

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
CURSO DE SISTEMAS DE INFORMAÇÃO

**UMA FERRAMENTA DE APOIO À DETERMINAÇÃO DE  
EQUIVALÊNCIAS SEMÂNTICAS ENTRE ESQUEMAS GML  
UTILIZANDO ONTOLOGIAS OWL**

LEONARDO ROMAN DA ROSA

Trabalho de conclusão de curso apresentado como parte  
dos requisitos para obtenção do grau de Bacharel em  
Sistemas de Informação.

Florianópolis - SC

2006 / 2

LEONARDO ROMAN DA ROSA

**UMA FERRAMENTA DE APOIO À DETERMINAÇÃO DE  
EQUIVALÊNCIAS SEMÂNTICAS ENTRE ESQUEMAS GML  
UTILIZANDO ONTOLOGIAS OWL**

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Ronaldo dos Santos Mello, Dr.

Banca examinadora

Lia Caetano Bastos, Dr.

Marília Abrahão Amaral, Dr

## Sumário

Sumário.....	3
Lista de Figuras.....	5
Lista de Reduções.....	7
Resumo.....	8
Palavras-chave.....	8
1 Introdução .....	9
2 Conceitos Relacionados.....	11
2.1 SIG e GML .....	11
2.2 A importância da GML.....	12
2.3 Interoperabilidade entre Esquemas GML.....	12
2.4 O Uso da Ontologia na Integração .....	13
3 Trabalhos Relacionados.....	16
3.1 Unificação Semântica de Esquemas Conceituais de BDGs.....	16
3.2 Interoperabilidade na representação de dados geográficos.....	17
3.3 Catálogos de Objetos Geográficos para Interoperabilidade entre BDGs.....	18
3.4 Ferramenta para Integração Semântica de Bancos de Dados Espaciais .....	18
4 Método para Determinação das Equivalências Semânticas entre Esquemas GML .....	20
5 Descrição da Ferramenta.....	22
5.1 Implementação .....	22
5.2 Árvores de Comparação.....	25
5.2.1 <i>Wrappers</i> das Árvores OWL e GML.....	26
5.3 Determinação da Equivalência Semântica .....	29
5.4 Métricas de Similaridade .....	30
5.4.1 Métricas para Valores Complexos.....	31
5.4.2 Métricas para Valores Atômicos .....	34
5.4.3 Exemplo de Determinação de Similaridade.....	34
5.5 Catalogação dos Mapeamentos.....	35
6 Estudo de Caso.....	37
7 Conclusão .....	39

8	Referências Bibliográficas .....	41
	Anexos .....	45
	Anexo 1 – Artigo.....	45
	Anexo 2 – Esquemas Utilizados no Estudo de Caso .....	56
	Anexo 3 - Código Fonte .....	112

## Lista de Figuras

Figura 1. Fragmento de documento GML .....	11
Figura 2. Fragmento de uma ontologia OWL .....	15
Figura 3. Visão geral do método .....	20
Figura 4. Diagrama de seqüência da ferramenta .....	22
Figura 5. Janela principal da ferramenta .....	24
Figura 6. Exemplo de nodo para um elemento OWL e para um elemento GML na estrutura de árvore canônica.....	25
Figura 7. Fragmento de árvore canônica para uma ontologia OWL.....	27
Figura 8. Fragmento de árvore canônica para um esquema GML.....	28
Figura 9. Exemplo da tela de Validação de Equivalências.....	30
Figura 10. Catalogação dos mapeamentos.....	35

## Lista de Tabelas

Tabela 1. Exemplo da lista de sinônimos .....	28
Tabela 2. Exemplo de equivalências entre tipos de dados .....	33
Tabela 3. Mapeamento entre cardinalidades de classes OWL e elementos GML .....	34
Tabela 4 – Similaridade entre elementos (GML 2) .....	38

## **Lista de Reduções**

BDG – Banco de Dados Geográfico

GIS – Geography Information System

GML – Geography Markup Language

OGC – OpenGIS Consortium

OWL – Web Ontology Language

RDF – Resource Description Framework

SIG – Sistema de Informação Geográfica

W3C – World Wide Web Consortium

XML – eXtensible Markup Language

## **Resumo**

Um dos problemas existentes na *Web* é a identificação semântica adequada do conteúdo de um *site*. O uso da XML procura resolver ambigüidades no significado desses conteúdos através da utilização de *tags* auto-descritivas. Em sistemas de informações geográficas existe um problema similar relacionado à integração semântica de esquemas GML.

Sendo um padrão aberto, a GML é utilizada para o armazenamento de informações geográficas codificadas em XML. Enquanto a GML facilita o intercâmbio de informações geográficas, ela não resolve por si só a questão da integração semântica de esquemas heterogêneos. Tal integração é importante para permitir troca e reutilização de informação geográfica entre diferentes sistemas de informações geográficas.

Este trabalho tem como propósito a criação de uma ferramenta que implementa um método de apoio à integração semântica entre esquemas GML. Para facilitar a análise de similaridade é utilizada uma ontologia OWL com a representação dos conceitos do domínio utilizado.

## **Palavras-chave**

XML, Extensible Markup Language, GML, Geography Markup Language, OWL, Web Ontology Language, Ontologias, SIG, GIS, Integração Semântica



## 1 Introdução

Existe uma crescente necessidade de troca de informações entre instituições geradoras de dados geográficos. Muitas pesquisas têm sido desenvolvidas a respeito da integração entre objetos geográficos armazenados em fontes de dados distintas e com estrutura sintática e semântica heterogênea [AZEVEDO 2006]. Órgãos públicos e privados que utilizam informações geográficas de diferentes fontes podem se beneficiar com a integração entre formatos de dados geográficos que mantêm as mesmas informações. Como exemplo, o governo de um estado poderia consultar os dados geográficos de seus municípios de forma simples com a integração dos Sistemas de Informação Geográfica de suas prefeituras. Dessa forma uma única base de dados seria pesquisada.

A Geography Markup Language (GML) é um padrão aberto para o transporte e armazenamento de informações geográficas em arquivos codificados na linguagem XML [OGC 2003]. Apesar de facilitar o intercâmbio de informações geográficas, a GML não resolve por si só a questão da integração semântica de esquemas de dados geográficos heterogêneos.

Como a versão atual da GML ainda apresenta limitações em sua representação semântica, uma forma de facilitar a interoperabilidade semântica entre dois Sistemas de Informação Geográfica (SIGs) é associar as definições de dados à ontologia que descreva o domínio, como propõe [LAKE 2002]. A Web Ontology Language (OWL) é a recomendação mais recente do W3C para a representação de ontologias, além de ser compatível com a arquitetura da Web Semântica [OWL 2006].

Neste contexto, este trabalho apresenta uma ferramenta de software para apoio a um método de determinação semi-automatizada de equivalências semânticas entre esquemas GML distintos. Seu propósito é validar na prática o método proposto por uma dissertação de mestrado em desenvolvimento no Grupo de Banco de Dados da UFSC [FROZZA 2006]. Dessa forma seus conceitos poderão ser efetivamente testados através da utilização de esquemas GML reais. A escolha de esquemas GML referentes ao domínio de cadastro urbano foi feita por se tratar de um domínio muito utilizado no planejamento da infra-estrutura das cidades. Esse domínio também é pouco explorado em outros trabalhos.

Em trabalhos relacionados, [BRAUNER 2004] introduz um conceito de mapeamento entre duas ontologias para criar um catálogo consolidado de objetos geográficos independentes, utilizando uma ontologia de referência. Já [MOROCHO 2003] apresenta uma ferramenta de integração semântica para dados espaciais. Em comum, abordam a interoperabilidade semântica entre SIGs em ambientes fortemente acoplados, dando ênfase na transformação de consultas. Este trabalho, por outro lado, enfatiza a questão da integração de dados, contribuindo para a interoperabilidade entre SIGs distintos.

Este trabalho possui outros 5 capítulos. O capítulo 2 apresenta os conceitos básicos da GML, seu uso em SIGs e discute a interoperabilidade entre esquemas GML. O capítulo 3 apresenta trabalhos relacionados e suas limitações em relação a esse trabalho. O capítulo 4 descreve o método de determinação de equivalências proposto. O capítulo 5 apresenta uma visão geral da ferramenta, seu funcionamento e sua interface. Por fim, o capítulo 6 apresenta as considerações finais e sugestões para trabalhos futuros.

## 2 Conceitos Relacionados

### 2.1 SIG e GML

Um SIG pode ser definido como um sistema computacional capaz de coletar, armazenar, gerenciar, manipular, recuperar, transformar e visualizar dados geo-referenciados, com habilidade para integrar grandes quantidades de informação e com capacidade de utilizar ferramentas analíticas para explorar estes dados [SILVA 2003].

A linguagem GML é uma gramática XML para a modelagem, transporte e armazenamento de informações geográficas [GML 2004]. A GML inclui uma variedade de objetos para descrição geográfica, tais como sistemas de coordenadas, geometria, topologia, unidades de tempo e medida.

```
<element name="LoteCTM" type="rc:LoteCTMType"
substitutionGroup="rc:_LoteCTM"/>
<complexType name="LoteCTMType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="LoteCTM" type="string"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

**Figura 1. Fragmento de documento GML**

Na Figura 1 é mostrado um exemplo de documento GML. Ele apresenta um elemento complexo de nome *LoteCTM*, que é do tipo *LoteCTMType*. Esse elemento é uma representação para um tipo de lote urbano no mundo real.

## **2.2 A importância da GML**

Por ser um padrão aberto, a GML tem sido muito utilizada como meio de armazenamento de dados geográficos de diferentes SIGs, da mesma forma como a XML é utilizada por uma grande variedade de aplicativos e sistemas das áreas mais diversas como formato para armazenamento de dados e informações.

A GML fornece um formato neutro tanto de fabricantes quanto de implementações. Esse formato é otimizado para a distribuição em rede. Tanto arquivos XML simples quanto arquivos XML comprimidos podem ser enviados de maneira que um usuário não precise esperar para descarregar um arquivo inteiro antes de abri-lo. Isso aumenta muito a usabilidade em um ambiente distribuído [PRINS 2003].

Existem muitas aplicações disponíveis para trabalhar com arquivos XML assim como com dados em GML. A GML padronizou a maneira com que informações geográficas são descritas. As definições da GML são descritas utilizando a recomendação XML Schema [XML SCHEMA 2007]. Arquivos XML Schema especificam as regras e o formato usado para codificar características espaciais abstratas, como linhas e polígonos em GML. É possível definir um esquema para aplicações específicas ou um documento de definição de tipos mais adequado a seu modelo de dados, estendendo os esquemas GML disponíveis.

## **2.3 Interoperabilidade entre Esquemas GML**

Informações geográficas são abstrações de fenômenos do mundo real. Elas são representações digitais de alguma parte do mundo real. Como tal,

estão sujeitas à cultura (regionalismos), ao domínio dos programadores sobre o tema, entre outros [CASANOVA 2005].

Diferenças a nível semântico entre esquemas GML dizem respeito aos aspectos relacionados à compreensão e utilização de seus dados, em diferentes aplicações, diferentes modelos de dados e diferentes interpretações desses dados.

O nível semântico da integração abrange a questão da unificação da nomenclatura utilizada para representar os fenômenos da realidade a serem modelados e os relacionamentos entre eles. Neste sentido, é necessário criar uma estrutura de organização do conhecimento, tal como um vocabulário controlado, uma taxonomia, um thesaurus ou uma ontologia [FROZZA 2006].

A opção do uso de uma ontologia deve-se exatamente pela maior facilidade de comparação entre dois esquemas GML se utilizarmos como modelo de comparação esta estrutura padronizada.

## **2.4 O Uso da Ontologia na Integração**

Em computação, uma ontologia pode ser encarada como um modelo de dados que representa um domínio. É utilizada para abstrair sobre os objetos deste domínio e as relações entre eles [WIKI 2006].

Segundo [DUARTE 2000], uma ontologia é uma descrição de conceitos e relações que existem em um domínio de interesse. Basicamente, uma ontologia consiste destes conceitos e relações, e suas definições, propriedades e restrições, descritas na forma de axiomas. Ontologias são úteis para apoiar a especificação e implementação de qualquer sistema de computação complexo. Uma ontologia pode ser desenvolvida para vários fins, mas, de uma forma geral, os seguintes propósitos são atingidos:

- Ajuda as pessoas a compreender melhor certa área de conhecimento: no desenvolvimento de uma ontologia, as pessoas envolvidas no processo se vêem diante de um desafio: explicar seu entendimento sobre o domínio em questão, o que as faz refletir e melhorar sua compreensão sobre esse domínio;
- Ajuda as pessoas a atingir um consenso em seu entendimento sobre uma área de conhecimento: geralmente, em uma determinada área de conhecimento, diferentes especialistas têm entendimento diferenciado sobre os conceitos envolvidos, o que gera problemas na comunicação. Ao se construir uma ontologia, essas diferenças são explicitadas e busca-se um consenso sobre seu significado e sua importância;
- Ajuda outras pessoas a compreender certa área de conhecimento: uma vez que haja uma ontologia sobre uma determinada área de conhecimento desenvolvida, uma pessoa que deseje aprender mais sobre essa área não precisa se reportar sempre a um especialista. Ela pode estudar a ontologia e aprender sobre o domínio em questão, absorvendo um conhecimento geral e de consenso.

Uma linguagem de definição de ontologias bastante utilizada atualmente é a OWL (Web Ontology Language) [OWL 2004]. Ela permite que informação semântica, contida em documentos OWL, possa ser processada por aplicações, ao invés de somente ser apresentada para visualização. A OWL pode ser utilizada para representar o significado de termos em vocabulários e o relacionamento entre estes termos.

```
<owl:Class rdf:ID="Servicos">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Edificado"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Edificado">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Lote"/>
  </rdfs:subClassOf>
</owl:Class>
```

**Figura 2. Fragmento de uma ontologia OWL**

A Figura 2 apresenta um fragmento de um esquema OWL. Nele é possível visualizar classes e subclasses. Nesse exemplo, a classe *Servicos* é subclasse da classe *Edificado*. E, ainda, *Edificado* é uma subclasse da classe *Lote*.

Essa linguagem também possui um suporte muito melhor à interpretação de conteúdo Web do que outras linguagens com abordagem semelhante como XML, RDF e RDF *Schema* (RDF-S). Isso é conseguido através de vocabulário adicional juntamente com uma semântica formal.

### 3 Trabalhos Relacionados

#### 3.1 Unificação Semântica de Esquemas Conceituais de BDGs

Em [HESS 2004] há uma proposta para solução de conflitos entre esquemas conceituais de bancos de dados geográficos, suportada por uma arquitetura de software, com duas etapas: uma sintática que deve converter os esquemas para o formato GML e a outra semântica que utiliza uma ontologia para a integração semântica. Esse modelo de resolução de conflitos entre esquemas conceituais de BDGs é proposto para ser universal, de forma a aceitar esquemas conceituais de diferentes domínios geográficos. Para solucionar os conflitos entre esquemas conceituais foi desenvolvida uma metodologia, suportada por uma arquitetura de software, a qual divide a fase de pré-processamento em duas etapas, uma sintática e uma semântica. A fase sintática visa converter os esquemas em um formato canônico, a Geographic Markup Language (GML). Para cada um dos diferentes modelos de dados um conjunto de regras foi desenvolvido e um *wrapper* implementado. Para suportar a etapa semântica da integração uma ontologia é utilizada para integrar semanticamente os esquemas conceituais dos diferentes projetos.

Por propor o tratamento de esquemas conceituais de diversos domínios geográficos, a metodologia proposta tende a apresentar diferentes níveis de eficácia em relação ao domínio escolhido. Dessa forma, os parâmetros de suas métricas necessitam de calibração constante. Sua ontologia também precisa ser completada conforme novos esquemas conceituais são processados.

Por outro lado, a ferramenta proposta nesse trabalho define bem seu domínio. Portanto, ela não apresenta as mesmas limitações em relação à



ontologia (já definida para o domínio em uso) e os níveis de ajuste dos pesos das métricas.

### **3.2 Interoperabilidade na representação de dados geográficos**

Uma abordagem para a interoperabilidade na representação de dados geográficos é feita por [SILVA 2003]. Seu estudo tem como objetivo analisar e validar a hipótese da viabilidade da integração entre duas soluções para representação de dados geográficos: GeoBR/INPE e GML 3.0/OGC. Esses modelos de representação foram escolhidos em virtude de seu uso no INPE, para fins de realização de um estudo de caso. O tratamento do tempo na representação da informação geográfica foi abordado e explorado dentro do contexto do caso estudado, utilizando os recursos da especificação GML 3.0 e estendida ao formato GeoBR. Para a validação do trabalho, ferramentas de conversão e visualização foram desenvolvidas e adaptadas, em particular adaptações no conversor TerraTranslator, no aplicativo geográfico para Internet denominado SpringWeb, e um demonstrativo do uso de SVG para dados animados.

Sua abordagem apresenta a possibilidade de integração entre duas soluções de representação de dados geográficos distintas. Nesse caso, ela fica restrita a modelos que utilizem somente os dois padrões apresentados. Como a ferramenta apresentada neste trabalho propõe a integração entre dois esquemas GML, ela é mais abrangente, pelo fato da GML ser um modelo mais difundido. Além disso, também utiliza padrões mais recentes da GML (3.1.1) do que os utilizados por [SILVA 2003].

### **3.3 Catálogos de Objetos Geográficos para Interoperabilidade entre BDGs**

[BRAUNER 2004] introduz um conceito de mapeamento entre duas ontologias para criar um catálogo consolidado de objetos geográficos independentes, utilizando uma ontologia de referência.

O conceito proposto é chamado de OGOC (Ontology-based Geo-Object Catalog) e estabelece uma estratégia para endereçar o problema de interoperabilidade entre fontes de objetos geográficos. Em seu trabalho, [BRAUNER 2004] considera que objetos geográficos, ou geo-objetos, são todos os tipos de dados que possuem alguma informação sobre sua localização espacial. O OGOC atuaria como um mediador genérico para várias fontes de geo-objetos, provendo serviços para o acesso e busca de dados e metadados. De forma a cumprir este requisito, o catálogo guardaria uma ontologia de referência, ontologias locais que descrevem fontes de objetos e mapeamentos entre a ontologia de referência e as ontologias locais. Ainda guardaria conjuntos de instâncias de geo-objetos e mapeamentos entre geo-objetos de referência e geo-objetos armazenados em cada fonte.

Ao contrário do que propõe [BRAUNER 2004], o trabalho aqui proposto enfatiza um domínio bem definido.

### **3.4 Ferramenta para Integração Semântica de Bancos de Dados Espaciais**

Em [MOROCHO 2003] é apresentada uma ferramenta de integração semântica para dados espaciais. Sua motivação é a variedade de estudiosos

de áreas como clima, ambiente, desenvolvimento sustentado, que podem se beneficiar através da integração de seus diversos dados geográficas.

O trabalho é focado no domínio dos SIGs e Bancos de Dados Espaciais. O foco em um domínio se deve à impossibilidade de integrar uma vasta quantidade e diversidade de dados geográficos sem perder informações. Utilizando padrões de conteúdo para aplicações geográficas, [MOROCHO 2003] desenvolveu um protótipo para a integração de fontes de dados espaciais heterogêneas.

Diferentemente deste trabalho, a proposta de [MOROCHO 2003] faz uso de modelos XMI (*XML Metadata Interchange*) para obtenção dos dados a serem integrados.

## 4 Método para Determinação das Equivalências Semânticas entre Esquemas GML

A ferramenta desenvolvida neste trabalho implementa um método para a determinação de equivalências semânticas entre dois esquemas GML distintos com o apoio de ontologias OWL. Este método é proposto por [FROZZA 2006] especificamente para o domínio do cadastro urbano. O método é semi-automático pois prevê a intervenção de um usuário especialista no domínio para validar as equivalências encontradas automaticamente entre dados geográficos.

A Figura 3 a seguir mostra uma visão geral do funcionamento do método.

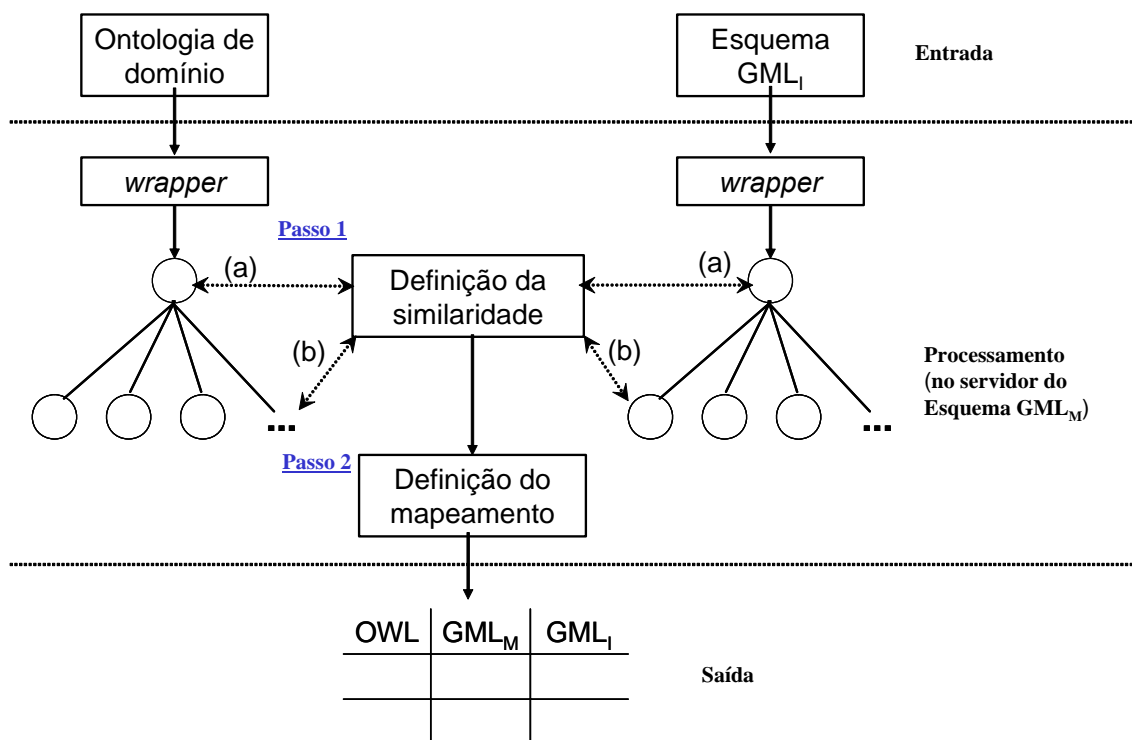


Figura 3. Visão geral do método

O esquema principal (GML<sub>M</sub>) representa os dados de um SIG local (GML<sub>M</sub>) e o outro esquema representa dados importados de um segundo SIG

remoto (GML<sub>I</sub>). O método utiliza, ainda, uma ontologia que representa os conceitos geográficos do domínio considerado.

O método é executado no servidor principal, considerando que já existe um mapeamento entre o esquema GML<sub>M</sub> e a ontologia de domínio. A entrada do método corresponde à ontologia de domínio, definida em OWL, e ao esquema GML<sub>I</sub>, os quais são transformados para um formato canônico (estrutura de dados em árvore) por *wrappers* específicos.

O método possui dois passos de execução. No primeiro passo é feita a determinação de equivalências semânticas entre o GML<sub>I</sub> e a ontologia de domínio. Esse passo analisa os elementos presentes em ambos os esquemas (GML<sub>I</sub> e ontologia) e determina os graus de similaridade semântica entre eles. Cada grau de similaridade é definido por métricas que seguem a classificação proposta por [DORNELES 2004]: métricas para valores complexos (MCV), adequados a elementos XML compostos, e métricas para valores atômicos (MAV), adequadas a valores de elementos simples e atributos XML.

No segundo passo, as equivalências semânticas encontradas entre os elementos do esquema GML<sub>I</sub> e a ontologia de domínio são mapeadas para os elementos equivalentes no esquema GML<sub>M</sub>. Para isso, é usado o mapeamento já existente entre a ontologia e o esquema GML<sub>M</sub>. Para armazenar esses mapeamentos, foi definido um catálogo relacional.

Maiores detalhes sobre os formatos canônicos, as métricas de similaridade e a catalogação dos mapeamentos são encontrados no próximo capítulo, que descreve a ferramenta.

## 5 Descrição da Ferramenta

### 5.1 Implementação

A ferramenta está implementada na linguagem Java (utilizada devido a sua alta portabilidade), através do ambiente de desenvolvimento Netbeans 5.0. A implementação está organizada em classes que foram criadas de forma a separar as diversas partes do método utilizado, facilitando a compreensão do código e a adição posterior de novas funcionalidades.

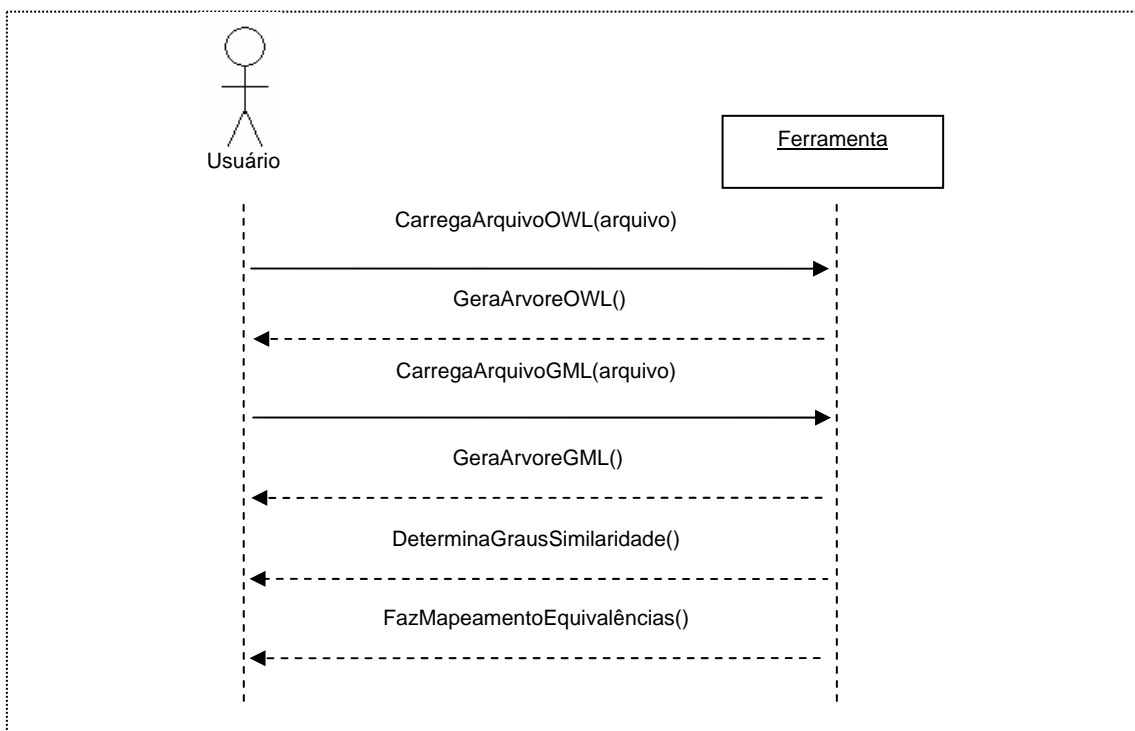


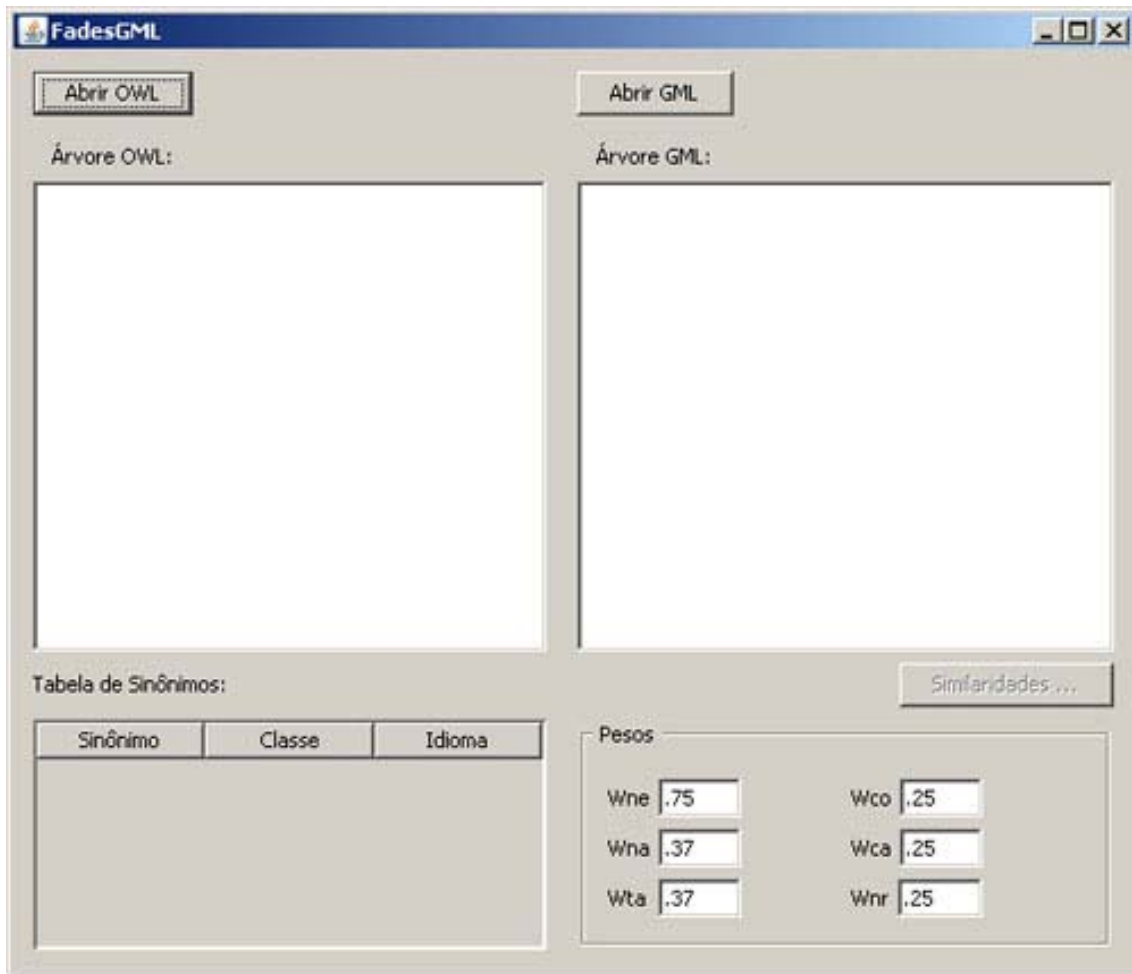
Figura 4. Diagrama de seqüência da ferramenta

A Figura 4 apresenta um diagrama de seqüência referente ao funcionamento da ferramenta. As funções apresentadas no diagrama são as seguintes:

- *CarregaArquivoOWL(arquivo)*: realiza a leitura e validação do arquivo correspondente à ontologia;

- *GeraArvoreOWL()*: cria e exibe a árvore com a estrutura da ontologia;
- *CarregaArquivoGML(arquivo)*: realiza a leitura e validação do arquivo correspondente ao GML<sub>I</sub>;
- *GeraArvoreGML()*: cria e exibe a árvore com a estrutura da GML<sub>I</sub>;
- *DeterminaGrausSimilaridade()*: aplica métricas de similaridade (ver capítulo 5.4) entre os elementos da ontologia e do GML<sub>I</sub>;
- *FazMapeamentoEquivalencias()*: retorna o mapeamento entre os elementos do esquema GML<sub>I</sub> e os elementos do esquema GML<sub>M</sub>.

Na Figura 5 é exibida a janela principal da ferramenta. Cada botão representa uma das funções utilizadas na interação do usuário com a ferramenta, mostradas na Figura 4.



**Figura 5. Janela principal da ferramenta**

Inicialmente, o usuário escolhe uma dentre duas opções: “Abrir OWL”, que corresponde à função *CarregaArquivoOWL()*, ou “Abrir GML”, correspondente à função *CarregaArquivoGML()*. Nestes dois casos, o sistema invoca a função para gerar a árvore (formato canônico) correspondente à opção escolhida pelo usuário.

Na geração da árvore da ontologia, uma tabela de sinônimos é gerada e preenchida. Essa tabela é parte da ontologia, estando definida em uma classe chamada *Dicionário*. Os sinônimos são armazenados como instâncias dessa classe. Ela é utilizada para manter a lista dos sinônimos dos conceitos presentes na ontologia. Após ter as duas árvores geradas, o usuário seleciona



o botão “*Similaridades...*” que invoca as funções *DeterminaGrausSimilaridade()* e *FazMapeamentoEquivalencias()*.

## 5.2 Árvores de Comparação

Como explicado no capítulo 4, o método requer como entrada uma ontologia referente ao domínio abordado e um esquema GML (GML<sub>I</sub>). Tanto a ontologia quanto o GML<sub>I</sub> devem passar por um pré-processamento em que são convertidos para um formato canônico. No caso, foi definida uma estrutura em árvore (árvore canônica), em que cada classe da ontologia e cada elemento do GML<sub>I</sub> são representados como um nodo (exemplos são mostrados na Figura 6). Esta forma de visualização se mostra adequada à representação de dados e esquemas XML [DORNELES 2004], tornando mais fácil o processamento.

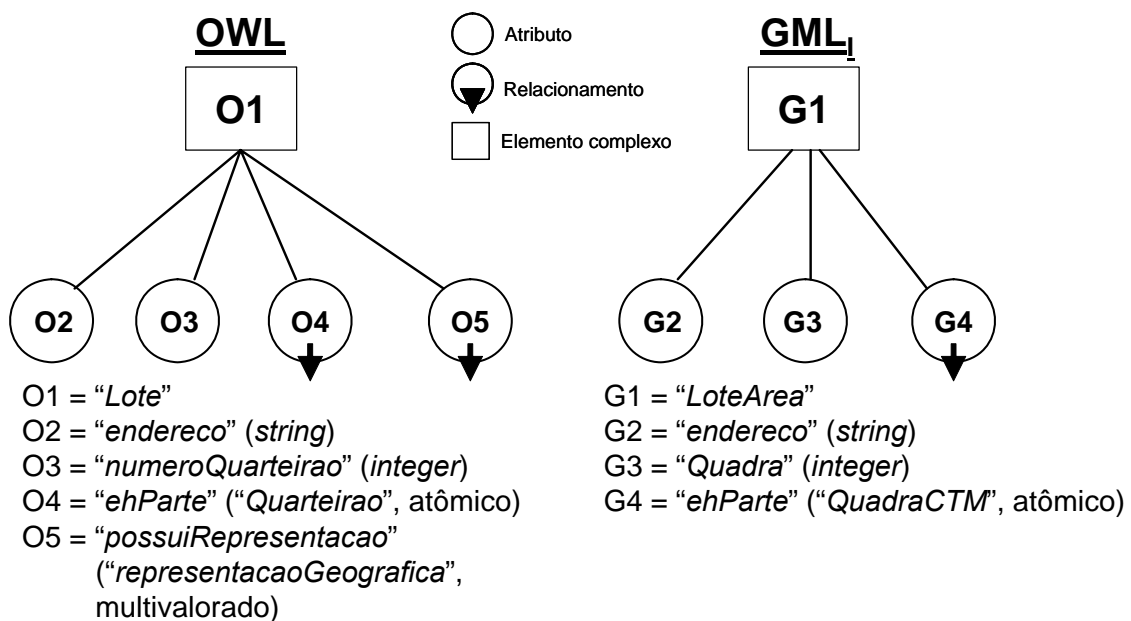


Figura 6. Exemplo de nodo para um elemento OWL e para um elemento GML na estrutura de árvore canônica

A estrutura das árvores canônicas possui a seguinte definição:

- *Elemento complexo*: identificador do nodo (identificador da classe ou do elemento GML);
- *Propriedades do nodo (elemento)*: atributos simples (*strings*, números etc.), atributos complexos (formados por outros atributos) e relacionamentos. Atributos e relacionamentos ainda podem receber um ou mais valores.

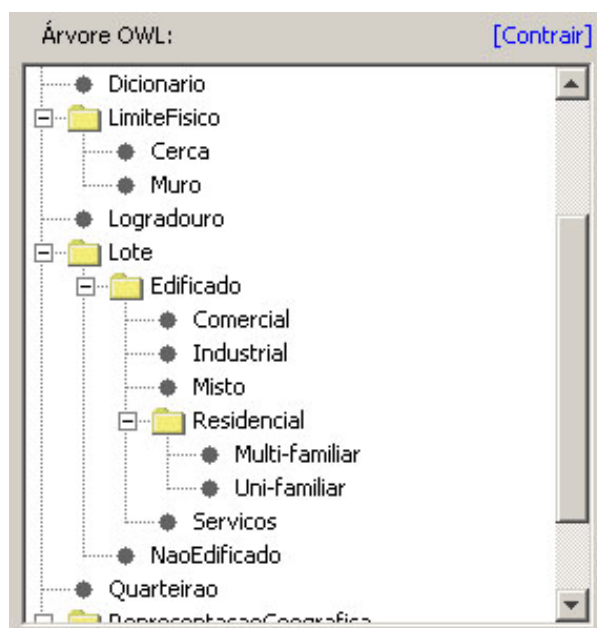
Como pode ser visto no exemplo da Figura 6, existe um elemento OWL chamado “*Lote*”, que tem dois atributos simples (“*endereco*” do tipo *string*; “*numeroQuarteirao*” do tipo *integer*) e dois relacionamentos (“*ehParte*”, relacionado ao conceito “*Quarteirao*” e com cardinalidade 1:1; “*possuiRepresentacao*”, relacionado com o conceito “*representacaoGeografica*” e com cardinalidade 1:n). De forma semelhante, o elemento GML “*LoteArea*” tem dois atributos simples (“*endereco*” do tipo *string*; “*Quadra*” do tipo *integer*) e um único relacionamento “*ehParte*” (que se relaciona com o elemento GML “*QuadraCTM*”, com cardinalidade 1:1).

Na ferramenta desenvolvida, os arquivos de entrada são abertos por módulos *wrappers* (classes *GeraArvoreOWL()* e *GeraArvoreGML()*), que convertem as estruturas OWL e GML para as árvores canônicas e as apresenta ao usuário. As duas árvores são apresentadas utilizando o componente *JTree* do Java

### 5.2.1 *Wrappers* das Árvores OWL e GML

Na montagem da árvore canônica da ontologia, cada classe é representada como um nodo  $NC_i$  da árvore e seus atributos e relacionamentos são representados como nodos filhos de  $NC_i$  [FROZZA 2006]. A Figura 7

mostra um fragmento da árvore canônica de uma ontologia OWL gerada pelo *wrapper* OWL.



**Figura 7. Fragmento de árvore canônica para uma ontologia OWL**

O processamento da ontologia para a geração da árvore é feito utilizando-se o *framework Jena*. Ele fornece um ambiente de programação em *Java* próprio para o desenvolvimento de aplicações de *Web Semântica*.

Já para o processamento da árvore canônica GML, optou-se por utilizar o *framework JDOM*. Ele proporciona uma biblioteca em *Java* para a manipulação de dados em arquivos XML em geral. Sua escolha foi feita por ser um projeto maduro para o tratamento do formato XML e pela falta de bibliotecas específicas em *Java* para o tratamento de arquivos GML.

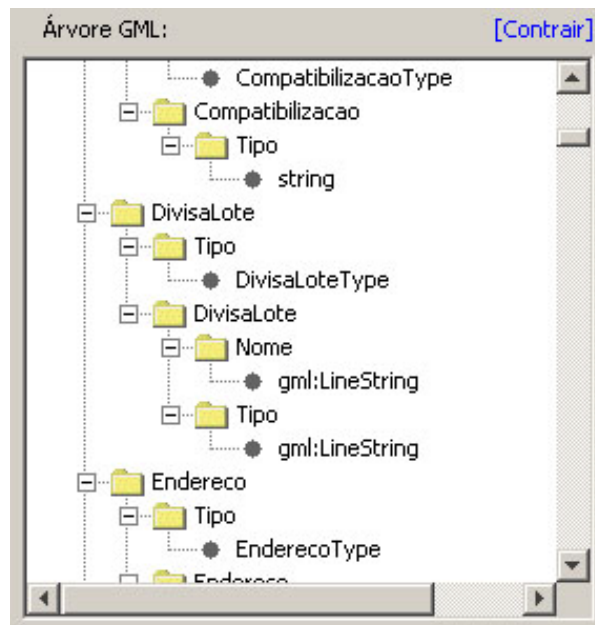
Além da montagem da árvore canônica, a classe *wrapper* correspondente à OWL cria uma lista de sinônimos a partir das instâncias de objetos de uma classe “*Dicionario*” na ontologia. Na Tabela 1 é possível visualizar um exemplo de uma lista de sinônimos correspondente a uma ontologia hipotética, no formato *sinônimo – classe – idioma*. Cada elemento dessa lista (coluna 1)

corresponde a um sinônimo de uma classe (coluna 2) na ontologia. Um sinônimo pode ser o próprio nome da classe da ontologia ou um outro sinônimo qualquer para esta classe. A indicação do idioma é utilizada na definição das métricas aplicadas pelo método.

**Tabela 1. Exemplo da lista de sinônimos**

SINÔNIMO	CLASSE	IDIOMA
Lote	Lote	pt
Parcel	Lote	en
Quadra	Quarteirao	pt
Quarteirao	Quarteirao	pt

A árvore canônica para o GML<sub>i</sub> é muito semelhante à árvore gerada para a ontologia, ou seja, cada elemento do esquema é convertido em um nodo  $NE_i$  da árvore. Seus atributos e relacionamentos são convertidos em nodos folha de  $NE_i$ . Um fragmento de uma árvore GML gerada é mostrado na Figura 8. Nela é possível visualizar nodos principais (como *DivisaLote*) e nodos folha (como *DivisaLoteType* e *gml:LineString*).



**Figura 8. Fragmento de árvore canônica para um esquema GML**

### 5.3 Determinação da Equivalência Semântica

Para fazer a determinação da equivalência semântica dos elementos, [FROZZA 2006] propõe vários passos. Cada nodo  $NE_i$  da árvore do esquema  $GML_i$  é testado contra cada nodo  $NC_j$  da ontologia para identificar se existe algum elemento  $NE_i$  idêntico ou similar representado na ontologia.

O identificador (nome) do elemento  $NE_i$  inicialmente é testado por igualdade contra a tabela de sinônimos (Tabela 1). Se um ou mais sinônimos correspondentes forem encontrados (por exemplo:  $NE_i = \text{“Lote”}$  e sinônimo =  $\text{“Lote”}$ ), uma métrica de similaridade de estrutura (métrica MCV) é aplicada em cada par  $NE_i - NC_j$ . Se nenhum sinônimo idêntico ao nome de  $NE_i$  for encontrado, uma nova busca é feita na tabela de sinônimos, agora aplicando uma *métrica de similaridade de nomes* (métrica MAV). Se forem encontradas similaridades de acordo com um grau aceitável (*threshold*), então a métrica de similaridade de estruturas é aplicada a todos os pares encontrados (por exemplo:  $NE_i = \text{“Lote”}$  e sinônimo =  $\text{“LoteCTM”}$ ). A definição de quê valor representa um grau de similaridade aceitável (*threshold*) para uma métrica deve ser configurada no início do processo pelo usuário.

Essas métricas retornam um valor de equivalência entre 0 e 1. A aceitação dos graus de similaridade encontrados é feita pelo usuário, que deve validar as equivalências geradas como forma de coibir eventuais validações incorretas geradas no processo. A Figura 9 mostra um exemplo de validação de equivalências exibido pela ferramenta.



**Figura 9. Exemplo da tela de Validação de Equivalências**

Ao final do processo, eventualmente podem ser encontrados mais de um par  $NE_i-NC_j$  com graus de similaridade muito próximos. Ao mesmo tempo, alguns elementos no esquema  $GML_i$  podem não ser reconhecidos pelo método e, por consequência, assumidos como não existentes na ontologia. Nestes casos, o usuário é chamado a interagir com a ferramenta, identificando qual é o par  $NE_i-NC_j$  equivalente ou se realmente o elemento  $NE_i$  não está representado na ontologia e deve ser desconsiderado.

#### **5.4 Métricas de Similaridade**

O método adapta dois tipos de métricas propostas por [DORNELES 2004] para a determinação dos graus de similaridade semântica entre os elementos do esquema  $GML_i$  e a ontologia de domínio.

As métricas do tipo MCV (*Métricas para Valores Complexos*) são aplicadas à estrutura dos dados. Para o método proposto, somente são encontradas estruturas semelhantes a tuplas. As métricas do tipo MAV (*Métricas para Valores Atômicos*) são aplicadas a conteúdos simples, como

*strings* e números. Elas são dependentes do domínio da aplicação, já que consideram as características dos dados da aplicação.

Este conjunto de métricas foi escolhido em [FROZZA 2006] por apresentar uma taxonomia adequada ao tratamento de dados XML. No caso, um elemento XML é tratado como nodo de uma árvore, considerando que pode ser atômico ou complexo. Elementos atômicos contêm valores simples, como uma cadeia de caracteres, uma data etc. Elementos complexos correspondem a estruturas formadas por outros elementos, atômicos ou complexos.

#### 5.4.1 Métricas para Valores Complexos

Três tipos de elementos complexos são considerados pelo método, sendo que para cada tipo há uma métrica MCV específica implementada pela ferramenta:

- Nodos  $NE_i$  e  $NC_j$ : têm um identificador e diversas propriedades que podem ser definições de atributos ou de relacionamentos. A similaridade dos pares  $NE_i-NC_j$  é dada pela Equação 1:

$$tupleSim(\varepsilon_p, \varepsilon_d) = \frac{\sum_{\varepsilon_p^i \cdot \eta = \varepsilon_d^j \cdot \eta} (sim(\varepsilon_p^i, \varepsilon_d^j))}{\max(m, n)} \quad \text{Eq. 5.1}$$

sendo:

- $P$ : conjunto de nodos  $NE_i$  da árvore do esquema GML<sub>i</sub>;
- $D$ : conjunto de nodos  $NC_j$  da árvore da ontologia;
- $\varepsilon_p$ : um nodo do conjunto  $P$  ( $NE_i$ );
- $\varepsilon_d$ : um nodo do conjunto  $D$  ( $NC_i$ );
- $n$  e  $m$ : número de nodos filhos de  $\varepsilon_p$  e  $\varepsilon_d$ , respectivamente.

Na métrica  $tupleSim()$  cada nodo filho  $\varepsilon_p^j$  de  $\varepsilon_p$  é comparado a um nodo filho  $\varepsilon_d^i$  de  $\varepsilon_d$  com mesmo nome ( $\varepsilon_p^j.\eta = \varepsilon_d^i.\eta$ ) e mesmas características.

A função  $max()$  retorna o maior número de nodos filho entre  $\varepsilon_p$  e  $\varepsilon_d$ .

- Definições de atributos simples: são compostas por um identificador e um tipo de dado:

$$attrSim(\varepsilon_p.\eta, \varepsilon_d.\eta) = \frac{nameSim(\varepsilon_p, \varepsilon_d) + typeSim(\varepsilon_p, \varepsilon_d)}{2} \quad \text{Eq. 5.2}$$

sendo:

- $\varepsilon_p$ : nodo filho do elemento da árvore GML;
  - $\varepsilon_d$ : nodo filho do elemento da árvore OWL;
  - $nameSim$ : similaridade dos nomes dos atributos;
  - $typeSim$ : similaridade dos tipos de dados.
- Definições de relacionamentos: são compostas pelo nome do relacionamento, o conceito relacionado e a cardinalidade:

$$relSim(\varepsilon_p.\eta, \varepsilon_d.\eta) = \frac{nameSim(\varepsilon_p, \varepsilon_d) + concSim(\varepsilon_p, \varepsilon_d) + cardSim(\varepsilon_p, \varepsilon_d)}{3} \quad \text{Eq. 5.3}$$

sendo:

- $nameSim$ : similaridade dos nomes dos relacionamentos;
- $concSim$ : similaridade dos conceitos referenciados pelos relacionamentos;
- $cardSim$ : similaridade das cardinalidades.

A métrica  $tupleSim()$  deve ser executada em duas etapas. Na primeira etapa consideram-se apenas os nodos filhos que correspondam a atributos de  $\varepsilon_p$  e  $\varepsilon_d$ , para os quais é aplicada a métrica  $attrSim()$ . A somatória das similaridades dos atributos gera um valor provisório de similaridade dos nodos. Na segunda etapa, são calculadas as similaridades para os nodos filhos que correspondam a relacionamentos, usando a métrica  $relSim()$ . Nesta métrica, o



valor passado para a função *concSim()* será o valor provisório obtido na etapa anterior. Essa seqüência de execução é necessária para evitar situações de *deadlock* existentes nas dependências entre os relacionamentos.

As métricas para valores atômicos são usadas para calcular a similaridade de nomes, tipos de dados e cardinalidades. As duas últimas métricas usam uma tabela de equivalência de tipos de dados (Tabela 2) e de cardinalidades (Tabela 3), respectivamente.

**Tabela 2. Exemplo de equivalências entre tipos de dados**

<b>OWL</b>	<b>GML</b>	<b>Grau similaridade</b>
Inteiro	<i>Integer</i>	1,0
Inteiro	<i>String</i>	0,5
Data	<i>Date</i>	1,0
<i>String</i>	<i>String</i>	1,0
Ponto	<i>Point</i>	1,0
Linha	<i>Curva</i>	1,0
Linha	<i>LineString</i>	1,0
Linha	<i>OrientableCurve</i>	1,0
Linha	<i>CompositeCurve</i>	1,0
Polígono	<i>Solid</i>	1,0
Polígono	<i>CompositeSolid</i>	1,0
Polígono	<i>Polygon</i>	1,0

**Tabela 3. Mapeamento entre cardinalidades de classes OWL e elementos GML**

Classe OWL	Elemento GML	Similaridade
<i>single</i>	<i>maxOccurs = 1</i>	1,0
<i>single</i>	<i>maxOccurs &gt; 1</i>	0,5
<i>multiple</i>	<i>maxOccurs = 1</i>	1,0
<i>multiple</i>	<i>maxOccurs &gt; 1</i>	1,0

#### 5.4.2 Métricas para Valores Atômicos

Algumas métricas de similaridade de *strings* são encontradas na literatura, como as métricas *Jaro Winkler*, distância de *Levenshtein* e distância de *Hamming* [CHAPMAN 2006]. O método considera as métricas *Jaro Winkler* e *Levenshtein*. Na ferramenta, pode-se selecionar automaticamente a melhor métrica a ser utilizada para definir a similaridade. Assim, o usuário fica livre para escolher a melhor métrica em cada situação. Por exemplo, para palavras na língua portuguesa, a melhor escolha em geral é a métrica *Jaro Winkler*. Para palavras em língua inglesa, a distância de *Levenshtein* é a melhor opção.

#### 5.4.3 Exemplo de Determinação de Similaridade

Um exemplo de determinação de similaridade gerada pela ferramenta considera os dados da Figura 6. Neste caso, aplicando as métricas *attrSim()* e *relSim()* aos elementos *Lote* e *LoteArea* para obter as similaridades entre atributos e relacionamentos, respectivamente, tem-se:

- $sim2 = attrSim(G2, O2) = 1$
- $sim3 = attrSim(G3, O3) = 0,78$
- $sim4 = relSim(G4, O4) = 0,89$

Na seqüência, determina-se o grau de similaridade final entre os dois elementos:

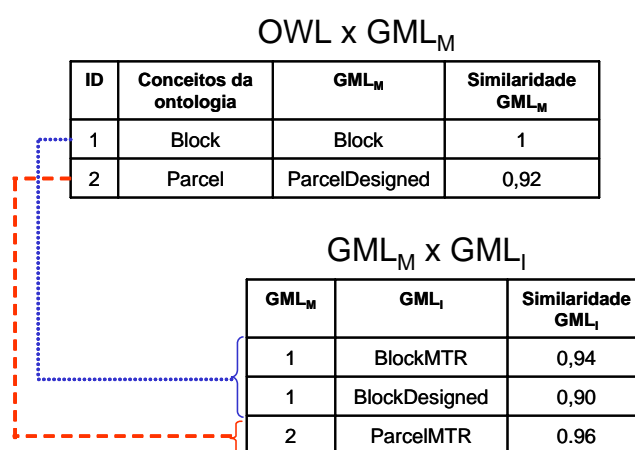
$$sim1 = tupleSim() = (1 + 0,78 + 0,89) / 4 = 0,67$$

Ao final, o valor obtido para a métrica *tupleSim()* precisa ser validado. Neste caso, a intervenção do usuário é prevista na ferramenta. Uma tela de validação é exibida e o usuário confirma as similaridades, como foi exemplificado na seção 5.3.

## 5.5 Catalogação dos Mapeamentos

Uma vez determinadas e confirmadas as equivalências semânticas entre o esquema GML importado e a ontologia, a ferramenta cataloga, em um banco de dados relacional de informações de mapeamento, quais elementos do GML<sub>I</sub> correspondem semanticamente a elementos do GML<sub>M</sub>. O Sistema Gerenciador de Banco de Dados (SGBD) escolhido para a catalogação dos mapeamentos foi o *PostgreSQL* em sua versão mais atual (8.2). Ele é um SGBD de código aberto e está disponível em muitas plataformas, como *Windows* e *Linux*. Ele também possui as características mais avançadas encontradas em SGBDs comerciais, sendo de fácil integração com a plataforma *Java*.

A Figura 10 mostra o esquema do banco de dados relacional definido.



**Figura 10. Catalogação dos mapeamentos**

A tabela superior na Figura 10 armazena as equivalências entre os conceitos da ontologia e do GML<sub>M</sub>, assim como seus graus de similaridade. A

tabela inferior mantém as correspondências encontradas entre elementos do  $GML_I$  e do  $GML_M$  (referenciado da primeira tabela) e seus graus de similaridade. As equivalências encontradas devem apresentar um grau mínimo de similaridade previamente definido para que façam parte da segunda tabela de catalogação.

## 6 Estudo de Caso

Um estudo de caso foi realizado com a ferramenta de forma de validar o seu uso. Para isso, foram utilizados uma ontologia padrão (Anexo 2 - A) e três esquemas GML. Para todos os esquemas testados, foram utilizados os seguintes pesos padrão:

- Peso de nome de elementos ( $W_{ne}$ ) = 0,8
- Peso de nome de atributos ( $W_{na}$ ) = 0,6
- Peso de tipo de atributos ( $W_{ta}$ ) = 0,4
- Peso de nome de relacionamentos ( $W_{nr}$ ) = 0,4
- Peso de cardinalidade de relacionamentos ( $W_{ca}$ ) = 0,2
- Peso de conceitos de relacionamentos ( $W_{ca}$ ) = 0,4
- *Threshold* (peso final de validação): 0,7

O primeiro esquema GML utilizado foi o da prefeitura de Lages (Anexo 2 - B). Este esquema define diversos conceitos do domínio do cadastro urbano. Entretanto, este esquema não apresentou nenhuma similaridade com a ontologia, considerando o *threshold* utilizado. Isto ocorreu devido ao fato dos atributos e relacionamentos de seus elementos diferirem muito da ontologia, mesmo em elementos com nomes idênticos.

O segundo esquema GML testado foi o da cidade de Belo Horizonte, desenvolvido por uma empresa local (Anexo 2 - C). Seu conteúdo é mais restrito que o GML anterior, abrangendo apenas quadras e lotes. Esse esquema apresentou forte similaridade com a ontologia, como pode ser visto na Tabela 4.

**Tabela 4 – Similaridade entre elementos (Ontologia X GML Belo Horizonte)**

<b>Conceito</b>	<b>Elemento GML</b>	<b>Similaridade</b>
Quarteirao	QuadraCTM	0.8359366
Quarteirao	QuadraProjetada	0.76431894
Cerca	Cerca	0.94750625
Muro	Muro	0.71766

Um terceiro esquema GML foi criado a partir de um esquema exemplo para o domínio urbano definido em [CAMARA 1999]. Esse esquema também não apresentou similaridades com graus relevantes.

Uma validação manual dos resultados a partir dos pesos mencionados mostrou que a precisão alcançada pelas validações automáticas da ferramenta não foi totalmente satisfatória. Esperava-se principalmente que alguns relacionamentos entre tipos de lotes aparecessem no resultado final, o que não ocorreu.

Os pesos foram configurados de forma a dar mais importância à nomenclatura dos elementos (nomes geralmente denotam mais a semântica dos dados). O *threshold* (grau de similaridade) foi definido como 70%, pois entende-se que é um valor mínimo para se ter uma confiabilidade dos resultados.

As similaridades encontradas ficaram dentro do esperado quando da análise manual das árvores canônicas dos elementos envolvidos. Entretanto, estes resultados também permitiram concluir que é necessária uma evolução constante da ontologia, de modo a contemplar novos conceitos ou mesmo detalhar conceitos já existentes, para que se possam determinar mais similaridades verdadeiras entre elementos.

## 7 Conclusão

Este trabalho apresenta uma ferramenta para apoio à determinação semi-automatizada de equivalências semânticas entre esquemas GML distintos. A ferramenta implementa o método descrito por [FROZZA 2006], específico para a determinação deste tipo de equivalência, gerando os mapeamentos das equivalências encontradas. Vale ressaltar que, como o método enfatiza somente o domínio do cadastro urbano, muitas partes do código são específicas para o domínio em questão. Assim sendo, nesta primeira versão, a ferramenta não está apta a trabalhar com esquemas GML de outros domínios.

As soluções encontradas em trabalhos relacionados geralmente abrangem qualquer domínio de aplicação, mesmo quando a validação de equivalências é feita em um único domínio. Também tratam ambientes fortemente acoplados, com ênfase na transformação de consultas ao invés de processos de integração de dados.

A contribuição dessa ferramenta é o apoio à interoperabilidade de dados geográficos baseados em GML, considerando que GML é uma recomendação atual para representação de dados geográficos. O foco da ferramenta é em dados urbanos, sendo possível a sua aplicação em *softwares* que possibilitem a troca de informações entre bases de dados geográficos sobre uma determinada cidade, por exemplo. Além disso, a ferramenta utiliza a tecnologia de ontologias na determinação das similaridades entre dados GML, o que confere um maior caráter semântico nas comparações.

Para versões posteriores da ferramenta, existe a possibilidade de expandi-la de forma a abranger outros domínios que também utilizem esquemas GML. Conforme a necessidade e o volume dos dados utilizados, a ferramenta também poderá fazer uso de bancos de dados para a criação das tabelas de sinônimos. Conforme salientado no estudo de caso, também é importante o desenvolvimento de um mecanismo de evolução da ontologia, para que se possa considerar mais características do domínio envolvido, de modo a aumentar a precisão das similaridades e permitir uma maior interoperabilidade.

Também é interessante propor a implementação do método como um *web service*. Dessa forma, a ferramenta funcionaria como um aplicativo cliente-servidor, utilizando a *Web* como meio de acesso e possibilitando diferentes interfaces de consulta.



## 8 Referências Bibliográficas

[AZEVEDO 2006] Azevedo, V. H. M. et al. Interoperabilidade entre Objetos Geográficos Heterogêneos. In: GEOINFO, 8., 2006, Campos do Jordão. **Anais...** São José dos Campos: INPE, 2006. p. 235-245

[BRAUNER 2004] Brauner, D. F.; Casanova, M. A.; Lucena, C. J. P. Geo-Object Catalogs to enable Geographic Databases Interoperability. In: GEOINFO, 6., 2004, Campos do Jordão. **Anais...** São José dos Campos: INPE, 2004. p. 235-246

[CAMARA 1999] CÂMARA, G. et al. A. Interoperability In Practice: Problems in Semantic Conversion from Current Technology to OpenGIS. In: INTERNATIONAL CONFERENCE ON INTEROPERABLE OPERATING SYSTEMS, 2., 1999. **Proceedings...** Zurich, 1999. p. 120-138.

[CASANOVA 2005] CASANOVA, M. A. et al. **Bancos de Dados Geográficos.** Curitiba: MundoGEO, 2005.

[CHAPMAN 2006] Chapman, J. **Sam's Strings Metrics.** Disponível em: <<http://www.dcs.shef.ac.uk/~sam/stringmetrics.html>>. Acessado em: 25 mai. 2006

[DORNELES 2004] DORNELES, Carina F. et al. **Measuring Similarity between Collection of Values.** In: ACM INTERNATIONAL WORKSHOP ON

WEB INFORMATION AND DATA MANAGEMENT (WIDM), Nov. 12-13. Proceedings, Washington DC, USA. ACM, 2004. p 56-63.

[DUARTE 2000] DUARTE, Kátia C.;FALBO, Ricardo de A. **Uma Ontologia de Qualidade de Software**. Anais do VII Workshop de Qualidade de Software, João Pessoa, Paraíba, Outubro 2000. p. 275-285.

[FROZZA 2006] Frozza, Angelo A.; Mello, Ronaldo S. Um Método para Determinar a Equivalência Semântica entre Esquemas GML. In: GEOINFO, 8., 2006, Campos do Jordão. Anais... São José dos Campos: INPE, 2006. p. 283-293.

[GML 2004] COX, Simon et al. **OpenGIS Geography Markup Language (GML) Implementation Specification**. OpenGIS Recommendation Paper. 2004. 601p.

[HESS 2004] HESS, Guillermo Nudelman. **Unificação Semântica de Esquemas Conceituais de Bancos de Dados Geográficos**. Porto Alegre: UFRGS, 2004. 110p. Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul, Programa de Pós Graduação em Computação, Porto Alegre, 2004.

[JDOM 2006] HUNTER, Jason et al. **JDOM v1.0 API Specification**. Disponível em <<http://www.jdom.org/docs/apidocs/>> Acesso em jun. 2006.

[LAKE 2002] LAKE, R. **Building the Geo-web at the local, regional, national and global levels.** In: DATABASE 2002 CONFERENCE, Out., 2002, Tokyo. Proceedings

[MOROCHO 2003] Morocho, V.; Pérez-Vidal, L.; Saltor, F. Semantic Integration on Spatial Databases: SITSD prototype. In: JORNADAS DE INGENIERÍA DEL SOFTWARE Y BASES DE DATOS, 8. **Proceedings...** Alicante: 2003. p. 603–612.

[OGC 2003] OGC. **Geography Markup Language (GML) Implementation Specification 3.0.** Open GIS Consortium, 2003.

[OWL 2004] MCGUINNESS, Deborah L.; VAN HARMELEN, Frank. **OWL Web Ontology Language Overview.** 2004. Disponível em: <<http://www.w3.org/TR/owl-features/>> Acesso em: jun. 2006.

[PRINS 2003] PRINS, Mark. **Is GML only for Internet GIS?** 2003. Disponível em: <[http://www.directionsmag.com/article.php?article\\_id=280&trv=1](http://www.directionsmag.com/article.php?article_id=280&trv=1)>. Acesso em: dez. 2006.

[SILVA 2003] BUSQUIM E SILVA, Rovedy A. **Interoperabilidade na representação de dados geográficos: GEOBR e GML 3.0 no contexto da realidade dos dados geográficos no Brasil.** São José dos Campos. 146p. Dissertação (mestrado). INPE. 2003.

[WIKI 2006] **Ontology (computer science) – Wikipedia.** Disponível em:  
<[http://en.wikipedia.org/wiki/Ontology\\_%28computer\\_science%29](http://en.wikipedia.org/wiki/Ontology_%28computer_science%29)> Acesso em  
jul. 2006.

[XML SCHEMA 2007] **W3C XML Schema.** Disponível em:  
<<http://www.w3.org/XML/Schema>> Acesso em fev. 2007.

## Anexos

### Anexo 1 – Artigo

(aceito na III Escola Regional de Banco de Dados, a ser realizado em Caxias do Sul, RS em abril de 2007)

## Uma Ferramenta de Apoio à Determinação de Equivalências Semânticas entre Esquemas GML

Leonardo Roman da Rosa, Angelo Augusto Frozza, Ronaldo dos Santos Mello

Depto. de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)  
Caixa Postal 476 – 88040-900 – Florianópolis – SC – Brasil

{lrrosa, frozza, ronaldo}@inf.ufsc.br

***Abstract.** One of the main difficulties of Geographic Information Systems (GIS) users is to provide information exchange between distinct systems. This paper presents a tool that gives support to the determination of semantic equivalence between GML schemas. This tool works on a little explored domain that has a great potential of practical application: urban registry. In order to facilitate the similarity analysis, it is used an OWL ontology that maintains the representation of the domain concepts.*

***Resumo.** Uma das principais dificuldades dos usuários de Sistemas de Informações Geográficas (SIG) é a troca de informações entre sistemas distintos. Este artigo apresenta uma ferramenta que tem como objetivo o apoio à determinação de equivalências semânticas entre esquemas GML. Esta ferramenta se aplica a um domínio pouco explorado e que possui grande potencial de aplicação prática: cadastro urbano. Para facilitar a análise de similaridade é utilizada uma ontologia OWL com a representação dos conceitos do domínio.*

### 1 Introdução

Existe uma crescente necessidade de troca de informações entre instituições geradoras de dados geográficos. Muitas pesquisas têm sido desenvolvidas a respeito da integração entre objetos geográficos armazenados em fontes de dados distintas e com estrutura sintática e semântica heterogênea [Azevedo *et al.* 2006]. Órgãos públicos e privados que utilizam informações geográficas de diferentes fontes podem se beneficiar com a integração entre formatos de dados geográficos que mantêm as mesmas informações.

A *Geography Markup Language* (GML) é um padrão aberto para o transporte e armazenamento de informações geográficas em arquivos codificados na linguagem XML [OGC 2003]. Enquanto a GML facilita o

intercâmbio de informações geográficas, ela não resolve por si só a questão da integração semântica de esquemas heterogêneos.

Como a versão atual da GML ainda apresenta limitações em sua representação semântica, é possível associar definições contidas em uma ontologia de forma a facilitar a interoperabilidade semântica entre dois SIGs. A OWL (*Web Ontology Language*) é a especificação mais recente do W3C para a representação de ontologias, além de ser compatível com a arquitetura da *Web Semântica* [OWL 2006].

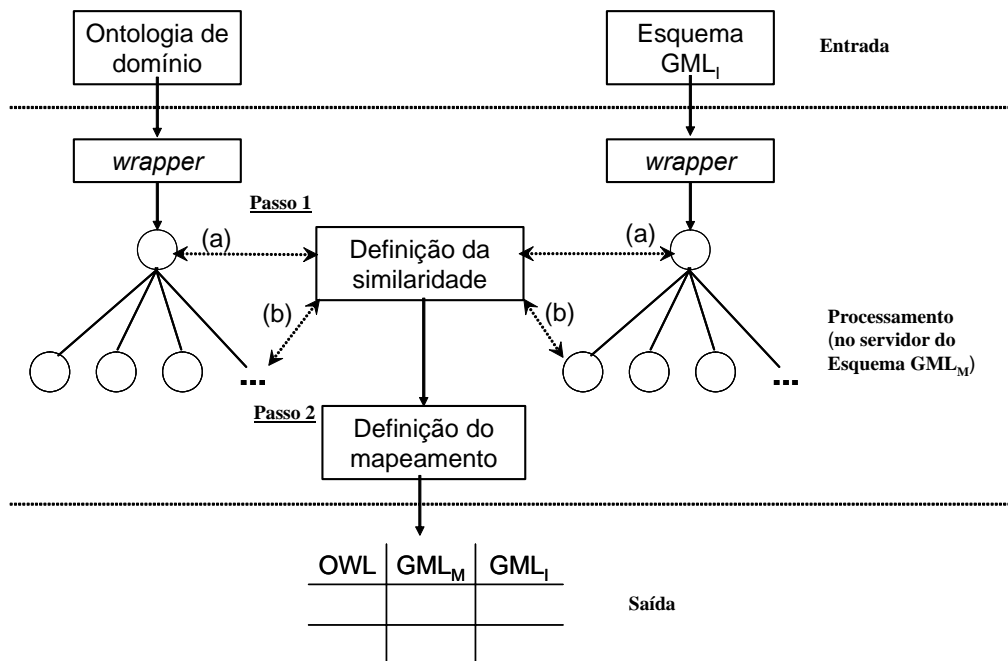
O objetivo deste artigo é apresentar uma ferramenta de *software* para apoio a um método de determinação semi-automatizada de equivalências semânticas entre esquemas GML distintos. São utilizados, especificamente, esquemas GML referentes ao domínio de cadastro urbano.

Em trabalhos relacionados, [Brauner, Casanova e Lucena 2004] introduzem um conceito de mapeamento entre duas ontologias para criar um catálogo consolidado de objetos geográficos independentes, utilizando uma ontologia de referência. Já [Morocho, Pérez-Vidal e Saltor 2003] apresentam uma ferramenta de integração semântica para dados espaciais. Em comum, abordam a interoperabilidade semântica entre SIGs em ambientes fortemente acoplados, dando ênfase na transformação de consultas. A ferramenta proposta nesse trabalho faz a integração de dados geográficos em um domínio de aplicação específico, com fontes de dados não interligadas.

O restante do artigo está organizado da seguinte maneira: a seção 2 descreve o método de determinação de equivalências proposto. A seção 3 apresenta uma visão geral da ferramenta e seu funcionamento. Por fim, a seção 4 apresenta as considerações finais deste trabalho.

## **2 Método para Determinação de Equivalências Semânticas entre Esquemas GML**

A Figura 1 mostra a arquitetura geral do método. Originalmente proposto por [Frozza e Mello 2006], ele tem como objetivo descobrir similaridades semânticas entre elementos de dois esquemas GML distintos: um esquema principal ( $GML_M$ ) que representa os dados de um SIG local ( $GML_M$ ) e outro esquema representando dados importados de um segundo SIG remoto ( $GML_I$ ). O método utiliza, ainda, uma ontologia que representa os conceitos geográficos do domínio considerado.



**Figura 1. Visão geral do método**

O método é executado no servidor principal, no qual já existe um mapeamento entre o esquema GML<sub>M</sub> e a ontologia de domínio. A entrada do método corresponde à ontologia de domínio e ao esquema GML<sub>I</sub>, os quais são transformados para um formato canônico por *wrappers* específicos.

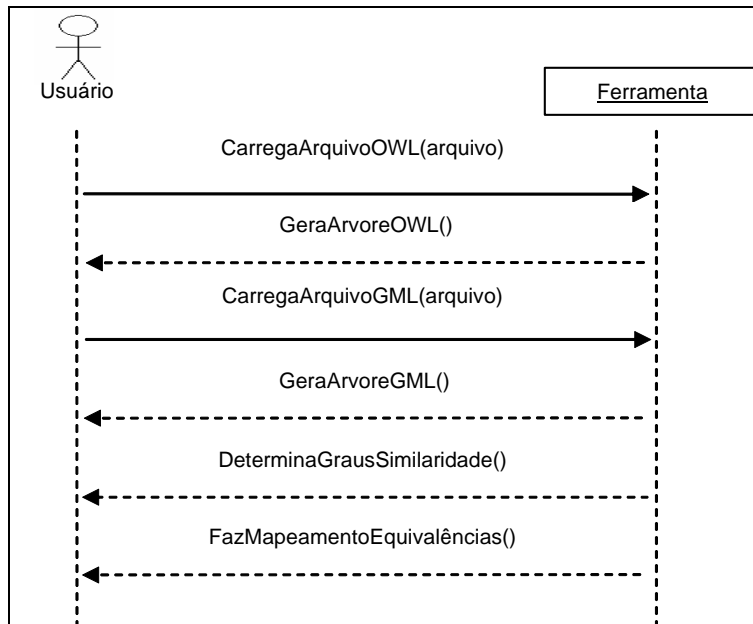
O método possui dois passos de execução. No primeiro passo é feita a determinação de equivalências semânticas entre o GML<sub>I</sub> e a ontologia de domínio. Esse passo analisa os elementos presentes em ambos os esquemas (GML<sub>I</sub> e ontologia), verificando os graus de similaridade semântica entre eles. Cada grau de similaridade é determinado por métricas que seguem a classificação proposta por [Dorneles *et al.* 2004]: métricas para valores complexos (MCV) e métricas para valores atômicos (MAV). As métricas são descritas na seção 3.4.

No segundo passo, as equivalências semânticas encontradas entre os elementos do esquema GML<sub>I</sub> e a ontologia de domínio são mapeadas para os elementos equivalentes no esquema GML<sub>M</sub>. Para isso, é usado o mapeamento já existente entre a ontologia e o esquema GML<sub>M</sub>, como descreve a seção 3.5.

### 3 Descrição da Ferramenta

#### 3.1 Implementação

A ferramenta está implementada na linguagem *Java* (utilizada devido a sua alta portabilidade), através do ambiente de desenvolvimento *Netbeans* 5.0. Ela está dividida em classes que foram criadas de forma a separar as diversas partes do método utilizado, facilitando a compreensão do código e a adição posterior de novas funcionalidades.



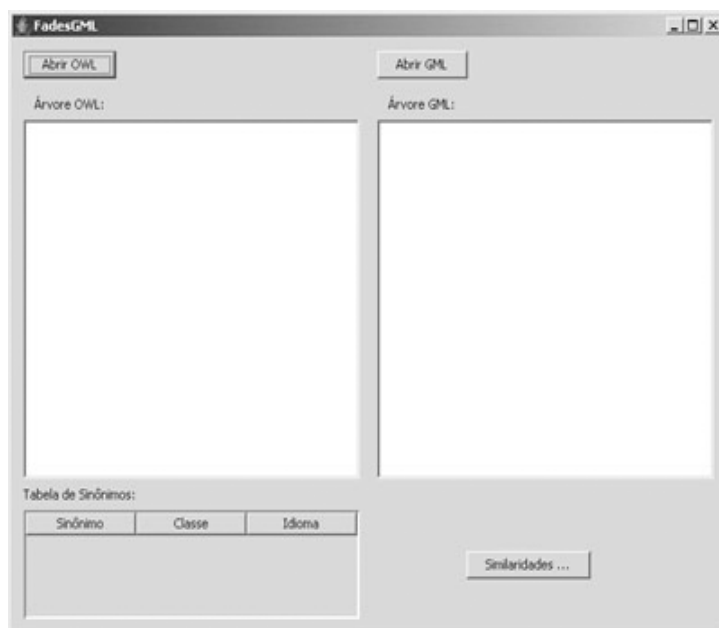
**Figura 2. Diagrama de seqüência da ferramenta**

A Figura 2 apresenta um diagrama de seqüência referente ao funcionamento da ferramenta. As funções apresentadas no diagrama são as seguintes:

- *CarregaArquivoOWL(arquivo)*: realiza a leitura e validação do arquivo correspondente à ontologia;
- *GeraArvoreOWL()*: cria e exibe a árvore com a estrutura da ontologia;
- *CarregaArquivoGML(arquivo)*: realiza a leitura e validação do arquivo correspondente ao GML<sub>I</sub>;
- *GeraArvoreGML()*: cria e exibe a árvore com a estrutura da GML<sub>I</sub>;
- *DeterminaGrausSimilaridade()*: aplica métricas de similaridade entre os elementos da ontologia e do GML<sub>I</sub>;
- *FazMapeamentoEquivalências()*: retorna o mapeamento entre os elementos do esquema GML<sub>I</sub> e os elementos do esquema GML<sub>M</sub>.

Na Figura 3 é exibida a janela principal da ferramenta. Cada botão representa uma das funções utilizadas na interação do usuário com a ferramenta, mostradas na Figura 2. Inicialmente, o usuário escolhe uma dentre duas opções: “Abrir OWL”, que corresponde à função *CarregaArquivoOWL()*, ou “Abrir GML”, correspondente à função *CarregaArquivoGML()*. Nesses dois casos, o sistema invoca a função para gerar a árvore correspondente à opção escolhida pelo usuário.



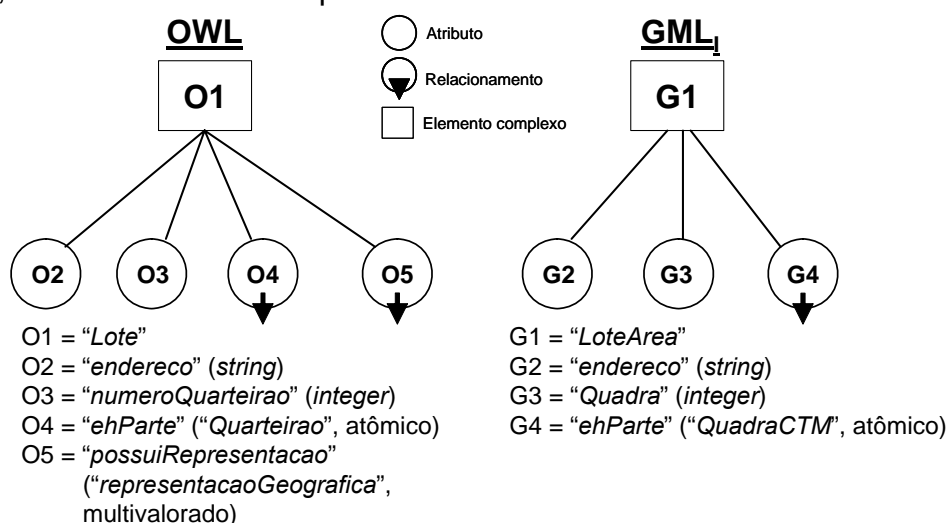


**Figura 3. Janela principal da ferramenta**

Na geração da árvore da ontologia, uma tabela de sinônimos também é preenchida (ver seção 3.2.1). Após ter as duas árvores disponíveis, o usuário seleciona o botão “*Similaridades...*” que invoca as funções *DeterminaGrausSimilaridade()* e *FazMapeamentoEquivalencias()*.

### 3.2 Árvores de Comparação

Como explicado na seção 2, o método requer como entrada uma ontologia referente ao domínio abordado e um esquema GML (GML<sub>I</sub>). Tanto a ontologia quanto o GML<sub>I</sub> devem passar por um pré-processamento em que são convertidos para um formato canônico. No caso, foi definida uma estrutura em árvore (árvore canônica), em que cada classe da ontologia e cada elemento do GML<sub>I</sub> são representados como um nodo (Figura 4). Esta forma de visualização se mostra adequada para representar dados e esquemas XML [Dorneles *et al.* 2004], tornando mais fácil o processamento.



**Figura 4. Exemplo de nodo para um elemento OWL e para um elemento GML na árvore canônica**

A estrutura das árvores canônicas possui a seguinte definição:

- *Elemento complexo*: identificador do nodo (identificador da classe ou do elemento GML);
- *Propriedades do nodo (elemento)*: atributos simples (*strings*, números etc.), atributos complexos (formados por outros atributos) e relacionamentos. Atributos e relacionamentos ainda podem receber um ou mais valores.

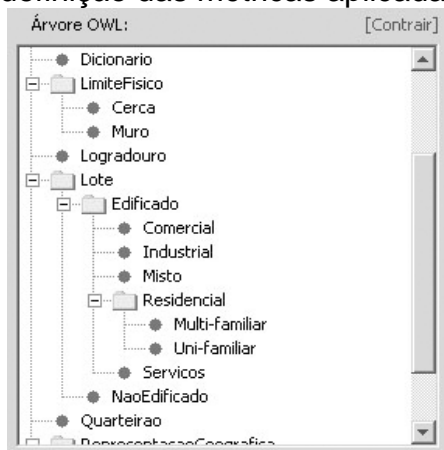
Como pode ser visto no exemplo da Figura 4, existe um elemento OWL chamado “Lote”, que tem dois atributos simples (“*endereco*” do tipo *string*; “*numeroQuarteirao*” do tipo *integer*) e dois relacionamentos (“*ehParte*”, relacionado ao conceito “*Quarteirao*” e com cardinalidade 1:1; “*possuiRepresentacao*”, relacionado com o conceito “*representacaoGeografica*” e com cardinalidade 1:n). De forma semelhante, o elemento GML “*LoteArea*” tem dois atributos simples (“*endereco*” do tipo *string*; “*Quadra*” do tipo *integer*) e um único relacionamento “*ehParte*” (que se relaciona com o elemento GML “*QuadraCTM*”, com cardinalidade 1:1).

Na ferramenta desenvolvida, os arquivos de entrada são abertos por módulos *wrappers* (classes *GeraArvoreOWL()* e *GeraArvoreGML()*), que convertem as estruturas OWL e GML para as árvores canônicas e as apresenta ao usuário. As duas árvores são apresentadas utilizando o componente *JTree* do *Java* (Figuras 5 e 6). As seções seguintes descrevem estas duas classes.

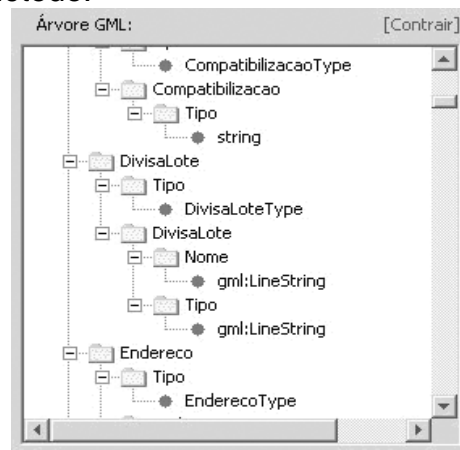
### 3.2.1 Wrappers das Árvores OWL e GML

Na montagem da árvore canônica da ontologia, cada classe é representada como um nodo  $NC_i$  da árvore e seus atributos e relacionamentos são representados como nodos filhos de  $NC_i$  [Frozza e Mello 2006]. A Figura 5 (a) mostra um fragmento da árvore canônica de uma ontologia OWL gerada pelo *wrapper* OWL.

Além da montagem da árvore canônica, a classe *wrapper* correspondente à OWL cria uma lista de sinônimos a partir das instâncias de objetos de uma classe “*Dicionario*” na ontologia. Na Tabela 1 é possível visualizar um exemplo de uma lista de sinônimos correspondente a uma ontologia hipotética, no formato *sinônimo – classe – idioma*. Cada elemento dessa lista (coluna 1) corresponde a um sinônimo de uma classe (coluna 2) na ontologia. Um sinônimo pode ser o próprio nome da classe da ontologia ou um outro sinônimo qualquer para esta classe. A indicação do idioma é utilizada na definição das métricas aplicadas pelo método.



(a)



(b)

Figura 5. Fragmentos de árvores canônicas para uma ontologia OWL (a) e para um esquema GML (b)

Tabela 1. Exemplo da lista de sinônimos

SINÔNIMO	CLASSE	IDIOMA
Lote	Lote	pt
Parcel	Lote	en
Quadra	Quarteirao	pt
Quarteirao	Quarteirao	pt

A árvore canônica para o GML<sub>1</sub> é muito semelhante à árvore gerada para a ontologia, ou seja, cada elemento do esquema é convertido em um nodo  $NE_i$  da árvore. Seus atributos e relacionamentos são convertidos em nodos folha de  $NE_i$ . Um fragmento de uma árvore GML gerada é mostrado na Figura 5 (b). Nela é possível visualizar nodos principais (como *DivisaLote*) e nodos folha (como *DivisaLoteType* e *gml:LineString*).

### 3.3. Determinação da Equivalência Semântica

Cada nodo  $NE_i$  da árvore do esquema GML<sub>1</sub> é testado contra cada nodo  $NC_j$  da ontologia para identificar se existe algum elemento  $NE_i$  idêntico ou similar representado na ontologia.

O identificador (nome) do elemento  $NE_i$  inicialmente é testado por igualdade contra a tabela de sinônimos (Tabela 1). Se um ou mais sinônimos correspondentes forem encontrados (por exemplo:  $NE_i = \text{"Lote"}$  e sinônimo = *"Lote"*), uma *métrica de similaridade de estrutura* (métrica MCV) é aplicada em cada par  $NE_i-NC_j$ . Se nenhum sinônimo idêntico ao nome de  $NE_i$  for encontrado, uma nova busca é feita na tabela de sinônimos, agora aplicando uma *métrica de similaridade de nomes* (métrica MAV). Se forem encontradas similaridades de acordo com um grau aceitável (*threshold*), então a métrica de similaridade de estruturas é aplicada a todos os pares encontrados (por exemplo:  $NE_i = \text{"Lote"}$  e sinônimo = *"LoteCTM"*)<sup>1</sup>. A definição de quê valor representa um grau de similaridade aceitável (*threshold*) para uma métrica deve ser configurada no início do processo.

Ao final do processo, eventualmente podem ser encontrados mais de um par  $NE_i-NC_j$  com graus de similaridade muito próximos. Ao mesmo tempo, alguns elementos no esquema GML<sub>1</sub> podem não ser reconhecidos pelo método e, por consequência, assumidos como não existentes na ontologia. Nestes casos, o usuário é chamado a interagir com a ferramenta, identificando qual é o par  $NE_i-NC_j$  equivalente ou se realmente o elemento  $NE_i$  não está representado na ontologia e deve ser desconsiderado. A Figura 6 mostra um exemplo de validação de equivalências exibido pela ferramenta.

<sup>1</sup> Estes novos nomes similares são cadastrados na tabela de sinônimos para consultas futuras.



Figura 6. Exemplo da tela de Validação de Equivalências

### 3.4 Métricas de Similaridade

O método adapta dois tipos de métricas propostas por [Dorneles *et al.* 2004] para a determinação dos graus de similaridade semântica entre os elementos do esquema GML importado e a ontologia de domínio.

As métricas do tipo MCV (*Métricas para Valores Complexos*) são aplicadas à estrutura dos dados. No caso deste trabalho, somente são encontradas estruturas semelhantes a *tuplas*. As métricas do tipo MAV (*Métricas para Valores Atômicos*) são aplicadas a dados simples, como *strings* e números. Elas são dependentes do domínio da aplicação, já que consideram as características dos dados da aplicação.

Este conjunto de métricas foi escolhido em [Frozza e Mello 2006] por apresentar uma taxonomia adequada ao tratamento de dados XML. No caso, um elemento XML é tratado como nodo de uma árvore, considerando que pode ser atômico ou complexo. Elementos atômicos contêm valores simples, como uma cadeia de caracteres, uma data etc. Elementos complexos correspondem a estruturas formadas por outros elementos, atômicos ou complexos.

São identificados três tipos de elementos complexos neste trabalho, sendo que para cada tipo há uma métrica MCV específica implementada pela ferramenta:

- Nodos  $NE_i$  e  $NC_j$ : têm um identificador e diversas propriedades que podem ser definições de atributos ou de relacionamentos. A similaridade dos pares  $NE_i-NC_j$  é dada pela equação 1:

$$tupleSim(\varepsilon_p, \varepsilon_d) = \frac{\sum_{\varepsilon_p^i.\eta = \varepsilon_d^j.\eta} (sim(\varepsilon_p^i, \varepsilon_d^j))}{\max(m, n)} \quad \text{Eq.1}$$

onde:

- $P$ : conjunto de nodos  $NE_i$  da árvore do esquema GML<sub>i</sub>;
  - $D$ : conjunto de nodos  $NC_j$  da árvore da ontologia;
  - $\varepsilon_p$ : um nodo do conjunto  $P$  ( $NE_i$ );
  - $\varepsilon_d$ : um nodo do conjunto  $D$  ( $NC_j$ );
  - $n$  e  $m$ : número de nodos filhos de  $\varepsilon_p$  e  $\varepsilon_d$ , respectivamente.
- Na métrica  $tupleSim()$  cada nodo filho  $\varepsilon_p^j$  de  $\varepsilon_p$  é comparado a um nodo filho  $\varepsilon_d^i$  de  $\varepsilon_d$  com mesmo nome ( $\varepsilon_p^j.\eta = \varepsilon_d^i.\eta$ ) e mesmas características. A função  $max()$  retorna o maior número de nodos filho entre  $\varepsilon_p$  e  $\varepsilon_d$ .

- Definições de atributos simples: são compostas por um identificador e um tipo de dado:

$$attrSim(\mathcal{E}_p.\eta, \mathcal{E}_d.\eta) = \frac{nameSim(\mathcal{E}_p, \mathcal{E}_d) + typeSim(\mathcal{E}_p, \mathcal{E}_d)}{2} \quad Eq.2$$

onde:

- $\mathcal{E}_p$ : nodo filho do elemento da árvore GML;
- $\mathcal{E}_d$ : nodo filho do elemento da árvore OWL;
- $nameSim$ : similaridade dos nomes dos atributos;
- $typeSim$ : similaridade dos tipos de dados.

- Definições de relacionamentos: são compostas pelo nome do relacionamento, o conceito relacionado e a cardinalidade:

$$relSim(\mathcal{E}_p.\eta, \mathcal{E}_d.\eta) = \frac{nameSim(\mathcal{E}_p, \mathcal{E}_d) + concSim(\mathcal{E}_p, \mathcal{E}_d) + cardSim(\mathcal{E}_p, \mathcal{E}_d)}{3} \quad Eq.3$$

onde:

- $nameSim$ : similaridade dos nomes dos relacionamentos;
- $concSim$ : similaridade dos conceitos referenciados pelos relacionamentos;
- $cardSim$ : similaridade das cardinalidades.

A métrica  $tupleSim()$  deve ser executada em duas etapas. Na primeira etapa consideram-se apenas os nodos filhos que correspondam a atributos de  $\mathcal{E}_p$  e  $\mathcal{E}_d$ , para os quais é aplicada a métrica  $attrSim()$ . A somatória das similaridades dos atributos gera um valor provisório de similaridade dos nodos. Na segunda etapa, são calculadas as similaridades para os nodos filhos que correspondam a relacionamentos, usando a métrica  $relSim()$ . Nesta métrica, o valor passado para a função  $concSim()$  será o valor provisório obtido na etapa anterior. Essa seqüência de execução é necessária para evitar situações de *deadlock* existentes nas dependências entre os relacionamentos.

As métricas para valores atômicos são usadas para calcular a similaridade de nomes, tipos de dados e cardinalidades. As duas últimas métricas usam uma tabela de equivalência de tipos de dados e de cardinalidades, respectivamente. Estas métricas não são detalhadas por questões de limitação de espaço.

Algumas métricas de similaridade de *strings* são encontradas na literatura [Chapman 2006], como as métricas *Jaro Winkler*, distância de *Levenshtein* e distância de *Hamming*. Na ferramenta é possível selecionar qual métrica é utilizada para definir a similaridade, usando como parâmetro o idioma na lista de sinônimos. Assim, a ferramenta fica livre para escolher a melhor métrica em cada situação. Por exemplo, para palavras na língua portuguesa, a melhor escolha pode ser a métrica *Jaro Winkler*. Para de palavras em língua inglesa, a distância de *Levenshtein* pode ser a melhor opção.

Um exemplo de determinação de similaridade gerada pela ferramenta considera os dados da Figura 4. Neste caso, aplicando as métricas  $attrSim()$  e  $relSim()$  aos elementos *Lote* e *LoteArea* para obter as similaridades entre atributos e relacionamentos, respectivamente, tem-se:

- $sim2 = attrSim(G2, O2) = 1$
- $sim3 = attrSim(G3, O3) = 0,78$
- $sim4 = relSim(G4, O4) = 0,89$

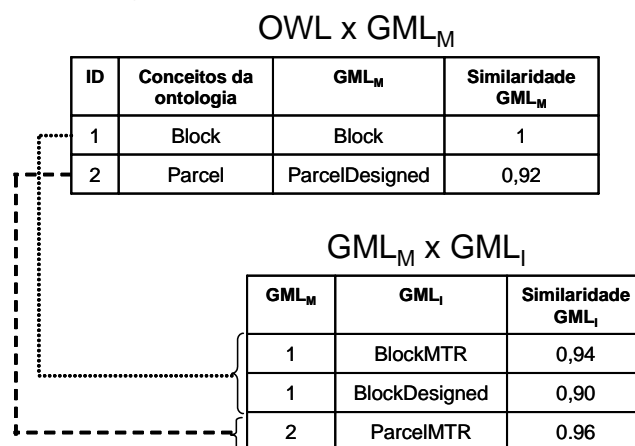
Na seqüência, determina-se o grau de similaridade final entre os dois elementos:

- $sim1 = tupleSim() = (1 + 0,78 + 0,89) / 4 = 0,67$

O valor obtido para a métrica  $tupleSim()$  aplicada ao exemplo pode ser considerado insuficiente para definir automaticamente os dois elementos como equivalentes. Neste caso, a intervenção do usuário é necessária no processo de avaliação.

### 3.5 Catalogação dos Mapeamentos

Uma vez determinadas as equivalências semânticas entre o esquema GML importado e a ontologia, a ferramenta cataloga, em tabelas relacionais de mapeamento, quais elementos do  $GML_I$  correspondem semanticamente a elementos do  $GML_M$ . A Figura 7 mostra o esquema relacional definido.



**Figura 7. Catalogação dos mapeamentos**

A tabela superior na Figura 7 armazena as equivalências entre os conceitos da ontologia e do  $GML_M$ , assim como seus graus de similaridade. A tabela inferior mantém as correspondências encontradas entre elementos do  $GML_I$  e do  $GML_M$  (referenciado da primeira tabela) e seus graus de similaridade. As equivalências encontradas devem apresentar um grau mínimo de similaridade previamente definido para que façam parte da segunda tabela de catalogação.

## 4. Considerações Finais

Este artigo apresenta uma ferramenta para apoio à determinação semi-automatizada de equivalências semânticas entre esquemas GML distintos. A ferramenta implementa o método descrito por [Frozza e Mello 2006], específico para a determinação deste tipo de equivalência. Uma vez que a ferramenta está em fase de conclusão, os resultados ainda são parciais, demonstrando que alguns pontos do método podem ainda ser refinados.

Para versões posteriores da ferramenta, existe a possibilidade de expandi-la de forma a abranger outros domínios que também utilizem esquemas GML. Conforme a necessidade e o volume dos dados utilizados, a ferramenta também poderá fazer uso de bancos de dados para a criação das tabelas de sinônimos. Ainda é possível propor a implementação do método como um *web service*. Dessa forma, a ferramenta funcionaria como um aplicativo cliente-servidor, utilizando a *Web* como meio de acesso e possibilitando diferentes interfaces de consulta.

## Referências

- Azevedo, V. H. M. *et al.* Interoperabilidade entre Objetos Geográficos Heterogêneos. In: GEOINFO, 8., 2006, Campos do Jordão. **Anais...** São José dos Campos: INPE, 2006. p. 235-245.
- Brauner, D. F.; Casanova, M. A.; Lucena, C. J. P. Geo-Object Catalogs to enable Geographic Databases Interoperability. In: GEOINFO, 6., 2004, Campos do Jordão. **Anais...** São José dos Campos: INPE, 2004. p. 235-246.
- Chapman, J. **Sam's Strings Metrics**. Disponível em: <<http://www.dcs.shef.ac.uk/~sam/stringmetrics.html>>. Acessado em: 25 mai. 2006.
- Dorneles, C. F. *et al.* Measuring Similarity between Collection of Values. In: ACM INTERNATIONAL WORKSHOP ON WEB INFORMATION AND DATA MANAGEMENT, 6., 2004. **Proceedings...** Washington: ACM, 2004. p. 56-63.
- Frozza, A. A.; Mello, R. S. Um Método para Determinar a Equivalência Semântica entre Esquemas GML. In: GEOINFO, 8., 2006, Campos do Jordão. **Anais...** São José dos Campos: INPE, 2006. p. 283-293.
- Morocho, V.; Pérez-Vidal, L.; Saltor, F. Semantic Integration on Spatial Databases: SITSD prototype. In: JORNADAS DE INGENIERÍA DEL SOFTWARE Y BASES DE DATOS, 8. **Proceedings...** Alicante: 2003. p. 603-612.
- OGC. **Geography Markup Language (GML) Implementation Specification 3.0**. Open GIS Consortium, 2003.
- OWL. **Web Ontology Language**. Disponível em: <<http://www.w3.org/2004/OWL/>>. Acessado em: 16 jul. 2006.

## Anexo 2 – Esquemas Utilizados no Estudo de Caso

### A – Ontologia

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Servicos">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Edificado"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Edificado">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Lote"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Multi-familiar">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Residencial"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="NaoEdificado">
    <rdfs:subClassOf rdf:resource="#Lote"/>
  </owl:Class>
  <owl:Class rdf:ID="Tributavel">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Conceitualizacao"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Comercial">
    <rdfs:subClassOf rdf:resource="#Edificado"/>
  </owl:Class>
  <owl:Class rdf:ID="Ponto">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="RepresentacaoGeografica"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Quarteirao"/>
  <owl:Class rdf:ID="Uni-familiar">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Residencial"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Cerca">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="LimiteFisico"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Logradouro"/>
  <owl:Class rdf:about="#Residencial">
    <rdfs:subClassOf rdf:resource="#Edificado"/>
  </owl:Class>
</rdf:RDF>
```



```

</owl:Class>
<owl:Class rdf:ID="Muro">
  <rdfs:subClassOf rdf:resource="#LimiteFisico"/>
</owl:Class>
<owl:Class rdf:ID="Cadastral">
  <rdfs:subClassOf rdf:resource="#Conceitualizacao"/>
</owl:Class>
<owl:Class rdf:ID="Dicionario"/>
<owl:Class rdf:ID="Linha">
  <rdfs:subClassOf rdf:resource="#RepresentacaoGeografica"/>
</owl:Class>
<owl:Class rdf:ID="Juridico">
  <rdfs:subClassOf rdf:resource="#Conceitualizacao"/>
</owl:Class>
<owl:Class rdf:ID="Poligono">
  <rdfs:subClassOf rdf:resource="#RepresentacaoGeografica"/>
</owl:Class>
<owl:Class rdf:ID="CoordXY"/>
<owl:Class rdf:ID="Industrial">
  <rdfs:subClassOf rdf:resource="#Edificado"/>
</owl:Class>
<owl:Class rdf:ID="Misto">
  <rdfs:subClassOf rdf:resource="#Edificado"/>
</owl:Class>
<owl:Class rdf:ID="Legal">
  <rdfs:subClassOf rdf:resource="#Conceitualizacao"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="_rgLogradouro">
  <rdfs:range rdf:resource="#Linha"/>
  <rdfs:domain rdf:resource="#Logradouro"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="_ehDelimitado">
  <rdfs:domain rdf:resource="#Quarteirao"/>
  <rdfs:range rdf:resource="#Logradouro"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="_rgLimiteFisico">
  <rdfs:domain rdf:resource="#LimiteFisico"/>
  <rdfs:range rdf:resource="#Linha"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="_coordenadasPonto">
  <rdfs:domain rdf:resource="#Ponto"/>
  <rdfs:range rdf:resource="#CoordXY"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="_possuiConceitualizacao">
  <rdfs:range rdf:resource="#Conceitualizacao"/>
  <rdfs:domain rdf:resource="#Lote"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="_coordenadasPoligono">
  <rdfs:domain rdf:resource="#Poligono"/>
  <rdfs:range rdf:resource="#CoordXY"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="_daAcesso">
  <owl:inverseOf>
    <owl:FunctionalProperty rdf:ID="_estaLocalizado"/>
  </owl:inverseOf>
  <rdfs:domain rdf:resource="#Logradouro"/>
  <rdfs:range rdf:resource="#Lote"/>

```

```

</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="_rgLote">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Lote"/>
  <rdfs:range rdf:resource="#Poligono"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="_ehDivididoEm">
  <rdfs:domain rdf:resource="#Quarteirao"/>
  <owl:inverseOf>
    <owl:FunctionalProperty rdf:ID="_ehParte"/>
  </owl:inverseOf>
  <rdfs:range rdf:resource="#Lote"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="_coordenadasLinha">
  <rdfs:domain rdf:resource="#Linha"/>
  <rdfs:range rdf:resource="#CoordXY"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="tipoUsoMisto">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Misto"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="testada">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Lote"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="sinonimo">
  <rdfs:domain rdf:resource="#Dicionario"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="numeroDocumentoLegal">
  <rdfs:domain rdf:resource="#Juridico"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="tipoDocumentoLegal">
  <rdfs:domain rdf:resource="#Juridico"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="numeroSetor">
  <rdfs:domain rdf:resource="#Quarteirao"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="tipoMaterial">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Cerca"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="valorVenal">
  <rdfs:domain rdf:resource="#Lote"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="numeroQuarteirao">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Quarteirao"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="ccordY">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>

```

```

<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
<rdfs:domain rdf:resource="#CoordXY"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="coordX">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
<rdfs:domain rdf:resource="#CoordXY"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="endereco">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:domain rdf:resource="#Lote"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="codigoQuarteirao">
<rdfs:domain rdf:resource="#Lote"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="numeroPlantaPrefeitura">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:domain rdf:resource="#Legal"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="tipoAlvenaria">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
<rdfs:domain rdf:resource="#Muro"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="numeroIdentificacao">
<rdfs:domain rdf:resource="#Lote"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:FunctionalProperty rdf:ID="fundos">
<rdfs:domain rdf:resource="#Lote"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="numeroUnidades">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:domain rdf:resource="#Multi-familiar"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="bairro">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:domain rdf:resource="#Quarteirao"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="proprietario">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:domain rdf:resource="#Tributavel"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="perimetro">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
<rdfs:domain rdf:resource="#Lote"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="tipoIndustrial">
<rdfs:domain rdf:resource="#Industrial"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>

```

```

<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="tipoComercial">
  <rdfs:domain rdf:resource="#Comercial"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#_estaLocalizado">
  <owl:inverseOf rdf:resource="#_daAcesso"/>
  <rdfs:domain rdf:resource="#Lote"/>
  <rdfs:range rdf:resource="#Logradouro"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="area">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#Lote"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="codigoEdificacao">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Edificado"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="numeroCertidao">
  <rdfs:domain rdf:resource="#Legal"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="tipoServico">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Servicos"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="_rgQuarteirao">
  <rdfs:domain rdf:resource="#Quarteirao"/>
  <rdfs:range rdf:resource="#Poligono"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="_ehLimitadoPor">
  <rdfs:range rdf:resource="#LimiteFisico"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <owl:inverseOf>
    <owl:InverseFunctionalProperty rdf:ID="_pertence"/>
  </owl:inverseOf>
  <rdfs:domain rdf:resource="#Lote"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="numeroCroqui">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Cadastral"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Numero do croqui que aparece o lote.</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#_ehParte">
  <rdfs:range rdf:resource="#Quarteirao"/>
  <rdfs:domain rdf:resource="#Lote"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <owl:inverseOf rdf:resource="#_ehDivididoEm"/>
</owl:FunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="#_pertence">

```

```

<rdfs:range rdf:resource="#Lote"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
<rdfs:domain rdf:resource="#LimiteFisico"/>
<owl:inverseOf rdf:resource="#_ehLimitadoPor"/>
</owl:InverseFunctionalProperty>
<Dicionario rdf:ID="_Quarteirao">
  <sinonimo xml:lang="pt">Quadra</sinonimo>
  <sinonimo xml:lang="en">Block</sinonimo>
  <sinonimo xml:lang="pt">Quarteirao</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Logradouro">
  <sinonimo xml:lang="pt">Logradouro</sinonimo>
  <sinonimo xml:lang="pt">Endereco</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Residencial">
  <sinonimo xml:lang="pt">Residencial</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Comercial">
  <sinonimo xml:lang="pt">Comercial</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Servicos">
  <sinonimo xml:lang="pt">Servicos</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_LimiteFisico">
  <sinonimo xml:lang="nl">LimiteFisico</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Lote">
  <sinonimo xml:lang="pt">Lote</sinonimo>
  <sinonimo xml:lang="en">Parcel</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Misto">
  <sinonimo xml:lang="pt">Misto</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Cadastral">
  <sinonimo xml:lang="pt">Cadastral</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Muro">
  <sinonimo xml:lang="pt">Muro</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Multi-Familiar">
  <sinonimo xml:lang="pt">Multi-Familiar</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Industrial">
  <sinonimo xml:lang="pt">Industrial</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Conceitualizacao">
  <sinonimo xml:lang="pt">Conceitualizacao</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Legal">
  <sinonimo xml:lang="pt">Legal</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Juridico">
  <sinonimo xml:lang="pt">Juridico</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Edificado">
  <sinonimo xml:lang="pt">Edificado</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_NaoEdificado">
  <sinonimo xml:lang="pt">NaoEdificado</sinonimo>
</Dicionario>

```

```

<Dicionario rdf:ID="_Uni-Familiar">
  <sinonimo xml:lang="pt">Uni-Familiar</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Cerca">
  <sinonimo xml:lang="pt">Cerca</sinonimo>
</Dicionario>
<Dicionario rdf:ID="_Tributavel">
  <sinonimo xml:lang="pt">Tributavel</sinonimo>
</Dicionario>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 2.2, Build 311) http://protege.stanford.edu -->

```

## B – Esquema GML de Lages

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:urb="lages" xmlns:ns1="http://www.opengis.net/gml" targetNamespace="lages"
elementFormDefault="qualified" version="2007-02-15">
  <!--=====-->
  <import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd"/>
  <import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gmlBase.xsd"/>
  <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/xlink/xlinks.xsd"/>
  <import/>
  <!--=====-->
  <!-- =====
PACK ORGANIZAÇÃO TERRITORIAL
===== -->
  <element name="OrgTerritorialPack" type="urb:OrgTerritorialType"
substitutionGroup="gml:_FeatureCollection"/>
  <complexType name="OrgTerritorialType">
    <complexContent>
      <extension base="gml:AbstractFeatureCollectionType"/>
    </complexContent>
  </complexType>

  <element name="_OrgTerritorialFeature" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="gml:_Feature"/>
  <element name="OrgTerritorialMember" type="urb:OrgTerritorialPropertyType"
substitutionGroup="gml:featureMember"/>
  <complexType name="OrgTerritorialPropertyType">
    <complexContent>
      <restriction base="gml:FeaturePropertyType">
        <sequence minOccurs="0">
          <element ref="urb:_OrgTerritorialFeature"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- =====
CLASSES ORGANIZAÇÃO TERRITORIAL
===== -->
  <element name="_SetorCensitario" type="gml:AbstractFeatureType" abstract="true"

```

```

substitutionGroup="urb:_OrgTerritorialFeature"/>
  <element name="SetorCensitario" type="urb:SetorCensitarioType"
substitutionGroup="urb:_SetorCensitario"/>
  <complexType name="SetorCensitarioType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:Polygon"/>
          <!--<element name="YY" type="integer"/>-->
          <element name="contidoEmMunicipio"
type="urb:MunicipioPropertyType">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">contemSetorCensitario</appinfo>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="SetorCensitarioPropertyType">
    <sequence>
      <element ref="urb:SetorCensitario" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!--=====>
  <element name="_Municipio" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
  <element name="Municipio" type="urb:MunicipioType"
substitutionGroup="urb:_Municipio"/>
  <complexType name="MunicipioType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:Polygon"/>
          <element name="Limites" type="string"/>
          <element name="contemSetorCensitario"
type="urb:SetorCensitarioPropertyType" minOccurs="0" maxOccurs="unbounded">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">contidoEmMunicipio</appinfo>
            </annotation>
          </element>
          <element name="contemDistrito"
type="urb:DistritoPropertyType" minOccurs="0" maxOccurs="unbounded">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">contidoEmMunicipio</appinfo>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="MunicipioPropertyType">
    <sequence>
      <element ref="urb:Municipio" minOccurs="0"/>
    </sequence>

```

```

        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->
    <element name="_Distrito" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
    <element name="Distrito" type="urb:DistritoType" substitutionGroup="urb:_Distrito"/>
    <complexType name="DistritoType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element ref="gml:Polygon"/>
                    <element name="codigoDistrito" type="integer"/>
                    <element name="municipio" type="string"/>
                    <element name="nomeDistrito" type="string"/>
                    <element name="contidoEmMunicipio"
type="urb:MunicipioPropertyType">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">contemDistrito</appinfo>
                        </annotation>
                    </element>
                    <element name="contemUnidadeTerritorial"
type="urb:UnidadeTerritorialPropertyType" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">contidoEmDistrito</appinfo>
                        </annotation>
                    </element>
                    <element name="contemSetor"
type="urb:SetorPropertyType" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">contidoEmDistrito</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="DistritoPropertyType">
        <sequence>
            <element ref="urb:Distrito" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->
    <element name="_UnidadeTerritorial" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
    <element name="UnidadeTerritorial" type="urb:UnidadeTerritorialType"
substitutionGroup="urb:_UnidadeTerritorial"/>
    <complexType name="UnidadeTerritorialType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element ref="gml:Polygon"/>
                    <element name="codigoUT" type="integer"/>
                    <element name="codigoDistrito" type="integer"/>
                    <element name="categoria" type="string"/> <!-- UTRE-
1/UTRE-2/UTRP-1/UTRP-2/UTIE-1/UTIP-1/UTOE-1/UTOE-2/UTOE-3/UTUP-1/UCCS-1 -->
                    <element name="areaUT" type="float"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

```



```

type="urb:DistritoPropertyType">
    <element name="contidoEmDistrito"
    <annotation>
        <appinfo source="urn:x-
gml:reverseProperty">contemUnidadeTerritorial</appinfo>
    </annotation>
    </element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="UnidadeTerritorialPropertyType">
    <sequence>
        <element ref="urb:UnidadeTerritorial" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_Setor" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="Setor" type="urb:SetorType" substitutionGroup="urb:_Setor"/>
<complexType name="SetorType">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element ref="gml:Polygon"/>
                <element name="codigoSetor" type="integer"/>
                <element name="codigoDistrito" type="integer"/>
                <element name="nomeSetor" type="string"/>
                <element name="contidoEmDistrito"
type="urb:DistritoPropertyType">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">contemSetor</appinfo>
                    </annotation>
                </element>
            </sequence>
            <element name="contemQuadra"
type="urb:QuadraPropertyType" minOccurs="0" maxOccurs="unbounded">
                <annotation>
                    <appinfo source="urn:x-
gml:reverseProperty">contidaEmSetor</appinfo>
                </annotation>
            </element>
            <element name="contemLoteamento"
type="urb:LoteamentoPropertyType" minOccurs="0" maxOccurs="unbounded">
                <annotation>
                    <appinfo source="urn:x-
gml:reverseProperty">contidoEmSetor</appinfo>
                </annotation>
            </element>
        </sequence>
    </extension>
</complexContent>
</complexType>
<complexType name="SetorPropertyType">
    <sequence>
        <element ref="urb:Setor" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

```

<!------->
<element name="_Quadra" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="Quadra" type="urb:QuadraType" substitutionGroup="urb:_Quadra"/>
<complexType name="QuadraType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:Polygon"/>
        <element name="codigoQuadra" type="integer"/>
        <element name="codigoSetor" type="integer"/>
        <element name="codigoBairro" type="integer"/>
        <element name="situacaoAtual" type="string"/><!--
real/projetada-->
        <element name="areaTerritorial" type="float"/>
        <element
type="urb:SetorPropertyType">
          name="contidaEmSetor"
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">contemQuadra</appinfo>
          </annotation>
        </element>
        <element
type="urb:BairroPropertyType">
          name="contidaEmBairro"
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">contemQuadra</appinfo>
          </annotation>
        </element>
        <element
type="urb:LotePropertyType" maxOccurs="unbounded">
          name="contemLote"
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">contidoEmQuadra</appinfo>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="QuadraPropertyType">
  <sequence>
    <element ref="urb:Quadra" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!------->
<element name="_QuadraProjetada" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element
name="QuadraProjetada"
type="urb:QuadraProjetadaType"
substitutionGroup="urb:_QuadraProjetada"/>
<complexType name="QuadraProjetadaType">
  <complexContent>
    <extension base="urb:QuadraType">
      <sequence>
        <element ref="gml:Polygon"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

<!--=====>
<element name="_QuadraReal" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="QuadraReal" type="urb:QuadraRealType"
substitutionGroup="urb:_QuadraReal"/>
<complexType name="QuadraRealType">
<complexContent>
<extension base="urb:QuadraType">
<sequence>
<element ref="gml:Polygon"/>
</sequence>
</extension>
</complexContent>
</complexType>
<!--=====>
<element name="_QuadraDeLoteamento" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="QuadraDeLoteamento" type="urb:QuadraDeLoteamentoType"
substitutionGroup="urb:_QuadraDeLoteamento"/>
<complexType name="QuadraDeLoteamentoType">
<complexContent>
<extension base="urb:QuadraType">
<sequence>
<element ref="gml:Polygon"/>
<element name="sobrepoeLoteDeLoteamento"
type="urb:LoteDeLoteamentoPropertyType" maxOccurs="unbounded">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">sobrepostoPorQuadraDeLoteamento</appinfo>
</annotation>
</element>
<element name="sobrepostoPorLoteamento"
type="urb:LoteamentoPropertyType" maxOccurs="unbounded">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">sobrepoeloteamento</appinfo>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="QuadraDeLoteamentoPropertyType">
<sequence>
<element ref="urb:QuadraDeLoteamento" minOccurs="0"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====>
<element name="_Bairro" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="Bairro" type="urb:BairroType" substitutionGroup="urb:_Bairro"/>
<complexType name="BairroType">
<complexContent>
<extension base="gml:AbstractFeatureType">
<sequence>
<element ref="gml:Polygon"/>
<element name="codigoBairro" type="integer"/>
<element name="nomeBairro" type="string"/>
<element name="numeroHabitantes" type="integer"/>

```

```

                <element name="leiAprovacao" type="string"/>
                <element name="dataAprovacao" type="date"/>
                <element name="contemQuadra"
type="urb:QuadraPropertyType" maxOccurs="unbounded">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">contidaEmBairro</appinfo>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="BairroPropertyType">
    <sequence>
        <element ref="urb:Bairro" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_Loteamento" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="Loteamento" type="urb:LoteamentoType"
substitutionGroup="urb:_Loteamento"/>
<complexType name="LoteamentoType">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element ref="gml:Polygon"/>
                <element name="codigoLoteamento" type="integer"/>
                <element name="codigoBairro" type="integer"/>
                <element name="codigoSetor" type="integer"/>
                <element name="situacaoAtual" type="string"/> <!--
Real/Projetada -->
                <element name="nomeLoteamento" type="string"/>
                <element name="nomeProprietario" type="string"/>
                <element name="numeroDecretoAprovacao"
type="string"/>
                <element name="dataDecretoAprovacao"
type="date"/>
                <element name="areaTotal" type="float"/>
                <element name="areaDeLotes" type="float"/>
                <element name="areaDeArruamento" type="float"/>
                <element name="areaEspacoPublicoInstitucional"
type="float"/>
                <element name="areaEspacoPublicoEspecial"
type="float"/>
                <element name="areaEspacoPublicoPreservacaoPermanente" type="float"/>
                <element name="areaNonEdificandi" type="float"/>
                <element name="contidoEmSetor"
type="urb:SetorPropertyType">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">contemLoteamento</appinfo>
                    </annotation>
                </element>
                <element name="sobrepoeLoteDeLoteamento"
type="urb:LoteDeLoteamentoPropertyType" maxOccurs="unbounded">
                    <annotation>

```

```

                                <appinfo                source="urn:x-
gml:reverseProperty">sobrepostoPorLoteamento</appinfo>
                                </annotation>
                                </element>
                                <element                name="sobrepoeQuadraDeLoteamento"
type="urb:QuadraDeLoteamentoPropertyType" maxOccurs="unbounded">
                                <annotation>
                                <appinfo                source="urn:x-
gml:reverseProperty">sobrepostoPorLoteamento</appinfo>
                                </annotation>
                                </element>
                                </sequence>
                                </extension>
                                </complexContent>
                                </complexType>
                                <complexType name="LoteamentoPropertyType">
                                <sequence>
                                <element ref="urb:Loteamento" minOccurs="0"/>
                                </sequence>
                                <attributeGroup ref="gml:AssociationAttributeGroup"/>
                                </complexType>
                                <!--=====>
                                <element name="_LoteamentoReal" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
                                <element                name="LoteamentoReal"                type="urb:LoteamentoRealType"
substitutionGroup="urb:_LoteamentoReal"/>
                                <complexType name="LoteamentoRealType">
                                <complexContent>
                                <extension base="urb:LoteamentoType">
                                <sequence>
                                <element ref="gml:Polygon"/>
                                </sequence>
                                </extension>
                                </complexContent>
                                </complexType>
                                <!--=====>
                                <element                name="_LoteamentoProjetado"                type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_OrgTerritorialFeature"/>
                                <element                name="LoteamentoProjetado"                type="urb:LoteamentoProjetadoType"
substitutionGroup="urb:_LoteamentoProjetado"/>
                                <complexType name="LoteamentoProjetadoType">
                                <complexContent>
                                <extension base="urb:LoteamentoType">
                                <sequence>
                                <element ref="gml:Polygon"/>
                                </sequence>
                                </extension>
                                </complexContent>
                                </complexType>
                                <!--=====>
                                <element name="_Unidadelmobiliaria" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
                                <element                name="Unidadelmobiliaria"                type="urb:UnidadelmobiliariaType"
substitutionGroup="urb:_Unidadelmobiliaria"/>
                                <complexType name="UnidadelmobiliariaType">
                                <complexContent>
                                <extension base="gml:AbstractFeatureType">
                                <sequence>
                                <element ref="gml:Polygon"/>
                                <element name="codigoUI" type="integer"/>

```

```

        <element name="codigoContribuinte" type="integer"/>
        <element name="numeroCadastr" type="string"/>
        <element name="inscricaoCadastral" type="string"/>
        <element name="categoriaUI" type="string"/><!--
Predial/Territorial -->
        <element name="propriedade" type="string"/><!--
Particular/Público Federal/Público Estadual/Público Municipal/Religioso/Entidade Sem Fins
Lucrativos -->
        <element name="posse" type="string"/><!--
Próprio/Alugado/Cedido/Invadido/Direito de Superfície -->
        <element name="incidenciaTributos" type="string"/><!--
- Normal/Isento de IPTU/Imune de IPTU/Isento de TSU/Isento de IPTU e de TSU/Exclusão
Lógica -->
        <element name="servidaoVila" type="boolean"/>
        <element name="propriedadeUnica" type="string"/><!--
Uma/Mais de Uma -->
        <element name="documentosPosse" type="string"/><!--
- Número da Matrícula/Ofício do RI/Data do Registro/Área do Terreno/Dim. Frente/Dim.
Fundos/Dim. Lado Direito/Dim. Lado Esquerdo -->
        <element name="aliquota" type="float"/>
        <element name="pertenceContribuinteP"
type="urb:ContribuintePPropertyType">
                <annotation>
                <appinfo source="urn:x-
gml:reverseProperty">possuiUnidadeImobiliaria</appinfo>
                </annotation>
        </element>
        <element name="pertenceTombamento"
type="urb:TombamentoPropertyType">
                <annotation>
                <appinfo source="urn:x-
gml:reverseProperty">possuiUnidadeImobiliaria</appinfo>
                </annotation>
        </element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="UnidadeImobiliariaPropertyType">
        <sequence>
                <element ref="urb:UnidadeImobiliaria" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_ContribuinteP" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
        <element name="ContribuinteP" type="urb:ContribuintePType"
substitutionGroup="urb:_ContribuinteP"/>
        <complexType name="ContribuintePType">
                <complexContent>
                        <extension base="gml:AbstractFeatureType">
                                <sequence>
                                        <element name="codigoContribuinte" type="integer"/>
                                        <element name="nome" type="string"/>
                                        <element name="cpfCGC" type="string"/>
                                        <element name="ruaCorrespondencia" type="string"/>
                                        <element name="numeroCorrespondencia"
type="integer"/>
                                        <element name="cep" type="string"/>

```

```

        <element name="bairro" type="string"/>
        <element name="complemento" type="string"/>
        <element name="cidade" type="string"/>
        <element name="uf" type="string"/>
        <element name="telefone" type="string"/>
        <element name="celular" type="string"/>
        <element name="email" type="string"/>
        <element name="debitoAutomatico" type="boolean"/>
        <element name="codigoBanco" type="string"/>
        <element name="codigoBanco" type="string"/>
        <element name="contaCorrente" type="string"/>

        <element
type="urb:UnidadelmobiliariaPropertyType" minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <appinfo
                    source="urn:x-
gml:reverseProperty">pertenceContribuinteP</appinfo>
            </annotation>
        </element>
    </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="ContribuintePPropertyType">
    <sequence>
        <element ref="urb:ContribuinteP" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_Tombamento" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element
name="Tombamento" type="urb:TombamentoType"
substitutionGroup="urb:_Tombamento"/>
<complexType name="TombamentoType">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element name="codigoTombamento" type="integer"/>
                <element name="codigoUI" type="integer"/>
                <element name="numeroTombamento" type="string"/>
                <element name="dataLeiTombamento" type="date"/>
                <element
                    name="categoriaTombamento"
type="string"/><!-- Federal/Estadual/Municipal -->
                <element
                    name="possuiUnidadelmobiliaria"
type="urb:UnidadelmobiliariaPropertyType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <appinfo
                            source="urn:x-
gml:reverseProperty">pertenceTombamento</appinfo>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="TombamentoPropertyType">
    <sequence>
        <element ref="urb:Tombamento" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>

```

```

</complexType>
<!--=====-->
<element name="_UIPredial" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="UIPredial" type="urb:UIPredialType"
substitutionGroup="urb:_UIPredial"/>
<complexType name="UIPredialType">
<complexContent>
<extension base="urb:UnidadeImobiliariaType">
<sequence>
<element ref="gml:Polygon"/>
<element name="possuiHabiteSe"
type="urb:HabiteSePropertyType" minOccurs="0" maxOccurs="unbounded">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">pertenceUIPredial</appinfo>
</annotation>
</element>
<element name="possuiConstrucao"
type="urb:ConstrucaoPropertyType" minOccurs="0" maxOccurs="unbounded">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">pertenceUIPredial</appinfo>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="UIPredialPropertyType">
<sequence>
<element ref="urb:UIPredial" minOccurs="0"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_HabiteSe" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="HabiteSe" type="urb:HabiteSeType"
substitutionGroup="urb:_HabiteSe"/>
<complexType name="HabiteSeType">
<complexContent>
<extension base="gml:AbstractFeatureType">
<sequence>
<element name="codigoHabiteSe" type="integer"/>
<element name="codigoUIPredial" type="integer"/>
<element name="numeroHabiteSe" type="string"/>
<element name="numeroSegundaViaHabiteSe"
type="string"/>
<element name="areaLiberada" type="float"/>
<element name="parteLiberada" type="float"/>
<element name="pertenceUIPredial"
type="urb:UIPredialPropertyType">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">possuiHabiteSe</appinfo>
</annotation>
</element>
</sequence>
</extension>

```



```

        </complexContent>
    </complexType>
    <complexType name="HabitesPropertyType">
        <sequence>
            <element ref="urb:Habites" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->
    <element name="_Construcao" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
    <element name="Construcao" type="urb:ConstrucaoType"
substitutionGroup="urb:_Construcao"/>
    <complexType name="ConstrucaoType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element name="codigoConstrucao" type="integer"/>
                    <element name="codigoUIPredial" type="integer"/>
                    <element name="categoriaConstrucao"
type="integer"/><!-- Inicial/Reforma/Ampliação/Demolição/Concluída -->
                    <element name="prevencaoCombateIncendio"
type="integer"/><!-- Instalado/Número Habite-se Bombeiros/Data Habite-se
Bombeiros/Inexistente -->
                    <element name="areaProjeto" type="float"/>
                    <element name="pertenceUIPredial"
type="urb:UIPredialPropertyType">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">possuiConstrucao</appinfo>
                        </annotation>
                    </element>
                    <element name="possuiAlvara"
type="urb:AlvaraPropertyType" minOccurs="0">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">pertenceConstrucao</appinfo>
                        </annotation>
                    </element>
                    <element name="responsabilidadeDeResponsavelTecnico"
type="urb:ResponsavelTecnicoPropertyType" minOccurs="0" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">responsavelPorConstrucao</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="ConstrucaoPropertyType">
        <sequence>
            <element ref="urb:Construcao" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->
    <element name="_Alvara" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>

```

```

<element name="Alvara" type="urb:AlvaraType" substitutionGroup="urb:_Alvara"/>
<complexType name="AlvaraType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="codigoAlvara" type="integer"/>
        <element name="codigoConstrucao" type="integer"/>
        <element name="numeroAlvara" type="string"/>
        <element name="dataAlvara" type="date"/>
        <element name="pertenceConstrucao"
type="urb:ConstrucaoPropertyType">
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">possuiAlvara</appinfo>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="AlvaraPropertyType">
  <sequence>
    <element ref="urb:Alvara" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_UITerritorial" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="UITerritorial" type="urb:UITerritorialType"
substitutionGroup="urb:_UITerritorial"/>
<complexType name="UITerritorialType">
  <complexContent>
    <extension base="urb:UnidadeImobiliariaType">
      <sequence>
        <element ref="gml:Polygon"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!--=====-->
<element name="_Lote" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="Lote" type="urb:LoteType" substitutionGroup="urb:_Lote"/>
<complexType name="LoteType">
  <complexContent>
    <extension base="urb:UITerritorialType">
      <sequence>
        <element ref="gml:Polygon"/>
        <element name="codigoLote" type="integer"/>
        <element name="codigoUI" type="integer"/>
        <element name="areaTerritorial" type="float"/>
        <element name="coeficienteTerritorial" type="float"/>
        <element name="testadaReal" type="float"/>
        <element name="valorMetroLinearTestada"
type="float"/>
        <element name="profundidadeMedia" type="float"/>
        <element name="testadaCorrigidaTerritorial"
type="float"/>
        <element name="valorVenal" type="float"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

type="boolean"/>
<element name="inadimplenciaTerritorial"
Plano/Aclive/Declive/Irregular -->
<element name="relevo" type="string"/> <!--
Regular/Irregular -->
<element name="dimensoes" type="string"/> <!--
Normal/Rochoso/Arenoso/Brejososo/Alagado/Inundável/Combinação -->
<element name="pedologia" type="string"/> <!--
Ao Nível/Mais Alto/Mais Baixo -->
<element name="nivelRelacaoRua" type="string"/> <!--
<!-- Distante até 100m/Até 500m/Mais de 500m -->
<element name="transporteColetivo" type="string"/>
<!-- Fossa Séptica/Sumidouro/Fossa Séptica+Sumidouro/Filtro Anaeróbico/Filtro
Aeróbico/Industrial/Céu Aberto/Inexistente -->
<element name="tratamentoEfluentes" type="string"/>
<!-- Concessionária Local/Poço/Rede+Poço/Inexistente -->
<element name="fornecimentoAgua" type="string"/>
<!-- Concessionária Local/Poço/Rede+Poço/Inexistente -->
type="urb:QuadraPropertyType">
<element name="contidoEmQuadra"
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">contemLote</appinfo>
</annotation>
</element>
<element name="possuiUIPredialIndividual"
type="urb:UIPredialIndividualPropertyType" minOccurs="0" maxOccurs="unbounded">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">pertenceALote</appinfo>
</annotation>
</element>
<element name="possuiUIPredialColetiva"
type="urb:UIPredialColetivaPropertyType" minOccurs="0" maxOccurs="unbounded">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">pertenceALote</appinfo>
</annotation>
</element>
<element name="formaZonaOODC"
type="urb:ZonaOODCPropertyType">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">coincideComLote</appinfo>
</annotation>
</element>
<element name="formaZonaPEUC"
type="urb:ZonaPEUCPropertyType">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">coincideComLote</appinfo>
</annotation>
</element>
<element name="possuiTestadaDeLote"
type="urb:TestadaDeLotePropertyType" maxOccurs="unbounded">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">pertenceALote</appinfo>
</annotation>
</element>

```

```

        </sequence>
    </extension>
</complexContent>
</complexType>
<complexType name="LotePropertyType">
    <sequence>
        <element ref="urb:Lote" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_LoteDeLoteamento" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="LoteDeLoteamento" type="urb:LoteDeLoteamentoType"
substitutionGroup="urb:_LoteDeLoteamento"/>
<complexType name="LoteDeLoteamentoType">
    <complexContent>
        <extension base="urb:LoteType">
            <sequence>
                <element ref="gml:Polygon"/>
                <element
name="sobrepostoPorQuadraDeLoteamento" type="urb:QuadraDeLoteamentoPropertyType"
maxOccurs="unbounded">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">sobrepoeloteDeLoteamento</appinfo>
                    </annotation>
                </element>
                <element name="sobrepostoPorLoteamento"
type="urb:LoteamentoPropertyType" maxOccurs="unbounded">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">sobrepoeloteDeLoteamento</appinfo>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="LoteDeLoteamentoPropertyType">
    <sequence>
        <element ref="urb:LoteDeLoteamento" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_ResponsavelTecnico" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="ResponsavelTecnico" type="urb:ResponsavelTecnicoType"
substitutionGroup="urb:_ResponsavelTecnico"/>
<complexType name="ResponsavelTecnicoType">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element name="codigoRespTecnico" type="integer"/>
                <element name="nomeCompleto" type="string"/>
                <element name="numeroCREA" type="string"/>
                <element name="endereco" type="string"/>
                <element name="telefone" type="string"/>
                <element name="responsavelPorConstrucao"

```

```

type="urb:ConstrucaoPropertyType" maxOccurs="unbounded">
    <annotation>
        <appinfo source="urn:x-
gml:reverseProperty">responsabilidadeDeResponsavelTecnico</appinfo>
    </annotation>
</element>
<element name="responsavelPorProjeto"
type="urb:ProjetoPropertyType" maxOccurs="unbounded">
    <annotation>
        <appinfo source="urn:x-
gml:reverseProperty">responsabilidadeDeResponsavelTecnico</appinfo>
    </annotation>
</element>
<element name="responsavelPorExecucaoDeProjeto"
type="urb:ExecucaoDeProjetoPropertyType" maxOccurs="unbounded">
    <annotation>
        <appinfo source="urn:x-
gml:reverseProperty">responsabilidadeDeResponsavelTecnico</appinfo>
    </annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="ResponsavelTecnicoPropertyType">
    <sequence>
        <element ref="urb:ResponsavelTecnico" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_Projeto" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="Projeto" type="urb:ProjetoType" substitutionGroup="urb:_Projeto"/>
<complexType name="ProjetoType">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element name="codigoProjeto" type="integer"/>
                <element name="codigoExecucao" type="integer"/>
                <element name="categoriaProjeto" type="integer"/> <!--
- Órgão Público/Profissional Liberal/Planta Popular/De Conjunto Popular Financiado -->
            </sequence>
            <element
name="responsabilidadeDeResponsavelTecnico"
type="urb:ResponsavelTecnicoPropertyType" maxOccurs="unbounded">
                <annotation>
                    <appinfo source="urn:x-
gml:reverseProperty">responsavelPorProjeto</appinfo>
                </annotation>
            </element>
            <element name="possuiExecucaoDeProjeto"
type="urb:ExecucaoDeProjetoPropertyType">
                <annotation>
                    <appinfo source="urn:x-
gml:reverseProperty">correspondeAProjeto</appinfo>
                </annotation>
            </element>
        </sequence>
    </extension>
</complexContent>

```

```

</complexType>
<complexType name="ProjetoPropertyType">
  <sequence>
    <element ref="urb:Projeto" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_ExecucaoDeProjeto" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="ExecucaoDeProjeto" type="urb:ExecucaoDeProjetoType"
substitutionGroup="urb:_ExecucaoDeProjeto"/>
<complexType name="ExecucaoDeProjetoType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="nome" type="string"/>
        <element
name="responsabilidadeDeResponsavelTecnico"
type="urb:ResponsavelTecnicoPropertyType" maxOccurs="unbounded">
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">responsavelPorExecucaoDeProjeto</appinfo>
          </annotation>
        </element>
        <element name="correspondeAProjeto"
type="urb:ProjetoPropertyType">
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">possuiExecucaoDeProjeto</appinfo>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="ExecucaoDeProjetoPropertyType">
  <sequence>
    <element ref="urb:ExecucaoDeProjeto" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_EspacoPublico" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="EspacoPublico" type="urb:EspacoPublicoType"
substitutionGroup="urb:_EspacoPublico"/>
<complexType name="EspacoPublicoType">
  <complexContent>
    <extension base="urb:UITerritorialType">
      <sequence>
        <element ref="gml:Polygon"/>
        <element name="codigoEspacoPublico"
type="integer"/>
        <element name="codigoUI" type="integer"/>
        <element name="categoria" type="string"/> <!-- Praça
Calçadão/Praça de Lazer/Praça Contemplativa/Praça Mista/Reserva/Parque -->
        <element name="destinacao" type="string"/> <!--
Institucional/Preservação/Dominial/Devoluta/Uso Especial -->
        <element name="patrimonio" type="string"/> <!--

```

```

Público Federal/Público Estadual/Público Municipal -->
    <element name="leiCriacao" type="string"/>
    <element name="dataCriacao" type="date"/>
    <element name="sanitarios" type="boolean"/>
    <element name="bancos" type="boolean"/>
    <element name="passeios" type="boolean"/>
    <element name="pavimentado" type="boolean"/>
    </sequence>
  </extension>
</complexContent>
</complexType>
<!------->
<element name="_UIPredialIndividual" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_OrgTerritorialFeature"/>
  <element name="UIPredialIndividual" type="urb:UIPredialIndividualType"
substitutionGroup="urb:_UIPredialIndividual"/>
  <complexType name="UIPredialIndividualType">
    <complexContent>
      <extension base="urb:UIPredialType">
        <sequence>
          <element ref="gml:Polygon"/>
          <element name="codigoUIPredialIndividual"
type="integer"/>
          <element name="codigoUI" type="integer"/>
          <element name="codigoLote" type="integer"/>
          <element name="ocupacao" type="string"/><!--
Residencial/Comercial/Prest.          Serviços/Residencial-Comercial/Residencial-Prest.
Serviços/Industrial/Templo/Serviço Público/Ensino-Cultura/Assist. e Serviço Social/Saúde
Pública/Clube/Esporte-Diversão/Associação de Classe/Agropecuário/Sem Uso -->
          <element name="tipoUIPredial" type="string"/><!--
Casa/Loja/Sala/Conjunto de Salas/Sobreloja-Subsolo/Casa com Loja/Casa com Sala/Barracão
Industrial/Galpão/Telheiro/Especial -->
          <element name="categoriaEstrutura" type="string"/><!--
- Metálica/Concreto/Alvenaria/Mista/Madeira -->
          <element name="cobertura" type="string"/><!--
Especial/Amianto/Alumínio/Zinco/Aço/Telha Cerâmica Colonial/Telha Cerâmica
Francesa/Laje/Palha-Cavaco/Madeira Tratada/Telha de Concreto -->
          <element name="vedacoes" type="string"/><!--
Especial/Tijolos Cerâmicos/Tijolos Concreto/Mista/Madeira/Sem -->
          <element name="revestimentoExterno"
type="string"/><!-- Tijolo Cerâmico Nú/Reboco/Tijolo à Vista/Especial -->
          <element name="instalacoesEletricas"
type="string"/><!-- Embutidas/Semi-Embutidas/Aparentes/Inexistentes -->
          <element name="pisos" type="string"/><!--
Especial/Taco/Cerâmica/Assoalho/Madeira/Tijolo/Concreto/Terra -->
          <element name="esquadrias" type="string"/><!--
Especial/Alumínio/Ferro/Madeira/PVC/Inexistente -->
          <element name="padraoAcabamento"
type="string"/><!-- Alto/Normal/Regular/Mau -->
          <element name="forro" type="string"/><!--
Especial/Eucatex/Laje/Madeira/Inexistente -->
          <element name="conservacao" type="string"/><!--
Boa/Regular/Ruim -->
          <element name="valorM2" type="float"/>
          <element name="emConstrucao" type="boolean"/>
          <element name="anoConstrucao" type="integer"/>
          <element name="areaEdificada" type="float"/>
          <element name="areaTerritorial" type="float"/>
          <element name="testadaProporcional" type="float"/>
          <element name="testadaCorrigidaPredial"

```

```

type="float"/>
        <element name="coeficientePredial" type="float"/>
        <element name="valorVenalUIPredial" type="float"/>
        <element
            name="inadimplênciaPredial"
            type="boolean"/>
        <element name="categoriaUso" type="string"/><!--
Residencial 01, 02, 03, 04, 05 Anexo III, Lei n. 1362 de 12/07/88, Prestação de Serviços 06,
07, ..., 19 Anexo III, Lei n. 1362 de 12/07/88, Comercial 20,..., 28 Anexo III, Lei n. 1362 de
12/07/88, Industrial 29, 30, 31 Anexo III, Lei n. 1362 de 12/07/88 -->
        <element
            name="pertenceALote"
            type="urb:LotePropertyType">
            <annotation>
                <appinfo
                    source="urn:x-
gml:reverseProperty">possuiUIPredialIndividual</appinfo>
            </annotation>
            </element>
            <element
                name="possuiEdificacaoDependente"
                type="urb:EdificacaoDependentePropertyType" minOccurs="0" maxOccurs="unbounded">
                <annotation>
                    <appinfo
                        source="urn:x-
gml:reverseProperty">pertenceAUIPredialIndividual</appinfo>
                </annotation>
            </element>
        </sequence>
    </extension>
</complexContent>
</complexType>
<complexType name="UIPredialIndividualPropertyType">
    <sequence>
        <element ref="urb:UIPredialIndividual" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====
<element name="_UIPredialColetiva" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element
    name="UIPredialColetiva"
    type="urb:UIPredialColetivaType"
substitutionGroup="urb:_UIPredialColetiva"/>
<complexType name="UIPredialColetivaType">
    <complexContent>
        <extension base="urb:UIPredialType">
            <sequence>
                <element ref="gml:Polygon"/>
                <element
                    name="codigoUIPredialIndividual"
                    type="integer"/>
                <element name="codigoUI" type="integer"/>
                <element name="codigoLote" type="integer"/>
                <element
                    name="ocupacao"
                    type="string"/><!--
Residencial/Comercial/Prest.
Serviços/Residencial-Comercial/Residencial-Prest.
Serviços/Industrial/Templo/Serviço Público/Ensino-Cultura/Assist. e Serviço Social/Saúde
Pública/Clube/Esporte-Diversão/Associação de Classe/Agropecuário/Sem Uso -->
                <element name="tipoUIPredial" type="string"/><!--
Casa/Loja/Sala/Conjunto de Salas/Sobreloja-Subsolo/Casa com Loja/Casa com Sala/Barracão
Industrial/Galpão/Telheiro/Especial -->
                <element name="categoriaEstrutura" type="string"/><!--
- Metálica/Concreto/Alvenaria/Mista/Madeira -->
                <element
                    name="cobertura"
                    type="string"/><!--
Especial/Amianto/Alumínio/Zinco/Aço/Telha
Cerâmica Colonial/Telha
Cerâmica Francesa/Laje/Palha-Cavaco/Madeira Tratada/Telha de Concreto -->

```



```

<element name="vedacoes" type="string"/><!--
Especial/Tijolos Cerâmicos/Tijolos Concreto/Mista/Madeira/Sem -->
<element name="revestimentoExterno"
type="string"/><!-- Tijolo Cerâmico Nú/Reboco/Tijolo à Vista/Especial -->
<element name="instalacoesEletricas"
type="string"/><!-- Embutidas/Semi-Embutidas/Aparentes/Inexistentes -->
<element name="pisos" type="string"/><!--
Especial/Taco/Cerâmica/Assoalho/Madeira/Tijolo/Concreto/Terra -->
<element name="esquadrias" type="string"/><!--
Especial/Alumínio/Ferro/Madeira/PVC/Inexistente -->
<element name="padraoAcabamento"
type="string"/><!-- Alto/Normal/Regular/Mau -->
<element name="forro" type="string"/><!--
Especial/Eucatex/Laje/Madeira/Inexistente -->
<element name="conservacao" type="string"/><!--
Boa/Regular/Ruim -->
<element name="valorM2" type="float"/>
<element name="emConstrucao" type="boolean"/>
<element name="anoConstrucao" type="integer"/>
<element name="areaPrivativa" type="float"/>
<element name="areaComum" type="float"/>
<element name="areaEdificada" type="float"/>
<element name="areaTerritorial" type="float"/>
<element name="testadaProporcional" type="float"/>
<element name="testadaCorrigidaPredial"
type="float"/>
<element name="coeficientePredial" type="float"/>
<element name="valorVenalUIPredial" type="float"/>
<element name="inadimplênciaPredial"
type="boolean"/>
<element name="categoriaUso" type="string"/><!--
Residencial 01, 02, 03, 04, 05 Anexo III, Lei n. 1362 de 12/07/88, Prestação de Serviços 06,
07, ..., 19 Anexo III, Lei n. 1362 de 12/07/88, Comercial 20,..., 28 Anexo III, Lei n. 1362 de
12/07/88, Industrial 29, 30, 31 Anexo III, Lei n. 1362 de 12/07/88 -->
<element name="pertenceALote"
type="urb:LotePropertyType">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">possuiUIPredialColetiva</appinfo>
</annotation>
</element>
<element name="possuiEdificacaoDependente"
type="urb:EdificacaoDependentePropertyType" minOccurs="0" maxOccurs="unbounded">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">pertenceAUIPredialColetiva</appinfo>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="UIPredialColetivaPropertyType">
<sequence>
<element ref="urb:UIPredialColetiva" minOccurs="0"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->

```

```

    <element name="_EdificacaoDependente" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_OrgTerritorialFeature"/>
    <element name="EdificacaoDependente" type="urb:EdificacaoDependenteType"
substitutionGroup="urb:_EdificacaoDependente"/>
    <complexType name="EdificacaoDependenteType">
        <complexContent>
            <extension base="urb:UIPredialType">
                <sequence>
                    <element ref="gml:Polygon"/>
                    <element name="codigoUIPredialDependente"
type="integer"/>
                    <element name="codigoUIPredialIndividual"
type="integer"/>
                    <element name="codigoUIPredialColetiva"
type="integer"/>
                    <element name="uso" type="string"/> <!--
Garagem/Área de Serviços/Churrasqueira/Mista/Outro Uso -->
                    <element name="tipoUIPredialDependente"
type="string"/> <!-- Edícula/Galpão/Telheiro -->
                    <element name="areaEdificada" type="float"/>
                    <element name="areaTerritorial" type="float"/>
                    <element name="testadaProporcional" type="float"/>
                    <element name="testadaCorrigidaPredial"
type="float"/>
                    <element name="pertenceAUIPredialColetiva"
type="urb:UIPredialColetivaPropertyType">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">possuiEdificacaoDependente</appinfo>
                        </annotation>
                    </element>
                    <element name="pertenceAUIPredialIndividual"
type="urb:UIPredialIndividualPropertyType">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">possuiEdificacaoDependente</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="EdificacaoDependentePropertyType">
        <sequence>
            <element ref="urb:EdificacaoDependente" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->
    <element name="_ZonaPEUC" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
    <element name="ZonaPEUC" type="urb:ZonaPEUCType"
substitutionGroup="urb:_ZonaPEUC"/>
    <complexType name="ZonaPEUCType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element ref="gml:Polygon"/>
                    <element name="codigoLote" type="integer"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

```

```

                                <element name="coincideComLote"
type="urb:LotePropertyType" maxOccurs="unbounded">
                                <annotation>
                                <appinfo source="urn:x-
gml:reverseProperty">formaZonaPEUC</appinfo>
                                </annotation>
                                </element>
                                </sequence>
                                </extension>
                                </complexContent>
                                </complexType>
                                <complexType name="ZonaPEUCPropertyType">
                                <sequence>
                                <element ref="urb:ZonaPEUC" minOccurs="0"/>
                                </sequence>
                                <attributeGroup ref="gml:AssociationAttributeGroup"/>
                                </complexType>
                                <!--=====-->
                                <element name="_ZonaOODC" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
                                <element name="ZonaOODC" type="urb:ZonaOODCType"
substitutionGroup="urb:_ZonaOODC"/>
                                <complexType name="ZonaOODCType">
                                <complexContent>
                                <extension base="gml:AbstractFeatureType">
                                <sequence>
                                <element ref="gml:Polygon"/>
                                <element name="codigoLote" type="integer"/>
                                <element name="coincideComLote"
type="urb:LotePropertyType" maxOccurs="unbounded">
                                <annotation>
                                <appinfo source="urn:x-
gml:reverseProperty">formaZonaOODC</appinfo>
                                </annotation>
                                </element>
                                </sequence>
                                </extension>
                                </complexContent>
                                </complexType>
                                <complexType name="ZonaOODCPropertyType">
                                <sequence>
                                <element ref="urb:ZonaOODC" minOccurs="0"/>
                                </sequence>
                                <attributeGroup ref="gml:AssociationAttributeGroup"/>
                                </complexType>
                                <!--=====-->
                                <element name="_TestadaDeLote" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
                                <element name="TestadaDeLote" type="urb:TestadaDeLoteType"
substitutionGroup="urb:_TestadaDeLote"/>
                                <complexType name="TestadaDeLoteType">
                                <complexContent>
                                <extension base="gml:AbstractFeatureType">
                                <sequence>
                                <element name="pertenceALote"
type="urb:LotePropertyType">
                                <annotation>
                                <appinfo source="urn:x-
gml:reverseProperty">possuiTestadaDeLote</appinfo>
                                </annotation>

```

```

                </element>
                <element name="possuiPasseio"
type="urb:PasseioPropertyType" minOccurs="0">
                    <annotation
                        <appinfo source="urn:x-
gml:reverseProperty">pertenceATestadaDeLote</appinfo>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="TestadaDeLotePropertyType">
    <sequence>
        <element ref="urb:TestadaDeLote" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_TestadaSecundariaDeLote" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="TestadaSecundariaDeLote"
type="urb:TestadaSecundariaDeLoteType"
substitutionGroup="urb:_TestadaSecundariaDeLote"/>
<complexType name="TestadaSecundariaDeLoteType">
    <complexContent>
        <extension base="urb:TestadaDeLoteType">
            <sequence>
                <element name="codigoLote" type="integer"/>
                <element name="extensao" type="float"/>
                <element name="muro" type="boolean"/>
                <element name="cerca" type="boolean"/>
                <element name="meioFio" type="boolean"/>
                <element name="categoriaRevestimentoPasseio"
type="string"/> <!-- Padrão/Ladrilho Hidráulico/Cimento/Pedra/Cerâmico/Misto/Com Floreira/
Sem Revestimento -->
                <element name="larguraPredominantePasseio"
type="float"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<!--=====-->
<element name="_TestadaPrincipalDeLote" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="TestadaPrincipalDeLote" type="urb:TestadaPrincipalDeLoteType"
substitutionGroup="urb:_TestadaPrincipalDeLote"/>
<complexType name="TestadaPrincipalDeLoteType">
    <complexContent>
        <extension base="urb:TestadaDeLoteType">
            <sequence>
                <element name="codigoTestadaPrincipal"
type="integer"/>
                <element name="codigoLote" type="integer"/>
                <element name="codigoSegmentoLogradouro"
type="integer"/>
                <element name="extensao" type="float"/>
                <element name="muro" type="boolean"/>
                <element name="cerca" type="boolean"/>
                <element name="meioFio" type="boolean"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

type="string"/> <!-- Padrão/Ladrilho Hidráulico/Cimento/Pedra/Cerâmico/Misto/Com Floreira/
Sem Revestimento -->
<element name="categoriaRevestimentoPasseio"
type="float"/>
<element name="larguraPredominantePasseio"
type="urb:SegmentoDeLogradouroPropertyType">
<annotation>
<appinfo source="urn:x-
gml:reverseProperty">pertenceATestadaPrincipalDeLote</appinfo>
</annotation>
</element>
</sequence>
</extension>
</complexContent>
</complexType>
<complexType name="TestadaPrincipalDeLotePropertyType">
<sequence>
<element ref="urb:TestadaPrincipalDeLote" minOccurs="0"/>
</sequence>
<attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!------->
<element name="_SegmentoDeLogradouro" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="SegmentoDeLogradouro" type="urb:SegmentoDeLogradouroType"
substitutionGroup="urb:_SegmentoDeLogradouro"/>
<complexType name="SegmentoDeLogradouroType">
<complexContent>
<extension base="gml:AbstractFeatureType">
<sequence>
<element ref="gml:LineString"/>
<element name="codigoSegmentoLogradouro"
type="integer"/>
<element name="codigoLogradouro" type="integer"/>
<element name="situacaoAtual" type="string"/><!--
Real/Projetado -->
<element name="pavimentacao" type="string"/><!--
Asfalto/Lajota/Paralelepípedo/Revestimento Primário/Inexistente -->
<element name="conservacaoSuperficie"
type="string"/><!-- Boa/Regular/Ruim -->
<element name="coletaLixo" type="string"/><!-- até 12
coletas-mês/13-24 coletas-mês/Inexistente -->
<element name="limpezaPublica" type="boolean"/>
<element name="conservacaoVias" type="boolean"/>
<element name="redeAgua" type="boolean"/>
<element name="redelluminacaoPublica"
type="boolean"/>
<element name="redeLuzDomiciliar" type="boolean"/>
<element name="redeEsgoto" type="string"/> <!--
Cloacal/Pluvial/Pluvial-Cloacal/Inexistente -->
<element name="larguraVia" type="float"/>
<element name="exensao" type="float"/>
<element name="codigoEnderecamentoPostal"
type="string"/>
<element name="pertenceATestadaPrincipalDeLote"
type="urb:TestadaPrincipalDeLotePropertyType">
<annotation>
<appinfo source="urn:x-

```

```

gml:reverseProperty">possuiSegmentoDeLogradouro</appinfo>
      </annotation>
    </element>
    <element name="possuiCanteiro"
type="urb:CanteiroPropertyType" minOccurs="0" maxOccurs="unbounded">
      <annotation>
        <appinfo source="urn:x-
gml:reverseProperty">pertenceASegmentoDeLogradouro</appinfo>
      </annotation>
    </element>
    <element name="pertenceALogradouro"
type="urb:SegmentoDeLogradouroPropertyType" minOccurs="1" maxOccurs="1">
      <annotation>
        <appinfo source="urn:x-
gml:reverseProperty">possuiSegmentoDeLogradouro</appinfo>
      </annotation>
    </element>
  </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="SegmentoDeLogradouroPropertyType">
  <sequence>
    <element ref="urb:SegmentoDeLogradouro" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_Passeio" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="Passeio" type="urb:PasseioType"
substitutionGroup="urb:_Passeio"/>
<complexType name="PasseioType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="pertenceATestadaDeLote"
type="urb:TestadaDeLotePropertyType">
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">possuiPasseio</appinfo>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="PasseioPropertyType">
  <sequence>
    <element ref="urb:Passeio" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_Canteiro" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="Canteiro" type="urb:CanteiroType"
substitutionGroup="urb:_Canteiro"/>
<complexType name="CanteiroType">
  <complexContent>

```

```

        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element ref="gml:Polygon"/>
                <element name="codigoCanteiro" type="integer"/>
                <element name="codigoSegmentoLogradouro"
type="integer"/>
                <element name="patrimonio" type="string"/> <!--
Público Federal/Público Estadual/Público Municipal -->
                <element name="categoria" type="string"/> <!-- Faixa
de Domínio/Rótula/Canteiro Central -->
                <element name="pertenceASegmentoDeLogradouro"
type="urb:SegmentoDeLogradouroPropertyType">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">possuiCanteiro</appinfo>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="CanteiroPropertyType">
    <sequence>
        <element ref="urb:Canteiro" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_Logradouro" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_OrgTerritorialFeature"/>
<element name="Logradouro" type="urb:LogradouroType"
substitutionGroup="urb:_Logradouro"/>
<complexType name="LogradouroType">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element ref="gml:LineString"/>
                <element name="codigoLogradouro" type="integer"/>
                <element name="nome" type="string"/>
                <element name="categoria" type="integer"/><!--
Rua/Avenida/Servidão/Alameda/Beco/Ferrovia/Rodovia Municipal/Rodovia Estadual/Rodovia
Federal/Privativa/Caminho/Escadaria/Passarela sobre Rio/Passarela sobre Via -->
                <element name="extensao" type="float"/>
                <element name="possuiSegmentoDeLogradouro"
type="urb:SegmentoDeLogradouroPropertyType" minOccurs="1" maxOccurs="unbounded">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">pertenceALogradouro</appinfo>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="LogradouroPropertyType">
    <sequence>
        <element ref="urb:Logradouro" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

```
<!------->
</schema>
```

## C – Esquema GML de Belo Horizonte

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:gml="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:urb="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ns1="http://www.opengis.net/gml" targetNamespace="quodras_lote" elementFormDefault="qualified" version="2007-01-27">
  <!------->
  <import namespace="http://www.opengis.net/gml"
    schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd"/>
  <import namespace="http://www.opengis.net/gml"
    schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gmlBase.xsd"/>
  <import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="http://schemas.opengis.net/gml/3.1.1/xlink/xlinks.xsd"/>
  <import/>
  <!------->
  <!-- =====
  PACK QUADRAS
  ===== -->
  <element name="QuodrasPack" type="urb:QuodrasPackType"
    substitutionGroup="gml:_FeatureCollection"/>
  <complexType name="QuodrasPackType">
    <complexContent>
      <extension base="gml:AbstractFeatureCollectionType"/>
    </complexContent>
  </complexType>
  <element name="QuodrasMember" type="urb:QuodrasPropertyType"
    substitutionGroup="gml:featureMember"/>
  <element name="_QuodrasFeature" type="gml:AbstractFeatureType" abstract="true"
    substitutionGroup="gml:_Feature"/>
  <complexType name="QuodrasPropertyType">
    <complexContent>
      <restriction base="gml:FeaturePropertyType">
        <sequence minOccurs="0">
          <element ref="urb:_QuodrasFeature"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- =====
  CLASSES FROM QUADRAS
  ===== -->
  <element name="_QuadraCTM" type="gml:AbstractFeatureType" abstract="true"
    substitutionGroup="urb:_QuodrasFeature"/>
  <element name="QuadraCTM" type="urb:QuadraCTMType"
    substitutionGroup="urb:_QuadraCTM"/>
  <complexType name="QuadraCTMType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:Polygon"/>
          <element name="numeroSetor" type="integer"/>
          <element name="numeroQuadra" type="integer"/>
          <element name="bairro" type="string"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ----->
```



```

                <element name="constituídoDeLotes"
type="urb:LoteAreaPropertyType" minOccurs="1" maxOccurs="unbounded">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">agregadoDe</appinfo>
                    </annotation>
                </element>
                <element name="possuiLote"
type="urb:LotePropertyType" minOccurs="1" maxOccurs="unbounded">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">pertenceAQuadraCTM</appinfo>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="QuadraCTMPropertyType">
    <sequence>
        <element ref="urb:QuadraCTM" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_QuadraProjetada" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_QuadrasFeature"/>
<element name="QuadraProjetada" type="urb:QuadraProjetadaType"
substitutionGroup="urb:_QuadraProjetada"/>
<complexType name="QuadraProjetadaType">
    <annotation>
        <documentation> Herda de QuadraCTM</documentation>
    </annotation>
    <complexContent>
        <extension base="urb:QuadraCTMType">
            <sequence>
                <element ref="gml:Polygon"/>
                <element name="numeroPlantaProjetada"
type="integer"/>
                <element name="codigoLogradouroProjetado"
type="integer" maxOccurs="unbounded"/>
                <element name="dataAprovacao" type="date"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<element name="_QuadraReal" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_QuadrasFeature"/>
<element name="QuadraReal" type="urb:QuadraRealType"
substitutionGroup="urb:_QuadraReal"/>
<complexType name="QuadraRealType">
    <annotation>
        <documentation> Herda de QuadraCTM</documentation>
    </annotation>
    <complexContent>
        <extension base="urb:QuadraCTMType">
            <sequence>
                <element ref="gml:Polygon"/>
                <element name="numeroPlantaReal" type="integer"/>
                <element name="codigoLogradouroReal"
type="integer" maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

type="integer" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- =====
PACK LOTES
===== -->
  <element name="LotesPack" type="urb:LotesType"
substitutionGroup="gml:_FeatureCollection"/>
  <complexType name="LotesType">
    <complexContent>
      <extension base="gml:AbstractFeatureCollectionType"/>
    </complexContent>
  </complexType>
  <element name="_LotesFeature" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="gml:_Feature"/>
  <element name="LotesMember" type="urb:LotesPropertyType"
substitutionGroup="gml:featureMember"/>
  <complexType name="LotesPropertyType">
    <complexContent>
      <restriction base="gml:FeaturePropertyType">
        <sequence minOccurs="0">
          <element ref="urb:_LotesFeature"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
<!-- =====
CLASSES FROM LOTE
===== -->
  <element name="_Cemiterio" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
  <element name="Cemiterio" type="urb:CemiterioType"
substitutionGroup="urb:_Cemiterio"/>
  <complexType name="CemiterioType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:Polygon"/>
          <element name="numeroLotes" type="integer"/>
          <element name="sobreLoteArea"
type="urb:LoteAreaPropertyType" minOccurs="1" maxOccurs="1">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">existeCemiterio</appinfo>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="CemiterioPropertyType">
    <sequence>
      <element ref="urb:Cemiterio" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
<!-- =====>

```

```

    <element name="_CEP" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
    <element name="CEP" type="urb:CEPType" substitutionGroup="urb:_CEP"/>
    <complexType name="CEPType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element name="codigoCEP" type="string"/>
                    <element name="localizaEndereco"
type="urb:EnderecoPropertyType" minOccurs="0" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">possuiCEP</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="CEPPropertyType">
        <sequence>
            <element ref="urb:CEP" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->
    <element name="_Compatibilizacao" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
    <element name="Compatibilizacao" type="urb:CompatibilizacaoType"
substitutionGroup="urb:_Compatibilizacao"/>
    <complexType name="CompatibilizacaoType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element name="numeroPlantaPrefeitura"
type="integer"/>
                    <element name="numeroRegistro" type="string"/>
                    <element name="compatibilizaLoteTributavel"
type="urb:LoteTributavelPropertyType" minOccurs="0" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">ehCompativel</appinfo>
                        </annotation>
                    </element>
                    <element name="compatibilizaLoteLegal"
type="urb:LoteLegalPropertyType">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">ehCompativel</appinfo>
                        </annotation>
                    </element>
                    <element name="compatibilizaLoteCTM"
type="urb:LoteCTMPropertyType" minOccurs="0" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">ehCompativel</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

```

```

        </complexContent>
    </complexType>
    <complexType name="CompatibilizacaoPropertyType">
        <sequence>
            <element ref="urb:Compatibilizacao" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->
    <!-- Generalização -->
    <element name="_DivisaLote" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
    <element name="DivisaLote" type="urb:DivisaLoteType"
substitutionGroup="urb:_DivisaLote"/>
    <complexType name="DivisaLoteType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element ref="gml:LineString"/>
                    <element name="pertenceLoteArea"
type="urb:LoteAreaPropertyType" minOccurs="1" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">possuiDivisaLote</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="DivisaLotePropertyType">
        <sequence>
            <element ref="urb:DivisaLote" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->
    <!-- Especialização disjunta / total -->
    <element name="_Cerca" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
    <element name="Cerca" type="urb:CercaType" substitutionGroup="urb:_Cerca"/>
    <complexType name="CercaType">
        <annotation>
            <documentation>Herda de DivisaLote</documentation>
        </annotation>
        <complexContent>
            <extension base="urb:DivisaLoteType">
                <sequence>
                    <element ref="gml:LineString"/>
                    <element name="tipoMaterial" type="integer"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <!--=====-->
    <!-- Especialização disjunta / total -->
    <element name="_Muro" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
    <element name="Muro" type="urb:MuroType" substitutionGroup="urb:_Muro"/>
    <complexType name="MuroType">

```

```

<annotation>
  <documentation> Herda de DivisaLote</documentation>
</annotation>
<complexContent>
  <extension base="urb:DivisaLoteType">
    <sequence>
      <element ref="gml:LineString"/>
      <element name="tipoConstrucao" type="integer"/>
    </sequence>
  </extension>
</complexContent>
</complexType>
<!--=====-->
<element name="_Endereco" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
  <element name="Endereco" type="urb:EnderecoType"
substitutionGroup="urb:_Endereco"/>
  <complexType name="EnderecoType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:Point"/>
          <element name="nomeLogradouro" type="string"/>
          <element name="numero" type="integer"/>
          <element name="dentroDeLoteArea"
type="urb:LoteAreaPropertyType" minOccurs="1" maxOccurs="1">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">possuiEndereco</appinfo>
            </annotation>
          </element>
          <element name="permiteOcupacaolmovel"
type="urb:OcupacaolmovelPropertyType" minOccurs="0" maxOccurs="unbounded">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">sobreEndereco</appinfo>
            </annotation>
          </element>
          <element name="possuiCEP"
type="urb:CEPPROPERTYTYPE" minOccurs="1" maxOccurs="1">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">localizaEndereco</appinfo>
            </annotation>
          </element>
          <element name="estaEmLoteTestada"
type="urb:LoteTestadaPropertyType" minOccurs="1" maxOccurs="1">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">possuiEndereco</appinfo>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
<complexType name="EnderecoPropertyType">
  <sequence>
    <element ref="urb:Endereco" minOccurs="0"/>
  </sequence>

```

```

        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->
    <element name="_OcupacaoImovel" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
    <element name="OcupacaoImovel" type="urb:OcupacaoImovelType"
substitutionGroup="urb:_OcupacaoImovel"/>
    <complexType name="OcupacaoImovelType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element ref="gml:Point"/>
                    <element name="tipoOcupacao" type="integer"/>
                    <element name="estaContidaLoteArea"
type="urb:LoteAreaPropertyType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">contemOcupacaoImovel</appinfo>
                        </annotation>
                    </element>
                    <element name="sobreEndereco"
type="urb:EnderecoPropertyType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">permiteOcupacaoImovel</appinfo>
                        </annotation>
                    </element>
                    <element name="possuiLoteTestada"
type="urb:LoteTestadaPropertyType" minOccurs="1" maxOccurs="1">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">possuiOcupacaoImovel</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="OcupacaoImovelPropertyType">
        <sequence>
            <element ref="urb:OcupacaoImovel" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->
    <element name="_LoteTributavel" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
    <element name="LoteTributavel" type="urb:LoteTributavelType"
substitutionGroup="urb:_LoteTributavel"/>
    <complexType name="LoteTributavelType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element name="numeroRegistroPrefeitura"
type="string"/>
                    <element name="ehLote" type="urb:LotePropertyType"
minOccurs="0" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">ehLoteTributavel</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

```

```

                </annotation>
            </element>
            <element name="ehCompative"
type="urb:CompatibilizacaoPropertyType" minOccurs="0" maxOccurs="unbounded">
                <annotation>
                    <appinfo source="urn:x-
gml:reverseProperty">compatibilizaLoteTributavel</appinfo>
                </annotation>
            </element>
        </sequence>
    </extension>
</complexContent>
</complexType>
<complexType name="LoteTributavelPropertyType">
    <sequence>
        <element ref="urb:LoteTributavel" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<element name="_LoteLegal" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
<element name="LoteLegal" type="urb:LoteLegalType"
substitutionGroup="urb:_LoteLegal"/>
<complexType name="LoteLegalType">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element name="numeroRegistroCartorio"
type="string"/>
                <element name="ehLote" type="urb:LotePropertyType"
minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">ehLoteLegal</appinfo>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="LoteLegalPropertyType">
    <sequence>
        <element ref="urb:LoteLegal" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<!-- Generalização -->
<element name="_LoteCTM" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
<element name="LoteCTM" type="urb:LoteCTMType"
substitutionGroup="urb:_LoteCTM"/>

```

```

<complexType name="LoteCTMType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="ehLote" type="urb:LotePropertyType"
minOccurs="0" maxOccurs="unbounded">
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">ehLoteCTM</appinfo>
          </annotation>
        </element>
        <element name="ehCompativel"
type="urb:CompatibilizacaoPropertyType" minOccurs="0" maxOccurs="unbounded">
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">compatibilizaLoteCTM</appinfo>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="LoteCTMPropertyType">
  <sequence>
    <element ref="urb:LoteCTM" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<!-- Especialização disjunta / total -->
<element name="_LoteCTMProjetado" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_LotesFeature"/>
<element name="LoteCTMProjetado" type="urb:LoteCTMProjetadoType"
substitutionGroup="urb:_LoteCTMProjetado"/>
<complexType name="LoteCTMProjetadoType">
  <annotation>
    <documentation> Herda de LoteCTM</documentation>
  </annotation>
  <complexContent>
    <extension base="urb:LoteCTMType">
      <sequence>
        <element name="numeroPlantaProjetado"
type="integer"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!--=====-->
<!-- Especialização disjunta / total -->
<element name="_LoteCTMReal" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
<element name="LoteCTMReal" type="urb:LoteCTMRealType"
substitutionGroup="urb:_LoteCTMReal"/>
<complexType name="LoteCTMRealType">
  <annotation>
    <documentation> Herda de LoteCTM</documentation>
  </annotation>
  <complexContent>
    <extension base="urb:LoteCTMType">
      <sequence>

```



```

                <element name="numeroPlantaReal" type="integer"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<!--=====-->
<!-- Generalização -->
<element name="_TestadaLote" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
<element name="TestadaLote" type="urb:TestadaLoteType"
substitutionGroup="urb:_TestadaLote"/>
<complexType name="TestadaLoteType">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element ref="gml:LineString"/>
                <element name="tamanhoTestada" type="float"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<!--=====-->
<!-- Especialização disjunta / parcial -->
<element name="_TestadaPrincipal" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
<element name="TestadaPrincipal" type="urb:TestadaPrincipalType"
substitutionGroup="urb:_TestadaPrincipal"/>
<complexType name="TestadaPrincipalType">
    <annotation>
        <documentation> Herda de TestadaLote</documentation>
    </annotation>
    <complexContent>
        <extension base="urb:TestadaLoteType">
            <sequence>
                <element ref="gml:LineString"/>
                <element name="metragemFrente" type="float"/>
                <element name="numeroSequenciaTrecho"
type="integer"/>
                <element name="agregadoDeLoteTestada"
type="urb:LoteTestadaPropertyType" minOccurs="1" maxOccurs="1">
                    <annotation>
                        <appinfo source="urn:x-
gml:reverseProperty">constituídoDeTestadaPrincipal</appinfo>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="TestadaPrincipalPropertyType">
    <sequence>
        <element ref="urb:TestadaPrincipal" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->
<!-- Especialização disjunta / parcial -->
<element name="_TestadaSecundaria" type="gml:AbstractFeatureType"

```

```

abstract="true" substitutionGroup="urb:_LotesFeature"/>
  <element name="TestadaSecundaria" type="urb:TestadaSecundariaType"
substitutionGroup="urb:_TestadaSecundaria"/>
  <complexType name="TestadaSecundariaType">
    <annotation>
      <documentation> Herda de TestadaLote</documentation>
    </annotation>
    <complexContent>
      <extension base="urb:TestadaLoteType">
        <sequence>
          <element ref="gml:LineString"/>
          <element name="metragemSecundaria" type="float"/>

          <element name="agregadoDeLoteTestada"
type="urb:LoteTestadaPropertyType" minOccurs="1" maxOccurs="1">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">constituídoDeTestadaSecundaria</appinfo>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="TestadaSecundariaPropertyType">
    <sequence>
      <element ref="urb:TestadaSecundaria" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>

<!--=====-->
<!-- SUPERCLASSE DE GENERALIZAÇÃO CONCEITUAL SOBREPOSTA DE
FORMA -->
  <element name="_Lote" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
  <element name="Lote" type="urb:LoteType" substitutionGroup="urb:_Lote"/>
  <complexType name="LoteType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="numeroQuadra" type="integer"/>
          <element name="numeroSetor" type="integer"/>
          <element name="numeroLote" type="integer"/>
          <element name="ehDefinido"
type="urb:LoteTributavelPropertyType">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">define</appinfo>
            </annotation>
          </element>
          <element name="pertenceAQuadraCTM"
type="urb:QuadraCTMPropertyType" minOccurs="1" maxOccurs="1">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">possuiLote</appinfo>
            </annotation>
          </element>
          <element name="ehLoteLegal"
type="urb:LoteLegalPropertyType" minOccurs="0" maxOccurs="unbounded">

```

```

                                <annotation>
                                <appinfo
                                source="urn:x-
gml:reverseProperty">ehLote</appinfo>
                                </annotation>
                                </element>
                                <element
                                name="ehLoteTributavel"
type="urb:LoteTributavelPropertyType" minOccurs="0" maxOccurs="unbounded">
                                <annotation>
                                <appinfo
                                source="urn:x-
gml:reverseProperty">ehLote</appinfo>
                                </annotation>
                                </element>
                                <element
                                name="ehLoteCTM"
type="urb:LoteCTMPropertyType" minOccurs="0" maxOccurs="unbounded">
                                <annotation>
                                <appinfo
                                source="urn:x-
gml:reverseProperty">ehLote</appinfo>
                                </annotation>
                                </element>
                                </sequence>
                                </extension>
                                </complexContent>
                                </complexType>
                                <complexType name="LotePropertyType">
                                <sequence>
                                <element ref="urb:Lote" minOccurs="0"/>
                                </sequence>
                                <attributeGroup ref="gml:AssociationAttributeGroup"/>
                                </complexType>
                                <!--=====-->
                                <!-- SUBCLASSE DE GENERALIZAÇÃO CONCEITUAL SOBREPOSTA DE FORMA
-->
                                <element name="_LoteArea" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
                                <element
                                name="LoteArea"
                                type="urb:LoteAreaType"
substitutionGroup="urb:_LoteArea"/>
                                <complexType name="LoteAreaType">
                                <complexContent>
                                <annotation>
                                <documentation>
                                Representa uma sub-classe de Lote, com representação
geométrica de área.
                                </documentation>
                                </annotation>
                                <extension base="urb:LoteType">
                                <sequence>
                                <element ref="gml:Polygon"/>
                                <element name="areaLote" type="float"/>
                                </sequence>
                                </extension>
                                </complexContent>
                                </complexType>
                                </sequence>
                                </complexType>
                                <complexType name="DivisaLotePropertyType" minOccurs="0" maxOccurs="unbounded">
                                <annotation>
                                <appinfo
                                source="urn:x-
gml:reverseProperty">pertenceLoteArea</appinfo>
                                </annotation>
                                </element>
                                </sequence>
                                </complexType>
                                <complexType name="OcupacaoImovelPropertyType" minOccurs="1" maxOccurs="unbounded">
                                <annotation>
                                <appinfo
                                source="urn:x-

```

```

gml:reverseProperty">estaContidaLoteArea</appinfo>
      </annotation>
    </element>
    <element name="existeCemiterio"
type="urb:CemiterioPropertyType" minOccurs="0" maxOccurs="1">
      <annotation>
        <appinfo source="urn:x-
gml:reverseProperty">sobreLoteArea</appinfo>
          </annotation>
        </element>
        <element name="possuiEndereco"
type="urb:EnderecoPropertyType" minOccurs="0" maxOccurs="unbounded">
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">dentroDeLoteArea</appinfo>
              </annotation>
            </element>
            <element name="agregadoDe"
type="urb:QuadraCTMPPropertyType">
              <annotation>
                <appinfo source="urn:x-
gml:reverseProperty">constituídoDeLotes</appinfo>
                  </annotation>
                </element>
              </sequence>
            </extension>
          </complexContent>
        </complexType>
        <complexType name="LoteAreaPropertyType">
          <sequence>
            <element ref="urb:LoteArea" minOccurs="0"/>
          </sequence>
          <attributeGroup ref="gml:AssociationAttributeGroup"/>
        </complexType>
        <!--=====-->
        <!-- SUBCLASSE DE GENERALIZAÇÃO CONCEITUAL SOBREPOSTA DE FORMA
-->
        <element name="_LoteTestada" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_LotesFeature"/>
        <element name="LoteTestada" type="urb:LoteTestadaType"
substitutionGroup="urb:_LoteTestada"/>
        <complexType name="LoteTestadaType">
          <complexContent>
            <annotation>
              <documentation>
                Representa uma sub-classe de Lote, com representação
geométrica de linha.
              </documentation>
            </annotation>
            <extension base="urb:LoteType">
              <sequence>
                <element ref="gml:LineString"/>
                <element name="perimetroLote" type="float"/>
                <element name="possuiOcupacaoImovel"
type="urb:OcupacaoImovelPropertyType" minOccurs="0" maxOccurs="unbounded">
                  <annotation>
                    <appinfo source="urn:x-
gml:reverseProperty">possuiLoteTestada</appinfo>
                      </annotation>
                    </element>

```

```

        <element name="possuiEndereco"
type="urb:EnderecoPropertyType" minOccurs="0" maxOccurs="unbounded">
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">estaEmLoteTestada</appinfo>
          </annotation>
        </element>
        <element name="constituídoDeTestadaPrincipal"
type="urb:TestadaPrincipalPropertyType" minOccurs="1" maxOccurs="1">
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">agregadoDeLoteTestada</appinfo>
          </annotation>
        </element>
        <element name="constituídoDeTestadaSecundaria"
type="urb:TestadaSecundariaPropertyType" minOccurs="0" maxOccurs="unbounded">
          <annotation>
            <appinfo source="urn:x-
gml:reverseProperty">agregadoDeLoteTestada</appinfo>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="LoteTestadaPropertyType">
  <sequence>
    <element ref="urb:LoteTestada" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
</schema>

```

## D – Esquema GML de [CAMARA 1999]

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:urb="LivroBDG" xmlns:ns1="http://www.opengis.net/gml" targetNamespace="LivroBDG"
elementFormDefault="qualified" version="2007-02-25">
  <!--=====-->
  <import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/feature.xsd"/>
  <import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gmlBase.xsd"/>
  <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/xlink/xlinks.xsd"/>
  <import/>
  <!--=====-->
  <!--=====
PACK CTM
===== -->
  <element name="CTMPack" type="urb:CTMType"
substitutionGroup="gml:_FeatureCollection"/>
  <complexType name="CTMType">
    <complexContent>
      <extension base="gml:AbstractFeatureCollectionType"/>
    </complexContent>

```

```

    </complexType>
    <element name="_CTMFeature" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="gml:_Feature"/>
    <element name="CTMMember" type="urb:CTMPropertyType"
substitutionGroup="gml:featureMember"/>
    <complexType name="CTMPropertyType">
        <complexContent>
            <restriction base="gml:FeaturePropertyType">
                <sequence minOccurs="0">
                    <element ref="urb:_CTMFeature"/>
                </sequence>
                <attributeGroup ref="gml:AssociationAttributeGroup"/>
            </restriction>
        </complexContent>
    </complexType>
<!-- =====
CLASSES CTM
===== -->
    <element name="_Municipio" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
    <element name="Municipio" type="urb:MunicipioType"
substitutionGroup="urb:_Municipio"/>
    <complexType name="MunicipioType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element name="codMunicipioIBGE" type="string"/>
                    <element name="populacaoMunicipio" type="integer"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

<!--=====----->
    <element name="_FronteirasMunicipais" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_CTMFeature"/>
    <element name="FronteirasMunicipais" type="urb:FronteirasMunicipaisType"
substitutionGroup="urb:_FronteirasMunicipais"/>
    <complexType name="FronteirasMunicipaisType">
        <complexContent>
            <extension base="urb:MunicipioType">
                <sequence>
                    <element ref="gml:Polygon"/>
                    <element name="area" type="float"/>
                    <element name="contemManchaUrbana"
type="urb:ManchaUrbanaPropertyType" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">contidoEmFronteirasMunicipais</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="FronteirasMunicipaisPropertyType">
        <sequence>
            <element ref="urb:FronteirasMunicipais" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>

```

```

</complexType>
<!--=====-->

  <element name="_FronteirasMunicipaisSimplificadas" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_CTMFeature"/>
  <element name="FronteirasMunicipaisSimplificadas"
type="urb:FronteirasMunicipaisSimplificadasType"
substitutionGroup="urb:_FronteirasMunicipaisSimplificadas"/>
    <complexType name="FronteirasMunicipaisSimplificadasType">
      <complexContent>
        <extension base="urb:MunicipioType">
          <sequence>
            <element ref="gml:Polygon"/>
            <element name="ehServidaPorRodovia"
type="urb:RodoviaPropertyType" maxOccurs="unbounded">
              <annotation>
                <appinfo source="urn:x-
gml:reverseProperty">serveAFronteirasMunicipais</appinfo>
              </annotation>
            </element>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  <complexType name="FronteirasMunicipaisSimplificadasPropertyType">
    <sequence>
      <element ref="urb:FronteirasMunicipaisSimplificadas" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
<!--=====-->

  <element name="_CidadePonto" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
  <element name="CidadePonto" type="urb:CidadePontoType"
substitutionGroup="urb:_CidadePonto"/>
    <complexType name="CidadePontoType">
      <complexContent>
        <extension base="urb:MunicipioType">
          <sequence>
            <element ref="gml:Point"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
<!--=====-->

  <element name="_Rodovia" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
  <element name="Rodovia" type="urb:RodoviaType"
substitutionGroup="urb:_Rodovia"/>
    <complexType name="RodoviaType">
      <complexContent>
        <extension base="gml:AbstractFeatureType">
          <sequence>
            <element ref="gml:LineString"/>
            <element name="numLogradouro" type="integer"/>
            <element name="serveAFronteirasMunicipais"
type="urb:FronteirasMunicipaisPropertyType" maxOccurs="unbounded">
              <annotation>

```

```

                                <appinfo source="urn:x-
gml:reverseProperty">ehServidaPorRodovia</appinfo>
                                </annotation>
                                </element>

                                <element name="sobrepostaAViaPrincipal"
type="urb:ViaPrincipalPropertyType" maxOccurs="unbounded">
                                <annotation>
                                <appinfo source="urn:x-
gml:reverseProperty">sobrepoerodovia</appinfo>
                                </annotation>
                                </element>

                                </sequence>
                                </extension>
                                </complexContent>
                                </complexType>
                                <complexType name="RodoviaPropertyType">
                                <sequence>
                                <element ref="urb:Rodovia" minOccurs="0"/>
                                </sequence>
                                <attributeGroup ref="gml:AssociationAttributeGroup"/>
                                </complexType>
                                <!--=====-->

                                <element name="_ManchaUrbana" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
                                <element name="ManchaUrbana" type="urb:ManchaUrbanaType"
substitutionGroup="urb:_ManchaUrbana"/>
                                <complexType name="ManchaUrbanaType">
                                <complexContent>
                                <extension base="gml:AbstractFeatureType">
                                <sequence>
                                <element ref="gml:Polygon"/>
                                <element name="contidoEmFronteirasMunicipais"
type="urb:FronteirasMunicipaisPropertyType">
                                <annotation>
                                <appinfo source="urn:x-
gml:reverseProperty">contemManchaUrbana</appinfo>
                                </annotation>
                                </element>
                                <element name="contemQuadraCTM"
type="urb:QuadraCTMPropertyType">
                                <annotation>
                                <appinfo source="urn:x-
gml:reverseProperty">contindaEmManchaUrbana</appinfo>
                                </annotation>
                                </element>
                                </sequence>
                                </extension>
                                </complexContent>
                                </complexType>
                                <complexType name="ManchaUrbanaPropertyType">
                                <sequence>
                                <element ref="urb:ManchaUrbana" minOccurs="0"/>
                                </sequence>
                                <attributeGroup ref="gml:AssociationAttributeGroup"/>
                                </complexType>
                                <!--=====-->

```



```

    <element name="_ViaPrincipal" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
    <element name="ViaPrincipal" type="urb:ViaPrincipalType"
substitutionGroup="urb:_ViaPrincipal"/>
    <complexType name="ViaPrincipalType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element ref="gml:LineString"/>
                    <element name="numLogradouro" type="integer"/>
                    <element name="tipoVia" type="string"/>
                    <element name="sobrepoeRodovia"
type="urb:RodoviaPropertyType">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">sobrepostaAViaPrincipal</appinfo>
                        </annotation>
                    </element>
                    <element name="pertenceALogradouro"
type="urb:LogradouroPropertyType">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">possuiViaPrincipal</appinfo>
                        </annotation>
                    </element>
                    <element name="compostaPorTrecho"
type="urb:TrechoPropertyType" minOccurs="0">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">compoeViaPrincipal</appinfo>
                        </annotation>
                    </element>
                    <element name="possuiCruzamentoViasPrincipais"
type="urb:CruzamentoViasPrincipaisPropertyType" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">pertenceAViaPrincipal</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="ViaPrincipalPropertyType">
        <sequence>
            <element ref="urb:ViaPrincipal" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->

    <element name="_CruzamentoViasPrincipais" type="gml:AbstractFeatureType"
abstract="true" substitutionGroup="urb:_CTMFeature"/>
    <element name="CruzamentoViasPrincipais"
type="urb:CruzamentoViasPrincipaisType"
substitutionGroup="urb:_CruzamentoViasPrincipais"/>
    <complexType name="CruzamentoViasPrincipaisType">

```

```

    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:Point"/>
          <element name="sobrepoeCruzamento"
type="urb:CruzamentoPropertyType" minOccurs="0">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">sobrepostoPorCruzamentoViasPrincipais</appinfo>
            </annotation>
          </element>
          <element name="pertenceAViaPrincipal"
type="urb:ViaPrincipalPropertyType" maxOccurs="unbounded">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">possuiCruzamentoViasPrincipais</appinfo>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="CruzamentoViasPrincipaisPropertyType">
    <sequence>
      <element ref="urb:CruzamentoViasPrincipais" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!--=====-->

  <element name="_Logradouro" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
  <element name="Logradouro" type="urb:LogradouroType"
substitutionGroup="urb:_Logradouro"/>
  <complexType name="LogradouroType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="numLogradouro" type="integer"/>
          <element name="tipoLogradouro" type="string"/>
          <element name="nomeLogradouro" type="string"/>
          <element name="possuiViaPrincipal"
type="urb:ViaPrincipalPropertyType" maxOccurs="unbounded">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">pertenceALogradouro</appinfo>
            </annotation>
          </element>
          <element name="possuiTrecho"
type="urb:TrechoPropertyType" maxOccurs="unbounded">
            <annotation>
              <appinfo source="urn:x-
gml:reverseProperty">pertenceALogradouro</appinfo>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

```

```

        </complexContent>
    </complexType>
    <complexType name="LogradouroPropertyType">
        <sequence>
            <element ref="urb:Logradouro" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
    <!--=====-->

    <element name="_Trecho" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
    <element name="Trecho" type="urb:TrechoType" substitutionGroup="urb:_Trecho"/>
    <complexType name="TrechoType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element ref="gml:LineString"/>
                    <element name="numLogradouro" type="integer"/>
                    <element name="tipoVia" type="string"/>
                    <element name="numSeqTrecho" type="integer"/>
                    <element
name="pertenceALogradouro"
type="urb:LogradouroPropertyType">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">possuiTrecho</appinfo>
                        </annotation>
                    </element>
                    <element name="compoViaPrincipal"
type="urb:ViaPrincipalPropertyType" minOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">compostaPorTrecho</appinfo>
                        </annotation>
                    </element>
                    <element name="possuiCruzamento"
type="urb:CruzamentoPropertyType" minOccurs="2">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">pertenceATrecho</appinfo>
                        </annotation>
                    </element>
                    <element name="emFrenteAQuadraCTM"
type="urb:QuadraCTMPropertyType" minOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">emFrenteATrecho</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="TrechoPropertyType">
        <sequence>
            <element ref="urb:Trecho" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>

```

```

<!--=====-->
    <element name="_Cruzamento" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
    <element name="Cruzamento" type="urb:CruzamentoType"
substitutionGroup="urb:_Cruzamento"/>
    <complexType name="CruzamentoType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element ref="gml:Point"/>
                    <element
name="sobrepostoPorCruzamentoViasPrincipais"
type="urb:CruzamentoViasPrincipaisPropertyType">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">sobrepoeCruzamento</appinfo>
                        </annotation>
                    </element>
                    <element name="pertenceATrecho"
type="urb:TrechoPropertyType" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">possuiCruzamento</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="CruzamentoPropertyType">
        <sequence>
            <element ref="urb:Cruzamento" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>

<!--=====-->
    <element name="_QuadraCTM" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
    <element name="QuadraCTM" type="urb:QuadraCTMType"
substitutionGroup="urb:_QuadraCTM"/>
    <complexType name="QuadraCTMType">
        <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element ref="gml:Polygon"/>
                    <element name="numQuadraCTM" type="integer"/>
                    <element name="contindaEmManchaUrbana"
type="urb:ManchaUrbanaPropertyType" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">contemQuadraCTM</appinfo>
                        </annotation>
                    </element>
                    <element name="emFrenteATrecho"
type="urb:TrechoPropertyType" minOccurs="0" maxOccurs="unbounded">
                        <annotation>
                            <appinfo source="urn:x-

```

```

gml:reverseProperty">emFrenteAQuadraCTM</appinfo>
    </annotation>
    </element>
    <element name="compostaPorLoteCTM"
type="urb:LoteCTMPropertyType" maxOccurs="unbounded">
    <annotation>
    <appinfo source="urn:x-
gml:reverseProperty">compoeQuadraCTM</appinfo>
    </annotation>
    </element>
    <element name="constituídaDeLoteCTMPoligono"
type="urb:LoteCTMPoligonoPropertyType" maxOccurs="unbounded">
    <annotation>
    <appinfo source="urn:x-
gml:reverseProperty">agregadoDeQuadraCTM</appinfo>
    </annotation>
    </element>
    </sequence>
    </extension>
    </complexContent>
</complexType>
<complexType name="QuadraCTMPropertyType">
    <sequence>
    <element ref="urb:QuadraCTM" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->

    <element name="_LoteCTM" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
    <element name="LoteCTM" type="urb:LoteCTMType"
substitutionGroup="urb:_LoteCTM"/>
    <complexType name="LoteCTMType">
    <complexContent>
    <extension base="gml:AbstractFeatureType">
    <sequence>
    <element name="numQuadraCTM" type="integer"/>
    <element name="numLoteCTM" type="integer"/>
    <element name="compoeQuadraCTM"
type="urb:LoteCTMPropertyType">
    <annotation>
    <appinfo source="urn:x-
gml:reverseProperty">compostaPorLoteCTM</appinfo>
    </annotation>
    </element>
    </sequence>
    </extension>
    </complexContent>
</complexType>
<complexType name="LoteCTMPropertyType">
    <sequence>
    <element ref="urb:LoteCTM" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!--=====-->

```

```

    <element name="_LoteCTMPoligono" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
    <element name="LoteCTMPoligono" type="urb:LoteCTMPoligonoType"
substitutionGroup="urb:_LoteCTMPoligono"/>
    <complexType name="LoteCTMPoligonoType">
        <complexContent>
            <extension base="urb:LoteCTMType">
                <sequence>
                    <element ref="gml:Polygon"/>
                    <element name="numQuadraCTM" type="integer"/>
                    <element name="numLoteCTM" type="integer"/>
                    <element name="agregadoDeQuadraCTM"
type="urb:QuadraCTMPropertyType">
                        <annotation>
                            <appinfo source="urn:x-
gml:reverseProperty">constituidaDeLoteCTMPoligono</appinfo>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="LoteCTMPoligonoPropertyType">
        <sequence>
            <element ref="urb:LoteCTMPoligono" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
<!--=====-->

    <element name="_LoteCTMFrente" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
    <element name="LoteCTMFrente" type="urb:LoteCTMFrenteType"
substitutionGroup="urb:_LoteCTMFrente"/>
    <complexType name="LoteCTMFrenteType">
        <complexContent>
            <extension base="urb:LoteCTMType">
                <sequence>
                    <element ref="gml:LineString"/>
                    <element name="numQuadraCTM" type="integer"/>
                    <element name="numLoteCTM" type="integer"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
<!--=====-->
    <element name="_LoteCTMPto" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="urb:_CTMFeature"/>
    <element name="LoteCTMPto" type="urb:LoteCTMPtoType"
substitutionGroup="urb:_LoteCTMPto"/>
    <complexType name="LoteCTMPtoType">
        <complexContent>
            <extension base="urb:LoteCTMType">
                <sequence>
                    <element ref="gml:Point"/>
                    <element name="numQuadraCTM" type="integer"/>
                    <element name="numLoteCTM" type="integer"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

```

```
        </complexContent>
    </complexType>
    <!--=====-->
</schema>
```

## Anexo 3 - Código Fonte

### Main.java

```
package fadesgml;
/**
 * FadesGML
 *
 * Ferramenta de Apoio a Determinação de Equivalências Semânticas entre
 * Esquemas GML Utilizando Ontologias OWL
 */
import javax.swing.UIManager;

public class Main {

    public Main() {

    }

    public static void main(String[] args) {

        // Configura o tema padrão
        try {
            UIManager.setLookAndFeel(
                //usa o tema padrão de cada plataforma (win, mac, nix)
                UIManager.getSystemLookAndFeelClassName() );
        }
        catch (Exception e) { }

        //cria a janela principal
        JFramePrincipal frmPrincipal = new JFramePrincipal();

        //abre a janela principal
        frmPrincipal.setVisible(true);

    }
}
```

### AttrSimItems.java

```
package fadesgml;
/**
 * FadesGML
 *
 * Ferramenta de Apoio a Determinação de Equivalências Semânticas entre
 * Esquemas GML Utilizando Ontologias OWL
 */
import javax.swing.UIManager;

public class Main {

    public Main() {
```



```

}

public static void main(String[] args) {

    // Configura o tema padrão
    try {
        UIManager.setLookAndFeel(
            //usa o tema padrão de cada plataforma (win, mac, nix)
            UIManager.getSystemLookAndFeelClassName() );
    }
    catch (Exception e) { }

    //cria a janela principal
    JFramePrincipal frmPrincipal = new JFramePrincipal();

    //abre a janela principal
    frmPrincipal.setVisible(true);

}
}

```

### CustomTreeCellRender.java

```

package fadesgml;

import com.hp.hpl.jena.ontology.ObjectProperty;
import com.hp.hpl.jena.ontology.OntClass;
import com.hp.hpl.jena.ontology.OntProperty;
import java.awt.Component;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeCellRenderer;

public class CustomTreeCellRender extends DefaultTreeCellRenderer{

    public CustomTreeCellRender() {
    }

    //<editor-fold defaultstate="collapsed" desc="getTreeCellRendererComponent">
    public Component getTreeCellRendererComponent(JTree tree, Object value, boolean
selected, boolean expanded, boolean leaf, int row, boolean hasFocus){

        JLabel label = (JLabel) super.getTreeCellRendererComponent(tree, value, selected,
expanded, leaf, row, hasFocus);
        DefaultMutableTreeNode node = (DefaultMutableTreeNode) value;
        if (node.getUserObject() instanceof String){
            String nodeLabel = node.getUserObject().toString();
            if ( nodeLabel.startsWith("___")){
                label.setIcon(new ImageIcon("images/rel.gif"));
                nodeLabel = nodeLabel.replace("___", "");
                label.setText(nodeLabel);
            }
            else if (nodeLabel.equalsIgnoreCase("Polygon")) //

```

```

nodeLabel.equalsIgnoreCase("string") ||
    nodeLabel.equalsIgnoreCase("LineString") ||
nodeLabel.equalsIgnoreCase("Point") ||
    nodeLabel.endsWith("____"){
    if (nodeLabel.endsWith("____")){
        nodeLabel = nodeLabel.replace("____", "");
        label.setText(nodeLabel);
    }
    label.setIcon(new ImageIcon("images/attr.gif"));
}
else if (!(nodeLabel.equalsIgnoreCase("Tipo") ||
nodeLabel.equalsIgnoreCase("minOccurs") ||
nodeLabel.equalsIgnoreCase("maxOccurs") || nodeLabel.equalsIgnoreCase("gml"))){
    if (!nodeLabel.startsWith("@@")){
        if (!nodeLabel.endsWith("--")){
            label.setIcon(new ImageIcon("images/class_elem.gif"));
        }
        else{
            nodeLabel = nodeLabel.replace("--", "");
            label.setText(nodeLabel);
        }
    }
}
if (nodeLabel.startsWith("@@")){
    nodeLabel = nodeLabel.replace("@@", "");
    label.setText(nodeLabel);
}
}
return label;
}
//</editor-fold>
}

```

## JFramePrincipal.java

```

/*
 * JFramePrincipal.java
 */

package fadesgml;

import com.ctc.wstx evt.WNamespace;
import com.hp.hpl.jena.rdf.model.Resource;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.io.File;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Enumeration;
import java.util.Iterator;
import java.util.List;
import java.util.NoSuchElementException;
import java.util.Vector;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumn;
import javax.swing.tree.DefaultMutableTreeNode;

```

```

import javax.swing.tree.DefaultTreeModel;
import org.jdom.Attribute;
import org.jdom.Document;
import org.jdom.Element;
import org.jdom.output.XMLOutputter;

/**
 *
 * @author Leonardo Roman da Rosa
 */
public class JFramePrincipal extends javax.swing.JFrame {

    WrapperOWL wrapperOWL = new WrapperOWL();
    WrapperGML2 wrapperGML = new WrapperGML2();
    JFrameValida validacao = new JFrameValida();
    Simil simil = new Simil();

    /** Creates new form JFramePrincipal */
    public JFramePrincipal() {

        initComponents();

        //centraliza a janela na tela
        Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
        int x = (screen.width - this.getWidth()) / 2;
        int y = (screen.height - this.getHeight()) / 2;
        this.setLocation(x, y);

        jLabelExpandirOWL.setVisible(false);
        jLabelExpandirGML.setVisible(false);

        jTreeGML.setCellRenderer(new CustomTreeCellRender());
        jTreeOWL.setCellRenderer(new OWLTreeCellRender());

        //monta o cabeçalho da tabela de sinônimos
        String[] cabecalhoTabelaSinonimos = {
            "Sinônimo",
            "Classe",
            "Idioma"
        };
        DefaultTableModel modelo = new DefaultTableModel(null, cabecalhoTabelaSinonimos) {
            //desabilita edição de células da tabela
            public boolean isCellEditable(int rowIndex, int mColIndex) {
                return false;
            }
        };
        jTableSinonimos.setModel(modelo);
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc=" Código Gerado ">
    private void initComponents() {

```

```

jButtonAbrirOWL = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
jTreeOWL = new javax.swing.JTree(new javax.swing.tree.DefaultTreeModel(null));
jLabelArvore1 = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
jTreeGML = new javax.swing.JTree(new javax.swing.tree.DefaultTreeModel(null));
jLabelArvore2 = new javax.swing.JLabel();
jButtonAbrirGML = new javax.swing.JButton();
jLabelExpandirOWL = new javax.swing.JLabel();
jLabelExpandirGML = new javax.swing.JLabel();
jScrollPane3 = new javax.swing.JScrollPane();
jTableSinonimos = new javax.swing.JTable();
jLabelSinonimos = new javax.swing.JLabel();
jButtonSimil = new javax.swing.JButton();
jPanelThreshold = new javax.swing.JPanel();
jLabelWne = new javax.swing.JLabel();
jLabelWna = new javax.swing.JLabel();
jLabelWta = new javax.swing.JLabel();
jLabelWco = new javax.swing.JLabel();
jLabelWca = new javax.swing.JLabel();
jLabelWnr = new javax.swing.JLabel();
jTextFieldWne = new javax.swing.JTextField();
jTextFieldWna = new javax.swing.JTextField();
jTextFieldWta = new javax.swing.JTextField();
jTextFieldWco = new javax.swing.JTextField();
jTextFieldWca = new javax.swing.JTextField();
jTextFieldWnr = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("FadesGML");
jButtonAbrirOWL.setText("Abrir OWL");
jButtonAbrirOWL.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonAbrirOWLActionPerformed(evt);
    }
});

jScrollPane1.setViewportView(jTreeOWL);

jLabelArvore1.setText("\u00c1rvore OWL:");

jScrollPane2.setViewportView(jTreeGML);

jLabelArvore2.setText("\u00c1rvore GML:");

jButtonAbrirGML.setText("Abrir GML");
jButtonAbrirGML.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonAbrirGMLActionPerformed(evt);
    }
});

jLabelExpandirOWL.setForeground(new java.awt.Color(0, 0, 255));
jLabelExpandirOWL.setText("[Expandir]");
jLabelExpandirOWL.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabelExpandirOWLMouseClicked(evt);
    }
});

```

```

jLabelExpandirGML.setForeground(new java.awt.Color(0, 0, 255));
jLabelExpandirGML.setText("[Expandir]");
jLabelExpandirGML.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabelExpandirGMLMouseClicked(evt);
    }
});

jTableSinonimos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane3.setViewportViewView(jTableSinonimos);

jLabelSinonimos.setText("Tabela de Sin\u00f4nimos:");

jButtonSimil.setText("Similaridades ...");
jButtonSimil.setEnabled(false);
jButtonSimil.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonSimilActionPerformed(evt);
    }
});

jPanelThreshold.setBorder(javax.swing.BorderFactory.createTitledBorder(" Pesos "));
jLabelWne.setText("Wne");

jLabelWna.setText("Wna");

jLabelWta.setText("Wta");

jLabelWco.setText("Wco");

jLabelWca.setText("Wca");

jLabelWnr.setText("Wnr");

jTextFieldWne.setText(".8");
jTextFieldWne.setToolTipText("Nome de Elemento (Wne >= 0 e <= 1)");

jTextFieldWna.setText(".6");
jTextFieldWna.setToolTipText("Nome de Atributo (Wna + Wta >= 0 e <= 1)");

jTextFieldWta.setText(".4");
jTextFieldWta.setToolTipText("Tipo de Atributo (Wna + Wta >= 0 e <= 1)");

jTextFieldWco.setText(".4");
jTextFieldWco.setToolTipText("Conceito Relacionado (Wco + Wca + Wnr >= 0 e <= 1)");

jTextFieldWca.setText(".2");
jTextFieldWca.setToolTipText("Cardinalidade (Wco + Wca + Wnr >= 0 e <= 1)");

jTextFieldWnr.setText(".4");

```

```

jTextFieldWnr.setToolTipText("Nome de Relacionamento (Wco + Wca + Wnr >= 0 e <=
1)");

org.jdesktop.layout.GroupLayout jPanelThresholdLayout = new
org.jdesktop.layout.GroupLayout(jPanelThreshold);
jPanelThreshold.setLayout(jPanelThresholdLayout);
jPanelThresholdLayout.setHorizontalGroup(

jPanelThresholdLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(jPanelThresholdLayout.createSequentialGroup())
.addContainerGap()

.add(jPanelThresholdLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(jLabelWne)
.add(jLabelWna)
.add(jLabelWta))
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

.add(jPanelThresholdLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING,
false)
.add(jTextFieldWta, 0, 0, Short.MAX_VALUE)
.add(jTextFieldWna, 0, 0, Short.MAX_VALUE)
.add(jTextFieldWne, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 43,
Short.MAX_VALUE))
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 57,
Short.MAX_VALUE)

.add(jPanelThresholdLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING,
false)
.add(jPanelThresholdLayout.createSequentialGroup()
.add(jLabelWnr)
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(jTextFieldWnr, 0, 0, Short.MAX_VALUE))
.add(org.jdesktop.layout.GroupLayout.TRAILING,
jPanelThresholdLayout.createSequentialGroup()
.add(jLabelWca)
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(jTextFieldWca, 0, 0, Short.MAX_VALUE))
.add(jPanelThresholdLayout.createSequentialGroup()
.add(jLabelWco)
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
.add(jTextFieldWco, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 43,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
.addContainerGap(62, Short.MAX_VALUE)
);
jPanelThresholdLayout.setVerticalGroup(

jPanelThresholdLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
.add(jPanelThresholdLayout.createSequentialGroup()
.add(11, 11, 11)

.add(jPanelThresholdLayout.createSequentialGroup()
.add(jLabelWne)
.add(jLabelWco)
.add(jTextFieldWne, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
.add(jTextFieldWco, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))

```

```

        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)

        .add(jPanelThresholdLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
            .add(jLabelWna)
            .add(jLabelWca)
            .add(jTextFieldWna, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .add(jTextFieldWca, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .add(jPanelThresholdLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
            .add(jLabelWta)
            .add(jLabelWnr)
            .add(jTextFieldWta, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .add(jTextFieldWnr, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
            .addContainerGap()
    );
    jPanelThreshold.setAccessibleContext().setAccessibleName("");

    org.jdesktop.layout.GroupLayout layout = new
org.jdesktop.layout.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(layout.createSequentialGroup()
                .addContainerGap()
                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                    .add(layout.createSequentialGroup()
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                            .add(layout.createSequentialGroup()
                                .add(jButtonAbrirOWL)
                                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED))
                            .add(org.jdesktop.layout.GroupLayout.TRAILING,
layout.createSequentialGroup()
                                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)
                                    .add(jScrollPane1, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
266, Short.MAX_VALUE)
                                    .add(org.jdesktop.layout.GroupLayout.LEADING, jScrollPane3,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 266, Short.MAX_VALUE)
                                    .add(layout.createSequentialGroup()
                                        .add(10, 10, 10)
                                        .add(jLabelArvore1)
                                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 143,
Short.MAX_VALUE)
                                        .add(jLabelExpandirOWL)))
                                    .add(16, 16, 16)))
                                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)
                                    .add(org.jdesktop.layout.GroupLayout.LEADING,
layout.createSequentialGroup()
                                        .add(10, 10, 10)
                                        .add(jLabelArvore2)

```

```

        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 158,
Short.MAX_VALUE)
        .add(jLabelExpandirGML))
        .add(org.jdesktop.layout.GroupLayout.LEADING, jButtonAbrirGML)
        .add(layout.createSequentialGroup())
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
        .add(jPanelThreshold, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .add(jScrollPane2, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 278,
Short.MAX_VALUE)))
        .add(layout.createSequentialGroup())
        .add(jLabelSinonimos)
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 348,
Short.MAX_VALUE)
        .add(jButtonSimil)))
        .add(ContainerGap())
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(layout.createSequentialGroup())
        .add(ContainerGap())
        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(layout.createSequentialGroup())
        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
        .add(jButtonAbrirOWL)
        .add(jButtonAbrirGML))
        .add(14, 14, 14)
        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
        .add(jLabelArvore2)
        .add(jLabelExpandirOWL)
        .add(jLabelExpandirGML)))
        .add(layout.createSequentialGroup())
        .add(37, 37, 37)
        .add(jLabelArvore1)))
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
        .add(jScrollPane2, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 244,
Short.MAX_VALUE)
        .add(jScrollPane1, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 244,
Short.MAX_VALUE))
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
        .add(jLabelSinonimos)
        .add(jButtonSimil))
        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING, false)
        .add(jScrollPane3, 0, 0, Short.MAX_VALUE)
        .add(jPanelThreshold, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .add(ContainerGap())
    );
    pack();
} // </editor-fold>

private void jButtonSimilActionPerformed(java.awt.event.ActionEvent evt) {

    this.determinaGrausSimilaridade();
}

private void jLabelExpandirGMLMouseClicked(java.awt.event.MouseEvent evt) {

```



```

if (jLabelExpandirGML.getText() == "[Expandir]") {
    //expande a árvore
    for (int r = 0; r < jTreeGML.getRowCount(); r++)
        jTreeGML.expandRow(r);
    jLabelExpandirGML.setText("[Contrair]");
}
else {
    //contraí a árvore
    for (int r = jTreeGML.getRowCount(); r > 0; r--)
        jTreeGML.collapseRow(r);
    jLabelExpandirGML.setText("[Expandir]");
}
}

private void jLabelExpandirOWLMouseClicked(java.awt.event.MouseEvent evt) {

    if (jLabelExpandirOWL.getText() == "[Expandir]") {
        //expande a árvore
        for (int r = 0; r < jTreeOWL.getRowCount(); r++)
            jTreeOWL.expandRow(r);
        jLabelExpandirOWL.setText("[Contrair]");
    }
    else {
        //contraí a árvore
        for (int r = jTreeOWL.getRowCount(); r > 0; r--)
            jTreeOWL.collapseRow(r);
        jLabelExpandirOWL.setText("[Expandir]");
    }
}

private void jButtonAbrirGMLActionPerformed(java.awt.event.ActionEvent evt) {

    final JFileChooser abrirArquivo = new JFileChooser();

    //Abre a a caixa de seleção de arquivos
    int vlrRetorno = abrirArquivo.showOpenDialog(this);

    if (vlrRetorno == JFileChooser.APPROVE_OPTION) {
        File arquivo = abrirArquivo.getSelectedFile();

        try {
            //Preenche a árvore com o retorno do wrapper
            //this.geraArvoreGML( wrapperGML.loadGMLFile(arquivo) );
            Document doc = wrapperGML.loadGMLFile(arquivo);
            this.geraArvoreGML( criaArvore(doc));
            clearTemporarySymbols(doc);
            saveGMLToFile(doc);
            jLabelExpandirGML.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

jLabelExpandirGML.setText("[Expandir]");

if ( (jTreeOWL.getModel()).getRoot() != null ) {
    if ( !jButtonSimil.isEnabled() ) jButtonSimil.setEnabled(true);
}
}

private void clearTemporarySymbols(Document document){
    Iterator it = document.getDescendants();

    while(it.hasNext()){
        Element elem = (Element) it.next();
        elem.setName(elem.getName().replace("___", ""));
        elem.setName(elem.getName().replace("---", ""));
    }
}

private void jButtonAbrirOWLActionPerformed(java.awt.event.ActionEvent evt) {

    final JFileChooser abrirArquivo = new JFileChooser();

    //Abre a a caixa de seleção de arquivos
    int vlrRetorno = abrirArquivo.showOpenDialog(this);

    if (vlrRetorno == JFileChooser.APPROVE_OPTION) {
        File arquivo = abrirArquivo.getSelectedFile();

        try {
            //Preenche a tabela de sinonimos
            this.preencheTabelaSinonimos(
                wrapperOWL.loadSinonimos(arquivo.toURI().toString()) );

            //Preenche a árvore com o retorno do wrapper
            DefaultMutableTreeNode root = wrapperOWL.loadOWLFile(arquivo);
            this.geraArvoreOWL( root );
            Document doc = GenerateTreeDocument(root);
            clearTemporarySymbols(doc);
            saveOWLToFile(doc);
            jLabelExpandirOWL.setVisible(true);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    jLabelExpandirOWL.setText("[Expandir]");

    if ( (jTreeGML.getModel()).getRoot() != null ) {
        if ( !jButtonSimil.isEnabled() ) jButtonSimil.setEnabled(true);
    }
}

/**
 * @param args the command line arguments
 */

```

```

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new JFramePrincipal().setVisible(true);
        }
    });
}

// Declaração de variáveis - não modifique
private javax.swing.JButton jButtonAbrirGML;
private javax.swing.JButton jButtonAbrirOWL;
private javax.swing.JButton jButtonSimil;
private javax.swing.JLabel jLabelArvore1;
private javax.swing.JLabel jLabelArvore2;
private javax.swing.JLabel jLabelExpandirGML;
private javax.swing.JLabel jLabelExpandirOWL;
private javax.swing.JLabel jLabelSinonimos;
private javax.swing.JLabel jLabelWca;
private javax.swing.JLabel jLabelWco;
private javax.swing.JLabel jLabelWna;
private javax.swing.JLabel jLabelWne;
private javax.swing.JLabel jLabelWnr;
private javax.swing.JLabel jLabelWta;
private javax.swing.JPanel jPanelThreshold;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTable jTableSinonimos;
private javax.swing.JTextField jTextFieldWca;
private javax.swing.JTextField jTextFieldWco;
private javax.swing.JTextField jTextFieldWna;
private javax.swing.JTextField jTextFieldWne;
private javax.swing.JTextField jTextFieldWnr;
private javax.swing.JTextField jTextFieldWta;
private javax.swing.JTree jTreeGML;
private javax.swing.JTree jTreeOWL;
// Fim da declaração de variáveis

//Efetua o preenchimento da tabela de sinônimos
private void preencheTabelaSinonimos(List<String[]> lista) {
    DefaultTableModel dtm = (DefaultTableModel) jTableSinonimos.getModel();
    for(int i = dtm.getRowCount() - 1; i >= 0 ; i--){
        dtm.removeRow(i);
    }
    for (int i = 0; i < lista.size(); i++){
        dtm.addRow( lista.get(i) );
    }
}

//Efetua o preenchimento da árvore JTree da ontologia
private void geraArvoreOWL(DefaultMutableTreeNode arvore) {

    //atribui o elemento raiz da árvore do parâmetro como raiz da JTree
    ( (DefaultTableModel)jTreeOWL.getModel() ).setRoot(arvore);
}

```

```

//Efetua o preenchimento da árvore JTree do modelo GML
private void geraArvoreGML(DefaultMutableTreeNode arvore) {

    //atribui o elemento raiz da árvore do parâmetro como raiz da JTree
    ( (DefaultTreeModel)jTreeGML.getModel() ).setRoot(arvore.getChildAt(0));

}

private DefaultMutableTreeNode criaArvore(Document documento) {

    // Cria o primeiro nodo da árvore com o elemento raiz do documento XML
    DefaultMutableTreeNode arvore = new
DefaultMutableTreeNode(documento.getRootElement().getName());

    // Adiciona os elementos do documento à arvore criada
    this.adicionaElementos(arvore, documento.getRootElement());

    return arvore;
}

private void adicionaElementos(DefaultMutableTreeNode arvore, Element elemento) {

    // Novo nodo para representar o elemento passado no parâmetro
    DefaultMutableTreeNode nodoAtual =
        new DefaultMutableTreeNode(elemento.getName());

    // Pega o nome do elemento
    String nome = elemento.getTextNormalize();

    // Adiciona o nodo com o elemento dentro do nodo raiz
    if( (nome != null) && (!nome.equals("")) )
        nodoAtual.add(new DefaultMutableTreeNode(nome));

    // Adiciona os atributos do elemento atual (como novos nodos)
    this.adicionaAtributos(nodoAtual, elemento);

    // Verifica elementos filho
    Iterator ponteiro = elemento.getChildren().iterator();

    // Adiciona os elementos filho no nodo do elemento atual
    while(ponteiro.hasNext())
        this.adicionaElementos( nodoAtual, (Element)ponteiro.next() );

    // Adiciona o nodo do elemento atual à árvore principal
    arvore.add(nodoAtual);
}

private void adicionaAtributos(DefaultMutableTreeNode nodoAtual, Element elemento) {
    //percorre o elemento em busca de atributos
    Iterator ponteiro = elemento.getAttributes().iterator();

    //cria um nodo para cada atributo do elemento
    while ( ponteiro.hasNext() ){

        Attribute atributo = (Attribute) ponteiro.next();
        DefaultMutableTreeNode nodoAtributo = new

```

```

DefaultMutableTreeNode(atributo.getName());
    nodoAtributo.add( new DefaultMutableTreeNode("@@" + atributo.getValue()));
    nodoAtual.add(nodoAtributo);
}
}

private void saveGMLToFile(Document documentoSaida){
    // Salva o xml lido em outro arquivo
    try {
        XMLOutputter saida = new XMLOutputter();

        //salva o xml criado em arquivo
        FileWriter salvar = new FileWriter("arvoreGML.xml");

        // saida.output(documento, salvar);
        saida.output(documentoSaida, salvar);
        salvar.close();

    } catch (java.io.IOException e) {
        e.printStackTrace();
    }
}

private void saveOWLToFile(Document documentoSaida){
    // Salva o xml lido em outro arquivo
    try {
        XMLOutputter saida = new XMLOutputter();

        //salva o xml criado em arquivo
        FileWriter salvar = new FileWriter("arvoreOWL.xml");

        // saida.output(documento, salvar);
        saida.output(documentoSaida, salvar);
        salvar.close();

    } catch (java.io.IOException e) {
        e.printStackTrace();
    }
}

private Document GenerateTreeDocument(DefaultMutableTreeNode rootNode){

    Element rootElement = new Element(rootNode.toString());
    addElements(rootElement, rootNode);
    Document outputDocument = new Document(rootElement);

    return outputDocument;
}

private void addElements(Element element, DefaultMutableTreeNode node){
    if (node.getChildCount() > 0){
        Enumeration e = node.children();
        while(e.hasMoreElements()){
            DefaultMutableTreeNode childNode = (DefaultMutableTreeNode) e.nextElement();

            String name = childNode.getUserObject().toString();

```

```

        if (childNodes.getUserObject() instanceof Resource){
            Resource res = (Resource) childNode.getUserObject();
            name = res.getLocalName();
        }

        if (childNodes.getUserObject() instanceof String){
            try {
                DefaultMutableTreeNode cardNode = (DefaultMutableTreeNode)
childNodes.getFirstChild();
                if (cardNode.getUserObject().toString().equals("Cardinalidade")){
                    DefaultMutableTreeNode valueNode = (DefaultMutableTreeNode)
childNodes.getFirstChild();
                    Element childElement = new Element(name);
                    element.addContent(childElement);

childNodes.setAttribute("Cardinalidade",valueNode.getUserObject().toString());
                    continue;
                }
            }
            catch(NoSuchElementException ex) {
            }
        }

        if (name == "Tipo"){
            DefaultMutableTreeNode value = (DefaultMutableTreeNode)
childNodes.getFirstChild();
            element.setAttribute("Tipo",value.getUserObject().toString());
        }
        else{
            Element childElement = new Element(name);
            element.addContent(childElement);
            addElements(childElement,childNodes);
        }

//        if (node.getUserObject() instanceof OntProperty){
//            OntProperty prop = (OntProperty) node.getUserObject();
//            if (!(prop.isObjectProperty())){
//                element.setAttribute("Tipo",name);
//                continue;
//            }
//            else{
//                //adicionar atributos minOccurs e maxOccurs e dar um continue;
//            }
//        }

    }
}

private void determinaGrausSimilaridade() {
    float wne = 0;
    float wna = 0;
    float wta = 0;
    float wco = 0;
    float wca = 0;
    float wnr = 0;
}

```

```

try {
    wne = Float.parseFloat( jTextFieldWne.getText() );

    wna = Float.parseFloat( jTextFieldWna.getText() );
    wta = Float.parseFloat( jTextFieldWta.getText() );

    wco = Float.parseFloat( jTextFieldWco.getText() );
    wca = Float.parseFloat( jTextFieldWca.getText() );
    wnr = Float.parseFloat( jTextFieldWnr.getText() );
} catch (Exception e) {
    JOptionPane.showMessageDialog(this,
        "Houve erros na digitação dos pesos.\nVerifique os números novamente.");
    return;
}

if ( ( wne > 1.0) || ( wne <= 0.0) ){

    JOptionPane.showMessageDialog(this,
        "O peso para o nome dos elementos (Wne) deve ficar entre 0 e 1 (inclusive).\n" +
        "Verifique esse peso novamente.");
    return;
}

if ( wna + wta != 1.0){

    JOptionPane.showMessageDialog(this,
        "A soma dos pesos de atributos (Wna e Wta) deve ser igual a 1.\n" +
        "Verifique esses pesos novamente.");
    return;
}

if ( wco + wca + wnr != 1.0){

    JOptionPane.showMessageDialog(this,
        "A soma dos pesos de relacionamentos (Wco, Wca e Wnr) deve ser igual a 1.\n" +
        "Verifique esses pesos novamente.");
    return;
}

//atribui os valores dos pesos
simil.ELEM_NAMESIM_WEIGHT = wne;
simil.ATTR_NAMESIM_WEIGHT = wna;
simil.TYPESIM_WEIGHT = wta;
simil.CONCSIM_WEIGHT = wco;
simil.CARDSIM_WEIGHT = wca;
simil.REL_NAMESIM_WEIGHT = wnr;

DefaultTreeModel gmlTree = (DefaultTreeModel) jTreeGML.getModel();
DefaultTreeModel owlTree = (DefaultTreeModel) jTreeOWL.getModel();
int qtdElemGml = gmlTree.getChildCount(gmlTree.getRoot());
int qtdElemOwl = owlTree.getChildCount(owlTree.getRoot());
List<String []> equivalencias = new ArrayList();
DefaultMutableTreeNode gmlNode = null;
DefaultMutableTreeNode owlNode = null;
String sinonimo = "";
String classeOwl = "";

for (int i = 0; i < qtdElemGml; i++) {
    //busca sinonimos iguais
    gmlNode = (DefaultMutableTreeNode)gmlTree.getChild(gmlTree.getRoot(), i);
}

```

```

sinonimo = this.buscaSinonimoIgual( gmlNode.toString() );
if ( sinonimo == "" )
    //não encontrou, busca sinonimos equivalentes
    sinonimo = this.buscaSinonimoEquiv( gmlNode.toString() );
if ( sinonimo == "" )
    //não encontrou nenhum sinonimo, verifica o próximo elemento
    continue;

//encontrou um sinônimo
//pega o nome da classe correspondente na lista de sinônimos
for (int j = 0; j < jTableSinonimos.getRowCount(); j++) {
    if ( jTableSinonimos.getValueAt(j, 0).toString() == sinonimo ){
        classeOwl = jTableSinonimos.getValueAt(j, 1).toString();
        break;
    }
}

//procura a classe na árvore da ontologia
for(int j = 0; j < qtdElemOwl; j++) {
    String classeAtual = owlTree.getChild(owlTree.getRoot(), j).toString().substring(
        owlTree.getChild(owlTree.getRoot(), j).toString().indexOf("#") + 1
    );
    if ( classeOwl.equals(classeAtual) ){
        owlNode = (DefaultMutableTreeNode)owlTree.getChild(owlTree.getRoot(), j);
        break;
    }
}

String[] elem = new String[3];
elem[0] = owlNode.toString().substring(owlNode.toString().indexOf("#") + 1);
elem[1] = gmlNode.toString();
elem[2] = String.valueOf(simFinal(owlNode, gmlNode));

equivalencias.add(elem);

} //fim do laço FOR principal

validacao.preencheLista(equivalencias);
validacao.setVisible(true);
}

private String buscaSinonimoIgual(String elemNEi) {
    String sinonimo = "";

    for (int i = 0; i < jTableSinonimos.getRowCount(); i++) {
        if ( elemNEi == jTableSinonimos.getValueAt(i, 0).toString() ) {
            sinonimo = jTableSinonimos.getValueAt(i, 0).toString();
            break;
        }
    }
    return sinonimo;
}

private String buscaSinonimoEquiv(String elemNEi) {
    String sinonimoAtual = "";
    String sinonimo = "";
    float equivalencia = 0f;

```



```

for (int i = 0; i < jTableSinonimos.getRowCount(); i++) {

    sinonimoAtual = jTableSinonimos.getValueAt(i, 0).toString();
    String lang = jTableSinonimos.getValueAt(i, 2).toString();
    if ( simil.simName(elemNEi, sinonimoAtual, lang) > equivalencia ) {
        equivalencia = simil.simName(elemNEi, sinonimoAtual, lang);
        sinonimo = sinonimoAtual;
    }
}

if ( equivalencia >= simil.ELEM_NAMESIM_WEIGHT )
    return sinonimo;
else return "";
}

private float simFinal(DefaultMutableTreeNode owlNode, DefaultMutableTreeNode
gmlNode){
    List<String[]> atrOwl = new ArrayList();
    List<String[]> relOwl = new ArrayList();
    List<String[]> atrGml = new ArrayList();
    List<String[]> relGml = new ArrayList();
    List<AttrSimItem> attrSimItems = new ArrayList();
    List<RelSimItem> relSimItems = new ArrayList();

    //separa os atributos da OWL
    for (int i = 0; i < owlNode.getChildCount(); i++){
        //verifica se não é um relacionamento
        if ( !owlNode.getChildAt(i).getChildAt(0).getChildAt(0).isLeaf() )
            continue;

        String nome = owlNode.getChildAt(i).toString().replace("---", "");
        if (nome.indexOf("#") > -1) nome = nome.substring(nome.indexOf("#") + 1);
        String tipo = owlNode.getChildAt(i).getChildAt(0).getChildAt(0).toString();

        String[] elem = new String[2];
        elem[0] = nome;
        elem[1] = tipo;
        atrOwl.add(elem);
    }

    //separa os relacionamentos da OWL
    for (int i = 0; i < owlNode.getChildCount(); i++){
        //verifica se não é um atributo
        if ( owlNode.getChildAt(i).getChildAt(0).getChildAt(0).isLeaf() )
            continue;

        String nome = owlNode.getChildAt(i).toString();
        if (nome.indexOf("#") > -1) nome = nome.substring(nome.indexOf("#") + 1);
        String card = owlNode.getChildAt(i).getChildAt(0).getChildAt(0).getChildAt(0).toString();

        String[] elem = new String[2];
        elem[0] = nome;
        elem[1] = card;
        relOwl.add(elem);
    }

    //separa os atributos da GML
    for (int i = 0; i < gmlNode.getChildCount(); i++){
        //verifica se não é um relacionamento

```

```

if ( (gmlNode.getChildAt(i).toString() == "Tipo" ) ||
    !gmlNode.getChildAt(i).getChildAt(0).getChildAt(0).isLeaf() )
    continue;

String nome = gmlNode.getChildAt(i).toString().replace("___", "");
String tipo = gmlNode.getChildAt(i).getChildAt(0).getChildAt(0).toString().replace("@@", "");

String[] elem = new String[2];
elem[0] = nome;
elem[1] = tipo;
atrGml.add(elem);
}

//separa os relacionamentos da GML
for (int i = 0; i < gmlNode.getChildCount(); i++){
    //verifica se não é um atributo
    if ( (gmlNode.getChildAt(i).toString() == "Tipo" ) ||
        gmlNode.getChildAt(i).getChildAt(0).getChildAt(0).isLeaf() )
        continue;

    String nome = gmlNode.getChildAt(i).toString().replace("___", "");
    String card = gmlNode.getChildAt(i).getChildAt(0).getChildAt(1).getChildAt(0).toString();
    if ( card.equals("@@1") ) card = "maxOccurs = 1";
    else card = "maxOccurs > 1";

    String[] elem = new String[2];
    elem[0] = nome;
    elem[1] = card;
    relGml.add(elem);
}

// monta a lista de atributos
int maxElem = atrOwl.size();
if (atrGml.size() < maxElem) maxElem = atrGml.size();
for (int i = 0; i < maxElem; i++){
    attrSimlItems.add(
        new AttrSimlItem(atrOwl.get(i)[0], atrGml.get(i)[0],
            atrOwl.get(i)[1], atrGml.get(i)[1])
    );
}

// monta a lista de relacionamentos
maxElem = relOwl.size();
if (relGml.size() < maxElem) maxElem = relGml.size();
for (int i = 0; i < maxElem; i++){
    float simNameValue = simil.simName(relOwl.get(i)[0], relGml.get(i)[0], "");
    float simCardValue = simil.simCard(relOwl.get(i)[1], relGml.get(i)[1]);
    relSimlItems.add( new RelSimlItem(simNameValue, 0f, simCardValue) );
}

int maxNodeCount = atrOwl.size() + relOwl.size();
if (maxNodeCount < atrGml.size() + relGml.size())
    maxNodeCount = atrGml.size() + relGml.size();

return simil.simTuple(attrSimlItems, relSimlItems, maxNodeCount);
}
}

```

## JFrameValida.java

```
/*
 * JFrameValida.java
 *
 * Created on 22 de Fevereiro de 2007, 02:25
 */

package fadesgml;

import java.awt.Dimension;
import java.awt.Toolkit;
import java.util.List;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author Leonardo
 */
public class JFrameValida extends javax.swing.JFrame {

    /** Creates new form JFrameValida */
    public JFrameValida() {

        initComponents();

        //centraliza a janela na tela
        Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
        int x = (screen.width - this.getWidth()) / 2;
        int y = (screen.height - this.getHeight()) / 2;
        this.setLocation(x, y);

        //monta o cabeçalho da tabela de sinônimos
        String[] cabecalho = {
            "Conceito OWL",
            "Elemento GML",
            "Similaridade",
            "Validar"
        };
        DefaultTableModel modelo = new DefaultTableModel(null, cabecalho);
        jTableValidar.setModel(modelo);
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc=" Código Gerado ">
    private void initComponents() {
        jScrollPane1 = new javax.swing.JScrollPane();
        jTableValidar = new javax.swing.JTable();
        jButtonConcluir = new javax.swing.JButton();
        jLabelQtdSim = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Valida\u00e7\u00e3o de Equival\u00eancias");
    }
}
```

```

jTableValidar.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Título 1", "Título 2", "Título 3", "Título 4"
    }
));
jScrollPane1.setViewportView(jTableValidar);

jButtonConcluir.setText("Concluir");
jButtonConcluir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonConcluirActionPerformed(evt);
    }
});

jLabelQtdSim.setText("Similaridades Encontradas:");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 453, Short.MAX_VALUE)
                .addGroup(layout.createSequentialGroup()
                    .add(jLabelQtdSim)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jButtonConcluir))
                .addGap())
            .addContainerGap());
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 200, Short.MAX_VALUE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jButtonConcluir)
                    .addComponent(jLabelQtdSim))
                .addGap())
            .addContainerGap());
pack();
} // </editor-fold>

private void jButtonConcluirActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
}

```

```

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new JFrameValida().setVisible(true);
        }
    });
}

// Declaração de variáveis - não modifique
private javax.swing.JButton jButtonConcluir;
private javax.swing.JLabel jLabelQtdSim;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTableValidar;
// Fim da declaração de variáveis

public void preencheLista(List<String[]> lista) {
    DefaultTableModel dtm = (DefaultTableModel) jTableValidar.getModel();

    for(int i = dtm.getRowCount() - 1; i >= 0 ; i--){
        dtm.removeRow(i);
    }

    for(int i = 0; i < lista.size(); i++) {
        dtm.addRow(lista.get(i));
    }

    jLabelQtdSim.setText("Similaridades Encontradas: " + String.valueOf(lista.size()));
}
}
}

```

### OWLTreeCellRender.java

```

package fadesgml;

import com.hp.hpl.jena.ontology.ObjectProperty;
import com.hp.hpl.jena.ontology.OntClass;
import com.hp.hpl.jena.ontology.OntProperty;
import java.awt.Component;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeCellRenderer;

public class OWLTreeCellRender extends DefaultTreeCellRenderer{

    public OWLTreeCellRender() {

```

```

}

//<editor-fold defaultstate="collapsed" desc="getTreeCellRendererComponent">
public Component getTreeCellRendererComponent(JTree tree, Object value, boolean
selected, boolean expanded, boolean leaf, int row, boolean hasFocus){

    JLabel label = (JLabel) super.getTreeCellRendererComponent(tree, value, selected,
expanded, leaf, row, hasFocus);
    Object object = ((DefaultMutableTreeNode) value).getUserObject();

    if (object instanceof String){
        String nodeLabel = (String) object;
        if (nodeLabel.startsWith("___")){
            nodeLabel = nodeLabel.replace("___", "");
            label.setText(nodeLabel);
            label.setIcon(new ImageIcon("images/class_elem.gif"));
        }
        else if (nodeLabel.endsWith("---")){
            label.setIcon(new ImageIcon("images/attr.gif"));
            nodeLabel = nodeLabel.replace("---", "");
            label.setText(nodeLabel);
        }
    }
    else{
        if (object instanceof OntClass){
            OntClass ontClass = (OntClass) object;
            label.setText(ontClass.getLocalName());
            label.setIcon(new ImageIcon("images/class_elem.gif"));
        }
        else if (object instanceof OntProperty){
            OntProperty prop = (OntProperty) object;
            if (prop.isObjectProperty()){
                label.setIcon(new ImageIcon("images/rel.gif"));
            }
            else{
                label.setIcon(new ImageIcon("images/attr.gif"));
            }
            label.setText(prop.getLocalName());
        }
    }
    return label;
}
//</editor-fold>
}

```

## RelSimItem.java

```

package fadesgml;

public class RelSimItem {

    private float simNameValue;
    private float simConcValue;
    private float simCardValue;

    public RelSimItem() {
    }
}

```

```
public RelSimItem(float simNameValue, float simConcValue, float simCardValue) {
    setSimNameValue(simNameValue);
    setSimConcValue(simConcValue);
    setSimCardValue(simCardValue);
}

public float getSimNameValue() {
    return simNameValue;
}

public void setSimNameValue(float simNameValue) {
    this.simNameValue = simNameValue;
}

public float getSimConcValue() {
    return simConcValue;
}

public void setSimConcValue(float simConcValue) {
    this.simConcValue = simConcValue;
}

public float getSimCardValue() {
    return simCardValue;
}

public void setSimCardValue(float simCardValue) {
    this.simCardValue = simCardValue;
}
}
```

### SimCardItem.java

```
package fadesgml;

public class SimCardItem {

    private String owlCard;
    private String gmlCard;
    private float similarity;

    public SimCardItem() {
    }

    public SimCardItem(String owlCard, String gmlCard, float similarity) {
        setOwlCard(owlCard);
        setGmlCard(gmlCard);
        setSimilarity(similarity);
    }

    public String getOwlCard() {
        return owlCard;
    }

    public void setOwlCard(String owlCard) {
```

```

        this.owlCard = owlCard;
    }

    public String getGmlCard() {
        return gmlCard;
    }

    public void setGmlCard(String gmlCard) {
        this.gmlCard = gmlCard;
    }

    public float getSimilarity() {
        return similarity;
    }

    public void setSimilarity(float similarity) {
        this.similarity = similarity;
    }
}

```

### SimTypeltem.java

```

package fadesgml;

public class SimTypeltem {

    private String owlType;
    private String gmlType;
    private float similarity;

    public SimTypeltem() {
    }

    public SimTypeltem(String owlType, String gmlType, float similarity) {
        setOwlType(owlType);
        setGmlType(gmlType);
        setSimilarity(similarity);
    }

    public String getOwlType() {
        return owlType;
    }

    public void setOwlType(String owlType) {
        this.owlType = owlType;
    }

    public String getGmlType() {
        return gmlType;
    }

    public void setGmlType(String gmlType) {
        this.gmlType = gmlType;
    }

    public float getSimilarity() {
        return similarity;
    }
}

```



```

    }

    public void setSimilarity(float similarity) {
        this.similarity = similarity;
    }
}

```

## Simil.java

```

package fadesgml;

import fadesgml.AttrSimItem;
import java.util.Hashtable;
import java.util.List;
import uk.ac.shef.wit.simmetrics.similaritymetrics.*;

public class Simil {

    //<editor-fold defaultstate="collapsed" desc="Atributos">
    private Hashtable types = new Hashtable();;
    private Hashtable cards = new Hashtable();;
    private JaroWinkler jaroWinkler = new JaroWinkler();;
    private Levenshtein levenshtein = new Levenshtein();;
    public static float ELEM_NAMESIM_WEIGHT = 0;
    public static float ATTR_NAMESIM_WEIGHT = 0;
    public static float REL_NAMESIM_WEIGHT = 0;
    public static float TYPESIM_WEIGHT = 0;
    public static float CONCSIM_WEIGHT = 0;
    public static float CARDSIM_WEIGHT = 0;
    //</editor-fold>

    //<editor-fold defaultstate="collapsed" desc="Construtor">
    public Simil() {
        registerType("int", "int", new Float(1.0f));
        registerType("int", "integer", new Float(0.9f));
        registerType("int", "positiveInteger", new Float(0.5f));
        registerType("int", "nonNegativeInteger", new Float(0.6f));
        registerType("int", "string", new Float(0.4f));

        registerType("float", "float", new Float(1.0f));
        registerType("float", "decimal", new Float(0.5f));
        registerType("float", "double", new Float(0.5f));
        registerType("float", "string", new Float(0.4f));

        registerType("dateTime", "dateTime", new Float(1.0f));
        registerType("dateTime", "duration", new Float(0.2f));
        registerType("dateTime", "time", new Float(0.5f));
        registerType("dateTime", "date", new Float(0.5f));
        registerType("dateTime", "gYearMonth", new Float(0.2f));
        registerType("dateTime", "gYear", new Float(0.2f));
        registerType("dateTime", "string", new Float(0.4f));

        registerType("date", "date", new Float(1.0f));
        registerType("date", "duration", new Float(0.2f));
        registerType("date", "time", new Float(0.0f));
        registerType("date", "dateTime", new Float(0.5f));
    }
}

```

```

registerType("date","gYearMonth",new Float(0.2f));
registerType("date","gYear",new Float(0.2f));
registerType("date","string",new Float(0.4f));

registerType("time","time",new Float(1.0f));
registerType("time","duration",new Float(0.2f));
registerType("time","date",new Float(0.0f));
registerType("time","dateTime",new Float(0.0f));
registerType("time","gYearMonth",new Float(0.0f));
registerType("time","gYear",new Float(0.0f));
registerType("time","string",new Float(0.4f));

registerType("boolean","booleand",new Float(1.0f));
registerType("boolean","string",new Float(0.5f));

registerType("Ponto","gml:Point",new Float(1.0f));
registerType("Linha","gml:Curve",new Float(1.0f));
registerType("Linha","gml:LineString",new Float(1.0f));
registerType("Linha","gml:OrientableCurve",new Float(1.0f));
registerType("Linha","gml:CompositeCurve",new Float(1.0f));
registerType("Poligono","gml:Solid",new Float(1.0f));
registerType("Poligono","gml:CompositeSolid",new Float(1.0f));
registerType("Poligono","gml:Polygon",new Float(1.0f));

registerCard("single","maxOccurs = 1",1.0f);
registerCard("single","maxOccurs > 1",0.5f);
registerCard("multiple","maxOccurs = 1",1.0f);
registerCard("multiple","maxOccurs > 1",1.0f);
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="public float simName(String a, String b, String
lang)">
public float simName(String a, String b, String lang) {
    int numLang = 0;
    if (lang.equals("pt")) numLang = 1;
    else if (lang.equals("en")) numLang = 2;

    switch(numLang) {
        case 1:
            return winkler(a, b);
        case 2:
            return levensh(a, b);
        default:
            return winkler(a, b);
    }
}
//    float similA = winkler(a, b);
//    float similB = levensh (a, b);

//    if ( similA > similB )
//        return(similA);
//    else return(similB);
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="public float simType(String owlType, String
gmlType)">
public float simType(String owlType, String gmlType){

```

```

Object value = types.get(owlType+gmlType);
if (value != null){
    return ((SimTypeItem)value).getSimilarity();
}

return 0;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="public float attrSim(AttrSimItem attr)">
public float attrSim(AttrSimItem attrSim){

    return          (Simil.ATTR_NAMESIM_WEIGHT          *
simName(attrSim.getOwlName(),attrSim.getGmlName(), "")) +
    (Simil.TYPESIM_WEIGHT * simType(attrSim.getOwlType(),attrSim.getGmlType()));
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="public float simConc(List<AttrSimItem>
attrSimItems, List<RelSimItem> relSimItems)">
public float simConc(List<AttrSimItem> attrSimItems, List<RelSimItem> relSimItems){

    float attrSum = 0;
    float relSum = 0;

    for(int i = 0; i < relSimItems.size(); i++){
        attrSum += attrSim(attrSimItems.get(i));
    }

    for (int i = 0; i < relSimItems.size(); i++){
        RelSimItem item = relSimItems.get(i);
        relSum += simRel(item);
    }

    return attrSum+relSum;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="public float simCard(String owlCard, String
gmlCard)">
public float simCard(String owlCard, String gmlCard){
    return ((SimCardItem)cards.get(owlCard+gmlCard)).getSimilarity();
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="public float simRel(RelSimItem relSim)">
public float simRel(RelSimItem relSim){

    return          Simil.REL_NAMESIM_WEIGHT*relSim.getSimNameValue()          +
Simil.CONCSIM_WEIGHT*relSim.getSimConcValue() +
    Simil.CARDSIM_WEIGHT*relSim.getSimCardValue();
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="public float simTuple(List<AttrSimItem>
attrSimItems, List<RelSimItem> relSimItems, int maxNodeCount)">
public float simTuple(List<AttrSimItem> attrSimItems, List<RelSimItem> relSimItems, int
maxNodeCount){

    float attrSum = 0;

```

```

float relSum = 0;

for(int i = 0; i < attrSimItems.size(); i++){
    attrSum += attrSim(attrSimItems.get(i));
}

for(int i = 0; i < relSimItems.size(); i++){
    RelSimItem item = relSimItems.get(i);
    item.setSimConcValue(attrSum);
    relSum += simRel(item);
}

return (attrSum + relSum) / maxNodeCount;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="private void registerType(String owlType, String
gmlType, float similarity)">
private void registerType(String owlType, String gmlType, float similarity){
    types.put(owlType+gmlType,new SimTypeItem(owlType,gmlType,similarity));
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="private void registerCard(String owlCard, String
gmlCard, float similarity)">
private void registerCard(String owlCard, String gmlCard, float similarity){
    cards.put(owlCard+gmlCard,new SimCardItem(owlCard,gmlCard,similarity));
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="private float winkler(String a, String b)">
private float winkler(String a, String b) {

    return( this.jaroWinkler.getSimilarity(a, b) );
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc="private float levensh(String a, String b)">
private float levensh(String a, String b) {

    return( levenshtein.getSimilarity(a, b) );
}
//</editor-fold>
}

```

## WrapperGML2.java

```

package fadesgml;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import javax.swing.border.EtchedBorder;
import javax.swing.tree.DefaultMutableTreeNode;

```

```

import org.jdom.Attribute;
import org.jdom.Content;
import org.jdom.Document;
import org.jdom.Element;
import org.jdom.JDOMException;
import org.jdom.filter.ElementFilter;
import org.jdom.input.SAXBuilder;
import org.jdom.output.XMLOutputter;

public class WrapperGML2 {

    private Element novoElemento = null;
    private ArrayList heranca = null;
    List elementos = null;
    List elementosTipoComplexo = null;

    public WrapperGML2() {
    }

    public Document loadGMLFile(File arquivo) throws JDOMException {

        Document documento = null;
        Document documentoSaida = null;

        // Constrói documento a partir do arquivo informado
        SAXBuilder construtor = new SAXBuilder();

        try {
            documento = construtor.build(arquivo);
        } catch (JDOMException ex) {

            ex.printStackTrace();

        } catch (IOException ex) {

            ex.printStackTrace();
        }

        // Constrói novo documento que será exposto na árvore
        documentoSaida = new Document(new Element("GML"));

        // Captura somente os elementos "complexType"
        elementosTipoComplexo = documento.getRootElement().removeContent( new
ElementFilter("complexType") );
        Iterator iTElementosTipoComplexo = elementosTipoComplexo.iterator();

        // Captura somente os elementos "element"
        elementos = documento.getRootElement().removeContent( new ElementFilter("element")
);

        // Instanciação das listas para guardar elementos
        // que não são colocados na árvore
        heranca = new ArrayList();

        // Navega por todos os elementos complexos do documento
        while ( iTElementosTipoComplexo.hasNext() ) {

            // Captura cada elemento "complexType" individualmente

```

```

Element nodo = (Element) iTElementosTipoComplexo.next();
String nomeTipoComplexo = nodo.getAttributeValue("name");

Iterator nodosArvore = null;
Element nodo2 = null;

// Captura todos os elementos "netos" de "complexType"
// os quais são os mesmos filhos de "complexContent"
nodosArvore = nodo.getDescendants( new ElementFilter("complexContent") );
if ( nodosArvore != null ) {
    try {
        nodosArvore = ((Element) nodosArvore.next()).getChildren().iterator();
    } catch (Exception e) {
        nodosArvore = null;
    }
}

if ( nodosArvore != null ) {

    // Captura o primeiro elemento "neto" de "complexType"
    nodo2 = (Element) nodosArvore.next();

    Element elementoTemp = null;
    String tipoElementoTemp = "";

    boolean acrescentarArvore = false;

    boolean encontrou = false;
    int index = 0;
    while ( !encontrou && index < elementos.size() ) {

        encontrou = false;

        elementoTemp = (Element) elementos.get(index);
        tipoElementoTemp = elementoTemp.getAttributeValue("type");

        if ( tipoElementoTemp.replace("urb:", "").equals( nomeTipoComplexo ) ) {
            //&& elementoTemp.getAttributeValue("name").indexOf("_") != 0 ) {

                //novoElemento = new Element("Elemento");
                novoElemento = new Element(elementoTemp.getAttributeValue("name"));

                // Se tipo complexo é "restriction",
                // o nome do novo elemento é o nome do tipo complexo
                if ( nodo.getDescendants( new ElementFilter("restriction") ).hasNext() ) {
                    // foi comentado porque o atributo nome já é o nome do elemento
                    //novoElemento.setAttribute( "Nome", nomeTipoComplexo );

                // Caso contrário, o atributo nome fica em branco
                // e o atributo tipo recebe o nome do tipo complexo
                } else {

                    String nomeElementoTemp = elementoTemp.getAttributeValue("name");
                    if ( nomeElementoTemp == null ) {
                        nomeElementoTemp = "";
                    }

                    // novoElemento.setAttribute( "name", "" );
                    //novoElemento.setAttribute( "Nome", nomeElementoTemp );
                    novoElemento.setAttribute( "Tipo", nomeTipoComplexo );
                }
            }
        }
    }
}

```

```

    }
    encontrou = true;

    }
    index++;
}

// Captura elementos sob o elemento "sequence" do "complexType"
if ( !nomeTipoComplexo.equals("QuadrasMemberType")
    && !nomeTipoComplexo.equals("LotesMemberType") ) {

    Iterator elementosSequence = obterElementosSobSequence( nodo );

    // Verifica elementos sob "sequence" e ajusta "minOccurs" e "maxOccurs"
    if ( elementosSequence != null ) {

        acrescentarArvore = criaNovoElemento( elementosSequence );
    }

    if ( acrescentarArvore || nomeTipoComplexo.equals("LoteType") ) {

        documentoSaida.getRootElement().addContent( novoElemento );
    }
}
}

Element elementoTempDDM = null;
Element elementoFilhoDDM = null;

if ( heranca != null ) {
    // elementoTempDDM = new Element("heranca");
    for ( int i = 0; i < heranca.size(); i++ ) {

        elementoFilhoDDM = (Element) heranca.get( i );
        elementoFilhoDDM = (Element) elementoFilhoDDM.detach();
        // elementoTempDDM.addContent( elementoFilhoDDM );
        documentoSaida.getRootElement().addContent( elementoFilhoDDM );
    }
    // documentoSaida.getRootElement().addContent( elementoTempDDM );
}

return documentoSaida;
}

private boolean criaNovoElemento(Iterator elementosSequence ) {

    boolean acrescentarArvore = false;
    Element elementoTemp = null;
    Element novoElementoFilho = null;

    while ( elementosSequence.hasNext() ) {
        elementoTemp = (Element) elementosSequence.next();

        Element elementoSequence = elementoTemp.getParentElement();
        Element elementoPaiSequence = elementoSequence.getParentElement();
        String nomePaiSequence = elementoPaiSequence.getName();
        String base = elementoPaiSequence.getAttributeValue("base");
        if ( base == null ) {
            base = "";
        }
    }
}

```

```

    }

    String relName = null;

    // Processa elementos "extension" (herança)
    if ( nomePaiSequence.toUpperCase().equals("EXTENSION") ) {

        // novoElementoFilho = new Element("Elemento");

        // Cria novo elemento filho e ajusta seus atributos "name" e "type"
        if ( elementoTemp.getAttributeValue("ref") != null ) {

            if (elementoTemp.getAttribute("name") != null){
                novoElementoFilho = new Element(elementoTemp.getAttribute("name").getValue());
            }
            else{
                novoElementoFilho = new Element(novoElemento.getName());
            }
            novoElementoFilho.setAttribute("Tipo", elementoTemp.getAttributeValue("ref"));
        }
        else if ( elementoTemp.getAttributeValue("name") != null ) {
            relName = elementoTemp.getAttributeValue("name");
            //APARTIR DESTA CARA, BUSCAR O NOME DO RELACIONAMENTO!
            String name = elementoTemp.getAttributeValue("type").replace("urb:", "");
            Element elem = null;
            for (int i = 0; i < elementosTipoComplexo.size(); i++) {
                Element e = (Element) elementosTipoComplexo.get(i);
                if (e.getAttributeValue("name").equals(name)){
                    elem = e;
                    break;
                }
            }

            if (elem != null){
                Iterator it = elem.getDescendants( new ElementFilter("element") );
                if ( it.hasNext() ) {
                    Element refElement = (Element) it.next();
                    novoElementoFilho = new Element(refElement.getAttributeValue("ref").replace("urb:", ""));
                    novoElementoFilho = new Element("___" + relName);
                    relElement = refElement;
                }
            }

            if ( elementoTemp.getAttributeValue("minOccurs") != null ) {
                novoElementoFilho.setAttribute("minOccurs", elementoTemp.getAttributeValue("minOccurs"));
                relElement.setAttribute("minOccurs", elementoTemp.getAttributeValue("minOccurs"));
            }
            else {
                novoElementoFilho.setAttribute( "minOccurs", "1" );
                relElement.setAttribute( "minOccurs", "1" );
            }
            if ( elementoTemp.getAttributeValue("maxOccurs") != null ) {
                novoElementoFilho.setAttribute("maxOccurs", elementoTemp.getAttributeValue("maxOccurs"));
            }
        }
    }

```



```

relElement.setAttribute("maxOccurs",elementoTemp.getAttributeValue("maxOccurs"));
    }
    else{
        //novoElementoFilho.setAttribute( "maxOccurs", "1" );
        relElement.setAttribute( "maxOccurs", "1" );
    }
    novoElementoFilho.addContent(relElement);
}
}else{
    //novoElementoFilho = new Element(elementoTemp.getAttributeValue("type"));
    novoElementoFilho = new Element(elementoTemp.getAttributeValue("name") +
"___");
    novoElementoFilho.setAttribute("Tipo",
elementoTemp.getAttributeValue("type"));
}

}
else novoElementoFilho = new Element("ElementoSemNome");

novoElemento.addContent( novoElementoFilho );

if ( ( base.toUpperCase().indexOf("ABSTRACTFEATURETYPE") == -1 ) && (
base.toUpperCase().indexOf("ABSTRACTFEATURECOLLECTIONTYPE") == -1 )){
    addSuperClassElements(novoElemento, base.replace("urb:", ""));
}

if ( base.toUpperCase().indexOf("ABSTRACTFEATURETYPE") > -1 ) {

    acrescentarArvore = true;

} else if ( base.toUpperCase().indexOf("ABSTRACTFEATURECOLLECTIONTYPE")
== -1 ) {

    heranca.add( novoElemento );
}
}

}

return acrescentarArvore;

}

private void addSuperClassElements(Element childElement, String baseName){
    boolean found = false;
    for(int i = 0; i < elementosTipoComplexo.size(); i++){
        Element elemComp = (Element) elementosTipoComplexo.get(i);
        if (elemComp.getAttributeValue("name").equals(baseName)){
            found = true;
            Iterator it = elemComp.getDescendants( new ElementFilter("element") );
            while ( it.hasNext() ) {
                Element refElement = (Element) it.next();
                if (refElement.getAttributeValue("type") != null){
                    String minOccurs = null;
                    String maxOccurs = null;
                    String relName = refElement.getAttributeValue("name");
                    String name = refElement.getAttributeValue("type").replace("urb:", "");
                    if ((childElement.getChild(relName.replace("___", "")) == null ||
childElement.getChild(relName.replace("___", "")).getChild(name) == null) &&
(childElement.getChild(relName) == null ||

```



```

    }

    if (!found){
        Element property = new Element(baseName);
        property.setAttribute("Tipo",baseName);
        childElement.addContent(property);
    }
}

// Captura elementos sob o elemento "sequence" de um complexType
private Iterator obterElementosSobSequence(Element elementoTipoComplexo) {

    Iterator sequenceElements = null;
    Iterator tempElements = elementoTipoComplexo.getDescendants( new
ElementFilter("sequence") );
    Element tempElement = null;

    if ( tempElements != null && tempElements.hasNext() ) {
        tempElement = (Element) tempElements.next();
        sequenceElements = tempElement.getChildren().iterator();
    }

    return sequenceElements;
}
}
}

```

## WrapperOWL.java

```

/*
 * WrapperOWL.java
 *
 * Adaptado de:
 * http://seed.expedita.com.pt/tiki-index.php?page=OWL+to+JTree+using+Jena+API
 *
 */

package fadesgml;

import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.util.iterator.*;
import com.hp.hpl.jena.rdf.model.*;
import java.io.File;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Enumeration;
import java.util.List;
import java.util.Map;
import java.util.NoSuchElementException;
import java.util.TreeMap;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
import java.util.Iterator;
import org.jdom.Document;
import org.jdom.Element;
import org.jdom.output.XMLOutputter;

/**

```

```

* Mostra graficamente (em forma de árvore) a hierarquia de classes de uma ontologia
*
*
*/
public class WrapperOWL extends JTree {

String sinonimos [][]; //colunas: sinonimo - classe - idioma

    public DefaultMutableTreeNode loadOWLFile(File arquivo) {

        OntModel model;

model=ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM_MICRO_RULE_INF);
        model.read( arquivo.toURI().toString() );

        Iterator itRootCls = model.listHierarchyRootClasses()
            .filterDrop( new Filter() {
                public boolean accept( Object o ) {
                    return ((Resource) o).isAnon();
                }
            });

//      while (itRootCls.hasNext()){
//          OntClass o = (OntClass) itRootCls.next();
//          System.out.println(o.getLocalName());
//      }

        //cria os nodos
        List rootNodes = createTreeNodes(itRootCls);

        //superclasse da OWL
        DefaultMutableTreeNode raiz = new DefaultMutableTreeNode("Ontourban");

        //adiciona os nodos ao nodo raiz
        Iterator it2 = rootNodes.iterator();
        while (it2.hasNext())
            raiz.add( (DefaultMutableTreeNode)it2.next() );

        return (raiz);
    }

    private List createTreeNodes(Iterator RootCls) {
        //O treemap contém a ontologia
        //O treemap permite listas ordenadas (para ordenar a árvore)
        Map map = new TreeMap();
        while (RootCls.hasNext()) {
            OntClass o = (OntClass) RootCls.next();
            if ((o.getLocalName().equalsIgnoreCase("Dicionario"))){
                continue;
            }
            map.put(o.getLocalName(), o);
        }

        //rootList contém todos os nodos de um nível
        List rootList = new ArrayList();
        Iterator a = map.values().iterator();
        while ( a.hasNext() )
            rootList.add(
                //new DefaultMutableTreeNode( ((OntClass)a.next()).getLocalName() );
                new DefaultMutableTreeNode((OntClass)a.next());
            );
    }
}

```

```

//lista as subclasses de cada classe
//classes anônimas e 'owl:Nothing' são descartadas
Iterator it = map.values().iterator();
for (int i = 0; i < map.values().size(); i++) {
    OntClass o = (OntClass) it.next();
    List sub;
    //-- teste de listagem de propriedades - adicionado 12/2006
    //System.out.println(o.getLocalName());
    //    for ( Iterator j = o.listDeclaredProperties(); j.hasNext(); ) {
    //
    //        OntProperty prop = (OntProperty)j.next();
    //    }

    // System.out.println("-----");
    //-- fim do teste -----
    DefaultMutableTreeNode temp = (DefaultMutableTreeNode)rootList.get(i);
    if ( o.hasSubClass() ) {

        sub = createTreeNodes(
            //filtro para retirar classes não utilizadas
            o.listSubClasses(true).filterDrop(
                new Filter() {
                    public boolean accept(Object o) {
                        return ((Resource)o.isAnon() ||
                            ((Resource)o.getLocalName().equals("Nothing"));
                    }
                }
            )
        );

        Iterator subit = sub.iterator();

        //adiciona todos os nodos filho
        while (subit.hasNext())
            //temp.add((DefaultMutableTreeNode)subit.next());
            rootList.add((DefaultMutableTreeNode)subit.next());
    }

    for ( Iterator j = o.listDeclaredProperties(); j.hasNext(); ) {
        OntProperty prop = (OntProperty)j.next();
        //DefaultMutableTreeNode node = new
DefaultMutableTreeNode(prop.getLocalName());
        DefaultMutableTreeNode node = null;
        boolean isGeo = false;
        if (prop.getLocalName().startsWith("_rg")){
            isGeo = true;
        }
        else{
            node = new DefaultMutableTreeNode(prop);
        }
        DefaultMutableTreeNode domain = null;
        DefaultMutableTreeNode range = null;
        if (prop.isObjectProperty()){
            //    for (ExtendedIterator iter = prop.listDomain(); iter.hasNext(); ){
            //        OntResource res = (OntResource) iter.next();
            //        if (o.getLocalName().equals(res.getLocalName())){
            //            domain = new DefaultMutableTreeNode(res.getLocalName());
            //            node.add(domain);
            //        }
            //    }

```

```

        for (ExtendedIterator iter = prop.listRange(); iter.hasNext(); ){
            OntResource res = null;
            res = (OntResource) iter.next();
            if ( (res.getLocalName() != null) && (!res.getLocalName().equals("Thing")) &&
(!res.getLocalName().equals("Resource"))){
                if ((isGeo) && (!res.getLocalName().equals("RepresentacaoGeografica"))){
                    //node = new DefaultMutableTreeNode(res.getLocalName() + "---");
                    node = new DefaultMutableTreeNode(prop.getLocalName() + "---");
                    DefaultMutableTreeNode      typeNode      =      new
DefaultMutableTreeNode("Tipo");
                    typeNode.add(new DefaultMutableTreeNode(res.getLocalName()));
                    node.add(typeNode);
                    break;
                }
                range = new DefaultMutableTreeNode("___" + res.getLocalName());
                if (prop.isFunctionalProperty()){
                    DefaultMutableTreeNode      cardNode      =      new
DefaultMutableTreeNode("Cardinalidade");
                    cardNode.add(new DefaultMutableTreeNode("single"));
                    range.add(cardNode);
                }
                else{
                    DefaultMutableTreeNode      cardNode      =      new
DefaultMutableTreeNode("Cardinalidade");
                    cardNode.add(new DefaultMutableTreeNode("multiple"));
                    range.add(cardNode);
                }
                node.add(range);
            }
        }
    }
}
else{
    //range = new DefaultMutableTreeNode(prop.getRange().getLocalName());
    range = new DefaultMutableTreeNode("Tipo");
    range.add(new DefaultMutableTreeNode(prop.getRange().getLocalName()));
    node.add(range);
}

temp.add(node);
}
}

return rootList;
}

public List<String[]> loadSinonimos(String arquivo) {

    List<String[]> lista = new ArrayList();
    OntModel      modelo      =
ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM_MICRO_RULE_INF);
    modelo.read(arquivo);

    String ns = "http://www.owl-ontologies.com/unnamed.owl#";
    OntClass classe = modelo.getOntClass( ns + "Dicionario" );
    Property sinonimo = modelo.getProperty( ns + "sinonimo");

    Iterator it = classe.listInstances();
    //retira as classes armazenadas no dicionario
    int i = 0;
}

```

```

while ( it.hasNext() ) {
    Individual ind = (Individual) it.next();
    ExtendedIterator it2 = ind.listPropertyValues(sinonimo);

    while ( it2.hasNext() ) {
        RDFNode res = (RDFNode) it2.next();
        String[] elem = new String[3];
        elem[0] = res.toString().substring(0, res.toString().indexOf("@")); //sinonimo
        elem[1] = ind.getLocalName().substring(ind.getLocalName().indexOf("_") + 1);
//classe
        elem[2] = res.toString().substring(res.toString().indexOf("@") + 1); //idioma
        lista.add(elem);
        i++;
    }
}

return lista;
}
}

```