

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 Andrade, Hugo Ferreira Martins, Ítalo Paiva Batista

Orientador: Edgard Costa Oliveira

Brasília, DF

2017



Dandara Pereira Aranha, Ebenézer Andrade, 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

2017

Listas de ilustrações

Figura 1 – Etapas do trabalho	9
Figura 2 – Cronograma do Projeto	10
Figura 3 – Diferentes definições de ontologias	17
Figura 4 – Exemplo de grafo com matriz de adjacência	19
Figura 5 – Exemplo de grafo com lista de adjacência	19
Figura 6 – modelo de banco de dados orientado a grafos	20
Figura 7 – Processos e atividades propostas pelo método Cyc	21
Figura 8 – Tabela abreviada do método Cyc	21
Figura 9 – Processo de desenvolvimento de ontologias.	22
Figura 10 – Tabela abreviada do método 101	22
Figura 11 – Método 101 para construção de ontologia	23
Figura 12 – Tabelas da base de dados da PRF	27
Figura 13 – Modelo de Dados atual das Ocorrências de Acidentes em Rodovias Federais	28
Figura 14 – Modelo de dados do software “Pé na Estrada”	29
Figura 15 – Arquitetura do software “Pé na Estrada”	29
Figura 16 – Componentes da ontologia	33
Figura 17 – Componentes da ontologia CAOVA	33
Figura 18 – Conceitos levantados com o <i>CardSorting</i>	34
Figura 19 – Modelo conceitual ideal da ontologia	36
Figura 20 – Visão geral do banco de dados	43
Figura 21 – Exemplo de registros no banco de dados	44
Figura 22 – Tela de ranking das rodovias federais	49
Figura 23 – Tela de sinalização dos acidentes em uma rota	50
Figura 24 – Tela de sinalização do trecho mais perigoso de uma rota	51

Lista de tabelas

Tabela 1 – Classes e Propriedades	35
Tabela 2 – Atributos da classe Acidente	37
Tabela 3 – Atributos da classe Veículo	37
Tabela 4 – Atributos da classe Ocupante	37
Tabela 5 – Atributos da classe Local	38
Tabela 6 – Atributos da classe Rodovia	38
Tabela 7 – Atributos da classe Estatística	38
Tabela 8 – Atributos da classe Causa	39
Tabela 9 – Atributos da classe TipoAcidente	39
Tabela 10 – Atributos da classe PostoPRF	39
Tabela 11 – Atributos da classe PostoPRF	40
Tabela 12 – Atributos da classe Hospital	40

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	10
1.7	Resultados Esperados	10
2	REFERENCIAL TÉORICO	13
2.1	Evolução da <i>Web</i>	13
2.2	Classificações de Terminologias	14
2.3	Ontologias	16
2.4	Linguagens de ontologias	17
2.5	Grafos	18
2.6	Banco de Dados Orientado a Grafos	19
2.7	Neo4j	20
2.8	Metodologias para a construção de ontologias	20
2.8.1	Método Cyc	20
2.8.2	Método 101	21
3	SOFTWARE 'PÉ NA ESTRADA'	27
3.1	Dados abertos	27
3.2	Informações da aplicação	27
4	CONSTRUÇÃO DA ONTOLOGIA	31
4.1	Metodologia utilizada para construção	31
4.2	Escopo da ontologia	31
4.3	Ontologias encontradas	32
4.4	Estrutura da ontologia	33
4.4.1	Classes e Propriedades	34
4.4.2	Atributos das classes	36
5	APLICAÇÃO DA ONTOLOGIA NO SOFTWARE	41
5.1	Modelagem das entidades	41

5.2	Testes da nova modelagem	43
6	CONCLUSÃO	45
 ANEXOS		 47
	ANEXO A – TELAS DO SOFTWARE PÉ NA ESTRADA	49
A.1	Tela de ranking das rodovias federais	49
A.2	Tela de sinalização dos acidentes em uma rota	50
A.3	Tela de sinalização do trecho mais perigoso de uma rota	51

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

A metodologia deste trabalho consiste em seis etapas que podem ser vistas na Figura 1.

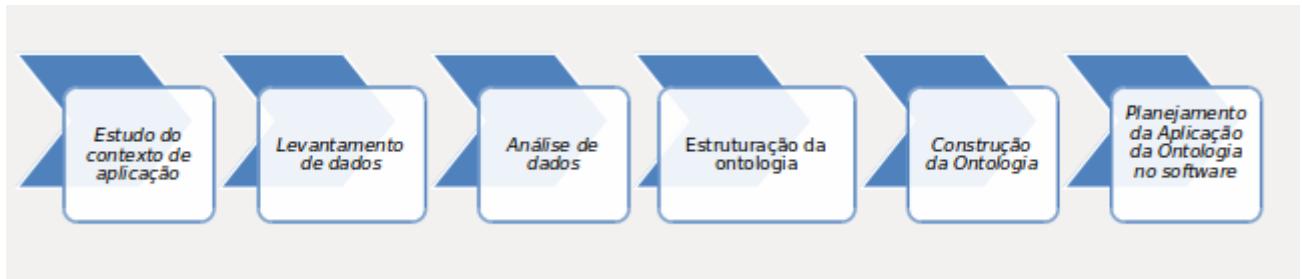


Figura 1 – Etapas do trabalho

Estudo do contexto de aplicação: essa etapa consiste na análise do contexto, dos usuários e do ambiente WEB ao qual a proposta será aplicada.

Levantamento de dados: essa etapa consiste na revisão de literatura e busca por ontologias e terminologias já existentes relacionadas a acidentes e aos dados das ocorrências no World Wide Web Consortium (W3C). O levantamento das ontologias ocorrerá por meio de duas expressões de busca, em sites de busca e em bases de dados, na língua inglesa, para obtenção de um resultado mais abrangente.

Expressão 1: “Car accident ontology”

Expressão 2: “Traffic accident ontology”

Análise de dados: essa etapa consiste na leitura dos artigos e verificação das ontologias encontradas para garantir que estejam alinhadas aos objetivos.

Estruturação da Ontologia: essa etapa consiste na elaboração de um modelo conceitual referente a ontologia.

Construção de Ontologias: essa etapa consiste na construção de uma ontologia de acidentes para o contexto do trabalho, baseada em ontologias existentes e na estruturação realizada.

Planejamento da aplicação da Ontologia no Software: essa etapa consiste na elaboração de um Planejamento de como a ontologia completa será finalizada e integrada ao “Pé na Estrada”

1.6 Cronograma

O Cronograma do Projeto pode ser visto na Figura 2.

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
Producir 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
Producir 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 2 – 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 implementação 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 3 – 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 Grafos

Grafos podem ser definidos como um conjunto de pontos relacionados entre si. Por exemplo, os pontos podem ser representados por pessoas, com linhas unindo pares de pessoas; ou pontos podem ser centros de comunicação que são relacionados por meio de *links* de comunicação. Uma abstração matemática de situações desse tipo originam o conceito de um grafo (??). Existem dois tipos de grafos: grafos com matriz de adjacência e grafos com lista de adjacência.

Grafo com matriz de adjacência: Neste tipo de grafo, cada vértice representa uma linha e/ou coluna de uma matriz de adjacência (??). A matriz 5x5 a seguir demonstra claramente esse tipo de grafo, pois cada linha ou coluna da matriz está sendo representada por uma das vértices do grafo.

Lista de adjacência: Neste tipo de grafo, cada vértice está associada a uma lista de vértices adjacentes (??). Cada posição de um dado vetor com n posições aponta para uma lista de adjacências. Como demonstra o exemplo a seguir:

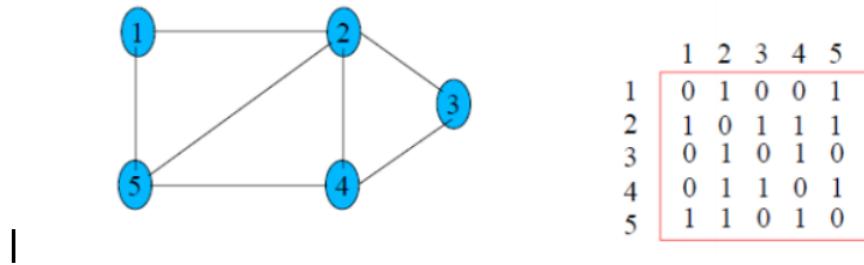


Figura 4 – Exemplo de grafo com matriz de adjacência. Fonte: (??)

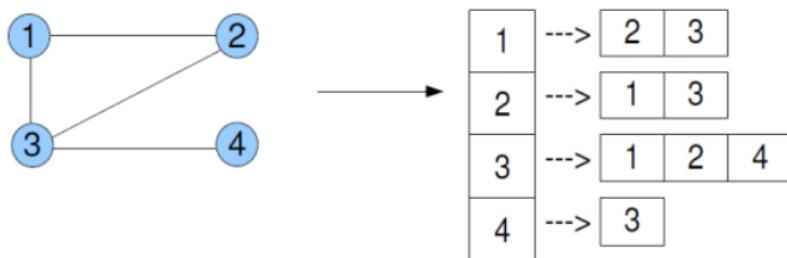


Figura 5 – Exemplo de grafo com lista de adjacência. Fonte: (??)

2.6 Banco de Dados Orientado a Grafos

O banco de dados orientado a grafos tem os mesmos conceitos gerais e simples dos bancos de dados relacionais, onde os pontos representam as classes e os objetos e as arestas representam os relacionamentos ou propriedades que podem existir entre esses objetos e classes (??).

Segundo ?? (??), a estrutura de um banco de dados orientado a grafos é a mesma de um banco de dados relacional, a diferença é que o seu esquema é representado por um grafo.

A ideia chave em um banco de dados orientado a grafos é a mesma de um banco de dados relacional. A instância pode ser vista conceitualmente como um grafo onde cada objeto e cada valor é representado por um nó exclusivo. Além disso, cada um desses nós, é rotulado por um nome de uma classe, semelhante ao modelo de banco de dados relacional. E por fim, as arestas representam os múltiplos relacionamentos ou propriedades existentes entre os objetos (??).

A imagem ilustra a representação de um modelo de banco de dados orientado a grafos modelado na ferramenta Neo4j.

Labeled Property Graph Data Model

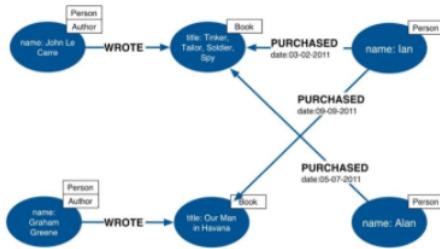


Figura 6 – modelo de banco de dados orientado a grafos. Fonte: (??)

2.7 Neo4j

Neo4j é um banco de dados orientado a grafos baseado em JVM. Como líder em banco de dados orientado a grafos, seu modelo é intuitivo e expressivo. Para dados altamente conectados, NEO4J é milhares de vezes mais rápido do que os bancos de dados relacionais, tornando-se ideal para gerenciar dados complexos em vários domínios, sejam eles financeiros, sociais, telecomunicações ou geoespaciais (??).

O Neo4j pode ser usado para fornecer respostas que são muito complicadas para bancos de dados relacionais, incluindo o fornecimento de diagramas para mostrar como as tabelas SQL se relacionam entre si e a cadeia mais curta de relações entre duas tabelas (??).

2.8 Metodologias para a construção de ontologias

2.8.1 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 7 ilustra os três processos que foram conduzidos no desenvolvimento da base de conhecimento Cyc em 1990 por Douglas Lenat e Ramanathan Guha (*apud* ??, ??).

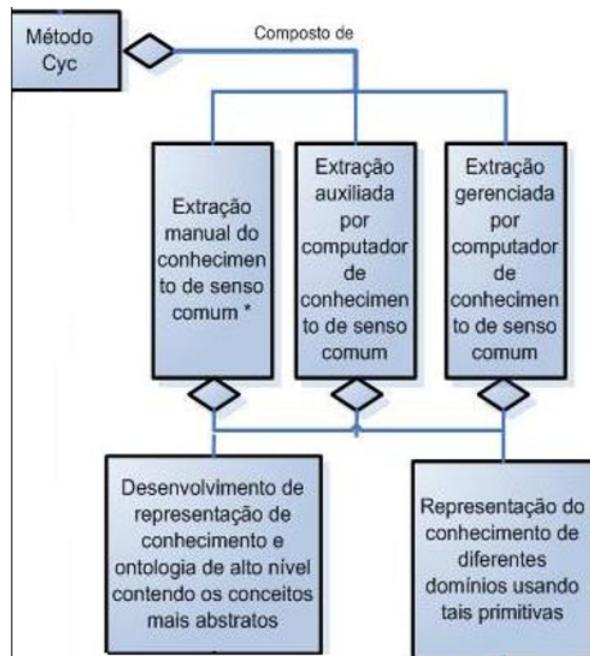


Figura 7 – Processos e atividades propostas pelo método Cyc. Fonte: (??)

Gerenciamento do projeto	Pré-desenvolvimento	Processos orientados					Pós-desenvolvimento	Processos integrais		
		Especificação de requisitos	Modelagem conceitual	Formalização	Implementação	Mantenção		Integração	Avaliação	Documentação
Ausente	Ausente	Simultânea à fase de implementação	Ausente	Ausente	Extração manual e codificação do domínio requerido. Codificação e extração do conhecimento com apoio de ferramentas computacionais como processamento de linguagem natural e aprendizado de máquina.	Ausente	Padrões tecnológicos de integração 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 8 – Tabela abreviada do método Cyc. Fonte: (??)

« « < HEAD =====

» » > 5c6b5878525c3b27d1ced18d8167b2e1280ec975

2.8.2 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 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 9.

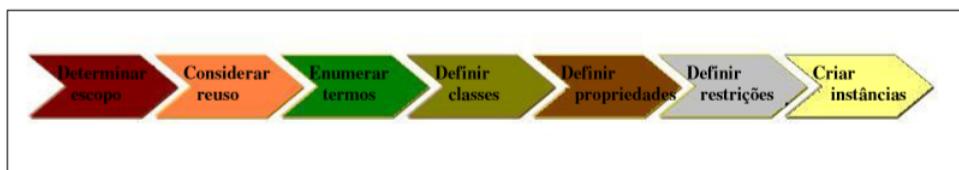


Figura 9 – Processo de desenvolvimento de ontologias. Fonte: (??)

Gerenciamento do projeto	Pré-desenvolvimento	Processos orientados				Processos integrais		
		Especificação de requisitos	Modelagem conceitual	Desenvolvimento	Pós-desenvolvimento	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 10 – Tabela abreviada do método 101. Fonte: (??)

« « « < HEAD =====

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.

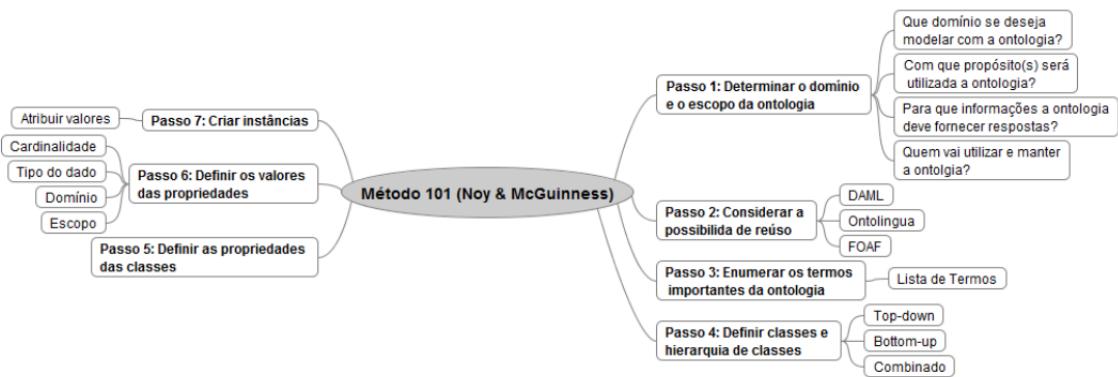


Figura 11 – Método 101 para construção de ontologias. Fonte: (??)

A Figura 11 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 (??):

- 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 compreensiva 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 descritivos 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 foram selecionadas as classes obtidas dos termos que foram listados no 3º Passo (Enumerar os termos im-

portantes 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. (??)

»»»> 5c6b5878525c3b27d1ced18d8167b2e1280ec975

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*.

3.1 Dados abertos

Os dados são disponibilizados, semestralmente, em planilhas baseadas no modelo da Figura 13, e para obter dados específicos de cada acidente, as informações precisam ser correlacionadas entre tabelas, que podem ser vistas na Figura 12. 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.
ocorrenciaapessoa	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.
ocorrenciaveiculo	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 12 – Tabelas da base de dados da PRF. Fonte: (??)

3.2 Informações da aplicação

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.

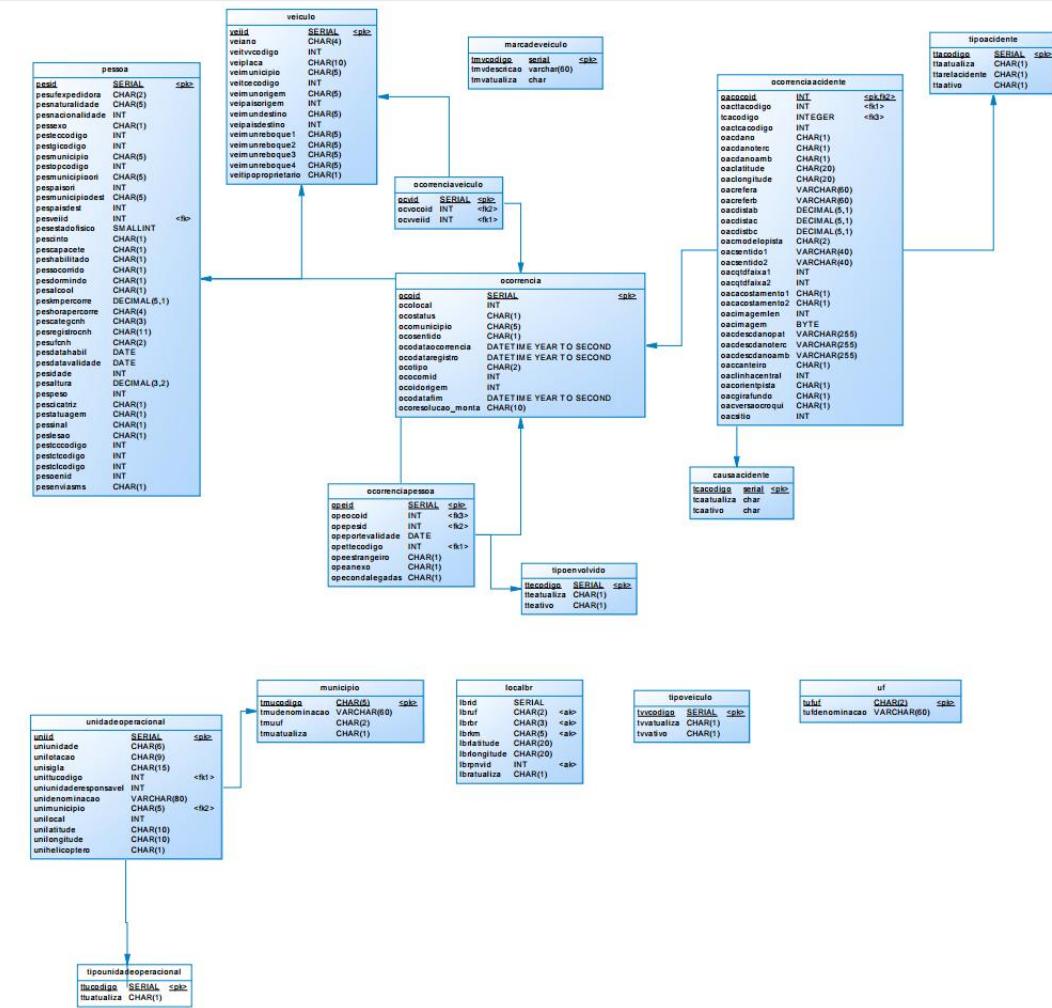


Figura 13 – Modelo de Dados atual das Ocorrências de Acidentes em Rodovias Federais.

Fonte: (??)

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 informação 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 14.

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 15.

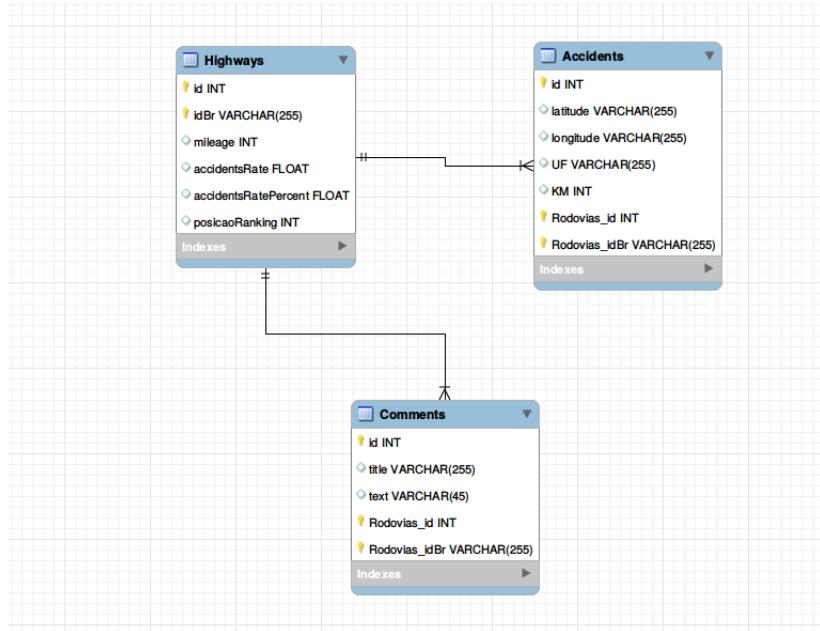


Figura 14 – Modelo de dados do software “Pé na Estrada”.

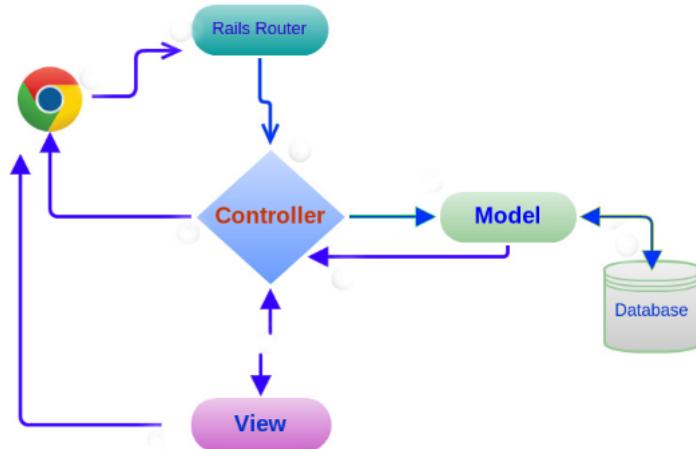


Figura 15 – Arquitetura do sistema. Fonte: Autores

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.

4 Construção da ontologia

Este capítulo apresenta detalhes da construção da ontologia proposta.

4.1 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.

4.2 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.

4.3 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 16 e na Figura 17.

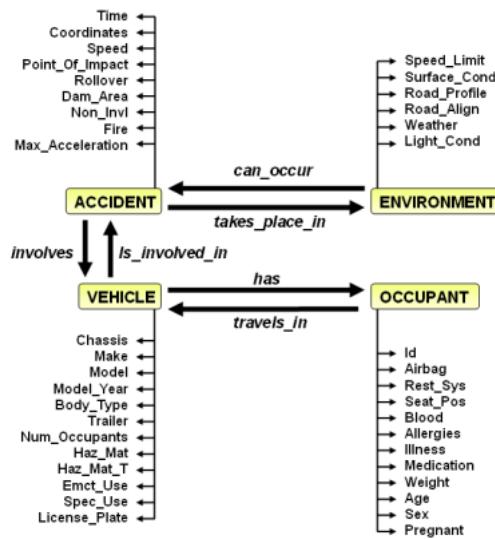


Figura 16 – Componentes da ontologia. Fonte: (??)

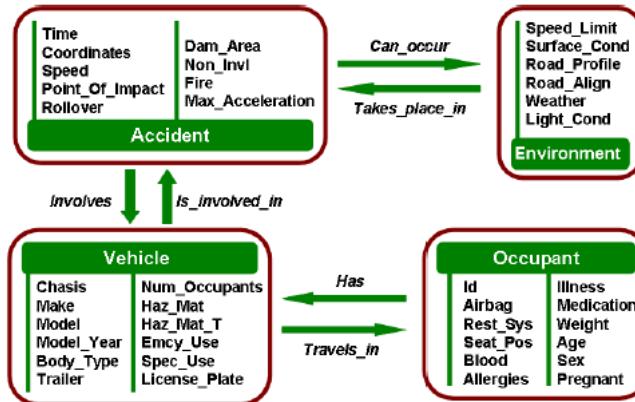


Figura 17 – Componentes da ontologia CAOVA. Fonte: (??)

4.4 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 18 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.

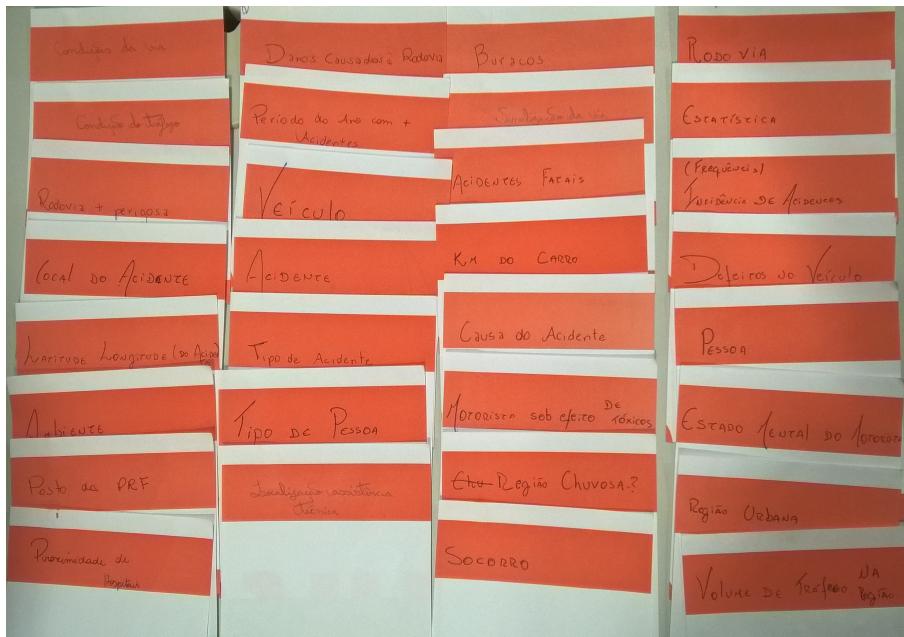


Figura 18 – Conceitos levantados com o *CardSorting*

4.4.1 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 19.

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 19.

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.

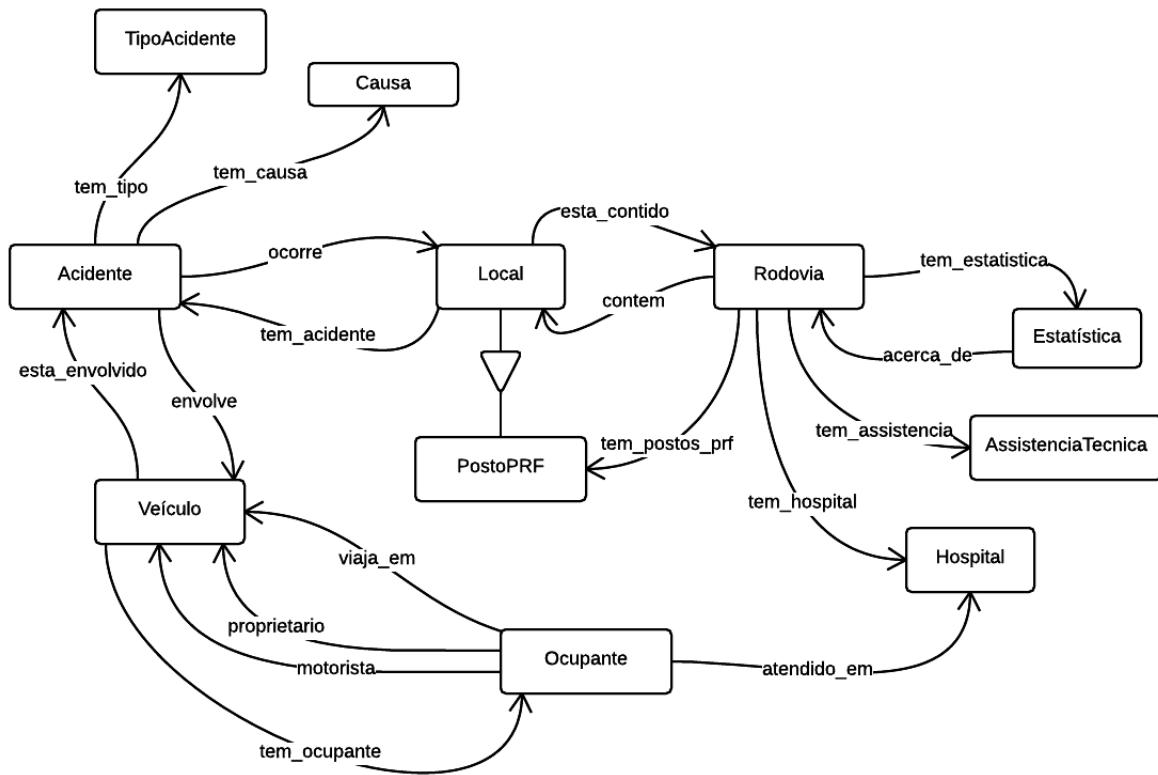


Figura 19 – Modelo conceitual ideal da ontologia a ser construída

4.4.2 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.

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.

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

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

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

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

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

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

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

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

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

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

5 Aplicação da ontologia no software

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 acompanhasssem 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.

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. Infelizmente, esse *framework* não é mais mantido, o que compromete a sua utilização no projeto.

Para ainda assim conseguir trazer uma camada semântica para o software, foi utilizado um banco de dados orientado a grafos, o Neo4J, para substituir a ontologia, uma vez que a ideia de ontologia se baseia no conceito de grafo, sendo perfeitamente possível representar uma ontologia como um grafo.

5.1 Modelagem das entidades

O principal relacionamento existente no “Pé na estrada” é dado pelas entidades Acidente e Rodovia, onde acidentes acontecem em uma Rodovia. Atualmente é feito um mapeamento relacional de “um para muitos” entre Rodovia e Acidente.

Para utilizar o Neo4j no Rails existe uma biblioteca chamada Neo4j.rb¹ que abstrai o acesso à instância do Neo4j e oferece classes para facilitar a modelagem em Ruby. O relacionamento foi mapeado para o Neo4J utilizando recursos do Neo4J.rb e as entidades na aplicação ficaram assim:

¹ <http://www.activerd.org/>

¹ <https://github.com/neo4jrb/neo4j>

```
class Accident
  include Neo4j::ActiveNode

  property :latitude
  property :longitude
  property :uf
  property :km

  has_one :out, :highway,
    type: :highway,
    model_class: :Highway
end

class Highway
  include Neo4j::ActiveNode

  property :br
  property :mileage

  has_many :in, :accidents,
    origin: :highway,
    model_class: :Accident
end
```

Embora essa modelagem inicial seja bastante simples (contando apenas com duas entidades), com essa nova estrutura de classes na camada *Model*, já temos uma melhor representação dos dados e uma interface de consulta mais poderosa.

5.2 Testes da nova modelagem

Para testar a nova estrutura orientada a grafos das *models*, o banco de dados foi carregado com parte dos dados dos acidentes do primeiro semestre de 2014 fornecidos pela PRF.

Foram carregados aproximadamente 13300 registros de acidentes e foi possível realizar as mesmas atividades realizadas anteriormente com o modelo relacional. Em questões de performance o banco de dados orientado a grafos perde um pouco para o banco de dados relacional, mas compensa pelo poder de semântica oferecido.

A Figura 20 ilustra uma parte do banco de dados, mostrando as rodovias, acidentes e seus respectivos relacionamentos cadastrados no Neo4j. Cada “bolinha” pode ser tanto uma rodovia quanto um acidente. Caso seja uma acidente, terá um relacionamento o ligando a outro registro, pois não existem acidentes que não aconteceram em uma rodovia. Registros isolados são rodovias que não possuem acidentes registrados.

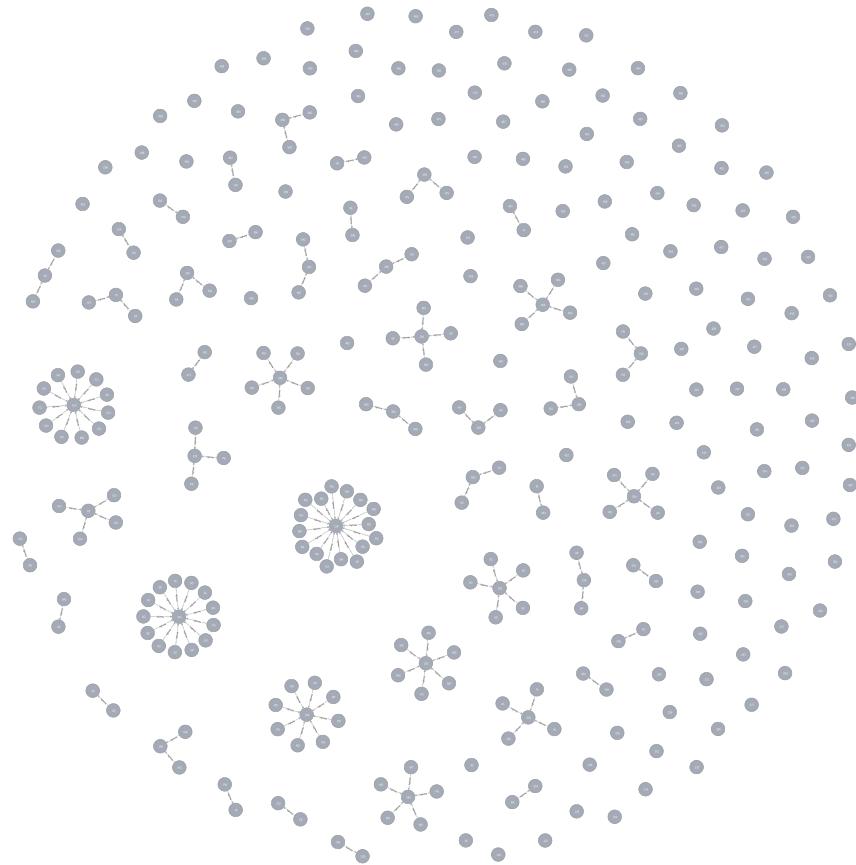


Figura 20 – Visão geral do banco de dados. *Imagen gerada pelo Neo4j.*

A Figura 21 apresenta um exemplo de três rodovias registradas no banco de dados, BR-285, BR-354 e BR-430. Como foi mencionado, cada “bolinha” representa um registro e cada “seta” representa uma relação entre os registros. No caso da 21 a relação “highway” representa que aquela rodovia é o local do acidente. Neste exemplo é possível ver que

ocorreram cinco acidentes na rodovia BR-285 no estado do Rio Grande de Sul, dois acidentes na BR-354 no estado de Minas Gerais e não ocorreram acidentes na rodovia BR-430.

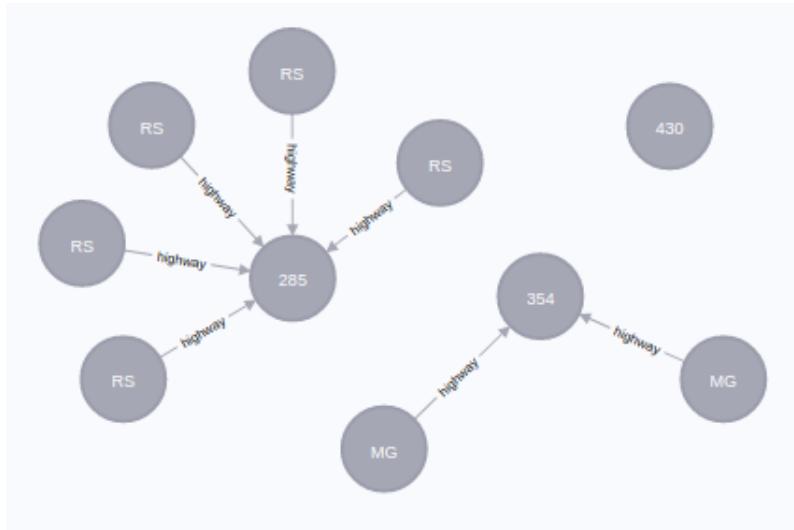


Figura 21 – Exemplo de registros no banco de dados. *Imagen gerada pelo Neo4j.*

Comparação com o banco de dados relacional

Considerando o cenário de recuperar a quantidade de acidentes ocorridos na rodovia BR-116 pelo nome da rodovia, a respectiva consulta em SQL ficaria assim:

Listing 5.1 – Consulta em SQL

```

SELECT COUNT(*) FROM accidents
INNER JOIN highways
ON highways . id = accidents . highway_id
WHERE highways . "br" = '116'
  
```

Já em CYPHER (linguagem de consulta do Neo4j) ficaria assim:

```

MATCH (node : ' Accident ')
MATCH (node)-[rel1 : ' highway ']->(result_highway : ' Highway ')
WHERE (result_highway . br = { result_highway . br })
RETURN count(result_highway)
AS result_highway | { : result_highway . br => "116" }
  
```

Em relação a performance, para 13300 registros, o banco de dados relacional consegue responder a esta consulta em uma média de 3ms, enquanto o banco de dados orientado a grafos responde em 17ms.

6 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 mais avançadas em dados mais interconectados. 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.

Foi possível ver que um banco de dados orientado a grafos pode ser mais intuitivo que um relacional, sendo até visualmente mais agradável que um banco de dados relacional. Embora o banco de dados orientado a grafos seja um pouco mais lento, os recursos fornecidos por ele valem a pena para dados muito conectados.

Anexos

ANEXO A – Telas do software Pé na Estrada

A.1 Tela de ranking das rodovias federais

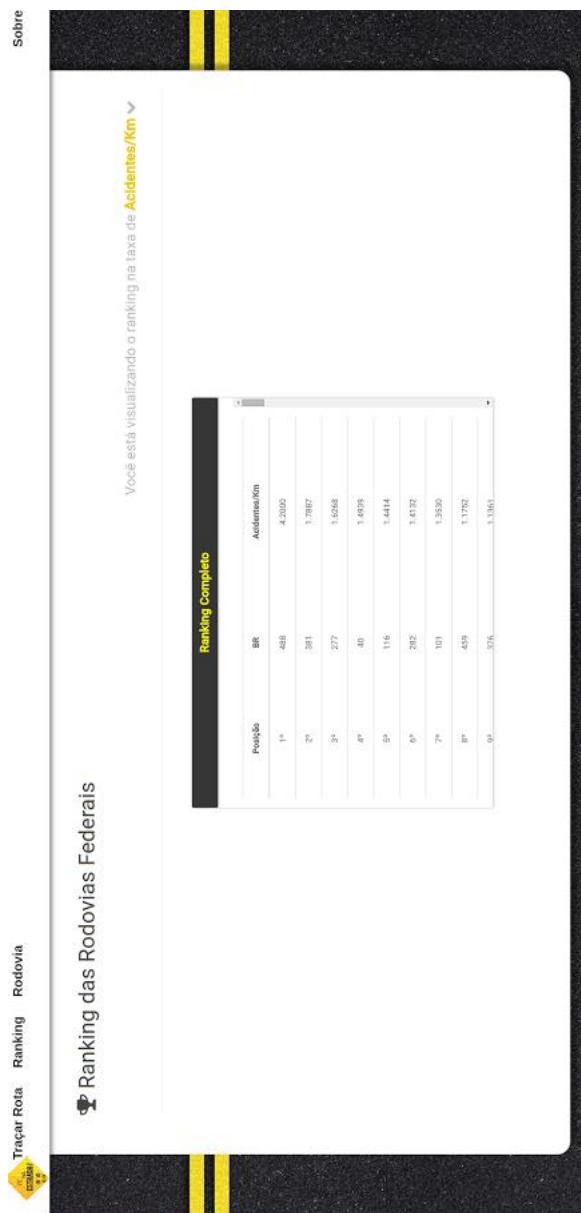


Figura 22 – Tela de ranking das rodovias federais.

A.2 Tela de sinalização dos acidentes em uma rota

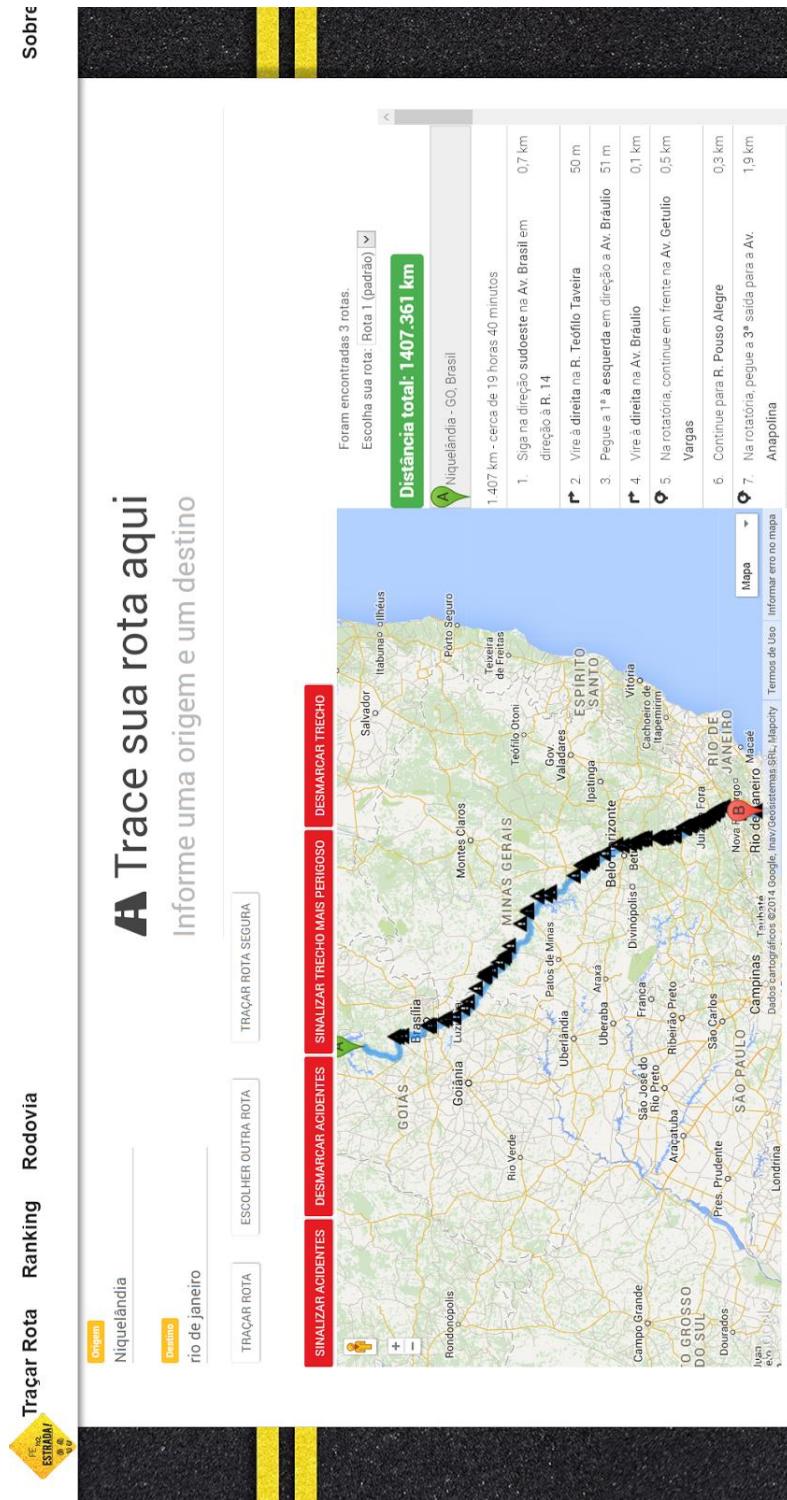


Figura 23 – Tela de sinalização dos acidentes em uma rota.

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



Figura 24 – Tela de sinalização do trecho mais perigoso de uma rota.