

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**UMA ARQUITETURA DE SISTEMA GEOGRÁFICO BASEADO EM  
CONHECIMENTO COM SUSTENTAÇÃO EPISTEMOLÓGICA**

**André Salvaro Furtado**

**Florianópolis – SC**

**2010**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**UMA ARQUITETURA DE SISTEMA GEOGRÁFICO BASEADO EM  
CONHECIMENTO COM SUSTENTAÇÃO EPISTEMOLÓGICA**

**André Salvaro Furtado**

Trabalho de Conclusão de Curso apresentado  
como parte dos requisitos para obtenção do grau  
de Bacharel em Sistemas de Informação

**Florianópolis – SC**

**2010**

**André Salvaro Furtado**

**UMA ARQUITETURA DE SISTEMA GEOGRÁFICO BASEADO EM  
CONHECIMENTO COM SUSTENTAÇÃO EPISTEMOLÓGICA**

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

**Orientador:**

---

Prof. Dr. Renato Fileto

**Banca Examinadora:**

---

Profa. Dra. Carina Friedrich Dorneles

---

Profa. Dra. Patricia Della Mea Plentz

**Florianópolis – SC**

**2010**

## RESUMO

Sistemas de Informação Geográfica (SIG) têm uma importância e presença crescente desde a última década do século XX. Contudo, a capacidade de descrição do espaço nestes sistemas mostra-se limitada, quando comparada às descrições feitas por outras ciências, como a Geografia. Em contraponto aos SIGs fortemente orientados pela estrutura física dos dados, este trabalho propõe a utilização de ontologias de domínio juntamente com uma ontologia de aplicação para adicionar uma nova camada de abstração entre a interface do usuário e os dados. Tal camada visa facilitar a utilização integrada de dados de fontes heterogêneas em aplicações e possibilitar interfaces baseadas em conhecimento para o usuário especificar e manipular diversos níveis de informação organizados segundo conceitos do domínio e da aplicação para possibilitar uma melhor descrição do espaço. A proposta foi parcialmente implementada em um protótipo de SIG Web, que relaciona os conceitos das ontologias com elementos de dados das fontes heterogêneas, possibilitando uma representação mais rica do espaço geográfico e permitindo ao usuário interagir com o SIG através de descrições semânticas. Alguns experimentos iniciais foram realizados sobre este protótipo, para avaliar a viabilidade da arquitetura e das funcionalidades propostas, visando seu aperfeiçoamento antes de investir em uma implementação mais completa.

**Palavras-chave:** Sistemas de Informação Geográfica; *Web Mapping*; Cartografia Temática; Ontologias; Web Semântica.

## **ABSTRACT**

Geographic Information Systems (GIS) have had a growing importance and presence since the last decade of the XX century. However, the ability of these systems to describe the space is limited when compared with the descriptions done in other sciences, like Geography. This work uses a different approach from that of the GIS tightly oriented to the physical structure of the data sources. It proposes the use of domain ontologies along with an application ontology to describe data elements and provide an new abstraction layer between the user's interface and the data. This layer was created to facilitate the integrated use of data from heterogeneous data and enable the use of knowledge-based interfaces that allow to the user specify and handle several levels of information organized according conceptualizations of the domain and the application to enable a better description of the space. It was partially implemented in an application prototype of a Web GIS, linking the concepts of both ontologies with different data sources, allowing the user to interact with the data using semantic descriptions that are in the domain ontology. Some initial experiments where performed over this prototype to evaluate the viability of the architecture and the functionalities proposed, aiming it's improvement before a complete implementation.

**Keywords:** GIS; *Web Mapping*; Thematic Cartography; Ontologies; Semantic Web.

# SUMÁRIO

LISTA DE FIGURAS.....	9
GLOSSÁRIO DE NOMES E ACRÔNIMOS.....	11
1 INTRODUÇÃO.....	13
1.1 OBJETIVOS.....	15
1.2 MOTIVAÇÃO E JUSTIFICATIVA.....	17
1.3 METODOLOGIA.....	18
1.4 ORGANIZAÇÃO DO TRABALHO.....	19
2 CARTOGRAFIA.....	21
2.1 SISTEMAS DE COORDENADAS.....	21
2.2 PROJEÇÕES CARTOGRÁFICAS.....	22
2.3 ESCALA.....	23
2.4 CARTOGRAFIA TEMÁTICA.....	23
2.5 SISTEMAS DE INFORMAÇÃO GEOGRÁFICA.....	23
3 A COMPUTAÇÃO E A GEOGRAFIA.....	26
3.1 O ESPAÇO.....	27
3.2 ELEMENTOS ESPACIAIS.....	28
3.3 O TEMPO.....	29
3.4 ELEMENTOS ESPACIALIZÁVEIS E EVENTOS.....	31
3.5 O CONHECIMENTO.....	32
3.6 A REPRESENTAÇÃO COMPUTACIONAL DOS ELEMENTOS DO ESPAÇO.....	33
4 PADRÕES E TECNOLOGIAS.....	35
4.1 OPEN GEOSPATIAL CONSORTIUM (OGC) .....	35
4.2 KEYHOLE MARKUP LANGUAGE (KML).....	35
4.3 BANCO DE DADOS.....	36
4.3.1 POSTGRESQL.....	36
4.3.2 POSTGIS.....	37
4.4 TECNOLOGIAS DE DESENVOLVIMENTO.....	38

4.4.1	GOOGLE WEB TOOLKIT (GWT)	38
4.4.2	OPENLAYERS	40
4.4.3	EXT-JS	41
4.4.4	PROTEGÉ	41
5	SISTEMA GEOGRÁFICO BASEADO EM CONHECIMENTO	42
5.1	REPRESENTANDO O ESPAÇO, A APLICAÇÃO E O DOMÍNIO	43
5.1.1	ONTOLOGIA PARA DESCRIÇÃO DO ESPAÇO	43
5.1.2	ONTOLOGIA PARA DESCRIÇÃO DA APLICAÇÃO	45
5.1.3	ELEMENTO DE UMA ONTOLOGIA VOLTADA AO DOMÍNIO	46
5.2	ARQUITETURA PROPOSTA	48
5.2.1	ONTOLOGIA	50
5.2.2	INDEXADOR E CLASSES INDEXADAS	51
5.2.3	FONTES DE DADOS E MAPEAMENTOS	52
5.2.4	RECUPERAÇÃO DO CONHECIMENTO	54
5.2.5	INTERFACE CARTOGRÁFICA E NÃO-CARTOGRÁFICA	55
6	IMPLEMENTAÇÃO	56
6.1	CARACTERÍSTICAS DO PROTÓTIPO	56
6.2	DEFINIÇÃO DA ESTRUTURA DE DADOS DAS CLASSES INDEXADAS	57
6.3	IMPLEMENTAÇÃO DO INDEXADOR	58
6.4	MAPEAMENTO DOS DADOS	59
6.5	MÓDULO PARA RECUPERAÇÃO DO CONHECIMENTO	60
6.6	TIPOS DE CAMADAS NO MAPA	61
6.7	CRIAÇÃO DA INTERFACE	62
6.7.1	ESTILO DOS ELEMENTOS	63
6.8	PROTÓTIPO IMPLEMENTADO	64
7	EXPERIMENTAÇÃO	68
7.1	FONTES DE DADOS	68
7.2	CENÁRIO INICIAL	69

<b>7.3 RESULTADOS.....</b>	<b>71</b>
<b>8 CONCLUSÃO E TRABALHOS FUTUROS.....</b>	<b>78</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>81</b>

## **LISTA DE FIGURAS**

Figura 1: Projeção Cartográfica

Figura 2: Imagem Vetorial do Centro de Florianópolis

Figura 3: Imagem Matricial do Centro de Florianópolis

Figura 4: Arquitetura de Comunicação do GWT

Figura 5: Adição de Camada de Estruturas de Conhecimento

Figura 6: Ontologia para a descrição do espaço

Figura 7: Ontologia para a descrição dos conceitos da aplicação

Figura 8: Elemento Hospital

Figura 9: Anotações das fontes de dados com base nos conceitos de aplicação e domínio

Figura 10: Arquitetura proposta

Figura 11: Mapeamento das entidades nas fontes de dados armazenadas em XML

Figura 12: Exemplo de Mapeamento de Dados

Figura 13: Modelos de interação com Camadas de Informação

Figura 14: Interface inicial do protótipo – Camada base do Google Maps

Figura 15: Seleção de elementos para interação com camadas de informação

Figura 16: Carregamento das ontologias

Figura 17: Recuperação dos dados presentes nas fontes externas

Figura 18: Cenário inicial do protótipo

Figura 19: Resultado do Experimento com dados de diferentes fontes integrados

Figura 20: Elemento FireStation

Figura 21: Resultado do Experimento 2

Figura 22: Elementos “NonNatural” carregados

Figura 23: Interação hierárquica com camada de informação

## **GLOSSÁRIO DE NOMES E ACRÔNIMOS**

AJAX – Asynchronous Javascript And XML

API – Application Programming Interface

BD – Banco de Dados

CSS – Cascading Style Sheets

ESRI – Environmental System Research Institute

EPSG - European Petroleum Survey Group

GIS – Geographic Information System

GML – Geography Markup Language

GWT – Google Web Toolkit

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

ICA - International Cartographic Association

JS – JavaScript

JSNI – JavaScript Native Interface

JSON – JavaScript Object Notation

OGC – Open Geospatial Consortium

OWL – Web Ontology Language

POJO – Plain Old Java Object

RPC – Remote Procedure Call

SHAPEFILE (SHP) – Formato proprietário de dados Geográficos da ESRI

SIG – Sistema de Informação Geográfica

SGBD – Sistema de Gerenciamento de Banco de Dados

SLD – Styled Layer Descriptor

SQL – Structured Query Language

SVG – Scalable Vector Graphics

UTM – Universal Transverse Mercator

XML – Extensible Markup Language

W3C – World Wide Web Consortium

WFS – Web Feature Service

WKT – Well-Known Text

WMS – Web Map Service

# 1 INTRODUÇÃO

A idéia de referenciar geograficamente elementos - unidades básicas de sistemas (HARVEY, 1969), como um Hospital em um Sistema de Saúde ou um rio em uma Rede de Drenagem - presentes nas atividades humanas e construir mapas é milenar e está fortemente ligada à história da civilização.

*“Mapmaking is an interesting subject to study and an activity that has enjoyed a long history, closely tied to the history of civilization itself. Maps date as far back as the fifth or sixth century B. C.”*

Dent (1996)

Praticamente tudo o que acontece – e existe -, acontece em algum lugar e saber onde algo acontece pode ser muito importante (LONGLEY, 2001), por isso é natural o surgimento de trabalhos visando mapear os elementos presentes espaço.

Estes trabalhos têm acompanhado a civilização ao longo de seu desenvolvimento e têm sido foco de ciências como a engenharia cartográfica, o mapeamento temático e, desde as últimas décadas do século XX, os SIG, Sistemas de Informação Geográfica (FRIENDLY & DENIS, 2006).

Sobre os Sistemas de Informação Geográfica, cabe destacar a importância da ciência da computação e dos sistemas de informação no esforço para representar elementos georreferenciados e na consolidação do geoprocessamento (ROCHA, 2000). Estas áreas têm apresentado significativos resultados tais como softwares para processar grandes volumes de dados georreferenciados, efetuar conversões de projeções cartográficas, realizar composições de camadas em mapas e possibilitar consultas com operadores espaciais, entre outras funcionalidades.

Muito por razão de sua origem, os Sistemas de Informação Geográfica acabaram por assumir um caráter fortemente cartográfico, comportando-se como “sistemas cartográficos de informação” (CÂMARA, 2000b). Conseqüentemente eles são baseados predominantemente em

uma abordagem geométrica para a análise, desconsiderando muito do estudado em outras áreas da Geografia, como conceitos relativos à importância da dimensão temporal, organização, descrição e representação do espaço.

Na década de 60 do século XX já se fazia presente na Geografia a discussão acerca da necessidade de interpretação dos fatos geográficos além da abordagem cartográfica.

*"Até o século XIX, não passava de um conhecimento descritivo e cartográfico. Já na segunda metade e fins desse século, começam a ganhar força a explicação e a interpretação dos fatos geográficos"* (CARVALHO, 1965).

Para buscar uma abordagem que vá além da cartografia, vise descrever os elementos e explicar os fatos geográficos através de critérios ainda objetivos, a Geografia Crítica no trabalho de Santos (2008) apresentou uma alternativa para descrição dos elementos do espaço utilizando quatro diferentes aspectos: (i) a forma - como o elemento é fisicamente; (ii) a função - o que o elemento faz, para que ele serve; (iii) estrutura - onde o elemento se encaixa, como os diferentes elementos estão organizados e interligados; e (iv) o processo - o que ocorre sobre a estrutura através das funções que os elementos possuem.

A dificuldade na representação de conceitos não-cartográficos por parte dos SIGs é um problema reconhecido e provém da dificuldade na criação de modelos computacionais para estes conceitos (CÂMARA et al., 2002). Entretanto, a evolução dos estudos na área de Representação do Conhecimento apresentou novas abordagens com grande poder de descrição, como as ontologias, e alternativas para o uso de ontologias em Sistemas de Informação (GUARINO, 1997) possibilitando a criação de Sistemas Geográficos baseados em Conhecimento.

Uma das possibilidades futuras apresentada em Camara *et al.* (2002) associa o uso das técnicas de representação do conhecimento e SIG como forma de se aproximar da base teórica proposta pela Geografia Crítica. Esta abordagem possibilita alternativas na resolução de alguns problemas e dificuldades encontrados pela atual geração de Sistemas de Informação Geográfica:

- Criar modelos para representar relações entre os elementos do espaço;
- Adicionar o conhecimento dos elementos do espaço e de sua organização a um SIG;
- Criar alternativas para a organização de informação sobre o espaço de maneira hierárquica, que possibilite a integração da informação em diferentes níveis;
- Facilitar a adaptação de um mesmo sistema para diferentes domínios, de modo a obter um maior aproveitamento do conhecimento do especialista de domínio que o obtido em um SIG genérico orientado pela estrutura física dos dados e também que o obtido em um SIG especializado em determinada área que necessitaria um trabalho de codificação para sua adaptação;

## **1.1 OBJETIVOS**

Este trabalho tem como objetivo a proposição de uma arquitetura de Sistema Geográfico de Informação baseado em Conhecimento. Tal arquitetura visa associar aos dados o conhecimento das estruturas presentes no espaço, adicionando outra camada de abstração (uma estrutura de conhecimento), para permitir a manipulação integrada de dados de fontes heterogêneas, de acordo com o conhecimento de domínio e não com base na descrição física das fontes de dados.

Os objetivos específicos deste trabalho são:

1. Propôr uma arquitetura de Sistemas Geográficos de Informação baseada em técnicas de Representação do Conhecimento que promova o reuso de dados e código usado para manipulá-los, que seja desenvolvido de acordo com tal arquitetura, minimizando alterações em bases de dados e código fonte.

2. Criar esboços de ontologias para a representação de conhecimento na arquitetura proposta: (i) uma ontologia de alto nível, que descreva os principais conceitos do espaço segundo a visão da geografia crítica; (ii) uma ontologia de aplicação que descreva os conceitos conhecidos pelo código da aplicação (atributos, medidas e geometria dos elementos,

entre outros) e (iii) um pequeno trecho de ontologia de domínio que descreva a organização urbana de uma cidade (com conceitos como Construção e Hospital) especializando a ontologia de alto nível para ilustrar a aplicação da proposta e iniciar a sua validação e aperfeiçoamento.

3. Implementar um protótipo de sistema geográfico de informação baseado na arquitetura proposta e efetuar experimentos, visando avaliar tal arquitetura quanto a seus objetivos, integrando dados provenientes de fontes de dados heterogêneas com diferentes níveis de detalhamento.

Os objetivos não funcionais da proposta são:

1. Realizar um diálogo com a Geografia, de modo a compreender descrições complexas do espaço existentes nessa ciência, para então criar modelos formais capazes de representar os elementos presentes em nossa sociedade que sejam relevantes para análises através do uso de ontologias.

2. Propor uma alternativa para a representação de elementos espaciais específicos de domínio nos Sistemas de Informação Geográfica, de modo que esta representação contemple os conceitos de organização e descrição do espaço vindos da Geografia e que esses conceitos fiquem armazenados em uma estrutura flexível (no caso, em uma estrutura de ontologia).

3. Permitir o desenvolvimento de sistemas aplicativos facilmente adaptáveis e extensíveis, com o auxílio de especialistas de domínio. Isso deve minimizar o trabalho de codificação de novas aplicações, mediante o reuso do código da aplicação SIG existente, efetuando alterações apenas na ontologia que descreve os conceitos de domínio (alterar um sistema voltado para a área da segurança pública para produzir um sistema que possa ser utilizado na área da saúde pública, por exemplo).

## 1.2 MOTIVAÇÃO E JUSTIFICATIVA

A evolução de técnicas computacionais de representação do conhecimento foi marcante ao longo da primeira década do século XXI, resultando no surgimento de alternativas com grande poder de descrição das estruturas conhecidas. Considerando esta evolução e o fato de outras ciências terem avançado na descrição do espaço, o uso de estruturas capazes de armazenar essa descrição do espaço (como ontologias) em Sistemas Geográficos de Informação é um rumo natural.

Outros trabalhos buscaram alternativas para esta união, abordando questões como recuperação da informação e interoperabilidade entre sistemas (FONSECA, 1999). Todavia a principal motivação do trabalho aqui apresentado é adicionar uma camada de abstração com descrições semânticas de elementos presentes no espaço visando uma descrição do espaço que implique em menor perda semântica em Sistemas de Informação Geográfica. Em outras palavras, pretende-se ir além do caráter geométrico dos Sistemas de Informação Geográfica, que geralmente são orientados pela organização física dos dados na fonte. Pretende-se assumir um caráter geográfico, através da adição de uma camada intermediária de abstração (uma ontologia) com o conhecimento descritivo dos elementos presentes no espaço.

Este trabalho pretende exercitar a idéia de Sistema Geográfico de Informação baseado em uma descrição conceitual do espaço, utilizando bases teóricas vindas principalmente da Geografia. Ao reunir conceitos de diferentes ciências em uma abordagem inter-disciplinar, pretende-se buscar um modelo para descrição e representação do espaço em SIG.

A necessidade de representações computacionais mais complexas do espaço – representações que levem em conta o conhecimento do domínio que se deseja representar - e uma visão integradora entre suas estruturas e processos são latentes (CÂMARA *et al*, 2002). Muito ainda precisa ser trabalhado neste sentido.

Outra justificativa vem do fato de Sistemas de Informação Geográfica terem se tornado bastante populares em tentativas de representação de diferentes domínios como a Saúde, Segurança, Planejamento Urbano, Social entre outros (LONGLEY, 2001). A necessidade de

trabalhar com todas essas áreas fez surgir dois tipos diferentes de SIG: (i) um com uma abordagem bastante genérica e mais preocupado em reproduzir os dados de uma base do que em organizá-la de acordo com o conhecimento do domínio; (ii) e outro com uma abordagem exatamente oposta, por demais especializado, dificultando sua adaptação e demandando muito trabalho de desenvolvimento para adaptá-lo a outro domínio.

Ao manter o conhecimento em uma estrutura de mais flexível e separada do código da aplicação, como uma ontologia, o trabalho aqui proposto torna a adaptação de um sistema para diferentes domínios muito mais simples. Isso valoriza o conhecimento do especialista de domínio que é muito mais familiarizado com os conceitos da sua área do que com a codificação de estruturas lógicas em linguagens de programação e/ou complexos relacionamentos em bases de dados.

### **1.3 METODOLOGIA**

O desenvolvimento deste trabalho iniciou com uma revisão bibliográfica de cinco principais temas: (i) teorias para a descrição do espaço da Geografia com foco na abordagem da Geografia Crítica, principalmente a abordagem de Milton Santos, descrita em suas obras “Espaço e Método” (SANTOS, 2008), “A Natureza do Espaço” (SANTOS, 2006), “Por uma Geografia Nova” (SANTOS, 1980) e “Metamorfoses do Espaço Habitado (SANTOS, 1988); (ii) conceitos básicos da Cartografia (IBGE, 2009) e da Geografia (HARVEY, 1969); (iii) teorias para representação do conhecimento por meio de ontologias e sua relação com Sistemas de Informação (GRUBER, 1995) (GUARINO, 1997); (iv) abordagens para combinar os conceitos de Sistemas de Informação Geográfica com estruturas de conhecimento (FONSECA, 1999); e (v) documentos contendo especificações da OpenGIS (OpenGIS, 2006) (OpenGIS, 2008)(OpenGIS, 2008b).

Após as leituras realizadas foram definidos os fundamentos do trabalho, as relações entre o conteúdo teórico da Geografia e descrições do espaço, do tempo e representação de elementos espaciais e informação, para então definir os modelos utilizados neste trabalho.

Finalmente foi implementado um protótipo que seguindo a arquitetura proposta. Tal implementação utilizou as linguagens Java e Javascript e como fontes de dados o SGBD PostgreSQL e o formato KML.

Com o protótipo implementado, o passo final foi a experimentação. Foi feita a modelagem simplificada de uma estrutura de conhecimento sobre a organização urbana da cidade de *Medford*. Esta estrutura foi utilizada no protótipo desenvolvido de acordo com a arquitetura visando validação desta através de experimentos que viessem a cumprir os objetivos do trabalho.

## **1.4 ORGANIZAÇÃO DO TRABALHO**

O segundo capítulo deste trabalho descreve conceitos presentes na Cartografia e Sistemas de Informação Geográfica: sistemas de coordenadas, projeções cartográficas, escala, cartografia temática e uma revisão de conceitos de SIG.

O terceiro capítulo apresenta a relação entre conceitos geográficos e conceitos computacionais para representação do espaço e do tempo, além de uma seção descrevendo brevemente ontologias.

O quarto capítulo descreve as tecnologias utilizadas no desenvolvimento do protótipo apresentado posteriormente no trabalho.

A arquitetura proposta para um Sistema Geográfico Baseado em Conhecimento é detalhada no quinto capítulo do trabalho. Este capítulo também explica o funcionamento de diferentes módulos presentes na arquitetura.

O sexto capítulo mostra o protótipo implementado de acordo com a arquitetura apresentada no capítulo anterior, com suas características funcionais e não-funcionais, descrevendo detalhes da implementação e interface.

No sétimo capítulo são apresentados os experimentos realizados, sendo estes avaliados em relação ao cumprimento dos objetivos do trabalho.

O último capítulo do trabalho é dedicado à conclusão e à descrição de possíveis trabalhos futuros.

## 2 FUNDAMENTOS BÁSICOS DA CARTOGRAFIA

A Cartografia é a ciência responsável por modelar o componente espacial da realidade de um determinado lugar (IBGE, 2010) e representá-lo através da produção de mapas, utilizando conceitos como projeções cartográficas, sistemas de coordenadas e escala.

A ICA (2010) define mapa como *“a symbolized image of geographical reality, representing selected features or characteristics, resulting from the creative effort of its author's execution of choices, and is designed for use when spatial relationships are of primary relevance.”*.

### 2.1 SISTEMAS DE COORDENADAS

Independente da forma escolhida para representar a Terra, para localizar um ponto em sua superfície é necessário um sistema de coordenadas.

Em representações planas, três tipos de coordenadas são mais utilizados:

- **Coordenadas Geográficas:** é o sistema mais comum, sendo representado por um par de coordenadas indicando a latitude e longitude em graus. (Ex: latitude - 27.59° e longitude -48.54°, coordenadas da cidade de Florianópolis).
- **Coordenadas Cartesianas:** com base em eixos X e Y definem a posição de um ponto no plano. (Ex: coordenada simples  $x=5$  e  $y=2$  em um determinado plano cartesiano).
- **Coordenadas Métricas:** usadas na projeção Universal Transversa de Mercator, são definidas com base na carta ao milionésimo da Projeção UTM, utilizando junto das coordenadas métricas a referência da Zona UTM a qual ela pertence. (Ex:  $x=742825.5926973042$ ,  $y=6945798.81878737$  na zona 22 do hemisfério sul é a coordenada UTM da cidade de Florianópolis).

## 2.2 PROJEÇÕES CARTOGRÁFICAS

A forma curva (elipsóide) da Terra é um problema básico para a representação cartográfica, posto que não é possível representar perfeitamente seu formato em uma estrutura plana em duas dimensões (IBGE, 2009).

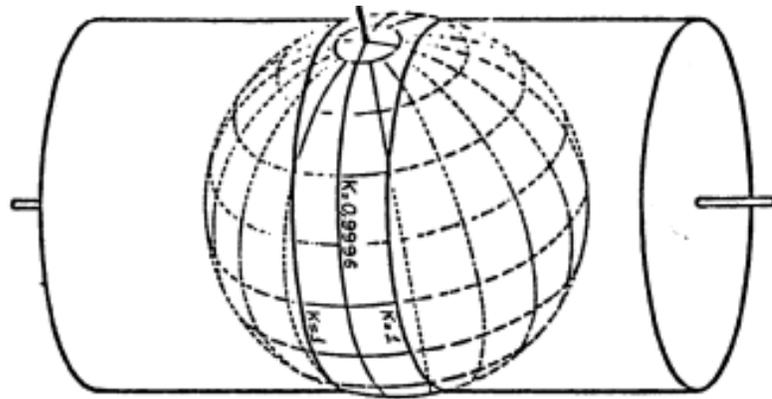


Figura 1: Projeção Cartográfica

Na tentativa de solucionar esse problema surgiram diferentes maneiras de representar a Terra em um plano, as projeções cartográficas. Contudo, representar perfeitamente a forma esférica da Terra não era algo possível, fazendo com que as diferentes formas de projeção “sacrificassem” algumas das variáveis envolvidas, resultando em distorções de área, distância e forma.

Para este trabalho foram utilizadas duas projeções, (i) uma conhecida como WGS84 – que utiliza o elipsóide de mesmo nome como base e possui como código o EPSG:4326 – código que a referencia no *EPSG Geodetic Parameter Dataset*) que possui o sistema de coordenadas geográficas presente nas fontes de dados da aplicação e (ii) uma projeção semelhante ao UTM (Figura 1), conhecida como *Spherical Mercator* comum em camadas base como as fornecidas pela Google (e que por isso é informalmente referenciada pelo código EPSG:900913 em referência à Google), que utiliza um sistema métrico para as coordenadas.

## 2.3 ESCALA

Escalas servem para indicar a proporção entre o tamanho apresentado na representação Cartográfica e o tamanho real de um elemento.

Elas são fundamentais para a compreensão de qualquer mapa e podem ser representadas de maneira numérica e/ou gráfica. Sua ausência pode tornar um determinado mapa completamente sem utilidade posto que qualquer definição relativa a medidas e distâncias depende da escala.

## 2.4 CARTOGRAFIA TEMÁTICA

A Cartografia Temática é responsável pela escolha da representação visual de dados em mapas, utilizando símbolos, cores, tamanhos, texturas entre outras variáveis visuais e técnicas de visualização (por exemplo, agrupamentos espaciais) de modo a facilitar a visualização da informação contida em um mapa, seus elementos cartográficos e os dados destes que se deseja representar.

*“The close relation of the numbers of the state (the origin of the word “statistics”) and its geography gave rise to the visual representation of such data on maps, now called “thematic cartography”* é a definição trazida por Friendly e Denis (2006) em seu trabalho dedicado a estabelecer os marcos históricos da Cartografia Temática.

## 2.5 SISTEMAS DE INFORMAÇÃO GEOGRÁFICA

Sistemas de Informação Geográfica é um termo definido por Câmara e Queiroz (2000) como: *“sistemas que realizam o tratamento computacional de dados geográficos e recuperam informações não apenas com base em suas características alfanuméricas, mas também através de sua localização espacial”*.

Estes sistemas possuem a capacidade de efetuar análises, transformações e operações espaciais sobre os dados, que por sua vez possuem duas maneiras principais de representação: vetorial e matricial.

Os dados com representação vetorial possuem um caráter fortemente geométrico possuindo três elementos básicos: ponto, linha e polígono, e composições entre esses elementos (Exemplificado na Figura 2). Sobre estes dados vetoriais os SIGs permitem a execução de filtros baseados em suas relações topológicas apresentadas por Egenhofer (1994).

Ao contrário dos dados vetoriais que são excelentes para representar elementos geográficos com limites bem definidos, os dados matriciais (ou *raster*) são ideais para a representação de espaços contínuos, não delimitados, apresentando uma estrutura que possui informação “anexada” a cada pixel (Exemplificado na Figura 3). Em Sistemas de Informação Geográfica essa forma de dado é muito útil na representação de camadas base como imagens de satélite e na criação de modelos que consideram medidas contínuas e não-delimitadas precisamente no espaço, como elevação, variação de temperatura e outras.

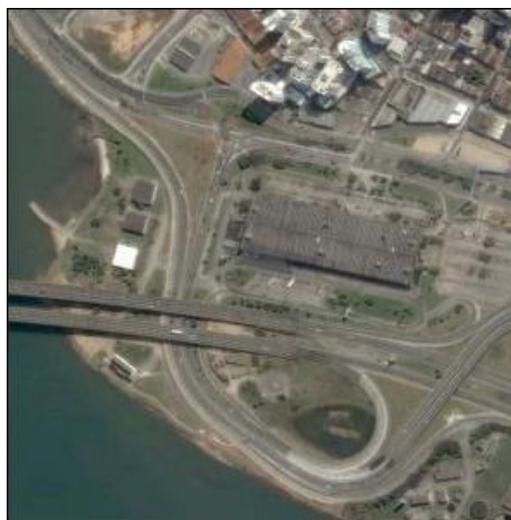


Figura 2: Imagem Vetorial (esquerda) e Figura 3: matricial (direita) do Centro de Florianópolis  
Obtida em <http://openstreetmap.org>.

Obtida em <http://bing.com/maps>.

Outro conceito importante nos Sistemas de Informação Geográfica é o de *feature*. Uma *feature* é uma abstração de um fenômeno do mundo real – como por exemplo uma construção, um campo aberto, uma universidade, um hospital. Uma *geographical feature* é um fenômeno do mundo real associado a uma determinada localização no espaço (OpenGIS, 2006).

O Open Geospatial Consortium também apresenta uma especificação completa de como é caracterizada uma *feature*:

*“Any feature may have a number of properties. These properties may be operations, attributes or associations. Any feature may also have a number of attributes: Spatial, Temporal, Quality, Location, Metadata, Thematic. A feature is not defined in terms of a single geometry, but rather as a conceptually meaningful object within a particular information or application community, one or more of the feature’s properties may be geometric.”* (OpenGIS, 2006)

Com base nessa especificação são implementadas aplicações e serviços (como o caso do WFS, padrão da OGC para distribuição de dados geográficos sob a abstração de *features*).

### 3 A COMPUTAÇÃO E A GEOGRAFIA

Estudar o espaço e descrevê-lo tem sido objeto de estudo de diversas áreas da ciência ao longo dos séculos, tendo especial destaque na Filosofia e na Geografia. A computação em sua tarefa de criar modelos que representem a realidade também apresentou por meio dos Sistemas de Informação Geográfica uma alternativa para a descrição do espaço. Contudo, esta descrição, baseada principalmente em operações diretas sobre os dados e em abstrações sob o conceito de *geographical features*, apresenta limitações quando comparada com a abordagem de outras ciências, como a Geografia, e não se mostra suficiente para a representação de leituras mais complexas do espaço - leituras que agreguem conhecimento da organização espacial.

Os modelos encontrados pelos referidos Sistemas de Informação Geográfica podem ser associados com descrições espaciais provenientes da Geografia, dado o caráter “transdisciplinar” que envolve o geoprocessamento abordado por Rocha (2000).

Entretanto o diálogo entre uma ciência exata e uma ciência humana nem sempre é fácil e direto. Câmara et al (2002) proporcionou esse diálogo com a Geografia e seu encontro com as representações computacionais do espaço, ao buscar bases epistemológicas para a ciência da Geoinformação, destacando a forte relação entre os SIGs com diferentes escolas da geografia tais como:

- A Geografia Idiográfica Hartshorniana:

*"A proposta de Hartshorne contribuiu para dar uma base metodológica para o uso do conceito de "unidade de área" em Geoprocessamento. A representação computacional correspondente aos conceitos de "unidade de área" em Hartshorne é o polígono fechado, que delimita cada região de estudo e um conjunto de atributos, tipicamente armazenados num banco de dados relacional."*

- a Geografia Quantitativa:

*"De uma forma bastante geral, pode-se dizer a forma privilegiada de representação computacional associada à Geografia Quantitativa são as superfícies, que no computador correspondem a grades regulares e malhas triangulares e as redes, representadas por matrizes de interação espacial."*

- a Geografia Crítica, que pode ser base de uma nova geração de SIG:

*"Podemos apenas especular sobre as representações computacionais que serão utilizadas neste contexto, que possivelmente incluam técnicas de Representação de Conhecimento."*

Promovendo este diálogo com a Geografia, em especial com as idéias de Santos (2006, 2008) um dos mais dedicados na proposição de um conceito universal para o espaço, este trabalho busca uma reunião entre a abordagem do espaço pela Geografia moderna com uma representação computacional em conjunto com técnicas de representação do conhecimento. Apresentando então uma arquitetura de Sistema Geográfico que considere o conhecimento acerca dos elementos presentes no espaço.

### **3.1 O ESPAÇO**

Como já referido anteriormente neste trabalho, o estudo do espaço e das relações que ocorrem no espaço foi objeto de uma série de discussões ao longo da história da Geografia com o objetivo de definir um objeto de estudo.

Cholley (1964) já discutia a dificuldade de se definir um domínio da Geografia em contraponto às ciências exatas e naturais. Ele afirma que essas ciências "se esforçam em decompor as combinações em seus elementos mais simples, para depois, considerá-los separadamente, enquanto a Geografia, por sua vez, toma a própria combinação como objeto de seu estudo". Em seguida ele apresenta a idéia das combinações como objeto de estudo da ciência geográfica.

Santos (1988) apresenta outra visão, ao destacar a importância de um esforço na descrição do espaço e de seus elementos e ao questionar a renúncia de uma definição objetiva

quando confrontada com a multiplicidade de visões acerca das mesmas coisas. *“Certo, dizemos nós, existem percepções diversas das mesmas coisas, pois há indivíduos diferentes. Mas deve-se por isso renunciar à aproximação de uma definição objetiva das realidades? Do contrário, não se saberia sequer por onde começar o trabalho científico. E estaríamos sempre à mercê de uma ambigüidade. Com efeito, para a questão que nos interessa, é preciso transformar num só o que parece um duplo problema. Trata-se de definir o espaço da geografia, seja ela uma geografia renovada ou redefinida, e estabelecer assim seu objeto e limites.”* (SANTOS, 1988)

A computação possui uma definição mais simples que a Geografia para descrever o espaço, o conceito de *features* geográficas apresentado pela OpenGIS (2008) em seu modelo de referência explica alguns conceitos:

*“A feature is an abstraction of a real world phenomenon. A geographic feature is a feature associated with a location relative to the Earth. A digital representation of the real world can be thought of as a set of features.”.*

Definindo o espaço como um conjunto de *features* e seus atributos que podem ser descritivos, temáticos, espaciais, medidas, etc.

### **3.2 ELEMENTOS ESPACIAIS**

Diferentes formas de descrição do espaço foram apresentadas ao longo dos últimos dois séculos pela ciência geográfica, este trabalho utilizará uma abordagem baseada no trabalho do geógrafo Milton Santos (SANTOS, 2008).

A abordagem utilizada neste trabalho é a apresentada no livro Espaço e Método (2008). Esta abordagem é baseada na definição de Harvey (1969) para elemento: “a unidade básica de um sistema”.

A partir desta definição de elemento ela define os diferentes aspectos presentes nos elementos do espaço: (i) forma - o aspecto visível de um elemento ; (ii) função - o quê é executado a partir daquela forma -; (iii) estrutura – as ligações entre os elementos; e (iv) o

processo – a dinâmica das modificações dos estados dos elementos interligados. Esses aspectos definem como estes elementos estão organizados e como eles interagem.

Através da definição de elemento e da conceituação destes quatro aspectos é possível pensar nestes elementos do espaço como objetos geográficos, isto é, elementos espaciais dotados de forma e função e presentes nas estruturas e processos de nossa sociedade.

A partir da definição de um objeto geográfico e de como ele é definido uma modelagem formal do objeto é não só possível, como uma interessante solução para trabalhar computacionalmente com os elementos presentes no espaço.

Para ilustrar estes conceitos pode-se utilizar um exemplo do mundo real. Seja a estrutura de um Sistema de Saúde. Ela é composta de elementos como hospitais, centros de distribuição de medicamentos e postos de saúde, entre outras formas espaciais, as quais são dotadas de funções. Tal estrutura de elementos interconectados é usada para realizar processos como o atendimento de pessoas doentes e campanhas de vacinação no espaço ao longo do tempo.

O conceito de elemento espacial é utilizado neste trabalho para descrever os elementos do mundo real que possuem forma e função e que sejam parte de uma estrutura e sobre a qual ocorram processos.

### **3.3 O TEMPO**

A dimensão temporal possui fundamental importância no estudo do espaço. A falta de uma visão conjunta de espaço e tempo torna impossível a compreensão e a visualização de fatores como a formação social e as transformações (porque não evoluções?) do espaço habitado.

Sendo assim, estudar o tempo, sua ligação com o espaço e suas propriedades torna-se necessário quando existe a intenção de representar o espaço. A base para o modelo desta

representação o conhecimento geográfico advindo de sua estrutura (e neste caso não só a estrutura física) e processos.

Santos (2006) ao descrever essa transformação do espaço (e de sua organização) ao longo do tempo aponta a atribuição de novas funções as mesmas formas, sendo estas provenientes de movimentos da sociedade: *“os movimentos da sociedade, atribuindo novas funções às formas geográficas, transformam a organização do espaço, criam novas situações de equilíbrio e ao mesmo tempo novos pontos de partida para um novo movimento.”*.

A diferenciação na interpretação de um mesmo espaço em tempos distintos em também é mencionada pelo autor: *“O que a caracteriza no Tempo 1 não é aquilo que a define no Tempo 2. ... No Tempo 2, encontraremos situações diferentes daquelas do Tempo 1.”* E define: *“Sem dúvida, a cada sistema temporal o espaço muda.”*.

Embora na Computação um modelo unificado para a questão temporal (e sua relação com o espaço) esteja distante, a área tem tido notáveis evoluções, resultados do esforço no sentido de compreender o tempo e suas propriedades de maneira a torná-lo representável computacionalmente.

Uma proposta de representação espaço-temporal é apresentada por Casanova et al (2005), onde são analisados os aspectos de ordem, variação e granularidade.

O conceito de granularidade é definido por três elementos básicos de representação *“instante, intervalo e elemento temporal”*. Sendo o instante *“um ponto no tempo cuja duração é infinitesimal”*, o intervalo *“um subconjunto de pontos do eixo temporal equivalente ao tempo decorrido entre dois pontos”* e o elemento temporal *“a união finita de intervalos de tempo”*.

Do conceito de ordem vem a importante definição de tempo circular, representado por eventos recorrentes, casos em que *“a periodicidade de sua ocorrência faz com que sempre se volte à mesma referência de tempo”*.

### 3.4 ELEMENTOS ESPACIALIZÁVEIS E EVENTOS

Nas duas seções anteriores foram tratados conceitos que podem ser formalizados para a descrição do espaço (objetos geográficos) e para a descrição do tempo (instante e período) utilizando de maneira conjunta os conceitos da Geografia e da Computação.

Com espaço e tempo definidos e formalizados é preciso enquadrar o homem neste espaço-tempo. Santos (2006) coloca o homem como um “ator” do espaço, o veículo de uma ação, o elemento que dá corporeidade a ação em um processo, porém apenas a partir de um novo conceito – o de evento - é possível encontrar uma maneira que formalize a ação do homem.

O mesmo autor define evento como “o resultado de um feixe de vetores, conduzido por um processo”. Um evento só é identificável quando se completa, quanto atinge um resultado. Este resultado ocorre em um lugar, o “depositário final, obrigatório, do evento”.

Eddington (apud Santos, 2006) coloca um evento como "um instante do tempo e um ponto do espaço".

Reunindo o resultado do processo colocado por Santos com a visão de Eddington chegamos a definição de um evento como um instante do tempo, um ponto do espaço e seu conteúdo informacional.

Une-se as idéias de Santos e Eddinton a visão de Hagerstrand em sua “Geografia do Tempo” quando apresenta a idéia de “mapear os tempos de uma realidade em movimento, através do artifício de "congelar" os eventos em padrões gráficos, de modo a que sejam analisados segundo seus respectivos conteúdos.” (HAGERSTRAND, 1973 apud SANTOS, 1997).

Desta maneira a ação do homem também ganha um caráter formal e torna-se representável computacionalmente, através dos eventos – pontos no espaço e tempo dotados de um conteúdo informacional.

A partir dos eventos foi atribuído ao homem e suas ações - que não são elementos espaciais e sim atores neste espaço – um novo conceito, o de elementos “espacializáveis”.

Elementos “espacializáveis” não possuem os mesmos aspectos dos elementos do espaço. Sendo o registro das relações do homem e dos elementos do espaço que o cerca.

### 3.5 O CONHECIMENTO

Representar computacionalmente o conhecimento é um desafio que tem motivado uma série de estudos na área de Inteligência Artificial ao longo de décadas. Diversas estruturas para a modelagem do conhecimento foram propostas na tentativa de obter uma maior capacidade de descrição e flexibilidade.

Uma destas estruturas tem se sobressaído a partir da metade da década de 90, ganhando espaço entre os sistemas de informação e buscando trazer conhecimento formalizado a estes, o uso de ontologias.

Segundo Gruber (1995) ontologia *“is an explicit specification of a conceptualization. The term is borrowed from philosophy, where Ontology is a systematic account of Existence. For AI systems, what “exists” is that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge”*.

Na abordagem que buscada neste trabalho o uso de ontologias para descrever o espaço e formas de representar este espaço e conhecimento parece ser um caminho natural.

Para isso, é importante compreender as formas de utilização de ontologias em conjunto com Sistemas de Informação, Guarino (1998) separou as ontologias em quatro diferentes tipos: ontologias de alto nível, ontologias de domínio, ontologias de tarefa e ontologias de aplicação.

Guarino (1998) também conceitua esses diferentes tipos de ontologias:

- Ontologias de alto nível descrevem conceitos gerais como espaço, tempo, matéria, objeto, evento, ação, etc., os quais são independentes de um problema ou domínio particular;

- Ontologias de domínio descrevem o vocabulário relacionado a um domínio genérico (como medicina, ou automóveis) especializando os termos presentes na ontologia de alto nível;
- Ontologias de tarefa descrevem uma atividade ou tarefa genérica (como diagnóstico ou venda) também a partir dos termos definidos na ontologia de alto nível;
- Ontologias de aplicação descrevem conceitos que dependem tanto de um domínio quanto de uma tarefa particular, sendo geralmente especializações de ambas as ontologias relacionadas.

A relação entre a arquitetura do Sistema Geográfico que se propõe e sua base no conhecimento representado, naturalmente passa pelos conceitos acima colocados, guiando a construção de conceitos organizados hierarquicamente – uma taxonomia – armazenada na estrutura de uma ontologia que servirá como fonte do conhecimento presente no sistema.

Por questões relativas ao escopo delimitado em razão do tempo, neste trabalho ao invés da utilização de ontologias completas, serão utilizadas ontologias simplificadas com conceitos relativos ao domínio que se deseja representar e a aplicação, visando a utilização de ontologias completas em trabalhos futuros.

### **3.6 A REPRESENTAÇÃO COMPUTACIONAL DOS ELEMENTOS DO ESPAÇO**

O componente temático da Cartografia é muitas vezes esquecido e mascarado por meio de paletas de cores e dégradés, que são selecionados aleatoriamente e aplicados sem critério.

A aplicação sem critério de cores sobre o desenho de geometrias mostradas em camadas de dados geográficos, trata apenas de um dos aspectos inerentes a representação (e aqui referimos apenas representações cartográficas) de elementos do espaço.

A OGC (Open Geospatial Consortium) – uma organização que busca estabelecer padrões abertos para tecnologias geoespaciais - possui um padrão para atender ao aspecto temático, o *Styled Layer Descriptor* (SLD). Apesar de esta ser uma alternativa muito boa para a descrição de camadas em mapas base, sua manutenção é difícil quando o objetivo é descrever diferentes

objetos geográficos e manter a possibilidade de fácil extensão dos tipos destes objetos. O ideal neste caso é que quem possua suas representações seja o próprio elemento espacial, tornando-o independente da descrição da camada e facilitando a extensibilidade.

Quando é utilizada uma abordagem baseada em estruturas que armazenam o conhecimento de um determinado domínio, é latente a necessidade da possibilidade de diferentes representações para um mesmo elemento.

Um mesmo elemento pode ser tratado de maneira diferente quando visto sob a perspectiva de domínios diferentes, ou seja, mesmo possuindo a mesma forma pode requerer uma representação diferente quando colocado em outro contexto.

Por exemplo, a representação de uma casa como um ponto é útil para analisar a densidade urbana em uma área, porém não é suficiente para um sistema que vise o cálculo da cobrança do IPTU – quando seria necessário o polígono do terreno.

Outro conceito chave é o caráter temporal das representações. Este conceito é necessário para representar processos dinâmicos e evolução de estruturas físicas ao longo do tempo. Elementos espaciais e espacializáveis podem surgir em um determinado momento e desaparecer em um momento posterior, podendo estar presentes no espaço ao longo de um período ou permanecendo registrados em um ponto da linha temporal como um instante.

Elementos espaciais também podem possuir estados distintos e sua representação pode variar de acordo com estes estados de maneira condicional (exemplos: uma loja com seus estados aberta/fechada, um hospital ativo/em reforma).

Este trabalho mantém os estilos de representação associados aos elementos espaciais e espacializáveis, possuindo funcionamento independente da camada em que estão colocados (como no caso do SLD).

## 4 PADRÕES E TECNOLOGIAS

Os padrões e tecnologias utilizados no desenvolvimento do protótipo a ser implementado para utilização como estudo de caso são apresentados nesta seção.

### 4.1 OPEN GEOSPATIAL CONSORTIUM (OGC)

Organização internacional que atua no desenvolvimento de padrões para serviços geoespaciais e de localização visando garantir a interoperabilidade entre diferentes sistemas que utilizam dados espaciais.

### 4.2 KEYHOLE MARKUP LANGUAGE (KML)

KML é uma linguagem XML desenvolvida inicialmente pela empresa Keyhole, comprada posteriormente pela Google e enviada em 2007 para o Open Geospatial Consortium, sendo em 2008, o KML 2.2 adotado pela OGC como uma de seus padrões abertos.

Tem grande poder de expressão suportando uma série de formas de descrição de dados geográficos, tanto 2D, quanto 3D. O padrão foi adotado pela OGC buscando atingir quatro objetivos:

*" \* That there be one international standard language for expressing geographic annotation and visualization on existing or future web-based online and mobile maps (2d) and earth browsers (3d).*

*\* That KML be aligned with international best practices and standards, thereby enabling greater uptake and interoperability of earth browser implementations.*

*\* That the OGC and Google will work collaboratively to ensure that the KML implementer community is properly engaged in the process and that the KML community is kept informed of progress and issues.*

*\* That the OGC process will be used to ensure proper life-cycle management of the KML Standard, including such issues as backwards compatibility. "*

Segundo o disponível em (OGC, 2009).

### **4.3 BANCO DE DADOS**

Banco de Dados é o resultado de técnicas de organizar um determinado conjunto de dados de forma que o seja possível o acesso a estes por parte de seus usuários.

Este acesso pode ser facilitado através da utilização de linguagens de consulta (ex: SQL), ter seu desempenho otimizado através do uso de técnicas de indexação.

Também existem implementações de extensões para os SGBDs que criam estruturas para novos tipos de dados (como por exemplo o tipo Geometry) e implementam funções (por exemplo Intersection ou Contains, operações geométricas) que podem ser utilizadas no controle destes.

#### **4.3.1 POSTGRESQL**

PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional de código aberto que tem sido bastante adotado entre as opções livres disponíveis, destacando-se pela robustez. É definido em seu site oficial (<http://www.postgresql.org/about/>) como:

“a powerful, open source object-relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness.”

### 4.3.2 POSTGIS

PostGIS é uma extensão de código aberto para o PostgreSQL que intenciona adicionar ao SGBD suporte a objetos geográficos implementando o padrão Simple Feature SQL proposto pelo Open Geospatial Consortium, suportando os diferentes dados geométricos simples e complexos:

- *Point* (Ponto)
- *Linestring* (Linha)
- *Polygon* (Polígono)
- *MultiPoint* (MultiPonto)
- *MultiLinestring* (MultiLinha)
- *MultiPolygon* (MultiPolígono)
- *GeometryCollections* (Coleções de Geometria)

Suportando também diferentes operadores geométricos e espaciais, entre os quais:

- *Union* (União)
- *Difference* (Diferença)
- *Intersects* (Intersecciona)
- *Touches* (Toca)
- *Crosses* (Cruza)

Além do suporte de operações de medidas. Estas e outras operações podem ser conferidas com maior detalhe na documentação do PostGIS (2010) disponível na web.

Outra possibilidade disponibilizada pela extensão é a criação de índices espaciais e o suporte a outros padrões da OGC como o Well-Known Binary (WKB) e o Well-Known Text (WKT) sendo esses, formatos binário e textual para representação de geometrias.

## 4.4 TECNOLOGIAS DE DESENVOLVIMENTO

Nesta seção são mostradas algumas das diferentes tecnologias, bibliotecas e frameworks utilizadas ao longo do desenvolvimento do protótipo servindo como estudo de caso para a arquitetura proposta neste.

### 4.4.1 GOOGLE WEB TOOLKIT (GWT)

O Google Web Toolkit é um framework da Google que tem como motivação facilitar a implementação de aplicações web. Sua estrutura divide a aplicação em duas partes básicas, a parte cliente que é executada no *browser* do usuário e a parte do servidor que fica em um servidor de aplicação Java. A grande facilidade trazida pelo GWT é a possibilidade de escrever código somente em Java, com a tradução do código para Javascript ficando a cargo do framework.

O GWT também fornece uma estrutura de comunicação RPC (Remote Procedure Calls) que abstrai a comunicação cliente-servidor, excluindo a necessidade (porém mantendo a possibilidade) de requisições HTTP diretas.

Esta estrutura funciona através da idéia de chamadas e respostas assíncronas, tendo uma interface síncrona e uma interface assíncrona, esta com um parâmetro a mais que permite especificar o que acontece no retorno da requisição efetuada no momento em que esta possuir uma resposta concluída no servidor estas interfaces possuem uma classe que as implementa no servidor. A arquitetura pode ser vista na Figura 4.

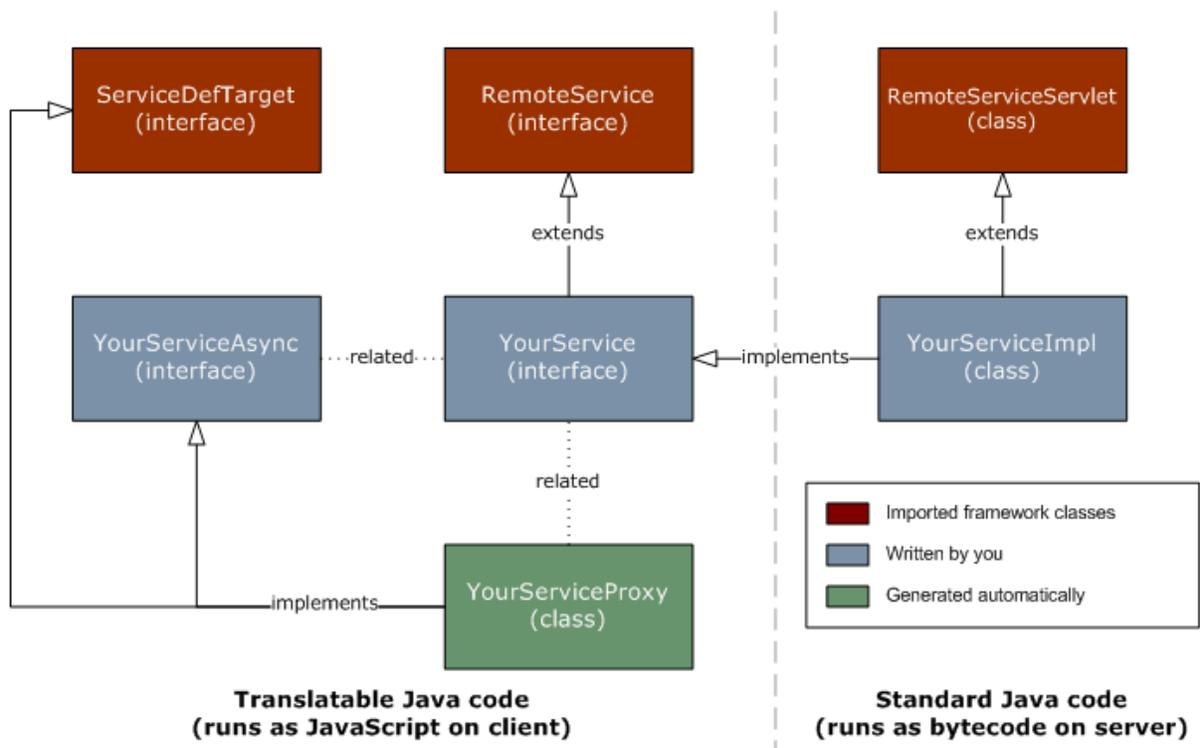


Figura 4: Arquitetura de Comunicação do GWT

Obtida em: <http://code.google.com/webtoolkit/doc/1.6/images/AnatomyOfServices.gif>

O framework permite também a possibilidade de escrever código diretamente Javascript através da utilização de métodos chamados de JSNI, *Javascript Native Interface*, com idéia semelhante a do JNI que permite a chamada de métodos de outras linguagens através do Java.

Suas limitações ficam por conta do suporte a determinados tipos de objetos, suportando na parte cliente classes mais comuns da API Java como os tipos básicos, *Arrays*, *Lists*, *Hashs* e *“POJOs”* que implementem a interface *Serializable* presente na API padrão do Java.

Entre outros frameworks para a criação de aplicações Web foi o escolhido para a implementação do protótipo deste trabalho por simplificar a comunicação e a estrutura do projeto, além de possibilitar o uso de outras bibliotecas Javascript desde que utilizando, criando ou estendendo *wrappers* (camadas que envolvem bibliotecas Javascript). A versão utilizada no desenvolvimento do projeto será a 2.0.3.

## 4.4.2 OPENLAYERS

O OpenLayers é uma API Javascript que tem a intenção de tornar fácil a exibição de mapas das mais diversas fontes de dados, conseguindo renderizar mapas em uma série de padrões, entre eles os padrões WMS e WFS da OGC até o ArcIMS da ESRI, como coloca a definição do próprio site da API (<http://www.openlayers.org>):

*“OpenLayers makes it easy to put a dynamic map in any web page. It can display map tiles and markers loaded from any source. Metacarta developed the initial version of OpenLayers and gave it to the public to further the use of geographic information of all kinds. OpenLayers is completely free, Open Source JavaScript, released under a BSD-Style License.”*

A API possui muitas das funções básicas necessárias para o seu objetivo de facilitar a colocação de mapas na web, a possibilidade de adicionar e utilizar alguns controles básicos, bem como implementações simples para gerir os *layers*, além de funções mais avançadas como a implementação de certas “estratégias” como cluster e formatadores para certos *schemas* XML.

Estas ferramentas junto a possibilidade de possuir uma camada vetorial e de criar features vetoriais permitem ao programador desenvolver aplicações de mapas bastante complexas.

Foi a tecnologia escolhida neste trabalho pela grande liberdade e compatibilidade que a biblioteca fornece além de a possibilidade de realizar necessárias adaptações no código da biblioteca. A versão utilizada no desenvolvimento do projeto será a 2.8.

### 4.4.3 EXT-JS

O ExtJS é uma biblioteca JavaScript que oferece entre outros recursos uma biblioteca de componentes visuais que permitem a criação de aplicações web com característica semelhante às aplicações desktop.

A biblioteca foi feita sob a licença LGPL - uma licença que deixava a biblioteca livre para uso comercial e distribuição em conjunto com softwares de código fechado. Porém a partir da versão 2.1 a Ext Foundation responsável por manter o projeto alterou a licença para GPL e passou a ter uma outra licença para uso comercial. A versão utilizada no desenvolvimento do projeto será a 2.0.2, ainda sob-licença LGPL.

### 4.4.4 EDITOR DE ONTOLOGIAS – PROTEGÉ

Um editor de ontologias é uma ferramenta de software utilizada para facilitar o desenvolvimento e manutenção de ontologias através de componentes visuais que abstraem a necessidade de um conhecimento prévio da estrutura do arquivo destino de uma determinada ontologia.

Para o desenvolvimento do estudo de caso mostrado na seção sete será utilizado o Protegé em sua versão 3.4.4. O *“Protegé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies.”* (PROTEGÉ,2010)

## 5 SISTEMA GEOGRÁFICO BASEADO EM CONHECIMENTO

Este trabalho propõe uma abordagem que compõe conceitos de representação de conhecimento e de Sistemas de Informação Geográfica.

O objetivo é facilitar a geração ou alteração de código de implementação, mediante a utilização de classes de elementos descritas em ontologias. Tais ontologias são usadas em uma camada intermediária de conhecimento que serve como interface entre a camada da aplicação e as fontes de dados, conforme ilustra a Figura 5.

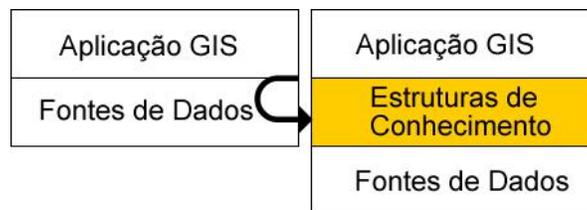


Figura 5: Adição de Camada de Estruturas de Conhecimento

Esta camada de Estruturas de Conhecimento inclui uma ontologia de aplicação (descrevendo os conceitos da aplicação), uma ontologia de domínio (descrevendo conceitos específicos de domínio) e mapeamentos entre conceitos dessas ontologias usados para descrever fontes de dados utilizadas e níveis de informação apresentados em uma aplicação SIG.

Desta forma a ontologia de aplicação é “genérica” ou seja, serve para qualquer domínio de aplicação, enquanto a ontologia de domínio é variável. Estas estruturas de conhecimento são carregadas pela aplicação que armazena as informações sobre estas classes em estruturas de dados permitindo de uma maneira semelhante à reflexão a recuperação da estrutura da classe (seu nome e atributos) posteriormente, permitindo que a aplicação conheça os elementos descritos na ontologia.

Através da utilização destas duas ontologias, esta arquitetura permite o isolamento dos elementos de domínio e seus atributos do código da aplicação que tem conhecimento prévio apenas dos conceitos presentes na ontologia de aplicação.

## **5.1 REPRESENTANDO O ESPAÇO, A APLICAÇÃO E OS ELEMENTOS DE DOMÍNIO**

Para representar o espaço é preciso ir além da descrição dos seus elementos, também definindo como se dá sua organização no espaço (SANTOS, 2008).

Nesta seção são mostrados os esboços das duas diferentes ontologias anteriormente referenciadas e um elemento espacial de uma ontologia voltada ao domínio: (i) o esboço de uma ontologia de alto-nível que contém os conceitos geográficos utilizados para descrição do espaço; (ii) uma ontologia que descreve os conceitos conhecidos pelo código da aplicação; e (iii) a classe de um elemento de domínio– grupo de classificação – presente na ontologia com conceitos de domínio.

Por exemplo, “*Providence Medford Medical Center*” é uma instância da classe Hospital e contém a descrição do elemento - conforme os conceitos das ontologias de domínio e aplicação – sendo então possível representá-lo como um elemento espacial na aplicação.

### **5.1.1 ONTOLOGIA PARA DESCRIÇÃO DO ESPAÇO**

Uma ontologia de alto nível busca descrever conceitos gerais que sejam aplicados a classes de diversos outros domínios (GUARINO, 1997). A ontologia utilizada neste trabalho para descrever uma organização base para o espaço tem uma idéia semelhante a da ontologia de alto nível, apresentada na Figura 6. Esta ontologia foi criada com a intenção de descrever uma idéia de organização do espaço tendo como sustentação as idéias de Santos (2008) e Harvey (1969) para a criação e hierarquia dos diferentes conceitos.

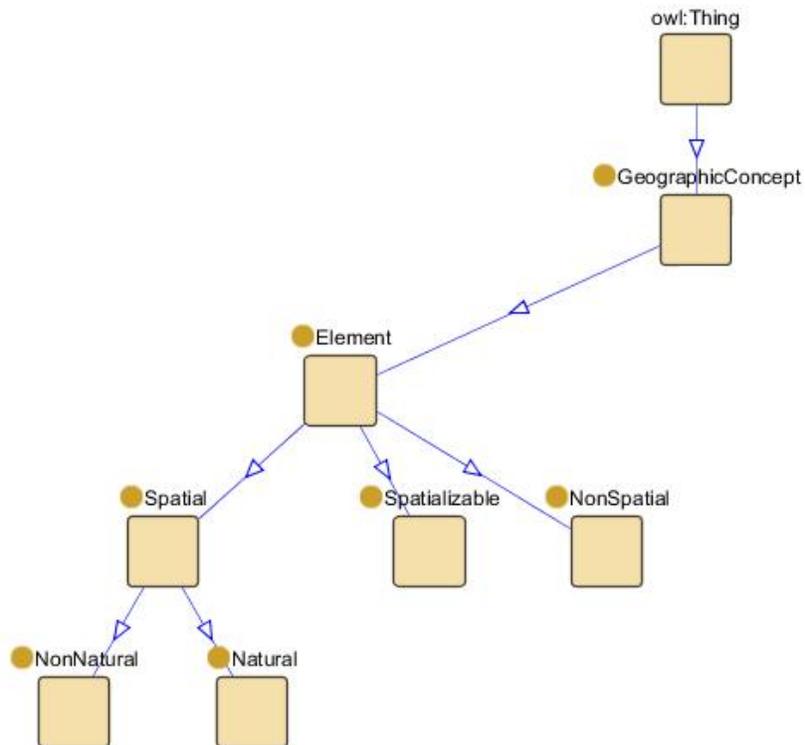


Figura 6: Ontologia para descrição do Espaço

A raiz da ontologia é o nodo *GeographicConcept*, e suas subclasses são conceitos geográficos aplicáveis para a descrição e organização do espaço citados na seção três: (i) elemento; (ii) elemento espacial; (iii) elemento “espacializável”; e (iv) elemento não-espacial.

A classe *Element* é pai de todos os futuros elementos a serem conhecidos pelo sistema e está dividida em três diferentes tipos de elementos:

1. espaciais (*Spatial*) – elementos que possuem uma localização no espaço (ex: um Hospital, uma Casa...);
2. “espacializáveis” (*Spatializable*) – elementos que não possuem uma localização mas podem ser associados diretamente a uma ou mais localizações no espaço (ex: um Caso de Doença, pode ser mostrado no espaço no hospital onde foi diagnosticado, na casa da pessoa doente, no lugar onde a doença foi contraída, etc.); e

3. não-espaciais (*NonSpatial*) – elementos que não podem ser associados diretamente a uma localização no espaço (leis naturais, taxonomias de elementos não-espaciais...).

Elementos espaciais são aqueles que em uma análise do mundo real baseada na proposta de Santos (2008) estão relacionados aos quatro aspectos, possuem forma, função e são parte das estruturas e processos presentes no espaço. As estruturas descrevem como estes elementos estão relacionados e através da função materializada que possuem, estes são parte dos processos.

Elementos “espacializáveis” e não-espaciais não possuem obrigatoriamente forma e função, porém podem também fazer parte das estruturas e processos estão relacionados a elementos espaciais.

A forma dos elementos espaciais no contexto aqui colocado guarda as possíveis representações computacionais do elemento (a união entre uma geometria e um estilo).

### **5.1.2 ONTOLOGIA PARA DESCRIÇÃO DA APLICAÇÃO**

A ontologia com conceitos de aplicação – mostrada na Figura 7 - é utilizada para descrever conceitos conhecidos pelo código da aplicação e que serão utilizados para descrever os elementos do domínio que se deseja representar.

A raiz é o *ApplicationConcept* e derivados dele estão: (i) conceitos simples (conceitos indivisíveis) como geometria (*Geometry*) que faz referência a uma geometria projetada em um plano, estilo (*Style*) que faz referência a um estilo CSS que através de suas propriedades formula um estilo visual e unidade de medida (*MeasureUnit*); (ii) conceitos híbridos como atributo (*Attribute*) que possui uma subclasse *SimpleAttribute* que é associada a um atributo com um único valor ou o conceito composto *MeasureAttribute* que além do valor possui uma unidade de medida; e (iii) conceito composto, como no caso da representação que é formada por uma ou mais geometrias, com um estilo e com um período durante o qual aquela representação é válida.

Através da associação destes conceitos na descrição dos elementos de domínio é possível fazer com que a aplicação compreenda e possa trabalhar de maneira genérica com os elementos de domínio sem a necessidade de possuir em seu código conhecimento destes elementos.

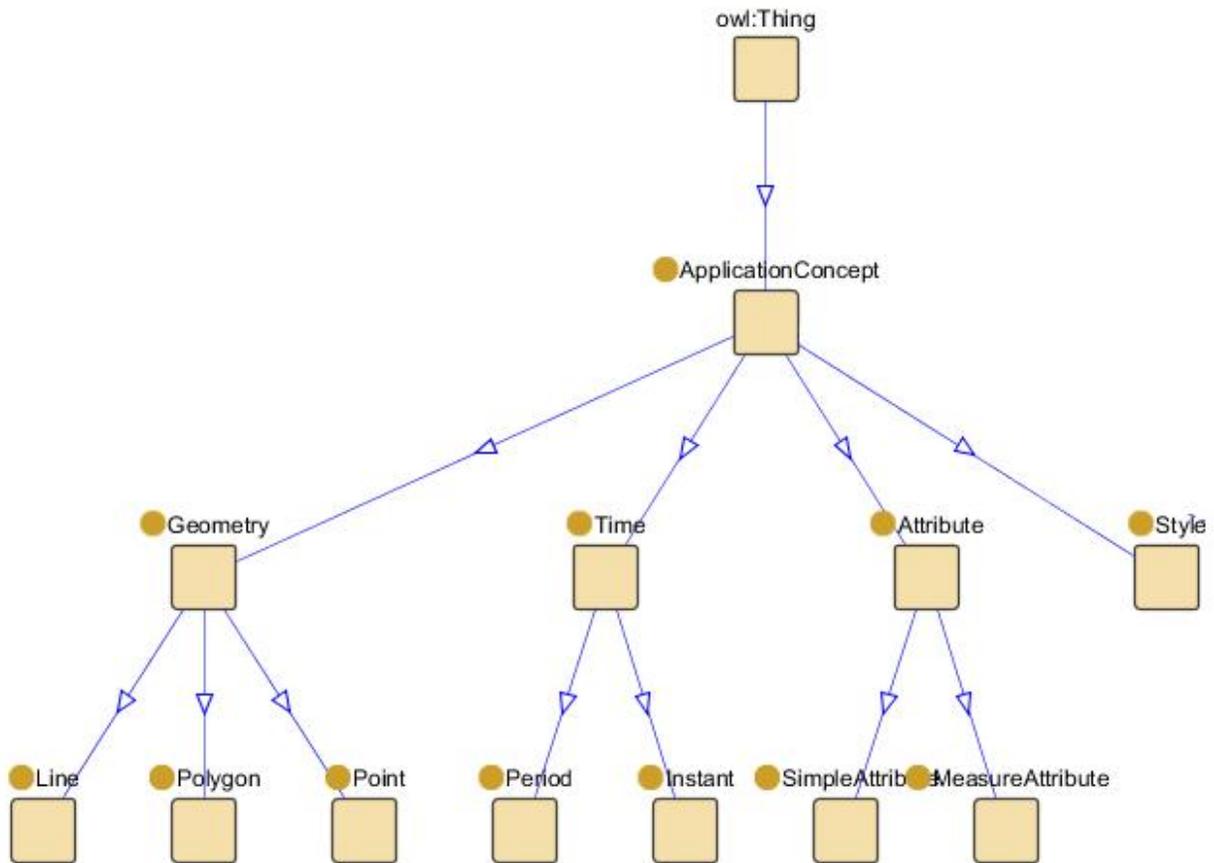


Figura 7: Ontologia para a descrição dos conceitos de Aplicação

### 5.1.3 ONTOLOGIA DE DOMÍNIO

A ontologia de domínio apresenta os conceitos referentes a um domínio específico (GUARINO, 1997). A idéia da ontologia de domínio utilizada neste trabalho é semelhante àquela proposta por Guarino, classificando e descrevendo os conceitos do domínio que se deseja utilizar, como o domínio da saúde, planejamento urbano, segurança entre outros.

Para a sua construção, deve-se contar com o conhecimento de um especialista de domínio que seja capaz de definir quais elementos e quais atributos destes elementos serão importantes para as análises futuramente realizadas.

Os elementos desta ontologia são as unidades básicas destes domínios, como um Hospital, um centro de Distribuição de Vacinas, Laboratórios que produzem vacinas, etc. Estes elementos são subclasses de um dos três tipos de elemento descritos na ontologia da organização do espaço. E além das classes concretas e instanciáveis (como Hospital) existem superclasses que visam apenas uma classificação/organização hierárquica do espaço (como Construção).

As relações entre os conceitos presentes na ontologia de domínio com a ontologia de aplicação apresentam uma descrição deste elemento de acordo com os atributos que possui. O exemplo demonstrado na Figura 8, de uma possível descrição o Elemento Hospital, ajuda a compreender melhor estas relações.

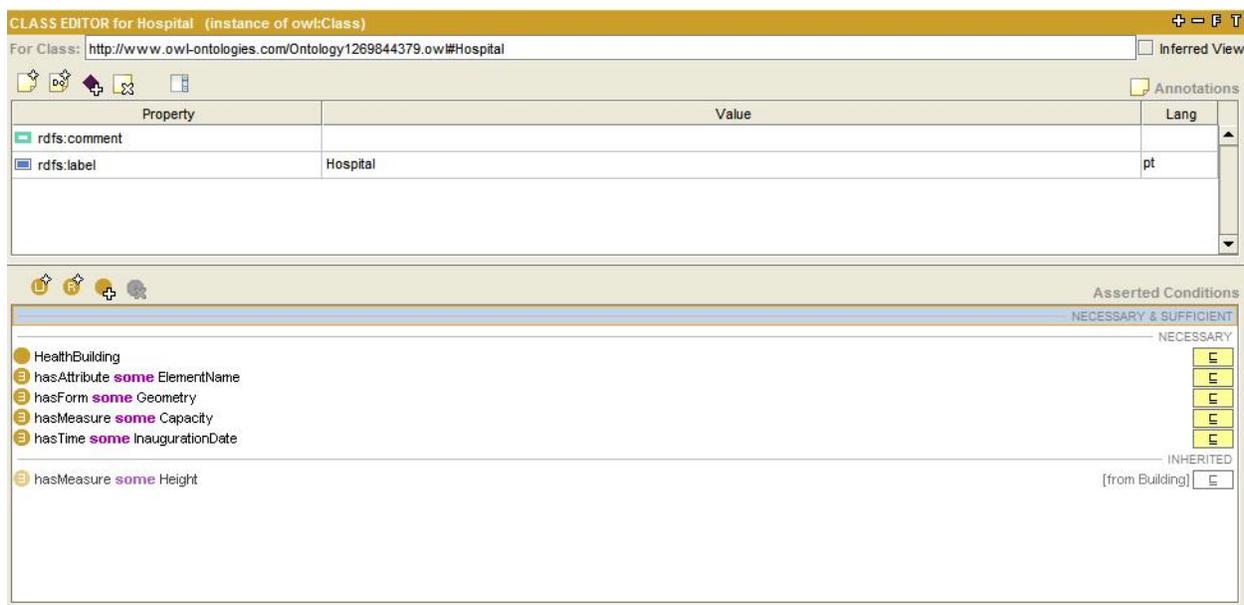


Figura 8: Elemento Hospital

Na Figura 8 é demonstrado o elemento Hospital, que é uma subclasse da classe Construção (*Building*) (que em sequência é filha da hierarquia *Element->Spatial->NonNatural-*

>*Building*). Como subclasse ele herda o atributo altura (*Height*) que é uma das subclasses do conceito *MeasureAttribute* na ontologia de aplicação e também possui seus diferentes tipos de relações (*hasAttribute*, *hasMeasure*, *hasTime*) com subclasses destes outros conceitos.

Com o elemento de domínio descrito desta maneira, a aplicação não precisa ter conhecimento acerca das características desse elemento em código fonte para trabalhar com ele posto que a aplicação conhece todos os tipos e nomes dos atributos de um elemento.

Desta forma a relação da aplicação com qualquer elemento de domínio ocorre de maneira genérica e transparente. São indiferentes para aplicação os valores ou atributos do elemento de domínio deste que este elemento respeite a estrutura que a aplicação é capaz de compreender, no caso os tipos de atributo descritos na ontologia de aplicação.

A hierarquia vista em *Building*->*Hospital* é utilizada para oferecer diferentes formas de filtragem e seleção de camadas de informação. Em determinado contexto, por exemplo, pode ser interessante filtrar todas as construções que tenham mais de 10 metros de altura, ou então exibir todos os tipos de construções com exceção de hospitais.

Propriedades também podem ser adicionadas à classe, como por exemplo a propriedade *label* que possui um rótulo a ser exibido como nome para aquela classe. É possível ainda associar esta propriedade a um idioma (coluna *Lang*) e criar rótulos para diferentes idiomas facilitando o processo de internacionalização da aplicação.

## 5.2 ARQUITETURA PROPOSTA

O fluxo de funcionamento pelos diferentes módulos presentes na arquitetura tem seu início na especificação da ontologia que serve como base ao sistema, uma vez especificada ao iniciar o sistema. Esta ontologia é lida pelo componente indexador, responsável por ler e retornar o conhecimento descrito na ontologia e transformá-lo em estruturas de dados pré-determinadas que possuam a capacidade de descrever os elementos do espaço e suas relações, criando uma descrição de meta-classes nestas estruturas de dados.

A comunicação entre a estrutura de meta-classes e as fontes de dados não ocorre de forma direta, havendo a necessidade de estruturas de mapeamento capazes de estabelecer relações entre a estrutura de dados e os dados propriamente ditos, integrando diferentes fontes. Estas relações são estabelecidas relacionando os termos descritivos presentes na ontologia de domínio com as entidades presentes nas fontes de dados, sendo os tipos destes dados definidos pelos conceitos presentes na ontologia de aplicação, conforme a Figura 9.

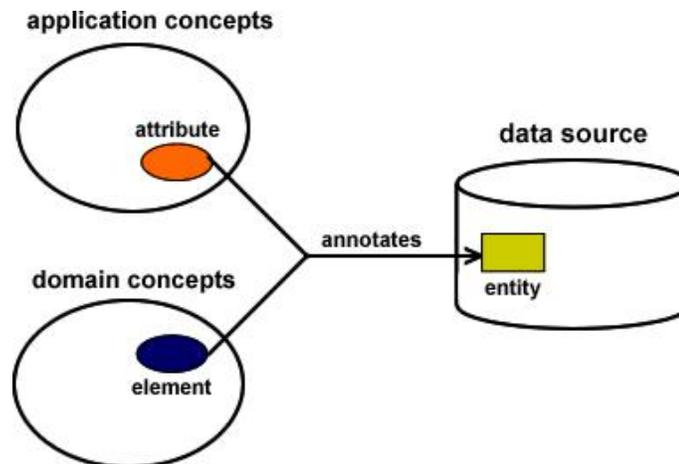


Figura 9: Anotações das fontes de dados com base nos conceitos de aplicação e domínio

Através do conhecimento da estrutura das classes presentes na ontologia que descreve o domínio e das anotações realizadas e consolidadas em um mapeamento de dados, a aplicação tem a capacidade de representar os dados presentes em diferentes fontes de dados, possibilitando a interação com estes através de componentes de interfaces como se fossem elementos de uma mesma fonte e possuindo a estrutura hierárquica descrita na ontologia.

A arquitetura proposta é apresentada na Figura 10.

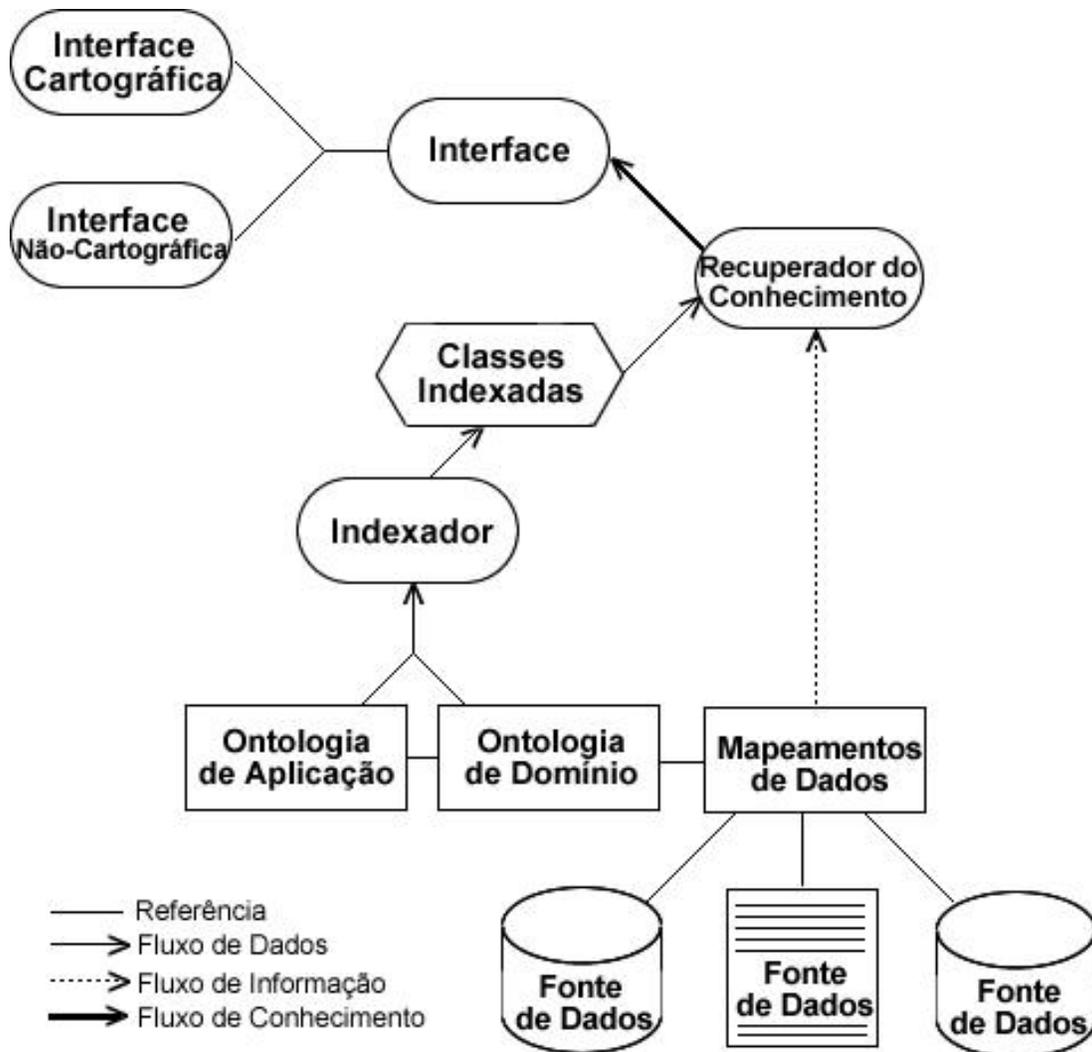


Figura 10: Arquitetura proposta

### 5.2.1 ONTOLOGIAS

As ontologias possuem papel base na arquitetura. A ontologia de alto nível organiza o espaço de acordo com uma hierarquia levando em conta os três diferentes tipos de elementos (que a aplicação conhece previamente); a ontologia com conceitos de aplicação contém os conceitos que o código da aplicação conhece (como os tipos de atributo); e por fim, a ontologia de domínio possui os elementos de domínio organizados de maneira hierárquica seguindo a organização dos tipos de elementos da ontologia de alto nível.

Através da relação entre os elementos descritos na ontologia de domínio com os conceitos presentes na ontologia de aplicação estes passam a ser compreensíveis para a aplicação.

### **5.2.2 INDEXADOR E CLASSES INDEXADAS**

Ler arquivos seguidamente é uma tarefa que apresenta um custo alto de tempo, sobretudo em uma arquitetura cliente-servidor quando o arquivo encontra-se localizado no servidor, o que tornaria quase inviável a leitura direta em um arquivo OWL (estrutura onde estão armazenadas as ontologias acima referidas). Em todas as vezes em que fosse necessário buscar alguma informação sobre os elementos do domínio ou seus atributos, como por exemplo, na seleção do elemento que se deseja carregar.

A solução aqui proposta para este problema é a criação de um módulo Indexador. Este módulo é o responsável por, no momento em que a aplicação é iniciada, percorrer todas as classes presentes no arquivo onde a ontologia está armazenada (um arquivo em formato OWL, por exemplo) para então ler e armazenar tudo o que for útil posteriormente para a aplicação (a hierarquia, o label, os atributos de uma classe) em uma estrutura de dados que contém as classes indexadas.

A idéia é semelhante ao conceito de reflexão presente nas linguagens de programação que permite ter acesso às diferentes estruturas de uma classe em tempo de execução, obtendo o nome daquela classe, seus atributos, métodos, etc.

A estrutura de dados que possui as classes indexadas fica então disponível para o acesso pela aplicação em sua camada cliente, sem a necessidade de efetuar novas leituras no arquivo ou requisições ao servidor. É aconselhável que a montagem desta estrutura de dados utilize como base estruturas de dados que possuam rápido acesso aos dados e uma chave pela qual os elementos possam ser retornados.

### 5.2.3 FONTES DE DADOS E MAPEAMENTOS

Fontes de dados com elementos espaciais encontram-se geralmente espalhadas em diferentes formatos e estruturas de acordo com as diferentes organizações que mantêm esses dados (no Brasil algumas destas organizações que podem ser citadas como exemplo são o IBGE, prefeituras, ministério do meio-ambiente, etc.), podendo variar no nível de detalhe (uma fonte ter mais atributos que a outra, por exemplo) ou no formato em que é armazenado (em uma das fontes pode estar em uma tabela em uma base de dados, em outra em um arquivo KML, por exemplo).

Para um SIG comum as fontes de dados seriam reproduzidas diretamente em diferentes camadas como se fossem unidades diferentes. Porém um dos objetivos deste trabalho é a integração destes dados de acordo com uma estrutura de conhecimento que contenha a descrição do elemento de domínio que se deseja representar. Então, por exemplo, dados de hospitais mesmo que vindo de fontes diferentes representam um elemento do mesmo tipo e fazem parte de uma estrutura onde estão organizados de forma que é possível encontrá-los em uma visão hierárquica do espaço então devem ser tratados como tal pelo sistema.

Como alternativa para resolução desta questão é necessário a criação de mapeamentos de dados que contenham as relações entre os atributos colocados no elemento de domínio presente na ontologia e os atributos nas fontes de dados. Não é parte do escopo deste trabalho definir maneiras complexas para mapeamentos de dados de fontes heterogêneas, porém para a implementação da proposta foram colocados dados em uma tabela em um SGBD e em um arquivo KML e o mapeamento foi realizado de maneira manual e armazenado em um arquivo que contém as relações mostradas nas Figura 11 e 12.

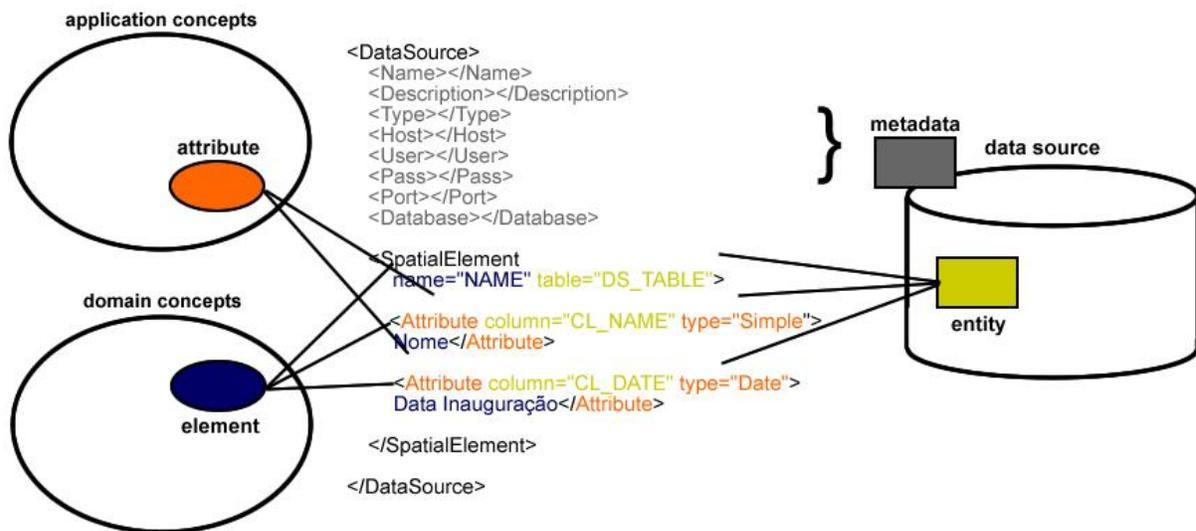


Figura 11: Mapeamento das entidades nas fontes de dados armazenadas em XML

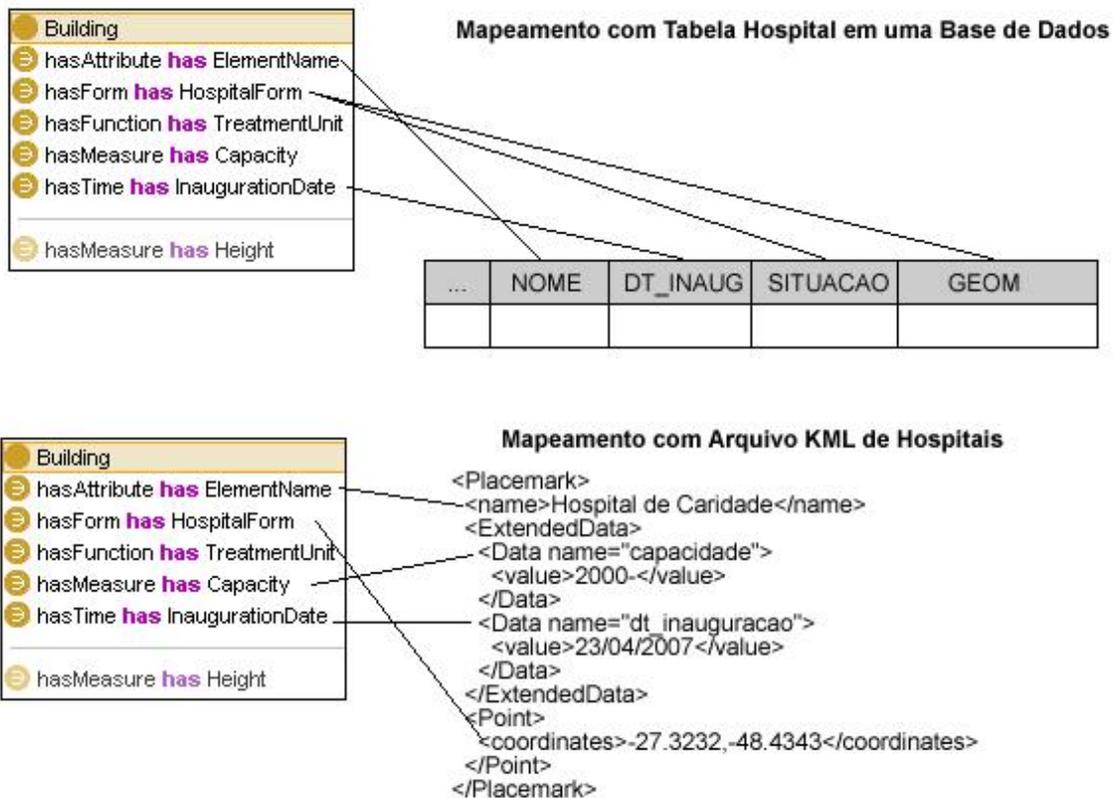


Figura 12: Exemplo de Mapeamento de Dados

Os mapeamentos guardam associações entre os atributos do elemento vindo da ontologia com conceitos e domínio e os atributos dos elementos na fonte de dados alvo, além das informações necessárias para acesso a fonte (localização do arquivo ou usuário, senha, host, no caso de um Banco de Dados).

Como pode ser notado na Figura 12, em alguns casos existe a relação direta entre o conceito na ontologia e na fonte de dados como no caso do *ElementName* com a tag *Name* no KML e com a coluna Nome no BD. Em outros casos a relação não é direta como por exemplo a forma, que pode ser ligada a uma geometria que será utilizada com uma representação genérica ou a uma geometria e uma situação (Ativo, Inativo, Em Reforma...) que define qual representação é utilizada. O conhecimento acerca do funcionamento dos mapeamentos está presente no código da aplicação.

#### **5.2.4 RECUPERAÇÃO DO CONHECIMENTO**

O módulo responsável pela recuperação do conhecimento reúne os dados sobre as estruturas das classes presentes nas classes indexadas com os dados das instâncias destas classes presentes nas fontes de dados, retornando então os elementos conhecidos para a interface.

Para executar tal tarefa, o módulo necessita conter em sua implementação o conhecimento de como recuperar dados dos diferentes tipos de fonte. Além disso precisa ser capaz de interpretar as relações colocadas nos mapeamentos (como as demonstradas na seção anterior) e o acesso as classes indexadas que descrevem os elementos que o sistema conhece conforme descrito na ontologia.

#### **5.2.5 INTERFACE CARTOGRÁFICA E NÃO-CARTOGRÁFICA**

A interface fica separada em duas partes: (i) a interface cartográfica que conterá os mapas e seus controles e (ii) a interface não-cartográfica que permite a interação com as estruturas do conhecimento armazenadas em ontologias.

Na interface cartográfica, o mapa serve como um *container* das informações mostrando as localizações geográficas dos elementos e sua disposição sobre camadas base (com os contornos dos municípios, arruamentos, etc.).

Segundo Dent (1996) é inquestionável o fato de que se consegue obter informações em mapas e que mapas também possuem importância na divulgação de suas descobertas para outros e são uma forma fácil de encontrar rapidamente padrões espaciais. E um SIG coloca o mapa em uma perspectiva mais dinâmica permitindo a interação do usuário no controle deste e das camadas de informação que podem ser representadas sobre o mapa.

Em um sistema geográfico baseado em conhecimento, a forma como está estruturada a interface não-cartográfica também possui um papel muito importante na interação com o usuário. Ela deve permitir ao usuário a visualização presente nas estruturas de conhecimento como as descrições dos tipos de objetos, permitir filtragens de acordo com os atributos presentes nestas estruturas, ter visualizações alternativas (textual, gráfica, etc.) para dados não-espaciais e conseguir refletir a interação com as camadas de informação de uma maneira hierárquica. Desta maneira, exibindo a organização hierárquica presente na ontologia através de uma visão da estrutura dos elementos do domínio, ao contrário dos SIGs comum que permitem a seleção apenas de camadas de dados. Esta situação é demonstrada na Figura 13.



Figura 13: Modelos de interação com Camadas de Informação

## 6 IMPLEMENTAÇÃO

Com a arquitetura proposta e seus componentes descritos, neste capítulo é descrita a implementação de um protótipo que segue o proposto até aqui, utilizando os padrões e tecnologias descritos no Capítulo 4 e a arquitetura descrita no Capítulo 5.

Com o protótipo implementado, são realizadas experimentações com a criação de uma ontologia que inclua conceitos de domínio, descrição dos atributos de seus elementos de domínio de acordo com os conceitos presentes na ontologia voltada para a aplicação. E a criação dos mapeamentos com pelo menos duas diferentes fontes de dados para as camadas de informação, estes dados mapeados serão usados em conjunto com uma terceira fonte com mapas base. Os experimentos são realizados com a intenção de efetuar a validação da arquitetura proposta.

### 6.1 CARACTERÍSTICAS DO PROTÓTIPO

O protótipo implementado segue a idéia de um *WebGIS*, possuindo uma arquitetura cliente-servidor acessível através de um navegador Web via internet ou intranet. Seguindo o padrão de uma aplicação *Web AJAX* que permite uma interação contínua com a interface através de componentes de interface (janelas, botões, barras de ferramentas, etc.). Esta abordagem surgiu em contraponto a navegação em fluxos bastante comum em páginas *Web* (interação permanecendo bloqueada quando um fluxo é requisitado, como exemplo no clique de um link que efetuará uma mudança de página).

A parte cliente foi desenvolvida em linguagem *Javascript* com a utilização das APIs *OpenLayers* para exibição de mapas na *Web* e *ExtJS* para a criação de componentes de interface, que utiliza também componentes do HTML e CSS. A parte cliente é utilizada através de um navegador Web que seja capaz de interpretar *Javascript*.

A parte servidor foi desenvolvida em linguagem Java e pode ser executada em um container Web (no caso, o *Tomcat*).

Para otimizar o processo de desenvolvimento, foi utilizado o framework Google Web Toolkit que possui um compilador responsável pela transformação de código Java em código Javascript. Deste framework também é aproveitada a estrutura de comunicação entre cliente e servidor baseada em chamadas de procedimentos remotos (RPC).

Como características funcionais o protótipo deve permitir a visualização e controle do mapa pelo usuário que possui opções de interação com o mapa como *pan*, *zoom* e *select*.

Deve permitir ao usuário trabalhar com as informações dispostas de maneira hierárquica e não em camadas permitindo a seleção de acordo com as diferentes classes de elementos de domínio.

Quanto aos dados, o protótipo deve ser capaz de acessar fontes de dados em uma base de dados PostgreSQL e em um arquivo KML e também suporta a exibição de camadas de mapa base comerciais (como as do *Google Maps*, *Bing Maps* e *Yahoo Maps*) através das APIs destes.

O protótipo permite ao usuário interagir com os elementos presentes no mapa e com o conhecimento acerca destes elementos presente nas ontologias.

## **6.2 DEFINIÇÃO DA ESTRUTURA DE DADOS DAS CLASSES INDEXADAS**

A estrutura de dados implementada para suportar a estrutura de classes indexadas é formada por uma estrutura composta de *POJOs* com a utilização de tipos de dados comuns a API padrão do Java como *String* e *int* e para guardar listas de dados utilizando estruturas de *HashMap*.

O modelo possui uma estrutura simplificada de acordo com a necessidade do que se deseja armazenar atualmente, a hierarquia de classes e suas propriedades. Duas classes *OWLClass* e *OWLProperty* mantêm os dados recuperados das estruturas de ontologia.

As classes são serializáveis (implementam a interface *Serializable*) para que a transferência delas entre o cliente e o servidor seja possível.

## 6.3 IMPLEMENTAÇÃO DO INDEXADOR

Este módulo foi implementado com base na API do Protegé (editor de ontologias) para efetuar a leitura de arquivos OWL.

O módulo percorre toda a estrutura presente no arquivo OWL, recupera os nomes das classes e suas relações com outras classes além de propriedades como os labels e os armazena na estrutura de dados apresentada na seção anterior, disponibilizando assim o acesso as descrições da ontologia sem a necessidade de constantes acessos ao arquivo OWL.

O algoritmo do indexador é um algoritmo recursivo que percorre uma estrutura de árvore com a busca em profundidade e indexa as classes presentes na ontologia contida na estrutura de ontologia, sua descrição em cinco passos:

- 1) Recebe como parâmetros um *OWLNamedClass* (classe Java com o modelo da classe conforme descrita na estrutura de ontologia) e uma *OWLClass* da classe pai (no caso do primeiro elemento é passado com o pai sem valor);

- 2) Instancia um objeto da classe *OWLClass* para armazenar os dados relativos a classe na estrutura de ontologia;

- 3) Itera por todas as propriedades daquela classe, armazenando essas propriedades na estrutura *OWLProperty* e adicionando-a ao *HashMap* de propriedades no objeto instanciado no passo anterior junto com a chave que é o nome da propriedade.

- 4) Se o pai não for sem valor (caso apenas do primeiro elemento da árvore), todas as propriedades do pai são adicionadas ao filho.

- 5) Itera pelas subclasses da classe atual (presentes na estrutura de ontologia), chamando recursivamente o método para cada subclasse até terminar as subclasses encontradas adicionando-as no *HashMap* de subclasses da classe pai.

Os passos descritos acima podem ser visualizados no algoritmo em pseudocódigo:

```

INDEXCLASS(OWLNamedClass owlClass, OWLClass parent)
  OWLClass indexedClass = NEW OWLClass(NAME(owlClass), parent)
  FOR EACH (PROPERTIES(owlClass))
    OWLProperty indexedProperty =
      NEW OWLProperty(NAME(property), VALUE(property))
    ADD_PROPERTY(indexedClass, indexedProperty)
  IF NOT_NULL(parent)
    ADD_PROPERTIES(indexedClass, PROPERTIES(parent))
  FOR EACH SUBCLASSES(owlClass)
    OWLNamedClass owlClassToIndex = owlSubClass
    OWLClass toIndexClass = INDEXCLASS(owlClassToIndex, indexedClass)
    ADD_SUBCLASS(indexedClass, toIndexClass)
  RETURN indexedClass

```

## 6.4 MAPEAMENTO DOS DADOS

O modelo utilizado para o mapeamento dos dados foi mostrado nas Figuras 11 e 12. A intenção dos mapeamentos é utilizar o conhecimento da descrição dos elementos do espaço presente nas ontologias para anotar as diferentes fontes de dados, permitindo que o sistema possua uma visão integrada das fontes de dados.

Esta visão integrada dos dados é acessada pelo sistema através do seu conhecimento da descrição dos elementos presentes nas ontologias.

O modelo do arquivo XML para anotação das fontes de dados é demonstrado abaixo, onde:

- *Name, Description, Type, Host, User, Pass* e *Port* são metadados, relativos à fonte de dados, que descrevem qual é o seu tipo (Base de Dados, KML..) e os dados necessários para acessá-la;
- *SpatialElement* é o elemento presente na ontologia, seu atributo *name* é o nome do elemento conforme colocado na ontologia. O atributo *table* serve para referenciar uma determinada tabela, no caso do acesso a uma tabela de *features*;
- *Attribute* é o atributo do elemento, podendo ser de diferentes tipos (*Simple, Measure, Time..*) de atributos reconhecidos pelo sistema e presentes na ontologia de aplicação, o

tipo é especificado no atributo *type*. Seu valor é o nome do atributo de acordo com o colocado na ontologia de domínio e seu atributo *column* é a referência a qual coluna ou atributo ele corresponde na fonte de dados.

```
<Source>
<DataSource>
  <Name> </Name>
  <Description> </Description>
  <Type> </Type>
  <Host></Host>
  <User></User>
  <Pass> </Pass>
  <Port></Port>
  <Database> </Database>
  <SpatialElement name="NOME_ELEMENTO" table="TABELA_ELEMENTO">
    <Attribute column="NOME_FISICO_ATRIBUTO"
      type="TIPO_ATRIBUTO"> NOME_ATRIBUTO</Attribute>
  </SpatialElement>
</DataSource>
</Source>
```

## 6.5 MÓDULO PARA RECUPERAÇÃO DO CONHECIMENTO

O módulo de recuperação do conhecimento é o módulo responsável pela integração entre a camada de conhecimento da aplicação, representada na ontologia de Domínio, com a camada das fontes de dados.

Para estabelecer a relação entre os elementos conhecidos presentes na ontologia de domínio e os dados presentes na fonte (com sua própria organização), este módulo utiliza os mapeamentos de dados apresentados (modelo apresentado na seção anterior) em conjunto

com a ontologia de aplicação que detém o conhecimento dos tipos de dados que a aplicação é capaz de compreender.

Este módulo tem a função de ler os mapeamentos, compará-los com as classes indexadas (com os dados sobre os elementos conhecidos provenientes da ontologia de domínio) de modo a relacionar a estrutura conhecida do elemento e apontar dados acerca de quais atributos ali descritos estão presentes nas diferentes fontes de dados mapeadas.

Após a leitura dos mapeamentos e com as relações entre elementos conhecidos e elementos na fonte de dados definidas, este módulo é capaz de acessar a fonte de dado de acordo com o tipo dessa fonte, especificado no mapeamento e recuperar esses dados alocando em estruturas de dados criadas de acordo com os diferentes tipos de dados suportados pela aplicação (e descritos na ontologia de aplicação), deixando esses dados disponíveis para a visualização na interface.

## **6.6 TIPOS DE CAMADAS NO MAPA**

As camadas no mapa utilizadas neste trabalho são classificadas em dois tipos: (i) camadas base e (ii) camadas de informação.

Camadas base são camadas de mapas que permitem ao usuário a interação apenas com as imagens ali representadas através da utilização de funcionalidades como mover o mapa e aplicar zoom, não sendo possível a seleção dos elementos ali renderizados.

São utilizadas como um "background" para as camadas de informação. Como nelas são renderizadas apenas imagens e não os desenhos de cada um dos elementos presentes de maneira individual. É necessário um recurso muito menor do navegador e como recursos em navegadores são limitados a renderização de apenas uma imagem garante uma melhor performance.

Neste trabalho são utilizadas camadas base comerciais (no caso as camadas do Google Maps). O protocolo WMS também retorna como resultado de uma requisição uma camada base com os dados renderizados em formato de imagem.

Camadas de informação são camadas que intencionam colocar elementos individuais (hospitais, estradas, ruas..) para que a interação com estes seja com a unidade deste elemento, permitindo sua seleção, visualização de seus atributos entre outras formas de interação.

Geralmente são organizados como camadas "planas" que espelham o conteúdo da fonte de informação (como por exemplo bases de dados ou arquivos KML). Este trabalho apresenta uma utilização destas camadas de informação com uma organização hierárquica que permite a interação com as informações também através de sua hierarquia, conforme descrita em uma estrutura de conceitos.

## **6.7 CRIAÇÃO DA INTERFACE**

Para criação da interface com o usuário foram levados em conta alguns conceitos de usabilidade – como o foco no Mapa e a separação do local onde estão as ferramentas para interação com o mapa e com as camadas de informação.

A interface é minimalista e simplificada, possuindo o mapa como foco de sua exibição, com barras de ferramentas no topo e na parte de baixo conforme o esquema da Figura 14.

A barra de ferramentas presente no topo contém as interações com camadas do mapa e camadas de informação enquanto a barra de ferramentas presente abaixo do mapa possui os controles para interação com o mapa.

A manipulação de camadas e a visualização detalhada de informações presentes no mapa é feita através de janelas flutuantes que podem ser diminuídas e movimentadas como demonstrado na Figura 15.

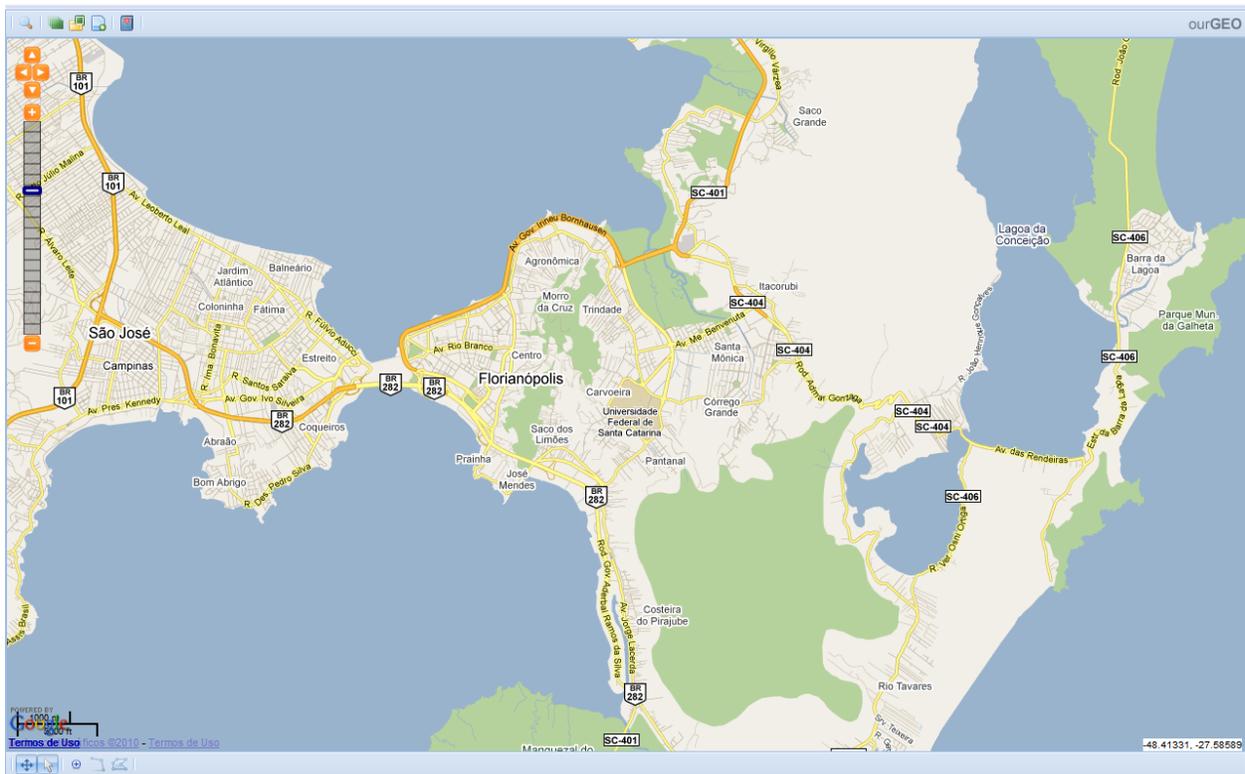


Figura 14: Interface inicial do protótipo– Camada base do Google Maps

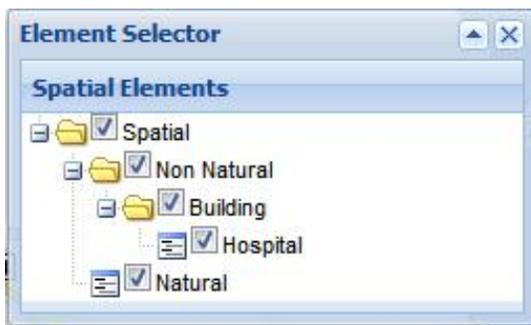


Figura 15: Seleção de elementos para interação com camadas de informação

### 6.7.1 ESTILO DOS ELEMENTOS

Para definição do estilo dos elementos a alternativa escolhida para deixá-los fora do código da aplicação foi estabelecer relações entre propriedades do CSS e propriedades de estilo utilizadas para a criação de *features* vetoriais para a renderização usando SVG pela API utilizada.

No momento de criação das *features* a classe responsável busca no CSS da página uma regra que possua o mesmo nome que o nome do elemento e realiza uma conversão entre as propriedades. As propriedades suportadas com sua respectiva conversão do CSS para o formato da API foram:

- *background-color* para *fillColor*;
- *opacity* para *fillOpacity*;
- *max-width* para *graphicWidth*;
- *max-height* para *graphicHeight*;
- *width* e *height* para *pointRadius*;
- *border-color* para *strokeColor*;
- *background-image* para *externalGraphic*; e
- *border-width* para *stroke-width*.

Dessa forma a definição do estilo dos elementos mostrados no mapa depende apenas do que está no CSS sendo facilmente editável e bastante flexível.

## 6.8 PROTÓTIPO IMPLEMENTADO

A carga de informação no protótipo pode ser dividida em duas partes. A primeira parte é o carregamento do conhecimento armazenado nas ontologias e a segunda parte o carregamento dos dados de acordo com este conhecimento.

A primeira parte ocorre quando a aplicação é iniciada. Nesse momento é disparado um evento para o módulo *Service*, um módulo que tem a responsabilidade de requisitar ao servidor os dados sobre a ontologia (no formato descrito na seção 6.2) através do serviço de comunicação *CommService*. Este serviço de comunicação realiza uma chamada ao módulo *OWL Indexer* (com funcionamento descrito na seção 6.3) que é responsável pela leitura do arquivo OWL que contém a ontologia e por retornar os dados armazenados da ontologia.

Estes dados retornados ao *CommService* (já convertidos para a estrutura de dados utilizada pelo protótipo) são devolvidos para o cliente através de uma resposta assíncrona

enviada ao módulo *Service*, que recebe os dados e armazena-os, deixando os dados das ontologias disponíveis para os componentes de interface que permitem a interação com a camada de informação, no caso, as especializações de *LayerSelector*, estrutura de interface que permite a seleção de informação.

A primeira parte é representada de maneira esquemática na Figura 16, mostrando o início da aplicação no *App Entry Point* e em seguida o processo acima explicado

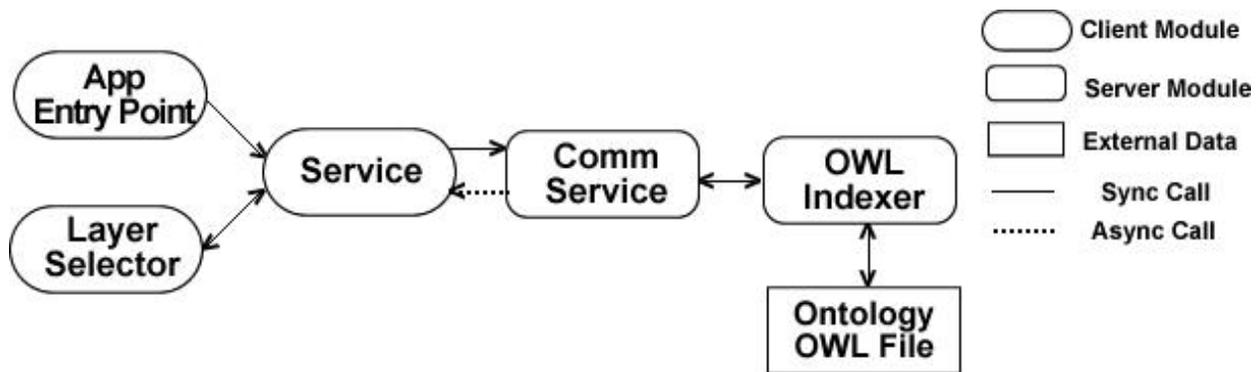


Figura 16: Carregamento das ontologias

A segunda parte ocorre quando o usuário, através da interação com algum dos componentes que interage com a camada de informação, requisita os dados de acordo com um dos conceitos da ontologia mostrados naquele componente – por exemplo, carregar todos os elementos *Building* através do componente para seleção de informação..

Após a interação do usuário, é repassado ao serviço de comunicação *CommService* o que ele deseja recuperar das diferentes fontes de dados – no exemplo acima todos os elementos *Building*. Essa informação acerca do que ele deseja recuperar é repassada ao módulo *DataSourceManager*, módulo que gerencia a comunicação com as diferentes fontes de dados externas.

Este módulo interage com dois outros diferentes módulos: i) *DataSourceXMLReader* e ii) *DataSourceConnector*. O *DataSourceXMLReader* é responsável por ler os mapeamentos das fontes dos dados do arquivo XML (*DataSource XML File*) – mapeados de acordo com o modelo mostrado na seção 6.4, e de armazenar esses mapeamentos em uma estrutura de dados interna do protótipo.

A partir da estrutura de dados com os mapeamentos retornada, o *DataSourceManager* verifica quais fontes ali presentes possuem os dados que ele deseja recuperar. Se a fonte possui aquele dado (como exemplo, elementos do tipo *Building*) ele efetua uma chamada a uma implementação da interface *DataSourceConnector* (a implementação correta é recuperada de acordo com o tipo apontado no mapeamento), passando como parâmetro o que deseja recuperar, os metadados da fonte e o mapeamento dos dados de acordo com os conceitos das ontologias.

A interface *DataSourceConnector* possui diferentes implementações para cada tipo de fonte de dados – por exemplo, existe um *PGConnector* para recuperar dados de uma base PostgreSQL e um *KMLConnector* para recuperar dados KML. Ao receber os parâmetros anteriormente mencionados o conhecimento do tipo da fonte é utilizado em conjunto com o mapeamento para recuperar o elemento desejado e devolver uma lista com todas as instâncias daquele elemento presentes na base ao *DataSourceManager*.

Após recuperar todas as instâncias de todas as fontes de dados do elemento, estes dados são adicionados a uma lista e devolvidos ao serviço de comunicação que de maneira assíncrona envia esses dados ao módulo *Map* para que possam ser plotados no mapa.

Com a lista com todos os elementos que serão plotados, o módulo *Map* percorre repassando individualmente seus dados ao módulo *FeatureBuilder*, módulo responsável por transformar o objeto com dados da determinada instância de um elemento em uma *feature* “plotável” no mapa.

O módulo *FeatureBuilder* é responsável pela criação da estrutura de atributos e a geometria do elemento que será plotado. O estilo é delegado por este módulo a um outro módulo, o *StyleBuilder*, que efetua a leitura a do arquivo CSS para recuperar o estilo correspondente a classe daquela *feature* e com a auxílio do módulo *CSS Parser* efetua as conversões citadas na seção 6.7.1, e devolve ao *FeatureBuilder* o estilo da feature no formato adequado.

A feature criada é retornada ao módulo *Map* que a adiciona ao mapa. Uma visão esquemática da estrutura é apresentada na Figura 17, com as interações com o Mapa sendo enviadas ao serviço responsável pela comunicação, o processo de recuperação do conhecimento conforme acima explicado e por fim a criação e exibição dessas *features* com auxílio do módulo *FeatureBuilder*.

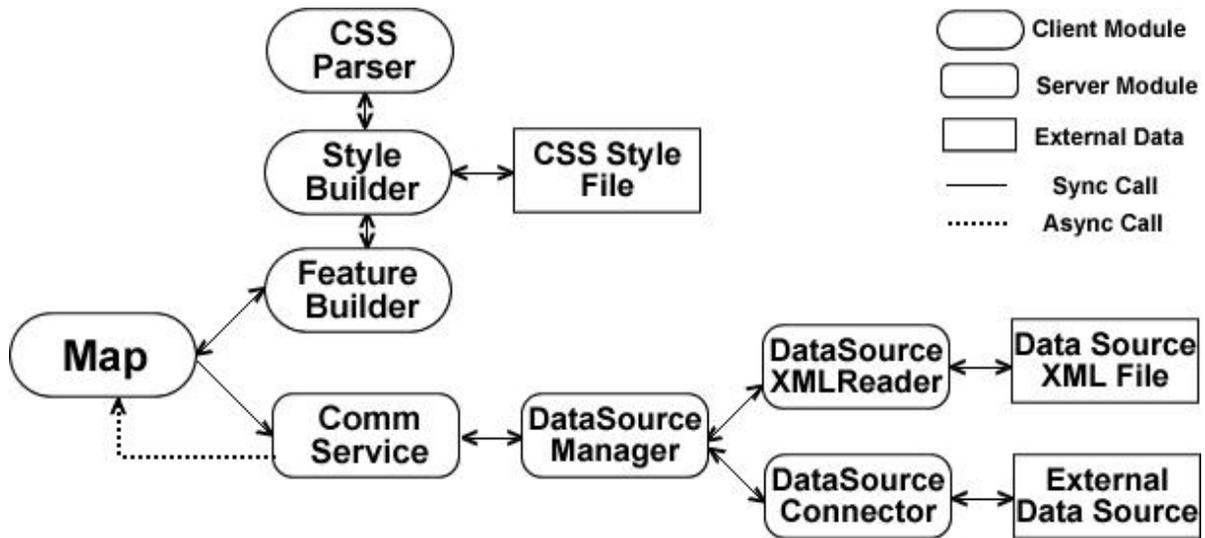


Figura 17: Recuperação dos dados presentes nas fontes externas

## 7 EXPERIMENTOS

Neste capítulo são descritos alguns experimentos realizados sobre o protótipo desenvolvido. Os resultados são analisados em relação ao cumprimento dos quatro objetivos apresentados na seção 1.1:

- 1) alteração de domínio sem necessidade de alteração no código;
- 2) integração de dados de fontes heterogêneas em diferentes níveis de detalhe; e
- 3) interação com camadas de informação de maneira hierárquica.

### 7.1 FONTE DE DADOS

Como fonte de dados base para a criação e demonstração do protótipo foram utilizados Shapefiles – arquivos para armazenamento de dados geográficos de formato proprietário da ESRI - com dados relativos a cidade de Medford no estado de Oregon nos Estados Unidos. Os dados foram levantados pela prefeitura da cidade de Medford e disponibilizado no formato de Shapefiles utilizando coordenadas geográficas (latitude e longitude em graus).

As camadas utilizadas vindas dos diferentes Shapefiles foram:

- **medford\_hospitals**: dados relativos aos hospitais da cidade representados como pontos;
- **medford\_firestation**: dados relativos a unidades do corpo dos bombeiros representado como ponto;
- **medford\_building**: dados relativos as construções presentes na cidade de Medford representados como polígonos;

Para utilização dos dados neste trabalho, focando a parte de integração de diferentes fontes de dados. Estes dados foram convertidos e colocados em tabelas dentro de uma base PostgreSQL e porções destes foram colocadas também em arquivos no formato KML.

A conversão dos dados do formato Shapefile para consequente importação pela base de dados foi realizada através do utilitário shp2pgsql disponibilizado na instalação do Postgis , através de sua execução no formato shp2pgsql [<opções>] <shapefile> <tabela> <base de dados>.

Os dados colocados em arquivos KML foram criados de maneira manual.

## 7.2 CENÁRIO INICIAL

Como cenário inicial, o protótipo possui as ontologias de descrição do espaço e de aplicação mostrados no Capítulo 5. Os mapas base já estão previamente definido e são as quatro camadas de mapa disponibilizadas pela Google Maps API (Normal, Híbrida, Satélite e Física).

Para iniciar as experimentações é inicialmente carregado o *Shapefile medford\_hospitals* em uma base de Dados PostgreSQL na tabela medford\_hospital.

Em seguida é criado e salvo na ontologia de domínio da aplicação o elemento Hospital com seus atributos, o elemento mostrado no Capítulo 5, também são criadas suas superclasses HealthBuilding e Building, sendo Building uma subclasse da classe NonNatural que é uma subclasse de Spatial (classe dos elementos espaciais).

O passo seguinte é o mapeamento entre os conceitos presentes na ontologia com os nomes físicos dos dados na base que é realizado com a adição do XML abaixo no arquivo DataSourcees.xml que serve como lugar para armazenar as fontes de dados da aplicação, tendo como resultado no arquivo:

```
<Source>
<DataSource>
  <Name>Postgre</Name>
  <Description>Base Postgree</Description>
  <Type>PGSQL</Type>
  <Host>localhost</Host>
```

```

<User>postgres</User>
<Pass>*****</Pass>
<Port>5432</Port>
<Database>postgis</Database>
<SpatialElement name="Hospital" table="medford_hospital">
    <Attribute column="name"
type="Simple">ElementName</Attribute>
    <Attribute column="the_geom"
type="Geometry">Form</Attribute>
</SpatialElement>
</DataSource>
</Source>

```

Após esta etapa é criada uma regra no arquivo de estilos (CSS) da aplicação com o nome do elemento (no caso #Hospital) que coloca os elementos hospital com pontos mostrados com raios de 6px e fundo vermelho seguindo as regras mostradas na seção anterior.

```

#Hospital {
    background-color: #FF0000;
    width: 6px;
}

```

Com o elemento adicionado a ontologia na estrutura de ontologia que o armazena (um arquivo .owl), os dados carregados na base, o mapeamento entre os termos da ontologia e os nomes físicos na base e o estilo criado o passo seguinte é abrir o protótipo e carregar os dados, obtendo como resultado o cenário inicial que pode ser visualizado na Figura 18.

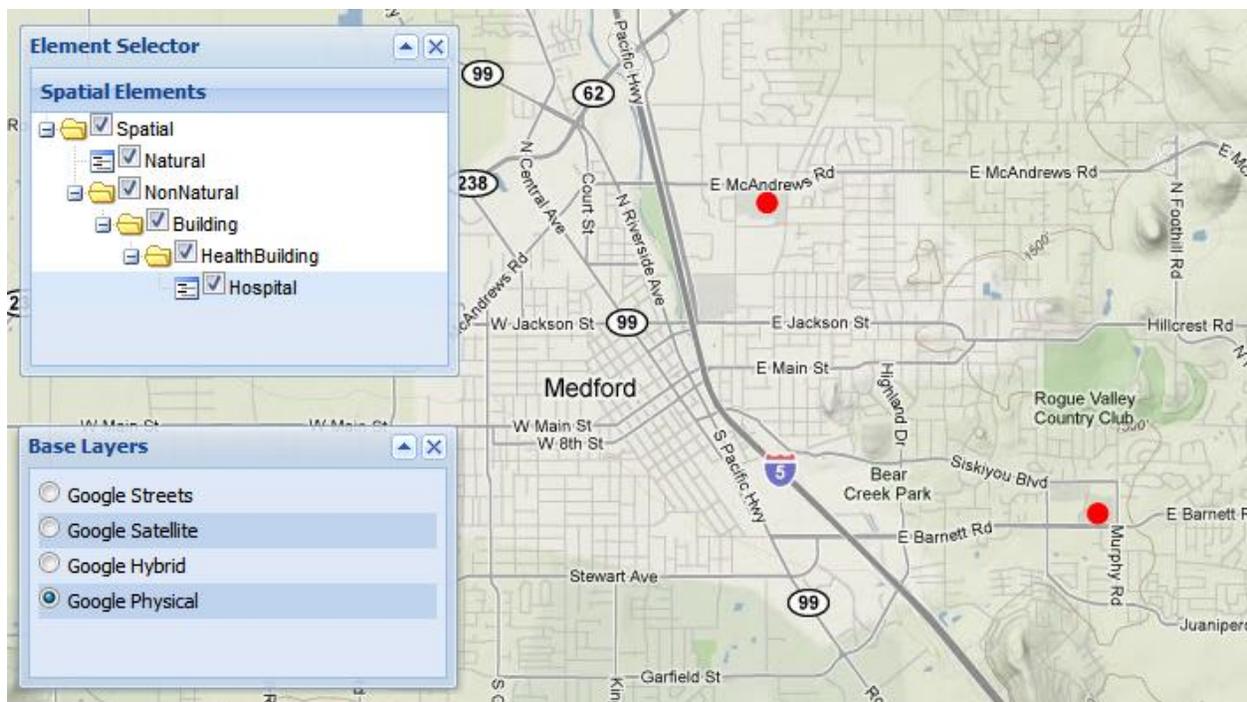


Figura 18: Cenário inicial do protótipo

### 7.3 EXPERIMENTOS E RESULTADOS

Nesta seção, são apresentados e realizados alguns experimentos de modo a validar a aplicação quanto ao cumprimento de seus objetivos. Os experimentos foram realizados com escopo minimalista de modo a facilitar a visualização da comprovação do conceito.

#### Experimento 1 - Integração de dados de fontes heterogêneas

Este experimento demonstra como é possível integrar dados de fontes heterogêneas através do mapeamento de conceitos das ontologias de domínio e aplicação.

Partindo do estado inicial e com os hospitais já carregados, nesse experimento é simulado um cenário onde um novo hospital é criado na cidade e o dado ainda não está disponível na fonte de dados que contém dados da prefeitura. Porém esse dado já está disponível em uma fonte que o disponibiliza em formato KML.

A fonte de dados que se deseja tornar disponível para visualização no Mapa é o seguinte arquivo KML que possui o nome Hospital.kml:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Document xmlns="http://www.opengis.net/kml/2.2"
xmlns:ns2="http://www.w3.org/2005/Atom"
xmlns:ns3="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
xmlns:ns4="http://www.google.com/kml/ext/2.2">
  <Placemark>
    <name>Medford Central Hospital</name>
    <ExtendedData>
      <Data name="rooms">
        <value>34</value>
      </Data>
    </ExtendedData>
    <Point>
      <coordinates>-122.8695,42.31058</coordinates>
    </Point>
  </Placemark>
</Document>

```

O arquivo possui elementos do tipo hospital. Para tornar esse aplicativo disponível na aplicação sob o conceito da ontologia é preciso adicionar um mapeamento para a nova fonte de dado:

```

<DataSource>
  <Name>KML</Name>
  <Description>Arquivo KML</Description>
  <Type>KML</Type>
  <Host>C:/Hospital.kml</Host>
  <SpatialElement name="Hospital">
    <Attribute column="name" type="Simple">Nome</Attribute>
    <Attribute column="the_geom"
type="Geometry">Geometry</Attribute>
  </SpatialElement>
</DataSource>

```

Como host é colocado a localização do arquivo. O elemento é então mapeado com base nos atributos que irá buscar na fonte de dados. Como o KML não possui nome para a geometria é utilizado o genérico “the\_geom” para indicar que a geometria deve ser buscada no arquivo.

Após o arquivo disponível e o mapeamento criado, para realizar o experimento basta entrar na aplicação, ir até o seletor de elementos e carregar os dados de hospital, tendo como resultado a Figura 19. Os pontos vermelhos são os hospitais, conforme o estilo especificado no cenário inicial. O hospital proveniente do arquivo KML é o terceiro ponto, mais a esquerda da imagem.

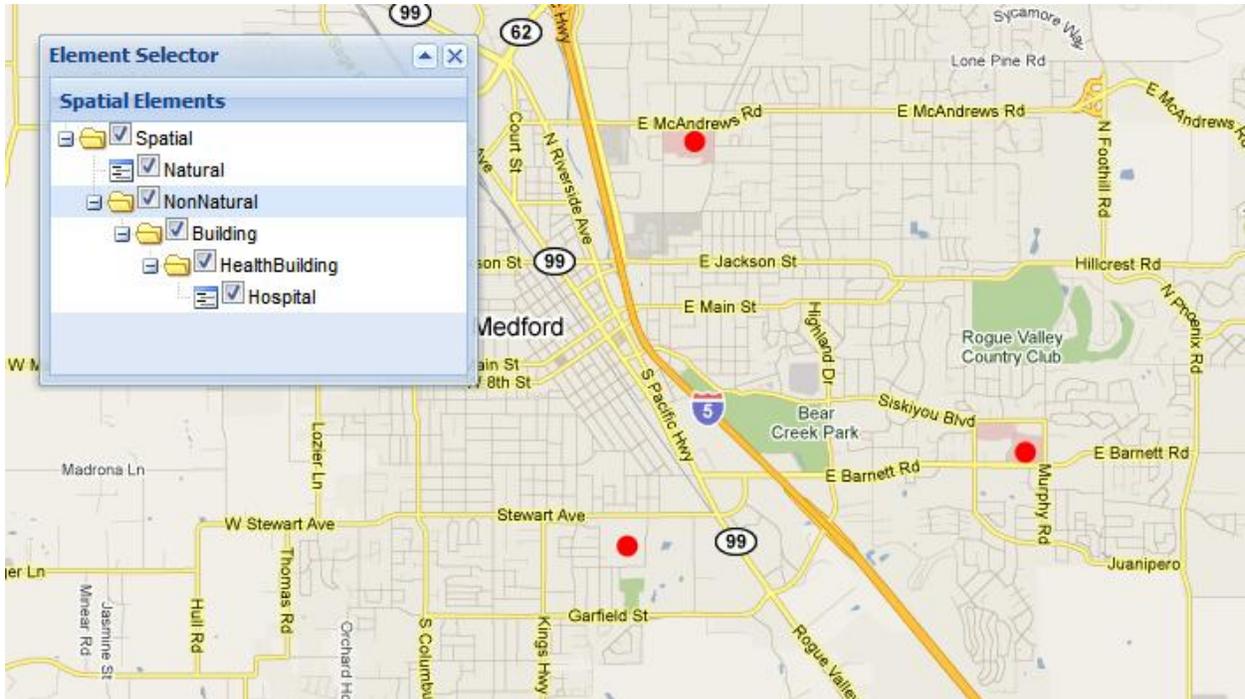


Figura 19: Resultado do Experimento com dados de diferentes fontes integrados

Com este experimento foi possível obter como resultado a comprovação do segundo objetivo apontado no início deste capítulo, sendo possível adicionar novas fontes de dados para um mesmo elemento descrito na ontologia, permitindo a integração de dados de diferentes fontes.

### **Experimento 2 - Alteração de elementos de domínio sem necessidade de alteração no código**

Este experimento tem a intenção de demonstrar como é possível, através da implementação do protótipo da arquitetura apresentada neste trabalho realizar uma alteração de domínio sem a necessidade de alteração no código da aplicação cumprindo o primeiro objetivo apresentado no início deste capítulo.

Para tal, será adicionado o elemento *FireStation* à ontologia de domínio mostrada no Capítulo 5 e serão mapeados seus dados. Estes dados previamente carregados em uma base de PostgreSQL provenientes de um Shapefile com as unidades do corpo de bombeiros na cidade de Medford.

Inicialmente foi adicionado o elemento com seus atributos na ontologia com os conceitos de domínio:

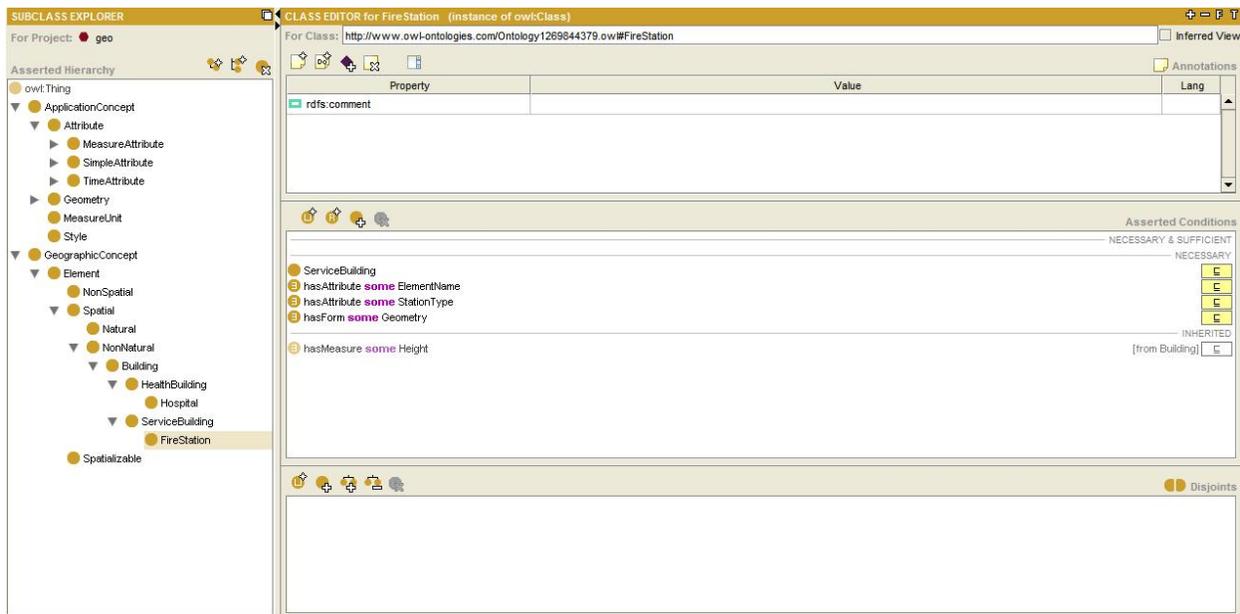


Figura 20: Elemento FireStation

O passo seguinte é a criação do mapeamento entre os conceitos do elemento criado e adicionado à ontologia com a fonte de dados (Shapefile carregado na tabela medford\_firestations na base de dados Postgree):

```
<SpatialElement name="FireStation" table="medford_firestations">
```

```
  <Attribute column="station" type="Simple">ElementName</Attribute>
```

```
  <Attribute column="the_geom" type="Geometry">Geometry</Attribute>
```

```
  <Attribute column="legend" type="Simple">StationType</Attribute>
```

</SpatialElement>

Após a adição do elemento na fonte de dados, que já estava mapeada por possuir o elemento hospital utilizado anteriormente, é criado um estilo para as estações de corpo dos bombeiros (FireStations). Foi utilizado um raio menor e como cor foi definido um tom de azul:

```
#FireStation {  
    background-color: #00008B;  
    width: 4px;  
}
```

A seguir, a aplicação é recarregada, o elemento é mostrado na árvore e seus dados são carregados através do menu *Load Data* (ao clicar com o botão direito sobre o elemento na aplicação) e se tem como resultado o visto na Figura 21. Os elementos *FireStation* são mostrados no mapa como círculos azuis (em menor tamanho) conforme o estilo.

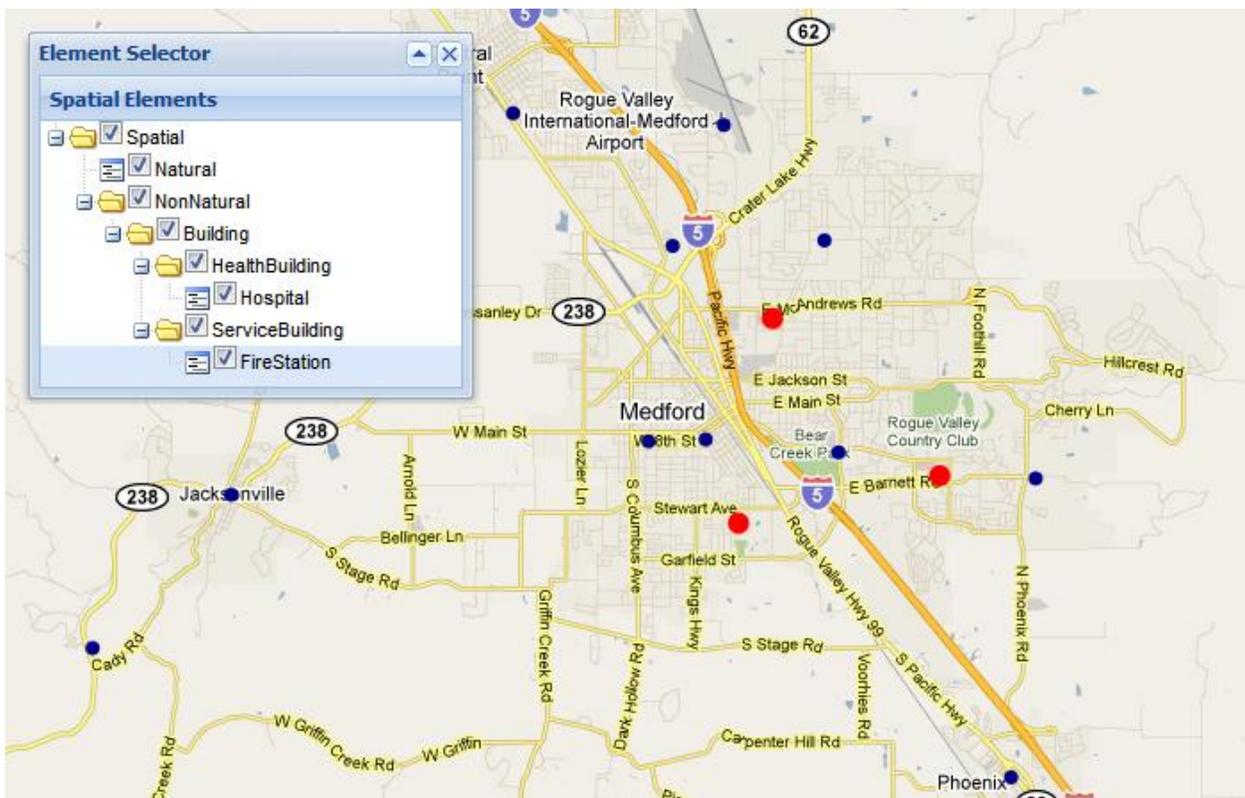


Figura 21: Resultado do Experimento 2

Com este experimento foi possível obter como resultado a comprovação do objetivo, sendo possível realizar alterações no domínio sem a necessidade de um novo trabalho de codificação.

### Experimento 3 - Interação com camadas de informação de maneira hierárquica

O experimento 3 demonstra como é realizada a interação com camadas de informação através de uma estrutura hierárquica ao invés da visão comum de camadas planas para os tipos de dado.

Inicialmente, é disparado o carregamento dos elementos a partir do nó “NonNatural” da árvore de seleção de elementos, esta ação carrega os elementos do tipo “NonNatural” e todos os elementos filhos deste nó, carregando por consequência os elementos “Hospital” e “FireStation” mapeados nos passos anteriores, conforme pode ser visto na Figura 22.

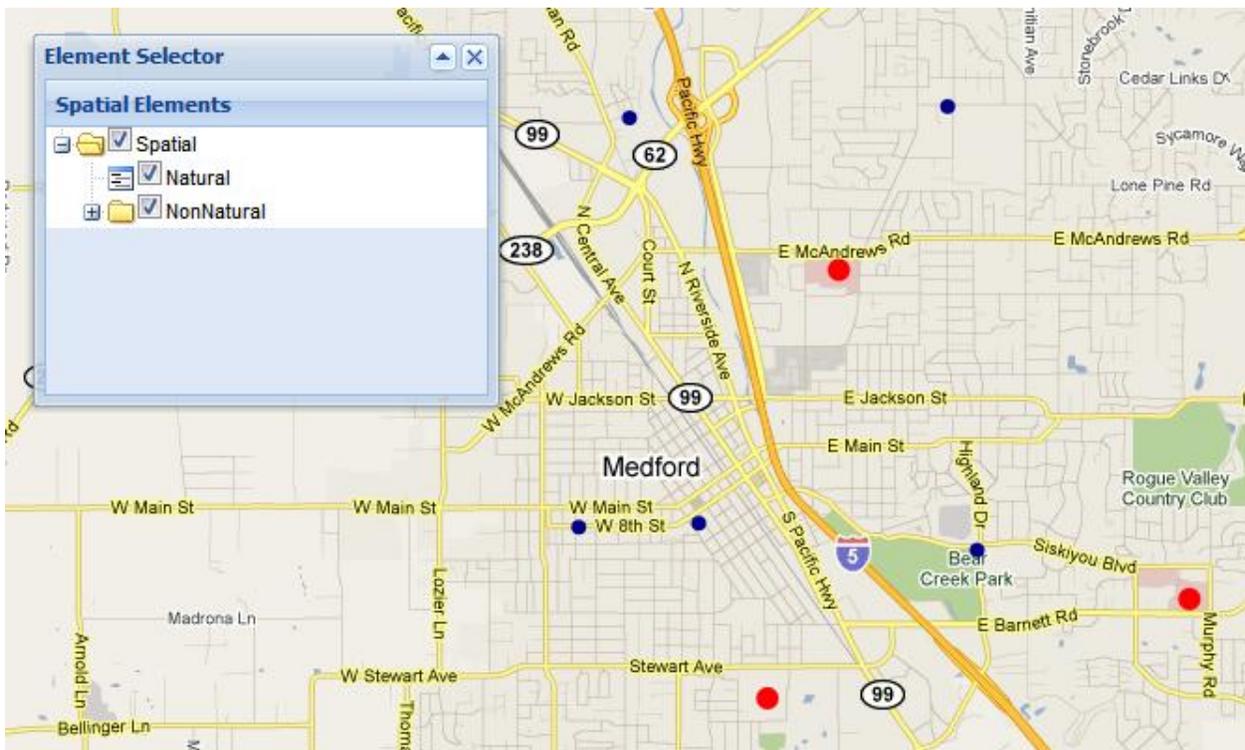


Figura 22: Elementos “NonNatural” carregados

A seguir, o nó “NonNatural” é desmarcado, e todos os seus filhos também são ocultados do mapa tendo como resultado o visto na Figura 23.

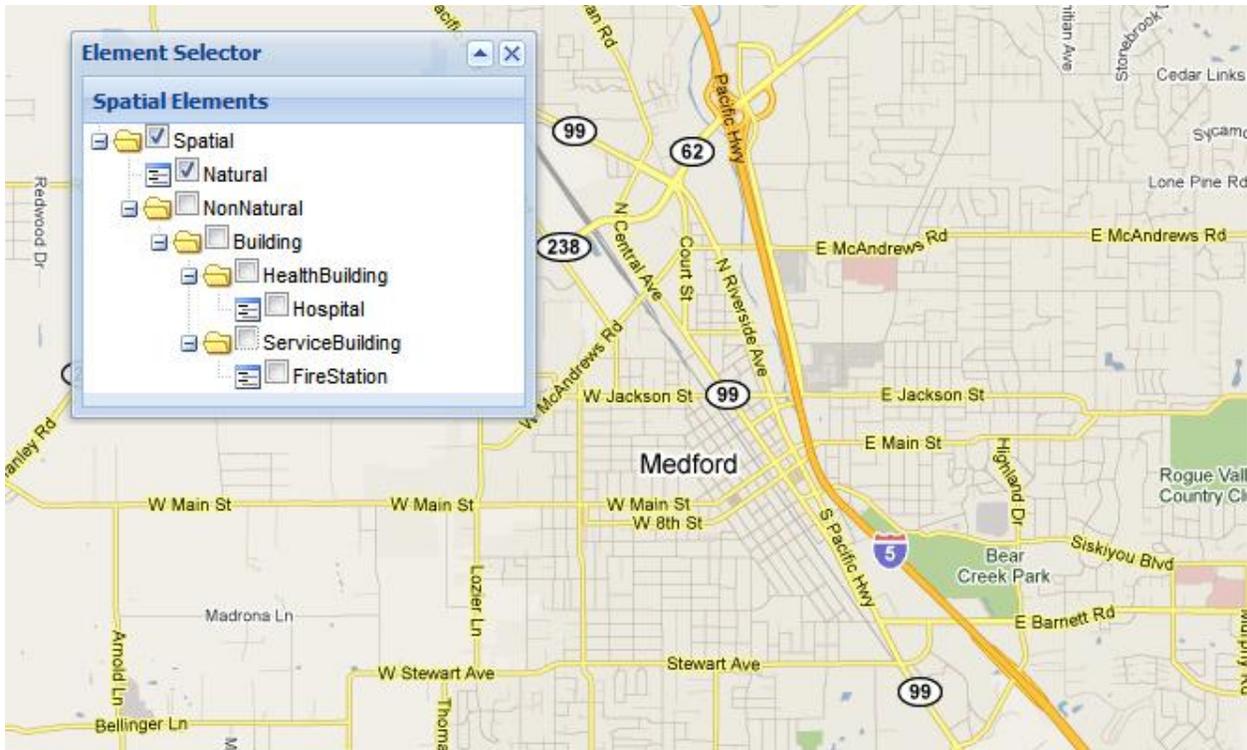


Figura 23: Interação hierárquica com camada de informação

Neste experimento, foi possível demonstrar de maneira simples a interação com as camadas de informação da aplicação utilizando a hierarquia descrita na ontologia de conceitos anteriormente apresentada.

O resultado esperado foi atingido, cumprindo o terceiro objetivo colocado no início do capítulo. Este conceito pode ser utilizado futuramente com filtragens e/ou análises através de uma visão integrada de dados provenientes de diferentes fontes.

## 8 CONCLUSÃO E TRABALHOS FUTUROS

A partir da primeira década do século XX uma série de trabalhos tem proposto a integração de dados e a utilização de semântica em Sistemas de Informação Geográfica. Contudo, ainda não existem soluções definitivas para a reunião de conhecimento formalizado com esses sistemas.

Este trabalho propôs uma arquitetura capaz de reunir estes conceitos. Ela facilita a integração de dados de fontes heterogêneas e utiliza uma forma hierárquica de organização de níveis de informação e interação com os dados, ainda pouco comum em SIG. Para definir esta hierarquia foi buscada inspiração na Geografia, principalmente no trabalho de Santos (2006 e 2008), que dá sustentação teórica ao modelo conceitual proposto para a descrição do espaço.

Com a proposição da arquitetura, foi implementado um protótipo seguindo o proposto para que fosse possível realizar experimentos e validar a arquitetura em relação aos objetivos inicialmente colocados pelo trabalho.

Esta proposta foi inspirada pela Geografia a partir dos conceitos mostrados no Capítulo 3 e utiliza uma ontologia para descrever o espaço em alto nível, uma ontologia de aplicação que contém os conceitos reconhecidos pela aplicação e uma ontologia de domínio com os conceitos relativos à área que se deseja representar.

A relação entre a ontologia de domínio e a ontologia de aplicação permite a criação de uma descrição dos elementos de domínio de acordo com conceitos da aplicação, conforme mostrado no Capítulo 5.

Com estas descrições, é possível mapear o conhecimento acerca dos elementos presente na ontologia de domínio para diferentes fontes de dados, permitindo a integração de fontes heterogêneas.

A hierarquia presente na ontologia de domínio serve como base para a criação de componentes de interface que permite a interação com os dados de domínio com a maneira hierárquica ali estabelecida.

Através da implementação do protótipo foi possível atingir os três principais objetivos apresentados na introdução deste trabalho:

- integração de dados de fontes heterogêneas; e
- interação com camadas de informação presentes sobre o mapa de maneira hierárquia.
- alteração de elementos de domínio – as descrições dos elementos presentes na ontologia de domínio – que se deseja representar na aplicação sem a necessidade de alteração de código;

Sendo estes cumpridos com sucesso, contudo, devido à questão da limitação de tempo para o desenvolvimento deste trabalho, as soluções apresentadas para a implementação do protótipo tem objetivo didático e de demonstração não sendo a melhor entre as soluções possíveis para implementação da arquitetura, deixando alguns temas como possíveis trabalhos futuros.

Entre os possíveis trabalhos futuros é necessário destacar alguns:

1) A criação automática ou semi-automática de mapeamentos dos dados relacionando conceitos e descrições as fontes. Como estava fora do escopo deste trabalho a questão dos mapeamentos, esta não foi trabalhada em profundidade apresentando apenas uma construção manual dos mapeamentos;

2) Desenvolvimento de uma estrutura de filtros (que permitam comparações aritméticas, lógicas e espaciais) que considere os atributos descritos na estrutura de ontologia utilizada para descrever os conceitos do domínio e da aplicação;

3) Adaptação para utilização da arquitetura com ontologias consolidadas e de maior porte permitindo o reuso dos conceitos descritos por essas ontologias;

4) O desenvolvimento das partes que relacionam as questões da estrutura e do processo de acordo com a teoria de Santos citada previamente neste trabalho;

5) Inserção do conceito de trajetórias - que podem ser vistas como seqüências de eventos - na implementação para que estas possam ser trabalhadas da mesma forma que os eventos, como elementos “especializáveis”.

## REFERÊNCIAS BIBLIOGRÁFICAS

CÂMARA, G. MONTEIRO, A. M. V. MEDEIROS, J. S. **Representações Computacionais do Espaço: Um Diálogo entre a Geografia e a Ciência da Geoinformação**. 2002.

CÂMARA, G. MONTEIRO, A. M. V. MEDEIROS, J. S. Fundamentos epistemológicos da ciência da geoinformação. In: G Câmara, C Davis, AM Monteiro (org.), *Introdução à Ciência da Geoinformação*, <<http://www.dpi.inpe.br/gilberto/livro/introd>>, 2000b.

CÂMARA G, QUEIROZ, G. R. **Arquitetura de Sistemas de Informação Geográfica**. In: G Câmara, C Davis, AM Monteiro (org.), *Introdução à Ciência da Geoinformação*, <<http://www.dpi.inpe.br/gilberto/livro/introd>>, 2000.

CARVALHO, A. SANTOS, M. **A Geografia Aplicada**. Boletim Geográfico. Rio de Janeiro, Mar/Abr, (nº 185), 1965. p. 249-258.

CASANOVA, M. *et al.* **Bancos de dados geográficos**. Curitiba: MundoGEO, 2005. Disponível em: <<http://www.dpi.inpe.br/gilberto/livro/bdados/material.html>>. Acesso em: 30 de Abril 2010.

DENT, B. **Cartography: Thematic Map Design**. William C Brown Pub, 2006. 433p.

EGENHOFER, M. F. MARK, D. M. HERING, J. **The 9-intersection: Formalism and its use for natural-language spatial predicates**. Technical Report 94-1, National Center for Geographic Information and Analysis, Santa Barbara, CA, 1994.

FONSECA, F. EGENHOFER, M. **Ontology-Driven Geographic Information Systems**. In: MEDEIROS, C. B. (Ed.) 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, MO, pp. 14-19.

FRIENDLY, M. DENIS, D. **Milestones in the History of Thematic Cartography, Statistical Graphics, and Data Visualization**, 2006.

GeoServer. **About**. Disponível em: <http://www.geoserver.org> . Acesso em: em 7 de Setembro de 2009 as 00:41.

GRUBER, T. R. **Toward Principles for the Design of Ontologies Used for Knowledge Sharing**. International Journal of Human and Computer Studies, **43**(5/6): 907-928, 1995.

GUARINO, N. **Formal Ontology and Information Systems**. In: Proceedings of FOIS'98, Trento, Itália. Junho, 1998.

HARVEY, D. **Explanation in Geography**. London: E. Arnold, 1969;

IBGE. **Noções Básicas de Cartografia**. Disponível: < [http://www.ibge.gov.br/home/geociencias/cartografia/manual\\_nocoos/representacao.html](http://www.ibge.gov.br/home/geociencias/cartografia/manual_nocoos/representacao.html) >. Acesso em: 2 de Novembro de 2009 as 04:18.

International Cartographic Association. **Mission**. Disponível em: < <http://icaci.org/mission> >. Acesso em: 9 de Maio de 2010.

LONGLEY, P.A. GOODCHILD, M. F. MAGUIRE, D. J., RHIND, D. W. Geographic Information Systems and Science. West Sussex UK: John Wiley, 2001;

OGC. **KML**. Disponível em: < <http://www.opengeospatial.org/standards/kml> >. Acesso em: 15 de Outubro de 2009 as 22:37.

OpenGIS® **Implementation Specification for Geographic Information - Simple feature access - Part 1: Common Architecture, Revision 3**, OpenGIS Consortium, Inc, OpenGIS Project Document Number 06-103r3, 2006.

OpenGIS® **Reference Model, Revision 4**, OpenGIS Consortium, Inc, OpenGIS Project Document Number 08-062r4, 2008.

OpenGIS® **Reference Model, Revision 4, OpenGIS Consortium, Inc, OpenGIS Project Document Number 08-062r4, 2008b.**

OpenLayers. **About.** Disponível em: < <http://www.openlayers.org> >. Acesso em: 6 de Setembro de 2009 as 22:11.

PERROUX, F. **L'Économie Du XX Siécle.** Paris: Press Universitaires de France, 1969.

PostgreSQL. **About.** Disponível em: < <http://www.postgresql.org/about/> >. Acesso em: 24 de Outubro de 2009 as 00:26.

Postgis. Spatial Relationship. Disponível em: < [http://postgis.refractory.net/documentation/manual1.4/ch07.html#Spatial Relationships Measurements](http://postgis.refractory.net/documentation/manual1.4/ch07.html#Spatial_Relationships_Measurements) >. Acesso em: 9 de Setembro de 2010.

Protegé. **Overview.** Disponível em: < <http://protege.stanford.edu/overview/> >. Acesso em: 16 de Maio de 2010.

CHOLLEY, A. **Observações Sobre Alguns Pontos de Vista Geográficos.** Boletim Geográfico, nº 179 e 180. CNG: Rio de Janeiro, 1964.

ROCHA, C. H. **Geoprocessamento: tecnologia transdisciplinar.** Juiz de Fora, MG, 2000.

SANTOS, M. **Por uma Geografia Nova.** 2ed. São Paulo: Editora Hucitec, 1980;

SANTOS, M. **METAMORFOSES DO ESPAÇO HABITADO, fundamentos teórico e metodológico Geografia.** São Paulo: Hucitec. 1988.

SANTOS, M. **A Natureza do Espaço: Técnica e Tempo, Razão e Emoção.** 4ed. São Paulo: Editora da Universidade de São Paulo, 2006;

SANTOS, M. **Espaço e Método.** 5ed. São Paulo: Editora da Universidade de São Paulo, 2008;

W3C. **Cascading Style Sheets.** Disponível em: < <http://www.w3.org/Style/CSS/> >. Acesso em: 8 de Maio de 2010.