

Utilizando-se Tecnologias Fontes Abertas para Projetar Web Mapas Estáticos Interativos

Nelson Marisco, *UFMS Campo Gande*

Jürgen Philips, *UFSC Florianópolis*

Humberto Rafael Pereira, *UFMS Campo Gande*

Conteúdo [esconder]

1. Introdução
2. Os Web Mapas
 - 2.1 Fatores influenciadores em projetos de Web mapas
3. Padrão vetorial para Web “Scalable Vector Graphics” (SVG)
4. Projetando o protótipo do Web Mapa
 - 4.1 Estabelecendo-se o modelo digital de superfície
 - 4.2 Extraíndo-se os dados do banco de dados
 - 4.3 Definindo-se o modelo digital cartográfico
5. Fluxograma do Trabalho
6. Análise e Resultados
 - 6.1 Fluxograma do Trabalho
 - 6.2 As interfaces gráficas com os usuários
7. Conclusões e Recomendações
 - 7.1 Conclusões
 - 7.2 Recomendações
8. Referências bibliográficas
9. URL

► **Resumo**

1. Introdução

A cartografia, historicamente, tem disponibilizado instrumentos adequados que atendem de forma satisfatória a representação dos dados geoespaciais. Os processos de revolução tecnológica, ora em curso, desponta aos cartógrafos, novos desafios, isto é, buscar novos aspectos e métodos de comunicação e de exploração desses dados. Evolutivamente, como uma nova tendência dentro da cartografia, a representação dos dados geoespaciais pode ser realizada por intermédio de produtos cartográficos veiculados através de novos meios de comunicação como a Internet, na qual destaca-se a World Wide Web (WWW) (KRAAK, 2001a p. 01) e (PETERSON, 2001).

Dentre as novas mídias emergentes, a Internet passa a ser um excelente meio veiculador e disseminador de informações, inclusive para dados geoespaciais. A Internet reúne através da World Wide Web, o som disponibilizado pelo rádio, a imagem proporcionada pela televisão, a escrita proporcionada pela imprensa e acrescentando a tudo isso, a possibilidade de interação com os usuários, caracterizando-a como multimídia. O caráter multimídia da World

Wide Web desponta novos desafios à cartografia e, principalmente, aos cartógrafos, como por exemplo:

- como construir e disponibilizar produtos cartográficos de forma eficiente e a um baixo custo, que venham atender às exigências e limitações dessas novas mídias emergentes?
- como possibilitar ao usuário construir seu próprio mapa mantendo os preceitos de um bom projeto cartográfico?

Essas são algumas das indagações a serem enfrentadas e, para uma adequada resposta, deve-se pensar o mapa ora como instrumento de exploração, ora como de comunicação dos dados geoespaciais. Essa nova forma de pensar o mapa é definida como processos de visualização cartográfica ou geovisualização.

Portanto, a Geovisualização “é uma integração da visualização científica, Cartografia exploratória, análise de imagens, visualização da informação, análise de dados exploratória (EDA) e Sistemas de Informação Geográfica (SIG) para produzir teorias, métodos e ferramentas para a exploração visual, análise, síntese e apresentação de dados geoespaciais” (MacEACHREN e KRAAK, 2001).

Dentro desse contexto, de revolução tecnológica e de evolução no processo de representação cartográfica, aumentam-se as demandas para explorar as possibilidades até então indisponível para o mapeamento em mídia papel, isto é, a animação, a interação, a escalabilidade e a generalização, utilizando-se novas tecnologias de códigos fontes abertas disponíveis para Web, aplicando-as ao processo de disseminação de informações relacionadas ao cadastro imobiliário urbano municipal pela Internet.

Dessa forma, propõe-se neste trabalho explorar as potencialidades das novas tecnologias fontes abertas para Web de forma que, com os resultados a serem alcançados, possam vir a atender às crescentes demandas por diminuição de custos em ações da gestão pública e a necessidade de uma maior democratização no processo de disseminação das informações relacionadas ao cadastro imobiliário urbano municipal, permitindo a diferentes usuários, com diferentes necessidades e interesses, acessar informações do tipo: tamanho da área do imóvel, valor, localização, características topográficas e pedológicas, tipo de zoneamento a que está sujeita a referida parcela, data da última transação com o imóvel, dentre outras informações, por meio de Web mapas interativos e/ou dinâmicos, dispostos pela Internet e/ou Intranet.

A adoção dessas novas tecnologias na gestão pública poderá levar a uma diminuição significativa em gastos com programas de computadores e equipamentos de informática, permitindo, com isso, maior agilização em processos que envolvam transações imobiliárias e maior participação popular nos processos de planejamento urbano.

Portanto, a hipótese deste trabalho centra-se na assertiva de que é possível utilizar fontes abertas XML, SVG, DOM, JavaScript e bancos de dados PostgreSQL em projetos de Web mapas interativos, com o intuito de disseminar informações concernentes ao cadastro imobiliário urbano municipal pela Internet, sem a necessidade de instalar nenhum outro tipo de programa no navegador a não ser o Plug-in Adobe SVG Viewer 3.0.

Para validar a hipótese desta pesquisa, utilizou-se informações concernentes a uma área piloto, com informações sobre as propriedades (lotes e edificações) e os tributos (planta de valores genéricos). Essas informações serão armazenadas em um servidor de banco de dados com característica “Open Source”. Não obstante, os Web mapas aqui pensados devem explorar de forma eficaz as potencialidades disponíveis na Web, sem perder de vista as reais limitações impostas pelo meio (tamanho do arquivo e área de apresentação).

Dos objetivos a serem alcançados na pesquisa destacam-se:

- Analisar a utilização das novas tecnologias disponíveis para Web em códigos fontes abertas, dentro do contexto da geovisualização.

- Investigar os diferentes formatos e modelos de dados que possam ser utilizados na construção de mapas interativos para Web.
- Desenvolver um protótipo de um Web mapa que seja uma interface de comunicação homem-máquina, utilizando os códigos fontes abertas.
- Fornecer uma modesta contribuição ao processo de ampliação e de popularização ao acesso às informações concernentes ao cadastro imobiliário urbano municipal no âmbito dos órgãos municipais e aos munícipes, no intuito de contribuir com o processo de modernização da gestão pública.

2. Os Web Mapas

Os Web mapas podem disponibilizar às pessoas usuárias a possibilidade de acessibilidade às informações geográficas de modo interativo, dinâmico e constantemente atualizado. Simples recursos como símbolos gráficos em um produto cartográfico interativo podem se tornar objetos ativos, clicáveis e recuperáveis de um servidor de banco de dados. As legendas podem ser apresentadas em janelas secundárias “pop-up”, as cores podem ser alteradas, objetos em movimentos podem ser disponibilizados, diferentes níveis de informação em diferentes escalas podem ser apresentados por meio de funções Zoom (in-out).

Os Web mapas passam, então, a ser a interface entre os usuários e os dados geoespaciais armazenados em um ou vários servidores de banco de dados, possibilitando uma interação homem-máquina. A interação poderá ser ampla ou restritiva, dependendo do tipo de informação a ser disponibilizada ao usuário e das suas funcionalidades interativas.

Os tipos de Web mapas mais encontrados na Internet são definidos por Kraak e Brown (2001) como sendo mapas estáticos não interativos, com finalidade apenas para visualizar; esses mapas são oriundos de produtos cartográficos em formato papel, convertidos em meio digital por meio da utilização de “scanners”, portanto não foram construídos para serem apresentados na Web e não exploram totalmente suas funcionalidades. Logo, o mais adequado é explorar as novas tecnologias para a Internet que permitam projetar Web mapas estáticos ou dinâmicos com interfaces e/ou conteúdos interativos (Figura 01).



Figura 1 : Características dos Web mapas após Kraak e Brown (2001, p.171)

2.1 Fatores influenciadores em projetos de Web mapas

Kraak (2001b, p.05) destaca que os produtos cartográficos na Web são influenciados por quatro grandes atores (Figura 02):

- os usuários;
- os provedores de dados;
- o ambiente de concepção dos produtos cartográficos, e
- o conteúdo dos produtos cartográficos.

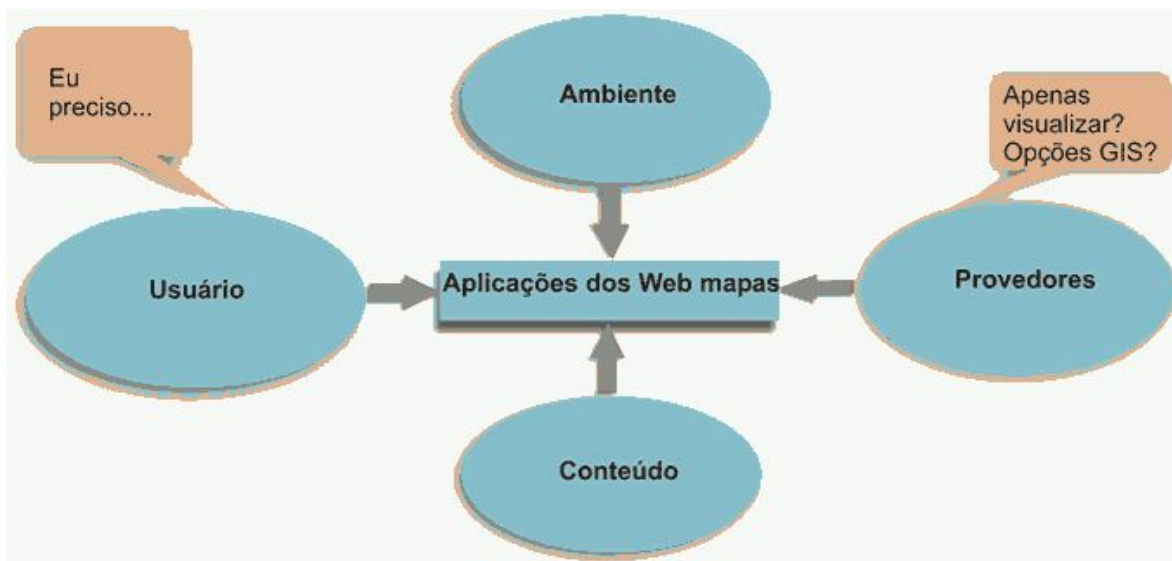


Figura 2 : Os quatros grandes atores influenciadores na construção de Web mapas.

Fonte: Kraak (2001b,p.05).

Dentro dos fatores influenciadores, destacam-se ainda:

A tradução gráfica: procura traduzir os diferentes fenômenos a serem representados através de símbolos pontuais, lineares e áreas dispondo-os espacialmente da melhor forma possível sobre o mapa, procurando uma melhor forma de comunicar as informações aos usuários.

A interface com o usuário: formada por apresentações de informações, de dados, de controles e de comandos. Ela, também, solicita e recebe as entradas de dados, de controles e de comandos (SHNEIDERMAN, 1997). Uma boa interface não levanta dúvida ou desorienta o usuário. Este, basicamente, interage com um produto cartográfico através do mouse, movendo-o ou clicando-o sobre os objetos ativos e digitando informações via teclado. Essa interação é induzida pela interface gráfica com o usuário (GUI).

O Armazenamento dos dados geoespaciais: considerando que a Internet, utiliza o paradigma computacional cliente-servidor, o que possibilita ao cliente interessado em informações geoespaciais, acessar um servidor ou vários servidores interconectados. O armazenamento dos dados passa ser de significativa importância.

3. Padrão vetorial para Web “Scalable Vector Graphics” (SVG)

Atualmente o SVG é o único a ser recomendado oficialmente como padrão pelo W3C, o que reforça a sua aplicação no desenvolvimento deste trabalho.

O SVG é um formato gráfico baseado na linguagem XML, tornando-o legível para o ser humano, facilitando sua manutenção e alteração. Os arquivos em SVG iniciam indicando a versão e uma ligação para uma DTD (Document Type Definition) fornecida pelo consórcio W3C, geralmente sendo chamados de prólogo.

O SVG proporciona:

- os diferentes tipos de transformações (translação, rotação, escala);
- a definição dos elementos geométricos: linhas retas (polyline), curvas, polígonos, linhas, círculos e elipses;
- objetos podem ser recortados (clipping), mascarados (masking) e compostos;
- os objetos podem ser coloridos, preenchidos, coloridos obedecendo a um gradiente, transparentes, marcados, e vários tipos de traços podem ser definidos (stroke-types);
- diferentes símbolos e textos podem ser construídos e utilizados;
- possibilidade de animação;
- possibilidade de interação, e com o auxílio dos CGI scripts, ou Java applets e/ou Java servlets, bancos de dados podem ser ligados;
- possibilidade de anexar atributos dos dados, informações sobre os dados (metadados) e extensibilidade (MathML, SMIL, XHTML).

Esses recursos disponibilizados no SVG são de fundamental importância para o processo de comunicação e exploração dos dados geoespaciais armazenados em bancos de dados através dos Web mapas. O arquivo SVG pode conter uma ou várias portas de visualização (viewport), sendo possível especificar um sistema de coordenadas relativas à área de visualização do SVG, (viewport coordinate system) e um sistema de coordenadas utilizado pelo usuário. Esses sistemas estão representados por (width) e (height) – esses atributos especificam o tamanho do mapa no papel e o atributo relacionado ao (viewbox) representa o sistema de coordenadas do usuário. Como a origem dos sistemas de coordenadas do SVG é definida no canto superior esquerdo da tela de vídeo do computador, para relacioná-lo ao sistema de coordenadas UTM (E, N) há a necessidade de se utilizar o sinal de negativo (-) nas coordenadas ao longo do eixo y (N), ou através de uma transformação de coordenadas utilizando o comando transform= “scale (1,-1)”. Essa segunda opção pode acarretar em dificuldades no momento da utilização dos elementos “use” e “text”.

Folhas de estilos podem ser definidas internamente ou externamente ao SVG para permitir a sua visualização. Os elementos gráficos possuem um único identificador o que possibilita o acesso a mudanças em seus atributos utilizando-se *scripts* (EISENBERG, 2002). Os elementos podem ser definidos e instanciados posteriormente várias vezes. E isso é de fundamental importância na definição de símbolos cartográficos para a Web, dando-se preferência aos marcadores, a fim de manter arquivos de tamanhos pequenos e também para eficientemente trocá-los, quando solicitados.

4. Projetando o protótipo do Web Mapa

Para a construção do protótipo do Web Mapa alguns requisitos foram adotados, dos quais destacam-se:

- Os usuários;
- Os dados cartográficos;
- O sistema de gerenciamento de banco de dados;
- O modelo Cliente/servidor a ser adotado, e
- A interface de comunicação com o usuário.

Considerado todos esses elementos e mais o fato de se estar propondo trabalhar com tecnologias fontes abertas para a Web, estabeleceu-se a seguinte configuração ao sistema computacional a ser utilizado (Tabela 01).

Sistemas e programas de computadores	Configurações
Hardware	AMD Athlon 1100 Mhz
Sistema operacional	MS Windows 2000 (futuramente testar no GNU-LINUX)
SGBD e extensões	PostgreSQL 7.3.3 (Cygwin) e PostGIS 0.8.0 (extensão espacial para o PostgreSQL)
Administrador do PostgreSQL para ambiente Windows	PgAdmII 1.6.0, PgAdm III 1.0.0 e DBTools 2.1
Servidor Web	Apache 2.0.47
Apresentação no cliente (scripts e plug-in)	ECMAScript/JavaScript e Adobe SVG viewer (versão 3.0) para sistema Windows.
Linguagem de programação do lado do Servidor (módulo)	PHP 4.3.4 e Perl 5.8.0
Browser (para testar a apresentação)	MS Internet Explore 5.0 – 5.5 - 6.0, Mozilla NS 1.4 e 1.5 e Opera 7.0 – 7.1 - 7.2 e Netscape Communicator 4.8

A implementação cartográfica objetiva disponibilizar, de forma interativa e dinâmica, via Internet, informações concernentes ao cadastro imobiliário urbano aos usuários em geral. As informações devem abranger os aspectos geométricos e fiscais do cadastro imobiliário com o intuito de atender a um maior espectro possível de usuários, futuramente aspectos legais deveram ser incorporados.

Na Web, são poucos os sites que disponibilizam informações sobre cadastro imobiliário. Com o intuito de aproveitar as potencialidades das novas tecnologias emergentes de fontes abertas, utiliza-se neste trabalho o paradigma cliente/servidor de três-camadas, tendo-se do lado do cliente a utilização das linguagens SVG, HTML e ECMAScript e, do lado do servidor, linguagens PHP e Perl em servidor Apache. Como servidor de banco de dados, utiliza-se o PostgreSQL. O esquema apresentado na Figura 03 procura representar a arquitetura Cliente/Servidor em três camadas a ser adotada.

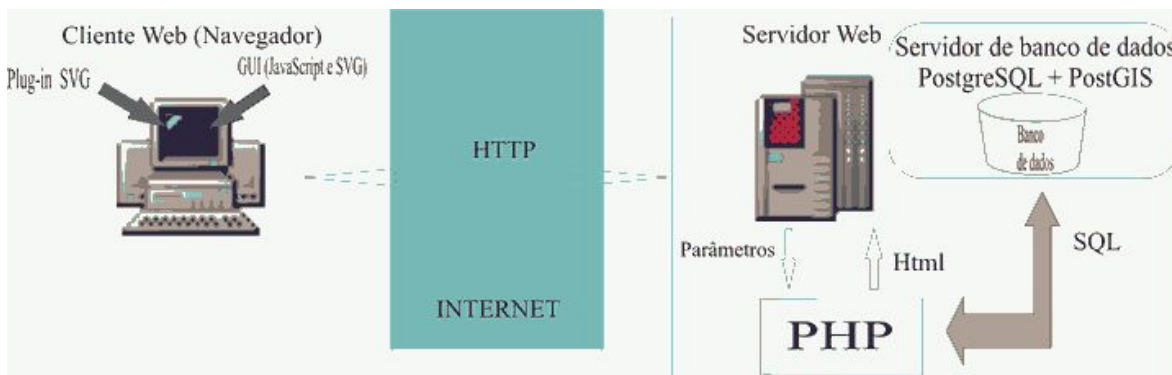


Figura 3 : Esquema da arquitetura Cliente/Servidor em três camadas a ser adotada.

4.1 Estabelecendo-se o modelo digital de superfície

Para estabelecer o modelo digital de superfície armazena-se os dados gráficos no PostgreSQL através da utilização da extensão PostGIS. Para isso, necessitou-se converter as bases cartográficas em formatos DGN para SHAPEFILE (.shp), uma vez que o PostGIS possui uma ferramenta, denominada de “shp2pgsql”, que permite criar declaração em SQL que pode ser utilizada para ler individualmente as diferentes camadas do arquivo shapefile, armazenando-as no banco de dados pré-definido. A tabela gerada contém geometrias baseadas em texto em conformidade com os padrões estabelecidos pelo “Open Gis Simple Features Specification for SQL”. Esse procedimento é efetivado utilizando-se o seguinte comando no Cygwin/PostgreSQL:

```
shp2pgsql lotes.shp lotes bdsj > lotes.sql
```

Esse comando pode ser traduzido da seguinte forma: converter o arquivo lotes.shp dentro de uma tabela chamada lotes no banco de dados bdsj, para gerar uma declaração SQL salvando-a em um arquivo denominado lotes.sql, resultando na declaração seguinte:

```
create table lotes (gid int4, ID int8, LAYER varchar, COLOR int4, LINETYPE varchar,
LINEWIDTH float8, ELEVATION float8);
select AddGeometryColumn('bdsj','lotes','the_geom','-1','MULTIPOLYGON',2);
begin;
Insert          into          lotes          values('0','1','Level
5','6','Continuous','0.0000','0.000000',GeometryFromText('MULTIPOLYGON(((736226.380000001
6944736.76 ,736226.930000001 6944728.08 ,736222.380000001 6944727.78 ,736222.070000001
6944732.37 ,736223.390000001 6944732.46 ,736223.110000001 6944736.55 ,736226.380000001
6944736.76 )))',-1) );
5','1','Continuous','0.0000','0.000000',GeometryFromText('MULTIPOLYGON(((736102.580000001
6945758.84 ,736110.440000001 6945767.79 ,736111.240000001 6945767.2 ,736115.200000001
6945764.22 ,736108.820000001 6945753.92 ,736102.580000001 6945758.84 )))',-1) );
```

O passo seguinte é construir a tabela no banco de dados. Para isso, utiliza-se o seguinte comando no Cygwin/PostgreSQL:

```
psql -U postgres -d bdsj -f lotes.sql
```

Esse comando possibilita a conexão ao banco de dados, utilizando o usuário padrão postgres, executando o arquivo lotes.sql, gerando a tabela lotes no PostgreSQL (Figura 04).

Com o intuito de diminuir o tempo de computação do lado do servidor, na hora da apresentação das diferentes camadas solicitadas pelo cliente, Neumann (2003) implementa um script em Perl que lê a coluna “the_geom” contendo as geometrias no padrão Open-GIS e

converte-as em SVG “dados path”. Assim, uma outra coluna denominada “svg_geom” é criada, armazenando-os na tabela. A desvantagem desse procedimento é que, caso o cliente deseje ver apenas uma porção de uma camada no viewBox, isso não será possível, uma vez que toda a camada é enviada para o cliente; entretanto, sua vantagem está na diminuição de tempo de computação do lado do servidor. Para todas as camadas do mapa, armazenadas no banco de dados, adotou-se o mesmo procedimento. A seta indicativa na Figura 04, destaca a coluna svg_geom na tabela lotes no PostgreSQL. Uma segunda questão a ser resolvida é que esse script em perl só lida com as especificações geométricas Open-GIS MULTIPOLYGONS e MULTILINESTRINGS, necessitando, em trabalhos futuros, estender às especificações restantes.

	oid	gid int4	id int8	layer varchar	color int4	linetype varchar	linewidth float8	elevation float8	the_geom geometry	svg_geom varchar	cod_lotes varchar
1	724010	0	1	Level 5	6	Continuous	0	0	SRID=-1,MUI M736226.38 - 1-4-2-EE		
2	724011	1	2	Level 5	6	Continuous	0	0	SRID=-1,MUI M735938 -69- 1-4-3-DD		
3	724012	2	3	Level 5	1	Continuous	0	0	SRID=-1,MUI M736102.58 - 1-4-98-174		
4	724013	3	4	Level 5	6	Continuous	0	0	SRID=-1,MUI M736192.35 - 1-4-2-B		
5	724014	4	5	Level 5	6	Continuous	0	0	SRID=-1,MUI M736209.82 - 1-4-2-C		
6	724015	5	6	Level 5	6	Continuous	0	0	SRID=-1,MUI M736218.09 - 1-4-2-D		
7	724016	6	7	Level 5	6	Continuous	0	0	SRID=-1,MUI M736223.19 - 1-4-2-222		
8	724017	7	8	Level 5	6	Continuous	0	0	SRID=-1,MUI M736228.44 - 1-4-2-A		
9	724018	8	9	Level 5	6	Continuous	0	0	SRID=-1,MUI M735879.4-6 1-4-186-79		
10	724019	9	10	Level 5	6	Continuous	0	0	SRID=-1,MUI M735885.44 - 1-4-186-94		
11	724020	10	11	Level 5	6	Continuous	0	0	SRID=-1,MUI M736166.44 - 1-4-3-394		
12	724021	11	12	Level 5	6	Continuous	0	0	SRID=-1,MUI M736180.95 - 1-4-2-36		
13	724022	12	13	Level 5	6	Continuous	0	0	SRID=-1,MUI M736282.51 - 1-4-28-194		
14	724023	13	14	Level 5	6	Continuous	0	0	SRID=-1,MUI M735879.68 - 1-4-40-402		
15	724024	14	15	Level 5	6	Continuous	0	0	SRID=-1,MUI M735873.31 - 1-4-40-411		
16	724025	15	16	Level 5	6	Continuous	0	0	SRID=-1,MUI M736273.22 - 1-4-19-20		
17	724026	16	17	Level 5	6	Continuous	0	0	SRID=-1,MUI M736291.97 - 1-4-28-194		
18	724027	17	18	Level 5	6	Continuous	0	0	SRID=-1,MUI M736409.56 - 1-4-5-211		
19	724028	18	19	Level 5	6	Continuous	0	0	SRID=-1,MUI M736113.41 - 1-4-3-310		
20	724029	19	20	Level 5	6	Continuous	0	0	SRID=-1,MUI M736146.06 - 1-4-3-574		
21	724030	20	21	Level 5	6	Continuous	0	0	SRID=-1,MUI M736380.23 - 1-4-1-27		
22	724031	21	22	Level 5	6	Continuous	0	0	SRID=-1,MUI M736394.45 - 1-4-4-106		
23	724032	22	23	Level 5	6	Continuous	0	0	SRID=-1,MUI M736310.11 - 1-4-102-38		
24	724033	23	24	Level 5	6	Continuous	0	0	SRID=-1,MUI M736291.5-6 1-4-2-217		
25	724034	24	25	Level 5	6	Continuous	0	0	SRID=-1,MUI M735745.01 - 1-4-34-322		
26	724035	25	26	Level 5	6	Continuous	0	0	SRID=-1,MUI M736290.61 - 1-4-2-216		

Figura 4 : Captura de tela do pgAdmin III apresentando as estruturas da tabela lotes. Em destaque a coluna svg_geom.

4.2 Extraindo-se os dados do banco de dados

Para a extração dos dados geoespaciais armazenados no banco de dados, adota-se neste trabalho o método “getURL()” e “parseXML()” (URL 4.1) e (UR L4.2). Esses métodos são atualmente extensões do Plug-in Adobe, e devem ser incorporadas nas recomendações do W3C para a versão do SVG 1.2. Esses métodos são essenciais em aplicações que necessitam

ser constantemente atualizadas, como são os casos dos Web mapas dinâmicos e interativos. O método “.getURL()” apanha os dados de uma dada URL, utilizando o método “HTTP GET”, requisitando um método “callback” para passar os parâmetro para a função em um objeto ECMAScript. O método “.parseXML()” converte uma fonte “string” dentro de “DocumentFragment” que pertença a um determinado documento XML, passando esses fragmentos XML para uma outra string.

Neste trabalho, os métodos “.getURL()” e “.parseXML()” são utilizados no scrip principal main.js em ECMAScript/JavaScript para pegar e passar os nomes das diferentes camadas e dos atributos não espaciais armazenadas no banco de dados e apresentá-las no browser, sendo então visualizadas pelo usuário.

4.3 Definindo-se o modelo digital cartográfico

Neste trabalho o modelo digital cartográfico é definido conforme apresentado na Tabela 02.

Feições	Símbolos	Nível de medição (percepção)	Variáveis visuais
Ruas	áreas	Qualitativo (associativo e seletivo)	Cor e forma
Quadras	áreas	Qualitativo (associativo e seletivo)	Cor e forma
Lotes	áreas	Qualitativo (associativo e seletivo)	Cor e forma
Edificações	áreas	Qualitativo(associativo e seletivo)	Cor e forma
Rios e córregos	linhas	Qualitativo (associativo e seletivo)	Cor e forma
Rodovias	linhas	Qualitativo (associativo e seletivo)	Cor e forma
Jardins e parques	pontos e áreas	Qualitativo(associativo e seletivo)	Cor, forma e tamanho
Piscinas	áreas	Qualitativo (associativo e seletivo)	Cor e forma

5. Fluxograma do Trabalho

A Figura 05 representa de forma esquemática, através de um fluxograma, as diferentes etapas cumpridas no desenvolvimento do protótipo.

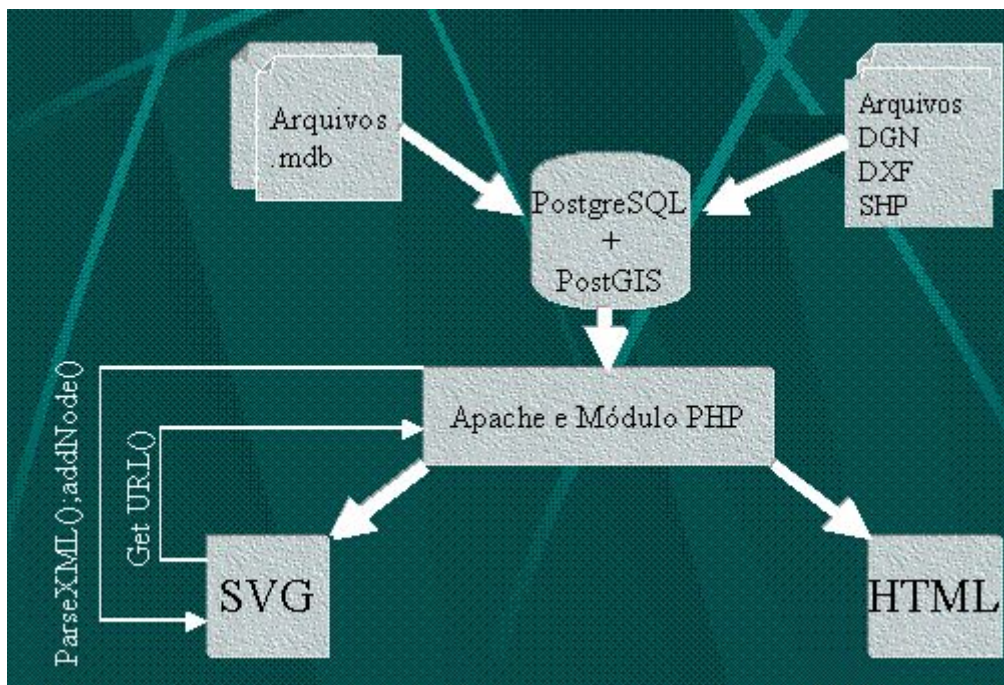


Figura 5 : Fluxograma representando as etapas de desenvolvimento do trabalho.

6. Análise e Resultados

6.1 Fluxograma do Trabalho

Os códigos implementados e utilizados neste trabalho estão armazenados em um servidor Apache e estruturados conforme esquema apresentado na Figura 06:

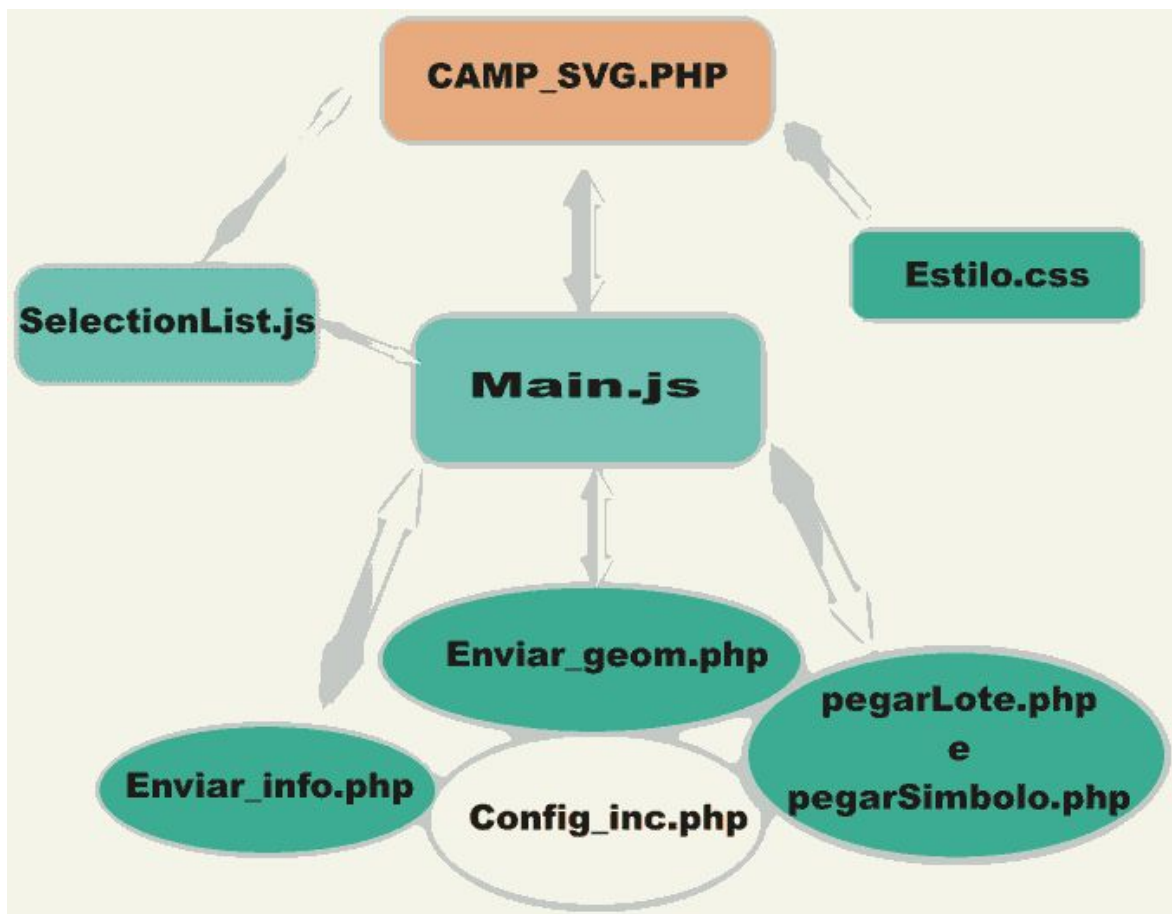


Figura 6 : Esquema de relacionamento entre os códigos implementados.

Esses arquivos possuem as seguintes funções:

O arquivo `camp_svg.php` é um script PHP que define toda a estrutura do Web mapa em código SVG no lado do servidor. Esse arquivo define o “layout” do Web mapa e os diferentes objetos da interface para interação com os usuários, com características intrínsecas ou extrínsecas. Nele estruturaram-se, também, as diferentes camadas do mapa que não foram armazenadas no banco de dados, como por exemplo: córregos, jardins e áreas verdes, piscinas, rodovias, pontes e viadutos, dentre outras.

O arquivo `main.js` é um arquivo estruturado em linguagem de programação ECMAScript/JavaScript, armazenado no servidor, que fornece todas as funções para as ações das interfaces proporcionadas pelos objetos de interação com os usuários e projetadas no arquivo `camp_svg.php`

O arquivo `config_inc.php` é um script PHP que possui a incumbência de fornecer os parâmetros de acesso ao banco de dados e os nomes das diferentes camadas do Web mapa armazenadas no servidor de banco de dados PostgreSQL, a todos os arquivos que o solicitarem.

O arquivo `enviar_geom.php` é um script PHP que possui a incumbência de extrair e passar a geometria das diferentes camadas armazenadas no PostgreSQL-PostGIS. Nesse script, utilizam-se o operador geométrico “overlaps - &&” do PostgreSQL e a função “BOX3D”, disposta pelo PostGIS para possibilitar consultas complexas. Nesse caso, o intuito é restringir a apresentação das camadas lotes e edificações ao tamanho da janela principal de

visualização do mapa, diminuindo a quantidade de dados a ser consultada e enviada pelo servidor, diminuindo o tempo de apresentação dos dados aos usuários do Web mapa.

O arquivo enviar_info.php é um script PHP que permite consultas aos dados não espaciais armazenados nas tabelas campinas e kobrat, estruturadas no PostgreSQL, passando os dados através do XML para serem apresentados em uma janela “pop up” no cliente, através da ação de clicar o botão direito do mouse sobre o lote consultado.

O arquivo pegarLote.php é um script PHP que permite consultar no banco de dados as geometrias dos objetos lotes, conforme solicitação feita pelo usuário através do código do lote digitado no objeto de interação campo de dados, implementada em SVG e disposta no arquivo camp_svg.php. Esse script utiliza a função OpenGIS “centroid(geometry)”, disponível no PostGIS para construir os centros geométricos dos objetos gráficos lotes.

O arquivo pegarSimbolo.php é um script php idealizado no script pegarLote.php que permite ao usuário selecionar através de uma caixa de seleção os diferentes atributos associados aos lotes. Nesse caso específico, selecionar os hospitais e os restaurantes mais próximos, possibilitando que o símbolo gráfico em SVG do atributo seja apresentado concomitantemente com o objeto gráfico lote, na janela de visualização do mapa. Os atributos dos lotes estão armazenados na coluna “símbolo”, na tabela lotes no PostgreSQL.

O arquivo selectionList.js é um script em linguagem ECMAScript/JavaScript, implementado por Hirtzler e Neumann (2003) (URL 6.1)), esse script cria a caixa de seleção no SVG de forma a possibilitar o usuário selecionar os respectivos elementos nela disponibilizados. Neste trabalho, foram feitas pequenas adaptações a esse script para permitir disponibilizar aos usuários, os atributos hospital e restaurante referentes à entidade lotes.

O arquivo estilo.css define uma folha de estilo externa que atribui estilo a todos os elementos gráficos e textuais do SVG que não são estilizados pela DTD.

6.2 As interfaces gráficas com os usuários

Considerando-se a existência dessa variedade de objetos de interação já implementados e aproveitando-se os esquemas projetados totalmente em SVG e DOM, e ações de interação com o usuário totalmente implementadas em script ECMAScript/JavaScript, optou-se em utilizar neste trabalho uma versão adaptada da interface interativa em SVG proposta por Neumann e Isakowski (2002), que fazem parte do projeto Carto.net do Instituto de Cartografia ETH de Zurique na Suíça (Ver (URL's 6.2 e 6.3)), como suporte para comprovar as potencialidades da utilização do servidor de banco de dados PostgreSQL e sua extensão PostGIS, como excelentes armazenadores de dados geoespaciais a serem utilizados no processo de construção de Web mapas estáticos com conteúdo e/ou interface interativa. Dentro dessa mesma filosofia, isto é, “Open-Source”, surge, também, o projeto GeoCliente proposto por Rogers (2003), do qual procurou-se incorporar, também, algumas funcionalidades desenvolvidas, já que os objetivos são semelhantes e com a vantagem de não haver a necessidade de instalação de nenhum outro tipo de programa de computador para visualizar o Web mapa. Dentre as funcionalidades incorporadas, destacam-se: legendas interativas, janelas de seleção de cores e janelas pop-up relacionando dados de consultas pré-definidas feitas ao servidor de banco de dados.

6.2.1 Funcionalidades com características extrínsecas

Janela suspensa “pop-up”

Através da ação de clicar o botão esquerdo do mouse sobre cada lote, uma janela suspensa é aberta e uma relação com as informações cadastrais relacionadas ao lote ativado é apresentada. Uma consulta pré-definida ao servidor de banco de dados PostgreSQL é responsável pela disposição das informações relativas aos dados cadastrais do lote ativado.

A codificação responsável por esse processo é disposta em um script PHP, denominado “enviar_info.php”. Essa funcionalidade é totalmente implementada em linguagem SVG e scripts PHP e ECMAScript/JavaScript. Detalhe dessa ação pode ser visto na Figura 07.

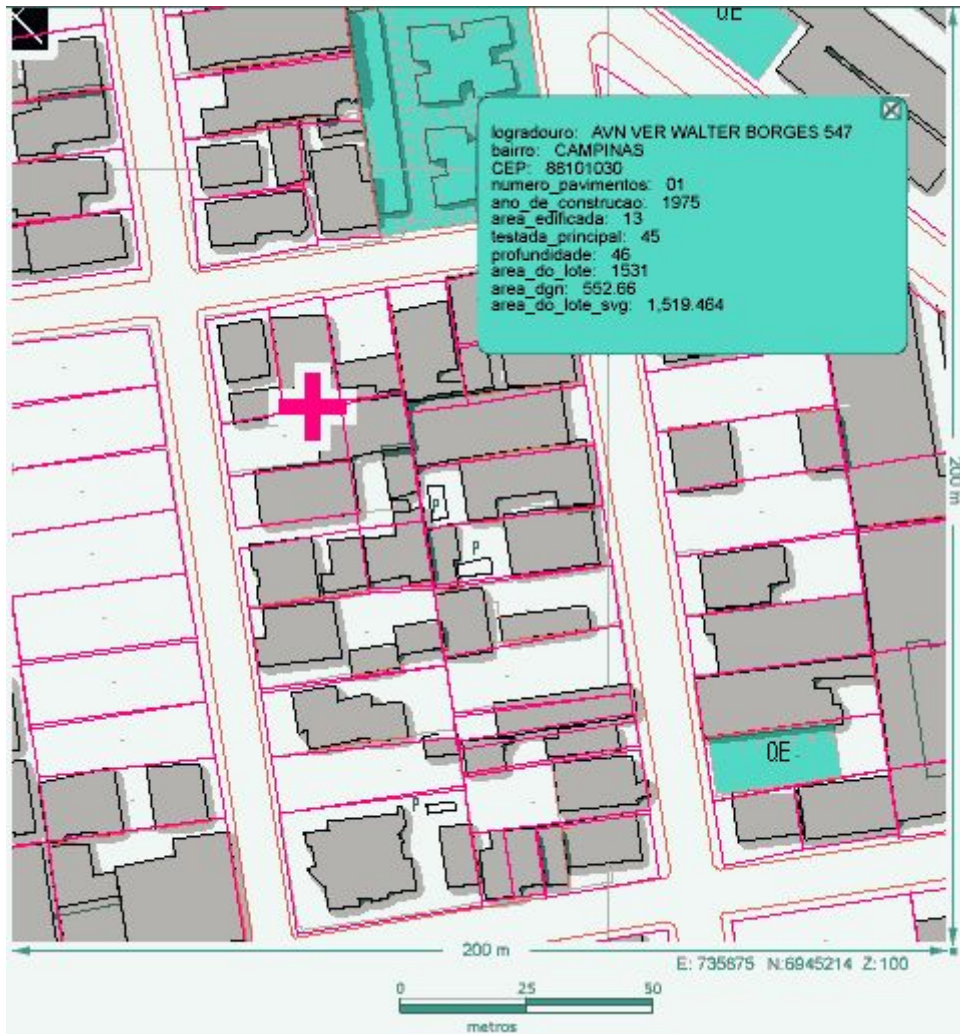


Figura 7 : Captura de tela do Web mapa representando uma janela pop-up.

Utilizaram-se neste trabalho diferentes objetos de interação com os usuários, que disponibilizam funcionalidades com características extrínsecas ao Web mapa projetado, dentre as quais destacam-se:

As caixas de checagem (seta – D) são um dos objetos de interação mais utilizados para controlar a apresentação das diferentes camadas de um Web mapa apresentado no browser, pois através delas o usuário pode selecionar uma ou todas as camadas para serem visualizadas ao mesmo tempo.

Neste trabalho, diferentes camadas armazenadas em um servidor de banco de dados PostgreSQL, como, por exemplo, lotes, edificações, ruas, quadras, são apresentadas aos usuários através da ação de clicar o botão direito do mouse sobre as caixas de checagem, habilitando-as ou desabilitando-as. A opção pela utilização do SVG, DOM e scripts ECMAScript/JavaScript se justifica pelo fato de, uma vez habilitada a camada no browser do cliente, não há mais necessidade de fazer novas consultas ao banco de dados, já que estes ficam armazenados no cliente. Evidentemente, havendo alterações dos dados no banco de dados, uma nova consulta faz-se necessária para atualizá-los no cliente.

A Figura 08 apresenta todas as caixas de checagem habilitadas com as respectivas camadas apresentadas.

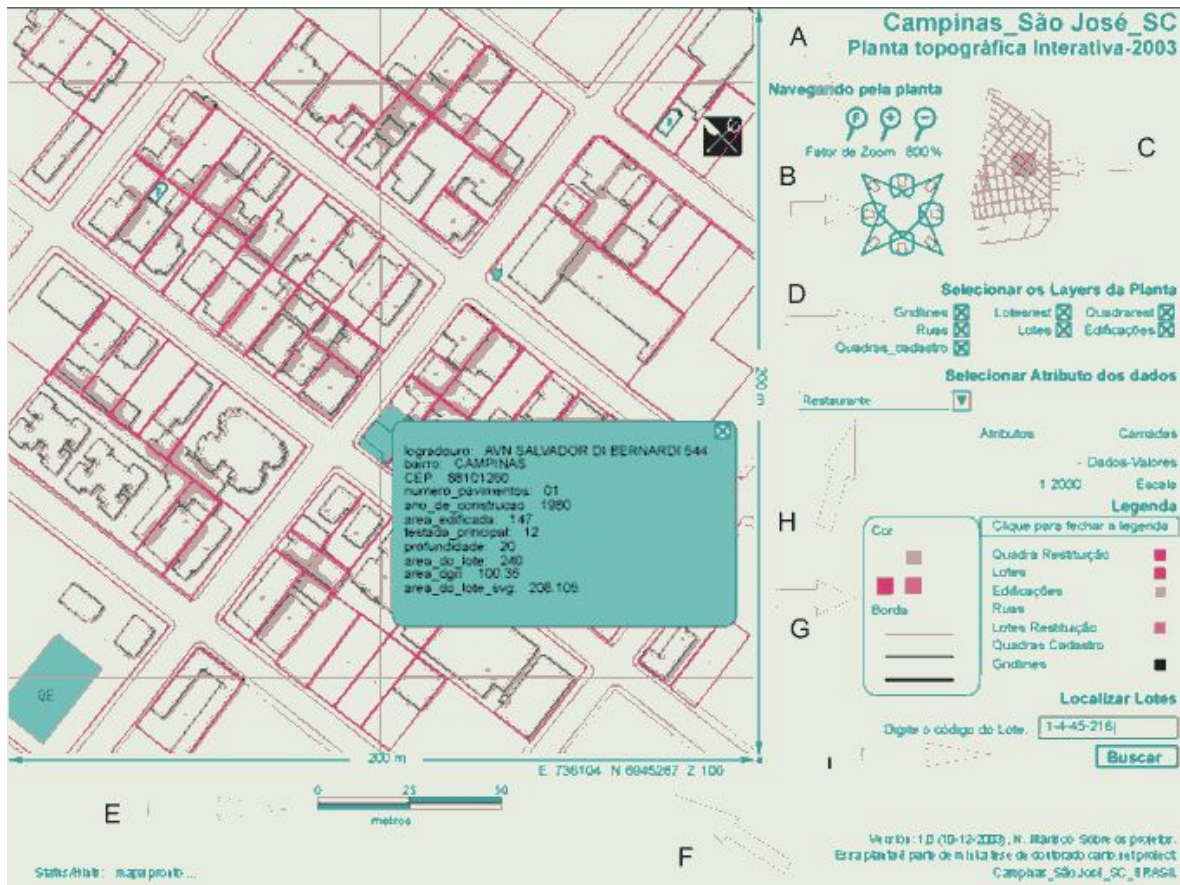


Figura 8 : Captura de tela do Web mapa projetado com a seta apontando para funcionalidades com características extrínsecas. Fonte: Adaptado de Neumann e Isakowski (2002).

As caixas de checagem são implementadas em código SVG e dispostas no arquivo camp_svg.php, sujeitas à ação do usuário e viabilizadas através das funções: "layerMenu(layerName)", "showAbout()", "checkBoxScript(evt)" e "onClick_ShowInfo(evt)" dispostas no arquivo main.js.

As setas a, b, c, d, e da Figura 08, representam respectivamente as interfaces de interação:

A) Navegação espacial ("Zooming", "Panning");

B) Função Panning;

C) Mapa para visão geral ("overview map");

E) Escala gráfica (dinâmica);

F) Mostrar coordenadas do usuário (planas UTM) ao passar o mouse sobre a área de apresentação do mapa;

G) Legenda Interativa.

6.2.1.1 Funcionalidades para consultas ao banco de dados

As funcionalidades para consultas ao banco de dados podem ser implementadas utilizando-se os seguintes objetos de interação: caixas de seleção, objetos ativos sobre o mapa, campos de textos, campo de dados, dentre outros.

O campo de dados (Figura 08 – seta I): possibilita ao usuário digitar textos ou números. Nesse caso, o usuário digita o código do lote, para estabelecer uma consulta ao lote de seu interesse, clicando com o botão direito do mouse sobre o botão de “busca”. Essa consulta disponibiliza na janela principal de visualização do mapa o lote consultado, permitindo que o usuário obtenha informações sobre o mesmo. Em seguida, o usuário pode clicar sobre o lote e uma janela “pop up” surgirá, apresentando os atributos não espaciais do lote armazenados no servidor de banco de dados PostgreSQL. A implementação dessa consulta possui um caráter restritivo, isto é, somente o usuário que possui a informação do código do lote poderá acessá-la. Justifica-se a escolha dos objetos campos de dados como interface de interação para a consulta de lotes ao banco de dados, devido ao fato de que a entrada dos dados pode ser feita utilizando-se um texto não muito extenso, nesse caso o código do lote. Na prática, entretanto, esses dados devem ser o número do Cadastro de Pessoa Física (CPF) do munícipe ou do registro imobiliário.

A implementação do campo de dados e do botão de consulta é feita em SVG. O script PHP, denominado “enviar_geom.php”, armazenado do lado do servidor, permite o acesso ao banco de dados. Nesse script, utilizam-se os operadores “centroid(geometry)” e “BOX3D(geometry)” do OpenGIS Simple Features Specification para o SQL, dispostos no PostGIS, no intuito de definir o centróide dos lotes e restringir o tamanho da área de apresentação das diferentes camadas do mapa ao usuário quando se efetiva a consulta. Isso reduz significativamente o tempo de apresentação da informação no browser. O script main.js em JavaScript/ECMAScript fornece as funções complementares para apanhar e passar corretamente os dados solicitados ao banco de dados.

Caixa de seleção (Figura 08 – seta H): a caixa de seleção implementada neste trabalho permite que o usuário selecione os atributos dos lotes. Esses atributos são apresentados através dos seus símbolos representados sobre a área. Neste trabalho, os símbolos implementados representam os restaurantes e os hospitais. Esses símbolos foram projetados em SVG por Held (2003), conforme (URL 6.4) e dispostos no arquivo camp_svg.php. A consulta ao banco de dados é viabilizada pelo script em PHP, denominado de pegarSimbolo.php. Esse script utiliza os operadores “POINT” e “distance”, ambos do PostGIS para, primeiramente, definir um ponto de referência e, em seguida, calcular a distância desse ponto ao objeto consultado, calculando-se concomitantemente o nível de zoom a ser disparado para apresentá-lo na janela de visualização principal.

O arquivo selectionList.js é um script ECMAScript desenvolvido por Hirtzler e Neumann para a implementação do objeto de interação com o usuário em SVG, conhecido como caixas de seleção. Dessa forma, utilizou-se esse script para a construção da caixa de seleção contendo os atributos restaurante e hospital para serem selecionados pelo usuário. Através do clicar do botão direito do mouse sobre a caixa de seleção, dispara-se uma consulta ao banco de dados. A ação de interação é fornecida pelas funções “function getObjectsFromDb()” e “function insertSimbol()”, implementadas em script ECMAScript/JavaScript, dispostas no arquivo main.js. Por questão de complexidade no processo de desenvolvimento da linguagem de programação, esta implementação faz com que apenas a informação mais próxima do ponto de referência seja apresentada ao usuário na janela principal de visualização do mapa.

7. Conclusões e Recomendações

7.1 Conclusões

No processo de construção do protótipo do Web mapa interativo, a arquitetura do sistema proposto encontra-se ancorada no paradigma computacional Cliente/Servidor, no qual o cliente solicita um serviço através do navegador Web e o servidor responde ao serviço solicitado. A arquitetura aqui idealizada procura refletir o modelo de três camadas na qual a

linguagem PHP exerce papel fundamental, tanto pela facilidade de seu aprendizado e utilização como por sua característica “Open Source”. Diferentes tecnologias podem ser empregadas, tanto do lado do cliente quanto do lado do servidor. Neste trabalho a opção pelas tecnologias fontes abertas e padronizadas como: XML, SVG, DOM, ECMAScript/JavaScript, Perl, PHP e um servidor de banco de dados PostgreSQL permitiu obter resultados como os apresentados neste trabalho. Deve-se isso ao fato de serem essas tecnologias “Open-Source” e por sua padronização, além da grande capacidade de suporte disponível na Web, facilitando a solução de problemas de forma rápida, barata e eficaz.

As diferentes funcionalidades disponíveis no Web mapa projetado, foram implementadas utilizando-se objetos gráficos de interação projetados totalmente em SVG, com ações de interação atribuídas a eles através de scripts em linguagem ECMAScript/JavaScript. Os diferentes objetos de interação com o usuário idealizados neste trabalho surgiram através dos estudos e das análises em diferentes projetos encontrados na Web, que exploram funcionalidades Web na construção de Web mapas interativos e/ou dinâmicos, utilizando-se como código para a apresentação gráfica o padrão SVG. Dessa forma, estabeleceram-se funcionalidades com características intrínsecas e extrínsecas, implementadas no Web mapa projetado no intuito de permitir ao usuário interagir com os dados geoespaciais, armazenados no servidor de banco de dados PostgreSQL dos quais, têm-se:

- Para desabilitar e habilitar diferentes camadas do Web mapa utilizaram-se os objetos de interação caixa de checagem. Para visualizar as diferentes camadas do mapa, o usuário clica sobre as caixas de checagem habilitando-as ou desabilitando-as. Camadas como ruas, lotes, quadras e edificações estruturadas em tabelas e armazenadas no PostgreSQL são apresentadas no navegador em padrão SVG.
- Para selecionar objetos e/ou atributos a serem apresentados sobre o mapa através de consultas pré-determinadas ao banco de dados, utilizou-se caixa de seleção. Essa implementação possibilita ao usuário selecionar, por exemplo: escolas, hospitais, postos de saúde, etc, visualizando os símbolos no mapa;
- Para funções de zoom, pan, arrastar e soltar, “overview mapping”, utilizaram-se objetos clicáveis e cursor do dispositivo de apontamento do mouse;
- Áreas sensíveis foram utilizadas para a apresentação de textos ou números aos usuários;
- Campo de texto e campo de dados foram utilizados para estabelecer consulta ao banco de dados. Essa implementação possibilita aos usuários digitarem o código do lote de interesse e através de uma consulta ao banco o objeto lote é apresentado em destaque no mapa;
- Janelas suspensas foram utilizadas para apresentar o resultado da consulta feita a tabelas contendo os atributos não espaciais, estruturadas no servidor de banco de dados. Essa implementação permite aos usuários obterem informações sobre os lotes, através da ação de clicar sobre os mesmos;
- Botões de comando, cursor do dispositivo de apontamento e caixas de atribuições são alguns dos objetos de interação empregados em ações de apresentação de coordenadas, de atributos dos objetos gráficos (área e perímetro), dentre outros dados apresentados aos usuários.

Todos os objetos de interação implementados no Web mapa projetado foram concebidos em linguagem SVG e DOM, com funções de interações projetadas em script ECMAScript/JavaScript. Justifica-se a opção de projetar toda a interface de interação com o usuário em SVG, por ser o SVG um XML e, portanto, um código padronizado que pode ser utilizado por diferentes equipamentos que suportem essa tecnologia, como os telefones celulares e assistentes pessoais digitais (PDA's). Uma outra questão a ser abordada é o caráter usável das interfaces propostas e utilizadas, isto é, elas já são bem familiares aos usuários de computadores pessoais facilitando sua usabilidade. A escolha da linguagem de

script ECMAScript/JavaScript pode ser justificada devido ao seu caráter padronizado e por estar definida nas especificações para o SVG e, principalmente, pelo seu caráter voltado a objetos.

Com a implementação e utilização desses objetos de interação, algumas funcionalidades exigidas para um ambiente de geovisualização ficam atendidas, como:

- Apresentações básicas: zoom, pan, escala, legendas;
- Orientação e identificação: mapa de visão geral e objetos ativos;
- Consultas aos dados: campo de dados; caixas de seleção e objetos ativos.

Este trabalho também confirma a viabilidade em se utilizar Sistema de Banco de Dados códigos fontes abertas (“Open Source”); nesse contexto, o PostgreSQL conjuntamente com a extensão PostGIS, como suporte ao armazenamento de dados geoespaciais, utilizados em projetos de construção de Web mapas interativos. A utilização do PostgreSQL e sua extensão espacial PostGIS fornece todo o suporte ao armazenamento dos dados gráficos e não gráficos utilizados ao longo do desenvolvimento deste trabalho. As camadas contendo os objetos geométricos representando os lotes, as edificações, as quadras e os arruamentos foram estruturados em tabelas e armazenadas no PostgreSQL, utilizando-se a extensão espacial PostGIS, sem maiores dificuldades. Deve-se destacar que, em processos de conversões de dados pré-existentes, esse procedimento está limitado à utilização de arquivos “shapefile”. No entanto, em etapas de obtenção e geração de novos dados, os dados gráficos e seus atributos não gráficos devem ser estruturados em uma única tabela, o que facilita o trabalho de utilização dos dados através do servidor de banco de dados PostgreSQL.

O PostGIS fornece as funções e os operadores que possibilitam consultar as geometrias (dados espaciais), estruturadas em tabelas, para serem apresentadas no navegador aos usuários. As consultas ao servidor de banco de dados foram implementadas utilizando-se da linguagem SQL. Os eficientes métodos “.getURL()” e “.parseXML()” viabilizaram a extração dos dados e a passagem dos parâmetros às funções, possibilitando assim a visualização dos dados no navegador.

Em função dos dados não espaciais encontrarem-se estruturados em tabelas separadas às tabelas que contêm os dados espaciais, viu-se diante da necessidade de se estabelecer uma relação entre eles. Identificaram-se cada objeto geométrico e estabeleceu-se uma relação com seus atributos não espaciais. Neste trabalho relacionaram-se os objetos referentes à camada lotes.

Assim, fica comprovada a importância em se estruturar em uma mesma tabela os dados espaciais e os seus atributos não espaciais, para o melhor aproveitamento da característica objeto relacional disponibilizada pelo PostgreSQL.

Os resultados obtidos com a utilização do padrão de apresentação gráfica SVG comprovam seu alto desempenho e sua excelente performance computacional à apresentação de Web mapas interativos na Web, através dos navegadores Internet Explorer 6.0, Opera 7.2 e do Netscape Communicator 4.8.

Devido ao fato de se estar trabalhando com representações cartográficas, que exigem um elevado nível de detalhamento das informações, optou-se em utilizar como padrão um nível de resolução gráfica para os desenhos dos traços das diferentes camadas do Web mapa em 300 dpi, isto é, 0.3 pixel. As influências no nível de percepção visual percebida através das variáveis visuais tamanho e cores, implementadas na legenda interativa, permitem ao usuário escolher a espessura dos traços, bem como, também, suas cores e as de seu preenchimento, para melhor visualizar as feições.

Para os tamanhos das fontes dos textos, referentes às características extrínsecas no Web mapa, recomenda-se que não sejam inferiores a 10pt (point), sob pena de não serem interpretados pelo usuário. Já para os textos referentes às características intrínsecas, estes são dependentes de suas funcionalidades e do nível de zoom de apresentação do Web mapa, podendo-se definir para eles valores inferiores a 10pt, sem prejuízo à interpretação.

Pode-se afirmar que atualmente, as limitações encontradas na utilização do padrão SVG em projetos de disseminação de Web mapas pela Internet, estão mais restritos aos constantes processos de evolução por que passam essas novas tecnologias e na necessidade de se instalar um Plug-in para sua visualização no navegador, além da eminente necessidade em se divulgar e demonstrar, aos potenciais usuários dessas novas tecnologias, suas enormes potencialidades, algumas das quais, demonstradas neste trabalho.

Ao considerar todos os elementos abordados anteriormente, conclui-se que o protótipo do Web mapa projetado enquadra-se dentro das características de ser um Web mapa estático com conteúdos e interfaces interativas, possuindo alta interatividade, destinado ao público em geral com objetivo de apresentar o conhecido. Portanto, fica comprovado o estreito relacionamento que existe entre as funções de visualização e comunicação de um produto cartográfico.

Por fim, entende-se que a importância em se incorporar tecnologias “Open Source”, para suportar as diversas atividades da gestão pública brasileira, está amparada na incessante busca de soluções para a seguinte dualidade: reduzir custos nos processos de gestão em contraponto com as necessidades de ampliar e melhorar os serviços prestados. Com a adoção de tecnologias códigos fontes abertas, vêm-se ampliadas essas possibilidades, isto é, de desenvolvimento tecnológico e, conseqüentemente, diminuição com gastos na aquisição de equipamentos e programas de computadores, além de impulsionar o desenvolvimento de aplicativos computacionais que venham a atender demandas e necessidades específicas.

Com os códigos fontes abertos, ampliam-se, também, as possibilidades de democratizar o acesso à informação, proporcionando, assim, a inclusão digital. As soluções propostas neste trabalho ajudam, também, a solucionar problemas de padronização dos dados a serem utilizados por diferentes órgãos da administração pública e privada, o que pode facilitar a interoperabilidade entre eles. Essas ações refletem em diminuição no tempo e nos custos de manutenção dos sistemas projetados, disponibilizando, concomitantemente, informações atualizadas e de alta confiabilidade.

7.2 Recomendações

Com o desenvolvimento deste trabalho deram-se os primeiros passos em direção à aquisição de conhecimento e do domínio das potencialidades na exploração das tecnologias de códigos fontes abertas para projetar Web mapas interativos e/ou dinâmicos, com o intuito de disseminar informações relativas ao cadastro imobiliário urbano municipal pela Internet. Logo, para a continuidade deste trabalho, algumas sugestões propositivas são apresentadas, juntamente com recomendações que possam ser incorporadas em pesquisas futuras:

- Testar o sistema proposto em ambiente GNU-LINUX;
- Testar a possibilidade de utilização da biblioteca Terralib;
- Aprimorar o script para a conversão dos dados gráficos armazenados no PostgreSQL em “Well-Known Text Representation of Geometry” (WKT) para o SVG, estendendo-o para as outras funções OpenGIS;
- Implementar escala animada (zoom dinâmico), isto é, variar a densidade de informações apresentada e a simbolização conforme o nível de zoom;
- Aperfeiçoar a simbolização dos diferentes atributos de forma a refletir a variação em escala;
- Estender o sistema para apresentar informações relacionadas às questões fiscais e legais do cadastro imobiliário urbano municipal.
- Ampliar o sistema para incorporar funções GIS, isto é, permitir a realização de análise espacial;

- Criar mecanismos que permita integrá-lo ao sistema de posicionamento por satélite (GPS) com intuito de possibilitar a localização e a orientação on-line do usuário, bem como a atualização dos dados cadastrais no banco de dados.
- Ampliar a participação de pesquisadores da área de computação para ampliar o nível de desenvolvimento.

8. Referências Bibliográficas

EISENBERG, J.D.: *SGV Essentials*, O'Reilly & Associates, Inc. United States of American, 335p, 2002

HELD, von G.: *Anforderungen an einen kartographischen Viewer für Business Intelligence Systeme*, Diplomarbeit, University of Applied Sciences – FH Karlsruhe 2003

KRAAK, M.J.: *Setting and needs for web cartography*, In: KRAAK, M.J. e BROWN, A., *Web Cartography – developments and prospects*, p.01-07, Editora: Taylor & Francis, Landon 2001a

_____: *Trends in Cartography*, In: KRAAK, M.J. e BROWN, A., *Web Cartography – developments and prospects*, p.08-19, Editora: Taylor & Francis, Landon 2001b

ORMELING, F. J.: *Cartography: visualization of geospatial data*, 2.ed., London, Pearson Education, 205p, 2003

MacEACHREN, A M; KRAAK, M.J.: *Research Challenges in Geovisualization*, In: *CARTOGRAPHY AND GEOGRAPHIC INFORMATION SCIENCE*, Vol. 28(1), p.3-12, 2001

MARISCO, N.: *Web Mapas Interativos como Interfaces aos Dados Geoespaciais: Uma Abordagem Utilizando-se Tecnologias Fontes Abertas*, Originalmente apresentada como tese de doutorado, Departamento de Engenharia Civil – Universidade Federal de Santa Catarina UFSC, Florianópolis 2004

NEWMANN, A.; ISAKOWSKI, Y.: *Interactive Topographic Web-Maps Using SVG*, In: *SVG/OPEN CARTO.NET DEVELOPER CONFERENCE – Zurich*, 15-17 julho, 2002. Disponível em [Paper SVG](#), Open 2002, Zurich, *Interactive Topographic Web-Maps Using SVG*, Acesso em: 14 jan. 2003

PETERSON, M.P.: *The Development of Map Distribution through the Internet*, In: *PROCEEDINGS OF THE 20th ICA CONFERENCE*, Beijing, China, 6-10/09/2001

ROGERS, N.: *Geopooling: SVG as an Enabling Technology for Sharing Geographic Knowledge*, In: *2nd ANNUAL CONFERENCE ON SCALABLE VECTOR GRAPHICS* - Vancouver, Canada, July 13-18, 2003. Disponível em [Geopooling](#), acessado em 24 jun. 2003

SHNEIDERMAN, B.: *Dynamic queries for visual information seeking*, *IEEE Software*, p.70-77, 1994

9. URL

- URL 4.1 [SVG-wiki - Get Url](#)
- URL 4.2 [carto:net - getURL & parseXML](#)
- URL 6.1 Script ECMAScript/JavaScript - selectionList
<<http://www.carto.net/papers/svg/gui/selectionlist/selectionList.js>
- URL 6.2 [Vienna - Social Patterns and Structures](#)
- URL 6.3 [Tuerlersee - Interactive Topographic Map Example](#)
- URL 6.4 Símbolos gráficos em SVG [carto:net - Cartographic symbol collection](#)

