

Centro Universitário Feevale

André Milke dos Santos

Extensões Espaciais de Sistemas Gerenciadores de Banco de Dados

Novo Hamburgo, julho de 2006.

André Milke dos Santos

Extensões Espaciais de Sistemas Gerenciadores de Banco de Dados

**Centro Universitário Feevale
Instituto de Ciências Exatas e Tecnológicas
Curso de Ciência da Computação
Trabalho de Conclusão de Curso**

Professor Orientador: Juliano Varella de Carvalho

Novo Hamburgo, julho de 2006.

RESUMO

A crescente utilização de Sistemas de Informação Geográfica (SIG) criou a necessidade de armazenamento de dados espaciais de forma mais adequada. Para tal fim utilizam-se Sistemas Gerenciadores de Banco de Dados (SGBD) Geográficos.

Estes tipos de SGBDs podem ser gerenciados pelas estratégias: Dual onde se utiliza SGBDs relacionais e arquivos em separado para o armazenamento dos dados espaciais; SGBDs relacionais que utilizam campos longos para o armazenamento de informações espaciais; ou SGBDs Objeto-Relacionais que possuem extensões espaciais para o tratamento destes dados. Este trabalho abordará os conceitos fundamentais a respeito de extensões espaciais para banco de dados e além disso, abordará também a aplicação da extensão espacial do SGBD PostgreSQL denominada PostGIS, no projeto de Dados Ambientais da Região do Vale do Rio dos Sinos, permitindo o acesso aos usuários, a algumas consultas referentes aos dados espaciais e descritivos armazenados.

Palavras-chave: Sistemas de Informação Geográfica, Extensão Espacial, Banco de Dados Espacial, Banco de Dados Geográfico, PostGIS, Mapserver.

ABSTRACT

The increasing use of Geographic Information Systems (GIS) created the necessity of to storage geographic data adequately. For to solve this question Database Management System (DBMS) Geographic are used.

These types of DBMS can be managed by strategies: Dual, where are used relational DBMS and files for to storage spatial data; Relational DBMS that uses long fields for the storage of spatial information; or object-relational DBMS (ORDBMS) that has spatial extensions to treat of these data. This work will approach the basic concepts regarding spatial extensions.

Moreover, it will also approach the application of DBMS spatial extension PostgreSQL, called PostGIS, in the project of Ambient Data of the Region of the Valley of the Rio dos Sinos, allowing to the users, make queries on the stored spatial and descriptive data.

Keywords: Geographic Information Systems, Spatial Extensions, Spatial Database, Database Geographic, PostGIS, Mapserver.

LISTA DE FIGURAS

Figura 1 - Componentes de um SIG	11
Figura 2 - Elementos da Estrutura Vetorial	12
Figura 3 - Estrutura Matricial	13
Figura 4 - Arquitetura Dual	15
Figura 5 - Níveis de Abstração de Aplicações Geográficas	21
Figura 6 - Notação gráfica para as classes do modelo OMT-G	23
Figura 7 - Geo-campos	24
Figura 8 - Geo-objetos	25
Figura 9 - Relacionamentos	26
Figura 10 – Cardinalidade	26
Figura 11 - Generalização.....	27
Figura 12 - Exemplos de Generalização espacial	28
Figura 13 - Agregação notação UML.....	29
Figura 14 - Agregação entre uma classe convencional e uma classe georreferenciada.....	29
Figura 15 - Agregação espacial “todo-parte”	29
Figura 16 - Diagrama de Classes do GeoFrame	30
Figura 17 – Diagrama de Temas.....	56
Figura 18 – Tema Potencial Elétrico	58
Figura 19 - Arquitetura Mapserver CGI com PostGIS	67
Figura 20 - Exemplo de um <i>Mapfile</i> simples com conexão ao banco de dados	69
Figura 21 - Exemplo utilizando a função intersects do PostGIS	71
Figura 22 - Exemplo utilizando duas feições	72
Figura 23 - Arquitetura Mapserver/Php Mapscript com PostGIS	75
Figura 24 - Tela Inicial	76
Figura 25 - Mapa de Referência	76
Figura 26 - Opção Mover	78

Figura 27 - Opção Aproximar	78
Figura 28 - Opção Afastar	79
Figura 29 - Opção Consultar	81
Figura 30 - Relatório gerado pela opção Consulta	81
Figura 32 - Grupo Temas.....	82
Figura 33 - Municípios por tipo de empresa.....	84
Figura 34 - Municípios por tipo de prop. rural	85
Figura 35 - Municípios atendidos/subestação.....	86
Figura 36 - Buffer Sub. Cidade Industrial	87
Figura 37 - Municípios contidos na bacia.....	88
Figura 38 - Concentração de Subestações	89
Figura 39 - Vizinhos de Novo Hamburgo	90
Figura 40 - Diferença entre município e bacia	91

LISTA DE ABREVIATURAS

GEOS	- <i>Geometry Engine Open Source</i>
INPE	- Instituto Nacional de Pesquisas Espaciais
MBR	- <i>Minimum Bounding Rectangle</i>
OGC	- <i>OpenGIS Consortium</i>
SGBD	- Sistema Gerenciador de Banco de Dados
SGBDOR	- Sistema Gerenciador de Banco de Dados Objeto-Relacional
SGBDR	- Sistema Gerenciador de Banco de Dados Relacional
SIG	- Sistemas de Informação Geográfica
SQL	- <i>Structured Query Language</i>
TDEs	- Tipo de Dados Espaciais
UML	- <i>Universal Modeling Language</i>
WKB	- <i>Well-Known Binary</i>
WKT	- <i>Well-Known Text</i>

SUMÁRIO

1. SISTEMAS DE INFORMAÇÃO GEOGRÁFICA.....	10
1.1 Categorização Vetorial	11
1.2 Categorização Matricial.....	13
1.3 Arquiteturas para Banco de Dados Geográficos	14
1.3.1 Arquitetura Dual.....	14
1.3.2 Arquitetura Integrada Baseada em Campos Longos	16
1.3.4 Arquitetura Integrada Baseada em Extensões Espaciais	17
1.3.5 Arquitetura Integrada Combinada	18
2. MODELO DE DADOS GEOGRÁFICOS	19
2.1 OMT-G	21
2.1.1 Classes	22
2.1.2 Relacionamentos	25
2.1.3 Generalização e Especialização	27
2.1.4 Agregação	29
2.2 GeoFrame.....	30
3. EXTENSÕES ESPACIAIS.....	33
3.1 OpenGIS	33
3.1.1 Modelo Essencial	34
3.1.2 Modelo Abstrato.....	34
3.2 PostGIS	37
3.2.1 Objetos Espaciais	37
3.2.2 Sistema de Coordenadas	39
3.2.3 Tabela GEOMETRY_COLUMNS	40
3.2.4 Tabela SPATIAL_REF_SYS.....	41
3.2.5 Índices.....	42
3.2.6 Consultas Espaciais	43
3.3 Oracle Spatial.....	46
3.3.1 Objetos Espaciais	46
3.3.2 SDO_GEOMETRY	46
3.3.3 Índices.....	48
3.3.4 Consultas Espaciais	48
3.4 MySQL	50
3.4.1 Tipos de Dados Espaciais do MySQL.....	50
3.4.2 Índices.....	51
3.4.3 Consultas Espaciais	51
3.5 Comparação Entre as Extensões Espaciais.....	53

4. MODELAGEM CONCEITUAL DE DADOS DO PROJETO DE CADASTRO DE DADOS AMBIENTAIS DO VALE DO RIO DO SINOS.....	55
5. MAPSERVER.....	60
5.1 Mapfile.....	62
5.2 Exemplos do Mapserver com o PostGIS.....	67
6. APLICATIVO MAPSERVER/PHP MAPSCRIPT.....	73
6.1 PHP Mapscript.....	74
6.2 Descrição da Aplicação.....	74
6.3 As opções do Grupo Navegação.....	77
6.4 As opções do Grupo Modo de Navegação.....	79
6.5 As opções do Grupo Temas.....	81
6.6 Consultas.....	82
6.6.1 Consultas Específicas.....	83
6.6.2 Consultas Genéricas.....	87
CONCLUSÃO.....	92
BIBLIOGRAFIA.....	94

INTRODUÇÃO

Atualmente, observa-se um grande crescimento de inclusão de técnicas para tratamento de dados geográficos nos sistemas de informação. Estes sistemas, denominados Sistemas de Informação Geográfica (SIG), começaram a ser desenvolvidos no início das décadas de 80 e 90 como simples sistemas “stand-alone¹”. Estes não tinham a capacidade de compartilhar ou gerenciar dados de forma eficiente, isto porque foram construídos com centenas de funções e constituídos de pacotes genéricos, dificultando muito sua utilização por pessoas leigas (Ferreira, 2003).

Para sanar parte destas deficiências, Bancos de Dados Espaciais e Geográficos foram incorporados aos SIGs, com o intuito de armazenar e gerenciar este tipo de informação, fornecendo suporte a consultas e diversas estruturas de índices e manipulações de dados espaciais.

Os dados espaciais são caracterizados pela localização geográfica sobre a superfície terrestre em certo instante. Dados espaciais são modelados para representar uma área ou polígono, linha, ponto ou algum objeto complexo, como por exemplo, uma rede de esgoto ou malha rodoviária. Este tipo de dado é armazenado através de um sistema de coordenadas (SILVA, 2005).

Segundo Silberschatz et al (1999, p.700), bancos de dados geográficos são banco de dados espaciais utilizados para armazenar informações geográficas, como mapas. Já Câmara et al (2005, p.2-3) utiliza o termo sistemas gerenciadores de dados

¹ Do inglês, autônomo. Sistema autônomo, que faz seu processamento sozinho.

geográficos como um mecanismo dentro de um SGBD que serve para armazenar dados e também a geometria dos objetos espaciais.

A partir da utilização de banco de dados geográficos, é possível construir aplicações para manipulação de dados espaciais de forma mais coesa e ainda usufruir das vantagens que possui um SGBD. Dentre elas, cita-se a persistência dos dados, integridade, segurança e um aumento da eficiência, pois com o uso de um SGBD, as informações podem ser extraídas e alteradas de modo mais prático e eficaz (SILBERSCHATZ et al., 1999).

Há três formas de trabalhar com dados geográficos. A primeira é a utilização da estratégia Dual, onde utiliza-se um SGBD como repositório dos atributos convencionais dos objetos geográficos e arquivos em separado para armazenar as representações geométricas destes (CÂMARA, 1995).

A segunda opção é a utilização de um SGBD Relacional (SGBDR) para armazenar os dados espaciais, tanto seus componentes espaciais e alfanuméricos. Outra opção é a utilização de extensões espaciais desenvolvidas sobre SGBDs Objeto – Relacional (SGBDOR), que definem funcionalidades e procedimentos capazes de armazenar, acessar e analisar dados espaciais (FERREIRA, 2003).

Utilizando-se de extensões espaciais pode-se usufruir de tipos de dados espaciais definidos por elas, tais como ponto, linha e polígono. Estas extensões permitem que tais dados sejam manipulados como qualquer outro tipo de dado de SGBD. Além desta característica, há a extensão da linguagem SQL para proceder consultas sobre estes dados, ofertando operações e funções para consultar relações espaciais.

A aplicabilidade de SIGs e SGBDs geográficos é vasta, podendo ser utilizados com grande propriedade na área rural, urbana e ambiental, pois podem auxiliar a tomada de decisões. Como exemplos de usabilidade pode-se citar o controle de tráfego urbano

de uma cidade, o registro e manutenção de dados a respeito de lotes urbanos e a criação de mapas interativos em páginas web².

O presente trabalho abordará conceitos sobre extensões espaciais e será dividido em seis capítulos distintos. O primeiro capítulo, SIG, traz uma explanação sobre os conceitos dos Sistemas de Informação geográfica, suas categorizações e arquiteturas. Em seguida, é abordada a Modelagem de Dados Geográficos, definindo-a e apresentando duas propostas de modelo de dados. O terceiro capítulo trata das extensões espaciais, abordando o padrão OpenGIS e três extensões espaciais disponíveis no mercado: Oracle Spatial, PostGIS e MySQL. O capítulo seguinte, Modelagem Conceitual de Dados do Projeto de Cadastro de Dados Ambientais do Vale do Rio do Sinos, trata da criação de um esquema conceitual, aplicando-se uma das propostas de modelos definidas no segundo capítulo. O quinto capítulo, Mapserver, traz uma explanação sobre o servidor de mapas Mapserver e exemplos práticos de utilização. O último capítulo, Aplicativo Mapserver/Php Mapscript, descreve a implementação de uma ferramenta para exibir os dados espaciais, contidos no banco de dados, em forma de mapa.

² “Os mapas para web podem disponibilizar aos usuários o acesso às informações geográficas de modo interativo, dinâmico e constantemente atualizado” (MARISCO, 2005 p. 76).

1. SISTEMAS DE INFORMAÇÃO GEOGRÁFICA

Segundo Câmara et al 2005, o termo Sistemas de Informação Geográfica (SIG) é a definição para sistemas que realizam o tratamento computacional de dados geográficos. A principal diferença deste para um sistema de informação é que o primeiro possui a capacidade de armazenar além dos atributos descritivos, as geometrias dos diferentes tipos de dados geográficos. Com essa definição é possível indicar as características principais do SIG:

- Inserir e integrar em uma única base de dados, informações espaciais e atributos não-espaciais provindos de diferentes meios de coleta de dados;
- Oferecer mecanismos para efetuar junções das informações, através de algoritmos de manipulação e análise, e também para consultar, recuperar e visualizar o conteúdo da base de dados geográfica.

A estrutura geral de um SIG é formada pelos seguintes componentes: Interface, Entrada e Integração dos Dados, Funções de Processamento, Visualização e Plotagem, Armazenamento e recuperação (CAM, apud ROCHA, 2001, p. 16).

A Figura 1 ilustra o relacionamento entre os componentes de um SIG.

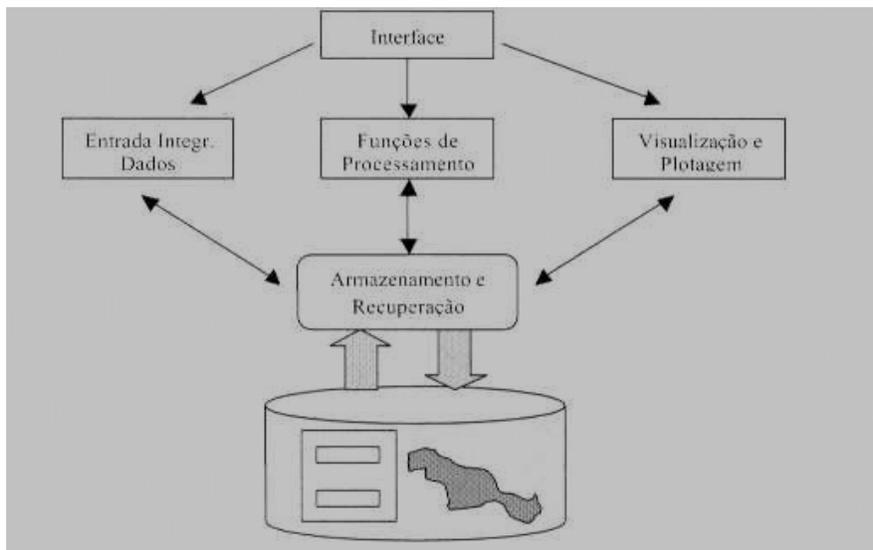


Figura 1 - Componentes de um SIG

Fonte: ROCHA, 2001, p.17.

Estes componentes se relacionam de forma hierárquica. No primeiro nível, se estabelece como o sistema é operado e controlado; No nível intermediário, um SIG deve ter mecanismos de processamento de dados espaciais; E no nível mais interno do sistema, encontra-se um Sistema Gerenciador de Banco de Dados Geográficos (SGBDG), que oferece armazenamento e recuperação dos dados espaciais.(ROCHA, 2001).

Pode-se dizer que um dado geográfico é um dado espacial, pois possui informações sobre posição, profundidade e localização, mas, além disso, ele pode conter outras informações mais detalhadas como tipo de solo, utilização da terra e área de queimadas se for um mapa para utilização agrícola, por exemplo.

Estes dados geográficos podem ser representados de duas formas distintas: vetorial ou matricial.

1.1 Categorização Vetorial

A categorização vetorial é representada a partir de um conjunto de objetos geométricos (ponto, linha ou polígono) que são referenciados geometricamente e a eles

associados atributos descritivos. Geralmente, mapas são representados vetorialmente, pois, por exemplo, estradas ou rodovias podem ser consideradas como um conjunto de linhas e estados ou municípios como polígonos. Segundo Ferreira 2003, uma representação vetorial é à busca da melhor tentativa de reprodução de um elemento, transpondo qualquer entidade ou elemento gráfico de um mapa, de maneira reduzida, a uma das três formas básicas, ilustrada na Figura 2, que são:

- Pontos: par ordenado de coordenadas espaciais (x, y). São utilizados para identificar localizações ou ocorrências no espaço. Exemplos: localização de pontes, ocorrência de doenças e localização de espécies de minerais;
- Linhas: formadas por uma cadeia de segmentos de reta, ou mais especificamente, por uma lista de coordenadas de pontos. São utilizados para representar formas unidimensionais. Exemplos: malha viária, rodovias, rios;
- Polígonos ou áreas: conjunto de linhas conectadas de forma que o último ponto de uma linha seja o primeiro da próxima. Exemplos: lotes de terra, bairros e municípios.

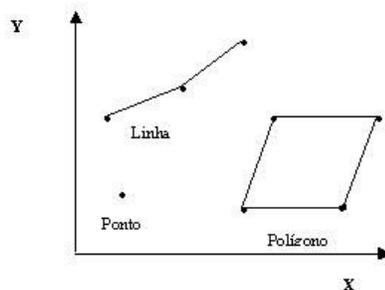


Figura 2 - Elementos da Estrutura Vetorial

Fonte: Adaptada de Silva (2002).

1.2 Categorização Matricial

A categorização matricial, também denominada raster ou tesselação, é utilizada para representar o espaço como um conjunto de *pixels* ou mapa de *bits*. Este tipo de representação, como o próprio nome indica, é composto por uma matriz bidimensional, onde, cada célula tem sua posição definida. Como exemplo pode-se citar uma foto aérea de uma área da Amazônia, onde existam diferentes tipos de vegetação. Cada célula da foto representa algum atributo, que pode ser diferenciado pela cor atribuindo, desta forma, dados a aquela posição.

Ao contrário da representação vetorial, onde cada fenômeno do mundo real está associado a um objeto espacial, na estrutura matricial os atributos dos fenômenos geográficos estão associados a grupos de células do mesmo valor [...]. A resolução de uma imagem matricial corresponde à dimensão linear mínima da menor unidade do espaço geográfico (célula) sendo considerada. Quanto menor a dimensão das células, maior a resolução da imagem matricial e, conseqüentemente, maior a quantidade de memória necessária para armazená-la. (LISBOA FILHO, 2002).

A Figura 3 apresenta a estrutura matricial, conforme citação acima:

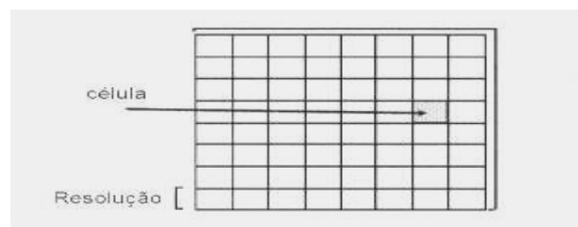


Figura 3 - Estrutura Matricial

Fonte: CÂMARA et al, 2005, p. 41.

1.3 Arquiteturas para Banco de Dados Geográficos

A integração entre atributos descritivos e dados espaciais é extremamente importante, senão essencial, para o desenvolvimento de um SIG, pois permite a análise conjunta dos atributos e a sua posição no espaço. Contudo, a preocupação com essa integração ocorreu após o desenvolvimento de SIGs e dos SGBDs atuais (SILVA, 2002).

Para integrar os atributos e os dados espaciais, Bancos de Dados Espaciais e Geográficos foram incorporados aos SIGs, com o intuito de armazenar e gerenciar as informações, fornecendo suporte a consultas e diversas estruturas de índices e manipulações de dados espaciais.

Conforme (Silva 2002), existem duas formas principais de integração entre SIGs e SGBDs, que são:

- Arquitetura Dual: que armazena os dados espaciais e atributos separadamente;
- Arquitetura Integrada: que armazena todos os dados em um SGBD, tanto os dados espaciais como os atributos.

1.3.1 Arquitetura Dual

A arquitetura dual, mostrada na Figura 4, possui um SGBD como repositório dos atributos convencionais dos objetos geográficos e arquivos em separado para armazenar as representações geométricas destes. Para associar os atributos convencionais e as representações geométricas utiliza-se de um identificador comum (CÂMARA, 1995).

Um exemplo de utilização da arquitetura dual é o SPRING, uma ferramenta de SIG desenvolvida pelo Instituto Nacional de Pesquisas Espaciais (INPE) (SILVA, 2002).

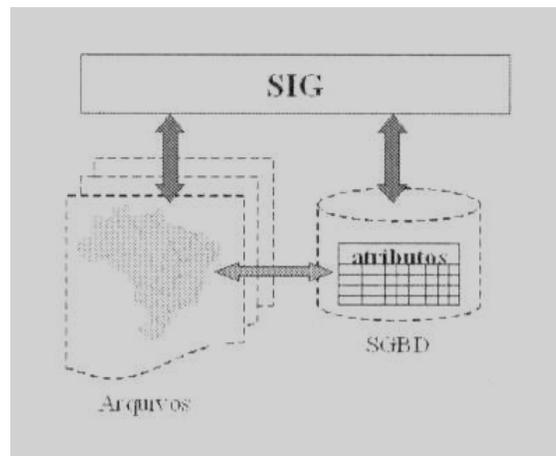


Figura 4 - Arquitetura Dual

Fonte: FERREIRA, 2003, p. 20.

Os principais problemas dessa arquitetura, segundo Ferreira 2003, são:

- Dificuldades no controle e manipulação dos dados espaciais;
- Dificuldade em manter a integridade entre o componente espacial e o SGBD repositório dos atributos convencionais;
- Lentidão no processamento por causa do processamento em partes distintas. A parte convencional da consulta é feita pelo SGBD e a espacial pelo aplicativo utilizando os arquivos proprietários;
- Falta de interoperabilidade entre os dados. Cada sistema produz seu próprio arquivo proprietário sem utilizar um padrão, tornando extremamente difícil à integração dos dados.

Em contrapartida a arquitetura dual, existe a arquitetura integrada, a qual subdivide-se em três grupos: baseada em campos longos, em extensões espaciais e combinada.

1.3.2 Arquitetura Integrada Baseada em Campos Longos

A Arquitetura Integrada Baseada em Campos Longos utiliza-se de um SGBD Relacional (SGBDR) para armazenar, tanto seus componentes espaciais como os alfanuméricos. Neste modo de trabalho são utilizados campos longos, denominados BLOBs³ para o armazenamento das informações espaciais. Desta forma, o controle da integridade de ambos, componentes espaciais e alfanuméricos, é feito pelo SGBDR (FERREIRA, 2003).

Os SGBD-R seguem o modelo relacional de dados, em que um banco de dados é organizado com uma coleção de relações cada qual com atributos de um tipo específico (CÂMARA et al, 2005, p. 183).

O uso dessa arquitetura tem como principais desvantagens:

- Um BLOB não possui semântica;
- Um BLOB não possui métodos de acesso;
- SQL oferece apenas operadores elementares de cadeias para tratar BLOBs.

SQL (Structured Query Language): linguagem padrão para manipular bancos de dados relacionais. Incluem recursos para definir estruturas de dados; consultar, inserir e modificar dados do banco de dados e especificar restrições de segurança (FERREIRA, 2003, p. 21).

Contudo, ao codificar-se dados espaciais em BLOBs, transfere-se à responsabilidade para o SIG de implementar os operadores espaciais obtendo a semântica dos dados, e métodos de acesso que possam ser úteis para a realização das consultas. (CÂMARA et al, 2005).

³ Campos binários longos, disponíveis em SGBDs Relacionais e SGBDs Objeto-Relacional. Que não contemplam uma variedade das operações presentes nestes SGBDs como: operações aritméticas, de conversão, de manipulação textual e operações com data (CÂMARA et al, 2005).

1.3.4 Arquitetura Integrada Baseada em Extensões Espaciais

A arquitetura integrada baseada em extensões espaciais consiste na utilização de extensões espaciais desenvolvidas sobre SGBDs Objeto – Relacional (SGBDOR), que definem funcionalidades e procedimentos capazes de armazenar, acessar e analisar dados espaciais (FERREIRA, 2003).

Os SGBD-OR estendem o modelo relacional, entre outras características, com um sistema de tipos de dados rico e estendível, oferecendo operadores que podem ser utilizados na linguagem de consulta. Possibilitam ainda a extensão dos mecanismos de indexação sobre os novos tipos (CÂMARA et al, 2005, p. 183).

Conforme Câmara et al 2005, esta arquitetura oferece as seguintes vantagens:

- Permite definir tipos de dados espaciais, equipados com operadores específicos;
- Permite definir métodos de acesso específicos para dados espaciais.

Segundo Güting apud Ferreira 2003, um SGBDOR que possui uma extensão para dados espaciais é chamado de Sistema de Banco de Dados Espaciais ou Geográficos e deve conter as seguintes características:

- Fornecer tipos de dados espaciais (TDEs), como ponto, linha e polígono, em seu modelo de dados e fornecer os meios para manipulá-los como qualquer outro atributo básico (string, inteiro, double);
- Estender a linguagem SQL para que se possa disponibilizar operações e consultas espaciais sobre TDEs;
- Adaptar outras funções para manipular TDEs eficientemente (Exemplo: indexação e junção espacial).

A utilização de extensões espaciais sobre SGBDOR é uma forma de evitar os problemas de armazenamento, contidos na utilização de BLOBs. Dentre suas vantagens pode-se citar os mecanismos de indexação, tanto para atributos numéricos quanto para espaciais e as funções previamente definidas para o tratamento dos dados espaciais (QUEIROZ, 2003).

Contudo, as extensões espaciais só tratam dados de representação vetorial, não dando suporte a dados matriciais e, portanto, existem escassos recursos para executar operações espaciais sobre estes.

1.3.5 Arquitetura Integrada Combinada

Para atender SIGs que manipulam dados representados tanto na forma vetorial quanto na matricial dispõe-se desta arquitetura.

A arquitetura integrada combinada é um misto da arquitetura integrada baseada em campos longos com a arquitetura integrada baseada em extensões espaciais. Com essa proposta as geometrias vetoriais são armazenadas utilizando a extensão espacial e as geometrias matriciais são armazenadas em BLOBs (CÂMARA et al, 2005).

2. MODELO DE DADOS GEOGRÁFICOS

O processo de abstração de objetos existentes no mundo real é de suma importância para o desenvolvimento de sistemas de informação. Isso também ocorre quando se está desenvolvendo um SIG, pois, além de ter de considerar o conjunto de dados descritivos de um objeto e seus relacionamentos com os demais, através da modelagem de dados, precisa-se designar uma representação para cada um deles. Esta representação deve ir ao encontro das finalidades da aplicação. Contudo, ocorre de aplicações distintas terem necessidade da mesma informação, mas em diferentes representações (DAVIS JR., 2000).

Conforme Davis Jr. 2000, em um SIG busca-se permitir que diferentes representações para um mesmo objeto convivam no mesmo banco de dados, se possível armazenando apenas uma delas e deduzindo as demais a partir desta representação primária.

Para determinar as necessidades de cada aplicação geográfica, com relação à representação de seus objetos, utiliza-se a modelagem de dados. Neste momento define-se qual modelo de dados será usado, para que ao fim do processo seja possível estabelecer um esquema de dados para a aplicação.

Um modelo de dados é um conjunto de conceitos que podem ser usados para descrever a estrutura e as operações em um banco de dados (ELMASRI e NAVATHE, apud CÂMARA et al, 2005, p. 93).

Câmara et al 2005, informa que modelos de dados para aplicações geográficas possuem necessidades adicionais, tanto com relação à abstração de conceitos e entidades quanto ao tipo de entidades representáveis e seus inter-relacionamentos. E cita diversas propostas para modelagem que buscam contemplar essas necessidades: GeoOOA, MODUL-R, GMOD, IFO, GISER, OMT-G, GeoFrame e MADS.

Modelos de dados geográficos variam de acordo com o nível de abstração. Para aplicações geográficas, são considerados quatro níveis distintos (Câmara et al, 2005), ilustrados na Figura 5:

- Nível do mundo real: Possui os fenômenos geográficos reais que devem ser representados. Exemplo: rios, ruas e lotes de terra;
- Nível de representação conceitual: Proporciona um conjunto de conceitos formais com os quais as entidades geográficas podem ser modeladas da forma como foram compreendidas pelo usuário, em um alto nível de abstração. Neste nível são definidas as classes básicas, contínuas ou discretas, que serão criadas no banco de dados;
- Nível de apresentação: Proporciona ferramentas para determinar os diferentes aspectos visuais que as entidades geográficas têm de assumir ao longo de seu uso em aplicações;
- Nível de implementação: Estipula padrões, formas de armazenamento e estruturas de dados para implementar fisicamente cada tipo de representação, os relacionamentos entre elas e as funções e métodos necessários.



Figura 5 - Níveis de Abstração de Aplicações Geográficas

Fonte: DAVIS JR., 2000, p. 14.

O presente trabalho apresentará duas possíveis propostas para modelagem de dados espaciais: OMT-G e a GeoFrame.

2.1 OMT-G

O modelo OMT-G parte das premissas definidas para o diagrama de classes da *Universal Modeling Language* (UML) (Rati97, apud Davis Jr., 2000), inserindo primitivas geográficas com o objetivo de maximizar a capacidade de representação semântica daquele modelo, minimizando desta forma a distância entre a concepção mental do espaço a ser modelado e o modelo de representação usual. Logo, o modelo OMT-G fornece primitivas para modelar a geometria e a topologia dos dados geográficos, oferecendo suporte a estruturas topológicas, estruturas de rede, múltipla representação de objetos e relacionamentos espaciais. O modelo, ainda permite a especificação de atributos alfanuméricos e métodos associados para cada classe. Seus principais diferenciais são a sua expressividade gráfica e sua capacidade de representação, uma vez que anotações textuais são substituídas pelo desenho de

relacionamentos explícitos, que representam a interação entre os diversos objetos espaciais e não espaciais (DAVIS JR., 2000).

O modelo OMT-G é fundamentado em três conceitos: classes, relacionamentos e restrições de integridade espaciais. Classes e relacionamentos definem os princípios básicos usados para criar diagramas de classes. A identificação de restrições de integridade espacial é uma atividade de suma importância no projeto de um SIG, e consiste na identificação de condições que necessitam ser garantidas para que o banco de dados esteja íntegro (CÂMARA et al, 2005).

2.1.1 Classes

As classes estipuladas pelo modelo OMT-G representam três principais grupos de dados (contínuos, discretos e não espaciais) que podem ser encontradas em aplicações geográficas, ofertando deste modo uma visão integrada do espaço modelado. Suas classes podem ser georreferenciadas ou convencionais (DAVIS JR., 2000).

A diferença entre classes convencionais e georreferenciadas permite que aplicações distintas compartilhem dados não espaciais, facilitando, desta forma, o desenvolvimento de aplicações integradas e a reutilização de dados. Uma classe georreferenciada apresenta um conjunto de objetos que têm representação espacial e que são associados a elementos reais localizados na terra (CÂMARA, 1995), adotando a proposta de campos e objetos. Uma classe convencional apresenta um conjunto de objetos com propriedades, comportamentos, relacionamentos e semânticas semelhantes, estes podem possuir algum tipo de relacionamento com objetos espaciais, não tendo, no entanto, propriedades geométricas (CÂMARA, 2005).

As classes georreferenciadas são especializadas em (Davis Jr. 2000):

- Geo-campos: são aquelas que representam objetos e fenômenos que são distribuídos ininterruptamente no espaço. Exemplo: Tipo de solo, relevo e tipo de vegetação;

- Geo-objetos: são aquelas que representam objetos geográficos particulares, individualizáveis e associados a elementos do mundo real. Exemplo: Edifícios, rios e árvores.

As classes convencionais são simbolizadas exatamente como na UML e as classes georreferenciadas são simbolizadas no modelo OMT-G, de maneira semelhante (Figura 6), incluindo no canto superior esquerdo um retângulo que é usado para denotar a forma geométrica da representação. Nos dois casos símbolos simplificados podem ser usados. Os objetos podem possuir ou não atributos não espaciais associados, listados no centro da representação completa. Métodos ou operações associadas são especificados na parte inferior da representação completa (CÂMARA, 2005).

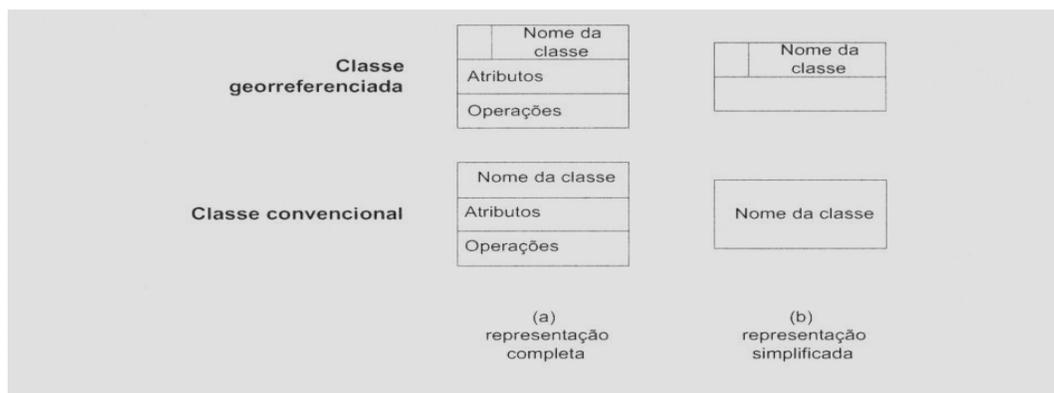


Figura 6 - Notação gráfica para as classes do modelo OMT-G

Fonte: CÂMARA et al, 2005, p. 101.

O modelo OMT-G oferece um conjunto fixo de alternativas de representação geométrica, usando uma simbologia que estabelece distinção entre geo-objetos (Figura 8) e geo-campos (Figura 7) (DAVIS JR., 2000).

O modelo OMT-G determina cinco classes descendentes de geo-campos: isolinhas, subdivisão planar, tesselação, amostragem e malha triangular (*triangulated irregular network, TIN*)(Figura 7), e duas classes descendentes de geo-objeto: geo-objeto com geometria e geo-objeto com geometria e topologia (Figura 8) (CÂMARA, 2005).



Figura 7 - Geo-campos

Fonte: CÂMARA et al, 2005, p. 101.

A classe geo-objeto com geometria representa objetos que tem somente propriedades geométricas, e é especializada em classes: ponto, linha e polígono e a classe geo-objeto com geometria e topologia representa objetos que tem, além das propriedades geométricas, propriedades de conectividade topológica, sendo exclusivamente voltada para a representação de estruturas em rede, tais como sistemas de esgoto e ou fornecimento de energia elétrica.

Essas propriedades de conectividade topológica estão presentes em classes descendentes que representam nós e arcos, da forma usualmente adotada na teoria de grafos. Os arcos podem ser unidirecionais, como redes de fornecimento de água, ou bidirecionais, como em redes de telecomunicações. Assim, as especializações previstas são denominadas nó de rede, arco unidirecional e arco bidirecional. O foco do modelo OMT-G com analogia a redes não está concentrado na implementação do relacionamento entre seus elementos, mas sim na semântica da conexão entre elementos de rede, que é um fator essencial para a composição de regras que garantam a integridade dos dados (DAVIS JR., 2000).

A Teoria dos Grafos é o ramo da matemática que estuda as propriedades de grafos. Um grafo é um conjunto de pontos, chamados vértices (ou nodos), conectados por linhas, chamadas de arestas (ou arcos) (WIKIPEDIA, 2005a).

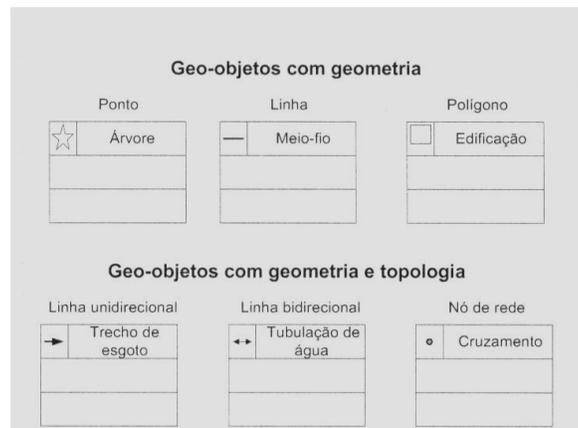


Figura 8 - Geo-objetos

Fonte: CÂMARA et al, 2005, p. 102.

2.1.2 Relacionamentos

O modelo OMT-G representa três tipos de relacionamento que podem ocorrer entre duas classes, que são (Davis Jr. 2000):

- Associação simples: representam relacionamentos entre classes distintas, convencionais ou georreferenciadas. São apontados por linhas contínuas (Figura 9);
- Relacionamentos espaciais: são aquelas que representam relações topológicas, métricas, ordinais e *fuzzy*⁴. São apontados por linhas pontilhadas (Figura 9);
- Relacionamentos topológicos em rede: são relacionamentos entre objetos que estão conectados um com o outro. São apontados por linhas pontilhadas paralelas (Figura 9).

⁴ “Uma abordagem de lógica nebulosa (*fuzzy logic*), que pode associar um *pixel* a várias classes temáticas através de graus de pertinência[..]” (Brule, apud MAS, 1996).

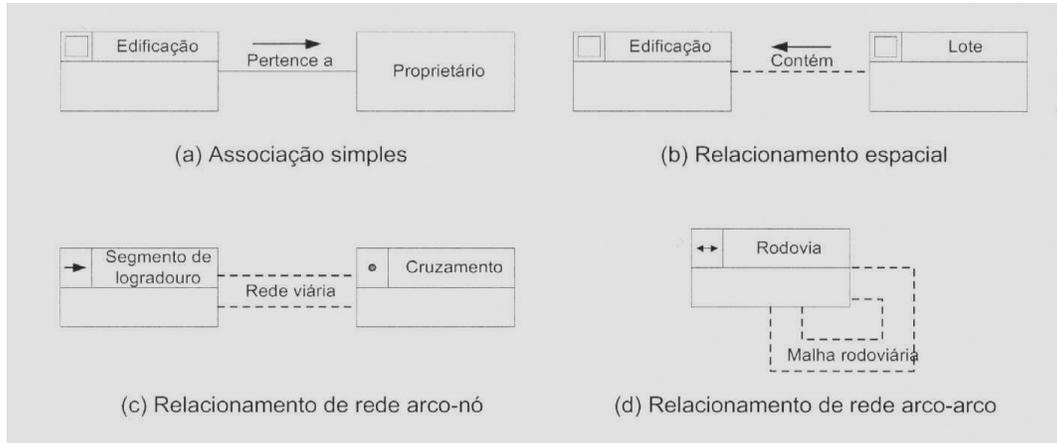


Figura 9 - Relacionamentos

Fonte: DAVIS JR., 2000, p. 19.

Os relacionamentos também são caracterizados por sua cardinalidade. A Figura 10 ilustra a notação de cardinalidade, que segue o padrão UML.

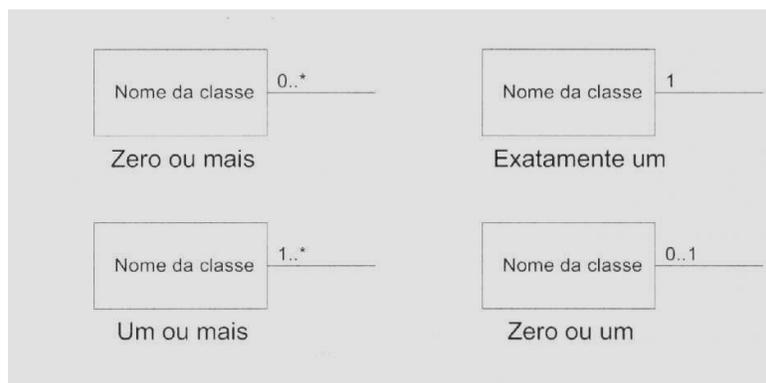


Figura 10 – Cardinalidade

Fonte: DAVIS JR., 2000, p. 19.

2.1.3 Generalização e Especialização

No modelo OMT-G, as abstrações de generalização e especialização se aplicam tanto a classes georreferenciadas quanto a classes convencionais, conforme definição da UML, onde um triângulo conecta a superclasse as suas subclasses. Cada generalização pode ter um discriminador, que indica qual propriedade ou característica está sendo abstraída pelo relacionamento de generalização. (DAVIS JR., 2000). A Figura 11 demonstra a notação UML e a Generalização Espacial.

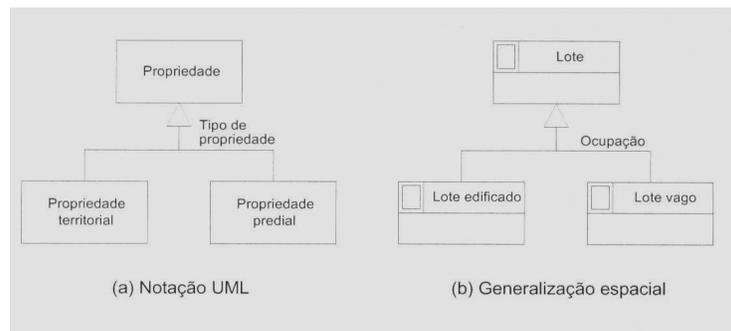


Figura 11 - Generalização

Fonte: DAVIS JR., 2000, p. 20.

As generalizações podem ser totais ou parciais. Uma generalização é total quando a união de todas as instâncias das subclasses é igual ao agrupamento total de instâncias da superclasse. A UML representa a totalidade com o uso de elementos de restrição, que são completo e incompleto. Já no modelo OMT-G foi adotada a colocação de um ponto no cume do triângulo para demonstrar a totalidade (Figura 12). (DAVIS JR., 2000).

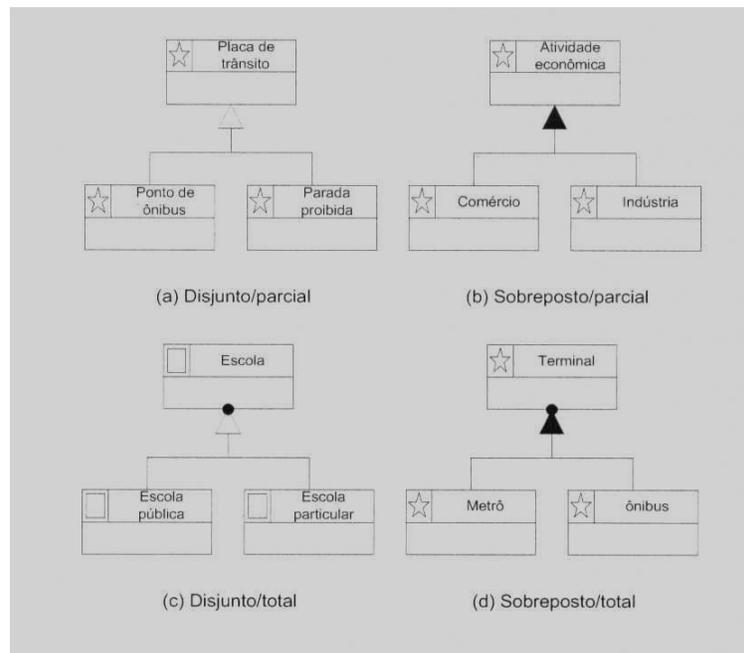


Figura 12 - Exemplos de Generalização espacial

Fonte: DAVIS JR., 2000, p. 20.

O modelo OMT-G possui uma notação para os elementos de restrição predefinida disjuncto⁵ e sobreposto⁶ da UML, ou seja, em uma generalização disjuncta o triângulo fica em branco e em uma generalização sobreposta ele é preenchido (Figura 12). (RBP, apud DAVIS JR., 2000, p. 20).

A combinação de disjunção e totalidade gera quatro tipos possíveis de restrição para generalização e especialização que são: Disjuncto parcial, Disjuncto total, Sobreposto parcial e Sobreposto total.

⁵ Disjuncto é quando existe a possibilidade de que uma subclasse possa ser enquadrada em outro tipo de subclasse numa determinada generalização ou especialização. Um exemplo seria se houvesse uma generalização onde houvesse uma superclasse Pessoa e subclasses Aluno, Professor e Funcionário, onde, temos a possibilidade de um aluno possa ser também um funcionário.

⁶ Sobreposto é o inverso de disjuncto não existe a possibilidade de uma subclasse ser enquadrada em outro tipo de subclasse.

2.1.4 Agregação

A notação para agregação utilizada no modelo OMT-G segue a UML (Figura 13). Uma agregação pode acontecer entre classes convencionais, entre classes georreferenciadas ou entre uma classe convencional e uma classe georreferenciada (Figura 14). Ao ocorrer uma agregação entre classes georreferenciadas, é necessária a utilização de agregação espacial (Figura 15). (DAVIS JR., 2000).

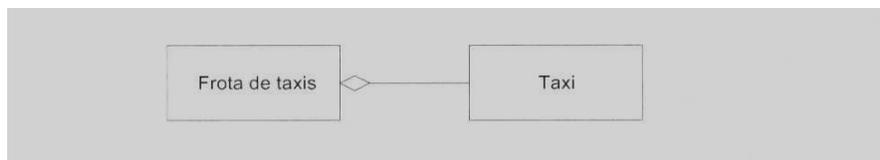


Figura 13 - Agregação notação UML

Fonte: DAVIS JR., 2000, p. 21.

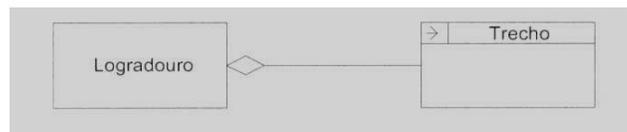


Figura 14 - Agregação entre uma classe convencional e uma classe georreferenciada

Fonte: DAVIS JR., 2000, p. 20.

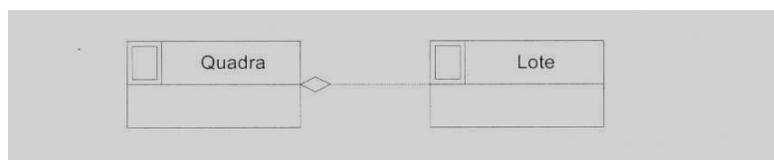


Figura 15 - Agregação espacial “todo-parte”

Fonte: DAVIS JR., 2000, p. 20.

Na agregação espacial é exposto o relacionamento topológico “todo-parte”. A agregação espacial denota que a geometria de cada parte deve estar contida na geometria do todo. (CÂMARA et al, 2005).

2.2 GeoFrame

O GeoFrame é um *framework* conceitual que disponibiliza um diagrama de classes básicas para auxiliar o projetista na modelagem conceitual de dados geográficos e na estipulação de padrões de análise em bancos de dados geográficos. (Lisboa Filho e Iochpe, apud LISBOA FILHO et al, 1999).

[...] um framework é uma estrutura de suporte definida em que um outro projeto de software pode ser organizado e desenvolvido. Tipicamente, um framework pode incluir programas de apoio, bibliotecas de código, linguagens de script e outros softwares para ajudar a desenvolver e juntar diferentes componentes do seu projeto. (WIKIPEDIA, 2005b).

O GeoFrame foi criado conforme as regras formais da orientação a objetos, utilizando a notação de diagrama de classes definida na linguagem UML. A Figura 16 mostra o diagrama de Classes do GeoFrame.(LISBOA FILHO et al, 2000a).

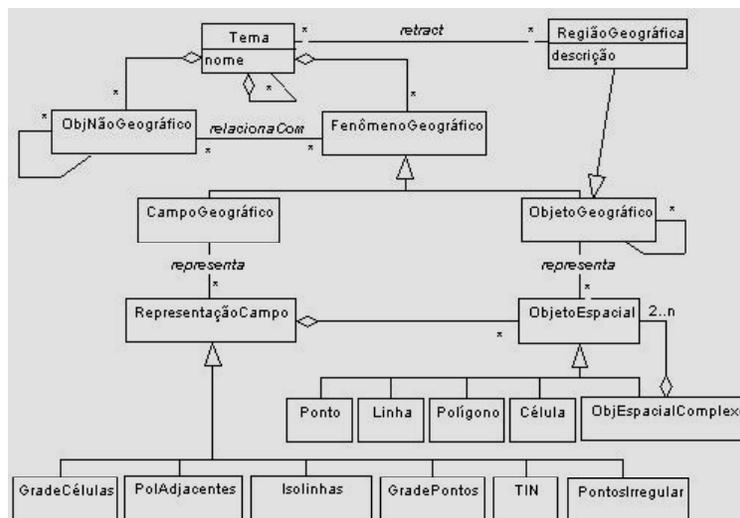


Figura 16 - Diagrama de Classes do GeoFrame.

Fonte: LISBOA FILHO et al, 2000b.

Descrição das classes do modelo GeoFrame(Lisboa Filho et al, 2002):

- *Tema e Região Geográfica*: Forma a base para a aplicação geográfica. Em cada região geográfica pode-se especificar uma coleção de temas⁷. O conjunto de classes que descrevem os fenômenos geográficos em temas funciona como um mecanismo para redução da complexidade em grandes esquemas de dados;
- *Objeto Não Geográfico*: Classe para objetos não geográficos, objetos que não possuem uma referência à posição geográfica dão origem as subclasses da classe *Objeto Não Geográfico*;
- *Fenômeno Geográfico*: A classe *Fenômeno Geográfico* generaliza qualquer fenômeno cuja localização em relação á superfície terrestre seja considerada;
- *Campo Geográfico*: A classe *Campo Geográfico* generaliza os fenômenos que se enquadram na visão de campo⁸;
- *Objeto Geográfico*: A classe *Objeto Geográfico* é uma generalização de todas as classes do domínio de aplicação que são percebidas na visão de objetos⁹;
- *Objeto Espacial*: A classe *Objeto Espacial* generaliza as classes necessárias para a especificação do componente espacial dos fenômenos geográficos percebidos na visão de objetos;

⁷ Conjunto de objetos geográficos, com descrições e formas comuns. Por exemplo, o conjunto de postes telefônicos de uma determinada cidade forma o tema “postes telefônicos”. (SILVA, 2002).

⁸ Idem a geo-campos, seção 2.1.1.

⁹ Idem a geo-objetos, seção 2.1.1.

- *RepresentaçãoCampo*: A classe *RepresentaçãoCampo* generaliza as classes necessárias para a especificação do componente espacial dos fenômenos geográficos percebidos na visão de campos.

3. EXTENSÕES ESPACIAIS

A utilização de extensões em SGBDOR estende o modelo de dados relacional fornecendo um sistema mais rico, acrescentando dados complexos e novas estruturas às linguagens de consulta, como a SQL, para tratar os dados acrescidos, onde, antes não poderiam ser trabalhados em um SGBD.(SILBERCHATZ et al, 1999).

Para o presente trabalho serão apresentadas as características e funcionalidades das extensões espaciais da Oracle, denominada Oracle Spatial, do PostgreSQL, denominada PostGIS e do MySQL incorporada recentemente na versão 4.1.

3.1 OpenGIS

Conforme Lakowski 1998, o mercado mundial de geoprocessamento se desenvolveu sem a preocupação de integração de informações geográficas entre sistemas. Formatos e estruturas de dados incompatíveis têm dificultado a integração de informações geográficas e limitado o uso da tecnologia. A indústria de Geoprocessamento, pressionada pela demanda da comunidade usuária, resolveu unir-se para avaliar a questão e projetar o futuro da tecnologia.

O projeto OpenGIS teve início em 1993, amparado por órgãos públicos e empresas comerciais que se aliaram para criar e fazer propostas técnicas de um formato aberto para softwares de geoprocessamento. O OGC (OpenGIS Consortium) foi

fundado em agosto de 1994 para desenvolver as especificações técnicas e criar um formato padrão para garantir a interoperabilidade entre sistemas.

Segundo Bogorny 1999, a proposta da OpenGIS, especificação abstrata, é criar e documentar um modelo conceitual que permita a criação de especificações de implementação.

A especificação abstrata consiste de dois modelos: o primeiro modelo é o essencial que representa os fatos do mundo real e o segundo é o abstrato que representa a descrição de como o software de SIG funcionará. (OGC99, apud BOGORNY, 1999).

Maiores detalhes sobre OpenGIS e a especificação abstrata podem ser encontrados em Bogorny 1999.

3.1.1 Modelo Essencial

O primeiro modelo da especificação abstrata OpenGIS, conforme citado anteriormente, é o essencial¹⁰. O mais simples entre os modelos concebe os conceitos do mundo real, ou seja, descreve o mundo como é. Expressando os objetos e comportamentos do mundo real como são.

3.1.2 Modelo Abstrato

O Modelo Abstrato é aquele que determina como representar os conceitos definidos no modelo essencial na implementação do software. Considera-se o modelo de especificação da implementação OpenGIS.

¹⁰ Para mais informações sobre o modelo essencial, <http://www.opengis.org/techno/guide.html>. (BOGORNY, 1999).

A OGC divide o modelo em dezessete tópicos, incluso o tópico zero que representa o relacionamento entre os demais. Para cada tópico é descrito o modelo essencial, contudo, será apresentado somente o modelo abstrato de forma sucinta.

Os tópicos e sua descrição seguem abaixo (Bogorny, 1999):

- *Feature Geometry* (Geometria de Feições): representação abstrata dos fenômenos do mundo real, associada a uma localização relativa a terra;
- *Spatial Reference Systems* (Sistemas de Referência Espacial): sua função é associar localizações no espaço a tuplas de coordenadas geométricas num espaço matemático e associar valores de coordenadas e geometrias a localizações no mundo real;
- *Locational Geometry Structures* (Estruturas Geométricas Locais): sua principal atribuição é suportar especificações geométricas de feições simples e seus sistemas de referência espacial;
- *Stored Functions and Interpolation* (Interpolação e Funções de armazenamento): funções definidas uma única vez e utilizadas em todos os pontos de um domínio;
- *The OpenGIS Feature* (Feições OpenGIS): feições definidas pelo OpenGIS;
- *The Coverage Type and its Subtypes* (O tipo Coverage e seus subtipos): São subtipos de feições, na especificação OpenGIS determinam-se como uma função que pode retornar um valor para um ponto geométrico;
- *The Earth Imagery Case* (Casos das Imagens da Terra): apresenta modelos essenciais e abstratos para a tecnologia que é usada amplamente através de SIG;

- *Relationship Between Features* (Relação entre feições): introduz uma abstração dos relacionamentos entre entidades do mundo real;
- *Quality* (Qualidade): especifica extensões para suportar a qualidade de objetos feições, geometria e coverage;
- *Features Collections* (Coleções de Feições): não há consenso entre a necessidade de coleções de feições, por enquanto, não foi determinado o modelo abstrato;
- *Metadata* (Metadados): associação de metadados a cada coleção de feições;
- *OpenGIS Service Architecture* (Arquitetura de Serviços OpenGIS): fornece a definição das partes essenciais de uma visão computacional de um sistema de informação para processamento geoespacial;
- *Catalog Services* (Serviços de Catálogo): a OGC usa o termo catálogo para descrever um conjunto de interfaces de serviços que suportam organização, descoberta e acesso a informações geoespaciais;
- *Semantics and Information Communities* (Semântica e Comunidades de Informação): criada para habilitar grupos (Exemplo: Ecologistas e Engenheiros Civis) a administrar eficientemente a semântica de seus dados geográficos, criando semânticas distintas;
- *Image Exploitation Services* (Serviço de Exploração de Imagens): descreve a categoria dos serviços de exploração de imagens necessárias;
- *Image Coordinate Transformation Services* (Serviço de Transformação de Coordenadas de Imagens): apresenta um modelo abstrato de serviços para transformação de coordenadas de imagens.

3.2 PostGIS

O PostGIS é um módulo de extensão espacial para o SGBDOR PostgreSQL que permite armazenar, recuperar e analisar dados espaciais, freqüentemente utilizados em um SIG. Foi desenvolvido pela Refrations Research Inc¹¹, como uma tecnologia de banco de dados espacial e vem sendo mantida pela mesma. (POSTGIS, 2005).

3.2.1 Objetos Espaciais

Os objetos espaciais que o PostGIS presta suporte são as feições¹² definidas pelo consórcio OpenGIS (OGC). Atualmente, ele suporta as características simples e a representação de APIs¹³, mas não os operadores de comparação e convolução¹⁴ definidos pela especificação de implementação de feições para SQL do OpenGIS. (POSTGIS, 2005).

O PostGIS lida com objetos espaciais dos tipos (QUEIROZ, 2003 e BÜCHNER, 2002):

- *point* (ponto): é uma geometria que representa uma simples localização no espaço, com coordenadas, e que não possui dimensão. Um ponto deve ter as coordenadas x e y, mas também pode ter a coordenada z. Exemplo de representação textual com coordenadas x e y: POINT(0 0);
- *linestring* (linha): é um objeto geométrico de uma única dimensão, armazenado como uma seqüência de pontos, com interpolação linear entre os pontos. Cada par consecutivo de pontos define um segmento da linha. Exemplo de representação textual com coordenadas x e y: LINESTRING(0 0, 1 1, 1 2), indica que a linha inicia no ponto de

¹¹ A Refrations é uma companhia de consultoria em GIS e banco de dados localizada em Victoria, British Columbia, Canadá, especializada em integração de dados e desenvolvimento de software. (POSTGIS, 2005).

¹² Feições são uma representação abstrata de fenômenos do mundo real, associada a uma localização relativa a terra. (BOGORNÝ, 1999).

¹³ API: Application Programming Interface. Interface entre o computador e os programas do usuário.

¹⁴ A convolução é uma técnica de processamento de imagens.

coordenadas POINT(0 0), passa pelo ponto POINT(1 1) e termina no ponto de coordenadas POINT(1 2);

- *polygon* (polígono): é um objeto geométrico com duas dimensões e plano, definido por um limite exterior e zero ou mais limites interiores. Cada limite interior define uma abertura no polígono. Polígonos devem ser topologicamente fechados, ou seja, as coordenadas do ponto inicial do polígono devem ser idênticas às do ponto final. Exemplo de representação textual, com coordenadas x, y e z, formado por um limite exterior e um limite interior: POLYGON((0 0 0, 4 0 0, 4 4 0, 0 4 0, 0 0 0), (1 1 0, 2 1 0, 2 2 0, 1 2 0, 1 1 0)), indica que o polígono exterior começa no ponto POINT(0 0 0) e termina na coordenada POINT(0 0 0), este polígono, conforme exemplo, possui um polígono interno que tem suas coordenadas iniciais e finais no ponto POINT(1 1 0);
- *multipoint* (multiponto): é uma coleção de elementos do tipo point (ponto). Exemplo de representação textual, com coordenadas x, y e z formado por dois pontos: MULTIPOINT(0 0 0, 1 2 1) ;
- *multilineestring*: é uma coleção de elementos do tipo linestring (linha). Exemplo de representação textual, com coordenadas x, y e z formado por dois objetos do tipo linestring: MULTILINESTRING((0 0 0 , 1 1 0, 1 2 1), (2 3 1, 3 2 1, 5 41));
- *multipolygon*: é uma coleção de elementos do tipo polygon (polígono). Exemplo de representação textual, com coordenadas x, y e z, constituído de dois polígonos, sendo que o primeiro possui um limite exterior e outro interior, e o segundo apenas um limite exterior: MULTIPOLYGON(((0 0 0, 4 0 0, 4 4 0, 0 4 0, 0 0 0), (1 1 0, 2 1 0, 2 2 0, 1 2 0, 1 1 0)), ((-1 -1 0, -1 -2 0, -2 -2 0, -2 -1 0, -1 -1 0)));
- *geometrycollection* (coleção de geometrias): é uma geometria formada por um conjunto de um ou mais tipos de objetos espaciais. Exemplo de representação textual, com coordenadas x, y e z, formado por dois

objetos distintos, um do tipo point e outro linestring:
GEOMETRYCOLLECTION(POINT(2 3 9), LINESTRING (2 3 4,
345));

Para tornar o PostGIS completamente compatível com o padrão OGC, além de utilizar as funções e especificações abstratas, a OGC orienta que o banco de dados deve possuir uma tabela para armazenar os dados espaciais, denominada GEOMETRY_COLUMNS, e outra para carregar os identificadores numéricos e descrições textuais de sistemas de coordenadas usados no banco de dados espacial, denominada SPATIAL_REF_SYS. (POSTGIS, 2005).

3.2.2 Sistema de Coordenadas

A definição para sistema de coordenadas conforme CAM 96 apud Lisboa Filho 2002, é a seguinte:

O sistema de coordenadas permite definir a localização de qualquer elemento sobre a superfície terrestre. Os sistemas mais empregados são os sistemas de coordenadas geográficas (ou terrestres) e os sistemas de coordenadas planas (ou cartesianas).

Num sistema de coordenadas geográficas, cada ponto é determinado a partir de um par de coordenadas da intersecção de um meridiano com um paralelo. Os meridianos são círculos da esfera terrestre que passam pelos pólos norte e sul. Estes são medidos em longitude, com início no meridiano de Greenwich, com valores que variam de 0° até +180° de longitude a leste e de 0° até -180 de longitude oeste. O paralelo do Equador determina as medidas de latitude, ele divide a terra em dois hemisférios: norte e sul, determinando as medidas de latitude como sendo 0° no paralelo do Equador, +90° no pólo Norte e -90° no pólo Sul. (LISBOA FILHO, 2002).

Os sistemas de coordenadas planas são formados por um par de eixos perpendiculares, onde a intersecção dos eixos representa a localização de qualquer ponto sobre determinado plano. Nestes sistemas, as coordenadas são representadas por valores (x, y) concebendo o valor do ponto sobre cada um dos eixos. Normalmente o

eixo horizontal é associado para a longitude, enquanto que o vertical fica associado a latitude, o que torna possível a conversão entre os sistemas de coordenadas, a partir de transformações matemáticas.(LISBOA FILHO, 2002).

3.2.3 Tabela GEOMETRY_COLUMNS

A tabela GEOMETRY_COLUMNS é a tabela que mantém a descrição das feições que serão armazenadas no banco de dados. A definição da tabela, conforme padrão OGC, é a seguinte (BÜCHNER, 2002):

```
CREATE TABLE GEOMETRY_COLUMNS (
  F_TABLE_CATALOG VARCHAR (256) NOT NULL,
  F_TABLE_SCHEMA VARCHAR (256) NOT NULL,
  F_TABLE_NAME VARCHAR (256) NOT NULL,
  F_GEOMETRY_COLUMN VARCHAR (256) NOT NULL,
  COORD_DIMENSION INTEGER NOT NULL,
  SRID INTEGER NOT NULL,
  TYPE VARCHAR (30) NOT NULL
)
```

As colunas da tabela são definidas da seguinte forma (POSTGIS, 2005):

- F_TABLE_CATALOG, F_TABLE_SCHEMA, F_TABLE_NAME: estas três colunas qualificam e denominam a tabela de feições correspondente, a qual contém uma coluna GEOMETRY. As condições “catalog” e “schema” são derivadas do Oracle. Não existe análogo de “catalog” no PostgreSQL, desta maneira a coluna é deixada em branco. Para “schema”, é usado o nome do banco de dados (tem como valor default public), a coluna “name” é reservada para o nome da tabela correspondente;
- F_GEOMETRY_COLUMN: esta coluna armazena o nome da coluna GEOMETRY na tabela de feições correspondente;

- COORD_DIMENSION: armazena a dimensão espacial das feições a serem armazenadas, que podem ter duas ou três dimensões;
- SRID: identificador do sistema de referência espacial usado para a coordenada geométrica das feições. É uma chave estrangeira para a tabela SPATIAL_REF_SYS;
- TYPE: coluna que armazena o tipo de objeto espacial da forma (point, linestring, polygon, multipolygon, multilinestring, multipolygon, geometriccollection). Este último atributo não faz parte da especificação OGC, contudo, é requerido para assegurar a homogeneidade das feições.

3.2.4 Tabela SPATIAL_REF_SYS

A tabela SPATIAL_REF_SYS carrega os identificadores numéricos e descrições textuais dos sistemas de coordenadas usadas no banco de dados espacial. A definição da tabela, conforme padrão OGC, é como segue (POSTGIS, 2005):

```
CREATE TABLE SPATIAL_REF_SYS (
  SRID INTEGER NOT NULL PRIMARY KEY,
  AUTH_NAME VARCHAR (256),
  AUTH_SRID INTEGER,
  SRTEXT VARCHAR (2048),
  PROJ4TEXT VARCHAR (2048)
)
```

As colunas são definidas da seguinte forma (SIMON & CHRISTL, 2005):

- SRID: Inteiro para identificação do Spatial Reference System¹⁵;
- AUTH_NAME: Nome atribuído ao sistema de coordenadas;

¹⁵ “O sistema de referência espacial (SRS) é um parâmetro textual que indica qual sistema de codificação de referência de coordenadas horizontais.” (PEREIRA, 2004).

- AUTH_SRID: Código dado pelo órgão que mantém o sistema de coordenadas;
- SRTEXT: Informação textual que representa o sistema de coordenadas;
- PROJ4TEXT: Coluna que contém a definição do sistema no formato reconhecido pelo Proj4¹⁶.

3.2.5 Índices

Índices são estruturas de dados criadas para maximizar o desempenho de acesso aos bancos de dados que, na maioria das vezes, estão armazenados em memória secundária. Em bancos de dados convencionais utilizam-se índices que aceleram o processamento da consulta, com estruturas de dados dinâmicas, como tabelas Hash e árvores B-Tree, tendo como chave um ou mais campos. Já para bancos de dados espaciais, esses índices não são aplicáveis, devido aos dados espaciais serem multidimensionais. Os índices espaciais consideram a localização dos objetos no espaço e processam consultas espaciais de forma eficiente, minimizando o conjunto de dados a ser verificado no processamento. (SILVA, 2002).

O PostgreSQL implementa três tipos de índices através do seu padrão que são: índices B-Tree, índices R-Tree e índices GIST.(POSTGIS, 2005).

Estes índices são definidos em Simon & Christl (2005) como:

- B-Tree: são índices binários que podem ser utilizados em comparações ou em consultas que os dados possam ser ordenados;
- R-Tree: é um índice específico para objetos espaciais, ele divide o espaço em retângulos consequentemente até que o menor retângulo para a forma seja encontrado;

¹⁶ Proj4 é uma biblioteca, incorporada ao PostGIS, que fornece funções para transformações de coordenadas.(POSTGIS, 2005).

- GIST: é um índice que se baseia em árvores genéricas de dados, dividindo os dados através de suas relações.

Segundo PostGIS (2005), o uso de B-Tree para indexação espacial é inadequada, pois, não há como mensurar os dados ao longo de um dado eixo (Ex.: qual dos pontos é maior, (0, 0), (0, 1) ou (1, 0)?).

O uso de R-Tree e GIST são as formas adequadas para indexação de dados espaciais. A utilização de R-Tree para o PostgreSQL, não é recomendada, pois sua implementação não é tão robusta para pesquisas espaciais. (SIMON & CHRISTL, 2005).

O PostGIS trabalha com um índice R-Tree implementado sob GIST para indexar seus dados espaciais.

3.2.6 Consultas Espaciais

Conforme Câmara et al 2005, um grande destaque desta extensão é o grande número de operadores espaciais disponíveis, dentre eles cita-se:

Operadores topológicos:

- *equals* (geometry, geometry): retorna verdadeiro se as duas geometrias passadas como parâmetro forem iguais;
- *disjoint* (geometry, geometry) : retorna verdadeiro se uma geometria está desconexa de outra;
- *intersects* (geometry, geometry): retorna verdadeiro se há intersecção entre as duas geometrias;
- *touches* (geometry, geometry): retorna verdadeira se as duas geometrias se tocam;

- *crosses* (geometry, geometry): retorna verdadeiro se a cruzamento espacial entre duas geometrias;
- *within* (geometry, geometry): retorna verdadeiro se uma geometria está dentro de uma outra geometria;
- *overlaps* (geometry, geometry): retorna verdadeiro se uma geometria sobrepõe outra geometria;
- *contains* (geometry, geometry): retorna verdadeiro se uma geometria é conteúdo espacial de outra geometria;
- *relate* (geometry, geometry): retorna a matriz de intersecção.

Operadores de construção de mapas de distância:

- *buffer* (geometry, double, [integer]): Retorna uma geometria que representa todos os pontos, da geometria recebida como parâmetro, que tem distancia menor ou igual ao parâmetro numérico recebido.

Operador para construção do Fecho Convexo:

- *convexhull* (geometry): retorna uma geometria que representa a casca convexa dessa geometria.

Operadores de conjunto:

- *interscetion* (geometry, geometry): retorna uma geometria que representa a intersecção do ponto fixo desta geometria com outra geometria;

- *geomUnion* (geometry, geometry): Retorna uma geometria que é a união do ponto fixo desta geometria com outra geometria;
- *symdifference* (geometry, geometry): retorna uma geometria que representa a diferença simétrica do ponto fixo desta geometria com outra geometria;
- *difference* (geometry, geometry): retorna uma geometria que representa a diferença do ponto fixo desta geometria com outra geometria.

Operadores métricos:

- *distance* (geometry, geometry): retorna a distância entre duas geometrias;
- *area* (geometry): retorna a área de uma determinada geometrias, se for um polígono ou multipolígono.

Centróide de geometrias:

- *centroid* (geometry): retorna o centróide¹⁷ da geometria como um ponto.

Validação (verifica se a geometria tem auto-intersecções):

- *isSimple* (Geometry): retorna verdadeiro se não tem nenhum ponto de intersecção própria ou tangência própria.

O suporte para utilização destes operadores espaciais é feito através da integração do PostGIS com a biblioteca GEOS (Geometry Engine Open Source)¹⁸. Essa

¹⁷ O centróide de uma geometria representa o centro do retângulo envolvente dos dados espaciais (*Bounding Box*) daquela geometria. (BÜCHNER, 2002).

biblioteca é uma tradução da API Java JTS para a linguagem C++. (CÂMARA et al, 2005).

3.3 Oracle Spatial

O Oracle Spatial é a extensão espacial do SGBDOR Oracle. Esta extensão contém um conjunto de funções e procedimentos que permitem armazenar, acessar e analisar dados espaciais em um banco de dados Oracle. (QUEIROZ, 2003).

O modelo de dados em que se baseia o Oracle Spatial consiste em uma estrutura hierárquica de elementos, geometrias e *layers*, em que *layers* são formados por um conjunto de geometrias, que por sua vez constituem-se de um conjunto de elementos. Um elemento pode ser do tipo ponto, linha ou polígono. Uma geometria pode ser composta por um ou mais elementos iguais ou não. (FERREIRA, 2003).

3.3.1 Objetos Espaciais

O Oracle Spatial suporta três tipos primários de geometrias: ponto (*point*), linha (*linestring*) e polígono (*polygon*). Além destes, aceita uma coleção de geometrias derivadas desses tipos primitivos, tais como: arcos circulares, círculos, linhas compostas e polígonos compostos. (SILVA, 2002).

Com base no modelo objeto-relacional, o Oracle Spatial define um objeto, para armazenar seus dados espaciais, denominado SDO_GEOMETRY. (QUEIROZ, 2003).

3.3.2 SDO_GEOMETRY

O objeto SDO_GEOMETRY contém a geometria em si, suas coordenadas e informações sobre seu tipo e projeção¹⁹. Em uma tabela espacial seus atributos são definidos como colunas de tipos básicos (NUMBER, VARCHAR2...) e a geometria,

¹⁸ <http://geos.refrations.net/>

¹⁹ Projeção cartográfica é a forma como se representa a superfície curva da terra em uma superfície plana, através da utilização de métodos matemáticos. (LISBOA FILHO, 2002).

como uma coluna de tipo SDO_GEOMETRY. (FERREIRA, 2003). A definição do objeto segue abaixo:

```
CREATE TYPE SDO_GEOMETRY AS OBJECT (
  SDO_GTYPE          NUMBER,
  SDO_SRID           NUMBER,
  SDO_POINT          SDO_POINT_TYPE,
  SDO_ELEME_INFO    SDO_ELEM_INFO_ARRAY,
  SDO_ORDINATES      SDO_ORDINATE_ARRAY
);
```

O objeto SDO_GEOMETRY é composto pelos seguintes atributos (FERREIRA, 2003):

- SDO_GTYPE: composto por quatro números, onde, os dois primeiros determinam a dimensão da geometria e os restantes o seu tipo. Esses tipos podem ser: 00 (não conhecido), 01(ponto), 02 (linha ou curva), 03 (polígono), 04 (coleção), 05 (multipontos), 06 (multilinhas) e 07 (multipolígonos);
- SDO_SRID: empregado para identificar o sistema de coordenadas, ou sistema de referência espacial, relacionado à geometria;
- SDO_POINT: é definido utilizando um objeto do tipo SDO_POINT_TYPE, que possui os atributos X, Y e Z para representar as coordenadas de um ponto. Somente é preenchido se a geometria for um ponto;
- SDO_ELEMENT_INFO: é um vetor de tamanho variável que guarda as características dos elementos que compõe a geometria. Cada elemento possui suas coordenadas armazenadas no vetor variável denominado SDO_ORDINATES que são interpretadas através de três números armazenados no SDO_ELM_INFO:

- SDO_STARTING_OFFSET: indica qual a posição da primeira coordenada do elemento no SDO_ORDINATES;
 - SDO_ETYPE: indica o tipo do elemento;
 - SDO_INTERPRETATION: indica como o elemento deve ser interpretado, em conjunto com o SDO_ETYPE.
- SDO_ORDINATES: é um vetor de tamanho variável que armazena as coordenadas da geometria.

3.3.3 Índices

O Oracle Spatial fornece dois tipos de indexação espacial, R-Tree e Quadtree, que podem ser utilizados de maneira simultânea. Contudo a Oracle recomenda explicitamente o uso do R-Tree devido ao desempenho superior.(CÂMARA et al, 2005).

Estes índices são definidos em Ferreira 2003, como:

- R-Tree: divide o dado em conjuntos de objetos geográficos, através da distribuição dos mínimos retângulos envolventes desses objetos no espaço;
- Quadtree: divide o espaço 2d, de forma recursiva, em quadros, independente da distribuição dos objetos geográficos no plano.

3.3.4 Consultas Espaciais

O modelo de consulta utilizado pela extensão é o baseado em duas etapas, denominadas: primeiro e segundo filtro. O primeiro filtro considera a proximidade das

geometrias pelo critério do mínimo retângulo envolvente, para minimizar a complexidade computacional, já o segundo filtro trabalha com as geometrias exatas, desta forma, tem um custo computacional mais elevado e só é aplicado ao resultado do primeiro filtro. (CÂMARA et al, 2005).

Para poder efetuar consultas e operações espaciais, o Oracle Spatial fornece alguns operadores e funções para serem utilizadas juntamente com a linguagem SQL. Para efetuar consultas de relações espaciais são utilizados os seguintes operadores (Ferreira, 2003):

- SDO_RELATE: verifica se as geometrias possuem uma determinada relação topológica;
- SDO_FILTER: implementa o primeiro filtro do modelo de consulta, ou seja, verifica se a interação entre os mínimos retângulos envolventes;
- SDO_WITHIN_DISTANCE: verifica se duas geometrias estão a uma determinada distância, recebida por parâmetro;
- SDO_NN: verifica n vizinhos mais próximos de uma geometria.

Dentre as funções pode-se citar (CÂMARA, 2005):

- SDO_BUFFER: gera uma nova geometria em volta ou dentro de uma outra, conforme uma distância passada por parâmetro;
- SDO_AREA: calcula a área de uma geometria;
- SDO_LENGTH: calcula o perímetro ou comprimento de uma geometria;
- SDO_DISTANCE: calcula a distância entre duas geometrias;

- SDO_INTERSECTION: cria uma nova geometria resultante da intersecção entre outras duas;

- SDO_UNION: cria uma nova geometria resultante da união entre outras duas;

- SDO_DIFERENCE: cria uma nova geometria resultante da diferença entre outras duas.

3.4 MySQL

O MySQL 4.1 introduz extensões espaciais para gerar, armazenar e analisar recursos geográficos. Atualmente estes recursos estão disponíveis apenas para tabelas MyISAM²⁰. O SGBD também implementa extensões espaciais seguindo especificações do OpenGIS Consortium (OGC).

3.4.1 Tipos de Dados Espaciais do MySQL

O MySQL fornece um conjunto de dados espaciais que correspondem as classes especificadas no modelo geométrico OpenGIS. Alguns destes tipos guardam valores únicos (MYSQL, 2005):

- GEOMETRY;
- POINT;
- LINESTRING;
- POLYGON.

O tipo *Geometry* é o mais genérico dentre eles, ele pode armazenar geometrias de qualquer tipo. Os demais restringem seus valores aos seus tipos específicos.

²⁰ “MyISAM é o tipo de tabela padrão no MySQL Versão 3.23. Ela é baseada no código ISAM e possui várias extensões úteis.”(MYSQL, 2005).

Outros tipos de dados são coleção de valores (MYSQL, 2005):

- MULTIPOINT;
- MULTILINESTRING;
- MULTIPOLYGON;
- GEOMETRYCOLLECTION.

Assim como o tipo *Geometry*, o *Geometrycollection* é o mais genérico dentre eles, podendo armazenar uma coleção de objetos de qualquer tipo. Os demais tipos restringem-se a coleções de tipos específicos, onde, *Multipoint* é uma coleção de tipos *point*; *Multilinestring* é uma coleção de tipos *linestring*; *Multipolygon* é uma coleção de tipos *polygon*.

3.4.2 Índices

O MySQL utiliza R-Tree para indexar colunas espaciais. Um índice espacial é construído usando o MBR²¹. Para grande parte das geometrias, o MBR é o retângulo mínimo envolvente, já para uma linha (*linestring*), o MBR é um retângulo degenerado, nas linhas e nos pontos respectivamente. (MYSQL, 2005).

3.4.3 Consultas Espaciais

O MySQL fornece um conjunto de funções para realizar operações em dados espaciais. Eles são divididos em quatro grandes grupos, de acordo com as operações que realizam (MYSQL, 2005):

- Funções que convertem geometrias entre diferentes formatos²²:

²¹ MBR – *Minimum Bounding Rectangle*.

²² Existem dois formatos padrão para representar geometrias em consultas: WKT (*Well-Known Text*) formato para troca de dados na forma ASC II e WKB (*Well-Known Binary*) formato para troca de dados em fluxos binários representados por valores BLOB. (MYSQL, 2005).

- *GeomFromText*: converte um valor string em formato de geometria e retorna um resultado;
 - *GeomFromWKB*: converte um valor binário em formato de geometria e retorna um resultado;
 - *AsText*: converte um valor em geometria em formato WKT e retorna a string resultante;
 - *AsBinary*: converte um valor em geometria em formato WKT e retorna o valor binário resultante.
- Funções que fornecem acesso a propriedades qualitativas ou quantitativas de uma geometria, cita-se entre elas:
 - *GeometryType*: retorna o nome do tipo da geometria;
 - *Dimension*: retorna a dimensão da geometria;
 - *SRID*: retorna um inteiro indicando o ID do sistema de referência espacial utilizado;
 - *Envelope*: retorna o retângulo mínimo envolvente (MBR).
 - Funções que descrevem relações entre duas geometrias, ainda não foram implementadas estão em fase de implementação;
 - Funções que criam geometrias de outras existentes, cita-se dentre elas:
 - *Envelope*: descrita anteriormente;

- *StartPoint*: retorna o ponto (*point*) inicial da linha (*LineString*) recebida como parâmetro;
- *EndPoint*: retorna o ponto (*point*) final da linha (*LineString*) recebida como parâmetro;
- *ExteriorRing*: retorna o anel exterior do polígono (*polygon*) recebido como parâmetro, como uma linha (*LineString*);
- *InteriorRingN*: retorna o n-ésimo anel exterior, onde n é um número recebido como parâmetro, para o valor polígono (*polygon*) recebido como parâmetro, como uma linha (*LineString*);
- *GeometryN*: retorna o n-ésimo geometria, onde n é um número recebido como parâmetro, para o valor de coleção de geometrias(*Geometry Colection*) recebido como parâmetro, como uma geometria(*geometry*).

3.5 Comparação Entre as Extensões Espaciais

A seguir apresenta-se um resumo comparativo entre as extensões espaciais, baseado em Queiroz 2003, PostGIS 2005 e MySql 2005:

- Oracle Spatial:
 - Determina tipos espaciais conforme SFSSQL;
 - Possui indexação R-Tree e Quadtree;
 - Possui operadores topológicos conforme matriz nove intersecções;
 - Possui operadores de conjunto;
 - Possui operadores de *Buffer Region*;
 - Possui transformação entre sistemas de coordenadas;

- Possui tabela de metadados das colunas geométricas, porém, de forma diferente da qual estipulada pelo OpenGIS.
- PostGIS:
 - Determina tipos espaciais conforme SFSSQL;
 - Possui indexação R-Tree ou R-Tree sobre Gist;
 - Possui operadores topológicos conforme matriz nove intersecções DE (GEOS);
 - Possui operadores de conjunto (GEOS);
 - Possui operadores de *Buffer Region* (GEOS);
 - Possui transformação entre sistemas de coordenadas (GEOS);
 - Possui tabela de metadados das colunas geométricas, conforme OpenGIS.
- MySQL:
 - Determina tipos espaciais conforme SFSSQL;
 - Possui indexação R-Tree;
 - Não implementado os operadores topológicos;
 - Não implementado os operadores de conjunto;
 - Não implementado os operadores de *Buffer Region*;
 - Não implementada a transformação entre sistemas de coordenadas;
 - Não implementada a tabela de metadados das colunas geométricas.

No decorrer deste capítulo pode-se fazer uma análise de que o PostGIS e Oracle Spatial são as extensões que estão mais habilitadas para tratar com dados espaciais, mediante ao MySQL que tem muitas de suas funcionalidade em fase de implementação.

Para o presente trabalho optou-se pela utilização, dentre as duas, do PostGIS, pois, além de ser uma arquitetura livre está mais dentro das conformidades estipuladas pela OGC.

4. MODELAGEM CONCEITUAL DE DADOS DO PROJETO DE CADASTRO DE DADOS AMBIENTAIS DO VALE DO RIO DO SINOS

O projeto de Cadastro de Dados Ambientais do Vale do Rio do Sinos visa a integração de dados ambientais relativos a esta região, em uma única base de dados. Informações estas relativas a cinco temas distintos: resíduos sólidos e líquidos depositados na região; impacto ambiental proveniente da indústria coureiro-calçadista; impacto ambiental de agronegócios; potencial energético da região; mapeamento dos recursos hídricos superficiais e subterrâneos, além das áreas de potencial interesse ambiental. Estes dados serão coletados por cada uma das cinco instituições participantes do projeto, respectivamente: UNILASALLE, UERGS, FEEVALE, LIBERATO e UNISINOS.

Para a elaboração do esquema conceitual descrito no decorrer deste capítulo foram levantados requisitos de parte do sistema que será desenvolvido, pois, a análise vem sendo feita de maneira progressiva, de acordo com a liberação de informações das instituições envolvidas.

O presente trabalho tratará com dados fictícios relativos ao potencial energético da região que englobam aspectos sócio-econômicos e ambientais, com base no levantamento de requisitos feito com a instituição LIBERATO porque, durante este trabalho, ela já tinha idéia clara de quais dados gostaria de armazenar no sistema. Desta forma busca-se dar início a elaboração de um diagnóstico do potencial energético da região e dos meios de melhoria na qualidade da distribuição de energia e da

possibilidade da inclusão de novos meios de captação energética para sanar problemas com relação a este ponto.

A ferramenta disponibilizará, em formato de mapa, diversas consultas relacionadas com dados que serão coletados pela instituição LIBERATO e com as feições geográficas contidas no mesmo banco de dados, possibilitando, desta forma, a visualização das informações distribuídas na região e permitindo uma análise espacial destas. Assim será possível visualizar com mais clareza possíveis instalações de meios de captação energética e também verificar as necessidades reais de demanda de energia elétrica por delimitação geográfica.

Com base no levantamento de requisitos, foi desenvolvido um esquema conceitual de dados geográficos utilizando o framework GeoFrame.

A princípio, identificou-se os temas que continham dados relevantes. Tais temas, denominados de sub-temas, foram agrupados em um tema mais genérico denominado Potencial Elétrico (Figura 17). Conforme o GeoFrame, os temas retratam uma região geográfica (Lisboa Filho et al, 2002). A região geográfica em questão é a região do Vale do Rio do Sinos, representada como uma instância da classe REGIÃO GEOGRÁFICA.

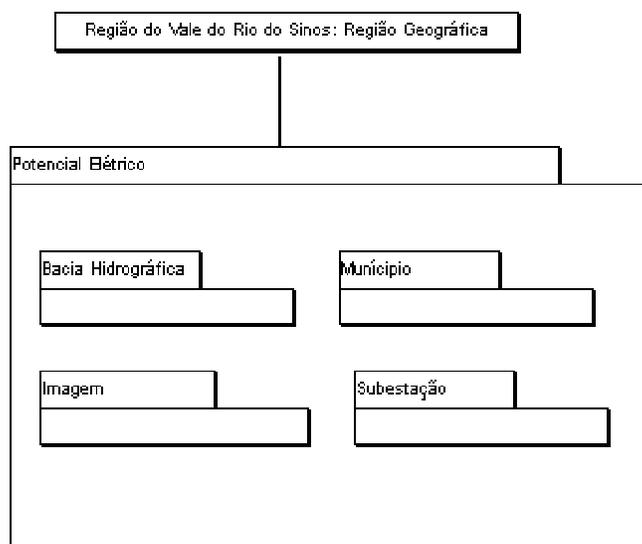


Figura 17 – Diagrama de Temas (figura nossa)

O tema Potencial Elétrico (Figura 18) agrega quatro sub-temas: Município, Bacia Hidrográfica, Imagem e Subestação. A seguir apresenta-se uma descrição das principais classes identificadas em cada tema.

O sub-tema Município engloba as classes que estão diretamente relacionadas com a classe Município e que representam aspectos econômicos e sociais relativos aos municípios da região. A classe Município, subclasse de OBJETO GEOGRÁFICO, é a classe principal deste sub-tema. Além dos seus atributos espaciais, considerada do tipo polígono, ela possui demais atributos não espaciais com dados importantes sobre o objeto em questão, tais como: Área Urbana, População Urbana, Potência Instalada e Potência Demandada.

O sub-tema Subestação contém as classes que tratam da distribuição de energia elétrica, através de subestações, tratando de níveis de potência, transformadores, alimentadores e carga de energia elétrica. Sua classe principal é a Subestação, subclasse de OBJETO GEOGRÁFICO, que contém atributos espaciais, considerada do tipo ponto, e seus valores de tensão de chegada e saída.

O sub-tema Bacia Hidrográfica é composto por apenas uma classe, denominada Bacia, subclasse de OBJETO GEOGRÁFICO, que contém atributos espaciais, considerada do tipo polígono, que representa a Bacia do Vale do Rio do Sinos.

O sub-tema Imagem é composto por apenas uma classe, denominada Foto Aérea, subclasse de REPRESENTAÇÃO CAMPO, que representa uma visão de campo do tipo grade de células. A utilização deste tema no presente trabalho, resume-se a fotos aéreas da bacia do vale do Rio do Sinos.

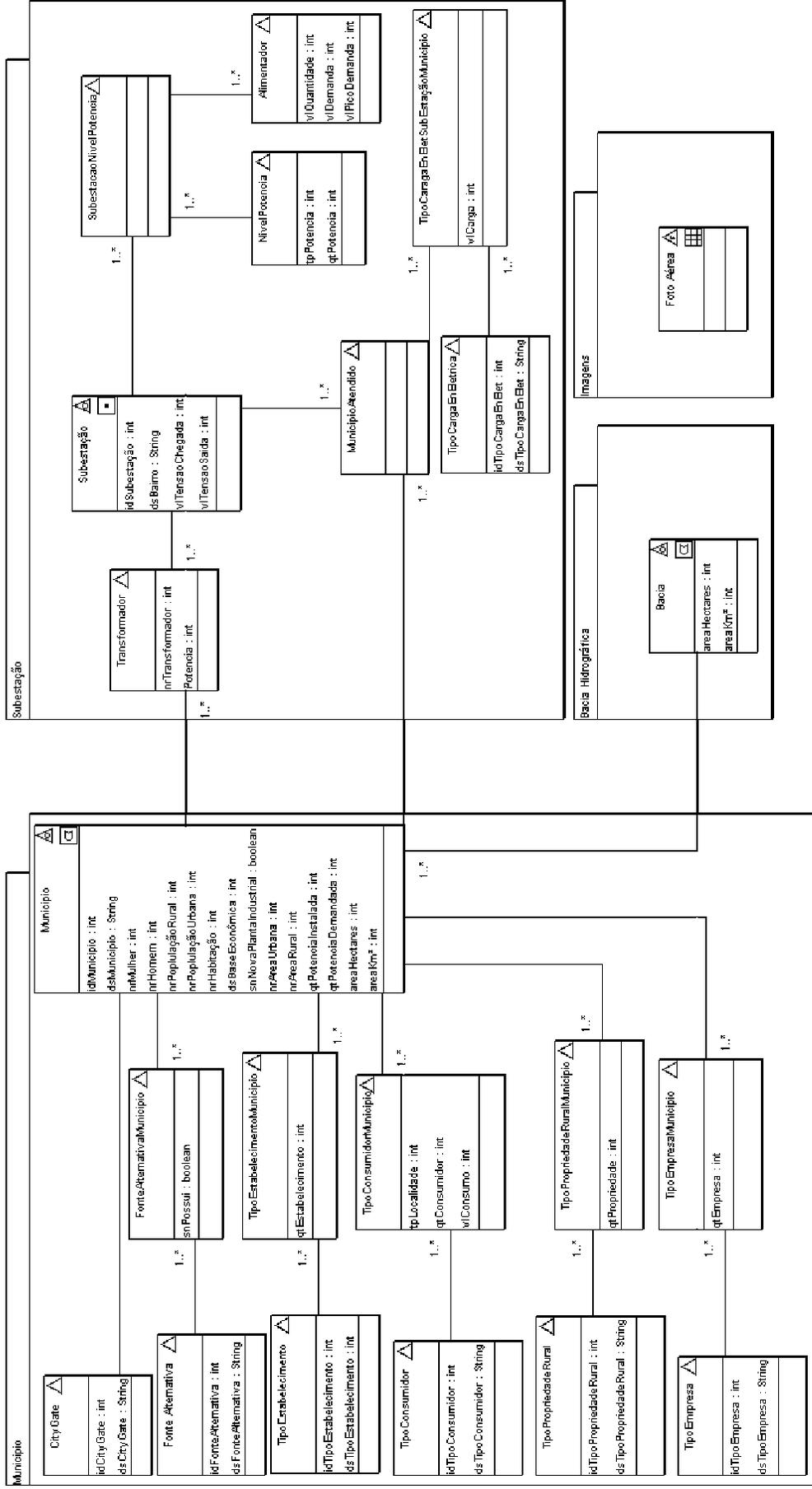


Figura 18 – Tema Potencial Elétrico (figura nossa)

A elaboração do esquema conceitual baseado no GeoFrame apresenta grandes vantagens. Dentre as quais, cita-se (LISBOA FILHO et al, 2000a):

- o esquema de dados final torna-se bem claro, uma vez que somente os elementos essenciais da aplicação são modelados;
- o uso de estereótipos permite, não sobrecarregando visualmente o esquema, a diferenciação entre objetos geográficos e fenômenos geográficos(campos e objetos);
- a divisão do diagrama de classes em temas, utilizando-se de pacotes, torna o esquema fácil de ser lido, uma vez que o leitor pode ler apenas uma parte pequena do esquema por vez;
- devido ao alto acoplamento entre classes de um mesmo tema, o número de associações entre classes de diferentes temas é mínimo, desta forma, contribuindo com a clareza do esquema.

No decorrer deste capítulo apresentou-se o esquema conceitual que será utilizado para o desenvolvimento do trabalho. A sua grande contribuição é a aproximação entre a modelagem conceitual e a implementação de aplicações geográficas, permitindo uma definição mais precisa dos objetos requisitados, seus parâmetros de visualização e suas operações. (CÂMARA et al, 2005).

5. MAPSERVER

A representação de dados geográficos pela internet pode ser feita de várias maneiras. As principais são: através de mapas estáticos em formato de imagem, onde, são inseridas imagens de mapas sem nenhuma possibilidade de interatividade com o usuário; mapas gerados a partir de formulários onde o usuário insere informações a respeito da região geográfica que ele deseja obter dados e o servidor retorna uma imagem com o que foi solicitado; navegação baseada em mapas-chave que exibe um mapa chave em forma de imagem ao usuário, onde se pode interagir com o mesmo através de cliques de mouse na imagem ou selecionando opções que ficam ao redor dela; ou a transmissão de dados vetoriais diretamente ao browser do usuário, onde, não se transmite a imagem, mas sim objetos geográficos com representação vetorial – neste último caso os objetos geográficos enviados ficam na memória da máquina do cliente permitindo ganho de tempo e aumento da interatividade, contudo, esta última forma de transmitir os dados tem um grande obstáculo, pois nenhum dos browsers disponíveis no mercado está preparado para receber e tratar os dados neste formato. Para solucionar este impasse, existe a possibilidade da criação de *plug ins*, contudo, haverá a necessidade de mudança no *plug in* para cada nova versão do browser. (CÂMARA et al, 2005).

O Mapserver se enquadra na navegação baseada em mapas-chave e foi escolhido para exibir as feições geográficas e consultas feitas à base de dados espaciais, devido ao fato de não requerer nenhum tipo de *plug in* para funcionar e ter um ótimo

desempenho. Ele é um servidor de mapas para internet e vem comumente sendo usado para acessar dados armazenados em PostGIS/PostgreSQL, não se limitando a utilizar apenas este gerenciador de banco de dados, pois também permite a utilização de Oracle Spatial e a extensão espacial do MySQL. Foi inicialmente concebido através do projeto ForNet da Universidade de Minnesota (UMN) em cooperação com a NASA e o Departamento de Recursos Naturais de Minnesota (MNDNR). Atualmente ele é mantido pelo projeto TerraSIP, que é patrocinado conjuntamente pela NASA, UMN e um consórcio de interesses de gerenciamento da terra. (MAPSERVER, 2006).

A forma de acesso aos dados contidos no PostGIS/PostgreSQL é a mesma utilizada por qualquer outro cliente PostgreSQL, usando a biblioteca libpq²³. Desta forma, isso demonstra que o Mapserver pode ser instalado em qualquer máquina que tenha a possibilidade de acessar um servidor PostGIS pela rede, uma vez que o sistema disponha das bibliotecas de gerenciamento de clientes do PostgreSQL. (BÜCHNER, 2002).

O Mapserver pode trabalhar de duas maneiras distintas: CGI ou Mapscript. A CGI é a forma mais simples, pois acessa-se diretamente o arquivo executável do Mapserver que está disposto em uma determinada pasta do servidor web, passando parâmetros de inicialização da aplicação *webmapping*²⁴, que processará estas requisições e retornará ao cliente (browser) o resultado esperado (imagens do mapa, legenda, barra de escala, mapa de referência, ou mesmo códigos HTML). O Mapscript é a combinação de recursos do Mapserver com determinada linguagem de programação, dispondo um nível maior de personalização que não pode ser conseguido utilizando-se o modo CGI. O modo Mapscript está disponível para as seguintes linguagens: PHP, Python, Perl, Ruby, TCL, Java e C#.

Dentre suas principais características, citam-se (MAPSERVER, 2006a):

- Saída avançada de cartografia:

²³ A libpq é uma interface para o PostgreSQL, desenvolvida em linguagem C que define um conjunto de funções que permite a programas cliente o envio de consultas ao PostgreSQL e o respectivo recebimento do resultados destas consultas. (POSTGRESQL, 2006).

²⁴ “É uma técnica para visualização de dados geográficos através de aplicações web (Internet/Intranet). Dependendo do tipo da aplicação, pode-se eventualmente inferir nos dados geográficos através da própria interface web”.(KANEGAE, 2005).

- Desenho de camadas e execução de aplicativos dependentes de escala;
 - Rotulação de camadas, incluindo mediação de colisão de rótulos;
 - Saídas direcionadas por modelos, altamente personalizáveis;
 - Fontes TrueType;
 - Automação de elementos de mapas (escala, mapa de referência e legenda);
 - Mapeamento temático usando classes baseadas em expressões lógicas ou expressões regulares.
- Uso de ambientes de desenvolvimento e linguagens de script populares:
 - PHP, Python, Perl, Ruby, Java, e C#.
 - Utilizável em diversas plataformas:
 - Linux, Windows, Mac OS X, Solaris, e outras.
 - Múltiplos formatos matriciais e vetoriais:
 - TIFF/GeoTIFF, EPPL7, e vários outros através da biblioteca GDAL;
 - *Shapefiles* ESRI, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL e muitos outros via OGR.

Para a utilização do Mapserver deve-se criar arquivos denominados *Mapfile*, nos quais estão contidos as definições das feições armazenadas no PostGIS, neste chamadas de *layers*, além das definições e configurações responsáveis pela execução de uma aplicação Mapserver, definindo como os mapas serão apresentados ao usuário. Na seção seguinte serão apresentados os principais objetos e propriedades que compõe um *Mapfile*.

5.1 *Mapfile*

O *Mapfile* é composto por um conjunto de objetos e propriedades, que estipulam o que será exibido no mapa, abaixo demonstrar-se-ão os principais (MAPSERVER, 2006a):

- CLASS: o objeto CLASS define classes temáticas para um determinado layer e cada layer precisa ter ao menos uma CLASS estipulada. No caso de

ter mais de uma CLASS, o usuário é quem determina os valores de atributos e as expressões. Algumas de suas principais propriedades são:

- COLOR: define a cor usada para pintar as feições em formato RGB;
 - EXPRESSION: contém a expressão que será usada para definição da classe que suporta comparação de strings, expressões regulares, expressões lógicas simples e funções string (existe apenas uma função que é a `length`²⁵);
 - KEYIMAGE: define o caminho absoluto da imagem da legenda;
 - LABEL: sinaliza a inicialização de um objeto LABEL;
 - MAXSCALE: escala máxima em que a CLASS é desenhada;
 - MINSSCALE: escala mínima em que a CLASS é desenhada;
 - MINSIZE: tamanho mínimo, em pixels, para o desenho do símbolo, valor default é 0;
 - NAME: nome para uso nas legendas, se não for informado não aparecerá na legenda;
 - OUTLINECOLOR: define a cor das linhas em polígonos, símbolos do tipo linha não o suportam;
 - SIZE: altura, em pixels, do símbolo que será usado;
 - STYLE: sinaliza a inicialização de um objeto STYLE, uma classe pode conter mais de um STYLE;
 - TEMPLATE: arquivo template ou URL que será usado na apresentação dos resultados da consulta ao usuário.
- LABEL: este objeto é usado para definir um rótulo para uma feição do mapa. Algumas de suas principais propriedades são:
 - COLLOR: cor para o desenho do texto do rótulo;
 - FONT: nome da fonte utilizada no rótulo (definida no FONTSET);
 - MAXSIZE: tamanho máximo da fonte(em *pixels*). Valor default é 256;
 - MINSIZE: tamanho mínimo da fonte(em *pixels*). Valor default é 4;
 - POSITION: posição relativa onde o rótulo deve ser desenhado, podem ser informados dois pontos para definição ou utilizar a opção AUTO que o Mapserver define os pontos evitando colisões com outros rótulos;
 - SIZE: define o tamanho do texto;
 - TYPE: tipo de fonte que será usado. Pode ser TRUETYPE ou BITMAP, quando for TRUETYPE é necessário a existência de um arquivo FontSet e também deve-se definir, no objeto LABEL, o nome da fonte True Type a ser usada.
 - LAYER: o objeto LAYER descreve as camadas usadas para constituir um mapa, as camadas são desenhadas na ordem em que aparecem no *Mapfile* (a primeira camada fica embaixo e a última fica em cima). Algumas de suas principais propriedades são:

²⁵ Função que retorna o comprimento de uma determinada String. Exemplo: EXPRESSION (`length(['NAME_EJ'] < 8)`).(MAPSERVER, 2006).

- CLASS: sinaliza a inicialização de um objeto CLASS;
 - CLASSITEM: define o nome do atributo no *Shapefile* ou nas feições contidas no PostGIS, que será usado como campo de classificação para esta camada;
 - CONNECTION: define a string de conexão com o banco;
 - CONNECTION TYPE: define o tipo de conexão, para camadas de PostGIS, é sempre “postgis”;
 - DATA: define o caminho completo para a base de dados espacial, se for uma camada PostGIS o formato deste parâmetro é “<coluna> from <tabela>” onde a coluna é a coluna espacial a ser traduzida no mapa;
 - DEBUG: habilita *debug* do objeto layer;
 - DUMP: permite a mudança de retorno dos dados para formato GML;
 - FEATURE: sinaliza a inicialização de um objeto FEATURE;
 - GROUP: nome do grupo a que pertence o layer;
 - LABELITEM: nome do atributo da tabela para usar como rótulo na classe;
 - LABELMAXSCALE: escala máxima para rotular;
 - LABELMINSCALE: escala mínima para rotular;
 - MAXSCALE: escala máxima para desenhar o *layer*;
 - MINSCALE: escala mínima para desenhar o *layer*;
 - NAME: nome curto para esse *layer*, limite máximo de vinte caracteres, deve ser único e ele é utilizado para interligar o *mapfile* e a interface *web*;
 - PROJECTION: sinaliza a inicialização de um objeto PROJECTION;
 - STATUS: define o status do *layer*, é recomendado utilizar o valor default quando estiver corrigindo algum problema e para uso com funcionamento normal utiliza-se *on* ou *off*;
 - TYPE: especifica como os dados serão desenhados, não necessariamente precisa ser o mesmo tipo de dados da feição do banco.
- LEGEND: este objeto define como a legenda será construída, componentes de legenda são construídos automaticamente nos objetos Class dos *layers* individuais. Algumas de suas principais propriedades são:
 - IMAGECOLOR: define a cor inicial da legenda (cor de fundo em formato RGB);
 - LABEL: sinaliza a inicialização de um objeto LABEL;
 - OUTLINECOLOR: define a cor da linha das caixas da legenda (em formato RGB);
 - POSITION: determina a posição em que a legenda ficará no mapa.
 - PROJECTION: este objeto define qual o sistema de coordenada será exibido no seu mapa, se em toda a base de dados do seu *mapfile* o sistema de coordenadas é o mesmo não é preciso definir, pois, o Mapserver assumirá o sistema de coordenada deste para a projeção.

- QUERYMAP: o presente objeto determina um mecanismo para criar um mapa após o processamento de uma consulta. Algumas de suas principais propriedades são:
 - COLOR: cor em que as feições serão destacadas em formato RGB (*default* é amarelo);
 - SIZE: tamanho do mapa em *pixels*;

- REFERENCE MAP: objeto que determina como o mapa de referência²⁶ será criado. A inicialização do objeto é feita utilizando a palavra chave REFERENCE. Algumas de suas principais propriedades são:
 - COLOR: define a cor de preenchimento do retângulo que será desenhado sobre o mapa de referência;
 - EXTENT: define o retângulo envolvente do mapa representado na figura;
 - IMAGE: define o caminho relativo ou completo da imagem utilizada como mapa de referência;
 - MAXBOXSIZE: se a largura ou altura do retângulo de referência for maior que o MAXBOXSIZE, então ele não será desenhado (valor em pixels);
 - MINBOXSIZE: se a largura ou altura do retângulo de referência for menor que o MINBOXSIZE, então ele não será desenhado (valor em pixels);
 - OUTLINECOLOR: define a cor da borda do retângulo de referência;
 - SIZE: define as dimensões em pixels da imagem do mapa de referência.

- SCALEBAR: objeto que determina como a barra de escala será criada. Algumas de suas principais propriedades são:
 - BACKGROUNDCOLOR: define a cor de fundo da barra de escala em formato RGB;
 - COLOR: define a cor de fundo dos segmentos ímpares em formato RGB;
 - LABEL: sinaliza a inicialização de um objeto LABEL;
 - STATUS: define se a imagem da barra de escala será criada;
 - STYLE: define o estilo da barra de escala (estilos válidos são 0 e 1);
 - UNITS: determina a unidade de medida da barra de escala (*feet, inches, kilometers, meters, miles, dd*).

- STYLE: este objeto contém parâmetros para a simbolização, estilos múltiplos podem ser aplicados em uma classe.

- WEB: define como uma interface *web* deverá operar. Algumas de suas principais propriedades são:

²⁶ Um mapa de referência é um mapa em proporção menor ao mapa principal, que exibe alguma sinalização, um ponto ou um retângulo conforme for o nível de *zoom* no mapa principal, para que o usuário possa situar-se em relação ao mapa que ele está navegando.

- EMPTY: página para qual os usuários serão enviados se ocorrer um erro na consulta;
 - ERROR: página para qual os usuários serão enviados se ocorrer um erro que não seja na consulta, como, por exemplo, clicar fora do mapa onde não exista um objeto geográfico;
 - FOOTER: *Template* utilizado após qualquer coisa ser enviada;
 - HEADER: *Template* utilizado antes de qualquer coisa ser enviada;
 - IMAGEPATH: informa o caminho relativo ou absoluto do diretório onde serão geradas as imagens temporárias;
 - IMAGEURL: informa a URL que será usada para localizar as imagens temporárias;
 - TEMPLATE: informa o caminho relativo ou absoluto do diretório da página html que será definida como modelo para definição da interface da aplicação.
- MAP: o objeto MAP é o elemento principal de um *Mapfile*, ele define o início de um arquivo *Mapfile* e todos os outros objetos e propriedades são definidos hierarquicamente abaixo dele. A palavra reservada END finaliza o bloco dos objetos. Algumas de suas principais propriedades são:
 - EXTENT: esta propriedade define o retângulo envolvente do mapa ao inicializar a aplicação;
 - FONTSET: define o caminho absoluto para o arquivo *Fontset*²⁷ a ser utilizado;
 - IMAGECOLLOR: define a cor de inicialização do mapa em formato RGB;
 - IMAGETYPE: define o formato de saída que será gerado (gif, png, jpeg, wbmp, gtiff, swf);
 - LAYER: sinaliza a inicialização de um objeto LAYER;
 - LEGEND: sinaliza a inicialização de um objeto LEGEND;
 - MAXSIZE: configura o tamanho máximo para a imagem do mapa(em pixels);
 - NAME: define o prefixo que será usado para nomear as imagens temporárias, usualmente se opta por um nome pequeno;
 - PROJECTION: sinaliza a inicialização de um objeto PROJECTION;
 - QUERYMAP: sinaliza a inicialização de um objeto QUERYMAP;
 - REFERENCE: sinaliza a inicialização de um objeto REFERENCE MAP;
 - RESOLUTION: seta o número de pixels por polegada para a imagem do mapa, o valor default é 72;
 - SCALE: escala computada para o mapa, definida muitas vezes por aplicação;
 - SCALEBAR: sinaliza a inicialização de um objeto SCALEBAR;
 - SHAPEPATH: define o caminho relativo ou absoluto dos arquivos de mapas;

²⁷ Fontset é o arquivo que recebe o tipo da fonte que será usada no mapa.

- SIZE: define as dimensões do mapa (largura por altura);
- STATUS: sinaliza se o mapa está ativo ou não;
- SYMBOL: sinaliza a inicialização de um objeto SYMBOL;
- UNITS: define a unidade de medida das coordenadas do mapa (feet, inches, kilometers, meters, miles, dd);
- WEB: sinaliza a inicialização de um objeto WEB.

Com a combinação destes objetos e propriedades pode-se representar de forma visual as diversas feições contidas no banco de dados ou em arquivos do tipo *Shapefile*²⁸. A seguir demonstrar-se-ão alguns exemplos de utilização do Mapserver em conjunto com o PostGIS, inclusive, com seus respectivos *Mapfiles*.

5.2 Exemplos do Mapserver com o PostGIS

A arquitetura que compõe o uso do Mapserver em modo CGI com o PostGIS, pode ser vista na Figura 19. O usuário faz uma requisição do seu *browser* ao servidor *web* Apache que, por sua vez transmite os parâmetros ao Mapserver que faz a verificação na base de dados e retorna a imagem do mapa ao Apache que remete ao usuário que a solicitou.

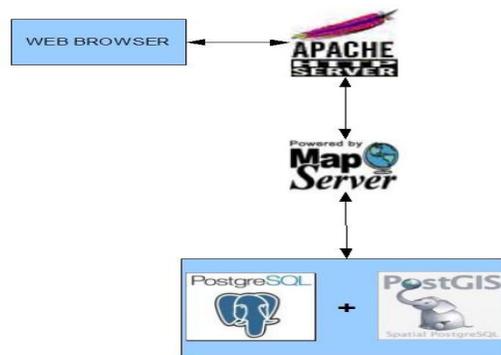


Figura 19 - Arquitetura Mapserver CGI com PostGIS

Fonte: Adaptado de MAPSERVER, 2006a.

²⁸ “*Shapefile* é um formato para o armazenamento de dados espaciais, utilizado pelo software ArcView, da ESRI.” (ESRI, apud BÜCHNER, 2002, p. 17).

Utilizando-se dos objetos discutidos anteriormente associados à extensão espacial PostGIS é possível criar inúmeras opções de saídas de mapa e de consultas diversas. Logo abaixo serão exibidos alguns exemplos de arquivos *Mapfiles*, contemplando o acesso ao banco de dados espacial e também o uso de algumas funções espaciais contidas no PostGIS e seus respectivos resultados.

As tabelas município e bacia contidas nos exemplos a seguir foram inseridas de arquivos *Shapefile*, através do comando *shp2pgsql* contido no SGBD PostgreSQL, o qual é responsável pela importação dos dados contidos em um *Shapefile* criando uma tabela com os dados do mesmo em um banco de dados.

Neste primeiro exemplo, exibir-se-á um *Mapfile* simples contendo acesso ao banco de dados espacial e mostrando todos os objetos espaciais contidos na tabela município.

```
MAP
  EXTENT 455213.5 6669742.5 594650.375 6817326
  SIZE 400 300
  STATUS ON
  NAME "Exemplo"
  LAYER
    CONNECTIONTYPE POSTGIS
    CONNECTION "dbname=TC user=postgres password=1234 host=127.0.0.1 port=5432"
    DATA "the_geom from municipio"
    NAME 'Municipio'
    STATUS DEFAULT
    TYPE POLYGON
    CLASS
      OUTLINECOLOR 0 0 0
      COLOR -1 -1 -1
    END
  END
END
```

Neste exemplo de *Mapfile* são definidos alguns objetos, dos quais o principal é o objeto MAP, que contém os demais. Em seguida, estipulam-se valores para alguns atributos do objeto MAP. No atributo EXTENT, por exemplo, informa-se o valor do retângulo envolvente para o mapa. O atributo SIZE estipula o tamanho da imagem gerada, e em NAME o nome do objeto MAP.

Outro objeto definido no exemplo anterior é o LAYER, o primeiro atributo informado neste, é o CONNECTION TYPE, definido como Postgis, e que determina o tipo de conexão que o objeto LAYER executará. Abaixo tem-se o CONNECTION, responsável por receber a *string* de conexão ao banco de dados. No exemplo em questão, a conexão é realizada em um banco com o nome “TC”, conforme definido em

dbname. O usuário que tem permissão de acesso é o “postgres”, informado em username, sua senha é “1234”, determinada em password, o *host* para acessar o banco é o 127.0.0.1, e a porta para acesso é a 5432, conforme *port*.

A definição do atributo DATA, em LAYER, determina o caminho completo para a base de dados espacial, neste caso, definiu-se “the_geom from municipio”, pois pretende-se exibir todas as geometrias da tabela municipio no mapa. Configurou-se STATUS como *default*, fazendo com que o *layer* sempre seja exibido. Para não exibir um *layer* determina-se o status como *off* e outra forma de exibi-lo também é utilizando o parâmetro *on*. O atributo TYPE estipula o tipo de exibição que a feição será desenhada no mapa e pode ser de três tipos: POLYGON, POINT e LINE. Para este Mapfile foi definido o tipo POLYGON.

Existe também um objeto CLASS dentro de LAYER, onde determinou-se um OUTLINECOLOR com valor “0 0 0”, cor preta, este atributo é responsável pela cor da linha do objeto CLASS, já o atributo COLOR recebeu os valores “-1-1-1 ”, sem cor (transparente), que define a cor de preenchimento da feição (Figura 20).



Figura 20 - Exemplo de um *Mapfile* simples com conexão ao banco de dados

(figura nossa)

No exemplo seguinte mais uma camada foi adicionada ao Mapfile anterior, um *layer*, que utilizará a função espacial *intersects*. Esta retorna verdadeiro se há

intersecção entre duas geometrias do PostGIS, para destacar os municípios vizinhos a Novo Hamburgo. Nesta consulta, fez-se um *select* composto, onde, verificou-se primeiro todos os municípios que tinham o nome diferente ao de Novo Hamburgo, fazendo uma intersecção com o município cujo nome fosse Novo Hamburgo, retornando, desta forma, o resultado de todos que faziam parte do primeiro grupo e que faziam fronteira com o município do segundo grupo, Novo Hamburgo.

Neste novo *layer* notam-se algumas diferenças quanto ao anterior. Em DATA tem-se uma consulta composta, onde a geometria a ser desenhada provém do resultado desta consulta que está entre os parênteses. A informação “As Foo”, significa que o conteúdo desta operação será, para o Mapserver, uma nova tabela denominada Foo. A determinação “using srid = -1”, serve para estipular o identificador do sistema de referência espacial e o “using unique gid” estipula um identificar único para cada linha da tabela. (MAPSERVER, 2006a). O Mapserver utiliza-se deste artifício, pois, para ele é complicado definir um identificador único e um srid quando são utilizados *selects* compostos(MAPSERVER, 2006a). O resultado deste exemplo pode ser visto na Figura 21.

```
MAP
EXTENT 455213.5 6669742.5 594650.375 6817326
SIZE 400 300
STATUS ON
NAME "Exemplo"
LAYER
  CONNECTIONTYPE POSTGIS
  CONNECTION "dbname=TC user=postgres password=1234 host=127.0.0.1 port=5432"
  DATA "the_geom from municipio"
  NAME 'Municipio'
  STATUS DEFAULT
  TYPE POLYGON
  CLASS
    OUTLINECOLOR 0 0 0
    COLOR -1 -1 -1
  END
END
LAYER
  CONNECTIONTYPE POSTGIS
  CONNECTION "dbname=TC user=postgres password=1234 host=127.0.0.1 port=5432"
  DATA "the_geom from (select m2.gid, m2.the_geom, m2.nome from municipio m1,
municipio m2 Where intersects(m1.the_geom, m2.the_geom) and m2.nome <>
'NOVO HAMBURGO' and m1.nome = 'NOVO HAMBURGO') As foo using
SRID=-1 using unique gid"
  NAME "Vizinho"
  STATUS ON
  TYPE POLYGON
  STATUS OFF
  CLASS
    name 'Vizinho'
    OUTLINECOLOR 0 0 0
    COLOR 255 255 0
  END
END
END
```

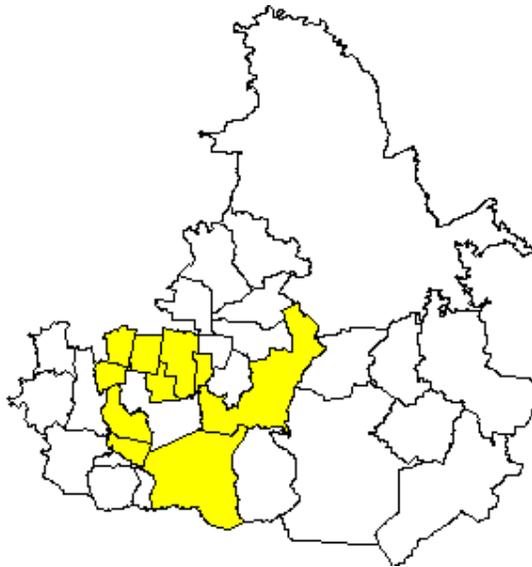


Figura 21 - Exemplo utilizando a função intersects do PostGIS (figura nossa)

Outro exemplo interessante é a utilização de duas feições diferentes no *Mapfile*. No próximo exemplo serão demonstrados dois *layers*, um contendo a feição município, que representa todos os municípios do Vale do Rio do Sinos, e outro contendo a bacia²⁹, que representa a bacia do Vale do Rio do Sinos. A ilustração deste exemplo pode ser visto na Figura 22.

```
MAP
  EXTENT 455213.5 6669742.5 594650.375 6817326
  SIZE 400 300
  STATUS ON
  NAME "Exemplo"
  LAYER
    CONNECTIONTYPE POSTGIS
    CONNECTION "dbname=TC user=postgres password=1234 host=127.0.0.1 port=5432"
    DATA "the_geom from municipio"
    NAME 'Municipio'
    STATUS DEFAULT
    TYPE POLYGON
    CLASS
      OUTLINECOLOR 0 0 0
      COLOR -1 -1 -1
    END
  END
  LAYER
    CONNECTIONTYPE POSTGIS
    CONNECTION "dbname=TC user=postgres password=1234 host=127.0.0.1 port=5432"
    DATA "the_geom from bacia"
    NAME 'Bacia'
    STATUS ON
    TYPE LINE
    CLASS
      COLOR 255 255 0
    END
  END
END
```

²⁹ A tabela bacia foi importada de um arquivo *Shapefile* através do comando `shp2pgsql`.

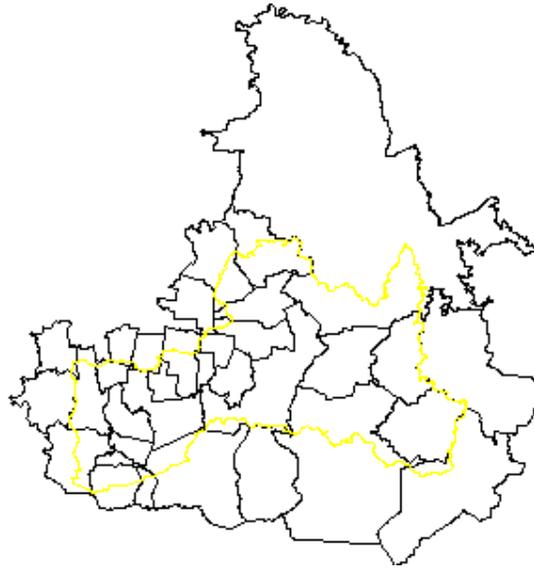


Figura 22 - Exemplo utilizando duas feições (figura nossa)

Ao longo deste capítulo apresentou-se o servidor de mapas Mapserver que será utilizado como ferramenta para visualização de mapas. Ele foi escolhido por ser amplamente utilizado com o PostGIS (BÜCHNER, 2002) e por estar em conformidade com a especificação para servidor de mapas na internet definida pelo OpenGIS (POSTGIS, 2005).

6. APLICATIVO MAPSERVER/PHP MAPSCRIPT

A aplicação prática foi desenvolvida utilizando-se do servidor de mapas Mapserver, da extensão espacial do PHP, denominada PHP Mapscript, que será discutida neste capítulo, e de algumas funções disponibilizadas pelo módulo PostGIS.

A ferramenta exibe, em formato de mapa, diversas consultas relacionadas a dados fictícios, pois a instituição LIBERATO começará a coleta de seus dados a partir do segundo semestre de 2006, e com as feições geográficas contidas no mesmo banco de dados. Permitindo, desta forma, a visualização das informações distribuídas na região, além dos temas definidos no esquema conceitual apresentado no capítulo 4.

As consultas disponíveis na aplicação vão além das consultas encontradas nos SGBDR tradicionais, devido a utilização da extensão espacial PostGIS, e enfocam também características espaciais.

Na seção a seguir, será feita uma descrição sucinta a respeito da extensão espacial do PHP, PHP Mapscript, que foi uma das ferramentas utilizadas para o desenvolvimento do aplicativo.

6.1 PHP Mapscript

O PHP Mapscript é um módulo PHP que é carregado dinamicamente, que insere e disponibiliza as funções e as classes do Mapserver dentro de algum ambiente PHP. Este módulo foi desenvolvido e é mantido pela empresa canadense DM Solutions Group. (BÜCHNER, 2002).

O PHP Mapscript é composto por um conjunto de constantes, classes e funções para acesso as classes do Mapserver, permitindo alterar dinamicamente os *Mapfiles*, de maneira singular, o que não pode ser realizado diretamente com o Mapserver. Por exemplo, pode-se determinar novos *layers*, dinamicamente, no meio de uma aplicação utilizando-se o PHP Mapscript, com a utilização do Mapserver em modo CGI, isto não é possível, pois todos os *layers* devem ser previamente determinados.

Dentre suas principais classes, cita-se (MAPSERVER, 2006b):

- MapObj: retorna um objeto mapa contido no *Mapfile* que foi recebido como parâmetro;
- LayerObj: retorna um objeto do tipo *Layer* através da classe MapObj ou pode ser criado um novo utilizando-se do seu construtor;
- PointObj: retorna um objeto do tipo ponto;
- LineObj: retorna um objeto do tipo linha;
- RectObj: retorna um retângulo com pontos geográficos determinados no seu construtor, comumente usado como retângulo envolvente para outros objetos;
- ShapeObj: retorna um *Shape*³⁰ através da classe LayerObj, no caso da utilização com banco de dados retorna a tabela que contém a feição importada do *shapefile* em questão, ou pode ser criado um novo através do seu construtor;
- ClassObj: retorna um objeto do tipo *Class* através da classe LayerObj ou pode ser criado um novo através do seu construtor.

Maiores detalhes sobre PHP Mapscript e suas classes podem ser encontrados em Mapserver 2006b.

6.2 Descrição da Aplicação

A presente aplicação foi concebida para ser uma ferramenta de exibição de resultados que acessa um SGBDOR, consultando e transformando o resultado de suas consultas, em uma saída visual para o usuário em formato de mapa. Este aplicativo foi

³⁰ Uma feição contida em um arquivo do tipo *shapefile*.

desenvolvido utilizando o servidor de mapas Mapserver, o Php Mapscript e a extensão espacial PostGIS.

A arquitetura que compõe o uso do Mapserver/Php Mapscript com o PostGIS , pode ser vista na Figura 23. O usuário faz uma requisição do seu *browser* ao servidor *web* Apache e este transmite os parâmetros ao Php Mapscript que faz as requisições necessárias ao Mapserver que, por sua vez, faz a verificação na base de dados e retorna a imagem do mapa ao Php Mapscript. Por fim, este remete ao servidor web Apache, que se encarrega de enviar os resultados ao usuário.

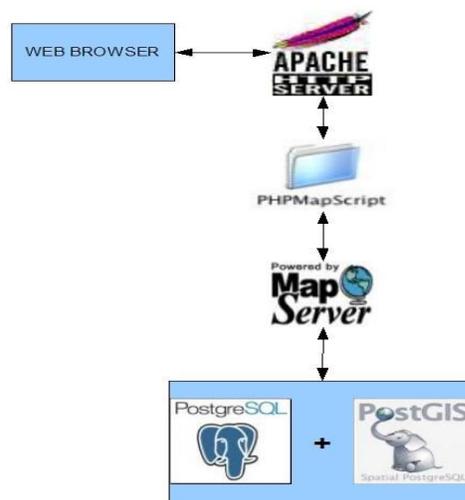


Figura 23 - Arquitetura Mapserver/Php Mapscript com PostGIS

Fonte: Adaptado de MAPSERVER, 2006a.

Todas as pesquisas são feitas através de comandos SQL diretamente ao banco de dados e através destas são gerados os mapas, com a exceção das opções de navegação que serão descritas a seguir. A tela inicial da aplicação está representada na Figura 24.

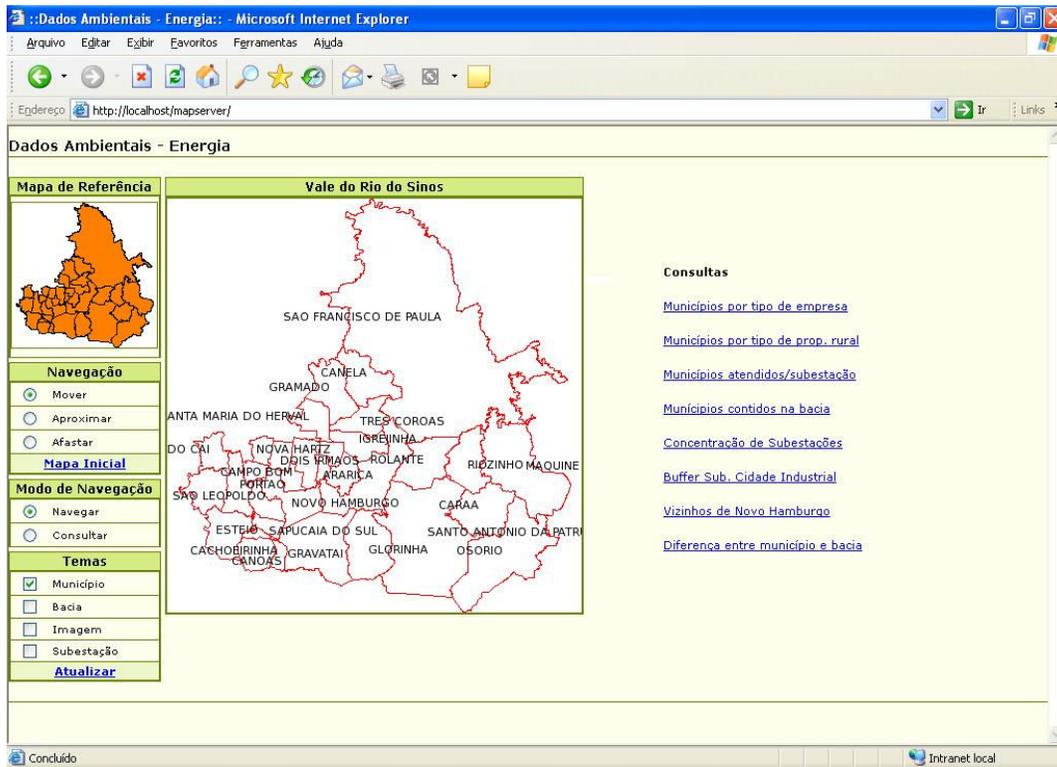


Figura 24 - Tela Inicial

A coluna da esquerda é subdividida em quatro grupos distintos, que são: Mapa de Referência, Navegação, Modo de Navegação e Temas. (figura nossa)

O primeiro grupo Mapa de Referência, possui um mapa de referência que serve para se ter uma referência em que área do mapa se está trabalhando, um quadrado em amarelo fica na região em questão, conforme exemplo na Figura 25.

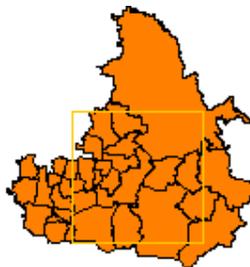


Figura 25 - Mapa de Referência (figura nossa)

O segundo grupo é o Navegação, que possui as opções: Aproximar, Afastar, Mover e o Mapa Inicial. Este grupo está diretamente relacionado com o próximo grupo, Modo de Navegação, este grupo possui duas opções que são Navegar e Consultar que

determinam qual tipo de navegação estará sendo utilizada no mapa. Nas seções seguintes será determinado o funcionamento destas opções.

A quarta opção, da primeira coluna, agrupa a lista dos temas disponíveis para visualização. Estes temas são as tabelas do banco de dados que possuem dados vetoriais. São eles: Município, Bacia e Subestação.

O tema imagem, também presente na quarta opção, refere-se a um mosaico georeferenciado, formado por fotografias aéreas da bacia do Vale do Rio do Sinos. A coluna da direita reúne as opções de consulta da aplicação, cada uma delas será vista em detalhes nas seções seguintes.

A coluna central, possui uma imagem intitulada Vale do Rio do Sinos que é a representação visual do mapa. De acordo com a consulta selecionada aparecerão mais informações nesta coluna, inclusive uma legenda para melhor entendimento do conteúdo do mapa.

Todas as opções trabalham em dois estágios, ou seja, primeiro deve-se marcar qualquer uma das opções e depois clicar em qualquer parte do mapa para efetuar a função escolhida. Há exceções para as consultas que ao serem clicadas imediatamente apresentam os resultados no mapa, a opção Mapa Inicial dentro do grupo Navegação retorna o mapa às suas dimensões iniciais e a opção Atualizar contida no grupo Temas serve para atualizar o mapa com os temas selecionados.

6.3 As opções do Grupo Navegação

A opção Mover permite que, ao clicar no mapa, o ponto que recebeu o clique torne-se o novo centro do mapa, permitido que o mapa se mova conforme a necessidade do usuário. Na Figura 26, efetuou-se um clique na cidade de São Sebastião do Caí para demonstrar o efeito.



Figura 26 - Opção Mover (figura nossa)

A função Aproximar, faz um *zoom* no mapa, ou seja, aproxima a visualização do mapa (Figura 27). Quando a função é ativada (clique no mapa), o ponto do mapa em que foi efetuado o clique passa a ser o novo centro do mapa e de acordo com o fator de aproximação o mapa será redesenhado. Para que seja possível esta aproximação, utiliza-se do método *zoompoint*, da classe *MapObj*, que determina um *zoom* sobre determinada posição XY passando-se um valor positivo como fator de *zoom*.



Figura 27 - Opção Aproximar (figura nossa)

A função Afastar, faz o inverso da função Aproximar, ou seja, afasta a visualização do mapa (Figura 28). Para tanto utiliza-se do mesmo método da classe MapObj, zoompoint, contudo, com um valor negativo como fator de *zoom*.



Figura 28 - Opção Afastar (figura nossa)

6.4 As opções do Grupo Modo de Navegação

A opção Navegar permite a navegação normal no mapa, utilizando-se das opções Afastar, Aproximar e Mover ao clicar no mapa sem nenhum outro tipo de retorno.

A opção Consultar permite a pesquisa de dados cadastrais sobre determinado município. Para obter uma resposta da consulta basta clicar diretamente no mapa sobre o município desejado e, após, será gerado um relatório sucinto, com informações referentes àquela feição (Figura 29).

A aplicação transforma o ponto do mapa onde foi clicado, que está em *pixels*, em coordenadas geométricas compatíveis com o mapa. Após verifica se existe algum tipo de consulta para aquele ponto, com o método `queryByPoint`, da classe MapObj. Se o valor retornado for verdadeiro é efetuado o procedimento percorrendo todos os

objetos *LayerObj*, que estão definidos no *MapObj* em questão, e verificando quais deles possibilitam este tipo de consulta através do método *getResults()*. Encontrando o *Layer*, cria-se um objeto *ShapeObj* a partir do método *getShape* do *Layer*, este objeto *Shape* disponibiliza todos os valores referentes a aquela feição no banco de dados para exibição no relatório. Conforme trecho do código fonte abaixo:

```

if (isset($HTTP_POST_VARS["mapa_x"]) &&
isset($HTTP_POST_VARS["mapa_y"])){
$map_pt = click2map( $HTTP_POST_VARS["mapa_x"],
                    $HTTP_POST_VARS["mapa_y"]
                    );
    $mypoint_query->setXY($map_pt[0],$map_pt[1]);
}
@$VERIFICA = @$map->queryByPoint($mypoint_query, MS_SINGLE, -1);

if($VERIFICA){

$layer->open();
    for ($j=0; $j < $layer->getNumResults(); $j++){
        $result = $layer->getResult($j);
        $shape = $layer->getShape($result->tileindex, $result ->shapeindex);
    }
}

```

No exemplo a seguir, selecionou-se a opção consulta e foi efetuado um clique no município de Santo Antônio da Patrulha (Figura 29). Com isso, será gerado um relatório que contém campos definidos na tabela do banco de dados que representa o *shape* obtido pelo método *getShape*, explicado anteriormente. Define-se quais atributos serão exibidos do *shape* no relatório através do método *values()* do *ShapeObj*. Por exemplo, no caso do relatório em questão, para exibirmos o número da população urbana, atribuímos a uma variável o valor do seguinte comando: *\$shape->values["nrpopulacaourbana"]*. Conforme trecho do código fonte abaixo:

```

$shape = $layer->getShape($result->tileindex, $result ->shapeindex);
$nrpopulacaourbana = $shape->values["nrpopulacaourbana"];

```



Figura 29 - Opção Consultar (figura nossa)

O resultado desta ação é o relatório ilustrado na Figura 30.

SANTO ANTONIO DA PATRULHA							
PIB	Área Urbana	População Urbana	Área Rural	População Rural	Habitacões	Potência Instalada	Potência Demandada
60000	200.35	14000	867	867	3000	40	38

Figura 30 - Relatório gerado pela opção Consulta (figura nossa)

Esta consulta é de suma importância, pois através dela o usuário poderá visualizar os dados cadastrados relacionados a cada município apenas clicando em um ponto dentro do município que deseja solicitar informações.

6.5 As opções do Grupo Temas

A qualquer momento pode-se selecionar quais os temas que se deseja visualizar, com a exceção de quando forem selecionadas consultas, pois elas já têm

como predefinidos os temas que lhes são necessários. Para escolher o tema, basta marcar as caixas de seleção existentes ao lado de cada tema.

A exibição ou não dos temas é definida alterando o status dos *layers* contidos dentro do objeto MapObj, através do método set do objeto LayerObj, passando como parâmetro ("status", 1) para exibição e ("status", 0) para ocultar o *layer*.

Quando o mapa for clicado novamente, os temas serão visualizados conforme a sua escolha. Isto vale também para as opções de Mover, Aproximar e Afastar que funcionam com os temas escolhidos. A opção Atualizar faz uma verificação de todas as opções de Temas selecionadas e as atualiza no mapa, redesenhando-no. A Figura 31 traz um exemplo desta funcionalidade, onde foram marcados os temas Imagem e Bacia.



Figura 31 - Grupo Temas (figura nossa)

6.6 Consultas

As Consultas trabalham com apenas um estágio, ou seja, ao clicar em uma das opções ela é desenhada no mapa. Para a exibição das consultas, fez-se a criação de um layer para cada uma delas e quando é acionada a opção se passa por parâmetro o nome do *layer* para a página. Defini-se o status de todos os *layers* para 0, não permitindo a exibição deles, e para o *layer* da consulta 1. Para que seja possível a captura de um *layer* pelo nome utiliza-se o método getLayerByName do objeto LayerObj.

As consultas no presente trabalho foram divididas em dois grupos distintos, o primeiro voltado para as necessidades específicas de informações relativas ao levantamento de requisitos feito junto ao LIBERATO, denominado no presente trabalho como Consultas Específicas e o segundo grupo foi feito para aplicar um número maior de operadores espaciais com o intuito de aprendizado e validação dos mesmos para incorporação em projetos futuros, denominado como Consultas Genéricas.

As duas subseções seguintes tratam destas consultas.

6.6.1 Consultas Específicas

A primeira opção de consulta, Municípios por tipo de empresa, retorna todos os municípios por tipo de atividade empresarial predominante. Para tanto, foi feita uma consulta ao banco de dados que se retorna a geometria de todos os municípios, fazendo uma junção com as tabelas que não possuem colunas geométricas TipoEmpresa e TipoEmpresaMunicípio (Figura 32).

Para tal fez-se à consulta abaixo:

```
select  max(tpr.dstipoempresa) as tipo, m.the_geom as the_geom, m.gid from  municipio m,
tipoempresamunicipio mtpr, tipoempresa tpr
where  m.gid = mtpr.idmunicipio
and    tpr.idtipoempresa = mtpr.idtipoempresa
group by m.the_geom, m.gid
```

As três tabelas fazem uma junção e retornam um número de tuplas. Estas são agrupadas pela geometria e pelo identificador único, mostrando apenas o máximo valor (número de tuplas) de cada grupo.

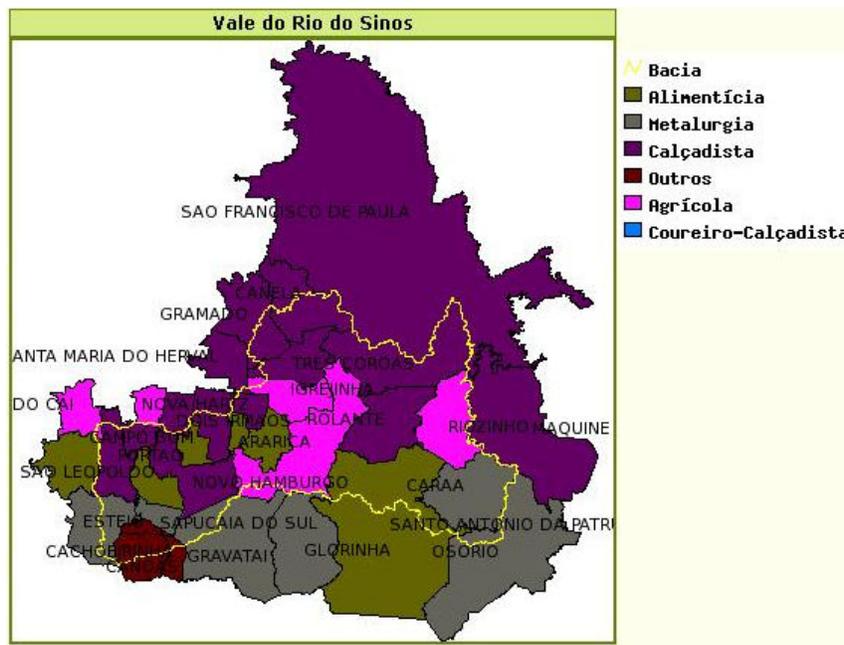


Figura 32 - Municípios por tipo de empresa (figura nossa)

A opção de consulta Municípios por tipo de propriedade rural retorna todos os municípios por tipo de propriedade rural predominante. Para tal, utiliza-se da mesma metodologia da consulta anterior, contudo, as tabelas que não possuem colunas espaciais são TipoPropriedadeRural e TipoPropriedadeRuralMunicipio (Figura 33). Para tal, fez-se a consulta abaixo:

```
select max(tpr.dstipopropriedaderural) as tipo, m.the_geom as the_geom, m.gid as gid
from municipio m, tipopropriedaderuralmunicipio mtpr, tipopropriedaderural tpr
where m.gid = mtpr.idmunicipio
and tpr.idtipopropriedaderural = mtpr.idtipopropriedaderural
group by m.the_geom, m.gid
```

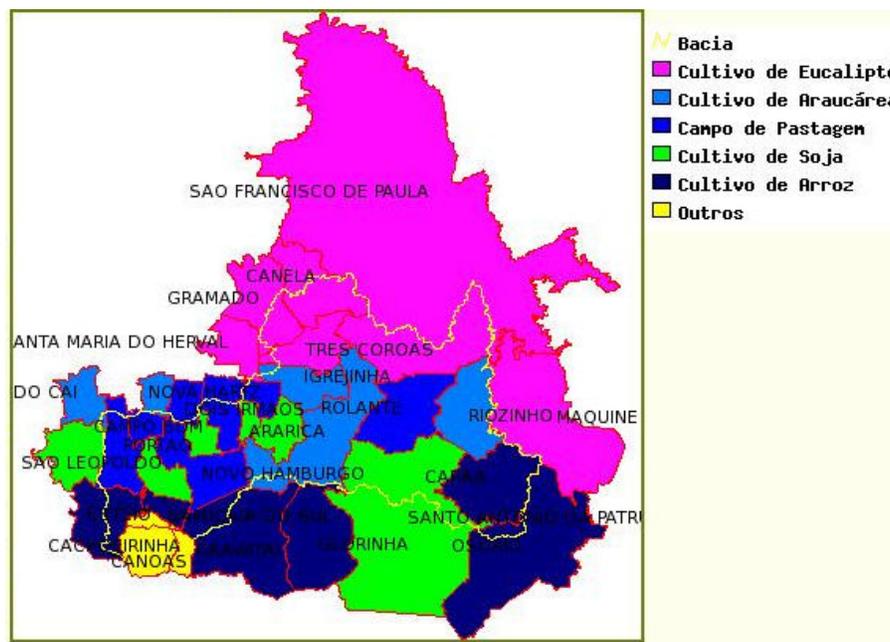


Figura 33 - Municípios por tipo de prop. rural (figura nossa)

A opção de consulta Atendidos por subestação retorna a geometria de todos os municípios que são atendidos por uma determinada subestação (Figura 34). Conforme consulta a seguir:

```
select m.the_geom as the_geom, s.dssubestacao as tipo, m.gid
from municipio m, municipioatendido ma, subestacao s
where m.gid = ma.idmunicipio
and ma.idsubestacao = s.idsubestacao
```

Selecionar todos os municípios que são atendidos por determinada subestação, consultando através da tabela municipioatendido, caso um dos municípios seja atendido por mais de uma subestação ele se enquadra em outros, na legenda, e tem como cor de preenchimento o preto.

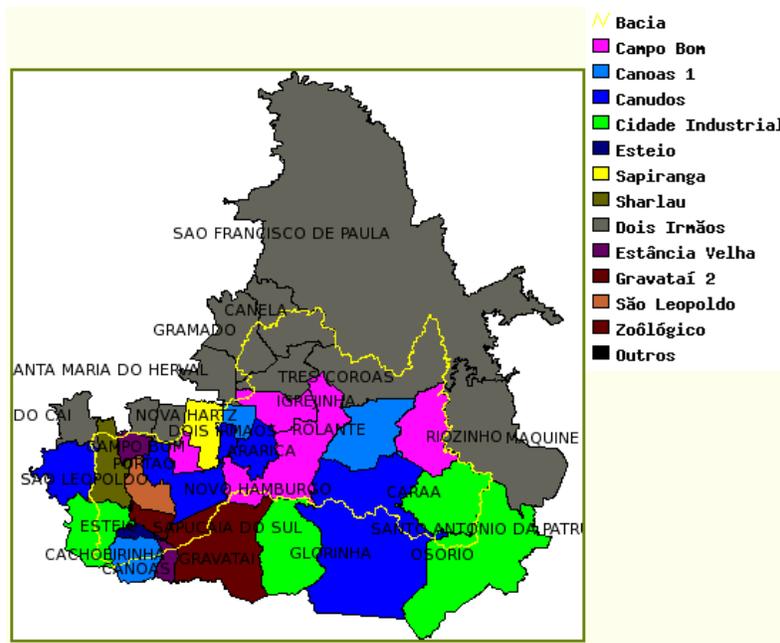


Figura 34 - Municípios atendidos/subestação (figura nossa)

A consulta Buffer subestação Cidade Industrial retorna um *buffer* de 10 Km do ponto em que está a subestação Cidade Industrial. Conforme demonstra-se na Figura 35 e é especificada na consulta abaixo:

```
select buffer((select AsText(the_geom) from Substacao where idsubstacao = 5), 10000 )
as the_geom, gid
from municipio m, substacao s
where s.idsubstacao = 5
and within(s.the_geom, m.the_geom)
```

Para a construção do *buffer* sobre a subestação Cidade Industrial, foi necessário selecionar a geometria da subestação pelo seu identificador único, estipular um raio para o *buffer*, neste caso 10000 metros, e selecionar o identificador único do município em que se localiza a subestação para poder desenhar no mapa. Para solucionar esta questão utilizou-se o operador *within* que retorna verdadeiro se uma geometria está contida em outra.

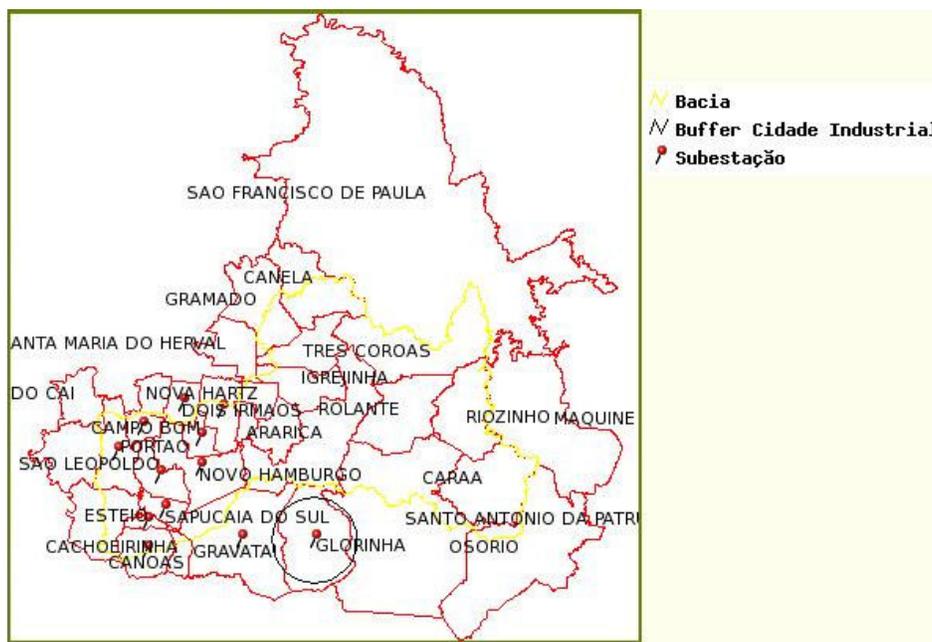


Figura 35 - Buffer Sub. Cidade Industrial (figura nossa)

Como se vê na Figura 35 foi desenhado um círculo em torno da subestação Cidade Industrial com um raio de 10 Km.

6.6.2 Consultas Genéricas

A consulta municípios contidos na bacia utiliza-se do operador topológico *Contains*, do PostGIS, para verificar todos os municípios que estão contidos na bacia do vale do Rio do Sinos. Este operador retorna todas as geometrias que estão totalmente contidas dentro de uma segunda geometria, conforme consulta abaixo(Figura 36).

```
select m.the_geom, m.gid
from bacia b, municipio m
where contains(b.the_geom, m.the_geom)
```

Seleciona todos os municípios que estão contidos na bacia, utilizando-se do operador *contains* que retorna verdadeiro se uma geometria está totalmente contida em outra.

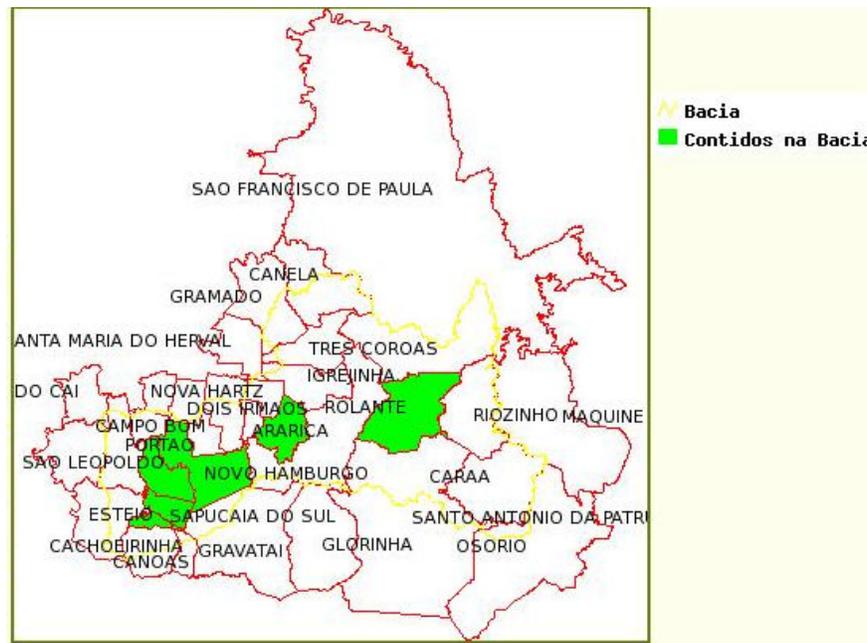


Figura 36 - Municípios contidos na bacia (figura nossa)

A opção Concentração de Subestações desenha uma circunferência com base no ponto central de todas as subestações. Para tanto, foi feita uma consulta para verificar o ponto central das subestações, dentro do retângulo envolvente destas. Definido este valor fez-se uma circunferência para delimitar a região com um raio de 30 Km, através do operador *buffer* (Figura 37). Conforme a consulta a seguir:

```
select distinct buffer (
    (select AsText(centroid(extent(the_geom)))from Subestacao), 30000
) as the_geom, gid
from Município
```

Para descobrir o ponto central de todas as subestações, foi necessária a utilização do operador espacial *extent*, que retorna o retângulo envolvente de uma geometria. Após definido o retângulo envolvente, descobriu-se o centróide deste retângulo com o operador *centroid*, que é o ponto central que se estava a procura. Definido o ponto central usou-se o operador *buffer* para desenhar a circunferência.

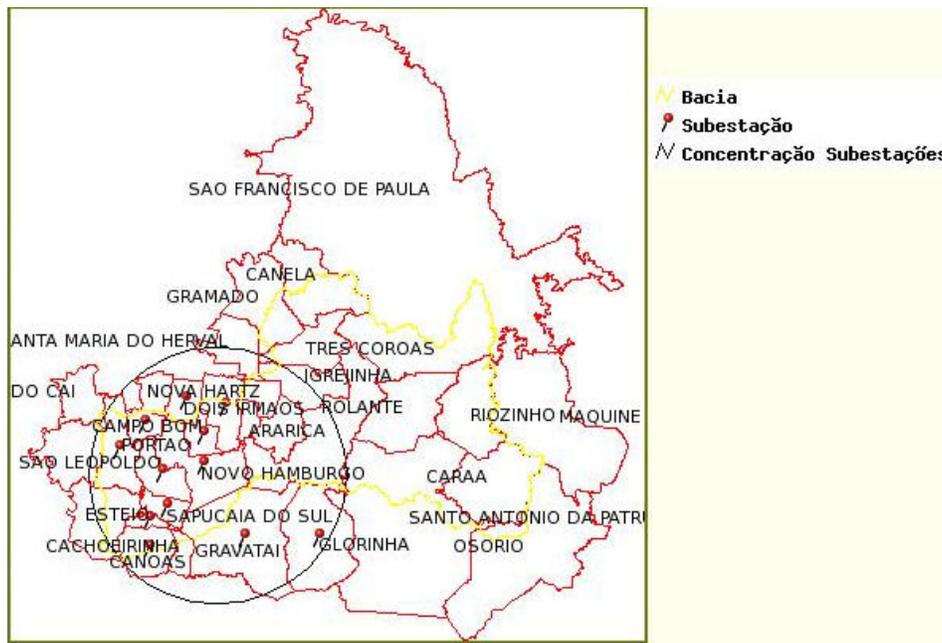


Figura 37 - Concentração de Subestações (figura nossa)

A opção Vizinhos de Novo Hamburgo, retorna todos os municípios que fazem intersecção com Novo Hamburgo e que não seja ele, conforme detalhes da consulta abaixo(Figura 38).

```
select m2.gid, m2.the_geom, m2.nome
from municipio m1, municipio m2
where intersects(m1.the_geom, m2.the_geom)
and m2.nome <> 'NOVO HAMBURGO'
and m1.nome = 'NOVO HAMBURGO'
```

Para determinar os municípios vizinhos de Novo Hamburgo, foi necessária a utilização do operador *intersects*, que retorna verdadeiro se uma geometria está tocando em outra. A consulta utilizou duas vezes a tabela municipio para a comparação, m1 e m2, onde, m1 é igual a Novo Hamburgo e m2 qualquer município diferente de Novo Hamburgo. Então fez a intersecção entre as duas tabelas, m1 e m2, para verificar se estas tinham uma intersecção. Após, verificada que a intersecção existia, foram desenhados no mapa todos os municípios m2.



Figura 38 - Vizinhos de Novo Hamburgo (figura nossa)

A consulta Diferença entre município e bacia, retorna uma nova geometria que é a resultante da geometria município menos a geometria bacia, para tanto foi necessário verificar se havia algum município que estava dentro da bacia e retirá-lo da consulta, pois, não há como fazer uma subtração de um mesmo espaço geométrico duas vezes. Segue abaixo a consulta e seu detalhamento em seguida(Figura 39).

```
select difference(m.the_geom, b.the_geom) as the_geom, m.gid
from municipio m, bacia b
where not within(b.the_geom, m.the_geom)
```

Para realizar a consulta que retorna uma geometria da diferença de outras duas, foi utilizado dois operadores, o *difference* e o *within*. Primeiramente, pretendia-se retirar a bacia de dentro do município criando uma nova geometria que fosse o resultado desta equação, para tanto utilizou-se o operador *difference*. Contudo, existiu a necessidade de fazer uma verificação se havia algum município contido na bacia, pois se houvesse a consulta iria retirar um município inteiro e na hora de criar a nova geometria iriam aparecer, aqueles municípios contidos, com valor vazío. Para resolver este problema utilizou-se do comando *within* que verifica se uma geometria está contida em outra, no caso em questão só deveria fazer a consulta para as geometrias que não estivessem contidas na bacia.



Figura 39 - Diferença entre município e bacia (figura nossa)

No decorrer deste capítulo foi apresentada a aplicação prática que buscou utilizar o conjunto de ferramentas, conceitos e metodologias descritas no decorrer do presente trabalho. Além do detalhamento das consultas utilizadas e da explicação do funcionamento do aplicativo.

CONCLUSÃO

O presente trabalho demonstrou os conceitos ligados a Extensões Espaciais, fazendo uma explanação inicial sobre Sistemas de Informação Geográfica (SIG), a fim de contextualizar suas duas categorias distintas de representação de dados geográficos, vetorial e matricial, além das diferentes arquiteturas de banco de dados disponíveis, entre as quais, inclui-se a arquitetura integrada baseada em extensões espaciais.

Foram abordadas no trabalho três extensões espaciais distintas, PostGIS, Oracle Spatial e MySQL. A partir deste estudo, foi escolhida a extensão espacial PostGIS para armazenar e tratar as coordenadas geográficas por estar em maior alinhamento com as definições do OpenGIS, por ser uma arquitetura livre e porque a base de dados do Projeto de Cadastro de Dados Ambientais do Vale do Rio do Sinos está sendo armazenada em PostgreSQL.

Em seguida, fez-se um levantamento de requisitos de parte do sistema que será desenvolvido, gerando-se um esquema conceitual de dados geográficos utilizando o framework GeoFrame. Montado o esquema conceitual e definidas as feições espaciais trabalhadas, criou-se a base de dados no PostgreSQL, utilizando a extensão espacial PostGIS para armazenar e tratar as coordenadas geográficas.

O servidor de mapas Mapserver mostrou-se uma ótima ferramenta para ser utilizado como meio de exibição do resultado das consultas espaciais em formato de mapa. Após o uso do módulo Php Mapscript, que permite uma maior flexibilidade ao utilizar o servidor de mapas em questão, construiu-se um aplicativo que dispõe de

consultas que atendem as necessidades específicas de informações relativas ao levantamento de requisitos feito junto ao LIBERATO e outras consultas para o aprendizado de operadores espaciais.

Baseado nestes estudos, foi desenvolvida uma ferramenta para exibição de resultados em formato de mapa que acessa o SGBDOR, consultando e transformando o resultado de suas consultas mapas, que podem ser facilmente manipulados através do aplicativo, onde, permite-se aplicar *zoom*, diminuindo ou ampliando a imagem, mover o mapa, inserir os temas definidos no esquema conceitual, além de algumas consultas predefinidas.

Abaixo são citadas algumas restrições do aplicativo desenvolvido e outras possibilidades de ampliação da pesquisa, que podem ser exploradas em trabalhos futuros:

- Permitir ao usuário a entrada de parâmetros para as consultas, contribuindo para uma busca mais eficiente de resultados esperados pelo usuário;
- Explorar outras consultas que venham a agregar informações importantes ao projeto;
- Aplicar esta tecnologia nas demais áreas do projeto, como, ao cadastrar uma subestação seja permitida a inserção das coordenadas geográficas através do formulário de cadastro;
- Ampliar o esquema conceitual, com os dados das outras instituições envolvidas, ampliando o número de temas e o aplicativo.

BIBLIOGRAFIA

1. BOGORNY, Vânia. **Um Estudo sobre o OpenGIS: A Proposta da OGC para Interoperabilidade e Distribuição em Sistemas de Informação Geográfica**. Porto Alegre, RS: PPGC da UFRGS, 1999. 79p.
2. BÜCHNER, Rogério Luiz. **Objetos Espaciais em Banco de Dados Relacional: Uma Aplicação prática utilizando PostGIS**. Ponta Grossa, PR: Especialização em Administração de Banco de Dados, Universidade Estadual de Ponta Grossa (UEPG), 2002. 54p.
3. **SPRING: Integrating remote sensing and GIS by object-oriented data modelling**. CÂMARA, G et al. *Garrido J Computers & Graphics*, 20: (3) 395-403, May-Jun 1996.
4. CÂMARA, G.. **Modelos, Linguagens e Arquiteturas para Bancos de Dados Geográficos**. São José dos Campos, SP: Ph.D., Instituto Nacional de Pesquisas Espaciais (INPE), 1995. 264p.
5. CÂMARA, G. et al. **Banco de Dados Geográficos**. São Paulo, SP: MundoGEO, 2005. 506p.
6. DAVIS JR., C. A. **Múltiplas Representações em Sistemas de Informação Geográficos**. Belo Horizonte, MG: Doutorado, Universidade Federal de Minas Gerais (UFMG), 2000.
7. FERREIRA, Karine Reis. **Interface para Operações Espaciais em Banco de Dados Geográficos**. São José dos Campos, SP: Mestrado, Instituto Nacional de Pesquisas Espaciais (INPE), 2003.100p.
25. KANEGAE, Eduardo Patto. **Introdução ao Mapserver**. Itajaí, SC: Mini-Curso, II Encontro Nacional de Usuários do MapServer, Univali, 2005.
9. LISBOA FILHO, J. **Projeto de Banco de Dados para Sistemas de Informação Geográfica**. In: Bertoldi, M. D. Marta. II Escola Regional de Informática RJ/ES. Vitória-ES: FAESA, 2002. p.109-147.

10. LISBOA FILHO, J. et al. **Modelagem de Banco de Dados Geográficos**. In: III Escola Regional de Informática do Centro-Oeste. Brasília – DF. SBC – Sociedade Brasileira de Computação, 2000a.
11. LISBOA FILHO, J. et al. **Modelagem Conceitual de Banco de Dados Geográficos: o estudo de caso do projeto PADCT/CIAMB**. In: Centro de Ecologia/UFRGS. (Org.). Carvão e Meio Ambiente. Porto Alegre 2000b, v. p. 440-458.
12. LISBOA FILHO, J. et al. **Projeto de banco de dados geograficos: mapeando esquemas GeoFrame para o SIG Spring**. In: I BRAZILIAN WORKSHOP ON GEOINFORMATICS - GEO-INFO99, 1999, Campinas. Anais. São José dos Campos: INPE, 1999.
13. MAPSERVER. **Mapserver**. Disponível em: <<http://mapserver.gis.umn.edu>> Acesso em: 05 abr. 2006.
14. MAPSERVER. **Mapserver PHP/Mapscript Class Reference**. Disponível em: <<http://mapserver.gis.umn.edu/doc35/phpmapscript-class-guide.html>> Acesso em: 05 abr. 2006b.
15. MARISCO, Nelson. et al. **Protótipo de Mapa para Web Interativo: Uma Abordagem Utilizando Código Aberto**. In: RBC - Revista Brasileira de Cartografia nº 56/1. Presidente Prudente – SP. jul. 2004.
16. MAS, Jean-François. **Classificação de Imagens Utilizando Dados Auxiliares numa Abordagem Nebulosa**. In: VIII Simpósio Brasileiro de Sensoriamento Remoto, 1996. Anais. Salvador – BA.
17. MYSQL. **Manual de Referência do MySQL 4.1**. Disponível em: <<http://postgis.refrations.net/docs/postgis.pdf>> Acesso em: 01 nov. 2005.
18. PEREIRA, Marconi de Arruda. **Uma Implementação do Serviço WMS sobre a Biblioteca TerraLib**. Mestrado, Pontifícia Universidade Católica (PUC-RIO), Rio de Janeiro, RJ.
19. POSTGIS. **PostGIS Manual**. Disponível em: <<http://postgis.refrations.net/docs/postgis.pdf>> Acesso em: 01 nov. 2005.
20. QUEIROZ, Gilberto Ribeiro de. **Algoritmos Geométricos para Bancos de Dados Geográficos: Da Teoria À Prática Na TERRALIB**. São José dos Campos, SP: Mestrado, Instituto Nacional de Pesquisas Espaciais (INPE), 2003. 143p.
21. ROCHA, Luciana Vargas da. **GeoFrame-T: um Framework Conceitual Temporal para Aplicações de Sistemas de Informação Geográfica**. Porto Alegre, RS: Mestrado, Universidade Federal do Rio Grande do Sul (UFRGS), 2001. 137p.
22. SILBERSCHATZ, Abraham et al. **Sistema de Banco de Dados**. São Paulo: Pearson Education do Brasil, 1999. 778p.

23. SILVA, Evaldo Oliveira da. Introdução a Sistemas de Informações Geográficas. **SQL – Magazine**. Disponível em:
<http://www.sqlmagazine.com.br/Colunistas/EvaldoOliveira/03_IntrSistInformGeogr.asp>
Acesso: 23 ago. 2005.
24. SILVA, Rosângela. **Banco de Dados Geográficos: Uma análise das arquiteturas Dual (Spring) e Integrada (Oracle Spatial)**. São Paulo, SP: Mestrado, Universidade de São Paulo (USP), 2002. p.137.
25. SIMON, Fernando; CHRISTL, Arnulf. **Introdução ao PostGIS**. Itajaí, SC: Mini-Curso, II Encontro Nacional de Usuários do MapServer, Univali, 2005.
26. WIKIPÉDIA, a enciclopédia livre. **Teoria dos grafos**. Disponível em:
<http://pt.wikipedia.org/wiki/Teoria_de_grafos>
Acesso: 24 out. 2005a.
27. WIKIPÉDIA, a enciclopédia livre. **Framework**. Disponível em:
<<http://pt.wikipedia.org/wiki/Framework>>
Acesso: 28 out. 2005b.