

# ONTOLOGIAS E PROTÉGÉ

TIAGO SEMPREBOM\*, MARCOS YUZURU CAMADA\*, IGOR MENDONÇA\*

\*Programa de Pós-Graduação em Engenharia de Automação e Sistemas  
Universidade Federal de Santa Catarina (UFSC) - Florianópolis, SC, Brazil

Emails: [tisemp@das.ufsc.br](mailto:tisemp@das.ufsc.br), [mcamada@das.ufsc.br](mailto:mcamada@das.ufsc.br), [igor@das.ufsc.br](mailto:igor@das.ufsc.br)

**Abstract**— This paper purpose to present the main concepts and application of ontology. Besides, it describes a ontology development tool called Protégé-2000. A use case is design to show how to handle this tool and more basic aspects in ontology build.

**Resumo**— Este trabalho tem o objetivo de apresentar os principais conceitos e aplicações de ontologia. Além disso, é feita uma descrição de uma ferramenta de construção de ontologias denominado Protégé-2000. Um caso de uso é projetado para ilustrar a utilização desta ferramenta, além dos aspectos fundamentais na construção de ontologias.

**Palavras-chave**— Ontologia, Protégé, Representação do Conhecimento.

## 1 Introdução

Nos últimos anos, o aumento exponencial dos dados disponíveis tem conferido importância significativa às técnicas de organização da informação. Essas técnicas fazem parte de um corpo de estudos que buscam melhorias no tratamento de dados, atuando na sua seleção, no seu processamento, na sua recuperação e na sua disseminação.

Muitas vezes, pedaços importantes da informação relevante estão dispersos em diferentes recursos de informação e o meio mais utilizado para encontrá-los é através de máquinas de busca (engenhos de busca). Entretanto, as máquinas de busca tradicionais retornam listas de recursos recuperados, oferecendo pouca ou nenhuma informação sobre as relações semânticas existentes entre eles. Por conseguinte, o usuário tem de despender uma quantidade substancial de tempo acessando-os e lendo-os, para então descobrir como esses recursos de informação estão relacionados. Neste contexto, uma abordagem que tem recebido atenção é a utilização de ontologias na organização do conteúdo dos dados. Uma ontologia é criada baseando-se no conhecimento de especialistas em um determinado domínio e define as regras que regulam a combinação entre termos e relações de um domínio do conhecimento. O que se busca, em última instância, são melhorias nos processos de recuperação da informação (Almeida and Bax, 2003).

As ontologias oferecem um meio de lidar com a representação de recursos de informação: o modelo de domínio descrito por uma ontologia pode ser usado como uma estrutura unificadora para dar semântica e uma representação comum à informação (Falbo et al., 2004).

Ontologias são utilizadas hoje em diversas áreas para organizar a informação. São encontradas na literatura diversas definições para as ontologias, diversos tipos, propostas para aplicação em

diferentes áreas de conhecimento e propostas para a construção de ontologias (metodologias, ferramentas e linguagens). Tal diversidade tem dificultado a escolha e a utilização das técnicas disponíveis para a manipulação de ontologias na organização da informação.

O objetivo deste artigo é apresentar os principais conceitos, tipos e aplicações de ontologias, traz também um estudo sobre umas das principais ferramentas utilizadas para implementar ontologias, o Protégé. Um caso de uso utilizando esta ferramenta também é introduzido.

Este trabalho foi dividido em 4 partes e está organizado como segue: A Seção 2 traz um levantamento dos principais conceitos, tipos e metodologias para o desenvolvimento de ontologias. Na Seção 3 são apresentados alguns projetos que fazem uso de ontologias e alguns repositórios de ontologias disponíveis na Internet. A Seção 4 faz um levantamento sobre a ferramenta de construção de ontologias Protégé-2000. Na Seção 5, um estudo de caso é mostrado, juntamente com alguns aspectos interessantes da modelagem e implementação do caso de uso. E finalmente, a Seção 6 discute as principais conclusões acerca do trabalho.

## 2 Ontologias

O termo ontologia é originário da Filosofia. O termo foi introduzido por Aritóteles, no contexto da pesquisa em ontologia filósofos tentam responder as questões: O que é um ser? e quais são as características comuns de todos os seres? O termo foi recentemente adotado pelas comunidades de Inteligência Artificial (IA) e gestão de conhecimento para se referir a conceitos e termos usados para descrever alguma área do conhecimento ou construir uma representação desse.

Em Inteligência Artificial pode-se assumir uma interpretação para ontologia como sendo um

conjunto de entidades com suas relações, restrições, axiomas e vocabulário. Um ontologia define um domínio, ou, mais formalmente, especifica uma conceitualização acerca dele (Gruber, 1995). Segundo Clark (Clark, 1999) uma ontologia é organizada em hierarquias de conceitos (ou taxonomias), pelo fato de não refletirem idealmente nenhum formalismo específico, pode-se considerar um ontologia como a materialização do nível do conhecimento.

### 2.1 Tipos de ontologias

Diferentes tipos de ontologias, de acordo com seu grau de genericidade podem ser delineados (Gomes-Perez, 1999):

- *Ontologias de representação*: definem as primitivas de representação – *frames*, axiomas, atributos e outros – de forma declarativa;
- *Ontologias Gerais*: trazem definições abstratas necessárias para a compreensão de aspectos do mundo, como tempo, processos, papéis, espaço, seres, coisas, etc;
- *Ontologias Centrais ou Genéricas*: definem os ramos de estudos de uma área e/ou conceitos mais genéricos e abstratos desta área. Por exemplo, regras básicas do direito;
- *Ontologias de Domínio*: tratam de um domínio mais específico de uma área genérica de conhecimento, como direito tributário, microbiologia, etc;
- *Ontologias de Aplicação*: procuram solucionar um problema específico de um domínio, como identificar doenças do coração, a partir de uma ontologia de domínio de cardiologia.

Existe ainda outra classificação para ontologias, aplicáveis apenas para as duas últimas citadas acima. São elas:

- *Ontologias de tarefas*: descrevem tarefas de um domínio – como processos, planos, metas, escalonamentos, etc – com uma visão mais funcional, embora declarativa, de um domínio;
- *Ontologias de domínio*: tem uma visão mais epistemológica<sup>1</sup> do domínio, focando nos conceitos e objetos do universo discurso.

### 2.2 Porque desenvolver ontologias?

Uma ontologia define um vocabulário comum para pesquisadores que compartilham informações em um domínio. Porque desenvolver ontologias? Alguns motivos interessantes são (Natalya and Deborah, 2001):

- Compartilhar conhecimento e estruturas de informação entre pessoas e agentes de software;
- Reutilização do conhecimento de um domínio;
- Explicitar hipóteses;
- Separar conhecimento de um domínio do conhecimento operacional;
- Análise do domínio de conhecimento.

### 2.3 Benefícios das ontologias

A utilização de ontologias em substituição à abordagem declarativa apresenta muitos benefícios, os mais relevantes podem ser verificados como segue:

- Reuso de ontologias: possibilidade de os desenvolvedores de sistemas baseados em conhecimento resusar ontologias e bases de conhecimento, mesmo com adaptações e extensões;
- Ontologias de Prateleira: disponibilização de uma vasta quantidade de ontologias prontas para uso e comunicação por agentes;
- Tradução entre linguagens e formalismos de representação do conhecimento: A tradução facilita o reuso de conhecimento, e pode vir a permitir comunicação entre agentes em formalismos diferentes, uma vez que este serviço encontra-se disponível um número cada vez maior de formalismos de representação de conhecimento. Outra forma de alcançar este objetivo é através da utilização de editores de ontologias, como por exemplo o Protégé-2000 (Noy et al., n.d.), onde pode-se gerar ontologias em CLIPS, Prolog, XML (*eXtensible Markup Language*), Jess<sup>2</sup>, RDF (*Resource Description Framework*), OIL (*Ontology Interface Layer*), etc;
- Acesso *on-line* a servidores de ontologias: capazes de armazenar milhares de classes e instâncias, que serviriam a várias empresas e grupos de pesquisa, servindo como base a todas elas, mantendo um vocabulário uniforme.

### 2.4 Como criar uma ontologia?

Não existe um padrão definido de como criar uma ontologia, existem diversas maneiras de se desenvolver um projeto, utilizando ontologias. Porém, existem sete passos que podem ser considerados na criação de uma ontologia (Natalya and Deborah, 2001):

1. *Determinar o Domínio e o Escopo da Ontologia*: O primeiro passo no desenvolvimento de uma ontologia é a definição do domínio e o escopo a ser tratado;

<sup>1</sup>Epistemologia ou teoria do conhecimento é um ramo da filosofia que trata dos problemas filosóficos relacionados à crença e ao conhecimento.

<sup>2</sup>[www.jessrules.com](http://www.jessrules.com)

2. *Reutilizar Ontologias Existentes*: No segundo passo, deve-se considerar a existência de uma ontologia semelhante ou se existe a possibilidade de refinar e estender uma ontologia existente para uma tarefa específica. Muitas ontologias já estão disponíveis de forma eletrônica e podem ser importadas para ambientes de desenvolvimento de ontologias (Seção 3.1);
3. *Levantar termos importantes*: Uma tarefa muito útil é levantar uma lista dos termos mais importantes a serem utilizados na ontologia. Quais as propriedades dos termos? O que podemos dizer sobre estes termos?;
4. *Definir classes e sua hierarquia*: Devemos definir as classes e as subclasses definindo as classes mais genéricas e realizando uma especialização destas na maioria das vezes;
5. *Definir propriedades das classes (slots)*: Uma classe sozinha não apresenta nenhuma informação. Uma vez definida quais são as classes, deve-se descrever a estrutura interna da idéia. Ex. uma classe de vinhos, pode-se ter os seguintes *slots*: cor, sabor, quantidade de açúcar, ano da safra e produtor;
6. *Restrições das Propriedades (facetas)*: Os *slots* podem possuir diferentes facetas descrevendo tipos de valores, valores permitidos, cardinalidade, etc. Ex. O nome de um vinho (slot nome) pode receber uma *string* de até 20 caracteres, esta restrição é a faceta do *slot*;
7. *Criação de instâncias*: O último passo é a criação de instâncias individuais para as classes na hierarquia. Define-se as instâncias para as classes: 1) escolhendo a classe, 2) criando uma instância individual da classe e 3) preenchendo os valores dos *slots*.

### 3 Projetos que Utilizam Ontologias

Atualmente, existem vários projetos que fazem uso de ontologias. Ontologias são utilizadas em projetos de domínios como gestão do conhecimento, comércio eletrônico, processamento de linguagens naturais, recuperação da informação na *web*, de cunho educacional, etc.

A tabela 1 apresenta alguns projetos que fazem uso de ontologias.

#### 3.1 Repositórios de Ontologias

Existem ontologias disponíveis para uso ou para modelar a construção de outras ontologias. A tabela 2 apresenta alguns repositórios de ontologias disponíveis na Internet.

Tabela 1: Projetos que fazem uso de ontologias

Projeto	Utilização
PlanetOnto	Gestão de conhecimento
MKBEEM	Comércio eletrônico
Oncoterm	Proc. de linguagens naturais
OntoSeek	Rec. de informação na <i>web</i>
Smartrainer	Projeto relacionado à educação

Tabela 2: Ontologias Disponíveis

Repositório	Sítio <i>web</i>
DAML	<a href="http://www.daml.org/ontologies/">www.daml.org/ontologies/</a>
Ontolingua	<a href="http://www.ksl-svc.stanford.edu">www.ksl-svc.stanford.edu</a>
Universal	<a href="http://www.ist-universal.org">www.ist-universal.org</a>

## 4 Protégé

### 4.1 Visão Geral

O Protégé é uma ferramenta que permite construir ontologias de domínio, personalizar formulários de entrada de dados, inserir e editar dados, possibilitando então, a criação de bases de conhecimento guiadas por uma ontologia. Sua interface gráfica, como exibida na figura 1, provê acesso a barra de menus e barra de ferramentas, além de apresentar cinco áreas de visualização (*views*) que funcionam como módulos de navegação e edição de classes, atributos, formulários, instâncias e pesquisas na base de conhecimento, propiciando a entrada de dados e a recuperação das informações (Paschoal, 2005).

### 4.2 Histórico do Protégé

Desenvolvido pelo departamento de informática médica da Universidade de Stanford, o projeto original do Protégé era uma ferramenta de aquisição de conhecimento limitada a um sistema especialista para oncologia.<sup>3</sup>

Foi modernizado gradativamente para acompanhar a evolução da tecnologia de SBC (Sistemas Baseados em Conhecimento), ou seja, servir para aquisição de conhecimento diretamente de especialistas de domínios com menos dependência de engenheiros de conhecimento, permitir diversos formalismos e estratégias de inferência, integrar tarefas (aquisição de ontologias e instâncias, ambiente de teste com inferência) num mesmo ambiente, criar automaticamente formulários para entrada de conhecimento, acessar ontologias via OKBC (Open Knowledge Base Connectivity) e a combinar de ontologias.

O Protégé foi desenvolvido utilizando a linguagem de programação Java e, posteriormente,

<sup>3</sup>Também chamada de Cancerologia, é a especialidade médica que estuda as neoplasias (tumores malignos) e a forma de como essas doenças se desenvolvem no organismo.

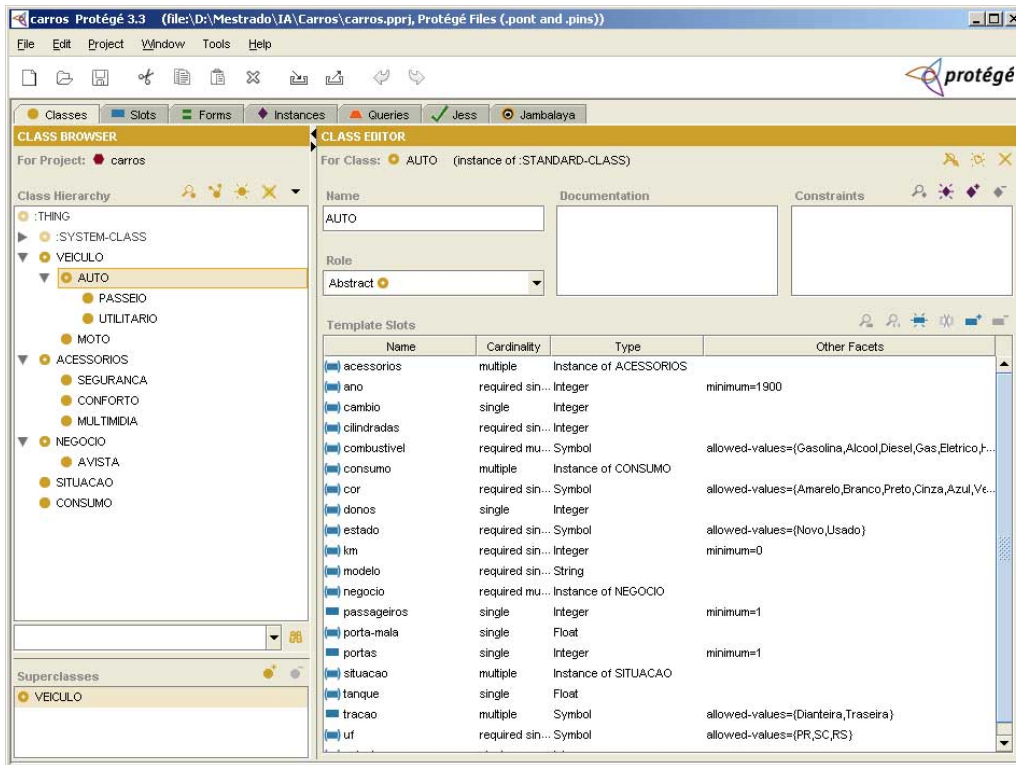


Figura 1: Interface gráfica do Protégé

optou por abrir seu código. Surgiu então uma arquitetura integrável a diversas aplicações, via componentes que podem ser conectados ao sistema. Como consequência desta decisão e de sua difusão, componentes de vários matizes, elaborados por grupos de pesquisa de usuários, puderam ser adicionados ao sistema, sem necessitar o redesenvolvimento. Foram aproveitados, por exemplo, o Jambalaya, um utilitário com animação e vários outros recursos em visualização de dados, e o Ontoviz, um componente que faz com que o gerador de gráficos Graphviz da AT&T produza gráficos com instâncias, heranças e outros tipos de relacionamento. (Freitas, 2004)

As formas de contribuição são:

1. *Backends*, permitem usuários armazenar e importar bases de conhecimento em vários formatos;
2. *Slot widgets*, são usados para mostrar e editar valores de *slot* ou suas combinações em modos de domínio específico e tarefa específica;
3. *Tab plug-ins*, são aplicativos baseados no conhecimento usualmente fortemente ligado a bases de conhecimento do Protégé.

#### 4.3 Arquitetura do Protégé

A arquitetura do Protégé é ilustrada na figura 2, onde a interface do usuário é definida pelos *plug-ins* e *slots* que fazem a interface frente à base

de conhecimento definidas pelo Protégé, representada pelo núcleo do Protégé que faz o intermédio com a base de armazenamento persistente podem ser em formato de banco de dados ou arquivos.

A modularização do Protégé é mostrada na figura 2 e está dividida em:

- Interface com o usuário;
- Núcleo do Protégé;
- Armazenamento persistente.

Essa arquitetura ajuda no crescimento da ferramenta, pois novos módulos são desenvolvidos preocupando-se somente com o módulo que se deseja, seja uma nova base de dados de armazenamento persistente ou um novo *plug-in* de interface com os usuários.

#### 4.4 Outras características do Protégé

Outras características do Protégé estão listadas abaixo. (Freitas, 2004)

- A linguagem axiomática PAL (*Protégé Axiomatic Language*). Permite a inserção de restrições e axiomas que incidem sobre as classes e instâncias de uma ou mais ontologias;
- A geração de arquivos de saída alteráveis. Atualmente, podem ser criados classes e instâncias em CLIPS - a base de conhecimento é gerada nativamente para esse motor de inferência;

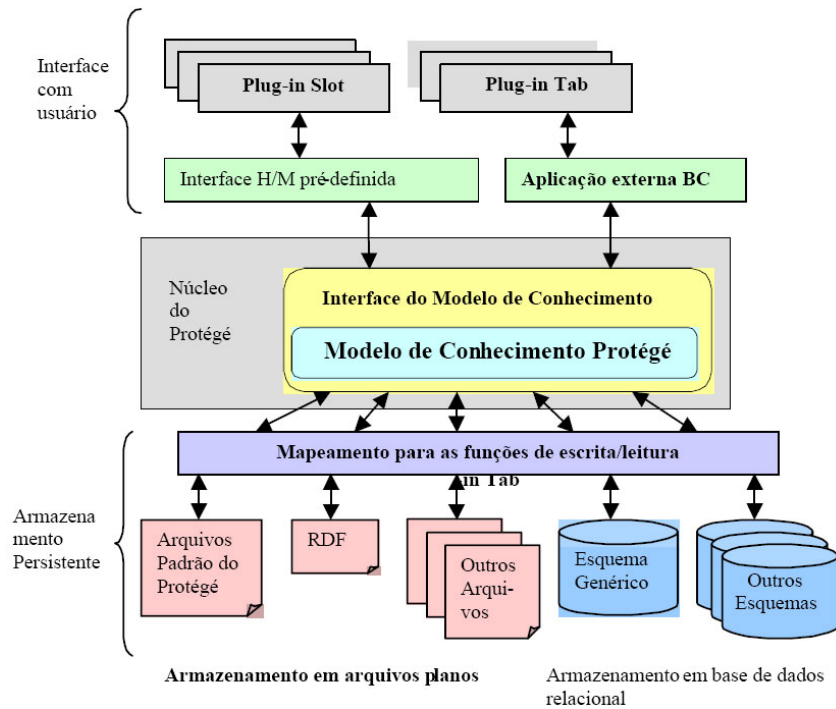


Figura 2: Arquitetura do Protégé

- Uma interface para entrada de conhecimento, incluindo um gerador automático de formulários para as classes definidas. Admite-se ainda a reposição da interface original por componentes mais adequados à aplicações específicas. Esta interface facilita sobremaneira o gerenciamento de conhecimento de uma ou mais ontologias.

Dentre os arquivos de saída suportáveis estão: Jess, F-Logic, Prolog, RDF, OIL, XML, Topic Maps.

## 5 Caso de Uso

Este trabalho modela um caso de uso de aplicação de ontologia com o Protégé. O domínio que se pretende representar é de sugestão de venda de veículos (carro ou moto) a partir de características do veículo determinadas pelo usuário. Na figura 3, é mostrado a hierarquia das classes modeladas. Nas subseções seguintes, é realizada uma descrição das classes modeladas.

### 5.1 Veículo

A classe *Veículo* é uma classe abstrata que define as entidades Veículos no mundo real. A entidade Veículo representar qualquer dos meios para transportar ou conduzir pessoas ou objetos, por exemplo, carro, moto, avião, etc. Por ser uma classe abstrata, ela não pode ser instanciada, servindo apenas como modelo para outras classes. Os *slots* pertencentes a esta classe são descritas na tabela 3.

Tabela 3: *Slots* da classe *Veículo*

Nome	Descrição
<i>acessorios</i>	Acessórios possíveis
<i>ano</i>	Ano de fabricação
<i>cilindradas</i>	Quantidade de cilindros
<i>combustivel</i>	Combustível utilizado
<i>consumo</i>	Consumo de combustível
<i>cor</i>	Cor
<i>estado</i>	Novo ou usado
<i>marcha</i>	Quantidade de marchas
<i>negocio</i>	Forma de pagamento
<i>numero_donos</i>	Quantidade de donos
<i>porta_mala</i>	Volume do porta mala
<i>quilometragem</i>	Quilometragem rodada
<i>tamanho_tanque</i>	Volume do tanque
<i>uf</i>	Estado de origem
<i>valvulas</i>	Quantidade de válvulas

### 5.2 Autos

A classe *Autos* é uma classe abstrata que define as entidades Autos. A entidade Autos representa um Veículo que possui, normalmente, quatro rodas conhecida comumente como carro. Ela herda as propriedades (*slots*) da classe *Veículo*. A tabela 4 contém os *slots* que esta classe possui, além das herdadas da classe *Veículo*.

### 5.3 Utilitário

A classe *Utilitario* é uma classe concreta que define os Autos do tipo Utilitário. A entidade Utilitário representa um Auto tipicamente de circula-

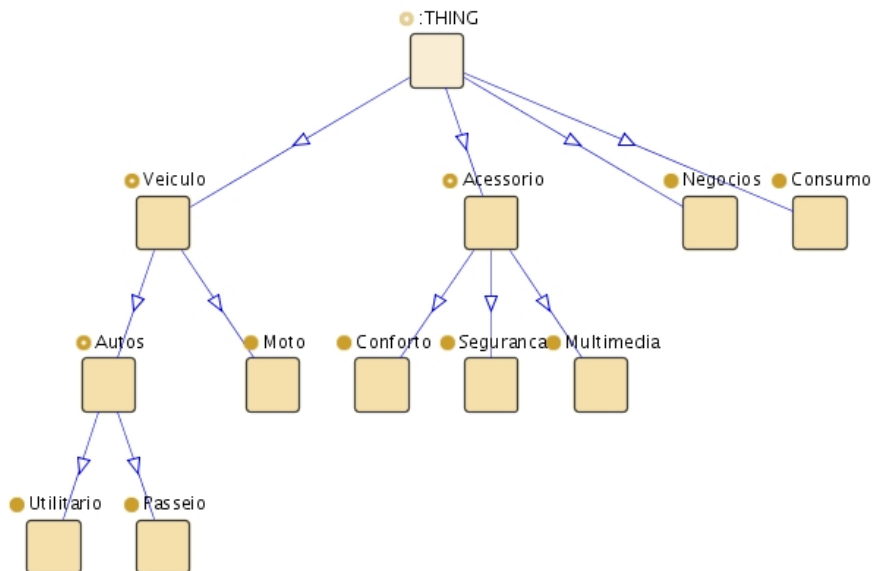


Figura 3: Hierarquia das classes

Tabela 4: Slots da classe Autos

Nome	Descrição
<i>cambio</i>	Mec., semi-aut. ou aut.
<i>modelo</i>	Modelo (nome)
<i>numero_passageiros</i>	Quant. de passageiros
<i>numero_portas</i>	Quant. de portas
<i>tracao</i>	Dianteira ou traseira

ção nas cidades. Ela herda as propriedades (*slots*) da classe *Auto*. A tabela 5 contém os *slots* que esta classe possui além das herdadas da classe *Auto*.

Tabela 5: Slot da classe Utilitario e Passeio

Nome	Descrição
<i>marca</i>	Marca (Fabricante)

#### 5.4 Passeio

A classe *Passeio* é uma classe concreta que define os Autos do tipo Passeio. Esta entidade representa um Auto utilizada para viagem, normalmente, com grande bagageiro. Ela herda as propriedades (*slots*) da classe *Auto*. Os *slots* desta classe é a mesma da classe *Utilitario* na tabela 5.

#### 5.5 Moto

A classe *Moto* é uma classe concreta que define as entidades Motos. Esta entidade representa os Veículos de duas rodas. Ela herda as propriedades (*slots*) da classe *Veiculo*. A tabela 6 contém os *slots* que esta classe possui além das herdadas da classe *Veiculo*.

Tabela 6: Slots da classe Moto

Nome	Descrição
<i>modelo</i>	Modelo
<i>partida</i>	Manual ou eletrônica
<i>refrigeracao</i>	Água ou ar
<i>versao</i>	Número da versão

#### 5.6 Acessorio

A classe *Acessorio* é uma classe abstrata que define as entidades Acessórios como um todo. A entidade *Acessorio* representa os utensílios que podem acompanhar o Veículo. Por ser abstrata, ela não pode ser instanciada, servindo apenas como modelo para outras classes. Os *slots* pertencentes a esta classe estão descritas na tabela 7.

Tabela 7: Slot da classe Acessorio

Nome	Descrição
<i>nome</i>	Nome

#### 5.7 Multimedia

A classe *Multimedia* é uma classe concreta que define os Acessórios do tipo Multimedia (rádio com mp3, *dvd* para carona, etc). Ela herda as propriedades (*slots*) da classe *Acessorio* na tabela 7.

#### 5.8 Seguranca

A classe *Seguranca* é uma classe concreta que define os Acessórios do tipo Segurança (*air bag*, sensor de colisão traseira, etc). Ela herda as propriedades (*slots*) da classe *Acessorio* na tabela 7.

### 5.9 Conforto

A classe *Conforto* é uma classe concreta que define os Acessórios do tipo Conforto (ar condicionado, etc). Ela herda as propriedades (*slots*) da classe *Acessorio*. Esta classe também herda as propriedades (*slots*) da classe *Acessorio* na tabela 7.

### 5.10 Negócios

A classe *Negocios* é uma classe concreta que define a forma de pagamento de um determinado Veículo. Os *slots* pertencentes a esta classe estão descritas na tabela 8.

Tabela 8: Slot da classe *Negocio*

Nome	Descrição
<i>valor</i>	Valor do pagamento

### 5.11 Consumo

A classe *Consumo* é uma classe concreta que define o tipo de consumo de um determinado Veículo. Os *slots* pertencentes a esta classe estão descritas na tabela 9.

Tabela 9: Slot da classe *Consumo*

Nome	Descrição
<i>local</i>	Local de consumo
<i>valor</i>	Valor do consumo em um local

## 6 Conclusões

Este artigo proporcionou uma visão geral sobre ontologias. Apresenta-se conceitos e definições importantes, tipos de aplicações e metodologias. Pesquisou-se o uso de ontologias, destacando projetos e repositórios de ontologias. O trabalho traz uma descrição sobre uma importante ferramenta de construção de ontologias, o Protégé. Um caso de uso descrevendo os principais aspectos na implementação de um ontologia também é abordado.

Um grande volume de pesquisas sobre o tema sugere a importância e utilidade das ontologias na tarefa de organizar informações dentro de um determinado domínio do conhecimento. As ontologias podem proporcionar melhorias no processo de recuperação da informação, organizando o conteúdo das fontes de dados que compõem o domínio em questão. Outra importante funcionalidade das ontologias vêm do fato delas possibilitarem a utilização de mecanismos de inferência para criar um novo conhecimento a partir do conhecimento existente. As ontologias também tem servido, em algumas áreas, como uma maneira de unificar certos conhecimentos e conceitos, servindo assim como um elemento estruturador.

Espera-se que este trabalho possa servir como base para outros estudos que utilizem algumas das abordagens aqui levantadas.

## Agradecimentos

Os autores agradecem aos colegas que apresentaram seus seminários, compartilhando assim o conhecimento assimilado e ao professor Guilherme Bittencourt.

## Referências

- Almeida, M. B. and Bax, M. P. (2003). *Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção*, Revista Ciência da Informação.
- Clark, D. (1999). *Mad cows, metathesaurim and meaning*, IEEE Intelligent Systems.
- Falbo, R., Ruy, F., Pezzin, J. and R., D. M. (2004). Ontologias e ambientes de desenvolvimento de software semântico, *IV Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento, JIISIC'2004* 1: 277–292.
- Freitas, F. G. (2004). *Ontologias e a Web Semântica*, Programa de Pós-Graduação em Informática, Universidade Católica de Santos.
- Gomes-Perez, A. (1999). *Tutorial on Ontological Engineering*, International Joint Conference on Artificial Intelligence - IJCAI'1999.
- Gruber, T. R. (1995). *Towards Principles for a Design of Ontologies Used for Knowledge Sharing.*, International Journal of Human and Computer Studies.
- Natalya, F. N. and Deborah, L. M. (2001). *Ontology Development 101: A Guide to Creating your First Ontology*, Stanford Medical Informatics Technical Report SMI.
- Noy, N., Ferguson, R. and Musen, M. (n.d.). The knowledge modelo of protege-2000: Combining interoperability and flexibility, *2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*.
- Paschoal, A. (2005). *Protégé + Ausubel + ITIL: uma proposta ontológica para compartilhamento de conhecimento na Seção de Suporte Operacional do Tribunal Superior Eleitoral.*, Brasília: UnB, 2005. f. 128: il. ; 29,5 cm. Monografia, Universidade de Brasília.