
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS GEODÉSICAS E
TECNOLOGIAS DA GEOINFORMAÇÃO

**AVALIAÇÃO DA INTEGRAÇÃO ENTRE SISTEMAS
DE INFORMAÇÕES GEOGRÁFICAS E SISTEMAS
GERENCIADORES DE BANCOS DE DADOS COM
EXTENSÕES ESPACIAIS**

AMÁLIA MARIA THORPE CHALEGRE

Orientador: Prof. Dr. José Luiz Portugal

Dissertação de Mestrado

Recife, 2011

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS GEODÉSICAS E
TECNOLOGIAS DA GEOINFORMAÇÃO

Amália Maria Thorpe Chalegre

AVALIAÇÃO DA INTEGRAÇÃO ENTRE SISTEMAS DE
INFORMAÇÕES GEOGRÁFICAS E SISTEMAS GERENCIADORES DE
BANCOS DE DADOS COM EXTENSÕES ESPACIAIS

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-Graduação em Ciências Geodésicas e Tecnologias da Geoinformação, do Centro de Tecnologia e Geociências da Universidade Federal de Pernambuco, como parte dos requisitos para obtenção do grau de Mestre em Ciências Geodésicas e Tecnologias da Geoinformação, área de concentração Cartografia e Sistemas de Geoinformação e defendida no dia 02/09/2011.

Orientador: Prof. Dr. José Luiz Portugal

Recife

2011

Catálogo na fonte
Biblioteca Margareth Malta, CRB-4 / 1198

C436a	Chalegre, Amália Maria Thorpe. Avaliação da integração entre Sistemas de Informações Geográficas e Sistemas Gerenciadores de Bancos de Dados com extensões espaciais / Amália Maria Thorpe Chalegre. - Recife: O Autor, 2011. v, 74 folhas, il., gráfs., tabs. Orientador: Prof. Dr. José Luiz Portugal. Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Ciências Geodésicas e Tecnologias da Geoinformação, 2011. Inclui Referências Bibliográficas e Anexos. 1. Engenharia Cartográfica. 2. SIG. 3. SGBD. 4. Programa computacional livre. 5. Interoperabilidade de dados espaciais. 6. OpenGIS. I. Portugal, José Luiz (Orientador). II. Título. UFPE 526.1 CDD (22. ed.) BCTG/2011-260
-------	---

AVALIAÇÃO DA INTEGRAÇÃO ENTRE SISTEMAS DE INFORMAÇÕES GEOGRÁFICAS E SISTEMAS GERENCIADORES DE BANCOS DE DADOS COM EXTENSÕES ESPACIAIS

POR

AMÁLIA MARIA THORPE CHALEGRE

Dissertação defendida e aprovada em 02/09/2011.

Banca Examinadora:

Prof. Dr. JOSÉ LUIZ PORTUGAL

Departamento de Engenharia Cartográfica - Universidade Federal de Pernambuco

Prof. Dra. LUCILENE ANTUNES CORREIA MARQUES DE SÁ

Departamento de Engenharia Cartográfica - Universidade Federal de Pernambuco

Prof. Dr. PAULO MARCIO LEAL DE MENEZES

Departamento de Geografia - Universidade Federal do Rio de Janeiro

DEDICATÓRIA

Aos meus pais Adauto e Alba (*in memoriam*) que, entre tantas outras coisas, me ensinaram o valor do aprender.

AGRADECIMENTOS

A Deus.

À minha filha Juliana e à minha irmã Ada, companheiras de todas as horas, pelo apoio incondicional.

Ao meu orientador Prof. Portugal, pelo conhecimento passado, pelo incentivo, paciência e compreensão durante o desenvolvimento deste trabalho.

Ao Serpro, pela oportunidade de fazer o mestrado. À Helena Cristina, Paulo Bastos e Patrícia Batista pela confiança e possibilidade de desenvolver o presente trabalho.

Aos professores do DECart, em especial ao Prof. Jaime Mendonça, meu primeiro professor no departamento, à Prof^a. Ana Lúcia, pelas sugestões para esse trabalho e ao longo do curso e à Prof^a. Lucilene, pelas sugestões para o desenvolvimento deste trabalho, por tudo que me ensinou e principalmente pela amizade.

Aos amigos: Carla e Binha pelo carinho e atenção constantes; Lilian (*in memoriam*), por me lembrar a importância dos desafios; Adriana e Mônica, pela parceria e apoio nos momentos difíceis; Mauro, pela troca de idéias e ajuda na solução dos problemas técnicos; Monteiro, por ter me apresentado o DECart;; Denise, Alexandra, Mary e Arianne pelo incentivo e Expedito, pelo apoio e troca de idéias.

Aos colegas do Mestrado. Um agradecimento especial à Rose e Ângela pelo companheirismo que transcendeu o período do curso.

Por fim, a todos que direta ou indiretamente contribuíram para o desenvolvimento deste trabalho.

SUMÁRIO

RESUMO.....	I
ABSTRACT.....	II
LISTA DE FIGURAS.....	III
LISTA DE QUADROS.....	IV
LISTAS DE SIGLAS E ABREVIATURAS.....	V
1. INTRODUÇÃO.....	1
1.1 - Objetivos da Pesquisa.....	2
1.1.1 - Objetivo Geral.....	2
1.1.2 - Objetivos Específicos	2
1.2 - Estrutura da Dissertação.....	2
2. PROGRAMA COMPUTACIONAL LIVRE.....	4
2.1 - A evolução dos programas computacionais livres	7
2.2 - Programas Computacionais Livres para Tecnologia da Geoinformação	8
2.2.1 – Experiências pelo mundo.....	9
3. CONSÓRCIO INTERNACIONAL OPEN GEOSPATIAL – OGC.....	14
3.1 – Principais Especificações OpenGIS	15
4. SISTEMAS GERENCIADORES DE BANCOS DE DADOS E SISTEMAS DE INFORMAÇÕES GEOGRÁFICAS.....	20
4.1. Sistemas Gerenciadores de Bancos de Dados	20
4.1.1 – Estruturas de armazenamento de dados espaciais.....	22
4.1.2 - Funções específicas para manipulação de dados espaciais.....	24
4.1.3 - Recurso de Indexação	25
4.1.4 – SGBD PostgreSQL/PostGIS	26
4.1.5 - SGBD MySQL	32

4.2 - Sistemas de Informações Geográficas.....	35
4.2.1 - Módulo de Tratamento de Dados Espaciais.....	36
4.2.2 – SIG gvSIG	38
4.2.3 – SIG OpenJUMP	39
4.2.4 – Quantum GIS	41
5. METODOLOGIA DA PESQUISA	42
5.1 Definição do processo de execução da análise	42
5.2 – Seleção dos SGBD e SIG	45
5.3 – Definição dos recursos tecnológicos	45
5.3.1 - Equipamentos computacionais	45
5.3.2 – Programas Computacionais.....	46
5.3.3 – Dados espaciais.....	46
6. COMPARAÇÃO DOS SGBD E ANÁLISE DAS INTEGRAÇÕES.....	47
6.1 – Comparação das estruturas de dados dos SGBD.....	47
6.2 – Análise das integrações	48
6.2.1 - Carga e transferência de dados nos bancos de dados.....	50
6.2.2 – Integração SGBD-SGBD	51
6.2.3 - Integração SIG - SGBD	53
6.2.3.1 - SIG e o PostGIS.....	53
6.2.3.2 - SIG e o MySQL.....	54
6.2.4 - Compartilhamento de dados pelos SIG	56
6.2.5 - Compartilhamento de dados entre SIG com acesso simultâneo a mais de um SGBD.....	57
7. CONCLUSÕES E RECOMENDAÇÕES	59
REFERÊNCIAS BIBLIOGRÁFICAS	62
ANEXO 1 – FUNÇÕES PARA MANIPULAÇÃO DE DADOS ESPACIAIS	68
ANEXO 2 – POSTGIS: TABELAS DE METADADOS E EXEMPLOS	71

RESUMO

CHALEGRE, Amália Maria Thorpe. **Avaliação da Integração entre Sistemas de Informações Geográficas e Sistemas Gerenciadores de Bancos de Dados com Extensões Espaciais**. Recife, 2011, 74p. Dissertação (Mestrado) – Centro de Tecnologia e Geociências, Universidade Federal de Pernambuco.

A pesquisa tem como objetivo analisar a integração entre Sistemas de Informações Geográficas (SIG) e Sistemas Gerenciadores de Bancos de Dados (SGBD) com extensões espaciais, em relação à interoperabilidade de dados geográficos vetoriais. Apoiada na recomendação do governo brasileiro sobre adoção de programas computacionais livres nas soluções de Tecnologias da Informação (TI) e na idéia de que, fortalecer esse conhecimento no ambiente acadêmico é uma forma de disseminá-lo, apenas SGBD e SIG enquadrados nessa categoria foram incluídos na pesquisa. Ressalta-se ainda que esses SGBD respeitam, parcial ou integralmente, as especificações do OGC - *Open Geospatial Consortium*. Na realização do estudo, inicialmente foram definidos os critérios para seleção dos programas e os aspectos a serem avaliados. O critério utilizado para escolha dos SGBD com extensões espaciais foi que tivessem o uso consolidado no gerenciamento de dados alfanuméricos. A escolha dos SIG, baseada no resultado de outras pesquisas, levou em consideração o tempo de maturação do programa, a qualidade do suporte disponível, a possibilidade de ser suportado por mais de um Sistema Operacional, o nível de complexidade para utilização e a velocidade de processamento. Os aspectos considerados na avaliação foram relativos às estruturas gráficas vetoriais integradas aos atributos alfanuméricos e o compartilhamento de dados pelos SIG, através do uso de SGBD com extensões espaciais. Trabalhou-se com dois SGBD, três SIG e o mesmo conjunto de dados. O resultado é um conjunto de considerações sobre a integração entre os SIG e os SGBD e recomendações que podem apoiar a seleção desses programas quanto aos aspectos estudados.

Palavras-chave: Sistemas de Informações Geográficas; Sistemas Gerenciadores de Banco de Dados; Programa Computacional Livre; Interoperabilidade de Dados Espaciais; OpenGIS.

ABSTRACT

The aim of the present research is to analyze the integration between Geographic Information Systems (GIS) and Database Management Systems (DBMS) with spatial extensions, regarding the interoperability of geographic data vector. Supported by the Brazilian Government recommendation on the adoption of free software for Information Technology (IT) solutions, and on the idea that the strengthening of this knowledge in the academic environment is one way to spread it, only DBMS and GIS which fall into this category were included in this research. It is worth noting that these DBMS observe, either wholly or in part, the specifications of the OGC - Open Geospatial Consortium. For this study, it was initially defined the criteria for selection of software and the aspects to be evaluated. The criterion used for choosing DBMS with spatial extensions was that they should have consolidated use in management alphanumeric data. The selection of GIS, based on the results of other researches, took into consideration the maturation time of the program, the quality of available support, the possibility of being supported by more than one operating system, the level of complexity involved in their use and the processing speed. The aspects considered in the evaluation were related to the vector graphic structures integrated to alphanumeric attributes and the sharing of data by GIS, through the use of DBMS with spatial extensions. Two DBMS, three GIS and the same data set were used in the study. The result is a set of considerations regarding the integration between GIS and DBMS and recommendations which can help in the selection of this software as regards the aspects dealt with in this study.

Keywords: Geographic Information Systems; Database Management Systems; Free Software; Spatial Data Interoperability; OpenGIS.

LISTA DE FIGURAS

Figura 1 - Modelo Cliente/Servidor	15
Figura 2 - Interação entre as Especificações OpenGIS e as aplicações	18
Figura 3 - Esquema de armazenamento OGC baseado em extensões espaciais	23
Figura 4 - Hierarquia de Classes Geométricas.....	24
Figura 5 - Processamento de consultas espaciais	26
Figura 6 - Tipos de dados espaciais do PostGIS	29
Figura 7 - Tipos de dados espaciais do MySQL.....	33
Figura 8 - Arquitetura de Sistemas de Informações Geográficas	36
Figura 9 - Esquema de carga dos arquivos no SGBD.....	43
Figura 10 - Esquema de transferência de dados entre SGBD	43
Figura 11 - Acesso dos SIG às tabelas associadas à geometria	43
Figura 12 - Alteração dos bancos de dados pelos SIG	44
Figura 13 - Compartilhamento de dados pelos SIG	44
Figura 14 - Compartilhamento de dados com acesso simultâneo a mais de um SGBD	45
Figura 15 - Ambiente cliente-servidor utilizado.....	48
Figura 16 - Criação de conexão entre o OpenJUMP e o PostGIS	49
Figura 17 - Carga das tabelas no PostGIS.....	51
Figura 18 - Carga das tabelas do MySQL	51
Figura 19 - Transfêrencia de dados entre SGBD	52
Figura 20 – Tabelas armazenadas no PostGIS e visualizadas no QGIS	54
Figura 21 - Geometrias visualizadas no QGIS armazenadas no MySQL.....	55
Figura 22 - Camadas criadas no OpenJUMP usando tabelas associadas á geometria do PostGIS criadas pelos SIG	57
Figura 23 - Compartilhamento de dados entre SIG através do SGBD	58

LISTA DE QUADROS

Quadro 1- Categorias dos programas computacionais	5
Quadro 2 - Mapeamento das operações realizadas pelos SIG sobre o PostGIS.....	53
Quadro 3 - Mapeamento das operações realizadas pelos SIG sobre o MySQL	56

LISTAS DE SIGLAS E ABREVIATURAS

API	<i>Application Programming Interface</i>
BLOB	<i>Binary Large Object</i>
CAMPUS	<i>Cadastre Data Management, Processing and Updating Software</i>
CITE	<i>Compliance & Interoperability Testing & Evaluation</i>
DDL	<i>Data Description Language</i>
DML	<i>Data Manipulation Language</i>
EPSG	<i>European Petroleum Survey Group</i>
FSF	<i>Free Software Foundation</i>
GDAL	<i>Geospatial Data Abstraction Library</i>
GIST	<i>Generalized Search Trees</i>
GML	<i>Geography Markup Language</i>
GPL	<i>General Public License</i>
ISO	<i>International Organization for Standardization</i>
JPP	<i>JUMP Pilot Project</i>
JTS	<i>Java Topology Suite</i>
LGPL	<i>Lesser GPL</i>
ODBC	<i>Open Database Connectivity</i>
OGC	<i>Open Geospatial Consortium</i>
REM	Menor Retângulo Envolvente
QGIS	<i>Quantum GIS</i>
SEXTANTE	<i>Sistema EXTremeno de ANálisis Territorial</i>
SFS	<i>Simple Features Interface Standard</i>
SGBD	Sistemas Gerenciadores de Bancos de Dados
SGBD-OR	Sistemas Gerenciadores de Bancos de Dados Objeto-Relacional
SIG	Sistemas de Informações Geográficas
SQL	<i>Structured Query Language</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TI	Tecnologia da Informação
WCS	<i>Web Coverage Service</i>
WFS	<i>Web Feature Service</i>
WMS	<i>Web Map Service</i>
XML	<i>eXtensible Markup Language</i>

1. INTRODUÇÃO

O Governo Federal brasileiro vem incentivando a adoção de programas computacionais livres pelas instituições públicas. Esta iniciativa tem como objetivos principais promover a independência tecnológica e melhorar as soluções adotadas na área de Tecnologia da Informação (TI), bem como reduzir os custos de aquisição dos programas computacionais proprietários pelos órgãos públicos (GUIA LIVRE, 2005).

Um exemplo é o emprego de programas computacionais livres na confecção do Plano Diretor para as cidades com mais de 20 mil habitantes e para municípios integrantes de regiões metropolitanas, previsto na Lei número 10257, denominada Estatuto da Cidade (BRASIL, 2001).

Uma característica desejável dos programas computacionais é a interoperabilidade. Essa pode ser entendida como a capacidade de diversos sistemas e organizações trabalharem em conjunto, de forma a garantir que esses elementos interajam para trocar informações de modo eficiente (BRASIL, 2011). Para promover a interoperabilidade é importante a adoção de padrões abertos. Os padrões abertos são disponibilizados publicamente e não são controlados por governos ou corporações CHEDE (2008).

O *Open Geospatial Consortium* (OGC) é um órgão internacional, com o objetivo de desenvolver padrões de *interfaces* geoespaciais abertos, visando promover a interoperabilidade geoespacial. Esses padrões são divulgados em forma de especificações direcionadas aos desenvolvedores de tecnologia, para tornar as informações espaciais e serviços acessíveis por vários tipos de aplicações, entre elas os Sistemas de Informações Geográficas – SIG (OGC, 2011b).

Uma das atribuições do OGC é a definição de padrões de armazenamento e manipulação de dados espaciais. Esses padrões são disponibilizados através dos Sistemas Gerenciadores de Bancos de Dados (SGBD).

O uso de SGBD com extensões espaciais padronizadas viabiliza o compartilhamento amplo e simultâneo de dados, seja pela utilização desses dados por vários programas de SIG diferentes, por sistemas corporativos que podem incluir funcionalidades de localização e análise de dados espaciais em seus requisitos, ou ainda sua disponibilização pela internet através de serviços previstos pelo OGC. Os programas de SIG também podem transferir para o SGBD a execução de análises espaciais que antes apenas os SIG dispunham .

A pesquisa visa analisar a integração entre programas de SIG x SGBD e SGBD x SGBD em relação à interoperabilidade dos dados geográficos. O resultado servirá de apoio ao desenvolvedor de aplicações em SIG na seleção de SGBD e SIG de forma a obter o compartilhamento de dados.

1.1 - Objetivos da Pesquisa

1.1.1 - Objetivo Geral

Analisar a integração entre programas de SIG x SGBD e SGBD x SGBD em relação à interoperabilidade dos dados geográficos.

1.1.2 - Objetivos Específicos

- Investigar a integração de programas de SIG com SGBD espaciais;
- Comparar as estruturas de dados dos SGBD espaciais;
- Avaliar o uso de SGBD e programas de SIG livres em relação à interoperabilidade de dados geográficos.

1.2 - Estrutura da Dissertação

Esta dissertação está dividida em 7 capítulos, sendo o primeiro uma introdução sobre o trabalho, onde estão descritos seus objetivos.

O segundo capítulo apresenta uma visão geral dos programas computacionais livres e sua evolução na área de Tecnologias da Geoinformação.

O terceiro capítulo descreve o *Open Geospatial Consortium* (OGC) e suas principais especificações, bem como o processo de verificação da conformidade entre os programas e as especificações.

O quarto capítulo aborda os Sistemas Gerenciadores de Bancos de Dados (SGBD) e os Sistemas de Informações Geográficas (SIG). Sobre os SGBD são apresentados alguns conceitos, e as características recomendadas pelo OGC para as extensões espaciais. São descritas também, as extensões espaciais dos SGBD PostgreSQL e sua extensão PostGIS e do MySQL, que foram escolhidos para serem analisados nesse trabalho. Sobre os SIG, é descrita a arquitetura integrada, que faz uso dos SGBD para o armazenamento e processamento dos dados espaciais, bem como são descritos os SIG gvSIG, OpenJUMP e Quantum GIS, programas selecionados para fazer parte desse trabalho.

O quinto capítulo descreve a metodologia da pesquisa, o sexto capítulo apresenta os resultados da análise e o sétimo capítulo apresenta a conclusão do trabalho.

O Anexo 1 descreve um subconjunto das funções para manipulação de dados espaciais recomendadas pelo OGC.

O Anexo 2 descreve as tabelas de metadados do PostGIS, os comandos para atualização da tabela *GEOMETRY_COLUMNS* e um exemplo de criação de tabela associada a geometria no PostGIS.

2. PROGRAMA COMPUTACIONAL LIVRE

O conceito de programa computacional livre defende a liberdade de uso do programa computacional como solução para as dificuldades provocadas pela limitação do conhecimento tecnológico decorrente do uso dos programas computacionais proprietários (UCHOA e FERREIRA, 2004). A principal característica que diferencia o programa computacional livre do proprietário é a transmissão de direitos legais e o modo de desenvolvimento, que no primeiro é colaborativo (GUIA LIVRE, 2005).

Os programas computacionais são protegidos por direitos autorais. O tipo de licença atribuída ao programa computacional define os direitos que o usuário tem sobre o mesmo. Nos programas computacionais livres, os detentores desses direitos, através de licença de software, dão permissão legal para que usuários possam utilizá-lo em liberdade (SABINO e KON, 2009). A *Free Software Foundation* (FSF), organização sem fins lucrativos criada em 1985, com o objetivo de garantir a liberdade para os usuários de computador, promovendo o desenvolvimento e utilização de programas computacionais livres, define que nesses programas, os usuários devem ter asseguradas quatro liberdades (FSF, 2010):

- A liberdade de executar o programa, para qualquer propósito (liberdade número 0);
- A liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades (liberdade número 1). O acesso ao código fonte é um pré-requisito para isso. Este código é o conjunto de linhas de programação que formam um programa computacional em sua forma original. É escrito em uma linguagem de programação e pode ser entendido por pessoas que conheçam a linguagem;
- A liberdade de redistribuir cópias de modo que se possa auxiliar outros usuários (liberdade número 2);
- A liberdade de distribuir cópias do programa modificado, de modo que toda a comunidade se beneficie (liberdade número 3). O acesso ao código fonte é um pré-requisito para isso.

Os programas computacionais são enquadrados em categorias, de acordo com os direitos que o usuário tem sobre ele, conforme mostra o Quadro 1 (GNU, 2010).

Quadro 1- Categorias dos programas computacionais

CATEGORIAS	DIREITOS
Programa computacional livre	Vem com permissão para uso, cópia e/ou distribuição do código, na forma original ou modificada, gratuitamente ou não. O código fonte do programa deve estar disponível.
Programa computacional de código aberto	É distribuído com seu código fonte. O tipo de licença define o que pode ser feito sobre o código fonte.
Programa computacional livre com <i>Copyleft</i>	Os termos de distribuição asseguram que todas as cópias de todas as versões tenham termos similares de distribuição. É uma forma de impedir que pessoas redistribuam cópias do programa computacional adicionando restrições a seu uso.
Programa computacional livre sem <i>Copyleft</i>	Os termos de distribuição permitem que cópias e versões possam incluir restrições adicionais de uso a ele. As versões modificadas desse programa podem tornar-se proprietárias.
Programa computacional de domínio público	Programa computacional que não está associado a direitos autorais (<i>copyright</i>). Se seu código fonte também é de domínio público, trata-se de um caso de programa computacional livre sem <i>copyleft</i> . Isso significa que suas cópias ou versões modificadas podem não ser livres.
Programa computacional GPL	Programa computacional que tem licença <i>General Public License</i> (GPL). Esta é uma licença pública que possui um conjunto de termos de distribuição que tornam um Programa computacional livre com <i>Copyleft</i> .
Sistema operacional GNU	O GNU é um sistema operacional livre equivalente ao UNIX. O desenvolvimento desse sistema operacional compõe o projeto GNU, de responsabilidade da FSF.

CATEGORIAS	DIREITOS
Sistema operacional GNU	O GNU é um sistema operacional livre equivalente ao UNIX. O desenvolvimento desse sistema operacional compõe o projeto GNU, de responsabilidade da FSF.
Programa computacional proprietário ou não livre	Programa cujo uso, redistribuição ou modificação é proibida ou requer que seja pedido permissão para fazê-lo. Um programa computacional proprietário pode ser desenvolvido com finalidades comerciais ou não.
Programa computacional gratuito (<i>Freeware</i>)	Termo normalmente usado para referenciar programas que permitem redistribuição, mas não permitem modificação e cujo código fonte não está disponível. Não são programas computacionais livres, porém tem distribuição e uso gratuitos.
Programa computacional <i>Shareware</i>	Programa que é gratuito por um período, após o qual há a cobrança para a continuação do uso. Não é programa computacional livre por duas razões: 1 – Normalmente o código fonte não está disponível e 2 - Não vem com permissão de cópia e instalação sem pagar uma licença, nem mesmo para uso em atividades sem fins lucrativos.
Programa computacional privado ou personalizado	É desenvolvido para um usuário, normalmente uma organização ou empresa. Esse usuário pode manter e usar o programa e não liberá-lo ao público.
Programa computacional comercial	Programa computacional desenvolvido por uma empresa que pretende obter lucro com o uso do programa.

Na arquitetura de sistemas pode ser adotada solução composta de programas computacionais livres e proprietários, dependendo das licenças dos programas computacionais envolvidos. Para exemplificar, um programa livre de SIG pode utilizar um SGBD proprietário e vice-versa, enquanto o sistema operacional de suas máquinas também pode ser livre ou proprietário.

Convém ressaltar que os termos programa computacional livre e programa computacional de código aberto muitas vezes são empregados como sinônimos, mas há diferenças entre eles. Para a FSF (2010), o programa computacional código aberto é uma metodologia de desenvolvimento, que procura as melhores formas de desenvolver produtos de maior qualidade. Já o programa computacional livre defende o respeito à liberdade dos usuários. A maior parte dos programas computacionais de código aberto são programas computacionais livres, e aqueles que não são, deixam de atender alguma das quatro liberdades.

O usuário normalmente tem dúvida a respeito da gratuidade do programa computacional livre. Segundo CAVALLINI E PICCARDI (2007) não há regras a respeito de cobrança pelo desenvolvimento e pela comercialização desse tipo de programa computacional, entretanto, devido ao direito de distribuir, prevalece a tendência da distribuição gratuita.

Para os desenvolvedores de programas computacionais livres, o mais importante não é a gratuidade, mas sim o direito de estudar o programa computacional, modificá-lo e distribuí-lo. Já para os usuários, o custo pesa na decisão em adotar a ferramenta.

2.1 - A evolução dos programas computacionais livres

Os programas computacionais livres vêm ganhando espaço no mundo, o que pode ser constatado através dos fóruns, encontros técnicos, seminários e projetos de lei em vários países (TAMBASCIA *et al.*, 2005).

CAVALLINI (2006) cita alguns fatores que fazem o programa computacional livre atraente, especialmente aos países em desenvolvimento:

- Recursos financeiros escassos levam a procura por soluções de baixo custo;
- Disponibilidade de programadores qualificados capazes de customizar e desenvolver os programas;

- Os recursos investidos no desenvolvimento permanecem no país. O dinheiro gasto com programa computacional proprietário muitas vezes é transferido a outros países.

CAVALLINI (2006) aponta também dificuldades da adoção de programas computacionais livres:

- A grande quantidade de produtos, dos quais nem todos estão maduros e totalmente funcionais, torna difícil identificar as melhores soluções para uma aplicação; e
- A falta de divulgação dos produtos.

2.2 - Programas Computacionais Livres para Tecnologia da Geoinformação

Na área de Tecnologias da Geoinformação, RAMSEY (2007) afirma que os programas computacionais de código aberto existentes, podem fornecer uma solução completa como alternativas aos programas computacionais proprietários na maioria dos projetos de sistemas. Na opinião de CAVALLINI (2006), vários produtos estão prontos para uso profissional, enquanto outros precisam continuar seu desenvolvimento para tornarem-se realmente competitivos.

STEINIGER e BOCHER (2009) apontam quatro indicadores como forma de perceber o crescimento da popularidade de programas computacionais livres para SIG: O primeiro indicador é a quantidade de novos projetos incluídos nos últimos dois anos no site FreeGIS.org. O FreeGIS.org tem como um dos seus objetivos disseminar os programas computacionais livre para SIG. O segundo indicador é crescimento do apoio financeiro que organizações governamentais vêm dando aos projetos de programas computacionais livre para SIG. O terceiro indicador é a quantidade de cópias de programas computacionais livres para SIG *desktop* baixadas nos últimos anos. O quarto indicador é o crescimento de relatos de casos de uso de programas computacionais livres para SIG.

Como um exemplo do uso de programas computacionais livre para SIG, CHEN *et al.* (2010) descrevem o trabalho que foi desenvolvido para definir uma

Plataforma de Gestão e Conhecimento dos Recursos Hídricos a ser utilizada por países em desenvolvimento. Uma Comissão Europeia vinculada ao *Joint Research Centre* foi responsável pelo projeto, que contou também com a colaboração de organizações nacionais e internacionais. A Plataforma integrará ferramentas de apoio e orientações para o setor de gestão de água desses países. As ferramentas de apoio serão baseadas em tecnologias SIG *desktop* de código aberto, visando proporcionar aos usuários uma tecnologia sustentável, tanto do ponto de vista financeiro quanto tecnológico.

Nesse trabalho, CHEN *et al.* (2010) definiram critérios de escolha e análises de programas computacionais de código aberto relacionados a área de Tecnologia da Geoinformação para compor a Plataforma. A análise teve início com centenas de programas que, após sucessivas aplicações de critérios de escolha, foi reduzida a catorze programas. Esses catorze programas foram submetidos a teste de instalação e desempenho. No final, quatro deles foram indicados como adequados ao propósito do estudo: o Quantum GIS (QGIS), o OpenJUMP, o gvSIG e o MapWindow GIS.

SVEEN (2008) apresenta um método para avaliar programas computacionais livres e de código aberto para SIG, com ênfase na utilização por um estabelecimento comercial. O método é composto de quatro etapas que devem ser executadas sequencialmente: (i) levantar as necessidades; (ii) reunir projetos candidatos e realizar uma filtragem inicial; (iii) criar uma ficha de avaliação para os candidatos restantes; (iv) classificar os projetos e fazer uma seleção manual. Os programas indicados foram: uDIG, gvSIG e OpenJUMP.

Outros exemplos de caso de uso de programas computacionais livre para SIG são as experiências de países que adotaram ou estão adotando uma política de código aberto para suas aplicações de cadastro de imóveis.

2.2.1 – Experiências pelo mundo

a) Bavária (Alemanha)

SEIFERT (2010) relata que a Alemanha tem um mapeamento cadastral de boa qualidade e que recentemente iniciou o desenvolvimento de uma infraestrutura de dados espaciais. Essa infraestrutura implementa um novo modelo conceitual de dados baseado em normas internacionais de Tecnologias da Geoinformação. A finalidade do trabalho é integrar o mapa cadastral, os registros de imóveis e o banco de dados topográfico em um único modelo, adicionando também os pontos de referência geodésica.

Na Bavária, o desenvolvimento de aplicativos de SIG é feito por funcionários do governo. Programas computacionais livres são utilizados na captura, no armazenamento de dados e na disponibilização de informações via *web*. Como forma de adquirir independência em relação a fornecedores de programas computacionais, foi tomada a decisão de desenvolver programas computacionais livres, sempre que possível.

Na avaliação do autor, em relação às tecnologias, os requisitos técnicos são satisfeitos pelos programas computacionais livres. Como limitação, é apontada a necessidade de pessoal técnico qualificado para lidar com as tecnologias envolvidas.

b) Bósnia e Herzegovina

HORISBERGER E OSMANOVIĆ (2010) relatam que no período de 2005 a 2007, foi desenvolvido um modelo de dados cadastrais para a Bósnia de acordo com os padrões ISO (*International Organization for Standardization*). O sistema de cadastro foi desenvolvido utilizando programas computacionais livres. A escolha foi motivada por limitações de recursos financeiros.

O aplicativo CAMPUS (*Cadastre Data Management, Processing and Updating Software*) foi implementado, segundo o modelo proposto, como um conjunto de extensões para gvSIG. O SGBD utilizado foi o PostgreSQL com extensão espacial PostGIS.

O CAMPUS atende as principais necessidades de gerenciamento e atualização de dados gráficos e textuais e devido à utilização de um modelo de dados padronizado, é interoperável com o banco de dados de registro de terra, não havendo inconsistências entre as duas bases de dados.

Para HORISBERGER E OSMANOVIĆ (2010), a flexibilidade oferecida pelas soluções de código aberto é um grande trunfo, já que o conhecimento adquirido permanecerá no país.

c) Camboja

PIEPER (2010) relata que, no Camboja, na década de 70, todos os registros de terra do país foram destruídos. O cadastro está sendo reconstruído a partir do zero através de um projeto de registro sistemático da terra, aldeia por aldeia. O projeto conta com o apoio do Banco Mundial e dos governos da Finlândia e da Alemanha.

O sistema computacional Camboja Digital foi desenvolvido para dar suporte ao registro de propriedade de terra. Inicialmente, o sistema utilizou programas computacionais proprietários, foram utilizados o MS Access e ArcView da ESRI ArcGIS. Após sete anos, devido ao crescimento da base de dados, surgiu a necessidade de um SGBD mais robusto e que permitisse compartilhamento de dados. A opção por programas computacionais livres foi motivada pela limitação de recursos financeiros.

O SGBD PostgreSQL com a extensão PostGIS foi adotado, mas o SGBD anterior ainda não foi totalmente desativado, devido à familiaridade do usuário com sua *interface*. O PostgreSQL e o MS Access estão instalados em cada um dos escritórios provinciais. Existe ainda o banco de dados nacional de registro da terra, também PostgreSQL, que é usado para acompanhar o progresso de registro nas províncias e para fins estatísticos. Periodicamente, cópias dos dados dos escritórios são enviadas para o banco de dados nacional.

O registro de propriedade de terra é composto do mapa e das informações sobre a propriedade. Por enquanto, o mapa ainda está sendo gerado a partir do ArcGIS, e apenas as informações sobre a propriedade estão no PostgreSQL. Ao considerar a possibilidade do sistema passar a ser multiusuário, o custo com as licenças do ArcView torna-se preocupante. Então, está sendo avaliada a migração para um programa de SIG livre. O gvSIG foi apontado como um forte candidato.

As vantagens citadas com a substituição do SGBD foram o uso simultâneo do banco de dados por vários usuários, o volume de dados que pode ser armazenado, a confiabilidade e a segurança no armazenamento e no processamento dos dados.

d) Cantão de Solothurn, Suíça

DÜSTER (2010) descreve que o Cantão de Solothurn, Suíça, iniciou a utilização de programas computacionais livres no desenvolvimento de aplicativos SIG para cadastro em 2001, com a introdução do UMN MapServer e a criação de um programa computacional cliente WebGIS. Em seguida, os dados espaciais foram migrados de *shapefiles* para o SGBD PostgreSQL com extensão espacial PostGIS. Em 2006, os SIG *desktop* QGIS e GRASS foram selecionados para substituir o programa computacional proprietário utilizado até o momento.

O principal trabalho executado pelos usuários é a visualização ou captura de dados espaciais com uma *interface* amigável. O QGIS oferece a maioria dos recursos que foram relacionados pelos usuários. Os recursos que faltam podem ser desenvolvidos localmente. Isso vem sendo incentivado no Cantão de Solothurn, que tem investido no desenvolvimento do QGIS desde 2007.

CÂMARA e ONSRUD (2004) destacam que os melhores projetos de programas computacionais de código aberto para SIG são de responsabilidade de empresas, que direcionam esforços para seus fins estratégicos. Defendem que os governos precisam estabelecer projetos públicos financiados, para o desenvolvimento de produtos computacionais de código aberto, adaptados às necessidades locais, a fim de criar os programas computacionais de que necessitam e alcançar a independência tecnológica. Os autores defendem ainda que, para

estabelecer uma política de uso de programa computacional livre, a recomendação de uso através da legislação não é suficiente.

Diante dos casos de uso reportados, é possível constatar o crescimento da utilização dos programas computacionais livres para SIG nos países em desenvolvimento. Pode-se verificar que caminhos comuns estão sendo seguidos por vários países na busca por independência tecnológica e por redução de custos. Por sua vez, a adoção de padrões é uma necessidade para alcançar a interoperabilidade. A junção desses dois fatores é importante para a evolução mais rápida das soluções de TI.

3. CONSÓRCIO INTERNACIONAL OPEN GEOSPATIAL – OGC

A inexistência de padrões para o desenvolvimento dos Sistemas de Informações Geográficas – SIG dificultava a comunicação entre os mesmos. Para sanar esse problema, em 1994, foi criado o *Open Geospatial Consortium* (OGC), inicialmente chamado de OpenGIS, com a missão de criar padrões abertos para permitir a interoperabilidade dos sistemas da área de Tecnologias da Geoinformação (UCHOA *et al.*, 2005a). O OGC foi criado por empresas privadas e posteriormente, instituições de ensino e pesquisa passaram a integrar o consórcio. Os padrões de interoperabilidade OGC mantêm a conformidade com os padrões propostos por outras instituições tais como a ISO.

O OGC trabalha em grupos que se responsabilizam pelos diferentes aspectos envolvidos na interoperabilidade dos SIG. Os principais produtos do trabalho do consórcio são as Especificações OpenGIS. Essas especificações são destinadas a apoiar aos desenvolvedores de programas computacionais na construção das *interfaces* de seus produtos (OGC,2010). Atualmente o termo OpenGIS é uma marca registrada do OGC, que faz referência as diversas especificações do consórcio.

Nas especificações OpenGIS é fornecida uma *interface* comum para desenvolver programas computacionais que irão se comunicar com outros programas computacionais escritos no mesmo padrão. A estrutura fornecida pelo OpenGIS inclui (CÂMARA e QUEIROZ, 2004) :

- O modelo de dados OpenGIS - que se trata de um modelo de dados espaço-temporais e de processos;
- Especificação para as principais linguagens de consulta a banco de dados para implementar o modelo de dados OpenGIS;
- Especificação para implementar o modelo de processo OpenGIS nos principais ambientes computacionais distribuídos.

As especificações OpenGIS dividem-se em duas categorias, as Especificações Abstratas e as Especificações de Implementação ou Padrões de Implementação. As Especificações Abstratas fornecem a base conceitual para a

maior parte das atividades de desenvolvimento das especificações OpenGIS. As Especificações de Implementação, por sua vez, detalham a estrutura de *interface* entre componentes de programas computacionais (OGC,2010).

3.1 – Principais Especificações OpenGIS

Antes de detalhar esse tópico, faz-se necessário os seguintes esclarecimentos:

Cliente-servidor – é um modelo computacional em que há duas entidades trocando informações: o cliente e o servidor. Normalmente essas entidades se encontram em computadores distintos, que são chamados respectivamente de cliente e servidor. O servidor tem a missão de atender as solicitações do cliente. A troca de informações segue uma sequência de ações: o cliente envia uma requisição de serviço ao servidor, que executa o processamento necessário e envia as informações processadas ao cliente (GORNÍ *et al.*, 2007). Para estabelecer a conexão entre o Cliente e o Servidor, utiliza-se programas específicos para essa função (API), disponibilizados pelos fabricantes de programas computacionais. O modelo Cliente/Servidor é apresentado na Figura 1.

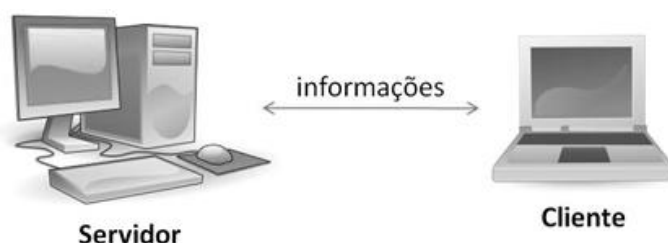


Figura 1 - Modelo Cliente/Servidor

API (*Application Programming Interface*) - é um conjunto de funções que auxiliam a troca de mensagens ou dados entre dois ou mais sistemas (FERREIRA, 2003).

ODBC (*Open Database Connectivity*) - é uma API padrão para utilização de SGBD. Criada pela Microsoft, atualmente é considerada uma *interface* universal de acesso de dados. Com ODBC, um aplicativo pode ser programado para acessar, exibir e modificar dados de vários bancos de dados diferentes simultaneamente (MICROSOFT, 2010).

SQL (*Structured Query language*) – linguagem padrão para manipular bancos de dados relacionais (FERREIRA, 2003).

XML (*eXtensible Markup Language*) - é uma linguagem cuja finalidade é permitir a troca de informações através da Internet. Permite que informações e serviços sejam codificados com estrutura e semântica significativas (YEUNG e HALL, 2007).

SGBD – é um conjunto de programas que gerencia um banco de dados em relação à declaração de sua estrutura de dados, gravação e leitura dos dados, recuperação de falhas de comunicação ou de gravação, controle de concorrência, de acesso e de segurança dos dados (NASSU e SETZER, 1999).

Entre as especificações de implementação OpenGIS, relaciona-se aquelas que tratam do armazenamento, acesso e compartilhamento de dados.

a) SFS (*Simple Features Specification for SQL*)

A SFS define um formato padrão para as aplicações tratarem o armazenamento, a consulta, a atualização e a análise de feições geoespaciais simples em SGBD através de uma API ODBC. O OGC define uma feição simples como uma composição de atributos espaciais e não espaciais, baseada em geometrias bidimensionais com interpolação linear ou planar entre os vértices. A SFS trata de feições geoespaciais simples usando elementos de dados vetoriais, tais como pontos, linhas e polígonos (UCHOA et al., 2005 e OGC, 2006b).

A SFS é dividida em duas partes que descrevem detalhadamente essa *interface*. A primeira define, conceitualmente, o modelo padrão para armazenamento

e acesso as feições pelas aplicações, inclusive as funções para análises topológicas. A segunda parte fornece uma implementação para o modelo definido na primeira parte, no padrão SQL.

O modelo de armazenamento e acesso de dados definido na SFS deve ser implementado pelo SGBD. As aplicações devem interagir com esse modelo.

b) GML (*Geography Markup Language*)

A especificação GML define uma linguagem baseada no formato XML para tratar o intercâmbio e armazenamento de feições geoespaciais. É composta de duas partes, uma delas é o esquema que descreve o documento e a outra parte contém as feições geoespaciais (OGC, 2010);

c) WFS (*Web Feature Service*)

A especificação WFS define padrões de *interfaces* e operações para acessar e manipular dados espaciais através do ambiente *web*. O formato utilizado para o intercâmbio de dados é a GML. A WFS trabalha com dados vetoriais. Este serviço pode ser solicitado por uma aplicação cliente utilizando as operações (QUEIROZ e FERREIRA, 2006):

- Selecionar feições com base em restrições espaciais e não espaciais;
- Criar uma nova instância da feição;
- Obter uma descrição das propriedades da feição;
- Excluir uma instância da feição;
- Atualizar uma instância da feição;
- Bloquear uma ou mais instâncias da feição.

d) WCS (*Web Coverage Service*)

A WCS especifica um padrão de *interface* e operações que permite a interoperabilidade para acesso a dados que representam fenômenos com variação contínua no espaço, através do ambiente *web*. Este serviço fornece os dados

juntamente com sua descrição e não imagens como é o caso da WMS (DAVIS JR. *et al.*, 2005).

e) WMS (*Web Map Service*)

WMS padroniza a *interface* dos serviços de produção de mapas dinâmicos na *web*. Define como a aplicação cliente solicita a geração dos mapas aos servidores e a maneira como estes servidores devem descrever e retornar os mapas (QUEIROZ e FERREIRA, 2006).

Os servidores disponibilizam informações sobre os mapas que podem ser solicitados e os parâmetros que devem ser informados na solicitação, tais como camadas e a área de interesse. A aplicação cliente efetua a solicitação, informando os parâmetros necessários. O servidor gera mapas georreferenciados, como uma imagem ou um conjunto de objetos gráficos, e envia para a aplicação cliente. A aplicação cliente organiza as imagens retornadas compondo seu mapa. Todo o processo de geração do mapa é efetuado no servidor de mapas.

A Figura 2 ilustra a interação entre as especificações OpenGIS e as aplicações.

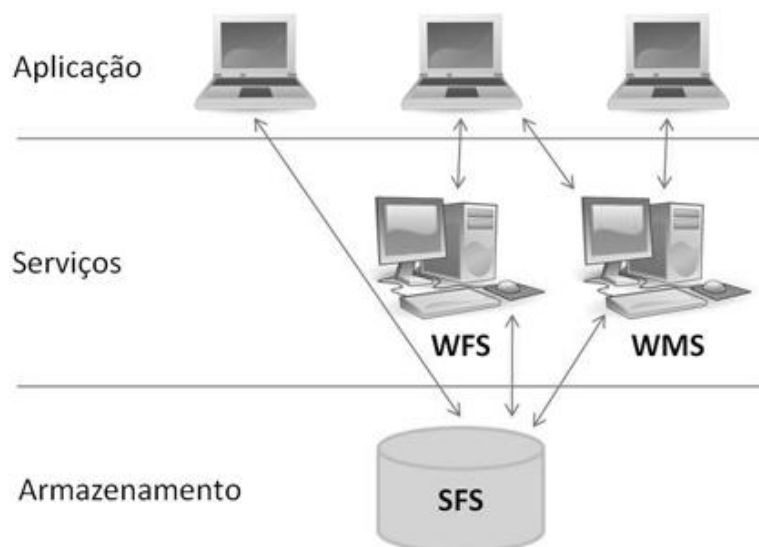


Figura 2 - Interação entre as Especificações OpenGIS e as aplicações

UCHOA e FERREIRA (2004) reforçam a importância da especificação SFS que define a organização dos dados espaciais no banco de dados geográfico e as funções de análise de um SIG. Por isso aconselham que uma instituição, ao contratar uma solução livre, exija que os dados vetoriais sejam armazenados de acordo com essa especificação.

Segundo UCHOA *et al.* (2005a), houve uma coincidência entre o momento que as especificações do OGC foram definidas e os projetos de programas computacionais livres expadiram-se para a área de Tecnologias da Geoinformação. Dessa forma os programas computacionais livres, em geral, já foram desenvolvidos seguindo os padrões recomendados pelo OGC.

Para verificar se um produto está de acordo com uma especificação OGC, existe o *Compliance & Interoperability Testing & Evaluation* (CITE) também conhecido como teste de conformidade. O teste de conformidade determina se o produto preenche todos os requisitos obrigatórios de uma determinada versão de uma especificação OGC. A verificação de conformidade não garante a interoperabilidade, apenas aumenta sua probabilidade (OGC,2011a).

O OGC disponibiliza junto às especificações os roteiros dos testes que são aplicados sobre os produtos. Assim, o consórcio sugere que a solicitação de verificação seja feita somente depois que for obtido sucesso nos testes executados pelo próprio fabricante. Uma vez atestada a conformidade, o fabricante paga uma licença anual e pode então associar o certificado de conformidade ao produto. O certificado representa uma garantia tanto para o fornecedor quanto para o cliente.

Até o momento os testes de conformidade existem apenas para produtos servidores. Os SGBD como servidores de dados podem ser submetidos a esses testes. No endereço <<http://www.opengeospatial.org/resource/products/compliant>> ficam registrados os produtos certificados e a indicação da especificação/versão da certificação.

4. SISTEMAS GERENCIADORES DE BANCOS DE DADOS E SISTEMAS DE INFORMAÇÕES GEOGRÁFICAS

4.1. Sistemas Gerenciadores de Bancos de Dados

Segundo RIGAUX *et al.* (2002), um Sistema Gerenciador de Banco de Dados (SGBD) é um conjunto de programas computacionais desenvolvido para gerenciar as estruturas e controlar o acesso a bancos de dados. São funcionalidades propiciadas pelo SGBD:

- Criar bancos de dados, através da definição das estruturas de dados;
- Armazenar os dados nas estruturas definidas;
- Manipular o banco de dados (incluir, alterar e remover dados);
- Recuperar dados de consultas específicas.

WORBOYS (1995) enumera algumas atribuições dos SGBD: garantir a integridade de dados, permitir que vários usuários utilizem e atualizem os dados simultaneamente, oferecer controle de transações, permitir bloqueio do dado para atualização, controlar o acesso, gerenciar grandes volumes de dados, oferecer mecanismos de cópia de segurança e recuperação, oferecer facilidades de acompanhamento de desempenho, dispor de uma linguagem padrão de definição e manipulação de estruturas de dados, entre outras.

Entre as atribuições dos SGBD, o controle de transações deve ser explicado mais detalhadamente. Segundo SILBERSCHATZ *et al.* (1999), uma transação é uma coleção de operações que desempenha uma função lógica única em uma aplicação. Uma transação é parte de um programa, delimitada por declarações de início e de fim, que acessa e pode atualizar vários registros de várias tabelas, sendo que essa atualização só tem sentido se for feita por inteiro. O controle de transações implica em o SGBD administrar a execução da transações de forma a garantir a integridade dos dados. Para isso, o SGBD deve implementar as seguintes propriedades:

- Atomicidade – ou todas operações de uma transação são efetuadas sobre o banco de dados ou nenhuma. Caso aconteça algum erro, o SGBD retorna os registros afetados para o estado anterior ao início da transação;

- Consistência – a execução de uma transação isolada preserva a consistência do banco de dados;
- Isolamento – garantia de que as transações concorrentes não tomam conhecimento umas das outras;
- Durabilidade – após o fim de uma transação, as modificações que foram feitas no banco de dados persistem mesmo em caso de falhas no banco de dados.

Dos diversos tipos de SGBD existentes, aqueles que têm seu emprego consolidado atualmente são o Relacional e o Objeto-relacional (QUEIROZ e FERREIRA , 2006).

O SGBD Relacional é baseado no modelo relacional proposto por Codd. Esse modelo considera as informações em uma base de dados como relações matemáticas que estão representadas através de tabelas bidimensionais. (MACHADO e ABREU, 1996).

A linguagem padrão para interagir com o SGBD Relacional é a SQL - *Structured Query Language*. A SQL é composta de duas sub-linguagens, a DDL (*Data Description Language*) e a DML (*Data Manipulation Language*). A DDL disponibiliza comandos para gerenciar as estruturas do banco de dados, tais como: criar, alterar, remover tabelas, colunas, índices e restrições de integridade. A DML disponibiliza comandos para manipular os dados armazenados no banco de dados, tais como: consultar, incluir, alterar e remover dados das tabelas (QUEIROZ e FERREIRA , 2006).

A SQL é baseada na álgebra relacional que, por sua vez, baseia-se em operações sobre conjuntos e operações relacionais. São exemplos das operações sobre conjuntos: união, interseção, diferença e produto cartesiano. E das operações relacionais são: seleção, projeção e junção (NASSU e SETZER, 1999).

Os tipos de dados que normalmente estão disponíveis nos SGBD Relacionais são: números inteiros, de ponto flutuante, cadeias de caracteres, datas e campos binários longos (BLOB - *binary large object*). Também se encontram disponíveis

várias operações sobre os tipos de dados, exceto para o tipo BLOB (QUEIROZ e FERREIRA, 2006). Esses tipos de dados, chamados tipos pré-definidos, são inadequados para o armazenamento dos dados complexos, como os dados espaciais.

Os SGBD objeto-relacionais (SGBD-OR) foram desenvolvidos para superar as limitações dos SGBD relacionais em lidar com dados complexos. Estendem o modelo relacional, permitindo a criação de novos tipos de dados e das suas respectivas funções de manipulação, bem como a criação de novos mecanismos de indexação (YEUNG e HALL, 2007). Os SGBD-OR viabilizaram a criação das extensões espaciais para tratamento desse tipo de dados.

As extensões espaciais devem disponibilizar estruturas de armazenamento e funções específicas para manipulação de dados espaciais, além de fornecer recursos de indexação.

4.1.1 – Estruturas de armazenamento de dados espaciais

O OGC, através da especificação SFS – *Simple Features Specification for SQL* define a estrutura de armazenamento de feições espaciais e a semântica dos operadores ou funções a serem utilizados em consultas espaciais. Essa especificação contempla apenas os dados espaciais em formato vetorial (OGC, 2006b).

A SFS apresenta duas formas de implementação das estruturas de dados no SGBD. Em uma delas, baseada nos tipos de dados pré-definidos, o uso do SGBD está restrito ao armazenamento dos dados espaciais, e na outra, baseada em SGBD com extensões espaciais, o SGBD provê além do armazenamento, a implementação das funções de apoio a análise dos dados (OGC, 2006b).

A Figura 3 apresenta o esquema de armazenamento baseado em extensões espaciais proposto. A Figura não foi traduzida porque na implementação pelos SGBD, os nomes das tabelas e das colunas permanecem os mesmos.

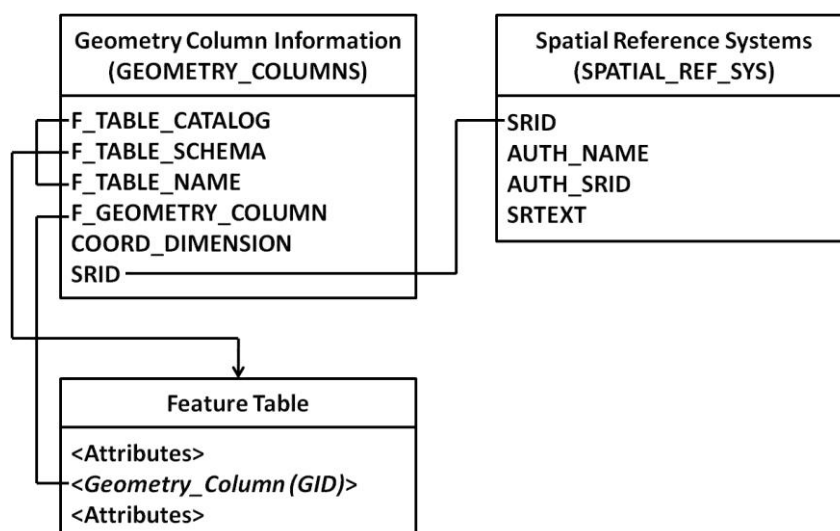


Figura 3 - Esquema de armazenamento OGC baseado em extensões espaciais

Fonte: OGC (2006b)

As tabelas da figura 3 são assim descritas:

- *SPATIAL_REF_SYS*: tabela de metadados que armazena informações sobre os Sistemas de Referência Espaciais cadastrados no banco de dados;
- *GEOMETRY_COLUMNS*: tabela de metadados do conteúdo da tabela *FEATURE_TABLE*. Os dados armazenados são: sistema de referência, nome da tabela, nome da coluna, entre outros;
- *FEATURE_TABLE*: Tabelas para o armazenamento das feições geoespaciais. As colunas representam os atributos e as linhas representam as feições. A geometria da feição é um dos atributos e é mapeada em uma coluna do tipo geométrico, que estende os tipos básicos oferecidos pela SQL.

O esquema de armazenamento descrito obedece à hierarquia de classes definida no modelo conceitual das geometrias que é especificado na primeira parte da SFS. A Figura 4 apresenta o principal diagrama de classes dessa especificação. Foi mantido o texto em inglês porque é assim que os tipos de geometria aparecem em todos os locais onde são implementados.

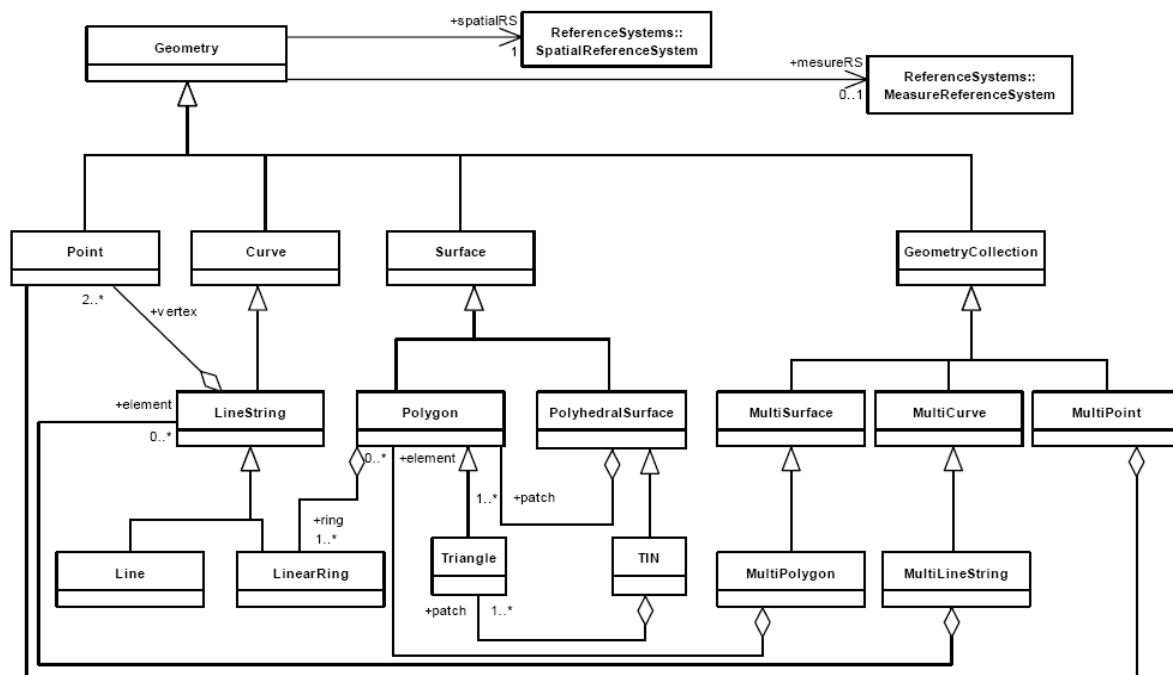


Figura 4 - Hierarquia de Classes Geométricas

Fonte: OGC (2006a)

Na hierarquia de classes apresentada na Figura 4, a superclasse *Geometry* é uma classe abstrata, ou seja, as demais classes herdam as suas características, mas não é possível criar um objeto dessa classe. A classe *Geometria* (*Geometry*) possui subclasses para representar os tipos geométricos ponto (*Point*), cadeia de linhas (*LineString*), polígono (*Polygon*), coleção de geometrias (*GeometryCollection*), coleção de pontos (*MultiPoint*), coleção de cadeias de linhas (*MultiLineString*) e coleção de polígonos (*MultiPolygon*). Cada objeto geométrico é associado a um sistema de referência espacial, que descreve o espaço de coordenadas no qual o objeto é definido (OGC, 2006a).

4.1.2 - Funções específicas para manipulação de dados espaciais

As funções para manipulação dos dados espaciais são definidas pelos métodos previstos em cada classe e que definem as operações permitidas para a classe. O OGC propõe um conjunto de métodos que são descritos em OGC (2006a) e MARTINS (2006).

A classe raiz *Geometria* (*Geometry*) apresenta métodos que são classificados em três diferentes tipos entre os quais se encontram: básicos, de relacionamentos

espaciais e de análises espaciais. Esses métodos são válidos para todas as classes derivadas da classe Geometria. Por sua vez, cada subclasse apresenta seus métodos específicos. O Anexo 1 apresenta um subconjunto das funções para manipulação de dados espaciais.

As funções espaciais são historicamente implementadas nos programas de SIG. Segundo ZLATANOVA e STOTER (2006), a pergunta sobre quem deve ser o responsável pelas funções de análise espaciais ainda está em aberto e será alvo de discussões. Os autores ponderam que, embora vários estudos demonstrem que é melhor executar operações espaciais no servidor de dados, os fornecedores de SIG podem não estar dispostos a abrir mão de análises espaciais, como também o desempenho das funções implementadas no SIG pode ser melhor.

4.1.3 - Recurso de Indexação

Os índices são recursos utilizados para melhorar o desempenho das consultas ao banco de dados, evitando a leitura exaustiva das tabelas para encontrar o resultado de uma consulta.

Em bancos de dados espaciais, os índices devem levar em consideração a localização espacial dos objetos, que são informações multidimensionais, o que torna sua implementação mais complexa que a dos índices unidimensionais sobre dados alfanuméricos.

Devido à complexidade, os índices espaciais trabalham com representações simplificadas dos objetos, como o menor retângulo envolvente (REM). Dessa forma, a maneira mais comum de executar uma consulta é dividir seu processamento em duas etapas. Na primeira etapa, chamada etapa de filtragem, são usados os índices espaciais com o objetivo de reduzir e rapidamente selecionar os possíveis candidatos que satisfaçam a consulta. A redução do espaço de busca é muito importante, pois a segunda etapa, a de refinamento, envolve a aplicação de algoritmos complexos e custosos à geometria exata dos candidatos selecionados na etapa anterior (QUEIROZ e FERREIRA, 2006). A Figura 5 representa a divisão de consultas em duas etapas.

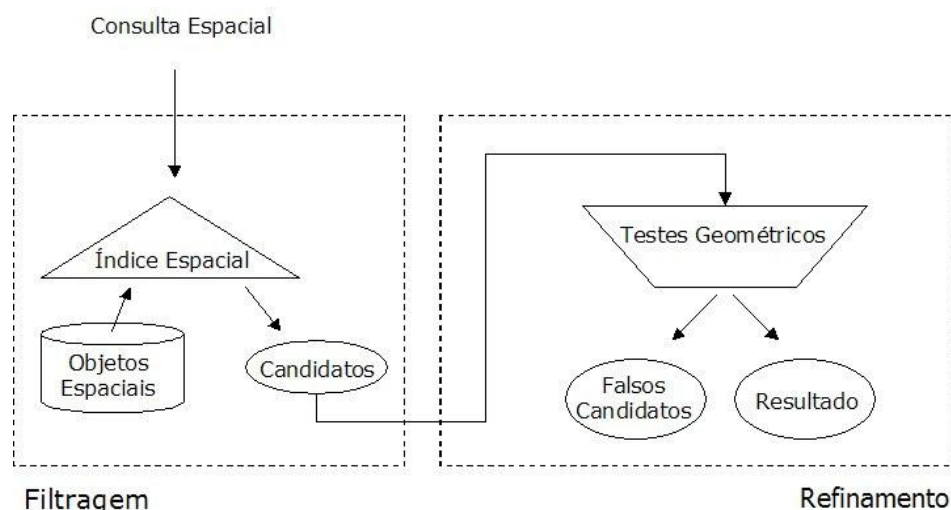


Figura 5 - Processamento de consultas espaciais

Fonte: QUEIROZ e FERREIRA (2006)

Segundo YEUNG e HALL (2007), as aplicações de mapeamento, análise e gerenciamento de dados espaciais funcionam de forma mais eficiente com o apoio de um sistema de banco de dados espacial.

Os SGBD envolvidos nessa pesquisa são o PostgreSQL/PostGIS e o MySQL.

4.1.4 – SGBD PostgreSQL/PostGIS

O PostgreSQL é um SGBD-OR de código aberto, descendente do Postgres. O projeto Postgres, com o objetivo de projetar um SGBD-OR, teve início em 1986 e se estendeu até 1994, na Universidade de Berkeley na Califórnia, sob a coordenação de Michael Stonebraker e patrocinado pela Defense Advanced Research Projects Agency (DARPA), pelo Army Research Office (ARO), pela National Science Foundation (NSF) e pela ESL, Inc. A primeira versão operacional foi liberada em 1987, e em 1988 o Postgres foi apresentado na Conferência ACM-SIGMOD. Quando o projeto foi encerrado em 1994, o Postgres estava na versão 4.2. Em seguida, no período de 1994 a 1995, dois alunos de Berkeley, Jolly Chen e Andrew Yu, adicionaram o suporte à linguagem SQL ao Postgres, que passou a se chamar Postgres95 e ter o código aberto. Após saírem da Universidade, apenas Chen continuou mantendo e desenvolvendo o Postgres95 junto com uma pequena equipe.

Em 1996, o nome foi alterado para PostgreSQL para refletir sua origem relacionada ao projeto Postgres e suas versões posteriores com SQL. A numeração da versão iniciou de 6.0, retomando a sequência do Postgres de Berkeley. (UCHOA *et al*, 2005b; POSTGRESQL, 2011)

Esse SGBD segue a arquitetura cliente-servidor. Assim, uma sessão do PostgreSQL consiste de dois programas. Um programa servidor chamado postmaster que gerencia os arquivos de banco de dados, aceita as conexões dos programas clientes com o banco de dados e executa as ações solicitadas pelos clientes no banco de dados. O outro programa, o cliente, que deseja executar operações sobre o banco de dados. O programa que faz o papel de cliente pode ser de natureza variada, como por exemplo: um aplicativo gráfico, um servidor de internet, uma ferramenta de manutenção e administração do banco, entre outros. O servidor PostgreSQL tem a capacidade de atender várias conexões de clientes simultaneamente (POSTGRESQL, 2011).

O cliente e o servidor podem estar na mesma máquina ou em máquinas diferentes. Nesse caso, a comunicação entre o cliente e o servidor se dá através de uma conexão de rede TCP/IP. Segundo PARKER (1996), TCP/IP (*Transmission Control Protocol/Internet Protocol*) é um conjunto de programas computacionais que fornece um método de transferência de informações entre máquinas. Cada máquina é associada a um endereço IP, que é um número de identificação única na rede. As portas são pontos de entrada e saída das máquinas, identificadas por números e que são associados às aplicações TCP/IP. O PostgreSQL normalmente é associado à porta número 5432.

Para aumentar a eficiência das consultas, o PostgreSQL implementa três tipos de índice:

- Índices Árvore-B: organiza os dados em árvores binárias. Aplica-se apenas a dados dos tipos básicos (inteiro, cadeia de caracteres, etc);
- Índices Árvore-R: constroem árvores abrindo os dados em retângulos, sub-retângulos, etc. Esse tipo de índice é utilizado em vários bancos de dados espaciais, entretanto a implementação do PostgreSQL para o R-

Tree não é robusta e apresenta limitações para o armazenamento das colunas;

- Índices GiST (*Generalized Search Trees* – Árvore de Busca Generalizada): forma genérica de indexação que pode ser utilizada para acelerar as pesquisas em todos os tipos de estruturas de dados que não podem ser indexados por índices normais (exemplo: arranjos de números inteiros, dados espectrais) (POSTGIS, 2010).

Entre os tipos de dados nativos, além dos tipos básicos (inteiro, cadeia de caracteres, etc) utilizados para representar os atributos alfanuméricos, o PostgreSQL implementa tipos geométricos, como ponto (*point*), segmento de linha (*lseg*), caixa (*box*), caminho (*path*), polígono (*polygon*) e círculo (*circle*), além de algumas funções geométricas, operadores e funções de conversão de tipos (POSTGRESQL, 2011). Entretanto, não oferece geometrias que possam representar objetos complexos, como os formados por conjunto de polígonos, como também, os operadores espaciais realizam a computação apenas sobre o retângulo envolvente das geometrias e não diretamente sobre elas (QUEIROZ e FERREIRA, 2005). Estes tipos de geometria não seguem os padrões *OpenGIS Consortium*, nem têm suporte de sistemas de coordenadas espaciais (OBE e HSU, 2011). Buscando resolver esses problemas, foi desenvolvida uma extensão espacial para o PostgreSQL.

O PostGIS é uma extensão espacial de código fonte aberto para o PostgreSQL, mantida pela empresa *Refractions Research Inc.*, que segue as especificação SFS do OGC. Essa extensão envolve o armazenamento de dados espaciais e diversas funcionalidades topológicas, possibilitando o desenvolvimento de SIG Corporativos e interoperáveis. (UCHOA *et al*, 2005b). Desde a versão 0.9, o PostGIS implementa todos os objetos e funções recomendados pela especificação SFS e estende o padrão implementando também para 3 dimensões (POSTGIS, 2010).

O PostGIS estende o PostgreSQL através da criação do tipo de dados *GEOMETRY* que, por sua vez, permite os subtipos: ponto (*point*), linha (*linestring*), polígono (*polygon*), multiponto (*multipoint*), multilinhas (*multilinestring*), multipolígono (*multipolygon*) e coleção de geometrias (*geometrycollection*), como mostra a Figura

6. *GEOMETRY* é um tipo de dados espacial usado para representar de forma planar os dados espaciais nas tabelas com feições geográficas. A seguir é mostrada a composição de uma tabela com uma coluna do tipo *GEOMETRY* chamada *the_geom*.

```
CREATE TABLE pe
(
  gid serial NOT NULL,
  geocodigo character varying(7),
  nome character varying(32),
  uf character varying(2),
  the_geom geometry,
  CONSTRAINT pe_pkey PRIMARY KEY (gid)
);
```

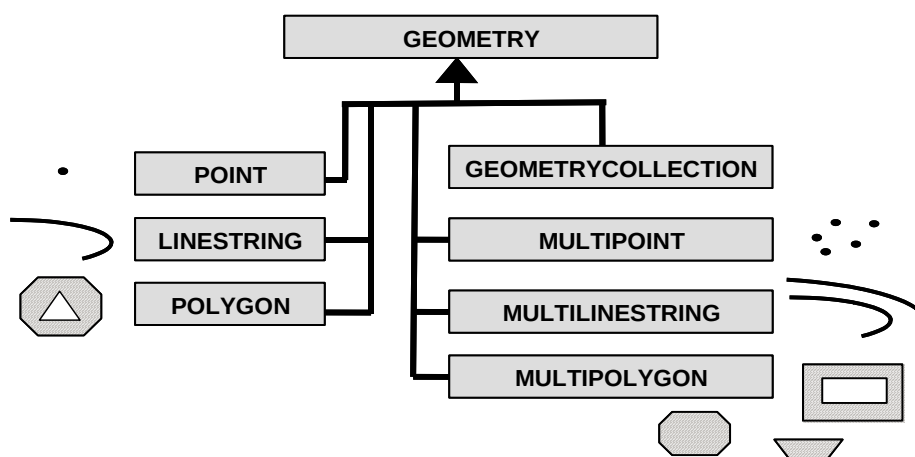


Figura 6 - Tipos de dados espaciais do PostGIS

Fonte: (QUEIROZ e FERREIRA, 2005)

Seguindo as recomendações da especificação SFS, o PostGIS implementa duas tabelas de metadados: *SPATIAL_REF_SYS* e *GEOMETRY_COLUMNS*. Essas tabelas são criadas no momento da instalação do PostGIS.

A tabela *SPATIAL_REF_SYS* é usada pelo PostGIS para catalogar todos os os sistemas de referência espaciais disponíveis no banco de dados. Contém o nome do sistema de referência espacial e os parâmetros necessários para transformar para outro sistema (OBE e HSU, 2011). No PostGIS essa tabela é inicialmente

carregada com mais de 3000 sistemas de referência, entretanto não contempla todas as possibilidades, podendo ser incluído outros sistemas de referência quando necessário. A maior parte dos sistemas de referência espaciais catalogados na tabela são definidos pela EPSG - *European Petroleum Survey Group* (www.epsg.org).

Na composição da tabela *SPATIAL_REF_SYS*, o PostGIS segue as recomendações da especificação SFS e acrescenta a coluna *PROJ4TEXT* para uso da biblioteca Proj4 no momento de executar as transformações. No Anexo 2 é mostrada a composição da tabela *SPATIAL_REF_SYS* do PostGIS e o significado de suas colunas.

Na composição da tabela *GEOMETRY_COLUMNS*, o PostGIS segue as recomendações da especificação SFS e acrescenta a coluna *TYPE*, que contém o tipo da geometria e pode assumir os valores: *POINT*, *LINESTRING*, *POLYGON*, *MULTIPOINT*, *MULTILINESTRING*, *MULTIPOLYGON*, *GEOMETRYCOLLECTION*. No Anexo 2 é mostrada a composição da tabela *GEOMETRY_COLUMNS* do PostGIS e o significado de suas colunas.

No PostGIS, a coluna do tipo *GEOMETRY* deve ter três restrições de verificação (*check constraints*) padrão, são elas:

- *enforce_dims_the_geom* - assegura que em todas as linhas da tabela a geometria armazenada na coluna tenha a dimensão definida na restrição;
- *enforce_geotype_the_geom* - assegura que em todas as linhas da tabela a geometria armazenada na coluna seja tipo definido na restrição;
- *enforce_srid_the_geom* - assegura que em todas as linhas da tabela a geometria armazenada na coluna pertençam ao sistema de referência definido na restrição.

Restrição de verificação são implementadas nos SGBD para permitir especificar que o valor de uma coluna deve satisfazer uma expressão fornecida. Normalmente a restrição é criada juntamente com a coluna afetada.

Quando uma coluna do tipo *GEOMETRY* é criada ou removida da base de dados, a tabela *GEOMETRY_COLUMNS* deve ser atualizada. Para evitar a atualização manual da tabela, que é trabalhosa e pode provocar erros, o PostGIS implementa cinco comandos que executam essa tarefa. Esses comandos estão descritos no Anexo 2, bem como um exemplo de como deve ser feita a criação de uma tabela com uma coluna do tipo *GEOMETRY* no PostGIS.

O PostGIS disponibiliza grande parte das funções definidas pela especificação SFS, bem como muitas outras que são exclusivas (OBE e HSU, 2011). São oferecidos mais de 300 funções e operandos através do uso das bibliotecas Proj4, GEOS e LibXML2. A biblioteca Proj4 é usada para fazer a transformação de coordenadas dos objetos. A biblioteca GEOS utilizada para fazer testes entre geometrias e a biblioteca LibXML2 utilizada em algumas funções de importação de geometrias (POSTGIS, 2010).

O Postgis implementa ainda o tipo *GEOGRAPHY* que fornece suporte para os recursos espaciais representados em coordenadas geodésicas expressas em unidades angulares. O caminho mais curto entre dois pontos sobre a esfera é um arco do círculo. Assim, cálculos de regiões – áreas, distâncias, comprimento, etc – devem ser calculados sobre a esfera, que envolve cálculos mais complexos que os efetuados sobre o plano. Uma restrição, é que o tipo *GEOGRAPHY* só trabalha com o sistema geodésico WGS-84. Ressalta-se que o formato padrão OGC são suportados, exceto para curvas. As funções da biblioteca GEOS não dão suporte ao tipo de dados *GEOGRAPHY* (POSTGIS, 2010).

Após a criação da tabela associada a geometria, os dados podem ser carregados através de comandos SQL. No caso de grandes volumes de dados, a carga pode ser feita através de utilitários. Para a conversão entre arquivos no formato *shape* e o PostGIS existe o *shp2pgsql*.

Existem *interfaces* gráficas livres que permitem a gerência do PostgreSQL de maneira simples: *phpPgAdmin* e *pgAdmin III*. Estas *interfaces* facilitam a gerência dos bancos de dados, tornando os processos mais intuitivos. (UCHOA *et al*, 2005b).

4.1.5 - SGBD MySQL

O MySQL é um SGBD relacional cliente-servidor que tem sua origem no Unireg, um sistema de banco de dados desenvolvido na década de 1980 por Michael Widenius para uma empresa sueca chamada TcX. Em 1995, quando Michael adicionou uma *interface* SQL para o Unireg, surgiu a primeira versão do servidor MySQL. Pouco tempo depois, David Axmark recomendou que o servidor MySQL fosse liberado sob um modelo de licenciamento duplo, onde ele poderia estar disponível para uso gratuito e também poderia ser utilizado em situações que requisitassem um modelo de licenciamento mais restritivo. Ainda em 1995, David e Michael, juntamente com Allan Larsson fundaram a empresa MySQL AB que forneceu suporte ao MySQL e desenvolveu a maior parte do seu código até o início de 2008, quando foi comprada pela Sun Microsystems (CABRAL e MURPHY, 2009). Em 2010, ao comprar Sun Microsystems, a Oracle passou a ter controle sobre o MySQL.

A respeito de licenças de uso, atualmente a Oracle fornece um modelo de licença dupla (GPL e comercial) para seu servidor de banco de dados MySQL e suas bibliotecas clientes. Para os projetos que envolvem apenas licenças GPL, a licença do MySQL é também GPL (MySQL, 2010). As demais situações devem ser verificadas.

O MySQL, embora seja um SGBD Relacional, implementa uma extensão espacial na qual disponibiliza um conjunto de tipos de dados que corresponde as classes propostas na especificação SFS e que são usados para representar de forma planar os dados espaciais nas tabelas com feições geográficas. São eles: geometria (*geometry*), ponto (*point*), linha (*linestring*), polígono (*polygon*), multiponto (*multipoint*), multilinha (*multilinestring*), multipolígono (*multipolygon*) e coleção de geometria (*geometrycollection*). O tipo de dados *geometry* pode armazenar valores de geometrias de qualquer tipo, bem como o tipo de dados *geometrycollection* pode armazenar coleções de objetos de qualquer tipo. Os demais tipos de dados são restritos ao tipo de geometria para o qual são definidos (MYSQL, 2011a). A Figura 7 apresenta os tipos de dados espaciais do MySQL. Na figura, as classes apresentadas em cor cinza, representam classes abstratas, ou seja, as demais

classes herdam as suas características, mas não é possível criar um objeto dessa classe.

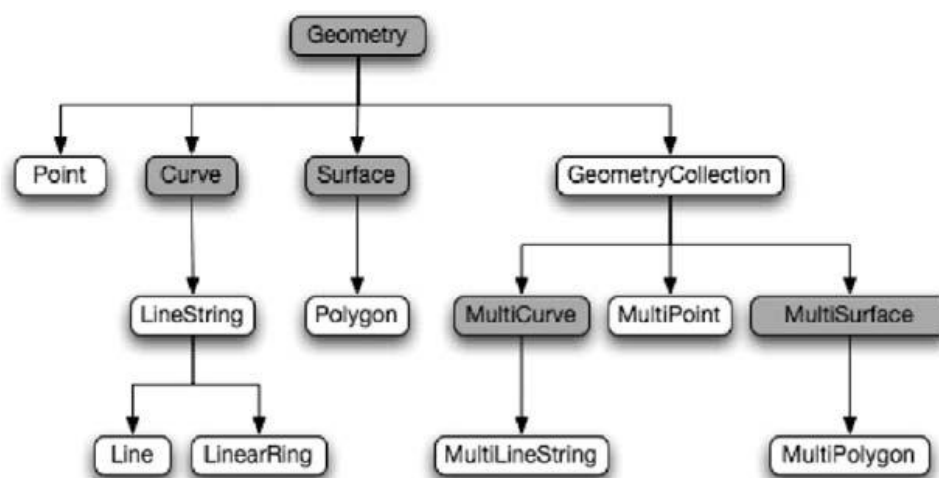


Figura 7 - Tipos de dados espaciais do MySQL

Fonte: (KARLSSON, 2011)

Em relação a compatibilidade com a especificação SFS, o MySQL na versão 5.5, apresenta algumas divergências como (MYSQL, 2011a):

- Não implementa as tabelas de metadados;
- A função `Length()` está implementada com o nome de `GLength()` devido a existir outra função no MySQL com a nome `Length()`;
- Implementa parcialmente as funções;
- Entre as funções implementadas, as que testam relacionamento entre geometrias não estão de acordo com a especificação.

As funções que não são implementadas pelo MySQL são: *Boundary(g)*, *IsEmpty(g)*, *IsSimple(g)*, *IsRing(ls)*, *Centroid(mpoly)*, *PointOnSurface(mpoly)*, *Buffer(g,d)*, *ConvexHull(g)*, *Difference(g1,g2)*, *Intersection(g1,g2)*, *SymDifference(g1,g2)* e *Union(g1,g2)* (MYSQL, 2011a).

As funções a seguir executam a análise sobre o retângulo envolvente e não sobre as geometrias como recomendado pelo OpenGIS: *Contains(g1,g2)*, *Crosses(g1,g2)*, *Disjoint(g1,g2)*, *Equals(g1,g2)*, *Intersects(g1,g2)*, *Overlaps(g1,g2)*, *Touches(g1,g2)*, *Within(g1,g2)* (MYSQL, 2011a).

O MySQL disponibiliza alguns tipos de tabela para o armazenamento e o gerenciamento dos dados. Cada um desses tipos tem suas características e, o usuário faz sua escolha de acordo com suas necessidades. Dos tipos de tabela que trabalham com a extensão espacial, os mais usados são o InnoDB e o MyISAM:

- InnoDB é um tipo de tabela que implementa os recursos envolvidos na segurança das operações e oferece controle de transações. A partir da versão 5.5, este é o tipo de tabela padrão (MYSQL, 2011a);
- MyISAM é um tipo de tabela muito utilizado em aplicações para internet. Não oferece controle de transação e é o único tipo de tabela que oferece suporte de índices sobre dados espaciais. Oferece rapidez no acesso e ocupação de espaço em disco reduzida. Nas versões anteriores a 5.5, era o tipo de tabela padrão do MySQL (MYSQL, 2011a; CABRAL e MURPHY, 2009).

Quanto aos recursos de indexação, até a versão 5.5, o MySQL implementa índice espacial sobre as colunas espaciais (geométricas) apenas para as tabelas MyISAM. O índice espacial disponibilizado é do tipo Árvore-R e é construído usando o retângulo envolvente das geometrias. Também é possível criar índices de Árvore-B, normalmente criado sobre colunas alfanuméricas, para colunas espaciais. Tabelas MyISAM suportam os dois tipos de índices e as demais tabelas suportam apenas índices de Árvore-B. Atualmente, as colunas para as quais são criados índices espaciais, devem ser de preenchimento obrigatório. Os índices de Árvore-B sobre colunas espaciais são utilizados apenas para pesquisa de valores exatos e não para intervalo de valores (MYSQL, 2011a).

Após a criação da tabela associada a geometria, os dados podem ser carregados através de comandos SQL. No caso de grandes volumes de dados, a carga pode ser feita através de utilitários como o shp2mysql, que converte arquivo shape em MySQL.

Para administrar o banco de dados MySQL, existem alguns programas que fornecem uma interface gráfica que facilitam essa tarefa. O MySQL Workbench é um desses programas. O MySQL Workbench é compatível com as versões 5.1 do MySQL e superiores, oferecendo três funcionalidades (MYSQL, 2011b):

- *SQL Development*: permite criar, gerenciar e configurar conexões com o banco de dados, oferecendo também a facilidade de executar consultas SQL utilizando um editor embutido;
- *Data Modeling*: Permite modelar graficamente o banco de dados e oferece facilidade de uso para editar tabelas, colunas, índices, gatilhos, visões, entre outros;
- *Server Administration*: permite administrar o servidor MySQL.

O MySQL Workbench tem duas versões, a *Community Edition* disponibilizada gratuitamente e a *Standard Edition* disponibilizada comercialmente, conta com mais recursos.

4.2 - Sistemas de Informações Geográficas

TOMLINSON (2007), afirma que uma definição simples não é suficiente para caracterizar os SIG, devido a sua abrangência, ao tipo de dados e ao tipo de operações que o compõem. Afirma ainda que o termo dado espacial tem um significado especial em SIG. O autor descreve os dados espaciais como dados que tem uma ligação geográfica, ou seja, parte dele está relacionada a uma referência geográfica, normalmente representada por combinações de pontos, linhas ou polígonos. Os dados espaciais são compostos por sua representação geográfica e descritiva.

CÂMARA *et al.* (1996), definem SIG como uma tecnologia que tem os seguintes componentes: *interface* com o usuário; entrada e integração de dados; funções de processamento; visualização e plotagem; e armazenamento e recuperação de dados. A Figura 8 mostra os componentes e o relacionamento entre eles.

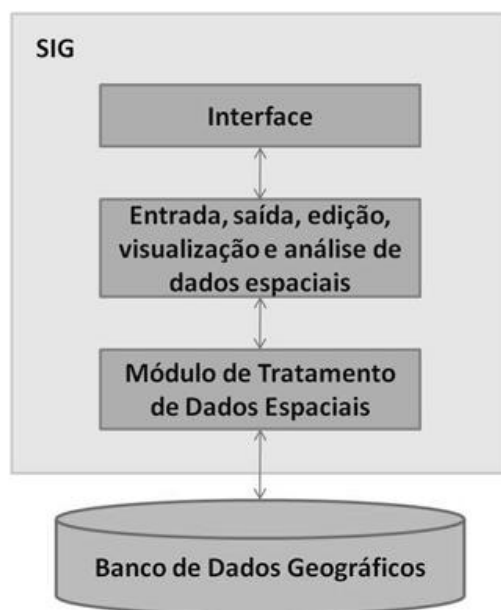


Figura 8 - Arquitetura de Sistemas de Informações Geográficas

A arquitetura apresentada é formada por três camadas, que se relacionam de forma hierárquica. A camada superior, *interface* homem-máquina, define como o sistema é operado e controlado. Na camada intermediária, o SIG deve ter mecanismos de entrada, saída, edição, visualização e análise dos dados espaciais. Na camada inferior, o Módulo de Tratamento de Dados Espaciais provê o mecanismo de armazenamento e recuperação dos dados espaciais.

Essa última camada, por ser alvo dessa dissertação, será a única detalhada.

4.2.1 - Módulo de Tratamento de Dados Espaciais

Uma mudança que vem ocorrendo nos programas computacionais para SIG é a forma como os dados são gerenciados. Os avanços dos SGBD levaram a criação de uma arquitetura chamada integrada. Essa arquitetura utiliza os recursos de um SGBD para gerenciar tanto as representações gráficas quanto os atributos descritivos dos objetos geográficos.

Segundo CÂMARA e QUEIROZ (2004), há basicamente duas diferentes arquiteturas integradas de gerenciamento de dados: arquitetura baseada em SGBD

relacionais e arquitetura baseada em extensões espaciais sobre SGBD objeto-relacionais.

A arquitetura integrada baseada em SGBD Relacionais se caracteriza por utilizar campos tipo BLOB para armazenar a representação geográfica do dado espacial. ZLATANOVA e STOTER (2006) apresentam como dificuldades dessa alternativa:

- A linguagem SQL oferece recursos limitados para o tratamento dos campos BLOB;
- Em consequência, o processamento do dado espacial é efetuado pelo SIG, que trata os dados no formato BLOB, ficando o SGBD apenas como repositório.

A arquitetura integrada baseada em extensões espaciais sobre SGBD Objeto-Relacionais permite a criação de estruturas padronizadas para armazenar os dados e oferece métodos para acesso, manutenção e análise dos dados espaciais em formato vetorial. As limitações dessa arquitetura, apontadas por FERREIRA (2003), são:

- Os mecanismos de controle de integridade dos dados espaciais não são totalmente definidos;
- As extensões da linguagem SQL ainda não são padronizadas.

Segundo LORENTZOS e VIQUEIRA (2006), os SGBD de última geração incluem as extensões de seus modelos, que permitem o armazenamento e gerenciamento de dados espaciais, bem como os SIG de última geração permitem o armazenamento de dados geográficos em extensões espaciais do SGBD.

Vale ressaltar que na arquitetura integrada, o SGBD a ser utilizado é informado ao SIG no momento da conexão com o mesmo. A partir daí, o Módulo de Tratamento de Dados Espaciais tem a capacidade de interagir com o SGBD em todas as necessidades de manipulação. O Módulo de Tratamento de Dados Espaciais exerce o papel de cliente e o SGBD do servidor. O SGBD e os SIG devem se comunicar, sendo necessário para tal, que ambos sigam o mesmo padrão de armazenamento de dados.

Os SIG envolvidos nessa pesquisa são o gvSIG, o OpenJUMP e o Quantum GIS.

4.2.2 – SIG gvSIG

O gvSIG é um sistema de informações geográficas, de código aberto com licença GPL, desenvolvido na linguagem de programação Java e está disponível para as plataformas Linux, Windows e Mac.

O projeto gvSIG teve início em 2004 como parte do projeto de migração de todos os sistemas de informação do Ministério da Infraestrutura e Transporte de Valência (Espanha) para programas computacionais livres (GVSIG, 2011a). Posteriormente três empresas espanholas e uma venezuelana criaram a *Asociación gvSIG (Asociación para la promoción de la Goemática libre y el desarrollo de gvSIG)*, uma associação sem fins lucrativos que tem como objetivo principal garantir a sustentabilidade do projeto gvSIG. Parte do lucro gerado através dos negócios da Associação é revertido para o projeto gvSIG (VIDAL *et. al*, 2009).

O gvSIG trabalha com a maior parte dos formatos padrão de dados *raster* e vetoriais. Entre os formatos de dados vetoriais estão: shp, gml, klm, dxf, dwg, dgn, WFS, WFS-T, ArcIMS, PostgreSQL 2D, MySQL, Oracle Spatial/Locator, ArcSDE. As possibilidades de acesso a dados do gvSIG podem ser encontradas em <<http://www.gvsig.org/web/projects/gvsig-desktop/devel/gvsig/1-10-0/formats/>> (GVSIG, 2011a).

As atividades executadas no gvSIG fazem parte de um projeto, que pode ser composto por documentos dos tipos: vista, tabela e mapa, descritos a seguir (REOLON, 2008):

- Vista: é o local destinado a visualização, manipulação, análise e edição das camadas de informação geográficas;
- Tabela: é o local destinado a criar, editar e visualizar atributos alfanuméricos das camadas de informações;

- Mapa: é o local destinado ao preparo do *layout* das saídas gráficas e descritivas.

Os projetos são salvos em arquivos com extensão <gvp> em local escolhido pelo usuário. O arquivo de projetos armazena as referências sobre a localização das fontes de dados utilizados (GVSIG, 2011a).

O gvSIG integra-se à biblioteca de algoritmos de análises geográficas SEXTANTE (*Sistema EXTremeno de ANálisis Territorial*), cujo desenvolvimento foi uma iniciativa do governo local de Extremadura, Espanha. Essa biblioteca disponibiliza mais de 300 algoritmos para dados *raster* e vetoriais. Além do gvSIG, outros programas livres de SIG, tais como OpenJUMP e uDIG integram-se à biblioteca. SEXTANTE é escrita em Java, distribuído sob licença MIT (OLAYA, 2011). A licença MIT, criada pelo *Massachusetts Institute of Technology*, redigida para o sistema Windows X, afirma que quem obtém uma cópia do programa computacional pode lidar com ele sem restrições (SABINO e KON, 2009).

Ressalta-se que o projeto gvSIG além do SIG *desktop*, tem o produto gvSIG *Mobile* para dispositivos móveis, ideal para captura e atualização de dados no campo (GVSIG, 2011b).

4.2.3 – SIG OpenJUMP

OpenJUMP é um Sistema de Informações Geográficas de código fonte aberto que teve sua origem no JUMP Workbench desenvolvido pela *Vivid Solutions Inc.* e pela *Refraction Research, Inc., British Columbia, Canadá*. Seu nome é uma abreviação de *Unified Mapping Platform*, onde o J indica a linguagem de programação Java e o Open o código fonte aberto. É um produto distribuído sob licença GPL e trabalha nas plataformas Windows, Linux e Mac (OPENJUMP, 2010b).

O OpenJUMP faz parte do *JUMP Pilot Project (JPP)*, uma organização cuja principal função é coordenar e incentivar o desenvolvimento cooperativo de

tecnologias baseadas no OpenJUMP. Ela possibilita o trabalho em conjunto de desenvolvedores de várias partes do mundo, de forma a evitar duplicação de esforços (OPENJUMP, 2010a).

O programa fornece nativamente suporte aos arquivos de dados vetoriais nos formatos: SHP, GML, JML, WKT e FME e, devido aos *plugins* desenvolvidos, trabalha também com arquivos em formato *raster*, com SGBD e com outros formatos vetoriais. Quanto a integração com os SGBD PostgreSQL/PostGIS e MySQL, o OpenJUMP lê as camadas de informações armazenadas nos dois SGBD, mas grava apenas no PostgreSQL/PostGIS (OPENJUMP, 2010b).

OBE e HSU (2011) apresentam como características relevantes do OpenJUMP, a quantidade de funcionalidades para análise estatística, processamento de geometria implementadas e também a possibilidade de efetuar consultas espaciais através de digitação de comando no PostGIS. As limitações apresentadas são em relação à leitura de arquivos muito grandes e ao suporte a projeções cartográficas (OPENJUMP, 2010a).

Quando é iniciado, o OpenJUMP abre uma janela para gerenciar uma tarefa ou projeto que é basicamente um conjunto de camadas de informações que podem ser exibidas juntas. Os projetos são salvos em arquivos com extensão <jmp> ou <jcs> em local escolhido pelo usuário. O arquivo de projetos armazena as referências sobre a localização das fontes de dados utilizados e as características das camadas tais como: cores, preenchimento, simbologia, entre outros (OPENJUMP, 2010b).

Segundo HSU e OBE (2008), toda a família de produtos JUMP tem como base central de sua funcionalidade a biblioteca JTS (*Java Topology Suite*). JTS é uma biblioteca para análises espaciais sobre geometrias em 2D, que segue a especificação SFS do OGC. É escrita em Java, de código aberto, e distribuída sob licença LGPL. LGPL (*Lesser GPL*) é uma licença para bibliotecas que permite que a mesma seja utilizada por programas livres e proprietários (GNU, 2011).

4.2.4 – Quantum GIS

O Quantum GIS (QGIS) é um SIG de código aberto distribuído sob licença GPL, escrito em C++ e que executa em Linux, Unix, Mac e Windows. Seu desenvolvimento foi iniciado em 2002, por Gary Sherman, com a intenção de desenvolver um visualizador de dados espaciais para Linux que fosse rápido e desse suporte a vários formatos de dados. Atualmente, existe a comunidade Quantum GIS que conduz o projeto. Parte do material do QGIS está hospedado no sítio da *Open Source Geospatial Foundation* (OSGeo) (QGIS, 2011).

OBE e HSU (2011) apresentam como características relevantes do QGIS, a facilidade de editar as geometrias no PostGIS, a facilidade de utilização, a *interface* amigável com outras ferramentas e a falta da funcionalidade para salvar uma camada de informações como uma nova tabela.

Entre as funcionalidades oferecidas pelo QGIS estão: Visualizar, criar, analisar, editar, gerenciar, exportar e explorar dados, bem como compor mapas e publicar mapas. A arquitetura do QGIS permite também ampliar suas funcionalidades através de *plugins*.

Os projetos QGIS são salvos em arquivos no formato XML e podem ser editados por quem conheça a linguagem. As informações armazenadas no projeto incluem as camadas adicionadas, as propriedades dessas camadas e a projeção para visualização do mapa (QGIS, 2010).

Para ampliar os formatos de dados que trabalha, o QGIS tem um *plugin* para a biblioteca GDAL / OGR para leitura e gravação dos dados espaciais (QGIS, 2010). GDAL / OGR é uma biblioteca de código aberto, sob licença MIT escrita em C++. GDAL (*Geospatial Data Abstraction Library*) é tradutora de dados espaciais de formatos *raster* e *OGR Simple Feature Library* é tradutora de dados espaciais de formato vetorial. A OGR é parte da GDAL. Ambas oferecem ainda utilitários que são executados na linha de comandos (GDAL,2011).

5. METODOLOGIA DA PESQUISA

A metodologia dessa pesquisa segue as etapas:

- Definição do processo de análise;
- Seleção dos SGBD e SIG;
- Definição dos recursos tecnológicos;
- Comparação dos SGBD e Análise das integrações.

Apoiada na recomendação do governo brasileiro sobre adoção de programas computacionais de código aberto nas soluções de Tecnologia da Informação (TI) e na idéia de que fortalecer esse conhecimento no ambiente acadêmico é uma forma de disseminá-lo, apenas SGBD e SIG enquadrados nessa categoria foram incluídos na pesquisa.

Com a finalidade de padronizar o processo de análise, um conjunto de dados foi utilizado para efetuar todas as operações necessárias, bem como as referidas operações foram executadas igualmente nas ferramentas do mesmo tipo.

5.1 Definição do processo de análise

O processo de análise foi dividido em quatro fases.

Na primeira fase são efetuadas cargas diretas dos arquivos em formato *shape* nos SGBD, sem a participação dos SIG. Essa carga visa oferecer uma estrutura de tabelas associadas à geometria, em formato padrão, que os SIG envolvidos na pesquisa devem ser capazes de trabalhar. Nessa fase é também investigada a possibilidade de transferência de dados espaciais entre os SGBD. A Figura 9 apresenta o esquema de carga no banco de dados e a Figura 10 o esquema da transferência de dados entre os SGBD.

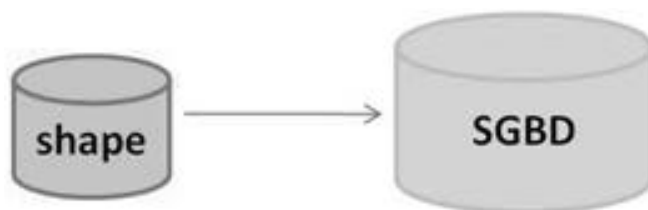


Figura 9 - Esquema de carga dos arquivos no SGBD

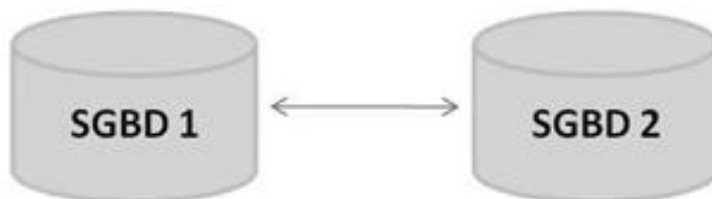


Figura 10 - Esquema de transferência de dados entre SGBD

Na segunda fase, os SIG selecionados são analisados quanto à sua integração com cada SGBD. Primeiramente, como retratado na Figura 11, é testado o acesso dos SIG às tabelas associadas à geometria que foram carregadas na primeira fase. Em seguida os SIG são investigados quanto à capacidade de alteração das tabelas associadas à geometria em relação às estruturas de dados e aos dados armazenados, como mostra a Figura 12.

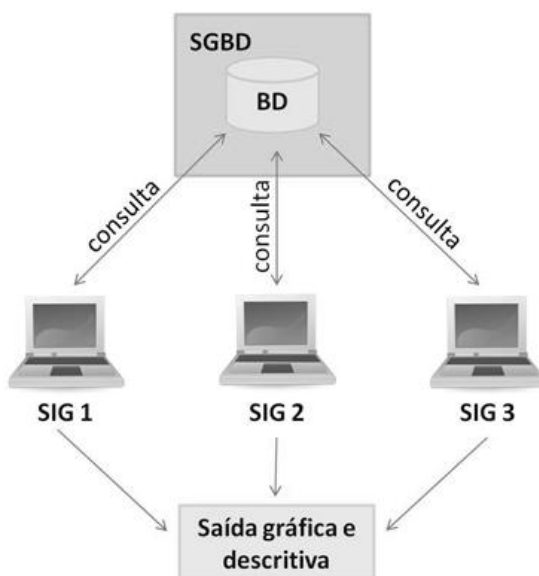


Figura 11 - Acesso dos SIG às tabelas associadas à geometria

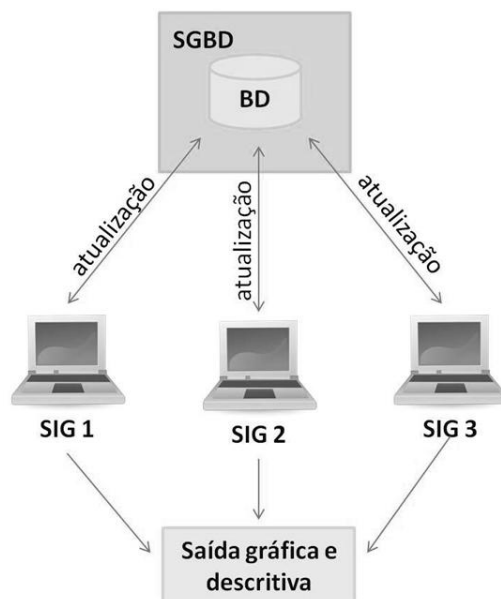


Figura 12 - Alteração dos bancos de dados pelos SIG

Na terceira fase, os SIG são analisados em relação ao compartilhamento dos dados armazenados nos SGBD. É verificada a capacidade de cada SIG acessar as tabelas associadas à geometria, criadas a partir dos demais SIG. A Figura 13 apresenta essa etapa.

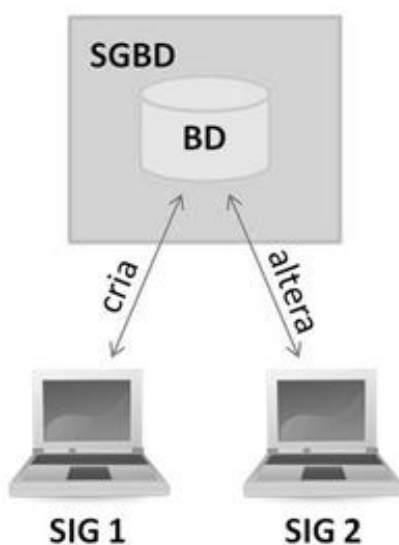


Figura 13 - Compartilhamento de dados pelos SIG

Na quarta fase, verifica-se a possibilidade de um SIG acessar tabelas associadas à geometria oriundas de mais de um SGBD, como também, a possibilidade de importação de projetos entre SIG. A Figura 14 apresenta o

esquema de compartilhamento de dados através do acesso simultâneo a mais de um SGBD.

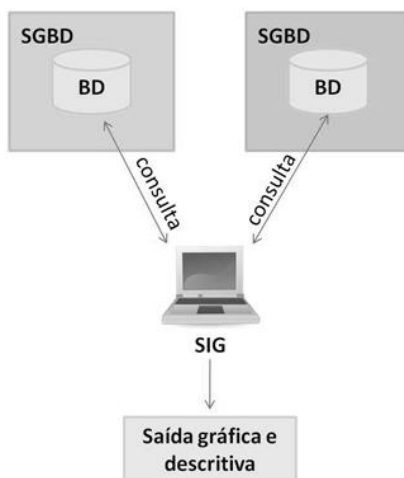


Figura 14 - Compartilhamento de dados com acesso simultâneo a mais de um SGBD

5.2 – Seleção dos SGBD e SIG

Foram selecionados para a análise os SGBD MySQL e PostgreSQL com a extensão espacial PostGIS, porque são de código aberto e têm suporte ao tratamento para dados espaciais. Estes SGBD têm o uso consolidado no gerenciamento de dados alfanuméricos.

A seleção dos SIG levou em consideração o tempo de maturação do programa, a qualidade do suporte disponível, a possibilidade de ser suportado por mais de um Sistema Operacional, o nível de complexidade para utilização e a velocidade de processamento (CHEN *et al.*, 2010; SVEEN, 2008). Com base nessas considerações selecionou-se os programas gvSIG, OpenJUMP e Quantum GIS (QGIS) para análise.

5.3 – Definição dos recursos tecnológicos

5.3.1 - Equipamentos computacionais

Os equipamentos utilizados na pesquisa têm a seguinte configuração:

- Notebook – Amazon - Intel (R) Core(TM)2Duo – CPU T5550 - 1.83GHz – 2 GB de RAM – HD 110 GB;
- Netbook – ASUS – Modelo Eee PC – Intel (R) Atom (TM) CPU N270 1.6GHz – 2 GB de RAM – HD 140 GB;
- Impressora jato de tinta.

5.3.2 – Programas Computacionais

Os programas computacionais utilizados na pesquisa foram os seguintes:

- PostgreSQL 8.4.7
- PostGIS 1.5
- pgAdmin III 1.10.2
- MySQL 5.5.8
- MySQL Workbench 5.2
- gvSIG 1.9
- OpenJUMP 1.3.1
- Quantum Gis 1.6.0 Cipiapó
- shp2pgsql
- shp2mysql
- ogr2ogr
- Microsoft Windows XP Professional versão 2002 – sistema operacional
- Microsoft Windows 7 Starter – sistema operacional

5.3.3 – Dados espaciais

Os dados espaciais utilizados foram fornecidos pelo Departamento de Engenharia Cartográfica da UFPE. Trata-se de um conjunto de três arquivos em formato *shape*. O primeiro arquivo, com geometria em forma de linha, corresponde às rodovias do estado de Pernambuco. O segundo arquivo, com geometria em forma de polígono, corresponde aos municípios do estado de Pernambuco, e o terceiro arquivo, com geometria em forma de pontos, corresponde aos nós das rodovias do estado de Pernambuco. Esse conjunto de dados contempla os três tipos básicos de geometria.

6. COMPARAÇÃO DOS SGBD E ANÁLISE DAS INTEGRAÇÕES

6.1 – Comparação das estruturas de dados dos SGBD

Na comparação das extensões espaciais entre os SGBD, foram considerados os aspectos: estrutura de dados, funções espaciais e recursos de indexação.

Na especificação SFS, existe apenas o tipo de dados *GEOMETRY*, sendo os tipos *POINT*, *LINESTRING*, etc, especializações desse tipo. O PostGIS segue esse padrão. O MySQL implementa cada um desses tipos separadamente.

Os utilitários *shp2mysql* e *ogr2ogr*, utilizados para a carga das tabelas criaram apenas colunas geométricas do tipo *GEOMETRY*. Para verificar o comportamento dos SIG com os demais tipos de dados do MySQL, foi incluída e preenchida manualmente uma coluna do tipo *LINESTRING* em uma tabela. Em seguida, a mesma foi acessada pelo gvSIG e pelo QGIS. Apenas o gvSIG reconheceu a coluna como geométrica e apresentou a camada. O QGIS tratou a coluna como um atributo de tipo desconhecido.

Ainda em relação às estruturas de dados, o PostGIS implementa as tabelas de metadados especificadas e o MySQL não. Isso pode causar problemas para os SIG que usam essas tabelas como apoio, bem como pode comprometer a possibilidade de conversão entre sistemas de referência.

Quanto as funções espaciais, no MySQL elas são parcialmente implementadas e ainda assim algumas não seguem a especificação SFS. O PostGIS não apresenta esse problema.

Em relação a indexação espacial, o MySQL disponibiliza o recurso apenas para tabelas do tipo MyISAM, que não oferece funcionalidades de controle de transação, o que pode colocar em risco a integridade dos dados.

O PostGIS tem uma implementação de tratamento espacial que cobre grande parte das recomendações OGC. Além disso, oferece o tipo de dados *GEOGRAPHY*,

que tende a se desenvolver e traz a possibilidade de efetuar cálculos sobre esferóides.

6.2 – Análise das integrações

Para execução dessa análise, preparou-se o ambiente seguindo a arquitetura cliente-servidor, onde o cliente recebeu os programas de SIG e o servidor os SGBD, conforme mostrado na Figura 15.

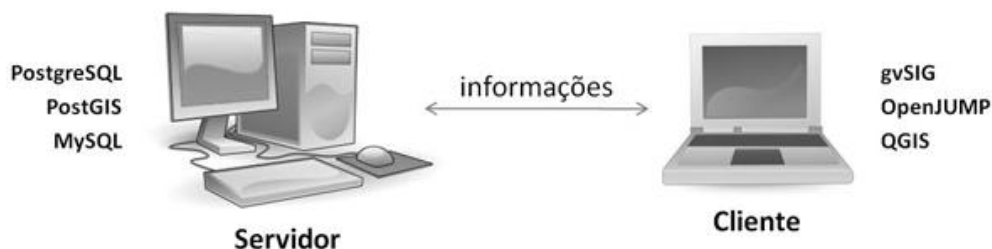


Figura 15 - Ambiente cliente-servidor utilizado

Os pontos que demandaram maior esforço na montagem do ambiente foram: (i) a instalação dos plugins quando estes não eram instalados junto com o programa. (ii) estabelecer a conexão entre o SGBD e os programas de SIG em máquinas distintas, prejudicada principalmente pela qualidade da rede utilizada que algumas vezes levou a achar que os problemas de rede eram de configuração.

A conexão entre o SIG e o SGBD é estabelecida através da *interface* do SIG, na qual são passados os seguintes parâmetros: o SGBD, o nome da base de dados, o endereço da máquina (IP) onde o SGBD se encontra, a porta de acesso utilizada pelo SGBD, o usuário e a senha. O PostgreSQL/PostGIS utiliza a porta 5432 e o MySQL a 3306. Caso o SGBD esteja instalado na mesma máquina que o cliente, o endereço IP pode ser substituído pelo termo *localhost*. A conexão fica cadastrada no SIG, de forma que pode ser utilizada sempre que o SIG acessar o mesmo banco de dados. A Figura 16 exemplifica a criação de uma conexão entre o OpenJUMP e o PostGIS.

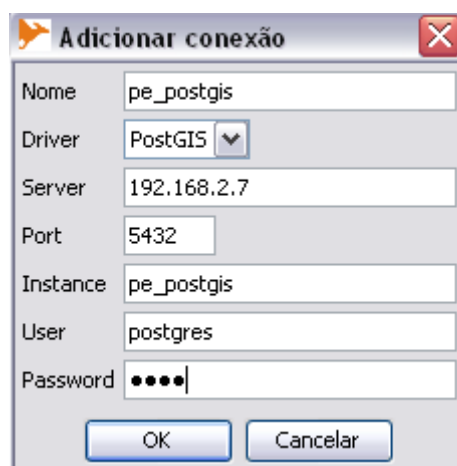


Figura 16 - Criação de conexão entre o OpenJUMP e o PostGIS

O endereço IP pode variar quando se trabalha com IP dinâmico, onde cada vez que um computador é ligado, recebe um número IP que representa o endereço do computador na rede durante aquela conexão. Para trabalhar de forma permanente com acesso remoto na arquitetura cliente-servidor, é necessário que a máquina servidora seja configurada para ter um endereço IP fixo. Dessa forma, as conexões estabelecidas e configuradas nas diversas aplicações não precisam ser alteradas a cada vez que a máquina servidora é reiniciada. No presente caso, optou-se por continuar trabalhando com o IP dinâmico uma vez que o estudo e a configuração de redes não é o foco da pesquisa.

Quanto às configurações para permitir o acesso remoto aos SGBD, foram necessárias duas ações. A primeira ação foi configurar a máquina onde estão os SGBD para permitir o acesso através das portas que os SGBD utilizam. Foram liberadas as portas 5432 e 3306 da máquina para permitir acesso ao PostgreSQL e MySQL respectivamente. A segunda ação foi configurar os SGBD para o acesso remoto, cujos passos estão descritos a seguir:

No PostgreSQL, dois arquivos de configuração foram alterados:

- No arquivo postgresql localizado em C:\Arquivos de programas\PostgreSQL\8.4\data, no bloco chamado # CONNECTIONS AND AUTHENTICATION a linha com o parâmetro listen_addresses foi customizada para listen_addresses = '*'.

- No arquivo pg_hba, localizado em C:\Arquivos de programas\PostgreSQL\8.4\data, no bloco # IPv4 local connections, a linha host all all 192.168.2.0/0 md5, foi incluída para permitir que qualquer máquina que faça parte da rede 192.168.2 possa acessar ao SGBD PostgreSQL. As configurações anteriores do SGBD continuaram válidas.

Na configuração do MySQL foi criado um usuário com permissão de acesso ao SGBD a partir de qualquer endereço IP. A esse usuário foram concedidos privilégios totais sobre o banco de dados através do comando:

```
Grant all on *.* to mysqluser@'%' identified by 'senha1';
```

6.2.1 - Carga e transferência de dados nos bancos de dados

O SGBD fornece, através da linguagem SQL, um conjunto de comandos para a manutenção das estruturas das tabelas e a manipulação dos dados nelas contidos. Por manutenção das estruturas entende-se criar, alterar e remover as tabelas do banco de dados e por manipulação dos dados entende-se incluir, alterar e excluir dados dessas tabelas. Entretanto, a manutenção e a manipulação das tabelas através dos comandos SQL exigem familiaridade com a linguagem, o que não é comum entre os usuários. Uma forma amigável de executar essas ações, quando dados espaciais estão envolvidos, é através dos programas de SIG ou de utilitários desenvolvidos para esse fim. Essa última opção foi escolhida, observando-se o seguinte:

A carga dos dados no PostGIS foi efetuada empregando-se o utilitário *shp2pgsql*, executado através de um *plugin* do pgAdmin, conforme mostra a Figura 17, não sendo detectada perda de informações tabulares e geométricas. Ressalta-se que durante sua operação, acontece também a criação da tabela associada à geometria. A tabela de metadados *GEOMETRY_COLUMNS* foi atualizada corretamente.

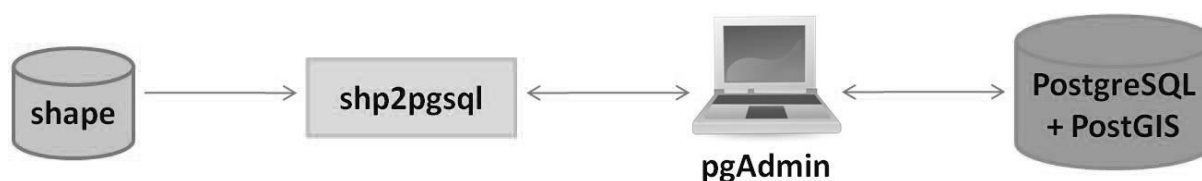


Figura 17 - Carga das tabelas no PostGIS

A carga dos dados no MySQL, esquematizada na Figura 18, foi efetuada empregando-se o utilitário `shp2mysql`, não sendo detectada perda de informações tabulares e geométricas. A criação da tabela associada à geometria também acontece durante sua operação. Apesar de não haver perda de informações, as colunas alfanuméricas foram criadas com um tamanho bem maior do que suas correspondentes no PostGIS e as colunas geométricas foram criadas com o tipo de dado *GEOMETRY*. A execução do `shp2mysql` gerou arquivos com comandos SQL para criar e carregar as tabelas associadas à geometria no MySQL. Então esse arquivo foi executado através do MySQL Workbench.

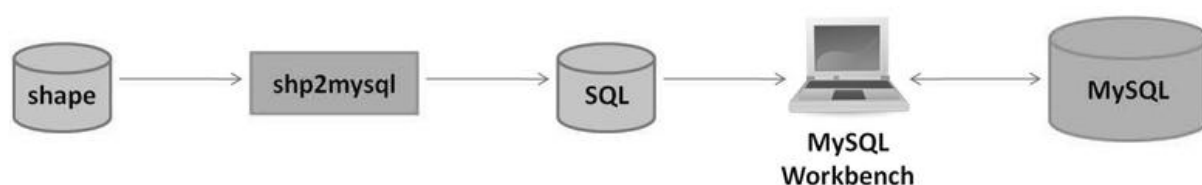


Figura 18 - Carga das tabelas do MySQL

Embora o MySQL disponibilize os tipos de dados *GEOMETRY*, *POINT*, *LINESTRING* e *POLYGON*, todas as tabelas foram criadas com a coluna geométrica do tipo *GEOMETRY*.

6.2.2 – Integração SGBD-SGBD

Na integração entre os SGBD buscou-se efetuar a transferência de dados espaciais entre eles. Foi empregado o utilitário `ogr2ogr`, como ilustra a Figura 19, ressaltando-se que a tabela de destino foi criada na mesma operação da carga. Na transferência MySQL – PostGIS não foi detectada perda de informações tabulares e

geométricas. É preciso chamar a atenção para o fato de que a tabela de destino foi criada de acordo com a tabela de origem, herdando o mesmo tipo de geometria e o tamanho das colunas alfabéticas que eram maiores que o necessário. Devido a inexistência da tabela de metadados *GEOMETRY_COLUMNS* no MySQL, a referida tabela foi atualizada no PostGIS com as seguintes características: (i) a coluna SRID (sistema de referência) das tabelas copiadas ficou como indefinido (valor -1); (ii) a coluna TYPE no PostGIS foi preenchida com “*GEOMETRY*” e não com o tipo de geometria armazenado na tabela. Foi ainda identificado que a restrição de integridade que garante que todas as linhas da tabela terão o mesmo tipo de geometria, não foi criada.

Na transferência PostGIS – MySQL o ogr2ogr não criou os atributos alfanuméricos das tabelas. Foi feita uma segunda tentativa, passando o nome das colunas alfanuméricas como parâmetro na execução do utilitário, mas também não funcionou. Então, a solução de contorno utilizada foi criar a tabela de destino utilizando comandos SQL e em seguida, carregar os dados através do mesmo utilitário.



Figura 19 - Transfêrencia de dados entre SGBD

A execução dos utilitários shp2pgsql, shp2mysql e ogr2ogr requer que sejam passados parâmetros como: a fonte e o destino dos dados, o sistema de referência, indicação para criar índice espacial, entre outras. Esses parâmetros podem ser obtidos digitando na linha de comando apenas o nome do utilitário. Informações mais detalhadas podem ser encontradas em <http://postgis.refrations.net/docs/ch04.html#shp2pgsql_usage> para o shp2pgsql e em <<http://www.gdal.org/ogr2ogr.html>> para o ogr2ogr.

6.2.3 - Integração SIG - SGBD

6.2.3.1 - SIG e o PostGIS

Os três SIG acessaram sem problemas as tabelas associadas à geometria contidas no PostGIS, criadas e carregadas pelo utilitário shp2pgsql. A Figura 20 apresenta a visualização no QGIS, de três camadas com dados oriundos dessas tabelas e parte das tabelas rodovias e pe. Nos demais SIG, o resultado é semelhante.

Em seguida foi investigada a capacidade dos SIG alterarem o banco de dados PostGIS em relação as estruturas de dados e ao conteúdo dos dados armazenados. Os resultados são apresentados no Quadro 2.

Quadro 2 - Mapeamento das operações realizadas pelos SIG sobre o PostGIS

Ação	gvSIG	OpenJUMP	QGIS
Gravar camada do SIG no banco de dados	Sim	Sim	Não (*)
Criar coluna alfanumérica	Sim	Não	Sim
Criar coluna geométrica	Não	Não	Não
Alterar a definição de coluna alfanumérica	Não (**)	Não	Não
Alterar a definição de coluna geométrica	Não	Não	Não
Remover coluna alfanumérica	Sim	Não	Sim
Remover coluna geométrica	Não	Não	Não
Remover tabela	Não	Não	Não
Incluir linha	Sim	Sim	Sim
Remover linha	Sim	Sim	Sim
Alterar conteúdo de coluna alfanumérica	Sim	Sim	Sim
Alterar conteúdo de coluna geométrica	Sim	Sim	Sim

(*) Uma camada de informações no QGIS não gera uma nova tabela associada à geometria. Para criar uma tabela associada à geometria no banco de dados, é necessário utilizar a biblioteca OGR que é disponibilizada através de *plugin*.

(**) o gvSIG permite alterar apenas o nome da coluna.

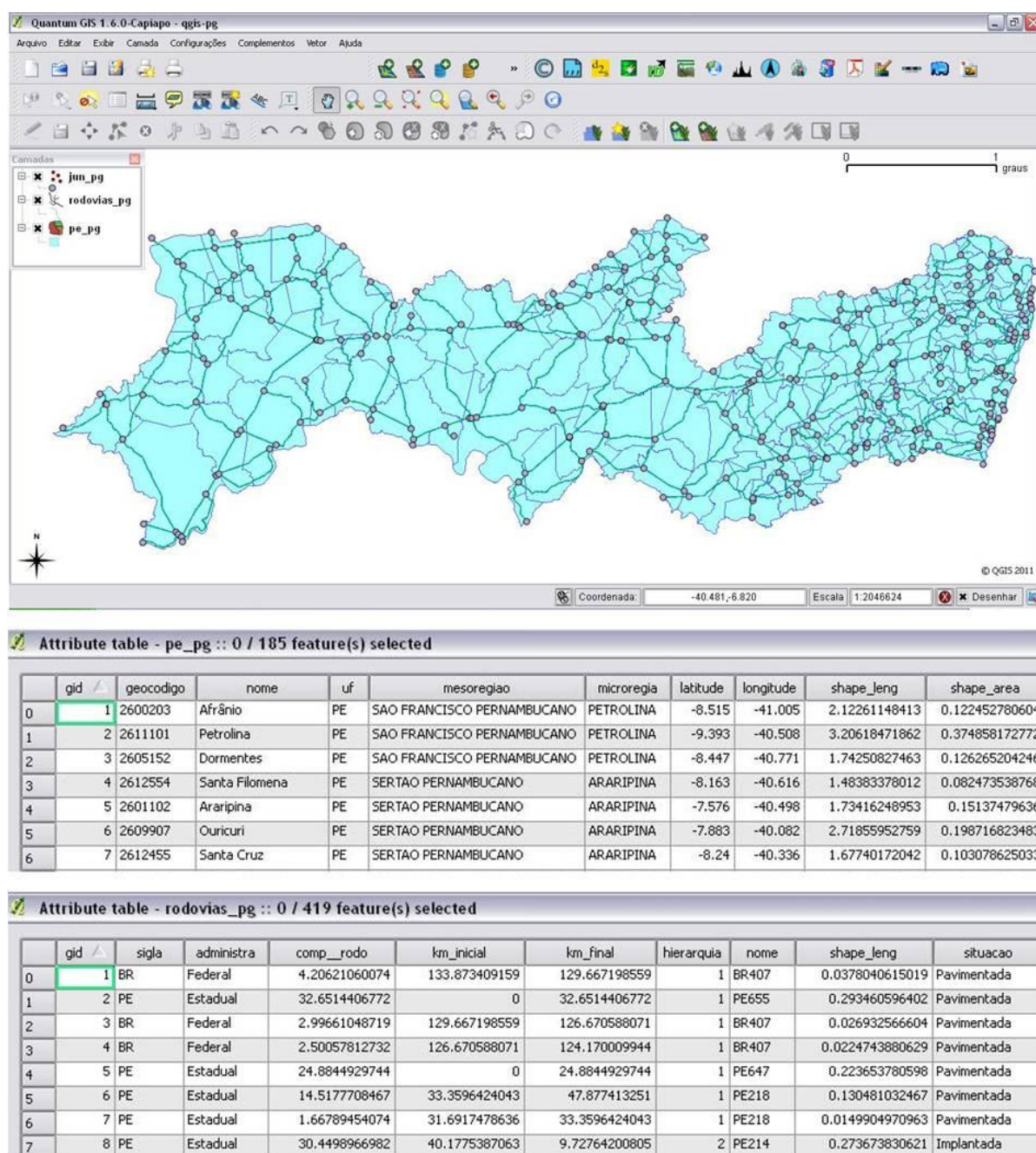


Figura 20 – Tabelas armazenadas no PostGIS e visualizadas no QGIS

6.2.3.2 - SIG e o MySQL

Não foi obtido sucesso nas tentativas de acesso feitas através do OpenJUMP. O comportamento apresentado corresponde ao descrito na documentação do *plugin* para os casos de consulta a dados com geometrias nulas ou consultas envolvendo

apenas atributos alfanuméricos. As mesmas tabelas foram acessadas pelo gcSIG e pelo QGIS.

O gvSIG e o qGIS acessaram sem problemas as tabelas associadas à geometria contidas no MySQL, criadas por comandos SQL, e carregadas pelo utilitário shp2mysql. A Figura 21 apresenta a visualização no QGIS de três camadas com dados oriundos dessas tabelas. No gvSIG o resultado é equivalente.

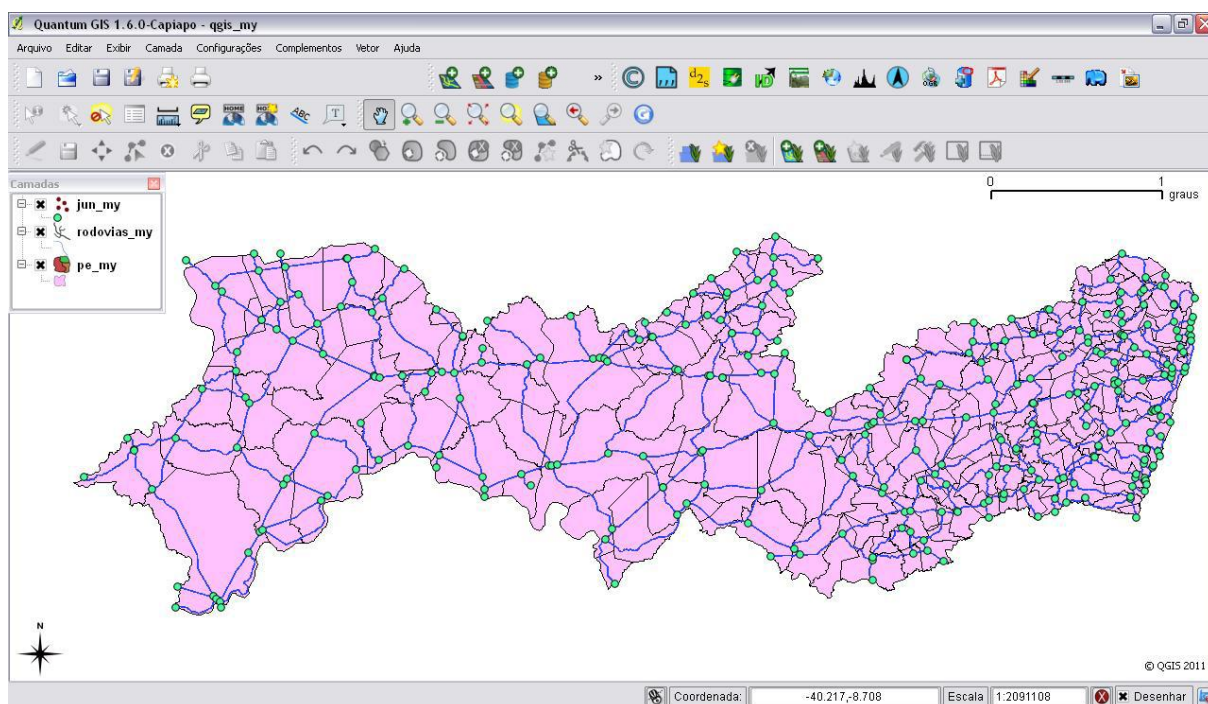


Figura 21 - Geometrias visualizadas no QGIS armazenadas no MySQL

Em seguida foi investigada a capacidade do SIG alterar o banco de dados MySQL, em relação as estruturas de dados e ao conteúdo dos dados armazenados. Os resultados são apresentados no Quadro 3.

Quadro 3 - Mapeamento das operações realizadas pelos SIG sobre o MySQL

Ação	gvSIG	QGIS
Gravar camada do SIG no banco de dados	Não	Não (*)
Criar coluna alfanumérica	Não	Não
Criar coluna geométrica	Não	Não
Alterar a definição de coluna alfanumérica	Não	Não
Alterar a definição de coluna geométrica	Não	Não
Remover coluna alfanumérica	Não	Não
Remover coluna geométrica	Não	Não
Remover tabela	Não	Não
Incluir linha	Sim (**)	Sim
Remover linha	Sim (**)	Sim
Alterar conteúdo de coluna alfanumérica	Sim (**)	Sim
Alterar conteúdo de coluna geométrica	Sim (**)	Sim

(*) Uma camada de informações no QGIS não gera uma nova tabela associada à geometria. Para criar uma tabela associada à geometria no banco de dados, é necessário utilizar a biblioteca OGR que é disponibilizada através de *plugin*. A tentativa de utilizar o *plugin* resultou em erro.

(**) No gvSIG é possível fazer manipulação dos dados, entretanto muitas vezes aparece mensagens de erro, apesar da operação se concretizar.

6.2.4 - Compartilhamento de dados pelos SIG

Na análise com o PostGIS, os três SIG envolvidos na pesquisa acessaram sem problemas as tabelas associadas à geometria que foram criadas através de outro SIG. No MySQL, esse teste não pode ser executado, uma vez que os SIG não criaram tabelas nesse SGBD. A Figura 22 ilustra o compartilhamento de dados no PostGIS, onde o nome da camada indica o SIG que gerou a tabela associada à geometria no banco de dados.

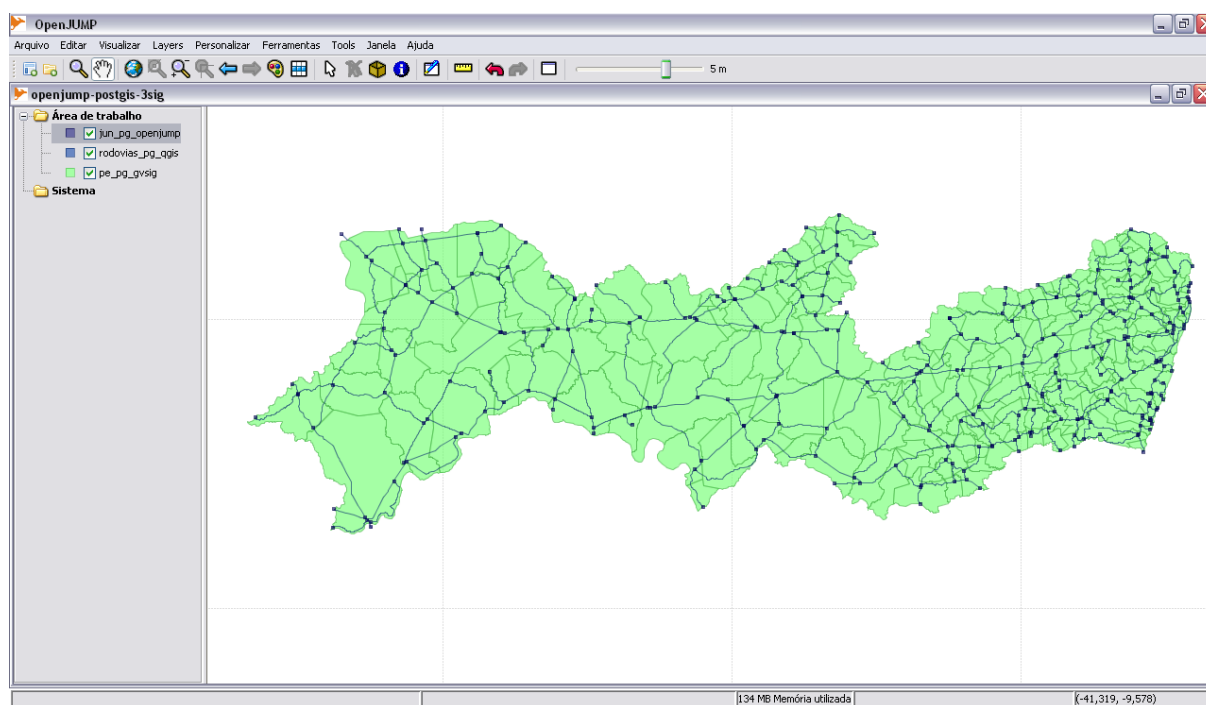


Figura 22 - Camadas criadas no OpenJUMP usando tabelas associadas à geometria do PostGIS criadas pelos SIG

6.2.5 - Compartilhamento de dados entre SIG com acesso simultâneo a mais de um SGBD

O gvSIG e o Quantum GIS apresentaram essa possibilidade. O OpenJUMP não obteve sucesso no acesso as tabelas do MySQL. A Figura 23 apresenta uma visualização do gvSIG, onde as camadas foram criadas a partir de tabelas associadas à geometria oriundas de mais de um SGBD. A fonte das camadas `pe_pg` e `jun_pg` é o PostGIS e da camada `rodovias_my` é o MySQL.

Quanto ao acesso de projetos entre os SIG, até o momento estes não são compartilhados entre os SIG envolvidos na pesquisa. Vale salientar que a forma de armazenamento dos projetos dos SIG não faz parte da especificação SFS, que trata apenas do armazenamento e manuseio dos dados espaciais através de SQL.

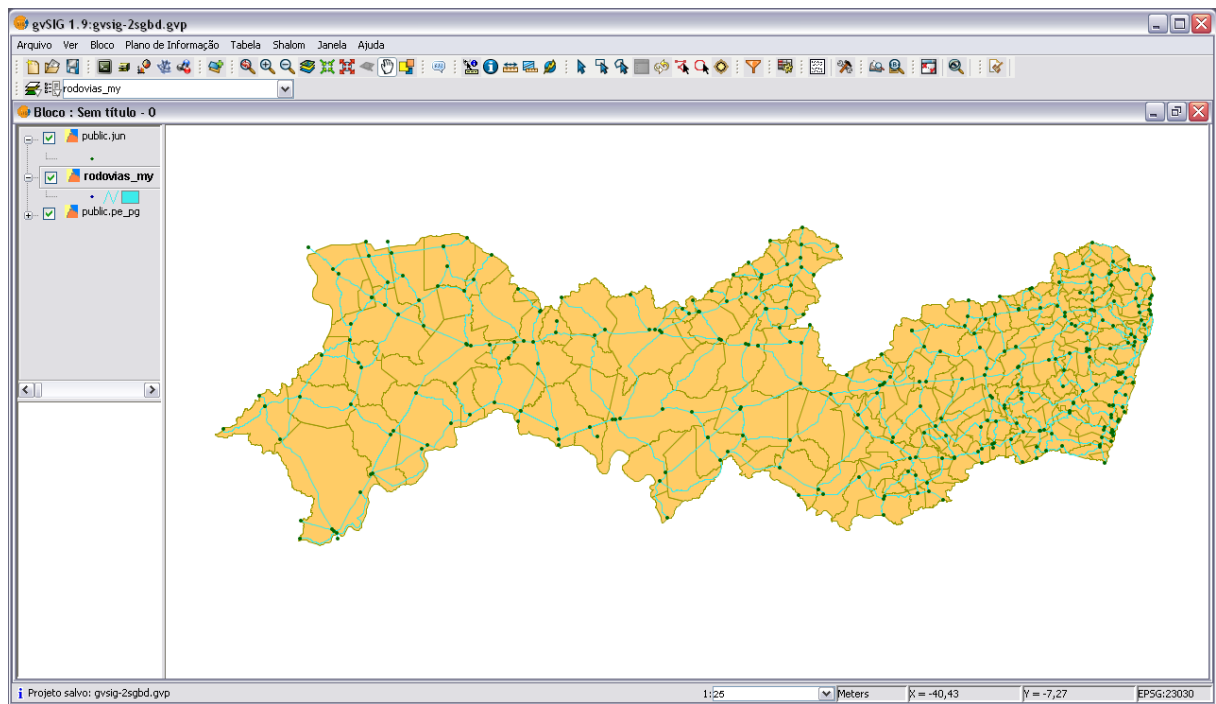


Figura 23 - Compartilhamento de dados entre SIG através do SGBD

7. CONCLUSÕES E RECOMENDAÇÕES

Este trabalho apresentou uma análise da integração entre Sistemas de Informações Geográficas (SIG) e Sistemas Gerenciadores de Bancos de Dados (SGBD) com extensões espaciais, em relação à interoperabilidade de dados geográficos vetoriais. Duas premissas foram definidas para a escolha dos programas: que tenham código fonte aberto e que os SGBD respeitem, parcial ou integralmente, as especificações do OGC - *Open Geospatial Consortium*. Foram utilizados os SGBD MySQL e PostgreSQL/PostGIS e os SIG gvSIG, OpenJUMP e Quantum GIS, além de um conjunto de dados espaciais em formato *shapefile*, fornecido pelo Departamento de Engenharia Cartográfica da UFPE.

Os SIG têm interface amigável e seu uso é intuitivo, embora as funcionalidades de análises espaciais não tenham sido exploradas. Para os SGBD, mesmo com o apoio de interfaces gráficas, é necessário algum conhecimento de suas funcionalidades para sua utilização e administração. Vale ressaltar que a adoção do ambiente utilizado nesse trabalho requer o acompanhamento profissional para a administração do SGBD e da rede de computadores.

Comparando as extensões espaciais do MySQL e do PostGIS, identificou-se que a implementação do PostGIS está mais próxima da especificação SFS da OGC que a implementação do MySQL.

Na transferência de dados entre SGBD através do utilitário ogr2ogr, foram encontradas dificuldades contornáveis tanto na transferência MySQL-PostGIS quanto na PostGIS-MySQL. É importante salientar que a mobilidade entre SGBD não é usual. Em geral adota-se um SGBD e trabalha-se com vários tipos de aplicações compartilhando os mesmos dados. Lembramos que é importante definir as necessidades da aplicação para a escolha do SGBD, de forma que ele possa cumprir todas as etapas e atender todos os usuários envolvidos.

Foi explorada a integração de cada um dos SIG com os SGBD, inclusive quanto à capacidade de modificar as bases de dados. Percebeu-se que as soluções para o PostGIS estão mais consolidadas. Neste houve a possibilidade de efetuar

operações de manutenção de tabelas e manipulação de dados, enquanto no MySQL apenas a manipulação foi obtida.

Percebeu-se ainda que há facilidade em utilizar, paralelamente, mais de um SIG sobre as mesmas bases de dados.

Foi confirmado que os SIG podem compartilhar os dados armazenados nos SGBD e que podem usar simultaneamente dados oriundos de mais de um SGBD.

Constatou-se que o uso do SGBD com extensões espaciais dentro dos padrões estudados, insere as informações geográficas no contexto das aplicações corporativas, permitindo que sistemas de informações tradicionais utilizem dados espaciais.

Considerando os resultados obtidos nas análises e as extensões espaciais utilizadas, chegou-se a conclusão que quanto maior a aderência entre a extensão espacial e a especificação SFS da OGC, mais fácil se torna a integração entre o SGBD e os SIG.

Para que a interoperabilidade de dados espaciais aconteça entre os SIG *desktop*, na forma estudada nesse trabalho, os SIG também devem obedecer a especificação SFS no armazenamento dos dados espaciais no SGBD.

Em continuidade a esse trabalho, sugere-se que a pesquisa seja estendida para incluir SGBD e SIG proprietários, visto que, pode haver interesse em adotar uma solução híbrida, onde programas livres e proprietários compartilhem dados. Também é sugerido que as demais funcionalidades dos SIG estudados sejam exploradas. Por fim, sugere-se complementar o estudo dos SIG, investigando a interoperabilidade através do uso WFS, WMS e WCS, serviços de internet especificados pelo OGC.

Finalmente, recomenda-se que na escolha de SIG e SGBD espaciais sejam considerados os pontos relacionados a seguir:

- O SGBD siga a especificação SFS da OGC;

- Os SIG sigam os padrões de interoperabilidade do OGC;
- Os SIG sejam multiplataforma;
- SGBD e SIG tenham o uso já consolidado;
- Os SIG e SGBD tenham a integração bem resolvida, com possibilidade de manutenção da base de dados através do SIG;
- SGBD e SIG disponham de uma estrutura de suporte que possa ser contratada ou consultada. Esse aspecto diz respeito mais diretamente aos programas computacionais livres;
- Que sejam consideradas as necessidades de conversão de dados na seleção dos SIG;
- Que sejam identificadas as funcionalidades necessárias aos SIG considerando que, uma vez garantido o compartilhamento de dados, pode ser adotado mais de um SIG.

REFERÊNCIAS BIBLIOGRÁFICAS

BRASIL - MINISTÉRIO DO PLANEJAMENTO, ORÇAMENTO E GESTÃO. **Guia de Interoperabilidade: Manual do Gestor**, 2011, 39 páginas. Documento técnico do governo brasileiro.

BRASIL. Lei Nº 10.257, de 10 de julho de 2001 - **Diário Oficial da União**. Brasília, DF, 11 jul. 2001. Disponível em: <http://www.planalto.gov.br/ccivil/leis/LEIS_2001/L10257.htm>. Acesso em: 16 jun. 2010.

CABRAL S., MURPHY K., **MySQL Administrator's Bible**. Wiley Publishing, Inc, 2009.

CÂMARA, G.; CASANOVA, MA.; HEMERLY, AS.; MAGALHÃES, GC.; MEDEIROS, I. **Anatomia de Sistemas de Informação Geográfica**. 1996. 205p.

CÂMARA, G.; QUEIROZ, GR. Arquitetura de Sistemas de Informação Geográfica. Capítulo 3 do livro Introdução à Ciência da Geoinformação. INPE, São José dos Campos, 2004.

CÂMARA, G.; ONSRUD, H. **Open Source GIS Software: Myths and Realities**. Open Access and the Public Domain in Digital Data and Information for Science, Washington, The National Academies Press, 2004.

CHEDE, CT. **Padrões abertos, interoperabilidade e interesse público**. poliTICs, número 2, novembro/2008, pag 29-35, Nupef/Rits. Disponível em: <http://www.politics.org.br/edicao_02/m4.html>. Acesso em: 04 ago. 2010.

CHEN, D.; SHAMS, S.; CARMONA-MORENO, C.; LEONE, A. **Assessment of Open Source GIS Software for Water Resources Management in Developing Countries**. Journal of Hydro-environment Research (2010), doi:10.1016/j.jher.2010.04.017

CAVALLINI, P. **Open Source and free GIS: A way out?** GIS development –The Geospatial Resource Portal, 2006. Disponível em: <http://www.gisdevelopment.net/technology/gis/techgis_004.htm>. Acesso em: 24 jun. 2010.

CAVALLINI, P.; PICCARDI, S. **Cosa significa Open Source? Cos'è il software libero? Applicazioni ai GIS**. MondoGIS – 59 – marzo/aprile 2007.

DAVIS JR., CA; BORGES, KAV.; SOUZA, LA.; CASANOVA, MA.; LIMA JR., PO. **O Open Geospatial Consortium**. Capítulo 11 do livro Bancos de Dados Geográficos. Editora MundoGEO. 2005.

DÜSTER, H. **Country Experiences: Cantão de Solothurn, Suíça** in FLOSS in Cadastre and Land Registration - Opportunities and Risks. Edited by Daniel Steudler, Mika-Petteri Töhrönen and Gertrude Pieper, April 2010. p 82-84.

FERREIRA, KR. **Interface para Operações Espaciais em Banco De Dados GEOGRÁFICOS**. Dissertação de Mestrado, INPE, 2003.

FSF – Free Software Foundation. Disponível em: <<http://www.fsf.org/>>. Acesso em: 24 jun. 2010.

GDAL. **GDAL - Geospatial Data Abstraction Library**. Disponível em: <<http://www.gdal.org/>>. Acesso em 02 Ago. 2011.

GNU. **Categories of free and non-free software**. Disponível em: <<http://www.gnu.org/philosophy/categories.html>>. Acesso em: 24 jun. 2010.

GNU. **Why you shouldn't use the Lesser GPL for your next library**. Atualizado em: Jul. 2011. Disponível em: <<http://www.gnu.org/philosophy/why-not-lgpl.html>>. Acesso em: Ago. 2011.

GORNI D.; GIANNOTTI M.; KNOPIK A.; BRITO P.; RODRIGUES M. **Open source web GIS Sistema de Informação Geográfica de Expedições**. Simpósio Brasileiro de Sensoriamento Remoto, Florianópolis, Brasil, 21-26 abril 2007, INPE. p. 3953-3960.

GUIA LIVRE. **Referência de Migração para Software Livre do Governo Federal / Organizado por Grupo de Trabalho Migração para Software Livre**. Brasília, 2005. 297 p.

GVSIG. **Manual de usuario gvSIG 1.9**. Última modificação 19/04/2011a. Disponível em <<http://www.gvsig.org/web/projects/gvsig-desktop/docs/user/gvsig-1-9-manual-de-usuario>>. Acesso em:10 jul. 2011.

GVSIG. **Qué es gvSIG mobile?** Última modificação 28/03/2011b. Disponível em <<http://www.gvsig.org/web/projects/gvsig-mobile/>>. Acesso em:10 jul. 2011.

HORISBERGER, J.; OSMANOVIĆ, K. **Country Experiences: Bósnia e Herzegovina** in FLOSS in Cadastre and Land Registration - Opportunities and Risks. Edited by Daniel Steudler, Mika-Petteri Töhrönen and Gertrude Pieper, April 2010. p 63-68.

HSU, L.; OBE, R. **OpenJump for PostGIS Spatial Ad-Hoc Queries**. Postgres OnLine Journal. 2008. Disponível em:

<<http://www.postgresonline.com/journal/archives/72-OpenJump-for-PostGIS-Spatial-Ad-Hoc-Queries.html>>. Acesso em: Jul. 2011.

KARLSSON, A. **GIS and Spatial Extensions with MySQL**. Disponível em: <<http://kambing.ui.ac.id/mysql/tech-resources/articles/4.1/gis-with-mysql.html>>. Acesso em: Ago. 2011.

LORENTZOS, NA.; VIQUEIRA, JRR. **Relational Formalism for the Management of Spatial Data**. The Computer Journal Vol. 49 No. 1, 2006

MACHADO, FNR.; ABREU M. **Projeto de Banco de Dados – Uma Visão Prática**. Editora Érica, 1996.

MARTINS, RG. **Proposta para o Banco de Dados do projeto WebMaps**. Dissertação de Mestrado, UNICAMP, Instituto de Computação. 2006.

MICROSOFT. **ODBC-Open Database Connectivity Overview**. Última revisão: 1 jul. 2010. Disponível em: <<http://support.microsoft.com/kb/110093/en-us?fr=1>>. Acesso em: 15 set. 2010.

MYSQL. **Commercial License for OEMs, ISVs and VARs**. Disponível em: <<http://www.mysql.com/about/legal/licensing/foss-exception/>>. Atualizado em: jul. 2010. Acesso em: 21 jul. 2011.

MYSQL. **MySQL 5.5 Reference Manual**. Disponível em: <<http://dev.mysql.com/doc/>>. Documento gerado em: 04 mai. 2011a (revisão: 25274).

MYSQL. **MySQL Workbench**. Disponível em: <downloads.mysql.com/docs/workbench-en.pdf>. Documento gerado em: 29 mar. 2011b (revisão: 25605).

NASSU, EA.; SETZER, VW. **Bancos de Dados Orientados a Objetos**. Editora Edgard Blucher LTDA, 1999.

OBE, RO; HSU, LS. **PostGIS in Action**. Manning Publications Co. 2011.

OGC - **OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture**06. Reference number: OGC 06-103r3. Version: 1.2.0., Editor: John R. Herring, 2006a.

OGC - **OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option**. Reference number: OGC 06-104r3. Version: 1.2.0., Editor: John R. Herring, 2006b.

OGC - OPEN GEOSPATIAL CONSORTIUM. **OGC Standards and Specifications**.
Atualizado em: 03/06/2010. Disponível em:
<<http://www.opengeospatial.org/standards>>. Acesso em: 20 jun. 2010.

OGC - OPEN GEOSPATIAL CONSORTIUM. **Compliance Testing**. Atualizado em:
25/07/2011a. Disponível em: <<http://www.opengeospatial.org/compliance>>.
Acesso em: 05 ago. 2011.

OGC - OPEN GEOSPATIAL CONSORTIUM. Disponível em:
<<http://www.opengeospatial.org/>>. Acesso em: 20 jul. 2011b.

OLAYA, V. **The SEXTANTE library**. Disponível em:
<<http://sextante.forge.osor.eu/index.html>>. Acesso em: 10 jul. 2011.

OPENJUMP. Atualizado em: Jul. 2010a. Disponível em
<<http://www.openjump.org/index.html>>. Acesso em: Jul. 2011

OPENJUMP. **New User Guide**. Atualizado em: Out. 2010b. Disponível em
<<http://sourceforge.net/apps/mediawiki/jump-pilot/index.php?title=Index>>. Acesso
em: Jul. 2011.

PIEPER, G. **Country Experiences: Camboja** in FLOSS in Cadastre and Land
Registration - Opportunities and Risks. Edited by Daniel Steudler, Mika-Petteri
Törhönen and Gertrude Pieper, April 2010. p 68-75.

QGIS - Quantum GIS Development Team. **Quantum User Guide version 1.6.0**
Copiado. 2010.

QGIS - Quantum GIS Development Team. **Quantum GIS Geographic Information
System**. Open Source Geospatial Foundation Project. 2011. Disponível em:
<<http://qgis.osgeo.org>>. Acesso em: 29 jul. 2011.

QUEIROZ, GR.; FERREIRA, KR. **SGBD com extensões espaciais**. Capítulo 8 do
livro Bancos de Dados Geográficos. Editora MundoGEO. 2005.

QUEIROZ, GR.; FERREIRA, KR. **Tutorial sobre Bancos de Dados Geográficos**.
Instituto Nacional de Pesquisas Espaciais. GeoBrasil, 2006.

PARKER, T. **Teach Yourself TCP/IP in 14 Days**. Second edition. Sams Publishing,
1996.

POSTGIS - **PostGIS 1.5.2 Manual**. Disponível em:
<<http://postgis.refractory.net/download/>>. Acesso em: 10 out. 2010.

POSTGRESQL - The PostgreSQL Global Development Group. **Postgres manual 8.4.6 Documentation**. Disponível em: <<http://www.postgresql.org/docs/manuals/>> Acesso em : 02 jun. 2011.

RAMSEY, P. The state of open source GIS, FOSS4G 2007 Conference, Vancouver, BC, Canada, 2007.

REOLON, CA. **Geotecnologias à cartografia temática: gvSIG**. Associação dos Geógrafos Brasileiros – 2008.

RIGAUX, P.; SCHOLL, M.; VOISARD, A. **Spatial Databases with application to GIS**. Morgan Kaufmann Publishers. 2002.

SABINO, V.; KON, F. **Licenças de Software Livre – História e Características**. Relatório Técnico RT-MAC-IME-USP 2009-01. Instituto de Matemática e Estatística – Universidade de São Paulo.

SEIFERT, M. **Country Experiences: Bavaria, Germany** in FLOSS in Cadastre and Land Registration - Opportunities and Risks. Edited by Daniel Steudler, Mika-Petteri Töhrönen and Gertrude Pieper , April 2010. p 59-63.

SILBERSCHATZ, A.; KORTH, HF.; SUDARSHAN, S. **Sistema de Bancos de Dados**. Terceira edição. São Paulo: Makron Books, 1999.

STEINIGER, S.; BOCHER, E. **An overview on current free and open source desktop GIS developments**, International Journal of Geographical Information Science, 1362-3087, Volume 23, Issue 10, 2009, Pages 1345 – 1370.

SVEEN, AF. Use of Free and Open Source GIS in Commercial Firms. Masters thesis, Norwegian University of Technology and Science. 2008.

TAMBASCIA, CA.; LI, LT.; TEIXEIRA, MD., CUBA, RM; GATTI, SD. **Software Livre na Área de SIG: Ficção ou realidade?** GeoBrasil, 2005 – Disponível em: <http://www.cpqd.com.br/file.upload/09_artigoforum_swlivre_em_sig_sigppt.pdf> Acesso em: 10 jun. 2010.

TOMLINSON, R. **Thinking about GIS**, Third Edition, ESRI Press, 2007.

UCHOA, HN.; FERREIRA, PR. **Geoprocessamento com Software Livre**. Versão 1.0 – 2004. Disponível em: <http://www.hidro.ufcg.edu.br/twiki/pub/Disciplinas/GeotecnologiaAplicada/geoprocessamento_software_livre_uchoa-roberto-v1.0.pdf>. Acesso em: 24 jun. 2010.

UCHOA, HN.; COUTINHO, RJC.; FERREIRA, PR.; COELHO FILHO, LCT.; BRITO, JLNS. **Arquitetura OpenGIS Baseada em Software Livre para Solução de Geoprocessamento**. Congresso da Sociedade Brasileira de Cartografia, 2005a.

UCHOA, HN.; COUTINHO, RJC.; FERREIRA, PR.; COELHO FILHO, LCT.; BRITO, JLNS. **Análise do modulo PostGIS (OpenGIS) para armazenamento e tratamento de dados geográficos com alta performance e baixo custo**. Congresso da Sociedade Brasileira de Cartografia, 2005b.

VIDAL, P., MONTESINOS, M., GONZÁLEZ, V., PEÑARRUBIA, F. **Asociación gvSIG**. Revista Open Planet n.3.- 5as Jornadas gvSIG - Diciembre 2009 pp 45-48

YEUNG, AKW.; HALL, GB. **Spatial Database Systems - Design, Implementation and Project Management** – The GeoJournal Library, Volume 87, Published by Springer, 2007.

WORBOYS, M. **GIS, a Computing Perspective**. Taylor & Francis Routledge, 1995.

ZLATANOVA, S.; STOTER, J. **The role of DBMS in the new generation GIS architecture**, Chapter 8 of Frontiers of Geographic Information Technology, Springer Berlin Heidelberg, 2006.

ANEXO 1 – FUNÇÕES PARA MANIPULAÇÃO DE DADOS ESPACIAIS

Métodos da classe Geometria (*Geometry*):

Métodos Básicos:

- *GeometryType(Geometry A)*: TipoGeometria - retorna uma cadeia de caracteres com o nome da sub-classe instanciada da classe *Geometry*;
- *SRID(Geometry A)*: Retorna um inteiro que é o ID do Sistema Espacial de Referência para um objeto *Geometry*;
- *Envelope(Geometry A)*: Envelope - retorna um objeto *Geometry* que é seu mínimo retângulo envolvente, o polígono definido pelo quadrante ((MINX, MINY), (MAXX, MINY), (MAXX,MAXY), (MINX, MAXY), (MINX, MINY));
- *AsText(Geometry A)*: ComoTexto - retorna uma cadeia de caracteres, que é uma representação textual conhecida para um objeto *Geometry*;
- *AsBinary(Geometry A)*: ComoBinário - retorna uma representação binária conhecida para um objeto *Geometry*;

Métodos para identificar relacionamentos espaciais:

- *Equals(Geometry A, Geometry B)*: Igual - retorna verdadeiro se as duas geometrias forem iguais espacialmente;
- *Disjoint(Geometry A, Geometry B)*: Disjunto - retorna verdadeiro se as geometrias não se tocarem;
- *Intersects(Geometry A, Geometry B)*: Intersecta - retorna verdadeiro se as geometrias se intersectarem;
- *Touches(Geometry A, Geometry B)*: Toca - retorna verdadeiro se existir pelo menos um ponto em comum entre as geometrias dadas;
- *Crosses(Geometry A, Geometry B)*: Cruza - retorna verdadeiro se as geometrias se cruzarem;
- *Within(Geometry A, Geometry B)*: Dentro - retorna verdadeiro se a geometria A contém a geometria B;
- *Contains(Geometry A, Geometry B)*: Contém - retorna verdadeiro se A está contida em B;

- *Overlaps(Geometry A, Geometry B)*: Sobrepõe - retorna verdadeiro se existe sobreposição entre as geometrias dadas;

Métodos para análise espacial:

- *Distance(Geometry A, Geometry B)*: Distância - retorna a distância entre duas geometrias;
- *Buffer(Geometry A, Geometry B, Double x)*: - Área de abrangência - recebe como parâmetros um objeto do tipo *Geometry* e um número real. Retorna o objeto do tipo *Geometry*, que representa todos os pontos cuja distância em relação ao objeto do tipo *Geometry* recebido é menor ou igual ao número real passado por parâmetro;
- *Intersection(Geometry A, Geometry B)*: Interseção - retorna um objeto do tipo *Geometry* formado pelo conjunto de pontos resultantes da interseção entre dois objetos do tipo *Geometry*;
- *Union(Geometry A, Geometry B)*: União - retorna um objeto do tipo *Geometry* formado pelo conjunto de pontos resultantes da união entre dois objetos do tipo *Geometry*;
- *Difference(Geometry A, Geometry B)*: Diferença - retorna um objeto do tipo *Geometry* formado pelo conjunto de pontos resultantes da operação lógica diferença entre dois objetos do tipo *Geometry*;

Métodos das subclasses da classe Geometria:

A seguir são descritos alguns métodos das classes Ponto, Linha e Polígono.

Métodos pertencentes à classe Ponto (*Point*):

- X: Retorna o valor da coordenada x para este ponto;
- Y: Retorna o valor da coordenada y para este ponto.

Os métodos pertencentes à classe Linha (*Line*) podem ser divididos em dois grupos: métodos próprios e herdados da classe Curva (*Curve*).

Métodos da classe Linha (*Line*):

- *NumPoints(Geometry A)*: NumeroDePontos - retorna o número de pontos do objeto do tipo Linha;
- *PointN(Geometry A, integer N)*: PontoN - retorna um objeto do tipo *Point* que é o enésimo ponto do objeto do tipo Linha.

Métodos herdados da classe Curva (*Curve*):

- *Length(Geometry A, Geometry B)*: Comprimento - retorna o comprimento do objeto do tipo Curva;
- *StartPoint(Geometry A)*: PontoInício - retorna um objeto do tipo Ponto que é o ponto inicial do objeto do tipo Curva;
- *EndPoint(Geometry A)*: PontoFinal - retorna um objeto do tipo Ponto que é o ponto final do objeto do tipo Curva;
- *IsClosed(Geometry A)*: Efechado - retorna verdadeiro se o objeto do tipo Curva é fechado;
- *IsRing(Geometry A)*: EAnel - retorna verdadeiro se o objeto do tipo Curva é fechado e simples, ou seja, não passa mais de uma vez pelo mesmo ponto.

Os métodos pertencentes à classe Polígono (*Polygon*) podem ser divididos em dois grupos: métodos próprios e herdados da classe Superfície (*Surface*).

Métodos da classe Polígono (*Polygon*):

- *ExteriorRing(Geometry A)*: AnelExterior - retorna um objeto do tipo *LineString* que é o anel exterior para um objeto do tipo Polígono;
- *NumInteriorRing(Geometry A)*: NumeroAnelExterior - retorna um inteiro que é número de anéis interiores para um objeto do tipo Polígono;

Métodos herdados da classe Superfície (*Surface*):

- *Area(Geometry A)*: Área - retorna a área da geometria;
- *Centroid(Geometry A)*: Centróide - retorna um objeto do tipo Ponto que é o centro matemático do objeto do tipo Superfície;
- *PointOnSurface(Geometry A)*: PontoSobreSuperficie - retorna um objeto do tipo Ponto que se encontra inserido na superfície em questão.

ANEXO 2 – POSTGIS: TABELAS DE METADADOS E EXEMPLOS

Composição da tabela *SPATIAL_REF_SYS* do PostGIS:

```
CREATE TABLE spatial_ref_sys (  
  srid INTEGER NOT NULL PRIMARY KEY,  
  auth_name VARCHAR(256),  
  auth_srid INTEGER,  
  srtext VARCHAR(2048),  
  proj4text VARCHAR(2048)  
)
```

Onde:

- **SRID:** Um valor inteiro que identifica a o Sistema de Referência Espacial de forma única, dentro do banco de dados;
- **AUTH_NAME:** o nome do padrão a que pertence a referência. Por exemplo: EPSG;
- **AUTH_SRID:** a identificação do Sistema de Referência como definido pela autoridade citada na coluna AUTH_NAME.
- **SRTEXT:** contém a definição de coordenadas do Sistema de Referência Espacial.
- **PROJ4TEXT:** contém a definição de coordenadas do Sistema de Referência, em forma de parâmetro, para uso da biblioteca Proj4.no momento de executar as transformações. Essa coluna não está prevista na definição da especificação SFS.

Um exemplo de preenchimento de uma linha da tabela *SPATIAL_REF_SYS*:

- **SRID:** 4618
- **AUTH_NAME:** "EPSG"
- **AUTH_SRID:** 4618
- **SRTEXT:**
"GEOGCS["SAD69",
 DATUM["South_American_Datum_1969",
 SPHEROID["GRS 1967 Modified", 6378160,298.25,
 AUTHORITY["EPSG","7050"]],

```
AUTHORITY["EPSG","6618"]],  
PRIMEM["Greenwich",0,AUTHORITY["EPSG","8901"]],  
UNIT["degree",0.01745329251994328,AUTHORITY["EPSG","9122"]],  
AUTHORITY["EPSG","4618"]]"
```

- PROJ4TEXT: "+proj=longlat +ellps=aust_SA +no_defs"

Composição da tabela **GEOMETRY_COLUMNS** do PostGIS:

```
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  
)
```

Onde:

- F_TABLE_CATALOG, F_TABLE_SCHEMA, F_TABLE_NAME: a qualificação completa para a tabela que contém a coluna geométrica;
- F_GEOMETRY_COLUMN: o nome da coluna geométrica na tabela de feições geométricas;
- COORD_DIMENSION: a dimensão espacial da coluna (2, 3 or 4);
- SRID: a identificação do sistema de referência na tabela SPATIAL_REF_SYS. Quando o sistema de referencia é indefinido, essa coluna é preenchida com o valor -1;
- TYPE: essa coluna não faz parte da especificação SFS. O PostGIS contém o tipo da geometria. Pode ser: *POINT*, *LINESTRING*, *POLYGON*, *MULTIPOINT*, *MULTILINESTRING*, *MULTIPOLYGON*, *GEOMETRYCOLLECTION*. No caso de coleções heterogêneas, deve ser usado o tipo *GEOMETRY*.

Comandos para atualização da tabela *GEOMETRY_COLUMNS*

Quando uma coluna do tipo *GEOMETRY* é criada ou removida da base de dados, a tabela *GEOMETRY_COLUMNS* deve ser atualizada. Para evitar a atualização manual da tabela, que é trabalhosa e pode provocar erros, o PostGIS implementa cinco comandos. São eles:

- *AddGeometryColumn*: cria uma coluna do tipo *GEOMETRY* em uma tabela e adiciona as informações sobre a coluna na tabela *GEOMETRY_COLUMNS*;
- *DropGeometryColumn*: remove uma coluna *GEOMETRY* de uma tabela e remove também os dados sobre a coluna da tabela *GEOMETRY_COLUMNS* ;
- *DropGeometryTable*: remove uma tabela que tem coluna tipo *GEOMETRY* e remove também todos os dados sobre a tabela removida da tabela *GEOMETRY_COLUMNS*;
- *UpdateGeometrySRID*: atualiza o SRID de todas as colunas do tipo *GEOMETRY*, atualiza as restrições de integridade do PostGIS e a tabela *GEOMETRY_COLUMNS* ;
- *Probe_geometry_columns*: (atualiza a tabela *GEOMETRY_COLUMNS* a partir da identificação, no banco de dados, das tabelas com restrição de integridade de geometria);
- *Populate_Geometry_Columns*: verifica todas as colunas do tipo *GEOMETRY* criadas no banco de dados, adiciona aquelas ainda não cadastradas, acrescenta restrições de integridade padrão para as colunas tipo *GEOMETRY* e identifica a falta de SRID e de tipo nas colunas geométricas já cadastrada. Se não for informada a tabela a ser verificada, todas as linhas da tabela *GEOMETRY_COLUMNS* são recriadas.

Exemplo de criação de uma tabela com uma coluna do tipo *GEOMETRY*

```
CREATE TABLE pe
(
  gid serial NOT NULL,
  geocodigo character varying(7),
  nome character varying(32),
  uf character varying(2),
  CONSTRAINT pe_pkey PRIMARY KEY (gid)
);
SELECT AddGeometryColumn('pe','the_geom','4618','POLYGON',2);
Alter table pe add constraint enforce_dims_the_geom CHECK
(st_ndims(the_geom) = 2);
Alter table pe add constraint enforce_geotype_the_geom CHECK
(geometrytype(the_geom) = 'POLYGON'::text OR the_geom IS NULL);
Alter table pe add constraint enforce_srid_the_geom CHECK
(st_srid(the_geom) = 4618);
```