

Marcela Lopes Neves

Tratamento de Dados Geográficos e  
Consultas Espaciais em Bancos de  
Dados Objeto-Relacionais

VIII Curso de Especialização em  
Geoprocessamento  
2005



UFMG

Instituto de Geociências  
Departamento de Cartografia  
Av. Antônio Carlos, 6627 – Pampulha  
Belo Horizonte  
cartog@igc.ufmg.br

TRATAMENTO DE DADOS GEOGRÁFICOS E CONSULTAS ESPACIAIS EM BANCOS  
DE DADOS OBJETO-RELACIONAIS

Monografia apresentada ao Curso de Especialização em  
Geoprocessamento do Departamento de Cartografia do  
Instituto de Geociências da UFMG.

Orientador: Clodoveu A. Davis Jr.

Belo Horizonte  
Instituto de Geociências  
2005

Neves, Marcela Lopes

Tratamento de Dados Geográficos e Consultas Espaciais em Bancos de Dados Objeto-Relacionais, 2005.

iv, 49 f.: il.

Monografia (Especialização) – Universidade Federal de Minas Gerais. Instituto de Geociências. Departamento de Cartografia, 2005.

Orientador: Clodoveu A. Davis Jr.

Geoprocessamento 2. Transporte Público 3. Modelagem de Dados 4. PostGIS. I. Título.

## **Resumo**

O presente trabalho apresenta um estudo sobre a implementação de um esquema conceitual de dados orientado a objetos, voltado para aplicações geográficas na área de transportes públicos, em um banco de dados objeto-relacional, no caso, o PostgreSQL. Considerando a necessidade que há por parte dos gestores de transporte público em utilizar sistemas georreferenciados, o esquema criado permite fazer consultas espaciais em um banco de dados com informações sobre bairros, arruamentos, linhas e paradas de ônibus.

## **Abstract**

The present work presents a study of the implementation of a object-oriented conceptual data scheme, adressed to geographic applications for public transport, in an object-related data base, in this case, the PostgreSQL. Considering the necessity the public transport managers has to use georeferenced systems, the scheme that was created allows to make spatial consults in a data base with informations of districts, streets, bus lines and bus stop.

## SUMÁRIO

<b>1- INTRODUÇÃO.....</b>	<b>5</b>
<b>2- REFERENCIAL TEÓRICO .....</b>	<b>8</b>
2.1- SGBDs e SIGs.....	8
2.2- OMT-G.....	10
2.3- O Modelo Objeto-Relacional.....	14
2.4- PostGreSQL.....	15
2.5- MapServer .....	19
<b>3 – ESTUDO DE CASO: SGBDOR E SIG EM TRANSPORTE PÚBLICO URBANO .....</b>	<b>20</b>
3.1 – Transporte público urbano.....	20
3.2 – Implementação do estudo de caso .....	22
<b>4- CONSIDERAÇÕES FINAIS.....</b>	<b>41</b>

## LISTA DE FIGURAS

Figura 1 - Arquitetura Genérica de um SIG Fonte [CAM1996] .....	8
Figura 2 - Geo-campos Fonte:[BORGES, 2005].....	12
Figura 3 - Geo-objetos com geometria Fonte:[BORGES, 2005] .....	12
Figura 4 - Relacionamentos Fonte:[BORGES, 2005] .....	13
Figura 5 - Cardinalidade Fonte:[BORGES, 2005] .....	14
Figura 6 - Tipos Geométricos do PostgreSQL Fonte:[QUEIROZ, 2005].....	15
Figura 7 - Tipos de Dados Espaciais do PostGIS Fonte:[QUEIROZ, 2005] .....	17
Figura 8 - Modelo de Dados para Transporte Público.....	23
Figura 9 - Área de Trabalho .....	24
Figura 10 - Representação dos Dados Inseridos na Base de Dados .....	25
Figura 11 - Mapa para visualização de bairros, pontos e itinerário das linhas de ônibus cadastrados na base de dados .....	37
Figura 12 - Representação visual do resultado da consulta 1 .....	39
Figura 13 - Representação visual do resultado da consulta 4.....	40

## LISTA DE TABELAS

Tabela 1 - Descrição da tabela <i>bairro</i> .....	26
Tabela 2 - Descrição da tabela <i>linha</i> .....	27
Tabela 3 - Descrição da tabela <i>trecho</i> .....	27
Tabela 4 - Descrição da tabela <i>itinerario</i> .....	27
Tabela 5 - Descrição da tabela <i>ponto</i> .....	28

## 1- Introdução

O uso de Bancos de Dados Objeto-Relacionais é hoje uma forte tendência no desenvolvimento de sistemas, pois possibilitam tratar dados e consultas complexas, dispensando a camada de persistência de dados entre o SGBD e a aplicação, e oferecendo um diferencial na qualidade da solução implantada.

Os Bancos de Dados Objeto-Relacionais (**SGBDOR**) suportam uma grande quantidade de requisições simultâneas e de locais diversos, como é o caso dos acessos via Internet.

Segundo OLIVEIRA (2005),

*“ O **SGBDOR** faz com que as regras de negócio fiquem mais consolidadas em um modelo único e sem interfaces de transformação de dados, que, independente de qualquer infraestrutura, é uma interface sempre considerada pelos desenvolvedores como um potencial gargalo de tráfego e integridade de informações.” (p.27)*

Os **SGBDOR** foram concebidos para apresentar boa performance no acesso, armazenamento e recuperação de dados complexos multimídia, como imagens, vídeos e sons. Tal performance dificilmente poderia ser obtida utilizando-se um banco de dados relacional. Os **SGBDOR** são aplicáveis no geoprocessamento, para armazenamento de uma base cartográfica com seus objetos geográficos e dados alfanuméricos, proporcionando pesquisas espaciais eficientes.

Existe uma crescente necessidade de manipulação de dados complexos, como imagens e objetos geográficos, de forma eficiente e rápida, considerando ainda o acesso simultâneo por

vários usuários. Para isso os dados precisam estar bem estruturados, num gerenciador banco de dados apropriado para consultas espaciais.

Para avaliar esses fatores em um estudo de caso prático, neste projeto de monografia será criado o projeto de um banco de dados orientado a objetos que permita fazer consultas sobre a rede de transporte urbano de uma cidade. Os sistemas de transportes coletivos das grandes cidades são frequentemente apontados como um dos mais críticos problemas urbanos. Uma das causas da baixa qualidade dos serviços prestados é a falta de informação sobre a operação destes sistemas. Os SIG são uma boa solução para este problema, já que fornecem para os órgãos gestores de transporte informações espaciais e alfanuméricas, permitindo uma melhor análise da situação. Entretanto, a maioria dos SIG disponíveis no mercado que propõem soluções para transporte público não são multiusuários ou não estão preparados para um grande volume de consultas simultâneas.

Assim, o principal objetivo desta monografia é criar e implementar um modelo de dados orientado a objetos para um SIG voltado para transporte público, utilizando a característica objeto-relacional disponibilizada pelo SGBDOR PostgreSQL e sua variação voltada para dados espaciais, o PostGIS. Serão projetadas e executadas consultas a dados espaciais de acordo com o modelo criado.

O banco de dados projetado para o estudo de caso contém elementos da rede de transporte urbano de uma cidade. As principais classes deste modelo são: bairro, linha de ônibus, ponto de parada de ônibus e endereço. A partir desse modelo é possível avaliar o atendimento de linhas de ônibus aos bairros, a distribuição espacial dos pontos de ônibus, o atendimento do transporte às áreas de referência da cidade, dentre outras aplicações.

Como esse trabalho visa avaliar as possibilidades de uso de um SGBDOR em um banco de dados espacial, e não desenvolver análises reais sobre a rede de transportes de uma cidade, não é necessário contar com uma base de dados real. Com isso, o banco de dados usado para testes teve, inclusive, alguns dados inseridos manualmente.



Para modelar o banco de dados, foi utilizado o modelo conceitual de dados geográficos OMT-G. O OMT-G é um modelo que parte do diagrama de classes da UML (Unified Modeling Language), e introduz primitivas geográficas, o que permite modelar a geometria dos dados geográficos. Segundo Borges (2005), esse modelo é baseado em três conceitos principais: classes, relacionamentos e restrições de integridade espaciais.

O esquema conceitual criado foi implementado no PostGIS, uma extensão geográfica, gratuita e de código fonte aberto do SGBDOR PostGreSQL, que visa permitir o gerenciamento de informações geo-espaciais. Sua distribuição oficial provê tipos geométricos, indexação espacial e operadores espaciais. Esse banco de dados foi escolhido por apresentar características de um banco objeto-relacional e por incorporar tipos de dados espaciais, fornecendo as funções e os operadores que possibilitam consultar as geometrias estruturadas em tabelas para serem apresentadas aos usuários, até mesmo através de um *browser*.

A partir do banco de dados modelado e implementado, foram projetadas e executadas algumas consultas aos dados espaciais.

Finalmente, o trabalho apresenta uma avaliação crítica da complexidade de criação do banco de dados utilizando a OMT-G e a sua implementação no PostGIS.

Desenvolvendo esta monografia já é possível observar as dificuldades de obter uma base de dados georreferenciada confiável e aprender sobre a importância da organização e modelagem de dados. Este trabalho servirá como base para um projeto maior que visa desenvolver um sistema para controlar toda a operação dos veículos do sistema de transporte urbano de uma cidade.

## 2- Referencial Teórico

### 2.1- SGBDs e SIGs

Segundo (CAMARA, 1996), numa visão abrangente pode-se considerar que um SIG tem os seguintes componentes (figura 1): interface com o usuário; entrada e integração de dados; funções de processamento; visualização e plotagem; e armazenamento e recuperação de dados (organizados sob a forma de um banco de dados geográfico).

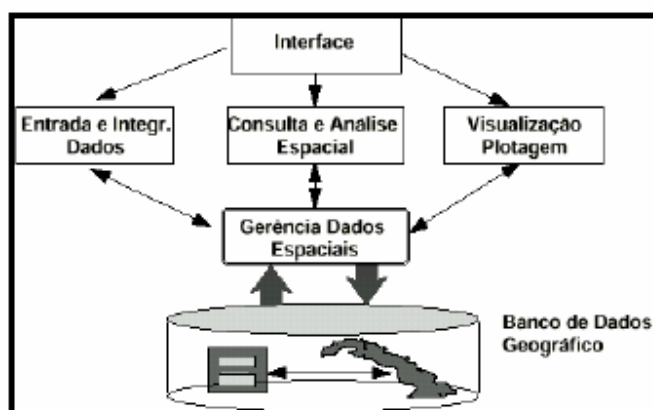


Figura 1 - Arquitetura Genérica de um SIG Fonte (CAMARA, 1996)

Esses componentes se relacionam de forma hierárquica. No nível mais próximo ao usuário a interface define como o sistema é operado e controlado. No nível intermediário, um SIG deve ter mecanismos de processamento de dados espaciais (entrada, edição, análise, visualização e saída). No nível mais interno, um sistema de gerência de banco de dados geográfico oferece armazenamento e recuperação dos dados espaciais e seus atributos.

Os SGBDs podem desempenhar um papel fundamental no desenvolvimento de novos aplicativos GIS, assim como facilitar o seu desenvolvimento.

O SGBD deve garantir que as propriedades convencionais sejam aplicáveis a dados geográficos. Estas propriedades incluem três requisitos importantes: eficiência (acesso e

modificações de grandes volumes de dados); integridade (controle de acesso por múltiplos usuários); e persistência (manutenção de dados por longo tempo, independentemente dos aplicativos que acessam o dado).

Sistemas de Informação necessitam de um banco de dados estruturado de forma a proporcionar o armazenamento dos dados gerados pelas regras de negócio de forma segura, eficiente e íntegra, e a tecnologia mais popular nesta área ainda é a relacional, que lida com dados planos na forma de tabelas em disco. Os dados gerados pelas aplicações atuais e futuras estão cada vez mais complexos, e uma estrutura plana com tabelas e relacionamentos comprovou ser pouco eficiente para a persistência de dados além de dificultar o equacionamento das questões de otimização de consultas, gerência de transações e controle de integridade e de concorrência (OLIVEIRA, 2005).

No caso de sistemas de informação geográfica, ocorre de cada sistema implementar seu próprio modelo de dados, não sendo possível definir um conjunto único de regras para o mapeamento de esquemas conceituais nos esquemas lógicos dos diversos SIGs existentes (LISBOA, 2001).

O armazenamento de dados dentro de um SGBD elimina o problema de uso de estruturas proprietárias, que dificultam a interoperabilidade entre sistemas. Elimina também o uso de armazenamentos heterogêneos, que provocam o armazenamento em locais diferentes, tornando a manutenção complicada, e a recuperação dos dados mais custosa (OLIVEIRA, 2005).

De acordo com ELMASRI e NAVATHE (2004), existem três categorias de modelo de dados: modelo conceitual, modelo de representação e modelo físico.

Os modelos conceituais de dados oferecem conceitos que estão próximos do modo como muitos usuários percebem os dados. Estes modelos utilizam os conceitos de entidade (representa um objeto ou conceito do mundo real), atributo (representa uma propriedade que

descreve uma entidade) e relacionamento (representa uma interação entre as entidades). Como exemplo de modelos deste nível temos a UML e OMT-G.

Os modelos de dados representativos fornecem conceitos que podem ser compreendidos por usuário finais, porém não estão muito afastados do modo como os dados estão organizados no computador. Esses modelos escondem alguns dos detalhes sobre o armazenamento, mas podem ser implementados num sistema de computador de uma forma direta. Estes modelos são muito utilizados, e incluem os modelos relacionais e os modelos objeto-relacionais.

Os modelos físicos de dados descrevem os detalhes de como os dados estão armazenados no computador, representando informações como formato de registros, ordenamento de registros e caminhos de acesso. Neste modelo existe a linguagem SQL (Structured Query Language) para consultas em bancos de dados. Um padrão revisado e mais expandido chamado SQL2 ou SQL-92 foi desenvolvido para consultas em bancos de dados relacionais, enquanto a SQL3 estende ainda mais a SQL com banco de dados objeto-relacionais. (ELMASRI e NAVATHE, 2004).

## **2.2- OMT-G**

As técnicas tradicionais de modelagem de dados ainda encontram diversas dificuldades em representar informações geográficas devido à complexidade destas informações no que diz respeito à localização, tempo e precisão de representação.

Segundo BORGES (2005), a modelagem do mundo real é uma atividade complexa porque envolve a discretização do espaço geográfico para a sua devida representação fazendo com que a modelagem de aplicações geográficas necessite de modelos mais adequados, capazes de capturar a semântica dos dados geográficos, oferecendo mecanismos de abstração mais elevados e independência de implementação. Apesar de toda a expressividade oferecida pelas

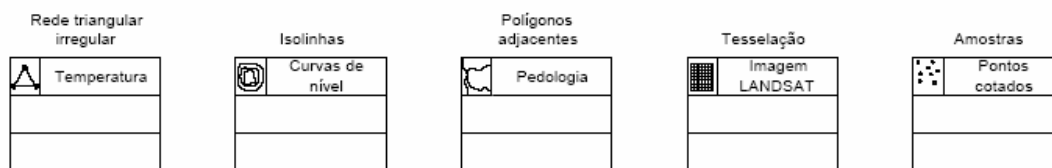
técnicas tradicionais de modelagem, dificuldades surgem devido ao fato de que os dados geográficos possuem aspectos peculiares, particularmente com respeito à codificação da localização espacial e do tempo de observação, bem como em relação ao registro de fatores externos, como sua precisão de obtenção.

O modelo OMT-G trabalha com os níveis de apresentação e representação, sendo que no nível de apresentação oferece ferramentas para a especificação dos diferentes aspectos visuais que as entidades podem assumir ao longo de seu uso em aplicações.

O modelo OMT-G baseia-se nas primitivas definidas para o diagrama de classes da Unified Modeling Language (UML) (Rational Software Corporation, 1997), introduzindo primitivas geográficas que aumentam a capacidade de representação semântica da UML. Portanto, o modelo OMT-G oferece suporte a estruturas topológicas “todo-parte”, estruturas de rede, múltiplas representações de objetos e relacionamentos espaciais além de prover primitivas para modelar a geometria e a topologia dos dados geográficos. O modelo permite também a especificação de atributos alfanuméricos e métodos associados para cada classe. Os principais pontos do modelo são suas representações gráficas e sua capacidade de codificação, uma vez que anotações textuais são substituídas pelo desenho de relacionamentos explícitos, que denotam a dinâmica da interação entre os diversos objetos espaciais e não espaciais.

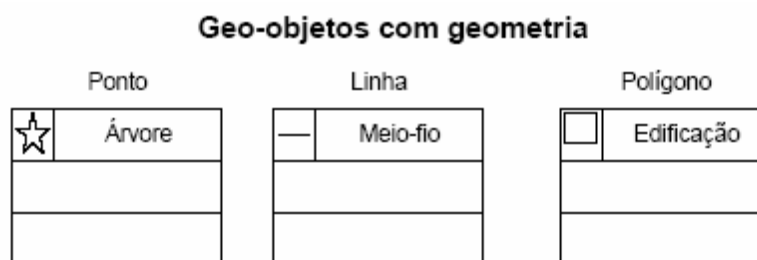
O OMT-G cria o conceito de *Classes Georreferenciadas*, que se especializam em classes do tipo *Geo-campo* e *Geo-Objeto* (BORGES,2005).

As classes do tipo *Geo-Campo* representam objetos distribuídos continuamente pelo espaço, e as classes do tipo *Geo-Objeto* representam objetos geográficos individualizáveis, que possuem identificação com elementos do mundo real. Esses objetos podem ter ou não atributos não-especiais. A figura 2 mostra exemplos de *Geo-Campos*.



**Figura 2 - Geo-campos Fonte (BORGES, 2005)**

Para a construção do modelo utilizado nesta monografia, foram utilizadas classes do tipo *geo-objeto com geometria*, que de acordo Borges representa objetos que possuem apenas propriedades geométricas, e é especializada nas classes *ponto*, *linha* e *polígono*. Essas classes possuem um padrão simbólico de representação representado na figura 3.



**Figura 3 - Geo-objetos com geometria Fonte (BORGES, 2005)**

Ainda de acordo com BORGES (2005), o modelo OMT-G representa os seguintes tipos de relacionamentos entre suas classes: associações simples, relações topológicas de rede e relações espaciais.

As relações simples representam relacionamentos estruturais entre objetos de diferentes classes, tanto convencionais como georreferenciadas. Esse relacionamento é representado por uma linha contínua ligando duas classes. Uma associação pode ter sobre o seu nome uma seta mostrando qual o sentido da relação.

As relações espaciais no modelo OMT-G têm por objetivo tornar explícita a interação espacial entre as classes quando for relevante para o propósito da aplicação. Todas as relações espaciais são representadas por linhas pontilhadas.

As relações em rede são relacionamentos entre objetos que estão conectados uns com os outros, sendo representados por nós e arcos.

A figura 4 representa as relações espaciais no padrão OMT-G.

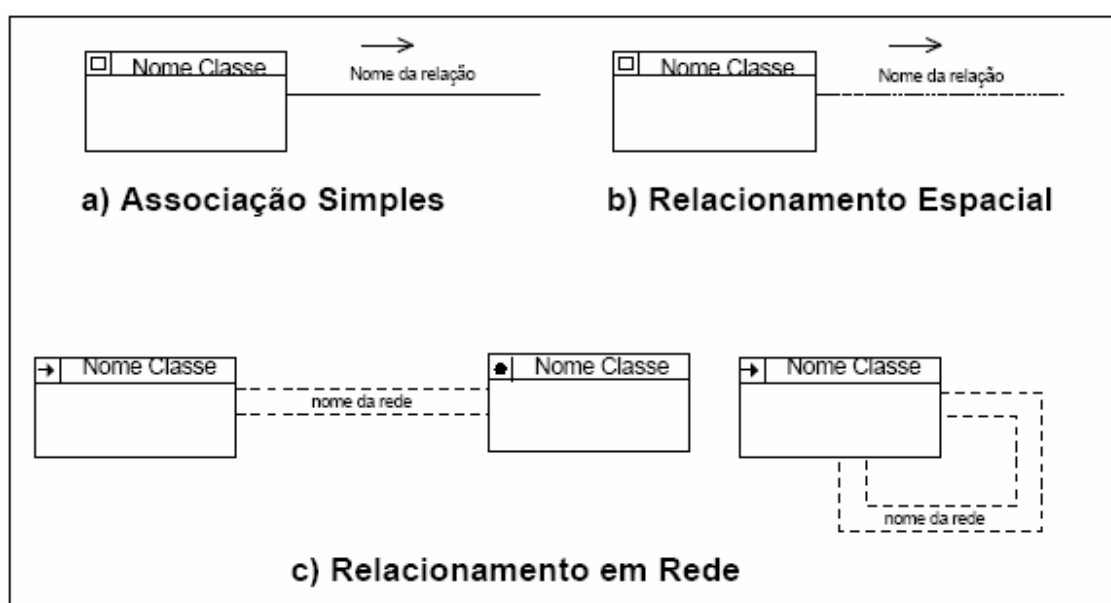


Figura 4 - Relacionamentos Fonte: (BORGES, 2005)

O modelo OMT-G considera as seguintes relações espaciais entre as Classes Georreferenciadas: *disjunto, contém, dentro de (contido), toca (encontra), cobre, coberto por, sobrepõe, adjacente, perto de, acima, abaixo, sobre, sob, entre, coincide, cruza, atravessa, em frente a, à esquerda e à direita*. Nesta monografia foram utilizadas as relações *contém* e *sobre*. A relação *contém* é utilizada quando a geometria da classe que contém envolve a geometria das classes contidas. A classe que contém deve ser do tipo *Polígono (Geo-Objeto)* ou *Subdivisão Planar*

(*Geo-Campo*). A relação *sobre* é utilizada no sentido de “em cima de” / “embaixo de”, no mesmo plano.

Os relacionamentos são caracterizados pela cardinalidade, que representa o número de instâncias de uma classe que pode estar associada a uma instância da outra classe. A notação de cardinalidade adotada pelo modelo OMT-G é a utilizada na UML, conforme figura 5.

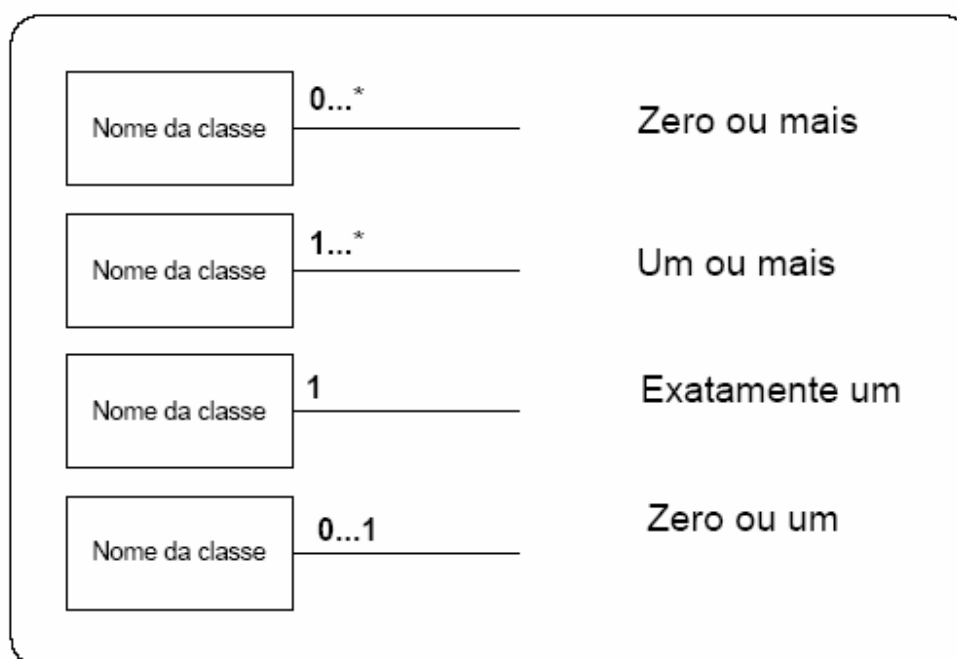


Figura 5 - Cardinalidade Fonte: (BORGES, 2005)

### 2.3- O Modelo Objeto-Relacional

O Modelo Objeto-Relacional é uma extensão do Modelo Relacional tradicional, incorporando novas funcionalidades e a capacidade de modelagem para tratar dados complexos (objetos) sobre estruturas relacionais (relações, usualmente mapeadas para tabelas físicas). Trata-se de uma modelagem mais natural e mais fácil de ser compreendida.

Neste modelo são utilizados componentes que são responsáveis por garantir que a modelagem seja feita de forma íntegra, segura e eficiente, mesmo quando milhares de usuários estiverem



conectados simultaneamente em uma aplicação na Internet e acessando um mesmo conjunto de objetos (OLIVEIRA, 2005).

O Modelo Objeto-Relacional apresenta os seguintes elementos: relações aninhadas, tipos complexos, herança, tipo referência, consultas, funções e procedimentos.

O SGBDs Oracle e PostGreSQL são dois exemplos de bancos de dados objeto-relacionais (FERREIRA, 2002). No caso de dados espaciais, existem extensões para modelagem física em ambos os SGBDOR, baseadas nas especificações do OpenGIS, porém existem variações relevantes entre os modelos internos de dados, semântica dos operadores espaciais, mecanismos de indexação e esquema de sintaxe da SQL estendida com tipos espaciais.

## 2.4- PostGreSQL

O PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional, gratuito e de código fonte aberto. Em sua distribuição oficial, o PostgreSQL disponibiliza tipos geométricos, recursos de indexação espacial e operadores espaciais. (QUEIROZ, 2005)

Os tipos geométricos do PostgreSQL estão apresentados na figura 6.

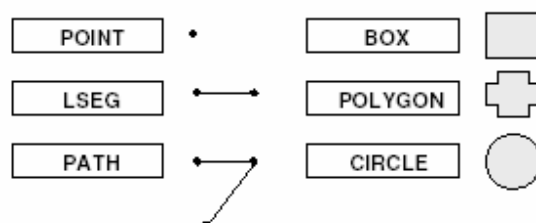


Figura 6 - Tipos Geométricos do PostgreSQL Fonte: (QUEIROZ, 2005)

Os pontos (*point*) são os blocos de construção bidimensionais fundamentais para os tipos geométricos. são especificados utilizando coordenadas x e y na forma de números de ponto flutuante.

Os segmentos de linha (*lseg*) são representados por pares de pontos que representam os pontos das extremidades do segmento de linha.

Os caminhos (*paths*) são representados por listas de pontos conectados. Os caminhos podem ser *abertos*, onde o primeiro e o último ponto da lista não são considerados conectados, e *fechados*, onde o primeiro e o último ponto são considerados conectados.

As caixas (*box*) são representadas por pares de pontos de vértices opostos da caixa. Os vértices são reordenados na entrada para armazenar o vértice direito superior e, depois, o vértice esquerdo inferior.

Os polígonos (*polygon*) são representados por uma lista de pontos (os vértices do polígono). Provavelmente os polígonos deveriam ser considerados equivalentes aos caminhos fechados, mas são armazenados de forma diferente e possuem um conjunto próprio de rotinas de suporte.

Os círculos (*circle*) são representados por um ponto central e um raio. (<http://www.javalinux.com.br:8080/pg74/datatype-geometric.html>)

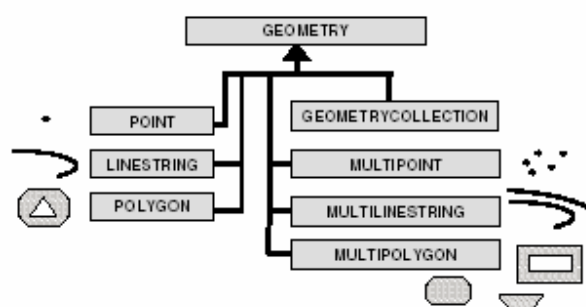
Para indexação, esse SGBD utiliza uma R-Tree, cuja implementação está limitada a dados com até 8Kbytes, sendo assim bastante limitada para dados geográficos reais. No entanto, o PostgreSQL permite a definição de uma R-Tree sobre o mecanismo de indexação conhecido como GiST (Generalized Search Trees). Este método de indexação não possui restrições de tamanho do dado a ser indexado e pode fornecer tanto as funcionalidades de uma B-Tree quanto de uma R-Tree e suas variantes (QUEIROZ, 2005).

O PostgreSQL apresenta apenas alguns poucos e limitados operadores. Estes operadores realizam a computação apenas sobre o retângulo envolvente das geometrias e não diretamente

nelas. Os tipos de dados disponíveis são simples, como polígonos, e não permitem a representação de buracos, não existindo também geometrias que permitam representar objetos mais complexos, como os formados por conjuntos de polígonos.

Como as funcionalidades oferecidas são bastante limitadas para o desenvolvimento de Sistemas de Informação Geográficas, uma extensão do PostgreSQL foi desenvolvida: o PostGIS, que é uma extensão geográfica, também gratuita e de código fonte aberto, que visa permitir ao SGBD PostgreSQL gerenciar informações geo-espaciais.

A figura 7 mostra os tipos de dados espaciais suportados pelo PostGIS.



**Figura 7 - Tipos de Dados Espaciais do PostGIS Fonte: (QUEIROZ, 2005)**

Segundo QUEIROZ (2005), a criação de uma tabela com tipo espacial no PostGIS é realizada em duas etapas. Na primeira são definidos os atributos alfanuméricos e na segunda é utilizada a função `AddGeometryColumn` para adicionar a coluna com o tipo espacial. Essa função implementada no PostGIS e especificada no OpenGIS, realiza todo o trabalho de preenchimento de uma tabela especial de metadados, denominada “`geometry_columns`”. Os parâmetros dessa função são:

- nome do banco de dados
- nome da tabela que irá conter a coluna espacial
- nome da coluna espacial
- sistema de coordenadas em que se encontram as geometrias da tabela

- tipo da coluna espacial, que serve para criar uma restrição que verifica o tipo do objeto sendo inserido na tabela
- tipo da coluna espacial, que serve para criar uma restrição que verifica o tipo do objeto sendo inserido na tabela
- dimensão em que se encontram as coordenadas dos dados

Após criar as tabelas, pode-se inserir os dados usando o comando SQL INSERT. Para isso, pode-se usar a representação textual das geometrias em conjunto com a função GeometryFromText, que recebe a representação textual e mais o sistema de coordenadas em que se encontra a geometria e compõe o objeto geográfico para armazenamento.

Colunas com tipos espaciais podem ser indexadas através de uma R-Tree. Os índices espaciais são usados em consultas que envolvem predicados espaciais, como no caso de consultas por janela, onde um retângulo envolvente é informado e as geometrias que interagem com ele devem ser recuperadas rapidamente. O operador && pode ser usado para explorar o índice espacial.

Abaixo estão listados os diversos tipos de operadores espaciais disponíveis no PostGIS:

- Operadores topológicos:

`equals(geometry, geometry)`

`disjoint(geometry, geometry)`

`intersects(geometry, geometry)`

`touches(geometry, geometry)`

`crosses(geometry, geometry)`

`within(geometry, geometry)`

`overlaps(geometry, geometry)`

`contains(geometry, geometry)`

`relate(geometry, geometry)`

- Operador de construção de mapas de distância:

`buffer(geometry, double, [integer])`

- Operador para construção do Fecho Convexo:

`Convexhull(geometry)`

- Operadores de conjunto:

`Intersection(geometry, geometry)`

`geomUnion(geometry, geometry)`

`symdifference(geometry, geometry)`

`difference(geometry, geometry)`

- Operadores métricos:

`distance(geometry, geometry)`

`area(geometry)`

- Centróide de geometrias:

`Centroid(geometry)`

- Validação

`isSimple(geometry)`

## 2.5- MapServer

O MapServer é software livre que disponibiliza um ambiente de desenvolvimento para construção de aplicações georreferenciadas na Internet. Ele possui diversas funcionalidades para pesquisa e visualização de sistemas de informações geográficas (GIS) armazenados em banco de dados como Oracle, MySQL, PostgreSQL, arquivos DBF, etc. (FERRARI, 2005)

Ainda segundo FERRARI(2005), o MapServer é uma aplicação do tipo servidor de mapas, disponibilizado para sistemas operacionais Unix e Windows, possuindo ainda uma interface de programação chamada MapScript que torna possível o uso de linguagens populares como PHP, Perl e até Java para a construção de aplicações mais complexas.

O MapServer dá suporte ao formato vetorial ShapeFile do ArcView. O formato de saída do aplicativo pode ser personalizado, permitindo a construção automática de legenda e barras de escala, a construção de mapas temáticos usando expressões lógicas ou regulares baseadas em

classes, entre outras facilidades. Este aplicativo provê condições suficientes de suporte para uma grande variedade de aplicações espaciais na Web. (MIRANDA, 2003)

Para publicar mapas, o MapServer utiliza um arquivo de configuração em formato texto, chamado de “*map file*” extensão .map, contendo uma descrição de todos os planos de informação (mapas temáticos) e seus parâmetros, como sistema de referência, nome do arquivo, rótulos a serem usados com o mapa, escala, cor, etc. As informações contidas neste arquivo serão usadas pelo MapServer quando estiver atendendo solicitações sobre mapas. (MIRANDA, 2003)

O papel do MapServer nesta monografia é possibilitar a visualização do resultado de consultas espaciais feitas no PostGis.

### **3 – Estudo de caso: SGBDOR e SIG em Transporte Público Urbano**

#### **3.1 – Transporte público urbano**

No Brasil, 80% da população vive nas cidades. As atividades econômicas da maioria das cidades dependem do transporte público, pois esse é o modo utilizado por grande parte dos clientes e trabalhadores do comércio, do setor de serviços e da indústria.

O transporte público urbano é, assim, imprescindível para a vitalidade econômica, a justiça social, a qualidade de vida e a eficiência das cidades modernas. As atividades de planejamento e gestão são vitais para garantir a qualidade e a eficiência do serviço de transporte público urbano. (FERRAZ, 2001)

Os sistemas de transporte público são frequentemente discutidos, principalmente os temas ligados à oferta, demanda, custos, qualidade e condições oferecidas aos usuários dos

transportes públicos, uma vez que a maioria dos deslocamentos para trabalho são feitos através deste meio.

O geoprocessamento tem demonstrado ser uma ferramenta de integração de bancos de dados informatizados e de visualização de informações, facilitando as atividades de planejamento, projeto, operação e monitoração dos sistemas de transporte e trânsito.

Um SIG pode relacionar uma informação importante de infra-estrutura do transporte, como condições de estradas, volume de tráfego, taxa de acidentes e capacidade de uma ponte, à sua localização (RODRIGUES, 2001).

Existem vários estudos realizados, que utilizam geoprocessamento como ferramenta para resolver problemas ligados ao transporte público, tais como:

- planejamento da capacidade e manutenção de vias
- análise para determinar pontos de embarque e desembarque das linhas de ônibus (PRADO, 2001)
- acompanhamento estatístico de acidentes de tráfego e assaltos a ônibus (CARVALHO, 2002)
- análise de itinerários das linhas ônibus, possibilitando uma análise da distribuição de linhas pelas áreas de uma cidade (PRADO, 2001)
- acompanhamento do posicionamento de veículos, utilizando receptores de GPS (Global Positioning System)
- determinar caminho ótimo entre pontos de origem e destino. (OLIVEIRA, 2000)

Através de bases de dados sobre as situações descritas acima, é possível confeccionar produtos cartográficos destinados às necessidades de planejamento, acompanhamento e operação de um sistema de transporte público.

A acessibilidade aos transportes coletivos é um dos temas mais importantes quando se trata de transporte público. Segundo JANUARIO (1995), um estudo de acessibilidade pode trazer

informações relevantes para a reprogramação do sistema de transporte ou para a estruturação de uma nova rede de transporte coletivo.

A acessibilidade de transportes considera a facilidade de acesso a um lugar, sistema ou região. Um estudo sobre acessibilidade visa quantificar ou medir as facilidades e/ou dificuldades de acesso, ou seja, chegada e saída a determinado local dentro de uma área considerada. A acessibilidade pode ser traduzida em termos de infra-estrutura, quantidade ou qualidade do transporte oferecido para se chegar a um local desejado.

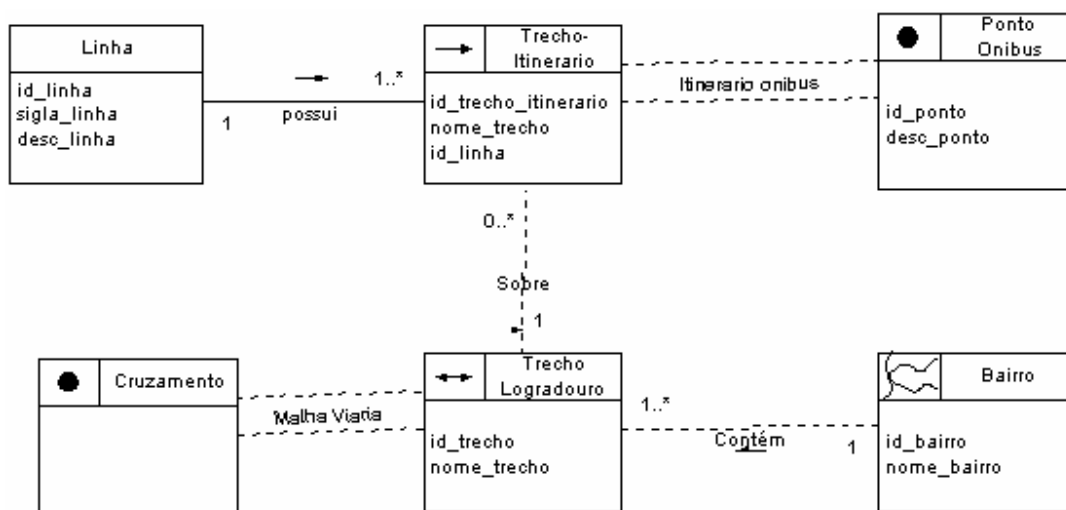
Segundo a ANTP (Associação Nacional de Transportes Públicos), o transporte público por ônibus tem objetivo de organizar a oferta de transporte para que se possa atender os desejos de deslocamentos das pessoas. A organização da oferta de transportes é feita considerando sempre vários aspectos relevantes, como a demanda, os custos e o nível de serviço desejado.

### **3.2 – Implementação do estudo de caso**

Para exemplificar a implementação de um banco de dados objeto-relacional e consultas espaciais sobre os dados inseridos nesse banco, foram seguidas as seguintes etapas:

1ª etapa) Criação de um modelo de dados orientado a objetos, baseado no sistema de transporte público de Belo Horizonte. Foram consideradas apenas algumas variáveis deste sistema: as que permitem fazer consultas relacionando logradouros, linhas de ônibus e pontos de parada de ônibus por bairros. O modelo abaixo (Figura 8) foi feito utilizando a metodologia OMT-G e o software Visio Professional, versão 5.0 para Microsoft Windows. Verifica-se que existe relacionamento espacial entre as principais classes: Trecho-itinerário, Ponto de ônibus, Trecho-logradouro e Bairro.

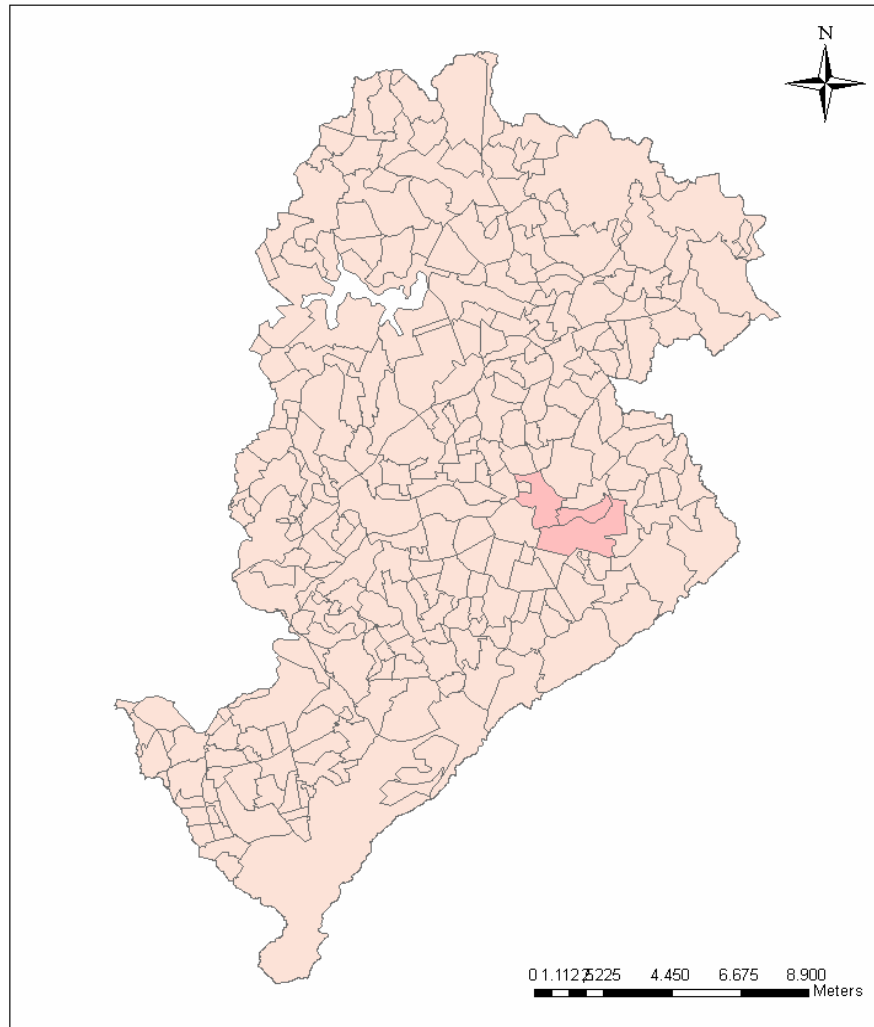




**Figura 8 - Modelo de Dados para Transporte Público**

2ª etapa) Selecionar dados para serem trabalhados. A partir de uma base de dados da Prodabel que contém dados sobre limites do município de Belo Horizonte, seus bairros e arruamentos, foram selecionados os dados relativos aos bairros de Santa Tereza, Santa Efigênia e Floresta. Para fazer essa seleção, foi utilizado o software MapInfo Professional, versão 7.0.

A área selecionada para ser trabalhada nesta monografia é a área em destaque na Figura 9:



**Figura 9 - Área de Trabalho**

Sobre a base trabalhada no MapInfo, foi utilizado o software ArcMap versão 9.0 para criar layers de linhas de ônibus e pontos de parada de ônibus. Os dados sobre as linhas, os itinerários e pontos de ônibus foram obtidos no site da BHTrans. Abaixo está o mapa completo (Figura 10) dos dados utilizados para povoar a base dados do banco criado no PostGIS.

## Linhas e Pontos de Ônibus dos Bairros Santa Tereza, Santa Efigênia e Floresta

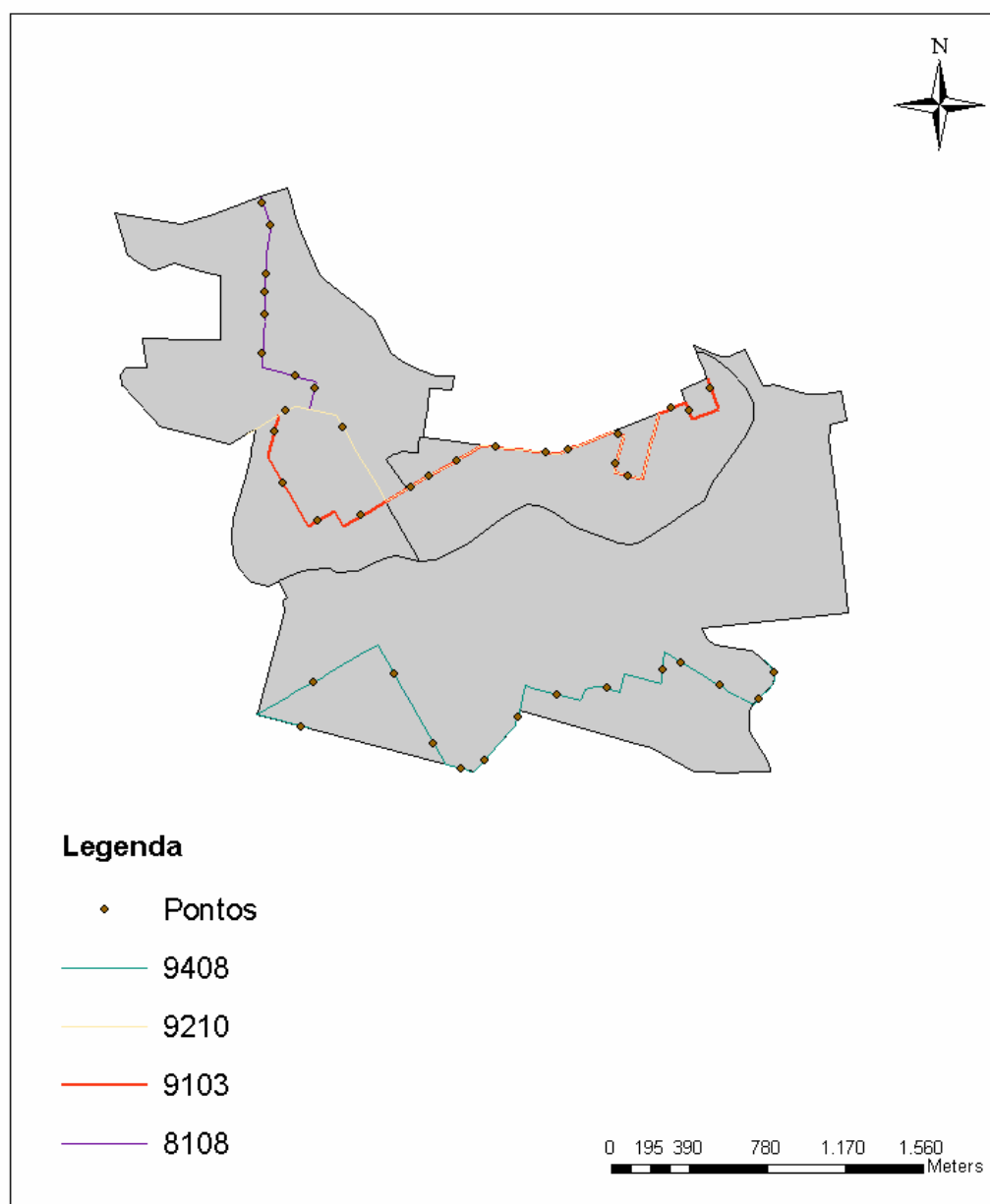


Figura 10 - Representação dos Dados Inseridos na Base de Dados

As linhas de ônibus selecionadas foram:

- 9103 – SANTA TEREZA/ SANTO ANTÔNIO (Em vermelho no mapa acima)
- 9102 – SANTA TEREZA/ PRADO (Em amarelo no mapa acima)
- 9408 – SANTA EFIGÊNIA/ PADRE EUSTÁQUIO (Em verde no mapa acima)
- 8108 – CIDADE NOVA (Em roxo no mapa acima)

Foram inseridos no mapa todos os pontos de ônibus que existem ao longo das linhas selecionadas (40 pontos). Não estão representados no mapa os itinerários completos das linhas de ônibus, apenas as partes que estão dentro dos limites dos bairros, e que serão objeto das consultas realizadas no banco de dados.

3ª etapa) Criação do Banco de Dados no PostGIS. Foram importados para o banco de dados PostGIS os arquivos em formato shape criados no ArcMap, sendo um arquivo para cada layer criado.

Foram criadas as seguintes tabelas no banco de dados: bairro (Tabela 1), linha (Tabela 2), trecho (Tabela 3), itinerario (Tabela 4) e ponto (Tabela 5).

<b>Tabela: bairro</b>			
<b>Descrição:</b> Tabela com os bairros da área de trabalho			
<b>Atributo</b>	<b>Formato</b>	<b>Tamanho</b>	<b>Descrição</b>
nubaipop	Int	4	Código que identifica o bairro
nobaipop	Varchar		Nome do bairro
the_geom	MULTIPOLYGON		Geometria que representa o bairro.

**Tabela 1 - Descrição da tabela *bairro***

<b>Tabela:</b> linha			
<b>Descrição:</b> Tabela com Linhas de ônibus			
<b>Atributo</b>	<b>Formato</b>	<b>Tamanho</b>	<b>Descrição</b>
id_linha	Int	8	Código único que identifica a linha
sigla_linha	Varchar		Número utilizado pelos usuários para identificação da linha.
desc_linha	Varchar		Identifica os bairros atendidos pela linha

**Tabela 2 - Descrição da tabela *linha***

<b>Tabela:</b> trecho			
<b>Descrição:</b> Tabela com os trechos de logradouro da área de trabalho			
<b>Atributo</b>	<b>Formato</b>	<b>Tamanho</b>	<b>Descrição</b>
nu_log	Int	4	Código que identifica o trecho de logradouro.
Nom_log	varchar		Nome do logradouro a que o trecho pertence.
the_geom	MULTILINESTRING		Geometria que representa o trecho.

**Tabela 3 - Descrição da tabela *trecho***

<b>Tabela:</b> itinerario			
<b>Descrição:</b> Tabela com os trechos de itinerário das linhas de ônibus			
<b>Atributo</b>	<b>Formato</b>	<b>Tamanho</b>	<b>Descrição</b>
nu_log	Int	8	Código que identifica o trecho de itinerario
nom_log	Varchar		Nome do logradouro a que o trecho pertence
id_linha	Int	8	Código que identifica a qual linha o trecho pertence
the_geom	MULTILINESTRING		Geometria que representa o trecho de itinerario.

**Tabela 4 - Descrição da tabela *itinerario***

<b>Tabela:</b> ponto			
<b>Descrição:</b> Tabela com os pontos de ônibus da área de trabalho			
<b>Atributo</b>	<b>Formato</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Int	4	Código que identifica o ponto de ônibus
descricao	varchar		Descrição do ponto de ônibus. (nome do trecho seguido de um número sequencial)
the_geom	POINT		Geometria que representa o ponto.

**Tabela 5 - Descrição da tabela *ponto***

4ª etapa) Consultas ao banco de dados, utilizando operadores espaciais. Os operadores testados foram: CONTAINS, TOUCHES, INTERSECTS e DISTANCE.

**1ª consulta:** Identificar os pontos de ônibus existentes no bairro de Santa Tereza

Comando SQL para PostGIS:

```
select p.id,p.descricao
from ponto p,bairro b
where b.nobaipop='SANTA TEREZA'
and CONTAINS(buffer(b.the_geom, 50), p.the_geom);
```

Resultado obtido:

*Id descrição*

- 1 "RUA SILVIANOPOLIS"
- 2 "RUA ALVINOPOLIS"
- 3 "RUA SALINAS\_1"
- 4 "RUA TENENTE VITORINO"
- 5 "RUA TENENTE DURVAL"
- 6 "RUA NORITA"
- 7 "RUA SALINAS\_2"
- 8 "RUA SALINAS\_3"
- 9 "RUA SALINAS\_4"

```

10 "RUA HERMILO ALVES"
11 "RUA HERMILO ALVES_2"
12 "RUA HERMILO ALVES_3"

```

Obs.: o comando buffer foi utilizado para que o resultado trouxesse também os pontos localizados nas ruas que fazem parte dos limites do bairro.

**2ª consulta:** Identificar as linhas de ônibus que passam por uma rua

Comando SQL para PostGIS:

```

select distinct l.desc_linha
from linha l, itinerario i, trecho t
where i.id_linha = l.id_linha
and t.nomlog = 'RUA NORITA'
and TOUCHES(t.the_geom, i.the_geom);

```

Resultado obtido:

*desc\_linha*

```

"SANTA TEREZA - PRADO"
"SANTA TEREZA - SANTO ANTONIO"

```

**3ª consulta:** Identificar as linhas de ônibus que passam por um determinado ponto de ônibus

Comando SQL para PostGIS:

```

select l.sigla_linha, l.desc_linha
from ponto p, itinerario i, linha l
where i.id_linha = l.id_linha
and p.id = 9
and INTERSECTS(buffer(p.the_geom, 10), i.the_geom);

```

Resultado obtido:

<i>sigla_linha</i>	<i>desc_linha</i>
"9103"	"SANTA TEREZA - SANTO ANTONIO"
"9210"	"SANTA TEREZA - PRADO"

**4ª consulta:** Identificar ruas próximas a um determinado ponto de ônibus. Neste caso, uma proximidade de 100m.

Comando SQL para PostGIS:

```
select distinct t.nomlog
from trecho t, ponto p
where p.id = 9
and (DISTANCE(p.the_geom, t.the_geom) < 100)
```

Resultado obtido:

*nomlog*  
"RUA GRAFITO"  
"RUA GABRO"  
"RUA MARMORE"  
"RUA SALINAS"  
"RUA HERMILO ALVES"

**5ª consulta:** Identificar pontos de ônibus próximos a uma determinada rua. Neste caso, a distância considerado entre o ponto de ônibus e a rua foi de 200m.



Comando SQL para PostGIS:

```
select p.id,p.descricao
from trecho t, ponto p
where t.nomlog = 'RUA NORITA'
and      (DISTANCE(t.the_geom, p.the_geom) < 200)
```

Resultado obtido:

<i>id</i>	<i>descricao</i>
3	"RUA SALINAS_1"
4	"RUA TENENTE VITORINO"
5	"RUA TENENTE DURVAL"
6	"RUA NORITA"
5	"RUA TENENTE DURVAL"
6	"RUA NORITA"

**6ª consulta:** Identificar pontos de ônibus por onde passam mais de uma linha.

Comando SQL para PostGIS:

```
select p.descricao
from ponto p, itinerario i, linha l
where  i.id_linha = l.id_linha
and INTERSECTS(buffer(p.the_geom, 10), i.the_geom)
group by p.descricao
having count(l.id_linha) > 1;
```

Resultado obtido:*descricao*

"RUA HERMILO ALVES"  
 "RUA TENENTE DURVAL"  
 "AVE DO CONTORNO\_2"  
 "RUA HERMILO ALVES\_2"  
 "RUA TENENTE VITORINO"  
 "RUA SALINAS\_3"  
 "RUA HERMILO ALVES\_3"  
 "RUA SALINAS\_4"  
 "RUA NORITA"  
 "AVE MEM DE SA\_1"  
 "RUA SALINAS\_2"

**7ª consulta:** Identificar todas as linhas que passam por um bairro

Comando SQL para PostGIS:

```
select distinct l.sigla_linha, l.desc_linha
from bairro b, itinerario i, linha l
where i.id_linha = l.id_linha
and b.nobaipop='SANTA TEREZA'
and CONTAINS(buffer(b.the_geom, 50), i.the_geom);
```

Resultado obtido:*sigla\_linha desc\_linha*

"9103" "SANTA TEREZA - SANTO ANTONIO"  
 "9210" "SANTA TEREZA - PRADO"

**8ª consulta:** Identificar quantidade de pontos em cada linha

Comando SQL para PostGIS:

```
select l.desc_linha, count(*) as quantidade
from ponto p, itinerario i, linha l
where i.id_linha = l.id_linha
and INTERSECTS(buffer(p.the_geom, 10), i.the_geom)
group by l.desc_linha;
```

Resultado obtido:

<i>desc_linha</i>	<i>quantidade</i>
"SANTA EFIGENIA"	15
"SANTA TEREZA - SANTO ANTONIO"	17
"CIDADE NOVA - FLORESTA"	9
"SANTA TEREZA - PRADO"	12

**9ª consulta:** Quantidade de Linhas que passam por cada ponto

Comando SQL para PostGIS:

```
select p.descricao, count(*) as quantidade
from ponto p, itinerario i
where INTERSECTS(buffer(p.the_geom, 5), i.the_geom)
group by p.descricao;
```

Resultado obtido:

descricao	quantidade
"RUA ALVINOPOLIS"	1
"RUA HERMILO ALVES"	2
"RUA MAESTRO DELE ANDRADE"	1
"AVE CARANDAI_1"	1
"RUA TENENTE DURVAL"	2
"RUA POUSO ALEGRE"	1
"AVE DO CONTORNO_2"	2
"RUA SAPUCAI"	1
"AVE PROFESSOR ALFREDO BALENA"	1
"AVE FRANCISCO SALES_2"	1
"RUA EUCLASIO_2"	1
"RUA JACUI_2"	1
"RUA MARECHAL DEODORO"	1
"RUA JACUI_4"	1
"RUA HERMILO ALVES_2"	2
"AVE CARANDAI_2"	1
"RUA TENENTE VITORINO"	2
"RUA JACUI_6"	1
"RUA SALINAS_3"	4
"RUA SALINAS_1"	1
"RUA HERMILO ALVES_3"	2
"RUA AQUILES LOBO"	1
"RUA JACUI_3"	1
"AVE DO CONTORNO_4"	1
"AVE MEM DE SA_2"	1
"AVE DO CONTORNO_1"	1
"RUA SILVIANOPOLIS"	1
"RUA JACUI_5"	1
"AVE FRANCISCO SALES_1"	1
"RUA SALINAS_4"	2
"RUA NORITA"	2
"RUA RESEDA"	1
"AVE MEM DE SA_1"	2
"AVE DO CONTORNO_3"	1
"RUA JACUI_1"	1
"RUA EUCLASIO_1"	1
"RUA ITAJUBA"	1
"RUA SALINAS_2"	2
"RUA TENENTE GARRO"	1
"AVE FRANCISCO SALES_3"	1

## 5ª etapa) Visualização de resultados utilizando MapServer

Para mostrar visualmente o resultado de algumas consultas no MapServer, foi preciso criar um arquivo texto com extensão .map, de acordo com o formato descrito no site do MapServer (<http://mapserver.gis.umn.edu/>). Quando executado, esse arquivo busca no PostGis os dados das consultas solicitadas.

O código abaixo é do arquivo do arquivo SistTransporte.map criado, e que contém um cabeçalho com informações sobre as dimensões do mapa gerado e as coordenadas do retângulo envolvente da área estudada, dentre outras informações de identificação do mapa que será visualizado. Além disso, contém a descrição de cada layer que será desenhado. Para cada layer há um comando SQL que fará busca no PostGIS dos dados espaciais que serão plotados no mapa. Os comandos SQL tem que vir embutidos no código, não existe ainda uma interface gráfica para construção destes comandos no MapServer. No caso do código descrito, serão buscados os bairros, os pontos e cada itinerário de linha como um layer diferente.

```

NAME "SISTEMA DE TRANSPORTE"
IMAGETYPE      PNG
EXTENT         610771 7795856 615059 7799067
SIZE          500 500
IMAGECOLOR    250 250 250
UNITS         METERS
SYMBOLSET     "/var/www/html/fonts/symbols.sym"

LAYER
  NAME "bairro"
  CONNECTIONTYPE postgis
  CONNECTION "user=postgres dbname=marcela host=127.0.0.1"
  DATA "the_geom from bairro"
  STATUS DEFAULT
  TYPE POLYGON
  CLASS
    STYLE
      COLOR 166 166 166
      OUTLINECOLOR 255 255 255
    END
  END
END

LAYER
  NAME "9103"

```

```
CONNECTIONTYPE postgis
CONNECTION "user=postgres dbname=marcela host=127.0.0.1"
DATA "the_geom from (select a.the_geom as the_geom, b.oid as oid from
itinerario a, itinerario b where a.id_linha = 1)as linha9103"
STATUS DEFAULT
TYPE LINE
CLASS
STYLE
    COLOR 220 22 12
END
END
END
```

```
LAYER
NAME "9210"
CONNECTIONTYPE postgis
CONNECTION "user=postgres dbname=marcela host=127.0.0.1"
DATA "the_geom from (select a.the_geom as the_geom, b.oid as oid from
itinerario a, itinerario b where a.id_linha = 4)as linha9210"
STATUS DEFAULT
TYPE LINE
CLASS
STYLE
    COLOR 236 243 41
END
END
END
```

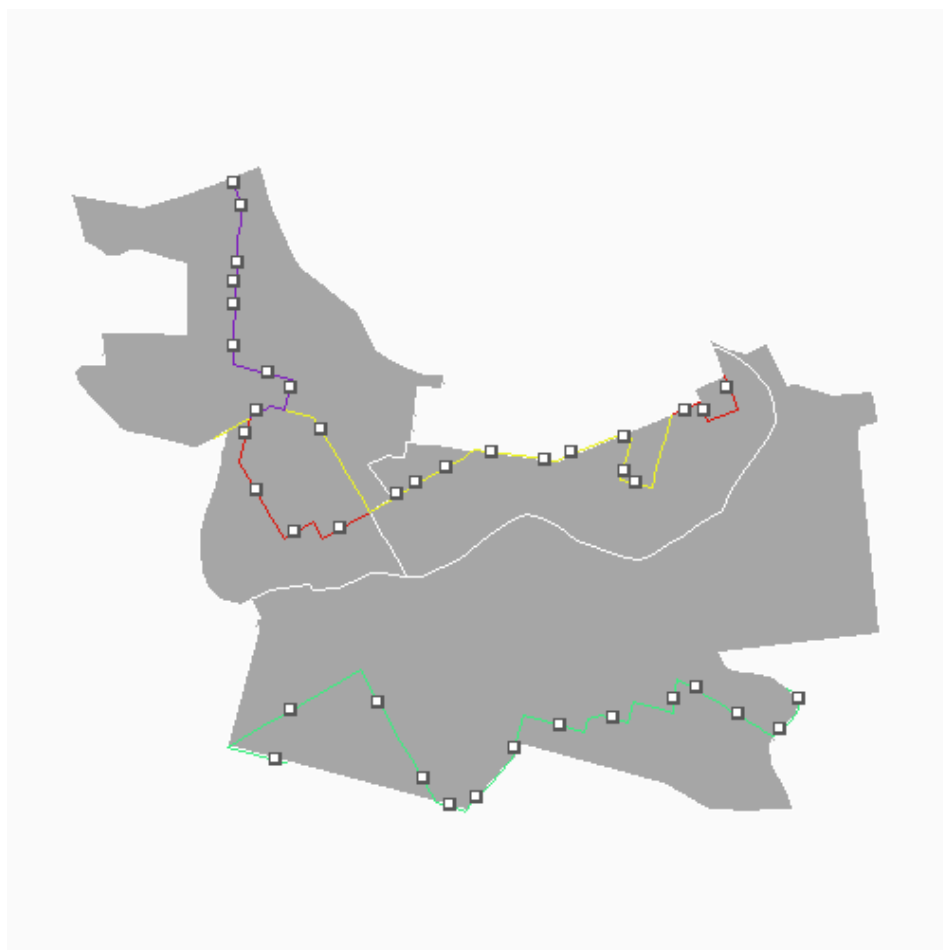
```
LAYER
NAME "9408"
CONNECTIONTYPE postgis
CONNECTION "user=postgres dbname=marcela host=127.0.0.1"
DATA "the_geom from (select a.the_geom as the_geom, b.oid as oid from
itinerario a, itinerario b where a.id_linha = 2)as linha9408"
STATUS DEFAULT
TYPE LINE
CLASS
STYLE
    COLOR 62 237 128
END
END
END
```

```
LAYER
NAME "8108"
CONNECTIONTYPE postgis
CONNECTION "user=postgres dbname=marcela host=127.0.0.1"
DATA "the_geom from (select a.the_geom as the_geom, b.oid as oid from
itinerario a, itinerario b where a.id_linha = 3)as linha8108"
STATUS DEFAULT
TYPE LINE
CLASS
STYLE
    COLOR 126 32 189
END
END
```

```
END

LAYER
  NAME "ponto"
  CONNECTIONTYPE postgis
  CONNECTION "user=postgres dbname=marcela host=127.0.0.1"
  DATA "the_geom from ponto"
  STATUS DEFAULT
  TYPE POINT
  CLASS
    STYLE
      SYMBOL 'localidade2'
      COLOR 237 62 231
      SIZE 7
    END
  END
END
END
```

Utilizando esse arquivo o resultado é apresentado na figura 11:



**Figura 11 - Mapa para visualização de bairros, pontos e itinerário das linhas de ônibus cadastrados na base de dados**

Para executar a consulta 1 (pontos de ônibus localizados no bairro de Santa Tereza), foi acrescentado ao arquivo “SistTransporte.map” o layer da consulta 1 (descrito abaixo). Para melhor visualização do resultado, foi retirado do código os layers de pontos e linhas de ônibus.

```
LAYER
  NAME "consulta1"
  CONNECTIONTYPE postgis
  CONNECTION "user=postgres dbname=marcela host=127.0.0.1"
  DATA "the_geom from (select x.the_geom as the_geom, y.oid as oid from
ponto x,ponto y, bairro b where b.nobaipop='SANTA TEREZA' and
CONTAINS(buffer(b.the_geom, 50), x.the_geom)) as consulta1"
  STATUS DEFAULT
  TYPE POINT
  CLASS
  STYLE
    COLOR 255 0 0
    SYMBOL 'localidade2'
    SIZE 7
  END
END
END
```

O resultado da consulta 1 está representado no Figura 12:



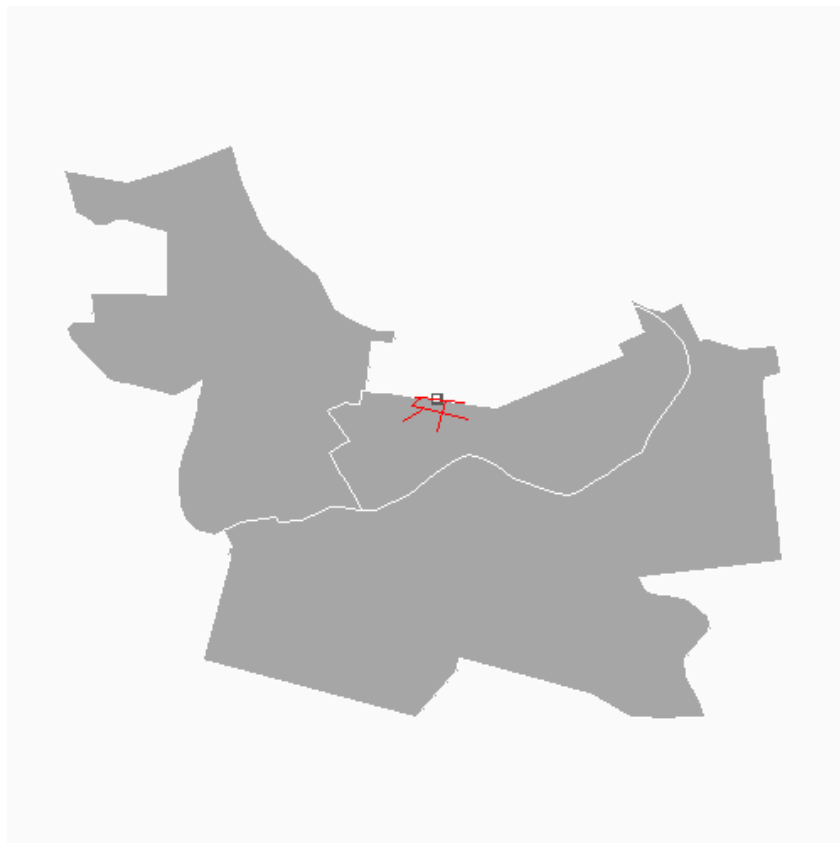


**Figura 12 - Representação visual do resultado da consulta 1**

Para executar a consulta 4 (ruas próximas a um determinado ponto de ônibus), foi acrescentado ao arquivo “SistTransporte.map” o layer da consulta 4 (descrito abaixo). Para melhor visualização do resultado.

```
LAYER
  NAME "consulta4"
  CONNECTIONTYPE postgis
  CONNECTION "user=postgres dbname=marcela host=127.0.0.1"
  DATA "the_geom from (select x.the_geom as the_geom, y.oid as oid from
trecho x,trecho y, ponto p
where p.id = 9 and (DISTANCE(p.the_geom, x.the_geom) < 100)) as
consulta4"
  STATUS DEFAULT
  TYPE LINE
  CLASS
  STYLE
    COLOR 255 0 0
  END
END
END
```

O resultado da consulta 4 está representado na figura 13:



**Figura 13 - Representação visual do resultado da consulta 4**

## 4- Considerações Finais

O estudo realizado atendeu aos objetivos propostos, já que foi possível criar um modelo direcionado para transporte público, implementá-lo no PostGIS, fazer consultas espaciais sobre os dados e visualizar os resultados de algumas consultas num browser.

Criar um modelo para tratamento de dados espaciais orientado a objetos e implementá-lo em um banco de dados objeto-relacional mostrou ser uma combinação eficaz no que diz respeito à compreensão do problema a ser estudado, e com isso a maior facilidade para fazer consultas e manipular dados.

O banco de dados objeto-relacional PostGIS mostrou-se eficaz nas consultas espaciais que foram propostas, além de ter mostrado que a sintaxe dos comandos SQL é relativamente simples. Não foi possível avaliar o desempenho do banco, mesmo observando que o resultado das consultas foi fornecido imediatamente. Só seria possível fazer uma avaliação mais realista do desempenho utilizando um volume maior de dados e uma plataforma multiusuário.

Através do MapServer foi possível visualizar o resultado das consultas propostas neste trabalho. Não foi complicado utilizar este aplicativo escrevendo os “*map files*”, entretanto para que ele seja mais interativo com os usuários, seria interessante ter uma ferramenta gráfica para construção dos comandos SQL.

No caso de transportes públicos, existem muitas aplicações para consultas espaciais, e já se observa hoje uma forte tendência por parte dos órgãos gestores de transporte na utilização de sistemas de informações georreferenciadas no controle diário de suas operações e tomada de decisões. A implantação de SIGs nas empresas muitas vezes esbarra na necessidade de uma base cartográfica georrefenciada com os dados da cidade, o que nem sempre existe. A formação dessa base pode ser um processo demorado, pois são bases grandes com forte necessidade de integridade entre os dados. Entretanto é um trabalho recompensado pelas vantagens da utilização dos SIGs, que, além de melhorar os processos da empresa, são indicação de qualidade.

## Referências Bibliográficas

- ANTP - <http://www.antp.org.br/> - Acesso em 10 de novembro de 2005.
- BHTRANS - <http://www.bhtrans.pbh.gov.br/bhtrans/index.asp>. Acesso em 01 de outubro de 2005.
- BORGES, Karla Albuquerque de Vasconcelos. **Modelagem de Dados Geográficos**. Apostila do curso de especialização de Geoprocessamento. 2005.
- BORGES, K.A.V.; DAVIS JR., C.A.; LAENDER, A.H.F. **Modelagem Conceitual de Dados Geográficos**. Capítulo 3 do livro Bancos de Dados Geográficos. Disponível em [www.dpi.inpe.br/livros/bdados](http://www.dpi.inpe.br/livros/bdados). Acesso em 01 de outubro de 2005.
- CAMARA, G.; CASANOVA, M.; HEMERLY, A.; MAGALHÃES, G. e MEDEIROS, C.; **Anatomia de Sistemas de Informação Geográfica**. Campinas: Instituto de Computação, UNICAMP, 1996.
- CARVALHO, A.; **O Geoprocessamento Como Recurso Para Análise das Ocorrências em Linhas de Ônibus em Belo Horizonte – Um Estudo de Caso**. Monografia Curso de Especialização em Geoprocessamento. 2002
- ELMASRI, R.; NAVATHE, S. **Sistemas de Banco de Dados. Fundamentos e Aplicações**. Ed. LTC, 2002.
- FERRARI, G.V.; SOUZA, C.L.; SOUZA, M.A.; **Mapa Interativo de Santa Catarina na Web: Servidor de Informações Georreferenciadas baseado em Software Livre**. Disponível em [www.mapainterativo.ciasc.gov.br/ArtigoMapaInterativo.doc](http://www.mapainterativo.ciasc.gov.br/ArtigoMapaInterativo.doc). Acesso em 12 de dezembro de 2005.
- FERRAZ, A.C.P.; TORRES, I.G.E.T. **Transporte Público Urbano**. Ed.RiMa, 2001.
- FERREIRA, E.A., **Um Método de Utilização de Dados de Pesquisa Embarque/Desembarque na Calibração de Modelos de Distribuição do Tipo Gravitacional**. Dissertação de Mestrado em Engenharia de Transportes. 1999.

FERREIRA, K.R.; QUEIROZ, G.R, PAIVA J.A; SOUZA, R.C.M; CÂMARA, G. **Arquitetura de Software Para Construção de Bancos de Dados Geográficos com SGBD Objeto-Relacionais**. 2002

JANUARIO, M.H. **Procedimentos Para Determinação de Índices de Acessibilidade de Transportes e Tratamento Cartográfico dos Mesmos**. (Dissertação de Mestrado). Rio de Janeiro: Instituto Militar de Engenharia, 1995.

LISBOA, J.; COSTA, A.C., IOCHPE, C. **Projeto de Banco de Dados Geográficos: mapeando esquemas *GeoFrame* para o SIG *Spring***. 2001.

MAPSERVER - <http://mapserver.gis.umn.edu/>. Acesso em 10 de dezembro de 2005.

MIRANDA, J.I.; **Publicando Mapas na WEB: Servlets, Applets ou CGI?**. Embrapa. 2003

OLIVEIRA, R. A. **Desenvolvendo Aplicações Livres de Modelagem Objeto/Relacional com Banco de Dados Orientado a Objetos**.  
Julho, 2005.

OLIVEIRA, S.P. **Implementação de Roteirização para a Malha Rodoviária do Estado de Minas Gerais**. Monografia Curso de Especialização em Geoprocessamento.2000

PRADO, V.L. **Estudo de Acessibilidade de Transportes no Município de Passos – MG**. Monografia Curso de Especialização em Geoprocessamento.2001.

QUEIROZ, G.R.; FERREIRA, K.R. **SGBD com extensões espaciais**. Capítulo 8 do livro Bancos de Dados Geográficos. Disponível em [www.dpi.inpe.br/livros/bdados](http://www.dpi.inpe.br/livros/bdados). Acesso em 01 de outubro de 2005.

RODRIGUES, A.R., **Radar: Cadastramento e Disponibilização Via internet**. Monografia Curso de Especialização em Geoprocessamento.2001

<http://www.javainux.com.br:8080/pg74/datatype-geometric.html>. Acesso em 11 de dezembro de 2005

