

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Tópicos Especiais em Engenharia de Software

DADOS SEMANTICAMENTE ESTRUTURADOS NO SOFTWARE “PÉ NA ESTRADA” UTILIZANDO BANCO DE DADOS ORIENTADOS A GRAFOS Neo4J

Autor: Dandara Pereira Aranha, Ebenézer, Hugo Ferreira
Martins, Ítalo Paiva Batista

Orientador: Edgard Costa Oliveira

Brasília, DF

2015



Dandara Pereira Aranha, Ebenézer, Hugo Ferreira Martins, Ítalo Paiva Batista

DADOS SEMANTICAMENTE ESTRUTURADOS NO SOFTWARE “PÉ NA ESTRADA” UTILIZANDO BANCO DE DADOS ORIENTADOS A GRAFOS Neo4J

Trabalho submetido à disciplina Tópicos Especiais em Engenharia de Software: Web Semântica do curso de graduação em Engenharia de Software da Universidade de Brasília.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Edgard Costa Oliveira

Brasília, DF

2015

Listas de ilustrações

Figura 1 – Diferentes definições de ontologias	14
Figura 2 – Processos e atividades propostas pelo método Cyc	17
Figura 3 – Tabela abreviada do método Cyc	17
Figura 4 – Procedimentos propostos na metodologia de Gruninger e Fox	19
Figura 5 – Tabela abreviada da metodologia de Gruninger e Fox	19
Figura 6 – Estágios do método de Uschold e King	20
Figura 7 – Tabela abreviada do método de Uschold e King	21
Figura 8 – Tabela abreviada do método Kactus	22
Figura 9 – Estágios e atividades do ciclo de vida da ontologia	23
Figura 10 – Tabela abreviada da metodologia Methontology	23
Figura 11 – Tabela abreviada do método Sensus	24
Figura 12 – Processo de desenvolvimento de ontologias.	25
Figura 13 – Tabela abreviada do método 101	25
Figura 14 – Tabela abreviada da norma ANSI/NISO Z39.19-2005	26
Figura 15 – Tabela abreviada da metodologia proposta no manual da BITI	26
Figura 16 – Tabelas da base de dados da PRF	27
Figura 17 – Modelo de Dados atual das Ocorrências de Acidentes em Rodovias Federais	27
Figura 18 – Modelo de dados do <i>software</i> “Pé na Estrada”	28
Figura 19 – Arquitetura do <i>software</i> “Pé na Estrada”	29
Figura 20 – Interface gráfica do Protégé	31
Figura 21 – Método 101 para construção de ontologia	32
Figura 22 – Componentes da ontologia	36
Figura 23 – Componentes da ontologia CAOVA	36
Figura 24 – Conceitos levantados com o <i>CardSorting</i>	37
Figura 25 – Modelo conceitual da ontologia	38
Figura 26 – Classes modeladas no Protégé	43
Figura 27 – Propriedades das classes modeladas no Protégé	43
Figura 28 – Cronograma de Construção da Ontologia na Aplicação	44
Figura 29 – Gráfico de Custo Planejado	45
Figura 30 – <i>Burndown</i> das Atividades	45
Figura 31 – Tela de ranking das rodovias federais	51
Figura 32 – Tela de sinalização dos acidentes em uma rota	52
Figura 33 – Tela de sinalização do trecho mais perigoso de uma rota	53

Lista de tabelas

Tabela 1 – Classes e Propriedades	38
Tabela 2 – Atributos da classe Acidente	39
Tabela 3 – Atributos da classe Veículo	39
Tabela 4 – Atributos da classe Ocupante	40
Tabela 5 – Atributos da classe Local	40
Tabela 6 – Atributos da classe Rodovia	40
Tabela 7 – Atributos da classe Estatística	41
Tabela 8 – Atributos da classe Causa	41
Tabela 9 – Atributos da classe TipoAcidente	41
Tabela 10 – Atributos da classe PostoPRF	42
Tabela 11 – Atributos da classe PostoPRF	42
Tabela 12 – Atributos da classe Hospital	42

Sumário

1	INTRODUÇÃO	7
1.1	Contexto	7
1.2	Problema	7
1.3	Justificativa	8
1.4	Objetivos do Projeto	8
1.4.1	Objetivos Específicos	8
1.5	Metodologia	9
1.6	Cronograma	9
1.7	Resultados Esperados	9
2	REFERENCIAL TÉORICO	11
2.1	Evolução da <i>Web</i>	11
2.2	Classificações de Terminologias	12
2.3	Ontologias	14
2.4	Linguagens de ontologias	15
2.5	Metodologias para a construção de ontologias	16
2.5.1	Método Cyc	16
2.5.2	Metodologia de Gruninger e Fox	17
2.5.3	Método de Uschold e King	19
2.5.4	Método Kactus	21
2.5.5	Metodologia Methontology	22
2.5.6	Método Sensus	23
2.5.7	Método 101	24
2.5.8	Metodologia e Norma para construção de vocabulários controlados	25
3	SOFTWARE 'PÉ NA ESTRADA'	27
3.1	Dados abertos	27
3.2	Pé na estrada	28
3.2.1	Planejamento da aplicação da ontologia no "Pé na Estrada"	28
4	CONSTRUÇÃO DA ONTOLOGIA	31
4.1	Protégé	31
4.1.1	Características do Protégé	31
4.2	Metodologia utilizada para construção	32
4.3	Escopo da ontologia	35
4.4	Ontologias encontradas	36

4.5	Estrutura da ontologia	37
4.5.1	Classes e Propriedades	37
4.5.2	Atributos das classes	39
4.5.2.1	Acidente	39
4.5.2.2	Veículo	39
4.5.2.3	Ocupante	39
4.5.2.4	Local	40
4.5.2.5	Rodovia	40
4.5.2.6	Estatística	41
4.5.2.7	Causa	41
4.5.2.8	TipoAcidente	41
4.5.2.9	PostoPRF	41
4.5.2.10	AssistenciaTecnica	42
4.5.2.11	Hospital	42
4.5.3	Modelagem da ontologia no Protégé	43
4.6	Cronograma para Construção da Ontologia na Aplicação	44
4.6.1	Esforço versus Impacto da Construção	45
5	CONCLUSÃO	47
ANEXOS		49
ANEXO A – TELAS DO SOFTWARE PÉ NA ESTRADA		51
A.1	Tela de ranking das rodovias federais	51
A.2	Tela de sinalização dos acidentes em uma rota	52
A.3	Tela de sinalização do trecho mais perigoso de uma rota	53

1 Introdução

Esse capítulo aborda o contexto no qual o trabalho está inserido, apresentando o problema a ser resolvido e os objetivos a serem alcançados. A metodologia do trabalho e o cronograma também são apresentados.

1.1 Contexto

De acordo com o Sistema de Notificação de Mortalidade (SIM), somente em 2012, 48.812 pessoas morreram em acidentes nas rodovias brasileiras e entre as maiores causas estão: imprudência, ingestão de álcool, curvas perigosas, pistas mal conservadas e mal projetadas. Segundo a Polícia Rodoviária Federal (2013), 20 rodovias no Brasil causam mais de 100 óbitos por ano.

Com a Lei de acesso à informação (LEI 12.527/2011) (??), que garante aos cidadãos o acesso a informações acerca dos órgãos públicos integrantes da administração direta, a Polícia Rodoviária Federal (PRF) passou a disponibilizar dados relacionados às ocorrências dos acidentes em rodovias federais. Todavia, são apenas dados disponibilizados em planilhas. Desse modo, não sendo possível fazer interpretações desses dados de forma fácil, eficiente e eficaz.

A PRF possui um software que faz representações visuais estatísticas, todavia é destinado a uso interno e, atualmente (Dezembro, 2017), está fora do ar . Com o intuito de transformar esses dados em informações visualmente apresentáveis para os cidadãos, a aplicação “Pé na Estrada” utiliza-os para exibir informações estatísticas acerca dos acidentes, com a finalidade de diminuir a ocorrência dos acidentes em rodovias federais. Contudo, o modelo de dados atual está mal estruturado e apresenta inconsistências e informações incompletas, o que diminui a representatividade estatística do software e também reduz a capacidade do software de mostrar valor ao usuário, limitando suas funcionalidades a somente mostrar os acidentes numa rota, disponibilizar um ranking das rodovias em relação aos acidentes ocorridos e fazer comentários sobre uma rodovia.

1.2 Problema

Considerando o contexto abordado, os problemas encontrados são as inconsistências e complicações na estruturação atual dos dados sobre acidentes ocorridos em rodovias federais, dados estes que são disponibilizados pela PRF e que são utilizados atualmente pelo software “Pé na estrada”.

Alunos da disciplina Tópicos Especiais em Engenharia de Software, no primeiro semestre de 2015, se propuseram a resolver esse problema através criação de uma camada de dados semânticos utilizando o ActiveRDF. Entretanto o mesmo atualmente não é mantido mais por seus criadores e os alunos não obtiveram os resultados esperados com sua utilização.

1.3 Justificativa

De acordo com o planejamento estratégico da PRF ((??)), deseja-se aprimorar a percepção de segurança dos usuários das rodovias federais e o registro de ocorrências.

Para o cumprimento desse planejamento estratégico, a disponibilização do software “Pé na Estrada” para o uso dos cidadãos seria louvável. Mas para tal, é necessário estruturar os dados referentes aos acidentes em rodovias federais, de modo a aperfeiçoar o seu uso e analisar estatisticamente os acidentes ocorridos nas rodovias federais para propor melhorias relacionadas ao trabalho realizado.

Esse trabalho foi iniciado em 2015 por alguns alunos da disciplina de Tópicos Especiais em Engenharia de Software da Universidade de Brasília com a criação de uma ontologia específica para a PRF, e o planejamento e modelagem de um modelo de dados semântico para o software "Pé na Estrada". Nesse semestre deseja-se finalizar o trabalho com a implementação desse modelo.

1.4 Objetivos do Projeto

O objetivo geral é dar continuidade ao trabalho desenvolvido pelos alunos da da Diciplina de Tópicos Especiais em Engenharia de Software em 2015 com a otimização do software “Pé na estrada” criando de uma camada de dados semânticos utilizando o banco de dados orientado a grafos Neo4J,de modo que ele possa ser utilizado pela PRF para disponibilizar para a população uma ferramenta de suporte para utilização das rodovias.

1.4.1 Objetivos Específicos

- Estudar e entender a ontologia de acidentes de trânsito criada pelos alunos da disciplina de Web Semântica especificamente para a PRF baseada em ontologias já existentes;
- Otimizar o modelo de dados semântico do software "Pé na Estrada"para uma melhor representação dos dados sobre ocorrência de acidentes em rodovias federais disponibilizados pela PRF;

- Implementar uma camada de dados semânticos utilizando o banco de dados orientado a grafos Neo4J.

1.5 Metodologia

1.6 Cronograma

O Cronograma do Projeto pode ser visto na Figura ??.

Nome	Duração	Ínicio	Fim	Recursos
⊖ Iniciação	11d?	01/04/2015	15/04/2015	
Estabelecer Tema e Objeto de Estudo	4d?	01/04/2015	06/04/2015	Emilie Morais
Pesquisar e Documentar por Tesauros, Ontologias e Vocabulário	5d?	06/04/2015	10/04/2015	Filipe Feitosa
Conceber referencial teórico	5d?	08/04/2015	14/04/2015	Italo Paiva
Produzir versão inicial do projeto	4d?	10/04/2015	15/04/2015	Kleber Brito
Revisar versão inicial do projeto	2d?	14/04/2015	15/04/2015	Matheus Silva
⊖ Elaboração	15d?	09/04/2015	29/04/2015	
Revisar Plano de Trabalho	8d?	09/04/2015	20/04/2015	Emilie Morais
Produzir relação entre ontologias existentes e o projeto	5d?	20/04/2015	24/04/2015	Filipe Feitosa
Planejar escopo da intervenção nas Ontologias	5d?	20/04/2015	24/04/2015	Italo Paiva
Planejar impacto técnico na aplicação	3d?	27/04/2015	29/04/2015	Kleber Brito
Entrega 1: Versão Inicial do Projeto	1d?	29/04/2015	29/04/2015	Emilie Morais, Filipe Feitosa
⊖ Execução	34d?	30/04/2015	16/06/2015	
Escrever sobre a ferramenta escolhida	5d?	30/04/2015	06/05/2015	Matheus Silva
Escrever sobre as metodologias existentes	1d?	01/06/2015	01/06/2015	Kleber Brito
Escolher metodologia a ser adotada	5d?	25/05/2015	29/05/2015	Matheus Silva
Realizar modelagem conceitual	3d?	01/06/2015	03/06/2015	Italo Paiva
Mesclar modelo conceitual com as ontologias existentes	3d?	01/06/2015	03/06/2015	Filipe Feitosa
Fazer amostra de programação dos dados em RDF	2d?	03/06/2015	04/06/2015	Emilie Morais
Definir propriedades e restrições das classes	1d?	03/06/2015	03/06/2015	Italo Paiva
Aplicar Ontologias na ferramenta	6d?	08/06/2015	15/06/2015	Matheus Silva
Documentar na versão intermediária do projeto	7d?	08/06/2015	16/06/2015	Emilie Morais, Filipe Feitosa
Entrega 3: Versão Final do Projeto	1d?	17/06/2015	17/06/2015	

Figura 1 – Cronograma do Projeto

1.7 Resultados Esperados

Espera-se com esse projeto, fornecer à PRF um modelo de dados mais representativo para o domínio de ocorrências de acidentes, que provoque uma melhoria na disponibilização dos dados abertos.

Ainda que represente um esforço considerável num período de 2 meses, a implantação da ontologia representa um aumento considerável na usabilidade da ferramenta, fornecendo dados semanticamente relacionados, possibilitando inclusive futuras implementações de inteligência artificial para filtragem dos dados. Dessa forma, considera-se que os benefícios para aplicação da ontologia superam o custo inerente para implantação e mudança da ferramenta.

A equipe de projeto também acredita que o software “Pé na estrada” pode ser adotado como uma ferramenta de suporte a utilização das rodovias federais brasileiras para os usuários das mesmas.

2 Referencial teórico

Esse capítulo aborda alguns conceitos chave referentes ao trabalho, como: Evolução da Web, Ontologias e suas linguagens.

2.1 Evolução da Web

A internet como conhecemos hoje surgiu a partir de uma idealização, em meados dos anos 60, de uma rede de comunicação militar alternativa capaz de resistir a um possível conflito nuclear mundial. Desenvolvida por um grupo de programadores e engenheiros eletrônicos com o conceito de uma rede sem nenhum controle central onde cada computador seria apenas um “elo” na transmissão das informações (??).

Desde então, a Web cresceu em um ritmo assustador. Porém, grande parte das páginas disponíveis na Web hoje em dia ainda mantém as características da chamada Web 1.0.

WEB 1.0

Segundo Vicentim (??), é a internet como ela surgiu. Sites de conteúdo simples e estáticos onde as informações eram interligadas através de diversos diretórios de links relacionados para leitores humanos e não para máquinas e programas de computadores (software). Os computadores eram utilizados meramente para exibir informação na tela. O conteúdo da página “somente leitura” fazia do usuário um mero espectador.

WEB 2.0

O termo Web 2.0 (e consecutivamente, o Web 1.0) foi criado pelo especialista no setor Tim O'Reilly, classificando essa nova forma de utilizar a internet como uma “web como plataforma”.

“Surgiu por volta de 2004 e ao contrário da Web 1.0, é caracterizada pela interatividade e participação do usuário final com a estrutura e conteúdo da página, fazendo do usuário um contribuidor” (??).

Porém, com toda essa expansão, surgiu um grande problema: A dificuldade de lidar com o excesso de informações inúteis e/ou erradas. A Internet atual é denominada Web Sintática. As páginas da Web não contêm informações sobre si mesmas, ou seja, os computadores não identificam que tipo de conteúdo está disposto e a que assunto(s) a página se refere, deixando o processo de interpretação a serviço dos seres humanos. O que nos faz pensar: Por que os computadores não podem realizar esse trabalho? A partir disso, vêm a motivação para uma Web do Futuro, a Web 3.0.

WEB 3.0

O termo Web 3.0 foi criado pelo jornalista John Markoff, do New York Times, baseado na evolução do termo Web 2.0 criado por O'Reilly em 2004. Outras denominações desse mesmo momento são “Web Semântica” ou “Web Inteligente”.

A ideia da Web Semântica surgiu em 2001, quando Tim Berners-Lee, James Hendler e Ora Lassila publicaram um artigo na revista Scientific American, intitulado: “Web Semântica: um novo formato de conteúdo para a Web que tem significado para computadores vai iniciar uma revolução de novas possibilidades.”

A Web 3.0 pode ser definida como:

Uma internet onde teremos toda informação de forma organizada para que não somente os humanos possam entender, mas principalmente as máquinas, assim elas podem nos ajudar respondendo pesquisas e perguntas com uma solução concreta, personalizada e ideal. É uma internet cada vez mais próxima da inteligência artificial. É um uso ainda mais inteligente do conhecimento e conteúdo já disponibilizado online, com sites e aplicações mais inteligentes, experiência personalizada e publicidade baseada nas pesquisas e no comportamento de cada indivíduo (??).

2.2 Classificações de Terminologias

As ontologias podem ser classificadas em diversos aspectos, sobre o seu espectro semântico, quanto a sua generalidade e quanto ao tipo de informação que representam. Elas foram propostas por alguns estudiosos da área.

As ontologias são classificadas quanto a generalidade, tem a generalidade da ontologia como principal critério de classificação. Foram identificadas por Nicola Guarino do seguinte modo:

- Ontologias de nível superior;
- Ontologias de domínio;
- Ontologias de tarefas;
- Ontologias de aplicação.

As ontologias quanto ao tipo de informação que representam foram propostas por Assunción Gomez Perez, Mariano Fernandez Lopez e Oscar Corcho. Nelas o tipo de informação a ser modelado é muito importante (??). E foram classificadas em:

- Ontologias para a representação do conhecimento;

- Ontologias gerais e de uso comum;
- Ontologias de topo e de nível superior (upper ontologies);
- Ontologias de domínio;
- Ontologias de tarefas;
- Ontologias de domínio-tarefa;
- Ontologias de métodos;
- Ontologias de aplicação.

O principal tipo de classificação de ontologias (pelo menos se tratando dentro do âmbito da web semântica) é a classificação para ontologias segundo o seu espectro semântico, que foram propostas na literatura por Lassila e Deborah McGuiness. Esse tipo de classificação difere-se desde a mais “leve” (lightweight) até a mais “pesada” (heavyweight). E essa forma, segundo o seu espectro semântico, classifica-se baseada na estrutura interna e no conteúdo das ontologias (??) . Foram divididas em:

- Vocabulários controlados;
- Glossários;
- Tesauros;
- Hierarquias tipo-de-informais
- Hierarquia tipo-de-formais;
- Frames;
- Ontologias que exprimem restrições de valores;
- Ontologias que exprimem restrições lógicas.

Os vocabulários controlados consistem em uma lista de termos referentes à aplicação. Glossário é onde está exposta uma lista de termos, assim como os vocabulários controlados, só que em linguagem natural. Já os tesauros são de extrema importância, pois um tesauro reúne um conjunto de relacionamentos entre termos que estão organizados dentro de uma taxonomia.

Taxonomia é definida pelo dicionário Merriam Webster (*apud ??, ??*) como: “O estudo dos princípios gerais da classificação científica: classificação sistemática,11 em particular, classificação ordenada de plantas e animais segundo relacionamentos naturais”.

Assim, é a classificação de entidades de informação segundo a hierarquia que representam. Com esses conceitos em mente é possível definir tesauros como uma taxonomia adicionada a um conjunto de relacionamentos semântico entre seus termos. Tesauros são consistentes, pois podem servir como indexadores a várias bases de dados através de uma terminologia (??).

Já as Hierarquias de tipos (formais e informais) têm a sua relevância. Na primeira, os relacionamentos de generalização são respeitados integralmente. E nas de tipo informal, os relacionamentos de generalização são realizados de maneira informal, sendo essa a diferença principal.

Os frames são compostos por classes e propriedades. As primitivas de um modelo de frames são classes que possuem propriedades chamadas atributos ou slots. Cada frame oferece o contexto para modelar um aspecto de domínio. Já os atributos, ou slots, se aplicam somente na classe a qual foram definidas, e não em um aspecto global (??).

Já as ontologias que exprimem restrições lógicas permitem expressar restrições em lógica de primeira ordem. E em relação às ontologias que exprimem restrições de valores elas trabalham oferecendo subsídio para restringir os valores assumidos pelas propriedades da classe em questão (??).

Para este trabalho, será utilizado um Glossário de termos técnicos rodoviários, criado pelo Departamento Nacional de Estradas de Rodagem (??).

2.3 Ontologias

A linguagem está presente na nossa sociedade e na própria constituição do homem. Ela evolui diariamente e sua complexidade está diretamente associada com todas as construções do homem, incluindo a Web. Neste contexto de evolução da Web e tecnologia de informação, ontologia pode ser definida como “a ciência que estrutura e arranja sistematicamente unidades do conhecimento (os conceitos) de acordo com os elementos de conhecimento (características) comuns” (DAHLBERG, 2006). Essa, entretanto, pode ser considerada uma definição simplista, pois está enquadrada apenas no campo da ciência da informação, de forma que o conceito pode ser expandido em diferentes áreas do conhecimento (??) sem se contrapor, dependendo da sua aplicação.

No que tange ao contexto do nosso projeto, os conceitos aplicados na ciência da informação e de sistemas de informação são satisfatórios e sustentam a aplicabilidade das técnicas e frameworks adotados.

De forma geral, podemos admitir que as ontologias são uma especificação explícita de uma conceituação, ou seja, a definição explícita de conceitos e suas relações, propriedades e restrições expressas formalmente (??). Ela permite especificar uma visão abstrata

ÁREA DE APLICAÇÃO	DEFINIÇÃO
Inteligência Artificial	GUARINO (1997) define ontologia como uma caracterização axiomática do significado do vocabulário lógico
Sistemas de Informação	É definida como um conjunto de conceitos e termos que podem ser usados para descrever alguma área do conhecimento ou construir uma representação para o conhecimento (SWATOUT, 1999)
Linguagem e Cognição	A ontologia refere-se a tudo que existe no mundo composto por objetos, mudanças e relações entre eles. Pode ser baseada no mundo, na mente/intelecto, na cultura ou na linguagem (DAHLGREN, 1995).
Banco de Dados	Conhecimento genérico que pode ser reusado em aplicações de tipos diferentes (MEERSMAN, 2002)

Figura 2 – Diferentes definições de ontologias. Fonte: (??).

e simplificada de determinado assunto que se deseja abordar.

O conceito, por sua vez, é compreendido como uma “unidade de conhecimento” (DAHLBERG, 2006), uma entidade que alimenta a construção de vocabulários e tesouros, dando subsídio à criação de uma ontologia. Sua aplicação prática é compor as ontologias de classes e instâncias e mesmo relacionamentos.

2.4 Linguagens de ontologias

A ideia inicial de Tim Bernes Lee para a web semântica é uma arquitetura baseada em camadas, de forma que as camadas se sobreponham e se completem adicionando cada vez mais semântica aos dados, mas mantendo a escalabilidade (??). Dessa forma é possível representar semanticamente uma página web da maneira mais simples para uma máquina que não possua todo o arcabouço semântico necessário para visualizar todo o leque de opções, da mesma forma que é possível outra máquina com capacidade de interpretar as camadas semânticas superiores também visualizar da forma mais completa, semanticamente, a página requisitada.

A primeira camada dessa arquitetura escalar da web é a Hypertext Markup Language (HTML), que lida com o conteúdo da informação e sua apresentação para os seres humanos, fornecendo pouca expressividade para os dados e suas estruturas (??). Subindo mais um degrau da arquitetura web, a forma mais simples de representar semanticamente um documento é feita por meio da linguagem eXtensible Markup Language (XML), que é uma linguagem de marcação flexível que permite a criação de tags pelo usuário, o que possibilita criar uma representação semântica para cada pedaço da informação de um documento e para as relações entre esses pedaços, relacionando-os por meio de aninhamento das tags. O XML começou a se preocupar com a representação da estrutura do documento apresentado (??).

O Resource Description Framework (RDF) é uma linguagem declarativa que se

baseia no XML, fornecendo uma maneira padronizada de utilizá-lo. Uma das propostas do RDF é fazer com que os recursos da web sejam legíveis e acessíveis por máquinas ((??) 2005). Permite descrever sentenças sobre propriedades e relacionamentos entre itens na web, destacando três componentes para a sentença: recurso, propriedade e valor, que se assemelham ao sujeito, predicado e objeto de uma oração qualquer da língua portuguesa, respectivamente. O RDF conta com uma linguagem de suporte para descrever o vocabulário que será utilizado pelo documento RDF, o RDF-Schema. O RDF veio para adicionar uma camada mais semântica para a web, sobrepondo a camada já fornecida pelo XML.

O RDF e o RDF-Schema (RDFS, combinação entre as duas linguagens) representam a base para a web semântica. Apesar de várias linguagens para ontologias importantes terem sido propostas ao longo do tempo com base em extensões do RDFS (como SHOE, Oil, DAML, DAML+Oil), será descrita apenas a Web Ontology Language (OWL), que acabou representando um resultado final dos esforços das outras linguagens, e é hoje uma recomendação do W3C no que tange a linguagens ontológicas (??).

A OWL é uma linguagem que foi projetada para resolver problemas inerentes às aplicações da Web Semântica, permitindo a construção de ontologias, explicitação de fatos sobre um domínio específico e racionalizar sobre ontologias e fatos (??). O objetivo dessa linguagem é representar conceitos e seus respectivos relacionamentos como uma ontologia, e possui três linguagens com diferentes níveis de expressividade: OWL Lite, OWL DL e OWL Full.

2.5 Metodologias para a construção de ontologias

2.5.1 Método Cyc

CyC é um projeto do *Microelectronics and Technology Computer Corporation (MCC) em Austin, Texas*, que fornece uma base para o raciocínio de senso comum através do desenvolvimento de ontologias para uma ampla variedade de aplicações específicas de domínio. Todo o conhecimento em CyC é representado de forma declarativa na forma de afirmações em uma variante da lógica de primeira ordem chamado CYCL. A própria base de conhecimento CYC contém afirmações simples, motor de inferência pode ser usado para derivar novas afirmações que utilizam esta base de conhecimento.

As ontologias subjacentes CYC são organizados em conjuntos de módulos conhecidos como microteorias. Cada microteoria captura o conhecimento e o raciocínio necessário para algumas de domínio particular, como espaço, tempo, causalidade, ou agentes.

Podem existir múltiplos microteorias para um determinado domínio, refletindo as diferentes perspectivas e suposições feitas para pessoas modelar esse domínio. Neste sentido, CYC não é uma ontologia integrado monolítica, ao contrário, é uma rede de

microteorias para um conjunto de domínios cuja união abrange os diferentes compromissos da ontologia que podem ser feitas dentro desses domínios (??).

A Figura 2 ilustra os três processos que foram conduzidos no desenvolvimento da base de conhecimento Cyc em 1990 por Douglas Lenat e Ramanathan Guha (*apud* ??, ??).

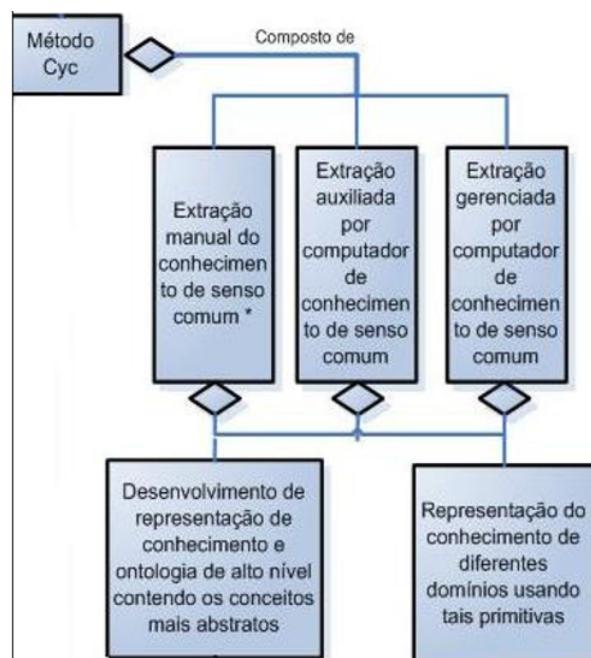


Figura 3 – Processos e atividades propostas pelo método Cyc. Fonte: (??)

Gerenciamento do projeto	Pré-desenvolvimento	Processos orientados				Pós-desenvolvimento	Processos integrais			
		Especificação de requisitos	Desenvolvimento				Integração	Avaliação	Documentação	
			Modelagem conceitual	Formalização	Implementação					
Ausente	Ausente	Simultânea à fase de implementação	Ausente	Ausente	Extração manual e codificação do conhecimento requerido. Codificação e extração do conhecimento com apoio de ferramentas computacionais como processamento de linguagem natural e aprendizado de máquina.	Ausente	Padrões ontológicos da indústria têm sido propostos no sentido de integrar sistemas computacionais à micro-teorias da base Cyc, como os padrões DAML, XML Schema e UML.	Ausente	Ausente	

Figura 4 – Tabela abreviada do método Cyc. Fonte: (??)

2.5.2 Metodologia de Gruninger e Fox

Esta metodologia baseia-se na experiência em desenvolvimento de ontologia do projeto Tove dentro do domínio de processos e atividades de negócios de modelagem. Essencialmente, ele envolve a construção de um modelo lógico de o conhecimento que está a ser especificado por meio da ontologia. Este modelo não é construído diretamente. Primeiro, uma descrição informal é feita das especificações para serem atendidas pela ontologia e, em seguida, esta descrição é formalizada. As medidas propostas são as seguintes:

1. Captação de cenários motivadores;
2. Formulação de perguntas informais de competência;
3. Especificação da terminologia da ontologia dentro de uma linguagem formal:
Obtendo terminologia informal;
Especificação da terminologia formal;
4. Formulação de questões formais de competência usando a terminologia da ontologia.
5. Especificação de axiomas e definições para os termos na ontologia dentro da linguagem formal.
6. Estabelecer condições para a caracterização da integralidade da ontologia.

Ontologias desenvolvida utilizando esta metodologia.

Aplicações utilizando ontologias desenvolvida com esta metodologia.

Empresa Projeto Workbench.

Abastecimento Integrado Cadeia Project Management agentes.

Análise da metodologia.(??)

A Figura 4 ilustra tal metodologia:

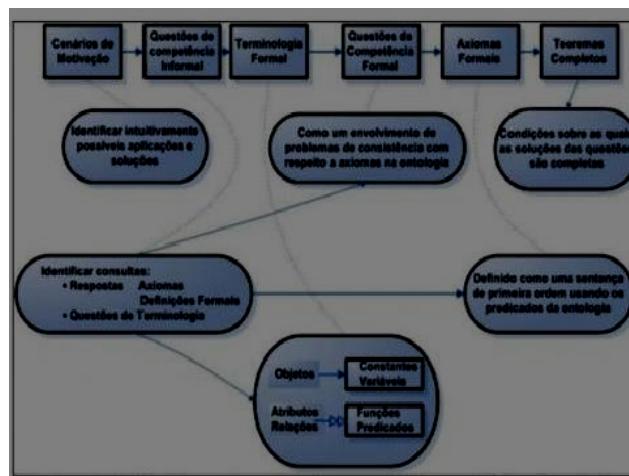


Figura 5 – Procedimentos propostos na metodologia de Gruninger e Fox. Fonte: (??)

Gerenciamento do projeto	Processos orientados					Processos integrais		
	Pré-desenvolvimento	Especificação de requisitos	Desenvolvimento	Pós-desenvolvimento		Integração	Avaliação	Documentação
Ausente	Descrição dos cenários de motivação.	Definição das questões de competência informal	Concepção da terminologia formal que servirá como linguagem para expressar as definições e as restrições dos axiomas. Pode ser dividida em dois passos: i) formalização das questões de competência a partir da terminologia formal concebida na fase de modelagem conceitual e ii) especificação dos axiomas formais para representar e resolver as questões de competência	As especificações dos axiomas em lógica de primeira ordem são implementadas na linguagem Prolog.	Ausente	Para integrar as ontologias envolvidas, a metodologia indica a formalização das diferentes perspectivas envolvidas de uma empresa para então integrá-las a ontologias de núcleo comum.	Os teoremas completos verificam se as soluções para as questões são completas.	Ausente

Figura 6 – Tabela abreviada da metodologia de Gruninger e Fox. Fonte: (??)

2.5.3 Método de Uschold e King

Uschold apresenta uma metodologia unificada, combinando as "melhores" partes de OE e Tove em um unificado método (apud Uschold, 1996). O primeiro passo é definir a finalidade da ontologia, ou seja, porque é a ontologia sendo construída. Isto pode ser feito de várias maneiras, por exemplo para identificar os utilizadores destinados, ou como no projeto com Tove, cenários de motivação e perguntas de competência, ou um documento de requisitos do usuário. Em seguida, o desenvolvedor deve decidir o nível de formalidade que a ontologia tem que ter. Na próxima fase, o desenvolvedor precisa encontrar os conceitos que deveriam estar na ontologia e as relações entre eles. Uschold prefere ir pelo means-out maneira ao definir termos e relacionamentos. Quando se trata de construir a ontologia o autor descreve quatro diferentes abordagens. O primeiro deles é pular as etapas anteriores e usar um editor de ontologias para definir os termos e axiomas (??).

Em segundo lugar, fazer os passos anteriores e, em seguida, iniciar uma codificação formal. A terceira abordagem é produzir um documento intermediário que consiste em

os termos e definições que apareceram na etapa anterior, este documento pode ser o resultado final, ou seja especificação do código formal ou documentação para ele ser. O quarto e aproximação final é identificar os termos formais a partir do conjunto de termos informais. A parte final presente no Uschold é o ciclo de avaliação ou a revisão, onde a ontologia desenvolvida é comparada com as questões da competência ou as necessidades dos utilizadores (??).

A Figura 6 apresenta tais estágios envolvidos na construção de uma ontologia:

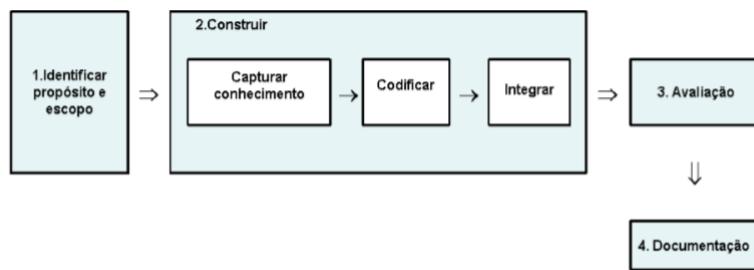


Figura 7 – Estágios do método de Uschold e King. Fonte: (??)

Gerenciamento do projeto	Pré-desenvolvimento	Processos orientados				Pós-desenvolvimento	Processos integrais		
		Especificação de requisitos	Modelagem conceitual	Desenvolvimento	Implementação		Integração	Avaliação	Documentação
Ausente	Ausente	Identificar a necessidade de construção, as classes de usuários e o uso pretendido da ontologia.	a) identificar os conceitos e relações chave no domínio de interesse; b) precisar primitivas textuais precisas e sem ambigüidades para os conceitos e relações; e c) identificar os termos com referência aos conceitos e relações de forma a conseguir um vocabulário consensual.	Ausente	Objetiva representar explicitamente a conceptualização "capturada" nas fases anteriores. Tal representação requer uma linguagem de representação formal como Prolog e Ontolingua, de maneira a criar o código.	Ausente	Verificar a possibilidade de integração com ontologias já existentes. Um método indicado é explicitar todas as pretensões subjacentes à ontologia de forma a alcançar uma correspondência semântica com a ontologia a qual se pretende integrar.	Pode ser feita através das especificações de requisitos, questões de especificidade, bem como da realidade imposta pelo mundo.	A documentação é definida de acordo com o propósito e escopo da ontologia. Todas as pretensões consideradas importantes da ontologia devem ser documentadas.

Figura 8 – Tabela abreviada do método de Uschold e King. Fonte: (??)

2.5.4 Método Kactus

Kactus é um método amplamente utilizado para o desenvolvimento de sistemas baseados em conhecimento, em que ontologias desempenham um papel importante. O projeto Kactus foi um projecto de acompanhamento que centrou-se na questão do desenvolvimento da ontologia. Uma abordagem de engenharia é adotada, salientando design modular, redesenhar e reutilizar. Uma ontologia é construída a partir de uma biblioteca de ontologias de pequena escala, que requer mapeamento entre as várias ontologias incluídas na desenvolvimento da nova ontologia. Dois tipos de funções de mapeamento são definidas entre o vocabulários de ontologias:

- (I) não há nenhuma mudança na semântica de expressões da ontologia mapeada.
- (II) uma alteração ocorre na semântica da ontologia mapeada como uma interpretação do que ocorre.

A seleção de ontologias relevantes a partir de uma biblioteca é suportado por um esquema de indexação para ontologias. Este caracteriza o contexto interpretação da utilização de uma ontologia segundo três dimensões: tipo de tarefa; método de resolução de problemas, e do tipo de domínio. O modular princípio de desenvolvimento é comum em engenharia ontológica e decorre do foco na reutilização. Embora o trabalho em funções de mapeamento está atualmente bastante limitado, a abordagem parece ter potencial em alfaiataria, ontologias para aplicação particular (??).

Gerenciamento do projeto	Pré-desenvolvimento	Processos orientados					Processos integrais		
		Desenvolvimento				Pós-desenvolvimento	Integração	Avaliação	Documentação
		Especificação de requisitos	Modelagem conceitual	Formalização	Implementação				
Ausente	Ausente	Lista de necessidades ou requisitos que precisam ser atendidos pela aplicação.	i) identificar os termos relevantes para o domínio da aplicação a partir dos requisitos; ii) construir um modelo preliminar, obtendo-se várias visões do modelo global; iii) refinar e estruturar a ontologia a fim de obter um modelo definitivo.	Ausente	Ausente	Ausente	Buscar por ontologias já desenvolvidas por outras aplicações no sentido de reutilização das mesmas. As ontologias reutilizadas demandariam refinamento e extensão para serem usadas na nova aplicação.	Ausente	Ausente

Figura 9 – Tabela abreviada do método Kactus. Fonte: (??)

2.5.5 Metodologia Methontology

A metodologia METHONTOLOGY é utilizada para construir ontologias a partir do zero (FERNANDÉZ et al., 1997 *apud* ,). O primeiro passo é especificar a finalidade da ontologia, o nível de formalidade e do escopo da ontologia. Em seguida, todo o conhecimento precisa ser coletado. Existem várias maneiras de fazer isso: através de *brainstorming*, entrevistas estruturadas e não estruturadas, análise formal e informal de textos e ferramentas de aquisição de conhecimento. Na fase de conceituação Fernández primeiro propos a construção de um glossário de termos de todo o conhecimento possivelmente útil no domínio determinado. Em seguida, os termos são agrupados de acordo com conceitos e verbos, e estes são reunidos para formar tabelas de fórmulas e regras. A próxima coisa a fazer é verificar se existem quaisquer ontologias já existentes que podem e devem ser usados. O resultado da fase de implementação é a ontologia codificada em uma linguagem formal, que pode ser avaliada (verificados e validados) de acordo com algumas referências. A parte final consiste na documentação (??).

Cada fase resulta em um documento que descreve a ontologia desenvolvida até agora. Durante o ciclo de vida temos a atividade de suporte. E nela se engloba a aquisição do conhecimento, documentação e avaliação conforme pode ser visto na figura 9:

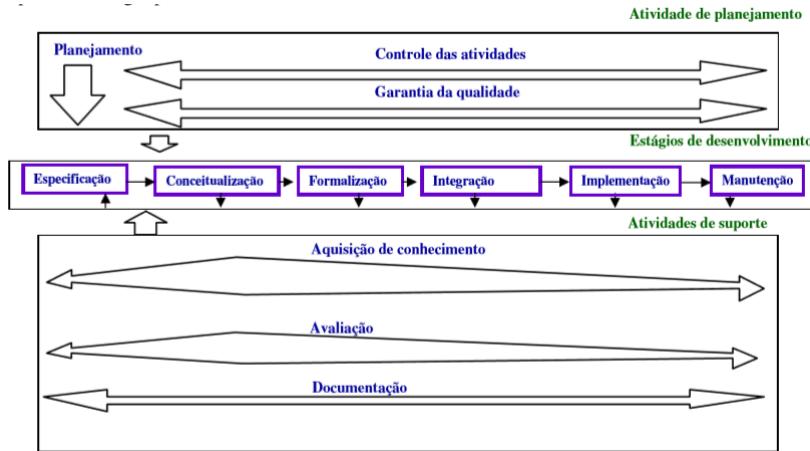


Figura 10 – Estágios e atividades do ciclo de vida da ontologia. Fonte: (??)

Gerenciamento do projeto	Pré-desenvolvimento	Processos orientados				Pós-desenvolvimento	Processos integrais			
		Desenvolvimento					Manutenção	Integração	Avaliação	
		Especificação de requisitos	Modelagem conceitual	Formalização	Implementação					
Planejamento das atividades; controle das tarefas e garantia da qualidade.	Ausente	Definição do propósito, do grau de formalização e do escopo.	Concepção da conceitualização através das representações intermediárias: i) glossário preliminar de termos; ii) árvores de classificação de conceitos; iii) diagramas de verbos; iv) dicionário de dados; v) tabelas de atributos de instâncias; vi) tabelas de atributos de classe; vii) tabela de constantes; viii) tabelas de fórmulas e regras; ix) árvores de classificação de atributos; e x) tabelas de instância.	Formalizar o modelo conceitual em uma linguagem formal como ontolingua e lógica descritiva. Entretanto, a formalização não é procedimento obrigatório, pois ferramentas como o ODE, por exemplo, já geram o código através da exportação da especificação da ontologia na linguagem de representação disponível.	Ambientes de desenvolvimento adequados à meta-ontologias e ontologias selecionadas na fase de integração. Uso de linguagens como Prolog, Ontolingua, dentre outras.	Orientação quanto a ajustes necessários nas representações intermediárias (ver tabela 16)	O resultado é um documento de integração que descreve, para cada termo aproveitado, o nome do termo e a definição do termo na meta-ontologia, o nome da meta-ontologia e o nome do termo na ontologia em construção.	Divide-se em verificação e validação da ontologia.	Recomendada em cada fase do ciclo de vida.	

Figura 11 – Tabela abreviada da metodologia Methontology. Fonte: (??)

2.5.6 Método Sensus

O método baseado na Sensus é uma abordagem top-down para derivar ontologias específicas de domínio de grandes ontologias. As etapas são as seguintes:

- 1) Uma série de termos são tomados como semente;
- 2) Esses termos de sementes estão ligados à mão para SENSUS;
- 3) Todos os conceitos no caminho dos termos de sementes para a raiz SENSUS estão incluídos;
- 4) Termos que poderiam ser relevantes no domínio e não têm ainda aparecido são adicionados;
- 5) Finalmente, para os nós que têm um grande número de caminhos através deles, toda a subárvore sob o nó é adicionado às vezes, baseada na idéia de que, se muitos dos gânglios em uma sub-árvore, foram encontrados para ser relevante, em seguida, os outros nós na

sub-árvore é provável que sejam relevantes. Esta metodologia utiliza a ontologia existente, onde a fusão será complexa devido a diferentes estruturas (??).

Gerenciamento do projeto	Pré-desenvolvimento	Processos orientados					Processos integrais		
		Especificação de requisitos	Modelagem conceitual	Desenvolvimento	Pós-desenvolvimento	Manutenção	Integração	Avaliação	Documentação
Ausente	Ausente	Ausente	i) identificar termos-chave do domínio; ii) ligar manualmente os termos-chave à ontologia SENSUS; iii) adicionar caminhos até o conceito de hierarquia superior da Sensus; iv) adicionar novos termos para o domínio; e v) adicionar subárvores completas.	Rede semântica	Utiliza a ferramenta Ontsaurus , a qual foi concebida para fins de desenvolvimento, gerenciamento e manutenção de bases de conhecimento, através da linguagem de programação LOOM .	Ausente	Ligar manualmente os termos-chave à ontologia SENSUS.	Ausente	Ausente

Figura 12 – Tabela abreviada do método Sensus. Fonte: (??)

2.5.7 Método 101

O método 101 foi criado por Natalya F. Noy e Deborah L. McGuinness com base em experiências no desenvolvimento das ontologias de alimentos e vinhos, utilizando para isso o conceituadíssimo software para a criação e edição de ontologias o Protégé-2000. Noy e McGuinness (2001, p.2) disseram em sua obra que não existe uma única maneira mais adequada para a construção de ontologias, e portanto não definiriam a melhor ontologia. A idéia desse método surgiu quando as autoras tiveram a idéia de compartilhar as suas experiências para a construção de ontologias, de modo a ser útil em algum momento para outras pessoas, em outros projetos. A origem do método segundo as autoras foram baseadas na literatura do POO (paradigmas orientados a objetos), contudo para o desenvolvimento de ontologias cada um deles, é claro, tem as suas particularidades, diferindo em alguns aspectos. (??) Para a execução do método 101, sete passos são considerados fundamentais no processo de construção da ontologia, conforme mostrado na figura 12.

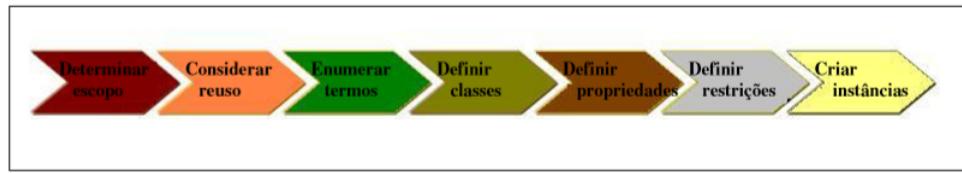


Figura 13 – Processo de desenvolvimento de ontologias. Fonte: (??)

Gerenciamento do projeto	Pré-desenvolvimento	Processos orientados				Pós-desenvolvimento	Processos integrais		
		Especificação de requisitos	Modelagem conceitual	Desenvolvimento	Implementação		Integração	Avaliação	Documentação
Ausente	Ausente	Determinar o domínio e o escopo da ontologia. Definição de questões de competência	i) enumerar termos; ii) definir classes; iii) organizar as classes em uma taxonomia; iv) definir slots e facetas; v) adicionar valores de slots para as instâncias.	Simultâneo à fase de modelagem conceitual através de uma linguagem de representação baseada em frames.	Baseada no módulo de frames ou módulo OWL da ferramenta Protégé.	Ausente	Considera a reutilização de ontologias com o propósito de: a) poupar esforço no desenvolvimento; b) interagir com as ferramentas que usam outras ontologias e c) utilizar ontologias validadas através do uso em aplicações.	Ausente	Na definição das classes e dos slots, a ferramenta Protégé possibilita fazer a documentação descrevendo-os em linguagem natural. Além disso, as premissas relevantes sobre o domínio também devem ser registradas como, por exemplo, lista de sinônimos e definições sobre limitações de escopo.

Figura 14 – Tabela abreviada do método 101. Fonte: (??)

2.5.8 Metodologia e Norma para construção de vocabulários controlados

O objetivo de vocabulários controlados é organizar as informações e para fornecer terminologia para catalogar e recuperar informações. Enquanto capturando a riqueza de termos de variantes, vocabulários controlados também promovem a coerência em termos preferenciais e da atribuição dos mesmos termos de conteúdo semelhante.

Dado que uma meta compartilhada da comunidade património cultural, é melhorar o acesso às artes visuais e informações de cultura material, vocabulários controlados são essenciais. Eles são necessários para a indexação fase porque sem eles catalogadores não vão usar consistentemente o mesmo termo para se referir à mesma pessoa, lugar ou coisa. No processo de recuperação, vários usuários finais podem usar diferentes sinônimos ou mais termos genéricos para referem-se a um dado conceito. Os usuários finais muitas vezes não são especialistas e, portanto, precisam para serem guiados porque eles podem não saber o termo correto.

As funções mais importantes de um vocabulário controlado são para reunir termos e sinônimos variantes para os conceitos e associar conceitos em uma ordem lógica ou classificá-los em categorias. Por exemplo: são um indicador de rosa e Catherine roda são a mesma coisa? Como está relacionado vidro pote metal para o vidro mais geral termo

marcado? As ligações e relações em um vocabulário controlado podem garantir que estas conexões são definidas e mantidas, tanto para catalogação quanto para recuperação (??).

Gerenciamento do projeto	Processos orientados						Processos integrais		
	Pré-desenvolvimento	Desenvolvimento				Pós-desenvolvimento	Integração	Avaliação	Documentação
		Especificação de requisitos	Modelagem conceitual	Formalização	Implementação				
Deveria ser parte integral de todo o projeto de construção do vocabulário controlado. Recomenda características necessárias aos sistemas de gerenciamento de vocabulários controlados.	Recomenda um estudo de viabilidade para a construção.	Ausente	Recomenda tratamentos em relação ao termo; organização das relações em hierárquica, associativa e de equivalência.	Recomenda tratar a representação dos relacionamentos dos termos.	Ausente	Recomenda procedimentos para a adição, a alteração e a exclusão de termos.	Ausente	Determina se o vocabulário controlado está fornecendo resultados de busca satisfatórios, que implica em uma boa relação entre precisão e revocação. Uma das tarefas da avaliação seria validar os termos antes de serem direcionados ao vocabulário controlado.	Recomendações quanto ao conteúdo da documentação.

Figura 15 – Tabela abreviada da norma ANSI/NISO Z39.19-2005. Fonte: (??)

Gerenciamento do projeto	Processos orientados						Processos integrais		
	Pré-desenvolvimento	Desenvolvimento				Pós-desenvolvimento	Integração	Avaliação	Documentação
		Especificação de requisitos	Modelagem conceitual	Formalização	Implementação				
Atividades de planejamento do projeto.	Ausente	Planejamento prévio à construção; i) delimitação da área; e ii) definição do público-alvo.	i) seleção de termos representativos do assunto; ii) definição dos termos; iii) agrupamento dos termos em categorias; e iv) organização das relações, que são classificadas em relação lógica e ontológica.	Tipos de exibição do tesouro, incluindo listas alfabéticas e visualizações gráficas, formatos impressos ou eletrônicos.	Ausente	A inclusão de novos termos deve ser proposta numa planilha específica.	Ausente	Critérios adotados para a adoção ou não de um tesouro existente.	Registro dos dados em planilhas, que irão possibilitar o registro do conhecimento sobre a concepção e a evolução do instrumento.

Figura 16 – Tabela abreviada da metodologia proposta no manual da BITI. Fonte: (??)

3 Software 'Pé na Estrada'

Esse capítulo traz informações sobre o *software* Pé na estrada, apresentando o modelo de dados fornecido pela PRF pela lei dos dados abertos, uma breve explicação das funcionalidades e estruturação atual do *software* e finalizando com as possíveis mudanças no *software* com a utilização da ontologia.

3.1 Dados abertos

Os dados são disponibilizados, semestralmente, em planilhas baseadas no modelo da Figura 17, e para obter dados específicos de cada acidente, as informações precisam ser correlacionadas entre tabelas, que podem ser vistas na Figura 16. As planilhas também contêm uma grande quantidade de dados, pois são divulgados cerca de 45.000 acidentes por semestre, além de uma quantidade exagerada de tabelas para o relacionamento, pouco eficiente, dos dados.

Nome das tabelas	Comentários de Colunas de tabelas
ocorrencia	É o registro de ocorrências confirmadas a partir das comunicações recebidas
ocorrenciaacidente	Cadastro de ocorrência envolvendo veículos.
ocorrenciapessoa	Cadastro de pessoas envolvidas no acidente.
tipoacidente	Qualifica os tipos de acidente
tipoenvolvido	Tabela que qualifica os envolvidos em uma ocorrencia de crime
tipounidadeoperacional	Qualificação dos tipos das unidades operacionais
tipoveiculo	Identifica os tipos de veículos
uf	Cadastro da UF onde foi feito a ocorrência.
unidadeoperacional	Qualificação dos tipos das unidades operacionais
Pessoa	Cadastro da pessoa envolvida na ocorrência.
ocorrenciaiveiculo	Cadastro da ocorrência referente ao veículo da pessoa da ocorrência.
Veiculo	Cadastro dos dados do veículo.
localbr	Armazena o local da OB onde aconteceu a ocorrência.
municipio	Cadastro do município da ocorrência.
Marcadeveiculo	Cadastro de marca de veículo.

Figura 17 – Tabelas da base de dados da PRF. Fonte: (??)

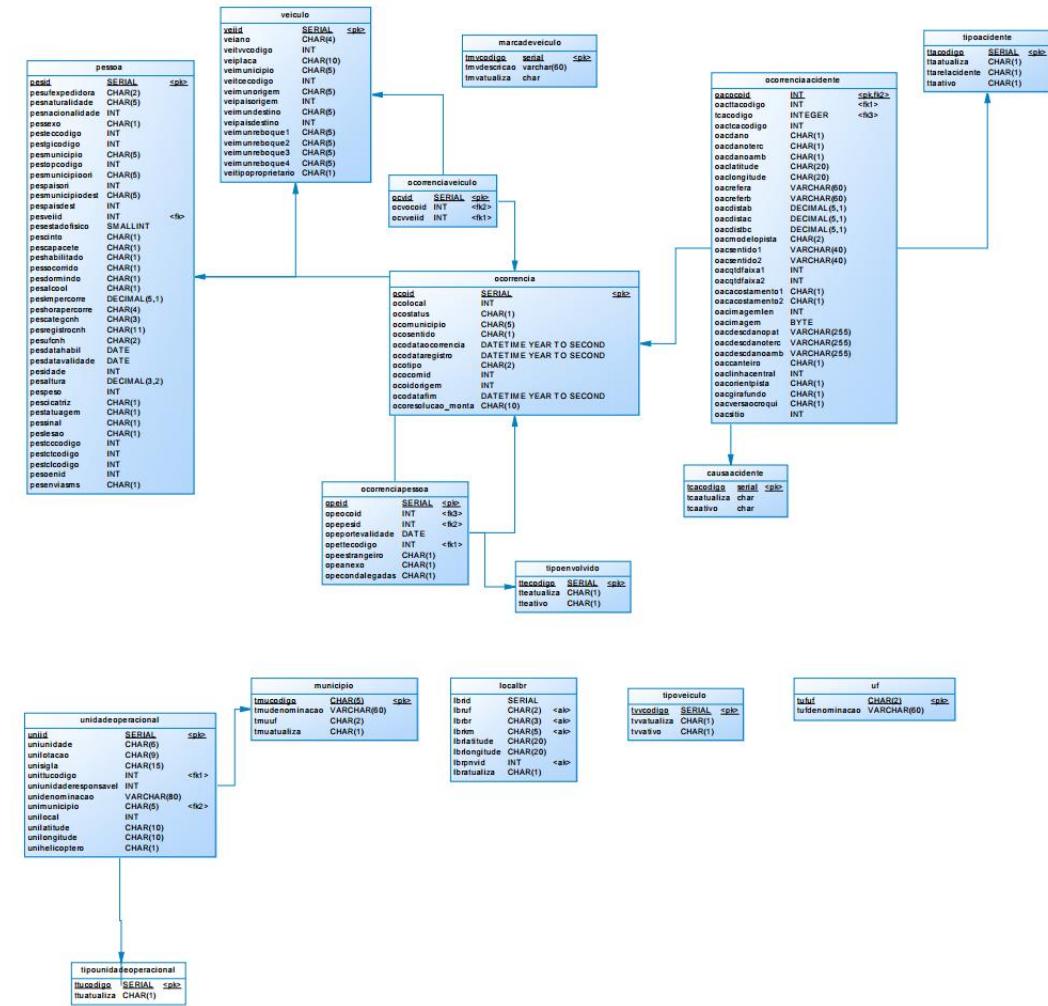


Figura 18 – Modelo de Dados atual das Ocorrências de Acidentes em Rodovias Federais.

Fonte: (??)

3.2 Pé na estrada

A disponibilização das ocorrências de acidentes em rodovias federais através da PRF, embora exista, não é realizada de forma visual. Todavia, consiste na disponibilização de planilhas complexas que contém muitos dados.

O software “Pé na estrada” foi desenvolvido, utilizando os dados divulgados pela PRF, com a finalidade de apresentar essas informações de forma mais atrativa e visual. Dessa forma, é um sistema que permite buscar informações sobre as rodovias federais, visualizando as rodovias com maiores índices de acidentes. A aplicação também permite traçar rotas para visualizar os trechos mais perigosos, e os acidentes ocorridos ao longo da rota. Há também a possibilidade dos usuários fazerem comentários sobre as rodovias. Algumas imagens do software podem ser vistas no Anexo A.

Os dados disponibilizados possuem muitas informações acerca do acidente, como tipo do acidente, dano causado à rodovia e tipo de veículo. Contudo, esse tipo de in-

formação não foi tratado na aplicação, pois esses dados estavam estruturados de forma complexa e inconsistente. Os relacionamentos entre os dados eram feitos por meio de uma grande quantidade de tabelas relacionadas, o que dificultaria a apresentação desses dados na aplicação, tornando o sistema lento e demasiado complexo de se manter. O modelo de dados utilizado na aplicação pode ser visto na Figura 18.

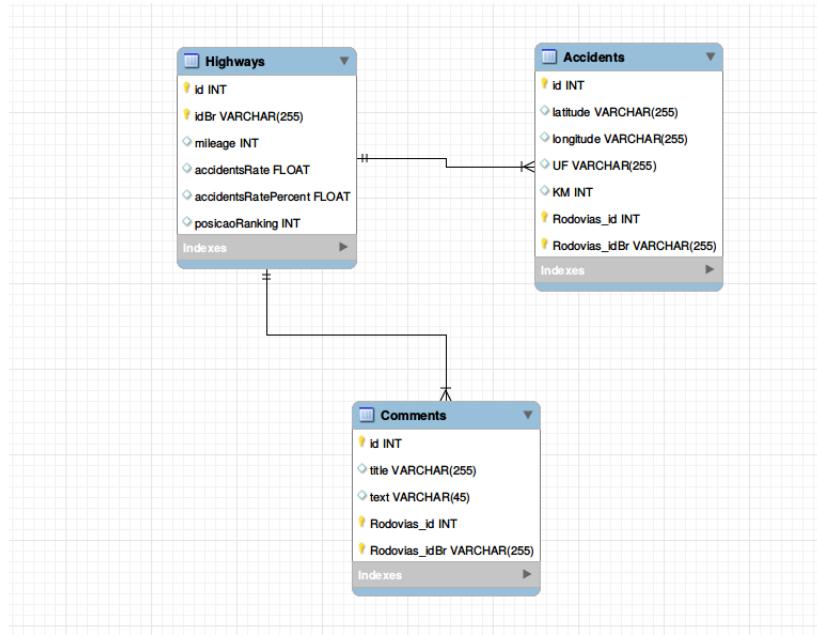


Figura 19 – Modelo de dados do software “Pé na Estrada”.

3.2.1 Planejamento da aplicação da ontologia no “Pé na Estrada”

Atualmente, o “Pé na estrada” disponibiliza informações apenas sobre os locais dos acidentes, e a quantidade de acidentes em cada rodovia. Esse tipo de informação seria mais relevante se acompanhassem informações acerca dos tipos de acidentes, danos causados a rodovia, tipo de veículo, envolvidos, entre outros dados que fossem relevantes para análise estatística e para representação gráfica.

Dessa forma, com o modelo de dados estruturado semanticamente, esses dados seriam encontrados e relacionados facilmente, possibilitando ao software disponibilizar essas informações ao usuário, até mesmo de forma mais visual.

Atualmente, o software não possui nenhuma camada semântica presente, possuindo apenas a camada visual do HTML, sendo necessária toda a estruturação das três principais camadas semânticas: o XML, o RDFS e a OWL, sendo esta última necessária para o uso da ontologia que será criada para representar os acidentes de trânsito.

O software “Pé na Estrada” foi desenvolvido na linguagem *Ruby* com a utilização do *framework Rails*. A partir disso, o sistema funciona com a arquitetura da Figura 19.

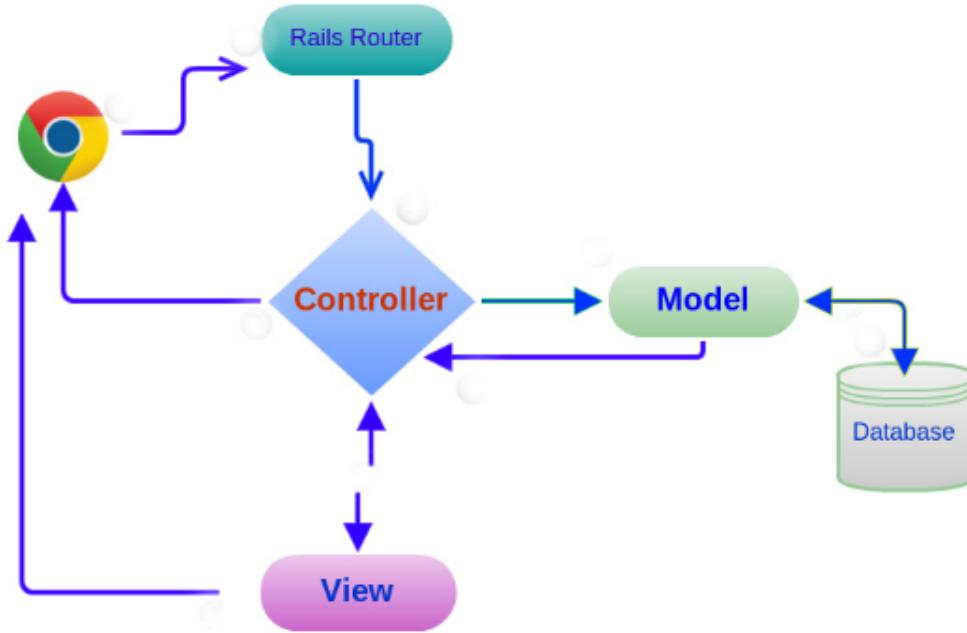


Figura 20 – Arquitetura do sistema. Fonte: Autores

O *Rails router* é responsável por estabelecer rotas que comunicam o sistema com o navegador. Essa rota é obtida na *Controller* que a partir disso gera a *View*. Os dados são obtidos e armazenados na base de dados através da *Model* que extende do *Active Record* que é a classe responsável por fornecer os meios de comunicação com a camada de dados.

Para fazer a comunicação da aplicação com a ontologia definida é necessário substituir a comunicação da *Model* que ao invés de estabelecer relação com a *Active Record* deve estabelecer relação com dados em RDF.

Existe um *framework* chamado *Active RDF*¹ que substitui a *Active Record* e estabelece um relacionamento do sistema com os dados contidos na ontologia.

¹ <http://www.activerdf.org/>

4 Construção da ontologia

4.1 Protégé

O Protégé é uma ferramenta que permite a construção de ontologias de domínio, como também, a personalização de formulários de entrada de dados e inserção e edição de dados, possibilitando assim, a criação de bases de conhecimento guiadas por ontologias. (??)

Para a construção da ontologia deste projeto, foi utilizada a versão (3.1) do Protégé. A Figura 20 ilustra a interface gráfica da ferramenta, que прôve acesso a ferramentas e menus, além de cinco áreas de vizualização (*views*) que se comportam como módulos de navegação e edição de classes, atributos, formulários e instâncias, propiciando entrada de dados, recuperação das informações e pesquisas na base de conhecimento. (??)

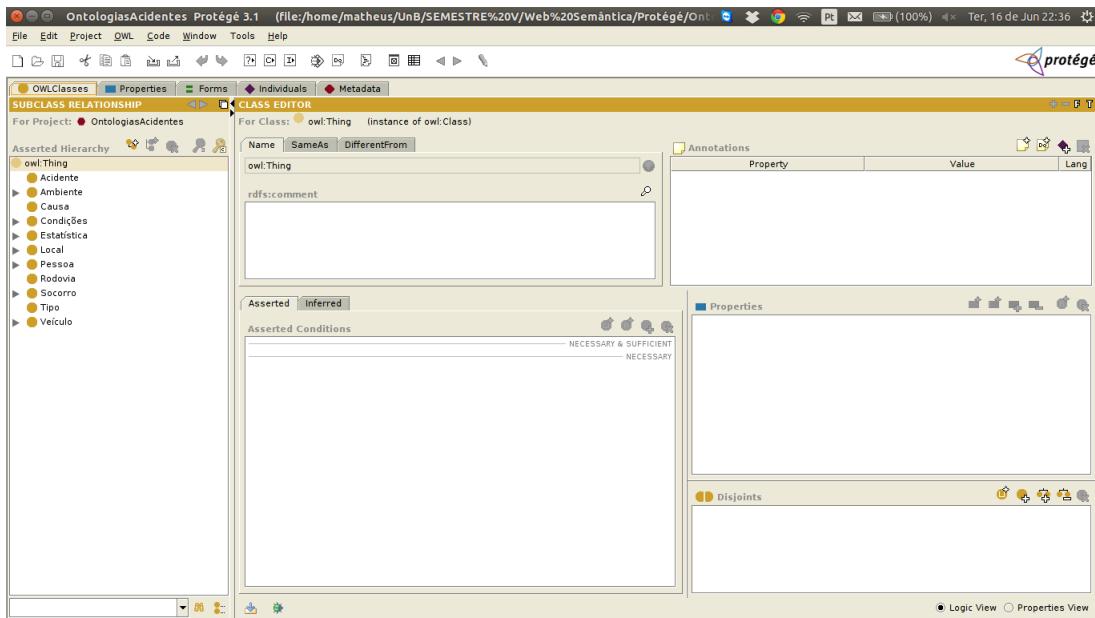


Figura 21 – Interface gráfica do Protégé

4.1.1 Características do Protégé

Esta seção apresenta algumas características da ferramenta Protégé que estão listadas abaixo: (??, ?? *apud* ??, ??).

- A linguagem PAL (*Protégé Axiomatic Language*, uma linguagem axiomática, permite a inserção de restrições e axiomas que incidem sobre classes e instâncias pertencentes a uma ou mais ontologias;

- A ferramenta provê a geração de arquivos de saída alteráveis, onde podem ser criadas classes e instâncias em CLIPS (motor de inferência) - a base de conhecimento é gerada nativamente para esse motor.
- Interface para entrada de conhecimento, com um gerador automático de formulários para classes definidas. A reposição da interface original por componentes mais adequados à aplicação é admitida. A interface da ferramenta facilita, sobretudo, o gerenciamento de conhecimento de uma ou mais ontologias.

Os arquivos de saída suportados são: Jess, F-Logic, Prolog, RDF, OIL, XML, Topic Maps. (??)

4.2 Metodologia utilizada para construção

A partir da análise conceitual dos métodos e metodologias para construção de ontologias abordados neste trabalho, o método escolhido para a realização deste projeto, levando em conta as etapas/fases propostas por cada uma dessas metodologias ou métodos no que tange ao processo de construção de ontologias, foi o método ***Ontology 101***.

Os aspectos observados na escolha do método foram: **a)** suporte literário bem consolidado e disponível (livros, artigos, guias e etc); **b)** o método é baseado na literatura do paradigma orientado a objetos; **c)** familiarização por parte dos integrantes do grupo com os conceitos e processos propostos pelo método.

Ontology 101.

De acordo com Breitman (??), o processo de construção de uma ontologia a partir do método 101, resumidamente, envolve as seguintes etapas:

- Definição das classes dessa ontologia;
- Disposição das classes em uma hierarquia taxonômica;
- Definição de propriedades e valores para os mesmos;
- Preenchimento dos valores das propriedades para cada instância.

A Figura 21 ilustra um esquemático do método 101 no qual é detalhado cada um dos seus sete passos.

1º Passo: Determinar o domínio e o escopo da ontologia

Para se determinar o domínio e o escopo da ontologia, Natalya F. Noy e Deborah McGuinness sugerem os seguintes questionamentos (??):

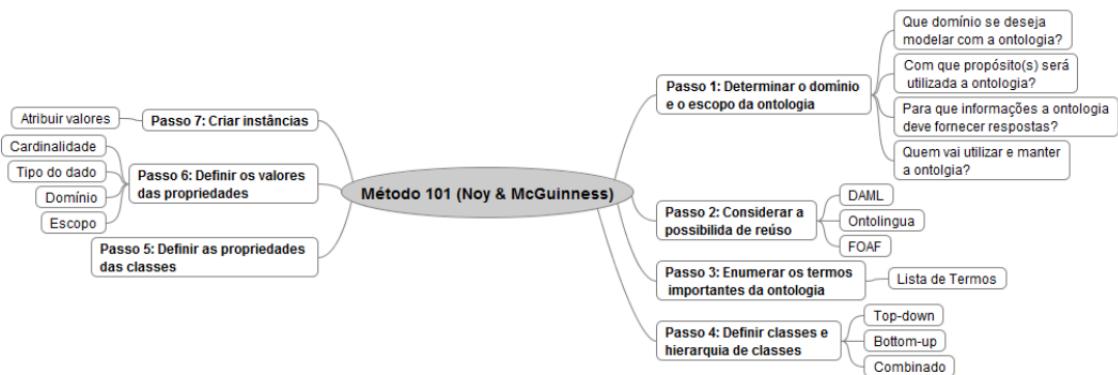


Figura 22 – Método 101 para construção de ontologias. Fonte: (??)

- Que domínio se deseja cobrir com a ontologia?
- Com que protótipo(s) será utilizada a ontologia?
- Para que informações a ontologia deve fornecer respostas?
- Quem vai utilizar e manter a ontologia?

Segundo Breitman (??), estas perguntas servirão para avaliar a ontologia após sua conclusão.

2º Passo: Considerar o reúso de outras ontologias

É importante considerar os termos que alguém já codificou em uma ontologia e se é possível refinar ou reutilizar modelos existentes para o domínio de nossa própria aplicação. Várias ontologias estão disponíveis eletronicamente, podendo ser facilmente importadas para editores e ambientes de desenvolvimento. (??)

Atualmente existem várias bibliotecas de ontologias que disponibilizam modelos para reúso. As bibliotecas do projeto Ontolingua, DAML e KACTUS, por exemplo, disponibilizam um grande número de ontologias que podem ser reutilizadas, refinadas ou adaptadas. (??)

3º Passo: Enumerar os termos importantes da ontologia

É útil fazer uma lista de todos os termos que deseja-se definir ou explicar aos usuários. Quais termos gostaríamos de abordar? Quais propriedades terão esses termos? O que gostaríamos de dizer sobre esses termos? É importante obter uma lista comprehensiva de termos sem se preocupar com a redundância entre os conceitos que eles representam, relacionamentos entre termos ou qualquer propriedade que eles venham a ter. (??)

4º Passo: Definir classes e a hierarquia de classes

Segundo Noy e McGuinness (??), definir as classes e a hierarquia entre elas e definir suas propriedades (5º Passo) são feitos de forma paralela, pois seria difícil fazer um após o outro. O curso natural, na prática, é definir uma classe e descrever suas propriedades logo após.

Existem muitas abordagens possíveis para se fazer uma hierarquia de classes: (??)

- A abordagem **topo-para-baixo** (*top-down*) inicia-se com a definição dos conceitos mais gerais do domínio (pai ou superclasse) e, posteriormente, esses conceitos são especializados em conceitos mais específicos (filhos ou subclasses). Essa abordagem é conhecida, também, como decomposição funcional.
- A abordagem **baixo-para-cima** (*bottom-up*), pelo contrário, inicia-se com a definição dos conceitos mais específicos (filhos ou subclasses) com o subsequente agrupamento dessas classes em conceitos mais gerais (pai ou superclasse). Esses agrupamentos são organizados de acordo com uma estratégia de generalização. (??)
- A **Combinação** (*middle-out*) é a conjunção das abordagens descritas anteriormente. Primeiramente, são definidos os conceitos mais notórios e então esses são especializados e generalizados adequadamente. (??)

De acordo com Noy e McGuinness (??), nenhuma das três abordagens é a melhor a ser utilizada, isso vai depender da visão pessoal do modelador do domínio. Geralmente, a **combinação** tende a ser a mais utilizada pelos desenvolvedores de ontologia, uma vez que os conceitos mais centrais são mais descriptivos no domínio da ontologia. (??)

5º Passo: Definir as propriedades das classes

Segundo Noy e McGuinness (??), as classes sozinhas não proveem subsídios suficientes para responder as questões de competência do 1º Passo (Determinar o domínio e o escopo da ontologia). Para isso é necessário criar uma estrutura interna de seus conceitos.

No passo anterior (Definir classes e a hierarquia das classes) já fora selecionadas as classes obtidas dos termos que foram listados no 3º Passo (Enumerar os termos importantes da ontologia). Muito dos termos remanescentes, provavelmente, representaram as propriedades das classes. Para cada propriedade deve-se determinar qual(ais) classe(s) este(s) descreve(m). (??)

De acordo com Noy e McGuinness (??), existem vários tipos de propriedades relativas a classes: intrínsecas, extrínsecas, partes, relacionamentos com classes e objetos, entre outras. Todas as subclasses ou filhos de uma classe (pai ou superclasse) herdam suas propriedades.

6º Passo: Definir os valores das propriedades

“Propriedades podem assumir diferentes valores, dependendo da expressividade da linguagem de ontologia que está sendo utilizada” (??). Um exemplo é a cardinalidade.

Alguns sistemas permitem cardinalidade única (um único valor) entre as propriedades e outros permitem cardinalidade múltipla (propriedades multivaloradas). (??)

Na linguagem OWL, por exemplo, é permitido utilizar tipos de dados no preenchimento dos valores das propriedades. (??)

Aqui está uma lista dos tipos de dados mais comuns:

- Cadeia de caracteres (*String*);
- Números (Às vezes valores mais específicos de ponto flutuante (*Float*) e inteiros são usados);
- Booleanos;
- Listas enumeradas de elementos.

7º Passo: Criar instâncias

O último passo é criar instâncias individuais das classes na hierarquia. Definir uma instância de uma classe consistem em: (1) Escolher a classe; (2) Criar uma instância individual dessa classe; (3) Preencher os valores de suas propriedades. (??)

4.3 Escopo da ontologia

A ontologia proposta neste trabalho será utilizada para o contexto de acidentes de veículos ocorridos em rodovias federais brasileiras, portanto, alguns conceitos inerentes a outros tipos de acidentes não se aplicam a este contexto. Por esse motivo, os modelos conceituais das ontologias encontradas foram analisados e utilizados como base para a construção da ontologia final.

Para este trabalho, é esperado apenas uma ontologia capaz de oferecer um arca-bouço mais semântico para o *software* "Pé na Estrada", de modo a provocar mudanças iniciais significativas no *software*. Para trabalhos futuros, pretende-se utilizar ontologias específicas para as entidades como Pessoa, Veículo, Rodovia e Hospital para aumentar a representatividade da ontologia, além de incluir atributos mais específicos para cada classe.

4.4 Ontologias encontradas

A fim de não ter retrabalho desnecessário, e de usar os conhecimentos previamente pesquisados por outros grupos de pesquisa, este grupo procurou em bases de periódicos e no próprio Google por ontologias que já existissem, e pudessem retratar de forma semântica os acidentes listados no software “Pé na Estrada”. As consultas muitas vezes não retornaram resultados satisfatórios, mas foi possível encontrar duas ontologias muito parecidas.

De posse das ontologias, é possível observar a similaridade entre as duas encontradas, até por que as duas são empregadas em soluções tecnológicas parecidas. Sua aplicação consiste na organização das informações de acidentes em estradas, para que as informações de um acidente de trânsito sejam devidamente transmitidas em redes veiculares Ad hoc, ou VANETs (??). Essas redes usam os carros como nós de uma rede para trafegar informações, que podem ser analisados e registrados em banco de dados. Na aplicação dos projetos encontrados ela também tem a utilidade de alertar ambulâncias, ou hospitais próximos que possam prestar socorro de forma rápida e eficiente. As duas ontologias possuem as mesmas entidades e seus relacionamentos são idênticos, evidenciando a robustez das mesmas. A diferença está nos atributos das entidades, com a adição de diversos campos que fornecem informações valiosas para a correta identificação do acidente de trânsito (??).

Desta forma, as duas ontologias são tomadas como base para o contexto deste projeto, pois se adequam muito bem dentro dos objetivos almejados. Será proposta a adição de uma entidade, de forma que seja possível também captar informações sobre a rodovia a qual o acidente ocorreu, diferentemente da entidade ambiente (“environment”), que procura detalhar as condições no momento do acidente. A entidade rodovia terá atributos que irão guardar as informações daquela rodovia, relacionadas ao contexto do projeto. O detalhamento dessa entidade será conduzido no trabalho seguinte.

As duas ontologias podem ser vistas na Figura 22 e na Figura 23.

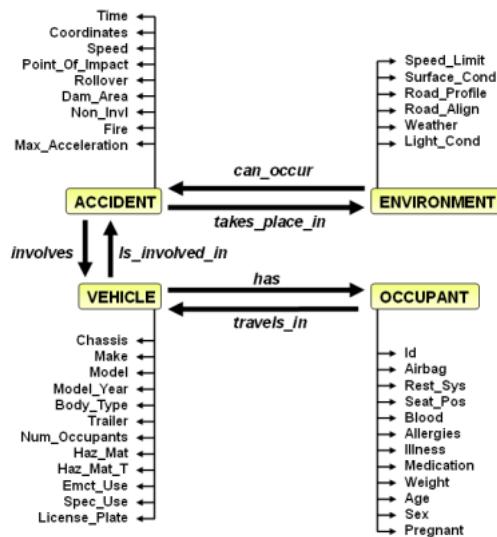


Figura 23 – Componentes da ontologia. Fonte: (??)

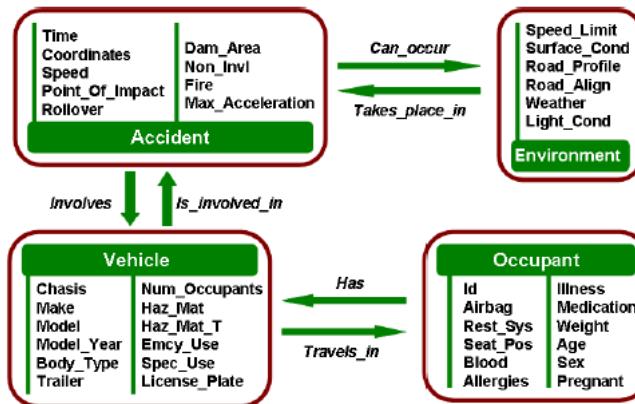


Figura 24 – Componentes da ontologia CAOVA. Fonte: (??)

4.5 Estrutura da ontologia

Com a escolha da metodologia 101 a primeira etapa consiste na definição de classes na ontologia. Para essa definição foi realizada uma técnica chamada *CardSorting*. A Figura 24 ilustra os conceitos levantados pela equipe com a técnica *CardSorting*.

Essa técnica consiste na escrita de termos em cartões que representem a interação do usuário com a aplicação, quais seriam as perguntas que ele faria ao sistema, o que ele procuraria. Dessa forma, pensando como usuário da aplicação foi realizado um levantamento de termos.

A partir do levantamento dos principais termos, foi realizado um mapa conceitual inicial com o intuito de organizar as classes em uma hierarquia e para identificar seus relacionamentos, que consiste na segunda etapa da construção.

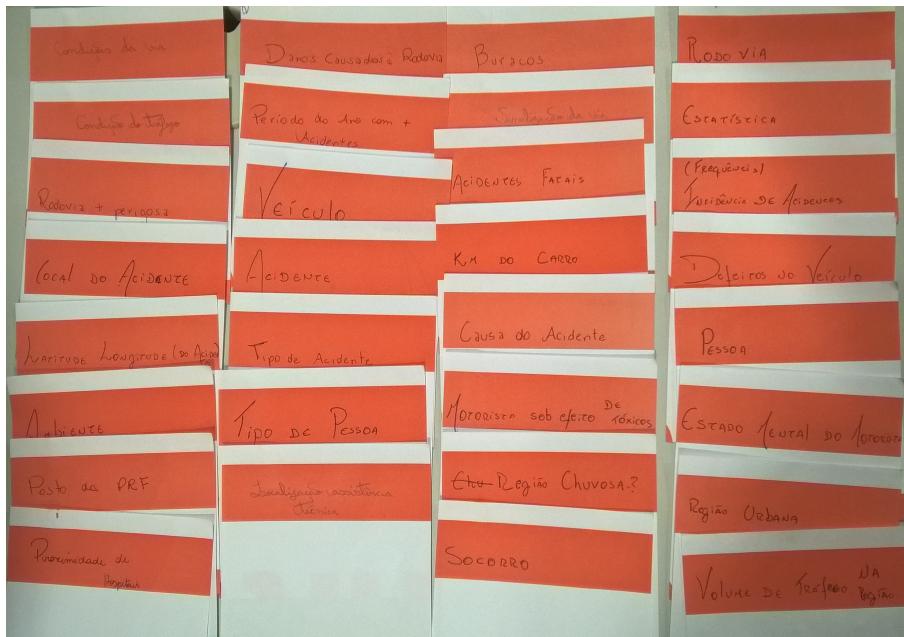


Figura 25 – Conceitos levantados com o *CardSorting*

4.5.1 Classes e Propriedades

Como não foi possível encontrar o código OWL das ontologias pesquisadas, a equipe decidiu aproveitar a modelagem conceitual feita nos trabalhos encontrados. A partir desses modelos, foi construído o modelo para a ontologia, ilustrado na Figura 25.

Após a elaboração do mapa conceitual inicial as propriedades entre as classes foram definidas e as ontologias encontradas foram utilizadas. Na tabela 1 está a lista de classes e as propriedades que elas possuem.

Os relacionamentos entre as classes podem ser vistos melhor no modelo conceitual da ontologia ilustrado na Figura 25.

Classe	Propriedade	Objeto (Classe)
Acidente	tem_tipo	TipoAcidente
	tem_causa	Causa
	ocorre	Local
	envolve	Veiculo
Veículo	esta_envolvido	Acidente
	tem_ocupante	Ocupante
Ocupante	viaja_em	Veiculo
	motorista	Veiculo
	proprietario	Veiculo
	atendido_em	Hospital
Local	tem_acidente	Acidente
	esta_contido	Rodovia
Rodovia	tem_estatistica	Estatistica
	contem	Local
	tem_postos_prf	PostosPRF
	tem_hospital	Hospital
Estatística	tem_assistencia	AssistenciaTecnica
	acerca	Rodovia
Causa	-	-
TipoAcidente	-	-
PostoPRF	-	-
AssistenciaTecnica	-	-
Hospital	-	-

Tabela 1 – Classes e Propriedades

O modelo conceitual apresentado se assemelha bastante com o padrão do grafo final que se obteria com a construção da ontologia.

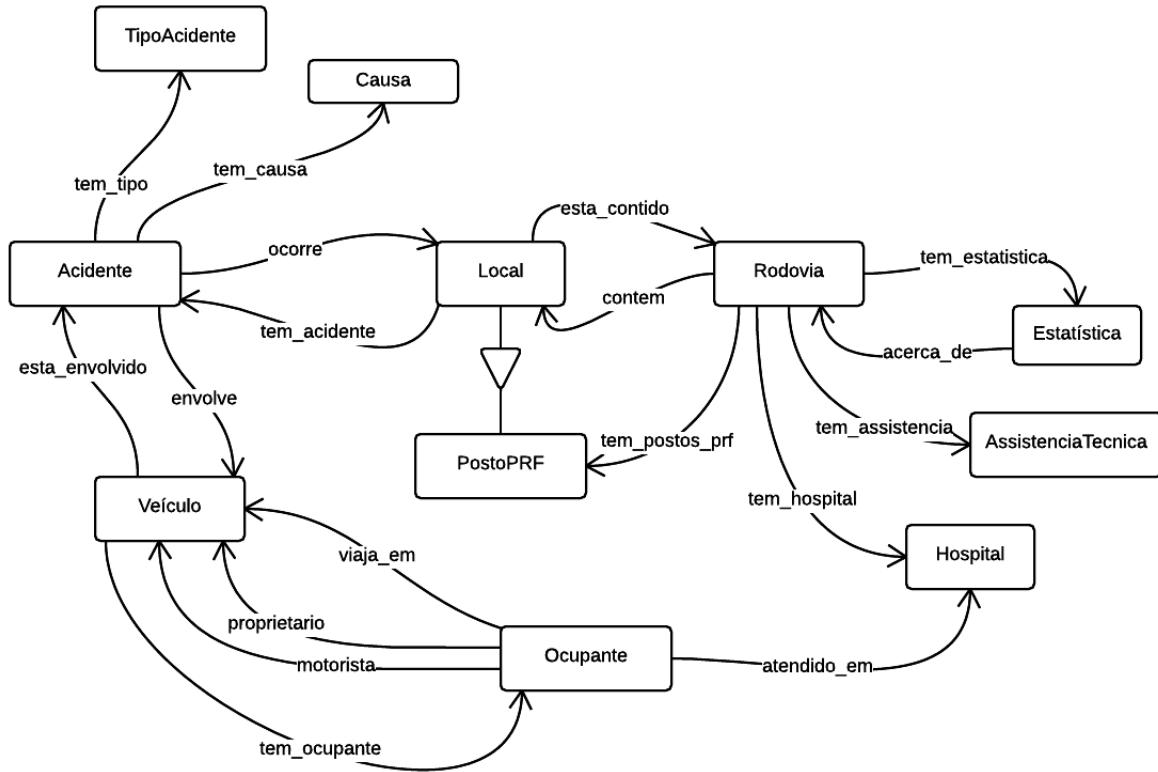


Figura 26 – Modelo conceitual da ontologia a ser construída

4.5.2 Atributos das classes

Apesar de não estar descrito no modelo conceitual apresentado, por questão de organização, as classes levantadas possuem atributos importantes que serão descritos nesse tópico.

Para a definição dos atributos das classes foram considerados alguns atributos definidos no modelo de dados atual da PRF.

4.5.2.1 **Acidente**

A classe Acidente pode ser considerada como a classe central para a aplicação e para a ontologia, uma vez que todas as outras classes estão relacionadas a um acidente ocorrido.

4.5.2.2 **Veículo**

A classe Veículo representa os veículos envolvidos em um acidente.

Classe	Atributo	Descrição
Acidente	data_acidente	Data da ocorrência do acidente
	hora_acidente	Horário da ocorrência do acidente
	danos_causados	Descrição dos danos causados pelo acidente

Tabela 2 – Atributos da classe Acidente

Classe	Atributo	Descrição
Veiculo	tipo_veiculo	Tipo do veículo (Ex.: Hatch, sedan, caminhonete)
	modelo_veiculo	Modelo do veículo
	ano_veiculo	Ano do veículo
	km_veiculo	Quilometragem do veículo no momento do acidente
	defeitos_veiculo	Defeitos encontrados no veículo no momento do acidente
	quantidade_pessoas	Quantidade de pessoas dentro do veículo no momento do acidente

Tabela 3 – Atributos da classe Veículo

4.5.2.3 Ocupante

A classe Ocupante representa as pessoas que podem estar dentro de um veículo no momento do acidente.

Classe	Atributo	Descrição
Ocupante	sexo	Modelo do veículo
	idade_pessoa	Idade da pessoa
	usava_cinto	Indica se o ocupante do veículo estava usando o cinto de segurança
	estado_mental*	Indica o estado mental do motorista no momento do acidente
	sob_efecto_toxicos*	Indica se o motorista estava sob efeito de tóxicos no momento do acidente
	uf_cnh_motorista*	UF da CNH do motorista

*Atributos específicos para instâncias com a relação 'motorista'.

Tabela 4 – Atributos da classe Ocupante

4.5.2.4 Local

A classe Local representa o local, dentro de uma rodovia, no qual um acidente ocorreu.

Classe	Atributo	Descrição
Local	latitude	Latitude em que ocorreu o acidente
	longitude	Longitude em que ocorreu o acidente
	km_rodovia	Indica em qual quilômetro da rodovia ocorreu o acidente
	ponto_referencia	Indica um ponto de referência próximo ao local do acidente
	regiao_urbana	Indica se a área do acidente é uma região urbana
	volume_trafego	Indica o volume de tráfego na região

Tabela 5 – Atributos da classe Local

4.5.2.5 Rodovia

A classe Rodovia representa as rodovias federais brasileiras. No domínio da aplicação, todo acidente ocorre em uma rodovia.

Classe	Atributo	Descrição
Rodovia	nome_rodovia	Nome da rodovia (Ex.: BR-060)
	extensao_rodovia	Indica o tamanho da rodovia
	estado_rodovia	Identifica quais estados brasileiros que a rodovia percorre
	posto_prf	Identifica um posto da PRF ao longo da rodovia

Tabela 6 – Atributos da classe Rodovia

4.5.2.6 Estatística

A classe Estatística representa as estatísticas que pode se inferir sobre os acidentes ocorridos em uma rodovia.

Classe	Atributo	Descrição
Estatística	nome_estatistica	Nome da estatística
	tipo_estatistica	Indica o tipo da estatística
	valor_estatistica	Identifica o valor da estatística
	desvio_padrao	Indica o desvio padrão da estatística calculada
	data_calculo	Indica a data em que foi calculada aquela estatística

Tabela 7 – Atributos da classe Estatística

4.5.2.7 Causa

A classe Causa representa as possíveis causas de um acidente. Para trabalhos futuros, pretende-se especializar esta classe nas diferentes causas de acidentes existentes

ou, se possível, utilizar uma ontologia de causas de acidentes.

Classe	Atributo	Descrição
Causa	descricao_causa	Descrição da causa do acidente
	tipo_causa	Indica o tipo da causa de acidente (Ex.: Falha técnica, condição da pista, etc.)
	periculosidade	Identifica o grau de periculosidade da causa

Tabela 8 – Atributos da classe Causa

4.5.2.8 **TipoAcidente**

A classe TipoAcidente representa os tipos de acidentes que podem estar relacionados a determinado acidente.

Classe	Atributo	Descrição
TipoAcidente	nome_tipo	Indica o nome do tipo de acidente
	descricao_tipo	Descrição do tipo de acidente

Tabela 9 – Atributos da classe TipoAcidente

4.5.2.9 **PostoPRF**

A classe PostoPRF representa os postos da PRF que podem estar localizados em uma rodovia. Os postos da PRF localizam-se ao longo das rodovias, o que os classificam como um tipo de local definido na classe Local. Logo, a classe PostoPRF possuem os mesmos atributos da classe Local, como latitude e longitude, e mais os seguintes:

Classe	Atributo	Descrição
PostoPRF	nome_posto	Indica o nome do posto da PRF (Ex.: 10º posto da Polícia Rodoviária Federal)
	encarregado_posto	Indica o nome do chefe responsável pelo posto da PRF
	telefone	Indica o número de telefone do posto da PRF
	email	Indica o endereço de <i>e-mail</i> do posto da PRF
	detalhes_posto	Informa detalhes sobre o posto da PRF (Ex.: Há local para pouso de aeronaves)

Tabela 10 – Atributos da classe PostoPRF

4.5.2.10 **AssistenciaTecnica**

A classe AssistenciaTecnica representa entidades que possam oferecer algum serviço de assistência aos motoristas, localizadas ao longo ou nas cidades mais próximas de uma rodovia.

Classe	Atributo	Descrição
AssistenciaTecnica	nome_assistencia	Indica o nome da unidade de assistência técnica
	tipo_assistencia	Indica o tipo de assistência oferecida (Ex.: Borracharia)
	endereco	Indica o endereço da unidade de assistência técnica
	telefone	Indica o número de telefone da unidade de assistência técnica

Tabela 11 – Atributos da classe PostoPRF

4.5.2.11 Hospital

A classe Hospital representa as unidades hospitalares localizadas próxima a uma rodovia. Para trabalhos futuros, pretende-se utilizar uma ontologia para a definição de hospitais.

Classe	Atributo	Descrição
Hospital	nome_hospital	Indica o nome do hospital
	especialidade	Indica qual a especialidade do hospital
	endereco	Indica o endereço do hospital
	telefone	Indica o número de telefone do hospital

Tabela 12 – Atributos da classe Hospital

4.5.3 Modelagem da ontologia no Protégé

As classes e propriedades definidas foram modeladas na ferramenta Protégé², de acordo com o modelo conceitual final apresentado na Figura 25.

As Figuras 26 e 27 apresentam, respectivamente, as classes e as propriedades que foram modeladas no Protégé.



Figura 27 – Classes modeladas no Protégé.

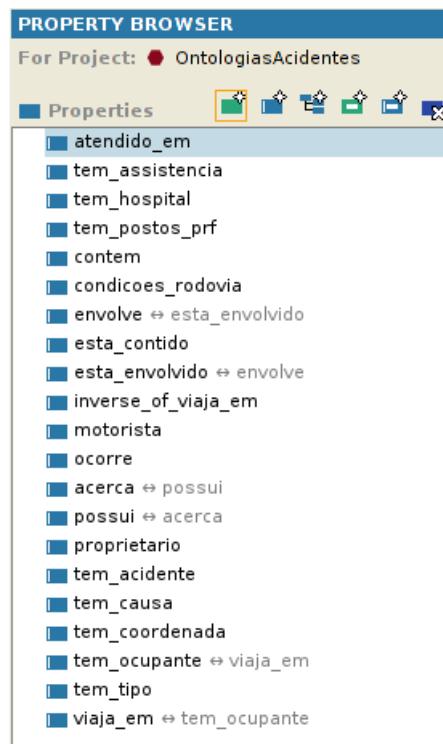


Figura 28 – Propriedades das classes modeladas no Protégé.

² <http://protege.stanford.edu/>

4.6 Cronograma para Construção da Ontologia na Aplicação

A aplicação da ontologia no software envolvem algumas mudanças estruturais na arquitetura e na refatoração de algumas funcionalidades, como uma busca no banco de dados por um acidente por exemplo. Desta forma, a equipe técnica definiu a construção da ontologia na aplicação como um projeto separado, com cronograma próprio, integrado com o cronograma geral do projeto. O objetivo é encarar a construção como uma mudança de requisitos, estudando o impacto e analisando as implicações do uso da ontologia dentro da ferramenta, com base nas métricas de qualidade previamente definidas.

		Nome	Duração	Ínicio	Fim	Predecessores
1		□Concepção	23d?	25/05/2015	24/06/2015	
2		Estudar implantação de Ontologia no Projeto	10d?	25/05/2015	05/06/2015	
3		Elaborar plano de avaliação do Impacto da mudança na aplicação	5d?	08/06/2015	12/06/2015	2
4		Validar especificação da Ontologia	1d?	17/06/2015	17/06/2015	
5		Elaborar Plano de Testes	7d?	15/06/2015	23/06/2015	3
6		Elaborar Projeto de implantação da mudança	5d?	18/06/2015	24/06/2015	3,4
7		□Desenvolvimento	23d?	22/06/2015	22/07/2015	
8		□Iteração 1	10d?	22/06/2015	03/07/2015	
9		Criar ambiente de teste	3d?	22/06/2015	24/06/2015	
10		Realizar deploy da aplicação no ambiente de teste	2d?	25/06/2015	26/06/2015	6
11		Desenvolver Módulo 1	5d?	29/06/2015	03/07/2015	10
12		Realizar Testes para Módulo 1	5d?	29/06/2015	03/07/2015	
13		□Iteração 2	5d?	06/07/2015	10/07/2015	
14		Testar Integração para Módulo 1	3d?	06/07/2015	08/07/2015	
15		Desenvolver Módulo 2	5d?	06/07/2015	10/07/2015	8
16		Realizar testes para Módulo 2	5d?	06/07/2015	10/07/2015	
17		□Iteração 3	8d?	13/07/2015	22/07/2015	
18		Testar Integração para Módulo 2	3d?	13/07/2015	15/07/2015	16
19		Desenvolver Módulo 3	5d?	13/07/2015	17/07/2015	13
20		Realizar Testes para Módulo 3	5d?	13/07/2015	17/07/2015	
21		Testar Integração para Módulo 3	3d?	20/07/2015	22/07/2015	
22		□Transição	12d?	23/07/2015	07/08/2015	
23		Validação: Realizar testes caixa preta na aplicação	5d?	23/07/2015	29/07/2015	21
24		Analizar métricas de qualidade	7d?	23/07/2015	31/07/2015	
25		Analizar impactos funcionais na aplicação	7d?	23/07/2015	31/07/2015	
26		Elaborar relatório final de análise para garantir a qualidade e funcionalidade	5d?	03/08/2015	07/08/2015	23,24,25

Figura 29 – Cronograma de Construção da Ontologia na Aplicação

4.6.1 Esforço versus Impacto da Construção

Com base no cronograma, é necessário observar que existe um grau de esforço que será empregado, com custos inerentes ao processo de software. A mudança mais relevante é o uso do framework *Active RDF*¹, no lugar do *Active Record* que faz deixa transparente o acesso de dados da camada *Model* com o banco de dados.

A hipótese é de que os benefícios da aplicação dessa mudança sejam relevantes a ponto de justificar o custo inerente à mudança do software. Para analisar a hipótese e refletir sobre suas implicações, foi necessário calcular os custos de produção. O valor considerado pela equipe técnica da hora de um aluno da UnB foi de dez reais e sessenta e cinco centavos, adicionados os valores de consumo de energia e mensalidade da internet. O valor do custo total foi baseado numa rotina de trabalho de 5 horas semanais, durante 58 dias, conforme a divisão de trabalho do cronograma.

O custo total da implementação da mudança na aplicação *Pé na Estrada* ficou em 3088,50 R\$. Esse valor, divido pelos 58 dias de trabalho, caracterizam 53,25 R\$ por dia de trabalho.

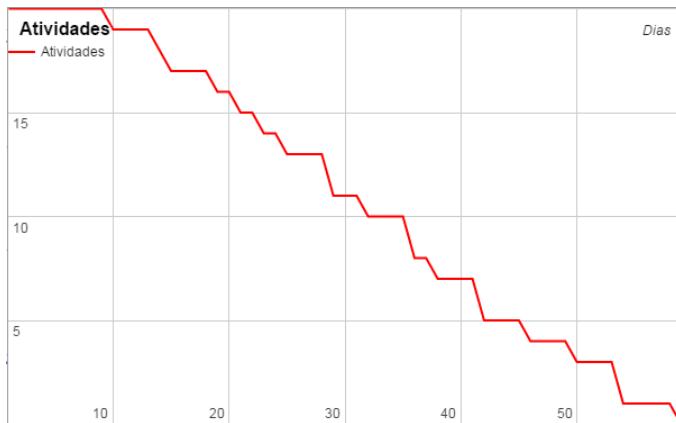


Figura 30 – Gráfico de Custo Planejado

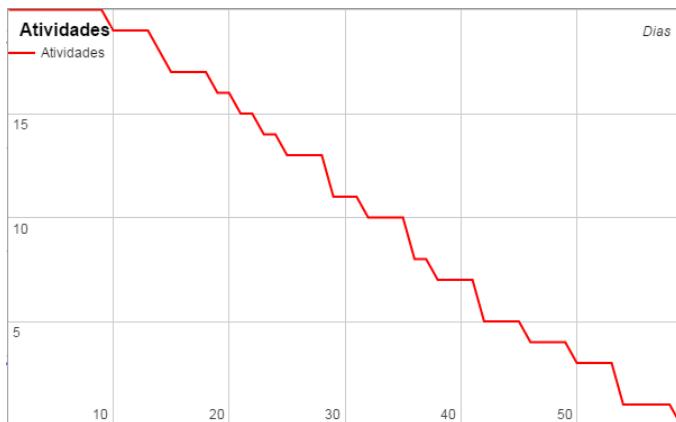


Figura 31 – Burndown das Atividades

A discussão sobre os resultados e sobre a hipótese levantada nesse seção serão discutidas no tópico **Resultados Esperados**.

5 Conclusão

A partir da realização deste trabalho foi possível conhecer sobre *Web Semântica* e suas potencialidades. Bem como, conhecer sobre a aplicação de ontologias e seu impacto no domínio de aplicações *Web*.

Embora as ontologias ainda não sejam amplamente utilizadas, já existem ontologias para vários domínios. No caso desse trabalho foram pesquisadas ontologias de acidentes e foi possível perceber que embora não se adequassem completamente ao contexto, seus conceitos puderam ser aproveitados.

A utilização da metodologia *ontology 101* permitiu a construção da ontologia em passos que resultaram na ontologia de acidentes para uso no “Pé na Estrada” com a reutilização de ontologias existentes.

Com a ontologia criada podemos amplificar a potencialidade do *software* “Pé na Estrada”, permitindo, por exemplo, pesquisas relacionando acidentes ocorridos em uma rodovia com as causas do acidente e com as condições da rodovia, ou pesquisar quais veículos estavam envolvidos em um acidente e quais pessoas estavam no veículo, possibilitando identificar o motorista, o proprietário e as demais pessoas no veículo e, ainda, é possível saber para qual hospital foram as pessoas envolvidas em um acidente. Todas essas novas possibilidades gerariam novas funcionalidades para o *software* de extremo valor para o usuário.

É possível perceber que com o uso de ontologias e *Web Semântica* possibilita o aumento da representatividade da aplicação, visto que expande o domínio do “Pé na Estrada” agregando mais valor ao usuário, no que se refere a quantidade de informações que o sistema fornece.

Anexos

ANEXO A – Telas do software Pé na Estrada

A.1 Tela de ranking das rodovias federais

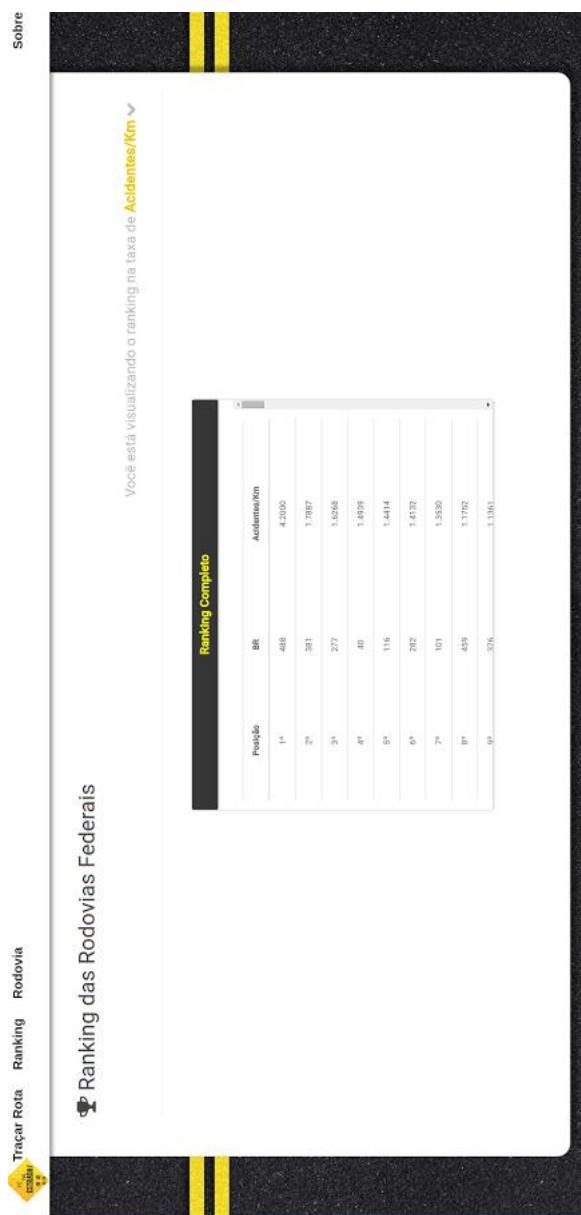


Figura 32 – Tela de ranking das rodovias federais.

A.2 Tela de sinalização dos acidentes em uma rota

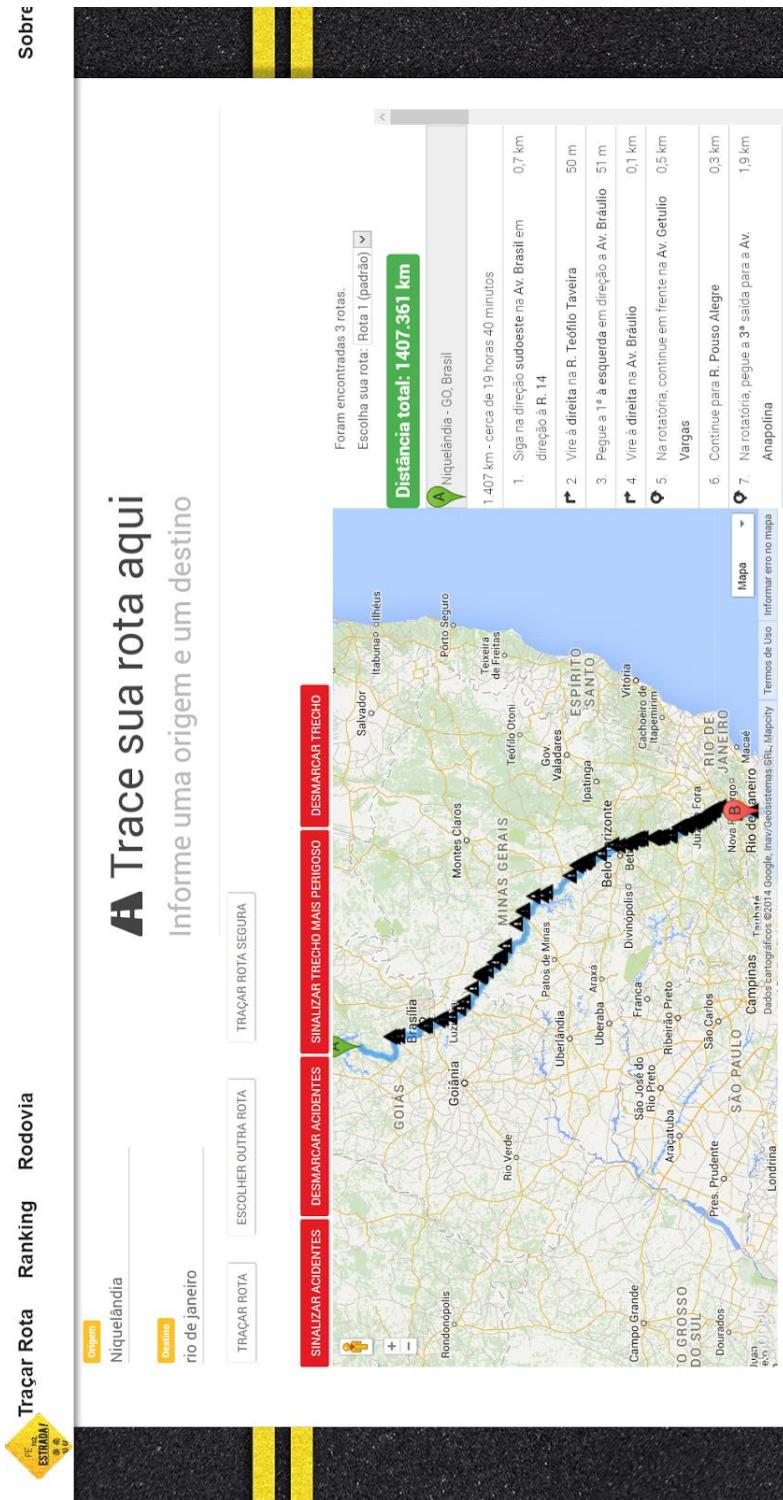


Figura 33 – Tela de sinalização dos acidentes em uma rota.

A.3 Tela de sinalização do trecho mais perigoso de uma rota



Figura 34 – Tela de sinalização do trecho mais perigoso de uma rota.