

UNIVERSIDADE FEDERAL DE SANTA CATARINA - UFSC
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL - PPGEC
ÁREA DE CONCENTRAÇÃO: CADASTRO TÉCNICO MULTIFINALITÁRIO

**POTENCIAL DA LINGUAGEM SVG *SCALABLE VECTOR GRAPHIC* PARA
VISUALIZAÇÃO DE DADOS ESPACIAIS NA INTERNET - ESTUDO DE CASO**

Dissertação submetida à Universidade Federal de Santa Catarina como requisito parcial exigido pelo Programa de Pós-Graduação em Engenharia Civil – PPGEC, para a obtenção do título de Mestre em Engenharia Civil.

Vamilson Prudêncio da Silva Júnior

Florianópolis, 2004.

Ficha Catalográfica

SILVA Jr., Vamilson Prudêncio da.

Potencial da linguagem SVG *Scalable Vector Graphic* para visualização de dados espaciais na Internet - Estudo de Caso. Vamilson Prudêncio da Silva Júnior. Florianópolis, 2004. 110 páginas; il; tabs; figs;

Orientador: Carlos Loch
Dissertação (Mestrado em Engenharia Civil) –
Programa de Pós-Graduação em Engenharia Civil – Cadastro
Técnico Multifinalitário – Universidade Federal de Santa
Catarina

1 - SVG; 2 - Mapas Interativos; 3 – Internet.

**POTENCIAL DA LINGUAGEM SVG *SCALABLE VECTOR GRAPHIC* PARA
VISUALIZAÇÃO DE DADOS ESPACIAIS NA INTERNET - ESTUDO DE CASO**

VAMILSON PRUDÊNCIO DA SILVA JÚNIOR

Dissertação julgada adequada para a obtenção do Título de Mestre em Engenharia Civil e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Civil - PPGEC da Universidade Federal de Santa Catarina - UFSC.

Prof^a. Henriette Lebre La Rovere - Coordenadora do PPGEC

Prof. Dr.- Carlos Loch - Orientador

COMISSÃO EXAMINADORA:

Dr. Carlos Loch - ECV/UFSC
Moderador

Dr. Ing. Jürgen Wilhelm Philips - ECV/UFSC

Dr. João Bosco Mangueira Sobral – INE/UFSC

Dr. João Rosaldo Vollertt Junior - CIRAM

Agradecimentos

A realização desse trabalho só foi possível graças à ajuda de muitos colegas de trabalho e amigos que durante o curso prestaram apoio valioso, com críticas construtivas, sugestões e incentivo. Agradeço aos colegas e professores do Programa de Pós-Graduação em Engenharia Civil, área de Cadastro Técnico Multifinalitário da Universidade Federal de Santa Catarina, que dividiram comigo as dificuldades e os méritos do trabalho. Destaco o apoio recebido de meu orientador que soube delinear o caminho à ser seguido para chegar ao resultado final desse trabalho.

Em especial agradeço aos colegas de trabalho da Epagri, que de uma forma ou de outra participaram do trabalho e incentivaram, com palavras de apoio principalmente nos momentos mais críticos, e que souberam encontrar no trabalho grande importância para a Instituição.

Resumo

SILVA JR., Vamilson Prudêncio da. **Potencial da linguagem SVG - *Scalable Vector Graphic* para visualização de dados espaciais na Internet - Estudo de Caso.** Florianópolis, 2004. 110 f. Dissertação (Mestrado) – Universidade Federal de Santa Catarina.

A expansão da Internet propiciou um meio para divulgação de informações com grande penetração em todo o mundo. Paralelamente, o impacto da informática sobre a cartografia causou surgimento de novas técnicas e recursos que potencializaram seu uso. Até recentemente os mapas digitais encontrados na Internet eram apresentados como imagens matriciais, com pouca ou nenhuma interatividade com o usuário, limitando-se a exibir o mapa na tela do computador. O surgimento da linguagem XML (*extensible Markup Language*) trouxe novas possibilidades para publicação de mapas na Internet, em função de sua capacidade de manipular dados gráficos. Este trabalho procura explorar o potencial da linguagem SVG, que é baseada em XML, para publicação de mapas no ambiente da Internet, apresentando ferramentas como um alternativa para qualquer instituição que necessite divulgar informações espaciais de maneira simplificada, sem necessariamente usar banco de dados ou servidor de mapas. Primeiramente faz-se uma abordagem teórica sobre disseminação dos dados espaciais na Internet e as linguagens de marcação envolvidas, mostrando suas principais características. Depois o trabalho aborda aspectos da seleção de mapas para publicação, a sua transformação para o formato SVG, implementação de *JavaScripts* nos arquivos para proporcionar interatividade e finalmente apresenta dois protótipos para montagem de uma página contendo o mapa e ferramentas para interação. Os protótipos foram projetados para casos onde os atributos e complexidade dos mapas é relativamente reduzida, dispensando a necessidade de um servidor de mapas. A linguagem SVG mostra-se promissora para apresentação de mapas na Internet devido a sua flexibilidade e grande capacidade de interação com o usuário, além de mostrar uma clara tendência de se tornar uma linguagem padrão nos próximos anos. O maior problema encontrado reside na necessidade de instalação de “plug-in” no navegador de Internet para visualização dos gráficos nesse formato.

Palavras-chave: SVG; mapas interativos; Internet.

Abstract

SILVA JR., Vamilson Prudêncio da. **Potential from language SVG - *Scalable Vector Graphic* for spatial data display over the Internet**. Florianópolis, 2004. 110 p. Dissertation (Masters degree) – Federal University of Santa Catarina.

The expansion of the Internet propitiated an environment for world wide information distribution. Simultaneously, the impact of informatics on cartography has caused the development of new techniques and resources that powered up its use. Until recently, digital maps encountered over the Internet were represented as raster images simple maps displays on computer screen with little or none interactivity with the user. The arising of XML Language (extensible Markup Language) brought new possibilities for map publications over the Internet, due to its good graphs manipulation capacity. This work explores the potential of SVG language, which is XML based, to publish maps into the Internet environment, as an alternative for institutions that need to disclose spatial information in a simplified way, without necessarily use a data base or map server. First a theoretical discussion is made about spatial data dissemination over the Internet and the markup languages involved, showing its principal characteristics. Secondly, the present work approaches the aspects of map selection for publication, its conversion to SVG format including JavaScript's implementations to improve its interactivity. Finally, the work presents two prototypes for an internet page construction containing a map and interaction tools. The prototypes have been conceived for cases where the quantity of information and thematic maps are relatively small, dispensing the need of a map server. This study shows that SVG language is a promising alternative for map display revealing a trend of becoming a standard language in years to come. The largest problem encountered is that, for a SVG graphic format display, a navigator "plug in" is needed.

Key words: SVG; interactive maps; Internet.

Sumário

1	Introdução	1
2	Objetivos.....	3
2.1	Geral	3
2.2	Específicos	3
3	Contexto da proposta de trabalho	4
4	Uma Revisão Teórica	8
4.1	SIG (Sistema de Informações Geográficas)	8
4.2	Web e Mapas	10
4.2.1	– Imagens Matriciais e Vetoriais	13
4.2.1.1	Imagens Raster.....	13
4.2.1.1.1	Resolução.....	14
4.2.1.1.2	Cor	14
4.2.1.1.3	Tamanho do arquivo	15
4.2.1.1.4	Formatos de arquivo	15
4.2.1.2	Imagens Vetoriais	15
4.2.1.2.1	Resolução.....	16
4.2.1.2.2	Cor	16
4.2.1.2.3	Tamanho de arquivo	16
4.2.1.2.4	Formatos de arquivo	16
4.2.2	Outras tecnologias	18
4.3	Linguagens de Marcação	19
4.3.1	SGML (<i>Standard Generalized Markup Language</i>)	20
4.3.1.1	Definição.....	20
4.3.1.2	Breve Histórico.....	21
4.3.1.3	Visão Geral	21
4.3.2	HTML (<i>HyperText Markup Language</i>)	22
4.3.2.1	Definição.....	22
4.3.2.2	Histórico e Objetivos	22
4.3.2.3	Estrutura de um documento HTML.....	23
4.3.3	XML (<i>eXtensible Markup Language</i>).....	24
4.3.3.1	Definição.....	24
4.3.3.2	Breve Histórico	25
4.3.3.3	Visão Geral	25
4.3.4	GML (<i>Geography Markup Language</i>).....	26
4.3.4.1	Definição.....	26
4.3.4.2	Breve Histórico	27
4.3.4.3	Visão Geral	27
4.3.4.4	Estrutura de um arquivo GML.....	28
4.3.5	SVG (<i>Scalable Vector Graphics</i>)	29
4.3.5.1	Definição.....	29
4.3.5.2	Breve Histórico.....	29
4.3.5.3	Visão geral	30
4.3.5.4	Compatibilidade:.....	31
4.3.5.5	Estrutura de um documento SVG	32
4.3.6	A Relação entre SGML/XML/HTML.....	34

4.4 Outras linguagens	35
4.4.1 Linguagem JAVA.....	35
4.4.1.1 Definição.....	35
4.4.1.2 Breve Histórico.....	36
4.4.1.3 Visão Geral.....	37
4.4.1.4 Como Java é executado	38
4.4.1.5 O que são Applets.....	39
4.4.2 JavaScript	40
4.4.2.1 Definição e origem	40
4.4.2.2 Visão Geral.....	41
4.4.2.3 Estrutura do JavaScript.....	42
4.5 Disseminação de Informações Geográficas na Internet.....	42
5 Desenvolvimento do trabalho	47
5.1 Seleção dos mapas para publicação.....	47
5.1.1 Caracterização do tema de trabalho.....	49
5.1.1.1 Material usado no levantamento.....	50
5.1.1.2 Divulgação dos resultados do Inventário.....	51
5.1.1.3 Aplicabilidade dos protótipos à outras áreas e temas	52
5.2 Geração do mapa em formato SVG.....	52
5.2.1 Script em Avenue	54
5.2.2 Conversor JAVA	55
5.2.2.1 Operação do conversor	58
5.3 Modelo 1 - Protótipo SVG simplificado.....	59
5.4 Modelo 2 – Protótipo GeoClient modificado	67
5.4.1 Operação do protótipo GeoClient modificado.....	70
6 Conclusões.....	81
6.1 A linguagem SVG.....	81
6.2 Os protótipos.....	83
6.3 Geral	84
7 Referências Bibliográficas.....	86
8 – Glossário.....	90

Lista de Tabelas

TABELA 1 - Visão geral de formatos de imagens usados na WEB	17
TABELA 2 - Funções JavaScript incorporadas ao arquivo “mapa_index.htm”	61
TABELA 3 - Arquivos do GeoClient modificado	69

Lista de Figuras

FIGURA 1 -Arquitetura de um sistema de informações geográficas.....	10
FIGURA 2 - Transmissão de Applets Java	40
FIGURA 3 – Localização da área de estudo no Estado de Santa Catarina.	49
FIGURA 4 – Localização da área de estudo no Oeste de Santa Catarina.	50
FIGURA 5 - Aspecto da ferramenta para ArcView.	54
FIGURA 6 – Aspecto do conversor escrito em JAVA.....	58
FIGURA 7 - Aspecto da página usando modelo 1.	67
FIGURA 8 - Aspecto da página usando protótipo 2.	71
FIGURA 9 - Uso da ferramenta de identificação.....	72
FIGURA 10 - Uso da ferramenta de consulta.	73
FIGURA 11 - Configuração de legenda.....	74
FIGURA 12 - Configuração do rótulo.....	75
FIGURA 13 - Aplicação de rótulos.....	76
FIGURA 14 - Visualizando atributos e gráficos.	78
FIGURA 15 - Esquema da metodologia proposta.....	79

1 Introdução

Muitas instituições de Pesquisa e Desenvolvimento geraram na última década, uma grande quantidade de mapas em formato digital, com riqueza de informações espaciais e procuram por maneiras eficientes de publicá-los, para que possam atingir seu objetivo final de levar informações à população, estudantes, pesquisadores, técnicos e governantes, gerando subsídios e ferramentas para o projetos de gestão ambiental, territorial e desenvolvimento. Paralelamente, as novas tecnologias criaram um novo cenário, onde os usuários de mapas já não se satisfazem com um simples mapa em papel, mas querem extrair dele uma quantidade maior de informações, gerando uma demanda social por informação espacial, resultado da globalização e de novos paradigmas de gestão territorial.

A expansão da Internet criou um ambiente adequado para divulgação de informações massivamente, e apresenta-se como uma oportunidade para publicação dos mapas, interativos ou não. Até pouco tempo os mapas disponíveis na Internet eram apenas imagens matriciais (em formato raster) sendo que as mais divulgadas na Web, estão no formato BMP, JPG ou GIF, e apresentam como principal desvantagem o tamanho dos arquivos, quando comparados à arquivos vetoriais. Além disso esses formatos oferecem apenas visualização, com poucas possibilidades de interatividade, como ampliação através de *zoom*, abrindo uma clara vantagem para os arquivos vetoriais quando se pretende usá-los para apresentação de mapas.

No entanto, com o aparecimento da linguagem XML (*eXtensible Markup Language*), que oferece condições de armazenamento, transferência e manipulação de dados gráficos, tornou-se possível publicar mapas vetoriais na Internet, ampliando bastante os recursos de interatividade. No capítulo 4.2 o trabalho aborda uma comparação entre imagens apresentadas na Internet na forma matricial e na forma vetorial.

Atualmente existem diversas soluções comerciais e de domínio público para publicação de mapas na *Web*, sendo que as mais avançadas oferecem ferramentas bastante sofisticadas. Entretanto, para muitos casos a quantidade de informações associadas ao mapa é pequena e não necessita de recursos tecnológicos mais avançados como um servidor de mapas e banco de dados acoplado. Mesmo assim, certos recursos mínimos devem estar presentes, como a realização de pesquisas, simples ou complexas, seleções, diretas ou por parâmetros

predefinidos, formatação de legenda, impressão de mapas, etc. Nessa linha, a disponibilização de dados espaciais na *Web* facilita o acesso de pessoas que necessitam manipular informações georreferenciadas e que não estão fisicamente em seu local de trabalho, minimizando os esforços em busca desta informação.

Este trabalho objetiva explorar o potencial da linguagem SVG (*Scalable Vector Graphics*) que é uma especificação da linguagem XML, como ferramenta para publicação de mapas interativos na Internet. O uso desse tipo de tecnologia com código fonte aberto em projetos de Web e mapas interativos é altamente recomendado, de acordo com o trabalho de MARISCO (2004). O autor afirma que com a adoção dessas tecnologias pelas gestões públicas e privadas, vêm-se ampliadas as possibilidades de desenvolvimento tecnológico e, conseqüentemente, uma 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.

2 Objetivos

2.1 Geral

Explorar o potencial da linguagem SVG (*Scalable Vector Graphics*) para visualização de dados espaciais na Internet.

2.2 Específicos

a) apresentar alternativas de divulgação via Internet de mapas digitais gerados pelas instituições de pesquisa e desenvolvimento, usando “software livre”, que permitam publicar mapas simplificados, com reduzida quantidade de informações, casos em que não há necessidade de integração com um servidor de mapas.

b) usar a linguagem SVG para montar protótipos de páginas de Internet para apresentação de mapas interativos.

c) explorar o potencial de ferramentas de baixo custo para publicação de mapas na Internet.

3 Contexto da proposta de trabalho

A necessidade de diversos setores da sociedade, como órgãos de Pesquisa e Desenvolvimento, órgãos governamentais, indústrias e empresas privadas de tratar informações gráficas e alfanuméricas de forma integrada foi, em parte, satisfeita pelo surgimento dos Sistemas de Informações Geográficas (SIG). Esses sistemas foram criados há aproximadamente 3 décadas, e de lá para cá, houve um processo de adaptação do sistema cartográfico usado até então para uma nova tecnologia oferecida pelos computadores. Assim a informática modernizou e agilizou muito o processo de geração dos mapas, além de permitir uma nova maneira de armazenamento: digital. Os sistemas são altamente sofisticados atualmente, com uma grande quantidade de funções de análise espacial, precisão cartográfica acurada, transformações entre diversos sistemas de projeções, e outros recursos. Outra grande vantagem que os sistemas informatizados incorporaram é a sua capacidade de interagir com sistemas de sensoriamento remoto, permitindo a junção de mapas digitais com imagens de satélites, abrindo novos campos em diversas áreas de pesquisa relacionadas às ciências da terra.

Como normalmente esses sistemas são executados em computadores pessoais, que no máximo estão ligados à mais algumas máquinas dentro de uma corporação através de uma rede interna, as possibilidades de compartilhamento de mapas fica restrita nesse ambiente de rede interna, e geralmente só para aquelas máquinas que possuem licença do software. Interações simples que o SIG oferece, como apontar para uma determinada área e mostrar informações, ampliar ou reduzir, só são possíveis porque o SIG está instalado na máquina do usuário. Para trocar mapas digitais com outras instituições, é necessário uso de mídia para gravação, pois normalmente os arquivos que compõe o mapa são bastante grandes.

Existe ainda o problema de SIGs diferentes entre as instituições, pois normalmente cada fabricante adota um formato de arquivo proprietário e incompatível com os concorrentes. De acordo com LIMA (2002), a expansão da rede mundial de computadores e a crescente popularização de SIGs em novos domínios de aplicações, propiciam um ambiente para realização de intercâmbio de informação espacial, impulsionado principalmente pelo custo de produzir o dado independentemente. Entretanto, a falta de modelos conceituais comuns acarreta problemas na troca de dados entre organizações utilizando SIGs distintos. Existem

esforços internacionais no sentido de estabelecer padrões tecnicamente coerentes e aceitos pela comunidade científica, entre os quais se destacam o SDTS (*Spatial Data Transfer Standard*) nos EUA, o SAIF (*Spatial Archive and Interchange Format*) no Canadá e o OGC (*Open GIS Consortium*), nos Estados Unidos. Por outro lado alguns formatos como “dxf” da AutoCad e “shapefile” da Esri estão se tornando mais comuns para importação e exportação de mapas.

Na última década muitas instituições geraram uma grande quantidade de mapas, armazenados em forma de arquivos eletrônicos, com riqueza de informações geográficas e atributos relacionados. Para que os pesquisadores, estudantes ou qualquer outro usuário possam visualizar esses mapas, necessitam de software adequado, normalmente um SIG produzido por empresas proprietárias dos formatos de arquivos manipulados pelo software. Como tais softwares normalmente não são de domínio público, há uma implicação de custos para que se possa usar os mapas digitais com seu potencial completo de informação.

Para o caso de SIGs instalados no computador do usuário, os fabricantes projetam seus softwares para que usem o máximo possível de recursos da máquina. Porém na Internet tal situação é inconcebível, já que não se sabe a configuração das máquinas ligadas à rede, tampouco o sistema operacional ou navegador de Internet (*browser*) de cada usuário.

Por outro lado, a rede Internet é um ambiente muito promissor para divulgação de mapas interativos por ser um meio de difusão de informações eficiente e barato, tanto que isso vem merecendo especial atenção dos órgãos de Pesquisa e Desenvolvimento, órgãos governamentais, indústrias e empresas privadas.

A rápida e expansão da Internet possibilita que usuários em regiões remotas possam acessar as informações espaciais, desde que haja uma série de processos pelos quais os arquivos eletrônicos que representam os mapas passam, de forma que no final possam ser visualizados pelos *browsers*.

Estes processos constituem um desafio técnico para os profissionais da área de geoprocessamento, pois existem hoje muitas formas de transmissão das informações e a capacidade de transmissão de dados na maior parte da Internet é ainda limitada, embora o futuro aponte para uma rapidez de acesso bem maior, o que exige processos de compactação e minimização de informações trocadas, porém, sem perder a eficiência.

Nesse ambiente, muitos pesquisadores e técnicos fazem uso da linguagem de marcação *Hyper Text Markup Language* (HTML), que é a linguagem em que a maioria das páginas de Internet são construídas, para divulgar seus dados, incorporando nessas páginas arquivos matriciais (raster) no formato JPF e GIF. Esse tipo de imagem pode conter perfeitamente um mapa, mas a linguagem usada não permite interatividade, e tão pouco suporta mapas vetoriais.

Com o surgimento da linguagem de marcação *eXtensible Markup Language* (XML), que cria condições de armazenamento e transferência de dados, logo surgiram novos padrões capazes de manipular dados gráficos. Assim, o consórcio *World Wide Web Consortium* (W3C) recentemente definiu um padrão de divulgação de mapas vetoriais na *Web* usando a especificação *Scalable Vector Graphics* (SVG), originalmente especificado pela Adobe como *Precision Graphics Markup Language* (PGML). Como o SVG é um padrão aberto (livre) a Microsoft em contrapartida definiu o *Vector Graphics Markup Language*, mais conhecido pela sigla VML. No entanto esse trabalho se atém ao SVG como alternativa para quem desenvolve aplicações para a Internet, por não ser um software proprietário.

Desde o surgimento dessas linguagens com seus novos recursos, começaram a surgir várias alternativas para publicação de mapas na *Web*. Nesse aspecto existe uma clara distinção entre os sistemas comerciais e os de domínio público. Embora o objetivo seja o mesmo, existe uma clara distinção entre as filosofias envolvidas, já que um lado considera que a tecnologia envolvida implica em direitos autorais, enquanto o outro trata a tecnologia como área de conhecimento público cuja divulgação pode (e deve) ser feita para desenvolvimento da área. De qualquer forma, as soluções comerciais são, do ponto de vista técnico, mais eficientes e vantajosas que as soluções de domínio público. Entretanto, existe uma grande quantidade de projetos que não necessitam de sistemas complexos para publicar seus mapas, dado a natureza mais simples das informações ali contidas, e nesses casos os sistemas de domínio público podem ser a opção mais adequada.

É necessário que as instituições de Pesquisa e Desenvolvimento também participem das discussões a respeito das diversas soluções possíveis para apresentação de mapas interativos (ou não) na Internet, e efetivamente comecem a dispor suas informações para os usuários.

Nesse contexto, este trabalho objetiva explorar o potencial da linguagem SVG como ferramenta para publicação de mapas interativos na Internet, propondo duas alternativas de protótipos para montagem de uma página, com interface amigável, contendo o mapa e algumas ferramentas para interação, visando facilitar a transmissão da informação contida no mapa para o usuário, dispensando a necessidade de um servidor de mapas. Naturalmente, os protótipos aqui apresentados são adequados para casos onde a quantidade de informações e mapas temáticos é relativamente reduzida.

4 Uma Revisão Teórica

Esse capítulo apresenta uma revisão bibliográfica sobre os principais assuntos tratados, começando por SIG, depois abordando aspectos gerais sobre publicação de mapas na *Web*, as principais linguagens de marcação usadas no desenvolvimento do estudo, as linguagens de programação JAVA e JavaScript e por fim algumas considerações sobre disseminação de dados geográficos na Internet.

4.1 SIG (Sistema de Informações Geográficas)

Um Sistema de Informações Geográficas (SIG) pode ser considerado um ambiente tecnológico valioso para as mais diversas áreas de conhecimento e de atuação sobre os meios físico e social. Dentre as inúmeras aplicações, em nível nacional e internacional, destaque é dado ao planejamento, à agricultura, à análise ambiental, à análise sócio-econômica e mesmo ao ensino e à pesquisa (MENEGUETTE, 1994).

Um primeiro questionamento que poderia ser feito se refere exatamente ao que significa SIG (ou GIS, do inglês *Geographical Information Systems*). Por tratar-se de uma tecnologia em franco processo de desenvolvimento, torna-se difícil chegar a uma definição de SIG que satisfaça ao mesmo tempo os envolvidos com seu uso, seu desenvolvimento, e até mesmo àqueles que fazem seu marketing. De acordo com GOODCHILD, 1997, há inclusive os que chegam a considerar SIG como uma ciência, e não como uma ferramenta.

Os Sistemas de Informações Geográficas têm uma gama muito grande de aplicações, havendo inclusive sistemas que, com frequência, usam as mesmas ferramentas de SIG, mas aplicam-nas em situações muito diferentes. Assim, os diferentes grupos de usuários chegam à sua própria definição de SIG.

De qualquer modo, antes de conceituar SIG, é conveniente revisar as definições de sistema, informação geográfica e sistema de informação. De acordo com MENEGUETTE, 1994:

Sistema é o conjunto ou arranjo de elementos relacionados de tal maneira a formar uma unidade ou um todo organizado, que se insere em sistema mais amplo;

Informação geográfica é o conjunto de dados ou valores que podem ser apresentados em forma gráfica, numérica ou alfanumérica, e cujo significado contém associações ou relações de natureza espacial;

Sistema de informação é o conjunto de elementos inter-relacionados que visam a coleta, entrada, armazenamento, tratamento, análise e provisão de informações.

Dentre as diversas definições de SIG, destacamos:

“SIGs constituem um conjunto de ferramentas para coleta, armazenamento, recuperação, transformação e exibição de dados espaciais do mundo real para um conjunto particular de propósitos” (BURROUGH, 1986).

"Qualquer conjunto de procedimentos manuais ou baseados em computador destinados a armazenar e manipular dados referenciados geograficamente" (ARONOFF, 1989);

"Um sistema de apoio à decisão que envolve a integração de dados espacialmente referenciados, em um ambiente para resolução de problemas" (COWEN, 1988);

"Um sistema de informações baseado em computador que permite a captura, modelagem, manipulação, recuperação, análise e apresentação de dados georreferenciados" (WORBOIS, 1995).

De acordo com CÂMARA *et al* (1996) o seguinte esquema representa a arquitetura de um SIG:

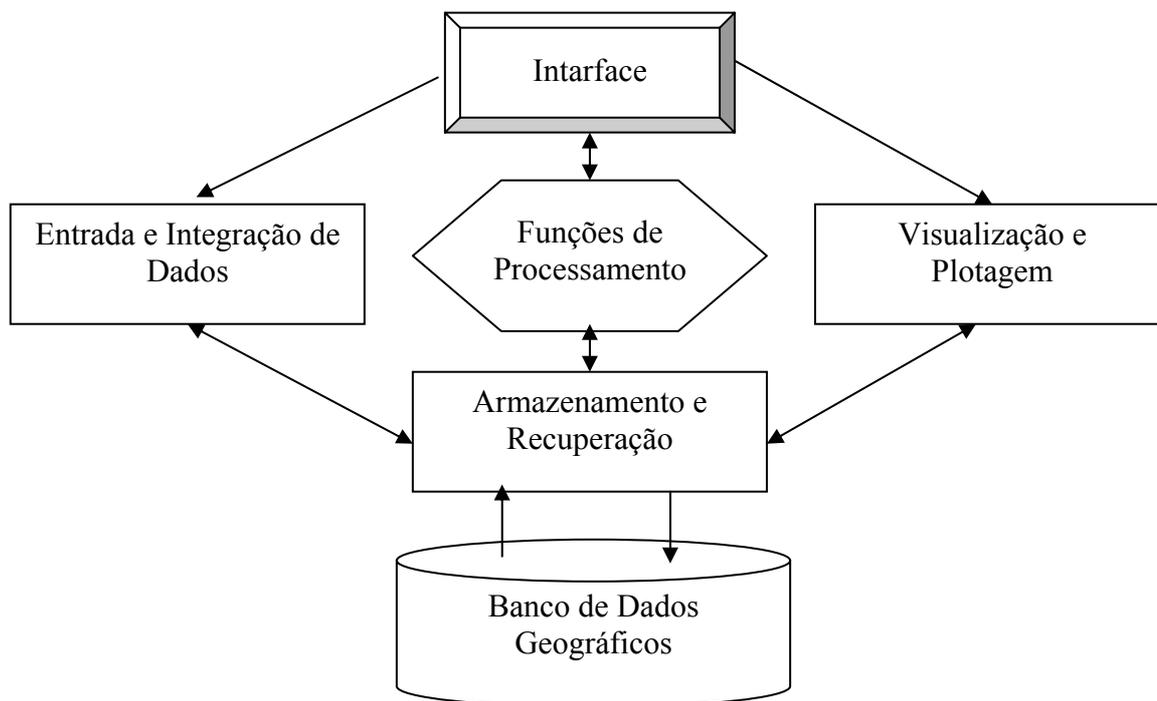


FIGURA 1 -Arquitetura de um sistema de informações geográficas

SIG, com seu conjunto de funções, pode ser visto como um processo ao invés de simplesmente como software e hardware. SIG's servem para tomada de decisão, portanto o modo no qual os dados são inseridos, armazenados e analisados dentro de um SIG deve refletir a maneira pela qual a informação será usada para uma pesquisa específica ou tarefa de tomada de decisão. Assim, ver o SIG como somente um software ou sistema de hardware pode implicar em perder de vista o papel crucial que ele pode desempenhar em um processo amplo de tomada de decisão.

4.2 Web e Mapas

A visualização cartográfica é o processo de tradução de informações espaciais em mapas através da utilização de métodos e técnicas cartográficas (SANTOS *et al*, 2001). Com

o advento dos ambientes digitais a produção de soluções de visualização cartográfica foi colocada diante de novas condições e desafios, envolvendo tanto possibilidades e recursos de expressão antes inexistentes, bem como novas restrições e perfis de utilização.

As instituições que trabalham com informações geográficas têm como um desafio crescente a publicação de dados através da Internet. Por sua natureza gráfica e bidimensional, o ambiente WWW (*World Wide Web*) oferece uma mídia adequada para a difusão da geoinformação. A médio prazo, espera-se que a disponibilidade *online* de grandes bases de dados espaciais e de ferramentas eficientes de navegação torne a geoinformação acessível de forma ampla, sem a necessidade de aquisição de *software* específico. Muitos esforços tem sido realizados neste sentido procurando estender as técnicas já utilizadas para bibliotecas digitais convencionais, que já utilizam a Internet para difundir seu acervo, para Bibliotecas Geográficas Digitais (BGD) onde os objetos a serem armazenados são capazes de representar dados Geográficos (OSSES, 2000).

A publicação de mapas através da rede Internet é um desafio técnico para os profissionais de SIG, de acordo com VASSOLER *et al*, (2001). O autor mostra que a falta de padrões dominantes e o tráfego gerado são obstáculos à disseminação desta fonte de informação. Contudo, à medida em que cresce a capacidade de processamento e de transferência de dados da Internet e se fortalecem iniciativas como as do OpenGis, mais e mais aplicações deverão tornar-se conhecidas do grande público.

O projeto OpenGis, é mantido pela *Open Geospatial Consortium* (OGC) e foi criado para fornecer especificações de interface para o processamento espacial que são abertamente disponíveis para o uso global, fornecendo os requisitos técnicos para interoperabilidade. O Projeto OpenGIS propôs um padrão de arquitetura de *software* para sistemas de informação geográfica (SIG), com base no qual possa se obter interoperabilidade entre produtos de SIG de diferentes fabricantes. Existe uma grande preocupação com relação à qualidade e à integridade espacial dos dados georreferenciados. Grande parte das aplicações geográficas exige a garantia de relacionamentos espaciais específicos na base de dados. Contudo, a maioria dos *softwares* de SIG atuais ainda não oferece suporte adequado para a definição ou garantia de integridade espacial (OGC, 2002).

Dentre as especificações do consórcio que visam promover a interoperabilidade entre dados geográficos, o que mais interessa para a publicação de mapas na Internet concerne ao

que foi publicada pela OGC na forma de *paper* sobre recomendação da *Geographic Markup Language* (GML) baseado na tecnologia XML. Essas linguagens serão abordadas mais adiante.

SANTOS *et al* (2001) destaca que a apresentação de mapas via *Web*, por sua importância crescente, vem merecendo especial atenção dos governos, da indústria e das instâncias de P&D como veículo de disponibilização e distribuição de informações geográficas e espaciais.

Nos últimos vinte anos tem ocorrido um desenvolvimento acelerado nas tecnologias de SIG e ciências correlacionadas com informação geográfica. *Softwares* de SIG são cada vez mais comuns e existem sistemas de diferentes tipos e tamanhos, variando conforme as necessidades dos usuários. Estão surgindo recentemente, diferentes soluções para acesso e distribuição de dados pela Internet.

CÂMARA *et al.* (2000) considera que “*Na maioria dos casos os desenvolvimentos têm sido direcionados para indústria, com algumas exceções, como o IDRISI e o SPRING. A competição entre as empresas de desenvolvimento de softwares e a demanda crescente dos usuários resulta em um número grande de soluções de alta qualidade, que são responsáveis por um aumento significativo na área de mercado de GIS*”.

No entanto, segundo OSSES (2000), ainda não se pode perceber um grande número de instituições disseminando dados geográficos além de suas fronteiras. Isto se deve em parte a problemas técnicos, parte a problemas de organização, e parte a natureza peculiar dos dados geográficos, que são complexos em estrutura e interpretação, rico em interrelacionamentos significativos, e dificuldade de seu entendimento ou uso.

Conforme CÂMARA & MEDEIROS (1998), o ambiente WWW, com o uso de HTML (*Hyper Text Markup Language*) apresenta alguns problemas para uso com banco de dados geográficos, principalmente porque a linguagem HTML é navegacional e não suporta noção de transação. Assim, para poder combinar apropriadamente HTML com ambiente de consulta típico de banco de dados geográficos, é necessário lançar mão, além dos recursos pictóricos do HTML, de uma linguagem de consulta com restrições espaciais. Esses autores afirmam ainda que o grande desafio em montar um ambiente de navegação é garantir rapidez de resposta e interatividade. Para garantir a rapidez é necessário usar mecanismos de generalização, que devem variar conforme o tipo de dados trabalhados.

A *World Wide Web Consortium* é o fórum oficial que desenvolve protocolos que promovem a integração e interoperabilidade na *Web*, garantindo seu crescimento e melhor aproveitamento potencial. Esta instituição tem apontado algumas linguagens específicas que podem, entre outras funções, apresentar informações geográficas na rede Internet. Entre elas destacam-se a XML (*eXtensible Markup Language*), SVG (*Scalable Vector Graphic*) e GML (*Geography Markup Language*), proposta pela OGC e recomendada pela W3C. (W3C CONSORTIUM, 2000).

4.2.1 – Imagens Matriciais e Vetoriais

Um aspecto fundamental na divulgação de mapas na WEB é o formato do arquivo que contém a imagem a ser apresentada. De acordo com MARISCO (2004), outra preocupação está no tamanho do arquivo em (Kb), uma vez que arquivos grandes levam muito tempo para serem baixados pela rede. A impaciência dos navegadores na Internet exige que as imagens e os conteúdos contidos em uma página na Internet levem no máximo 20 segundos para serem apresentadas, caso contrário, esses passam para uma outra página. Em geral, pode-se dizer, que boa parte dos produtos cartográficos estáticos veiculados pela Internet estão em formato padrão matricial ou bitmapas (GIF e JPEG).

Esse formatos de imagens matriciais diferem conceitualmente e estruturalmente dos arquivos vetoriais. Imagens raster são criadas através de processos de escaneamento de um desenho artístico ou “pintando” com um programa de edição de imagens, como o Corel Photo Paint ou Adobe Photo Shop. Já as imagens vetoriais são criadas através de processos de desenho de ilustração vetorial, como o Corel Draw ou Adobe Illustrator. A palavra vetor é sinônimo de linha. Imagens vetoriais também podem ser criadas através de processos de conversão de uma imagem raster usando um programa de conversão como o CorelTrace ou EuroVector.

4.2.1.1 Imagens Raster

Uma imagem raster é uma coleção de pontos chamados pixels. Cada pixel é um minúsculo quadrado colorido. Quando uma imagem é escaneada, é convertida em uma

coleção de pixels, chamada de imagem raster. Gráficos escaneados e gráficos web (como arquivos JPEG e GIF) são as formas mais comuns de imagem raster.

4.2.1.1.1 Resolução

A resolução de uma imagem raster ou imagem escaneada é expressa em termos de pontos por polegada ou dpi (do inglês *dots per inch*). A resolução de impressoras e aparelhos scanner também é medida em dpi. Uma impressora laser típica de computador desktop, possui de 300 a 600 dpi. Mas impressoras com alta resolução (acima de 2.500 dpi) são capazes de produzir saídas limpas e detalhadas. A qualidade de saída de um equipamento de impressão é dependente da resolução da imagem de origem. Então uma imagem de 300 dpi será impressa com a mesma qualidade em uma impressora laser de 300 dpi como em uma plotter com 2.500 dpi. Tomando-se uma imagem de 300 dpi e aumentando o tamanho da imagem, estaremos criando uma imagem com “jaggies”, ou estourando os pixels. O que acontece é que cada pixel quadrado torna-se maior, e suas bordas podem tornarem-se visíveis, prejudicando a qualidade visual. Diminuindo-se o tamanho da imagem, cada pixel fica menor, e a imagem manterá a definição original, embora em menor tamanho, sem prejudicar a qualidade inicial da imagem. Em outras palavras, imagens raster não são muito boas para ampliação. A qualidade de uma imagem raster impressa, é dependente da resolução (dpi) da imagem raster original, da capacidade e tecnologia da impressora e se a imagem teve sua escala aumentada ou não.

4.2.1.1.2 Cor

Quando se escaneia uma imagem colorida, um grande número de cores será necessário para rasterizar a reprodução da imagem original com acurácia. Se for escaneada com esquema de 16 milhões de cores, a maioria dos olhos humanos não percebe a diferença entre a imagem original e a imagem rasterizada. Mas se a mesma imagem fosse escaneada usando uma paleta de 256 cores, seria impossível de reproduzir as cores originais acuradamente. Para contornar isso, alguns scanners usam um processo chamado “dithering” para aproximar as cores que não existem na paleta.

Esse processo produz um padrão de pontos distintos para aproximar a cor original da imagem. Porém, o processo deteriora a qualidade da imagem raster gerada. Para mudar uma determinada cor em uma imagem raster, é necessário isolar uma cor específica ou uma escala

de cores, e instruir o software à mudar tal cor. Isto pode ser um desafio para trabalhar com softwares de edição de imagens.

4.2.1.1.3 Tamanho do arquivo

Para reproduzir uma imagem raster, o software de gráficos precisa guardar uma grande quantidade de informações, incluindo a localização exata e a cor de cada pixel no conjunto de pixels. Isto resulta em grande tamanho de arquivos para as imagens. Resoluções altas e grandes quantidades de cores, produzem arquivos grandes. Uma imagem típica de 2 X 3 polegadas com 150 dpi em preto e branco, gastará menos que 70Kb. O mesmo arquivo salvo com 300 dpi e 24 milhões de cores, terá 100 vezes maior tamanho, algo em torno de 7Mb. Quando se cria imagens raster, o tamanho do arquivo torna-se uma limitação. Arquivos grandes tendem à tornar mais lento a performance da máquina e o trabalho do disco rígido. Transferir estes arquivos pela Internet (acima de 1Mb) requer conexões de alta velocidade, nos dois sentidos, tanto para uploads como downloads.

4.2.1.1.4 Formatos de arquivo

Arquivos raster mais comuns: BMP (Windows Bitmap), PCX (Paintbrush), TIFF (Tag Interleave Format), JPEG (Joint Photographics Expert Group), GIF (Graphics Interchange Format) , PNG (Portable Network Graphic), PSD (Adobe PhotoShop) and CPT (Corel PhotoPAINT). Destes, BMP, JPG e GIF são os mais comuns na Internet.

4.2.1.2 Imagens Vetoriais

Uma imagem vetorial é uma coleção de linhas e curvas conectadas, que representam objetos. Quando se cria uma imagem vetorial em um programa de ilustração, os nós, ou pontos são inseridos e linhas e curvas são desenhadas, conectando-os. Este é o mesmo princípio de “ligar os pontos”. Cada nó, linha e curva é definido no desenho pelo programa de edição gráfica através de uma descrição matemática. Cada aspecto do objeto vetorial é definido por uma equação, incluindo a localização do nó, comprimento e espessura da linha. Objetos de texto são criados por conexão entre nós e linhas e curvas. Cada letra de uma determinada fonte é representada como um objeto vetorial. Imagens vetoriais são orientadas a objeto, enquanto imagens raster são formadas pelos pixels. Em uma imagem vetorial, as cores são como roupas sobre um esqueleto (estrutura de linhas e pontos).

4.2.1.2.1 Resolução

Imagens vetoriais são definidas por equações matemáticas e não por pixels. Elas podem ter a escala aumentada ou diminuída sem nenhuma perda de qualidade. Quando um programa de edição de imagens aumenta ou diminui uma imagem vetorial, ele simplesmente multiplica a descrição matemática de cada objeto por um fator de escala. Já que imagens vetoriais não perdem qualidade quando têm sua escala aumentada ou diminuída, elas podem ser impressas em qualquer resolução que a impressora seja capaz de produzir. Ao contrário das imagens raster, a qualidade não é limitada por pontos por polegada ou pela resolução do scanner.

4.2.1.2.2 Cor

Já que as imagens vetoriais são compostas por objetos e não por pixels, é possível alterar as cores individualmente para cada objeto. Colorir objetos vetoriais é similar a colorir desenhos à lápis em um livro. Um software adequado pode habilitar o usuário à simplesmente clicar no interior de um objeto e definir sua cor, assim como espessura de linhas. Nesse sentido, colorir uma imagem vetorial é muito mais fácil do que uma imagem matricial.

4.2.1.2.3 Tamanho de arquivo

Imagens vetoriais não precisam guardar informações de cada pixel, mas somente as descrições matemáticas de seus objetos. Por essa razão arquivos vetoriais tendem a ser menores. Uma imagem de 5 por 10 cm representada em um arquivo vetorial, terá o mesmo tamanho de arquivo do que a mesma imagem com 60 por 120 cm. O tamanho do arquivo é o mesmo porque a única diferença entre eles seria um número que define o tamanho da imagem. Uma imagem raster precisaria armazenar uma grande quantidade de pixels a mais para representar o aumento da imagem, o que significaria um arquivo bem maior.

4.2.1.2.4 Formatos de arquivo

Arquivos vetoriais comuns: EPS (Encapsulated PostScript), WMF (Windows Metafile), AI (Adobe Illustrator), CDR (CorelDraw), DXF (AutoCAD), PLT (Hewlett Packard Graphics Language Plot File). Os formatos mais comuns para Internet: SVF (Simple Vector Format), DWF (Drawing Web Format), Flash (formato exclusivo da Macromedia),

PDF (Portable Document Format), SVG (Scalable Vector Graphics). A TABELA 1 ilustra as principais características dos formatos de arquivos usados para apresentação de imagens na WEB.

TABELA 1 - Visão geral de formatos de imagens usados na WEB

Formato	Uso	Modulo de Visualização	Tamanho do arquivo*	Formato interno	Nível de interatividade
Matriciais					
JPG	frequente	navegador	111.129 b	binário	0
GIF	frequente	navegador	27.464 b	binário	0
BMP	frequente	navegador	284.726 b	binário	0
PNG	raro	navegador	19.162 b	binário	0
Vetoriais					
SVF	ultrapassado	Plug-in	95.410 b	binário	1
DWF	raro	Plug-in/applet	144.083 b	binário	2
Flash	frequente	Plug-in	4.807 b	binário	3
PDF	frequente	Plug-in	10.374 b	binário/ascii	1
SVG	raro(novo)	Navegador/ Plug-in	19.796 b	ascii	4
WebCGM	raro	Navegador/plugin	6.382 b	binário	2
<p>* Tamanho de arquivo: O tamanho do arquivo (expresso em bytes) é relativo, refere-se à uma mesma imagem, com mesma definição de cores e linhas, gerada com 552 X 514 px, exportada para todos os formatos acima pelo software CorelDraw 11, usando as opções "default" para que se obtenha todas as imagens com a mesma qualidade de saída.</p> <p>Nível de interatividade: 0 = simplesmente exibe. 1 = zoom, layers, links nos objetos. 2 = scripts externos para acessar gráficos. 3 = animação. 4 = total controle dos objetos e animação.</p>					

Fonte: NEUMANN e WINTER (2001), modificada.

Quanto ao tamanho do arquivo, cabe ressaltar que para fins de comparação, gerou-se a mesma imagem em todos os formatos, cuidando para que definição de cores e espessura de linhas fossem equivalentes. A imagem gerada, equivale à um tamanho de impressão de 194,65 X 181,47 mm em todos os formatos. Devido à estas dimensões, na tabela alguns arquivos matriciais ficaram equivalentes ou até menores do que alguns formatos vetoriais. No entanto, se a dimensão da imagem fosse aumentada para o dobro, ou seja, 1104 x 1028 px, o tamanho dos arquivos matriciais aumentaria proporcionalmente (com alguma variação em função da tecnologia de compactação), enquanto os arquivos vetoriais manteriam o mesmo tamanho ou muito próximo, já que os vetores são os mesmos e a única mudança seria o fator de escala. Ainda quanto à tamanho de arquivos, é necessário considerar que o tamanho do arquivo vetorial aumentará proporcionalmente ao nº de vértices da imagem, portanto se o nível de detalhamento de uma imagem (nº de vértices) for muito alto, o arquivo vetorial poderá ser maior do que o equivalente matricial.

4.2.2 Outras tecnologias

Muitos serviços criados recentemente, a partir de novas tecnologias, necessitam da interação entre mapas e Web para apresentar seu produto final. Um exemplo claro disso é o LBS. De acordo com MAGON & SHUKLA (2001) o serviço baseado na localização, ou *Location Based Service* (LBS), pode ser definido como a capacidade de encontrar a localização geográfica de um dispositivo móvel e de fornecer serviços baseados nessa informação de localização. Um exemplo prático seria uma pessoa solicitando uma informação específica, como qual seria a localização do supermercado mais próximo de sua própria localização dentro de uma cidade. Esse serviço pode ser disponibilizado através da Internet ou de uma rede de computadores.

Os autores também afirmam que os serviços de informação desenvolvem-se rapidamente ao longo dos anos e novas dimensões são agregadas, como a capacidade de localização de objetos e até pessoas. A união de sistemas de informações geográficas, Internet, comunicação remota, técnicas de localização e dispositivos móveis, deu origem aos serviços baseados na localização, causando maior impacto na forma de fazer navegação e na condução de negócios comerciais. A evolução apenas começou e muitas oportunidades e tecnologias surgiram, e estão sendo aproveitadas pela área comercial.

Conhecer a própria localização tem sido de suma importância para o homem, que já adaptou vários métodos para isso, ao longa da história. Desde perguntas verbais, marcos no terreno até as estrelas já foram usados para isso. Na idade média, usava-se mapas e compasso para encontrar sua localização. Com o crescimento da tecnologia as necessidades criaram dispositivos mais sofisticados para encontrar a localização, e a seqüência natural está sendo o surgimento de técnicas que permitam tirar proveitos sociais e comerciais dessas tecnologias.

4.3 Linguagens de Marcação

Segundo ALMEIDA (2002), usa-se a palavra "marcação" para descrever anotações ou marcas em um texto, que tem por objetivo dar instruções ao desenhista ou datilógrafo sobre a maneira como uma parte do texto deveria ser representada. O autor cita exemplos, como um sublinhado ondulado que indicaria negrito, símbolos especiais para passagens a serem omitidas ou impressas com uma fonte especial, dentre outras. Com o advento da informática, a formatação e a impressão de textos se tornaram automatizadas, e o termo foi estendido para todos os tipos de códigos de marcação em textos eletrônicos. Todos os textos impressos são codificados com sinais de pontuação, uso de letras maiúsculas e minúsculas, regras para a disposição do texto na página, espaço entre as palavras etc. Estes elementos são um tipo de "marcação", cujo objetivo é ajudar o leitor na determinação de onde uma palavra termina e onde outra começa, ou identificar características estruturais (por exemplo, cabeçalhos) ou simples unidades sintáticas (por exemplo, parágrafos e sentenças). Codificar ou "marcar" um texto para processamento por computadores é também um processo de tornar explícito o que é conjetural. Indica como o conteúdo do texto deve ser interpretado.

Concordando com o exposto, BAX (2001) afirma que marcas inseridas no documento de forma implícita (pelo programa, em resposta a determinado comando do usuário) ou explícita (pelo usuário) indicam **como** o processador deve dispor o texto na página; qual fonte de caracteres usar e muitas outras características tipográficas. Estas marcas ou códigos são tipicamente específicos a um sistema de formatação proprietário. Cada *software* editor ou compilador de textos possui seu próprio conjunto de códigos com significado apenas para

aquele sistema, que deverá rodar em um determinado sistema operacional ou em uma máquina específica.

ALMEIDA (2002) entende por "linguagem de marcação", um conjunto de convenções utilizadas para a codificação de textos. Uma linguagem de marcação deve especificar que marcas são permitidas, quais são exigidas, como se deve fazer distinção entre as marcas e o texto e qual o significado da marcação.

Uma marca ou *tag* é tudo o que não for considerado conteúdo em um documento, segundo BAX (2001). Elas indicam a função (o propósito) da informação no documento, em vez de como ela deve ser apresentada, ou seja, sua aparência física. A idéia básica é a de que o conteúdo do documento deve estar separado do estilo usado em sua apresentação. Cabe, à aplicação que interpreta a linguagem de marcação, formatar o texto em tempo real e apresentá-lo aos usuários. Assim, nos programas de processamento de textos modernos, a marcação procedimental é escondida do usuário, porém ela ainda está lá, presente.

As linguagens de marcação encontram atualmente sua melhor aplicação nos arquivos HTML, mais conhecidos como "páginas da Internet", os quais são interpretados por *softwares* populares (navegadores ou *browsers*), sendo que os mais utilizados no Brasil, são o Netscape e o Internet Explorer.

4.3.1 SGML (*Standard Generalized Markup Language*)

4.3.1.1 Definição

Standard Generalized Markup Language, ou simplesmente SGML, é o padrão internacional para definir a descrição da estrutura de diferentes tipos de documento eletrônicos.

4.3.1.2 Breve Histórico

Esta linguagem foi criada há aproximadamente 30 anos como um esforço para se definir uma linguagem de marcas para a representação de informações em texto (EDWARDS citado por BAX, 2001). A linguagem foi reconhecida como um padrão ISO (8879) em 1986. SGML não é um conjunto predeterminado de marcas, e sim uma linguagem para se definirem quaisquer conjuntos de marcas, uma linguagem *autodescritiva*; cada documento SGML carrega consigo sua própria especificação formal, conhecido como *Data Type Document* (DTD).

De acordo com TURNER (2003), Charles Goldfarb junto com seus colegas, começaram a montar os conceitos do código já em 1969, trabalhando em projetos de pesquisa da IBM. O projeto foi crescendo junto com a expansão da informática, mas somente em 1980, foi publicado o primeiro trabalho sobre SGML, no *American National Standards Institute* (ANSI). Em 1985 um grupo de usuários do SGML foi criado e muitas empresas mostravam interesse pelo código. Na sequência dois importantes projetos aplicaram a SGML: o *Electronic Manuscript Project* da *Association of American Publishers* (AAP), e, em 1987, a documentação do *Computer-aided Acquisition and Logistic Support* (CALS) iniciativa do Departamento de Defesa dos Estados Unidos. A partir desse projeto, a SGML foi requisitada em centenas de postos de comandos do exército americano. Depois disso, devido à sua consistência e praticidade, a linguagem foi adotada como padrão que gerou outras linguagens derivadas, durante a expansão da Internet.

4.3.1.3 Visão Geral

A linguagem de marcação SGML é um padrão internacional, não proprietário e de código aberto, utilizado já há bastante tempo para troca eletrônica de dados e que pode ser utilizada por diferentes sistemas informatizados. Um dos objetivos do SGML é garantir que documentos codificados de acordo com suas regras possam ser transportados de um ambiente de *hardware* e *software* para outro, sem perda de informação. Tanto o HTML (atual padrão em uso na Internet) quanto o XML derivam do SGML e, portanto, apresentam características similares (ALMEIDA, 2002).

4.3.2 HTML (HyperText Markup Language)

4.3.2.1 Definição

É uma linguagem franca para publicação de hipertexto na World Wide *Web*. É um formato não proprietário baseado em SGML, e pode ser criado e processado por uma grande variedade de ferramentas, desde simples editores de texto até sofisticados *softwares* de autoria. A linguagem HTML usa *tags* como `<h1>` e `</h1>` para estruturar o texto em cabeçalhos, parágrafos, listas, hiperlinks, etc. (W3C CONSORTIUM, 2004).

4.3.2.2 Histórico e Objetivos

A linguagem foi desenvolvida no CERN, um centro de pesquisa internacional sobre energia e física, próximo à Geneva, por Tim Berners-Lee e rapidamente se tornou popular com o aparecimento do *browser* MOSAIC. De 1990 à 1995 a linguagem HTML sofreu uma série de extensões por parte de diversos grupos e organizações: HTML 2.0 da IETF, HTML+, HTML 3.0. Depois disso, em 1996, os esforços do grupo de trabalho do World Wide *Web* Consortium, levaram ao aparecimento do standard HTML 3.2. Já em 1997 o HTML 4.0 surgiu como uma extensão do HTML 3.2 que permite a utilização de folhas de estilo (*style sheets* – arquivos no formato “css”), mecanismos de scripts, frames, objetos incorporados, e alguns mecanismos de acessibilidade para pessoas incapacitadas. Finalmente, em 1999 o HTML 4.01 corrige um certo número de erros e incongruências encontradas na recomendação anterior (ROCHA, 2000).

Atualmente a recomendação do W3C é o XHTML 1.0 (*Extensible HyperText Markup Language*). Esta especificação é uma reformulação do HTML 4.01 na linguagem XML (*Extensible Markup Language*), e combina a força e aceitação do HTML 4 com o poder da linguagem XML.

Esta linguagem objetiva publicar documentos que contêm texto, imagens, fotografias, e outras informações, além de incluir ligações para outros documentos da *Web*, que podem ser acessados com um simples *click* do mouse. Também objetiva incluir diretamente nos documentos todo o tipo de aplicações multimídia e incorporar formulários para pesquisa de

informações, realizar transações com sistemas remotos, efetuar reservas, encomendar produtos, etc.

4.3.2.3 Estrutura de um documento HTML

Em geral um arquivo HTML inicia com a especificação da versão de HTML utilizada no documento (*Data Type Document*): HTML 4.01 *Strict* DTD pode conter todos os elementos e atributos que não foram marcados como desaconselhados (*deprecated*):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

```
"http://www.w3.org/TR/html4/strict.dtd">
```

Depois da definição do tipo de documento, todo o resto do documento é englobado pela etiqueta (*tag*) HTML. O cabeçalho contém informação sobre o documento como o título, o autor, palavras-chave que serão utilizadas por motores de busca, etc. A informação contida no cabeçalho não é geralmente apresentada.

Todo o documento html tem que ter um título, que deve ser uma frase curta mas indentificadora do documento.

A linguagem HTML permite que os autores coloquem no documento informação sobre o próprio documento, denominada meta-informação. Por exemplo, é possível indicar o autor do documento ou indicar palavras-chaves.

Na seqüência insere-se o corpo do documento. É nele que se encontra o conteúdo (informação) propriamente dita. A forma como esse conteúdo é apresentado pode variar consoante os dispositivos e o *software* que o usuário dispõe para acessar os documentos. Para os *browsers* gráficos mais comuns, o corpo do documento pode ser encarado como uma "tela" onde se posicionam elementos de texto, cores, figuras, gráficos, etc. No entanto, o mesmo documento pode ser acessado por *browsers* de texto, em que a informação relativa aos elementos gráficos deve estar presente de outra forma. Podemos pensar ainda que esse conteúdo possa vir a ser processado por um sintetizador de voz, ou mostrado no reduzido visor de um notepad.

As folhas de estilo (extensão “css”) devem ser utilizadas para tratar da apresentação da informação (aspectos visuais), para que o corpo do documento contenha o mínimo possível de atributos visuais. Além disso, é possível usar a mesma folha de estilos para vários documentos.

O exemplo à seguir ilustra a estrutura de um arquivo HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
  <TITLE>HTML básico</TITLE>
  <LINK rel="stylesheet" type="text/css" href="estilos.css">
</HEAD>
<BODY>
  ... corpo do documento...
</BODY>
</HTML>
```

4.3.3 XML (*eXtensible Markup Language*)

4.3.3.1 Definição

De acordo com o W3 CONSORTIUM (2000) XML é um simples e muito flexível formato de texto derivado de SGML (ISO 8879) para intercâmbio de dados estruturados.

Atualmente, diversas áreas do conhecimento discutem sobre a possibilidade de melhor aproveitar o grande volume de informações disponível na Internet, transformando-a em algo mais gerenciável e útil. Algumas propostas em estudo contemplam a adoção da linguagem de marcação XML em conjunto com procedimentos complementares (como, por exemplo,

padrões de metadados em formato eletrônico) que permitam conferir elementos semânticos à Internet (ALMEIDA, 2002).

4.3.3.2 Breve Histórico

A linguagem *Extensible Markup Language* (XML) é o resultado do trabalho de um grupo de especialistas estabelecido em 1996 pelo W3C, com o objetivo de propor uma simplificação de SGML que fosse voltada às necessidades específicas da *Web* (BRYAN citado por BAX, 2001).

Já ALMEIDA (2002) coloca que XML é uma linguagem derivada da SGML e foi idealizada por Jon Bosak, engenheiro da Sun Microsystems. Bosak era conhecedor e usuário da SGML e apresentou ao W3 Consortium sua idéia de explorar o SGML em aplicações voltadas para Internet. Em 1996, foi criado o XML, inicialmente como uma versão simplificada do SGML, e, em fevereiro de 1998, o XML tornou-se uma especificação formal, reconhecida pelo W3 Consortium.

4.3.3.3 Visão Geral

Assim como HTML, XML também é uma linguagem expressa em arquivos de texto puro (ASCII), concebida especialmente para armazenar e transmitir dados.

Os modelos ou formatos de dados conhecidos como proprietários possibilitam às empresas que detêm seus direitos alterá-los arbitrariamente, o que lhes oferece grande vantagem, porém em uma rede como a Internet, que conecta diferentes tipos de computadores e plataformas, a informação deve ser acessível, sem restrições impostas por formatos de dados proprietários (BRYAN, 1997).

De acordo com ALMEIDA (2002), XML tem uma característica adicional importante que permite ao autor do documento a definição de suas próprias marcas, conferindo à linguagem XML algumas propriedades semânticas, que possibilitam melhorias significativas em processos de recuperação e disseminação da informação.

A linguagem XML não possui elementos e marcas predefinidas, portanto não especifica como os autores vão utilizar metadados, sendo que existe total liberdade para utilizar qualquer método disponível, desde simples atributos, até a implementação de padrões mais complexos. ALMEIDA (2002) cita como exemplos de padrões de metadados para meios eletrônicos o *ISO11179*, *Dublin Core*, *Warwick Framework*, *RDF — Resource Description Framework*, e *PICS — Platform for Internet Content Selection*. Este último foi originalmente projetado para ajudar pais e professores à controlar o conteúdo acessado pelas crianças na Internet, mas também serve para prover outros recursos, incluindo sinais de código e privacidade. A plataforma PICS é um entre diversos *softwares* de serviços de filtragem que estão sendo desenvolvidos (WWW ISO URL, 1997).

MARCHAL, citado por ALMEIDA (2002), mostra algumas áreas em que o XML pode ser útil no curto prazo, com vantagens significativas em relação ao HTML: na manutenção de grandes *sites*, no intercâmbio da informação entre organizações, no gerenciamento de conteúdo de *sites*, nas aplicações de comércio eletrônico, nas aplicações científicas com o uso de novas linguagens de marcação para fórmulas matemáticas e químicas, em dispositivos computacionais alternativos (como os *palmtops* e *handhelds*).

4.3.4 GML (Geography Markup Language)

4.3.4.1 Definição

A GML é uma codificação XML para o transporte e armazenamento de informação geográfica, incluindo tanto as propriedades espaciais quanto as não espaciais de objetos geográficos (VASSOLER *et al.* 2001).

A GML foi especificada pelo Projeto OpenGis (OGC) e fornece um meio para estruturar a informação geográfica com o objetivo de conseguir a interoperabilidade de uma infra-estrutura de dados geoespaciais. (SANTOS *et al.* 2002). Seguindo a tendência do uso de padrões para intercâmbio de dados, o OpenGIS usa o padrão XML para definir uma forma de codificar dados geográficos e, segundo LIMA (2002), foi para isso que especificou a linguagem GML.

4.3.4.2 Breve Histórico

Segundo VASSOLER *et al.* (2001), essa especificação foi proposta pelo Projeto Open GIS (OGC), em 1999, tornando-se, sua primeira versão, um padrão em maio de 2000, tendo a segunda versão publicada em fevereiro de 2001. Entre os objetivos da GML, estão:

- a) prover um meio de codificar informações espaciais tanto para transferência quanto armazenamento de dados, especialmente no contexto da Internet;
- b) estabelecer os fundamentos para GIS na Internet de maneira modular, para que possa haver desenvolvimento contínuo da linguagem;
- c) ser capaz de separar conteúdo espacial e não espacial de sua apresentação.

4.3.4.3 Visão Geral

As aplicações que visam publicar informações geográficas na *Web* baseiam-se em três esquemas pré-definidos dentro da estrutura da GML (VASSOLER *et al.* 2001): o esquema de geometria (*geometry schema*), o esquema de feições geográficas (*feature schema*) e o esquema de *XLink* (*Xlink schema*). O esquema de geometria inclui definições de tipos para elementos geométricos abstratos e concretos, indo até a definição de tipos complexos para suporte aos tipos geométricos. O esquema de feições geográficas modela as propriedades geométricas como uma associação que liga uma feição a um objeto do esquema geométrico. O esquema de *Xlink* provê atributos para suportar requisitos de ligação, visando referenciar feições específicas num documento GML.

Segundo VASSOLER *et al.* (2001) a GML baseia-se no conceito de orientação a objetos. Para maior clareza no entendimento do modelo, a sua documentação foi construída utilizando a linguagem de modelagem gráfica UML (*Unified Modeling Language*). O autor afirma ainda que a GML foi projetada para separar o que é conteúdo do que é apresentação visual. Uma vez que GML é uma aplicação XML, ela pode ser apresentada em uma variedade de formatos, incluindo gráficos *raster* e vetoriais, texto, som e voz. A geração de saídas gráficas, tais como mapas, é uma das apresentações mais comuns da GML, podendo ser

diretamente compiladas por *applets* gráficas ou apresentadas por meio de uma tecnologia gráfica XML, tal como X3D ou SVG (*Scalable Vector Graphics*).

De acordo com SANTOS *et al* (2001), para desenhar um mapa com GML é necessário transformar o GML em um formato gráfico, o que pode ser feito através de manipulação direta, por exemplo, uma *applet* JAVA usando DOM (*Document Object Model*), ou ainda transformando as Características Simples codificadas em XML em formato gráficos, tais como: SVG, VML (*Vector Markup Language*), ou VRML (*Virtual Reality Markup Language*), os quais também são codificados em XML.

A GML não contém nenhuma informação sobre como as características codificadas podem aparecer, independente de qualquer especificação gráfica. Assim, a representação visual de uma estrutura GML depende do uso de recursos gráficos adicionais. Transformar a GML em SVG, VML ou VRML é recomendado para visualização de dados.

4.3.4.4 Estrutura de um arquivo GML

```
<?xml version="1.0" encoding="UTF-8"?>

  <rdf:RDF xml:lang="en"
    xmlns:rdf="http://www.w3.org/1999/02/22 -rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

    <rdfs:Class rdf:ID="Geometry">
      <rdfs:comment>

        ...

      </rdfs:comment>
    </rdfs:Class>

    <rdfs:Class rdf:ID="Feature">
      <rdfs:comment>

        ...

      </rdfs:comment>
    </rdfs:Class>

    <rdfs:Class rdf:ID="GeometryCollection">
      <rdfs:subClassOf rdf:resource="#Geometry"/>
      <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Container"/>
      <rdfs:comment>

    </rdfs:RDF>
```

4.3.5 SVG (*Scalable Vector Graphics*)

4.3.5.1 Definição

Segundo SANTOS *et al* (2001), a *Scalable Vector Graphics* é uma linguagem de design e animação vetorial para a Internet, que permite a concepção de textos e imagens, bem como de gráficos 2-D acessíveis, dinâmicos e re-utilizáveis em *Extensible Markup Language* (XML).

Resumidamente, pode ser definida como uma linguagem de marcação para descrever gráficos bidimensionais em XML, que usa elementos para descrever textos, trajetórias (de linhas e curvas), e imagens. Uma vez definida a imagem, pode-se transformá-la e manipulá-la de muitas formas usando os recursos da linguagem.

4.3.5.2 Breve Histórico

De acordo com WILKPEDIA (2004), SVG tornou-se uma recomendação oficial da W3C em setembro de 2001. A SVG foi desenvolvida em um longo processo depois que a empresa Macromedia e a Microsoft introduziram a linguagem VML e em resposta as empresas Adobe e Sun Microsystems lançaram uma linguagem dentro da filosofia de *software* livre, conhecido como PGML, do inglês *Precision Graphics Markup Language*, em 1998. Segundo PELKIE, 2001, a linguagem SVG evoluiu do PGML e incorpora recursos de ambas as linguagens. MIRANDA (2003), afirma que o consórcio (W3C) definiu um padrão de divulgação de mapas vetoriais na *Web* usando a especificação (SVG), originalmente especificado pela Adobe como (PGML). Existe também o padrão da Microsoft, *Vector Graphics Markup Language*, mais conhecido pela sigla VML. Mas a opção para quem desenvolve aplicações para a Internet é o SVG, por não ser proprietário.

Segundo SANTOS *et al* (2001), a SVG assim como a GML, é uma aplicação XML recomendada pela W3C, o que dá a essa tecnologia credibilidade. O objetivo desta aplicação é a descrição de gráficos vetoriais 2D, textos e imagens *raster*. Em relação a interatividade SVG propõe uma rica variedade de eventos. Todos os Elementos podem ser animados usando suas “expressões” (cor, preenchimento, forma, transparência, posição, etc.)

4.3.5.3 Visão geral

Contrariamente às imagens constituídas por *pixels* de formato GIF ou JPEG, as imagens vetoriais são obtidas a partir de figuras geométricas, logo é necessário determinar a textura e o comportamento (deslocação, deformação, etc.). Deste modo, a transferência de figuras abstratas e ligeiras é mais rápida. Além disso, a XML baseia-se nas linhas de código em modo de texto. Desta maneira, os arquivos SVG são igualmente concebidos em modo de texto, admitindo a inclusão de palavras-chave e conseqüentemente a sua indexação aos motores de pesquisa. O *Flash*, por sua vez, integra apenas linguagem binária que é hermética, tornando impossível a sua inclusão em endereços na Internet. (TEK – SAPO, 2001).

Na recomendação do W3C, o SVG apresenta-se como uma esperança para a indústria de conteúdos, que contribui para a interoperabilidade na rede Internet. Tal como o *Flash*, também o SVG possibilita a criação de imagens de pequeno tamanho em bytes, que usam o sistema de desenho vetorial, o que é preferível para transmissão pela Internet.

Os visualizadores SVG, como o *SVGViewer* da Adobe, simulam o "algoritmo do Pintor", ou seja, a tela vai sendo pintada na medida que o código é interpretado, podendo ocorrer a sobreposição de imagens se assim estiver definido na seqüência de código. Nessas sobreposições é possível se obter efeitos de transparência. A criação de elementos para posterior associação em grupos também se encontra disponível. Ainda, é possível realizar operações de transformação geométricas elementares através de matrizes 3x3.

De acordo com PRESCOD (2003), SVG possui uma série de vantagens para formatos gráficos em uso como JPEG ou GIF. Por ter um formato de texto simples, os arquivos do SVG são legíveis e, geralmente, menores do que as imagens gráficas comparativas. As imagens SVG possuem recursos de zoom e são escalonáveis, ou seja, os usuários podem aproximar uma área em particular de um gráfico, como por exemplo um mapa, e não ter nenhuma degradação de imagem. Por serem escalonáveis, as imagens do SVG podem ser impressas em alta qualidade em qualquer resolução. Textos dentro de uma imagem baseada em SVG, como o nome de uma cidade em um mapa, podem ser tanto selecionados ou procurados. Aplicativos escritos em SVG podem se tornar acessíveis para descrever a informação visual em texto detalhado. Finalmente, o SVG suporta *scripting* e animação, o que permite gráficos dinâmicos e interativos.

O SVG dispõe de três tipos de objetos gráficos:

- formas gráficas vetoriais (ex. linhas, curvas e áreas contornadas por elas)
- imagem *raster* / digital
- texto

Arquivos em SVG podem ser dinâmicos (com movimento) ou interativos. O *Document Object Model* (DOM) para SVG, que inclui o XML DOM completo, permite animação de gráficos vetoriais via ECMAScript. Eventos como *onmouseover* e *onclick* podem ser direcionados para qualquer objeto gráfico (WIKIPEDIA, 2004).

SVG é uma forma bem diferente de apresentação de gráficos na *Web*, se comparado às formas até então tradicionais, primeiramente porque é uma linguagem baseada em texto para descrever como os gráficos vetoriais podem ser compilados. Isto contrasta com os formatos GIF, JPG e até arquivos Macromedia Flash, que são binários. Como SVG usa XML, é possível gerar o código a partir de qualquer editor de texto (PELKIE, 2001).

4.3.5.4 Compatibilidade:

SVG é baseado em texto e trabalha em conjunto com as tecnologias atuais da *Web*, como HTML, GIF, JPEG, PNG, SMIL, ASP, JSP, e JavaScript, e, através de plugins, é suportada pelos *browsers* mais usados atualmente, como Internet Explorer, Netscape, Mozilla e Opera.

O padrão SVG foi e está sendo desenvolvido e suportado pelas maiores companhias de *software* gráfico do mundo. Sun Microsystems, Adobe, Apple, IBM, e Kodak são algumas das organizações mais conhecidas que estão envolvidas no desenvolvimento do SVG. O *software MapInfo* já disponibiliza meios de exportação para o formato SVG, assim como o *ArcView* também permite o uso de extensões que incluem a possibilidade de exportação para SVG. Espera-se que a SVG atinja o status de formato padrão para distribuição de imagens vetoriais na Internet (PRESCOD, 2003).

Isso garantirá a interoperabilidade entre os produtos de aplicações gráficas distintas. Também é esperado que as próximas versões dos *browsers web* suportem diretamente o SVG,

não havendo mais a necessidade de *plugin*. No universo dos sistemas de informação geográfica, a expectativa é semelhante, no sentido de que as companhias de *Software GIS* também suportem a SVG em suas aplicações para projetos *Web*.

De acordo com WILKPEDIA (2004), o *browser* Amaya suporta a linguagem sem necessidade de plugins. No entanto, outros *browsers* precisam de um plugin específico, como o Adobe SVG Viewer ou Corel SVG Viewer. Uma versão especial do *browser* Mozilla, chamada "Croczilla", suporta atualmente, parte dos recursos SVG. No projeto KDE (pacote de *softwares* para Linux) o *browser* Konqueror tem uma versão que suporta quase todos os recursos de SVG, chamado KSVG. O *browser* da Apple, Safari, também prevê suporte total à SVG no futuro. Programadores da linguagem Java podem usar as ferramentas do projeto Batik SVG Toolkit, para compilar, gerar e manipular gráficos em SVG.

4.3.5.5 Estrutura de um documento SVG

Conforme SANTOS *et al* (2001), a SVG utiliza sistema cartesiano de coordenadas. A origem está situada no canto superior esquerdo da tela. A representação de um mapa requer a inversão do eixo dos valores y ou a colocação do sinal de menos "-" nos valores desse eixo. Ainda, podemos criar um grupo e transformar a coordenada invertendo-a através de uma operação matricial. Assim como ocorre em outros formatos XML, a SVG pode ter sua folha de estilo modificada automaticamente sempre que sua definição é mudada. A SVG é totalmente compatível com CSS (*Cascating Style Sheet*) o que confere maior flexibilidade ao formato SVG na exibição de fontes, cores preenchimentos, linhas e posições.

De acordo com as observações encontradas em W3SCHOOLS (2004), há três maneiras de definir um documento SVG na *Web*:

Como uma página SVG (*standalone*);

Como um elemento incorporado (*embedded*) em um documento XHTML

Como um documento XHTML com declaração do tipo "*namespace*".

Atualmente, as duas primeiras formas são mais usadas. O exemplo seguinte define um arquivo SVG como uma página (*standalone*). O arquivo precisa ser salvo com a extensão “.svg”:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="300" height="300" x="0" y="0">
....
....
....
</svg>
```

A primeira linha no exemplo contém a declaração XML. Como o SVG é uma aplicação XML, deve incluir a declaração XML.

A segunda e terceira linha definem a declaração do tipo de documento (*Document Type Declaration - DTD*). O DTD descreve a linguagem e sintaxe permitidas em SVG.

A quarta linha contém a tag `<svg>`. Esta tag "avisa" o *browser* que se trata de um documento SVG. A tela do documento SVG é definida pelos atributos de largura (`width`) e altura (`height`). Se estas não forem definidas, a tela do documento SVG ocupará as dimensões do *browser*. Os atributos “x” e “y” especificam onde a tela será localizada dentro da janela do *browser*.

A quinta, sexta e sétima linha representam o espaço onde seriam declarados os elementos SVG.

Na última linha `</svg>` encerra o documento.

Como o SVG é baseado em XML, muitos dos mecanismos de busca da Internet não serão capazes de encontrar documentos com extensão SVG. Para resolver isso, o documento SVG pode ser incorporado em uma página XHTML, como no exemplo a seguir:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<body>
<object data="test.svg" width="500"
height="500" type="image/svg+xml">
<embed src="test.svg" width="500"
height="500" type="image/svg+xml" />
</object>
</body>
</html>
```

No exemplo pode-se notar o uso das *tags* <object> e <embed>. De acordo com o padrão XHTML só é necessário usar a *tag* <object>, no entanto, o *browser* Netscape não suporta essa *tag*, o por isso é conveniente usar também a *tag* <embed>.

Usando esse método o documento pode ser encontrado por mecanismos de busca da Internet.

4.3.6 A Relação entre SGML/XML/HTML

Uma linguagem de marcação é diferente das linguagens de programação tradicionais, como: C, Basic, Pascal, Visual Basic, etc. Essas últimas especificam uma maneira para se calcular, executar ações e tomar decisões. Já o SGML e o XML permitem maneiras de descrever o dado para armazenamento, transmissão ou processamento por um programa.

XML, SGML e HTML não têm os mesmos propósitos, conforme evidenciado por ALMEIDA (2002) a seguir:

- **SGML** é o *Standard Generalized Markup Language* definido pela ISO 8879[2] e representa um padrão internacional para definição de estrutura e conteúdo de diferentes tipos de documentos eletrônicos. A SGML pode ser chamada de "língua mãe" e é usada para descrever tipos diferentes de documentos em muitas áreas da atividade humana, desde transcrições de antigos manuscritos irlandeses até documentação técnica para aviões de guerra; de registros de pacientes em unidades médicas até notação musical.

- **HTML** é o *Hypertext Markup Language* definido pela IETF-RFC1866 e consiste de uma aplicação específica do SGML utilizada na Internet. O HTML define um tipo de documento simples, com marcações fixas projetadas para uma classe de relatórios técnicos de uso comum em escritórios, como, por exemplo, cabeçalhos, parágrafos, listas, ilustrações e algumas possibilidades para hipertexto e multimídia. É o padrão atualmente em uso na Internet (IETF, 1995).

- **XML** ou *Extended Markup Language* é uma versão abreviada do SGML, que possibilita ao autor especificar a forma dos dados no documento, além de permitir definições semânticas. Um arquivo eletrônico XML pode conter, simultaneamente, dados e a descrição da estrutura do documento, através do DTD-*Data Type Definitions* (gramáticas que conferem estrutura ao documento XML). O XML obtém benefícios omitindo as partes mais complexas e menos utilizadas do SGML.

Entre os objetivos estabelecidos na especificação da linguagem XML, de acordo com o W3 CONSORTIUM (2000), destaca-se: ser diretamente utilizável na Internet; ser legível por humanos; possibilitar um meio independente para publicação eletrônica; permitir a definição de protocolos para troca de dados pelas empresas (independentemente da plataforma de *hardware* e *software*); facilitar às pessoas o processamento de dados pelo uso de *softwares* de baixo custo; facilitar a utilização de metadados que auxiliam na busca de informações; aproximar "produtores" e "consumidores" de informação.

4.4 Outras linguagens

4.4.1 Linguagem JAVA

4.4.1.1 Definição

É uma linguagem de programação, baseada na linguagem C, desenvolvida pela empresa SunMicrosystems.

4.4.1.2 Breve Histórico

Java foi desenvolvida por um grupo de pesquisadores da SUN Microsystems por volta de 1990, pouco antes da explosão da Internet. Inicialmente elaborada para ser a linguagem-base de projetos de software para produtos eletrônicos, principalmente eletrodomésticos (SCHÜTZER & MASSAGO, 1999). Os projetistas de sistemas de controle desses processadores, descontentes com linguagens convencionais de programação, como C, propuseram a criação de uma linguagem específica para uso em processadores de aparelhos domésticos, como geladeiras e torradeiras. Todo o descontentamento dos projetistas residia no fato de que programas escritos e compilados em C são fortemente dependentes da plataforma para a qual foram desenvolvidos. Como o ramo de eletro-eletrônicos está em constante evolução, a cada novo liquidificador lançado no mercado com um novo processador embutido, um novo programa deveria ser escrito e compilado para funcionar no novo compilador, ou então, na melhor das hipóteses, para reaproveitar o antigo programa, no mínimo ele teria de ser re-compilado para o novo processador.

No início de 1990, Naughton, Gosling e Sheridan começaram a definir as bases para o projeto de uma nova linguagem de programação, apropriada para eletrodomésticos, sem os problemas já tão conhecidos de linguagens tradicionais como C e C++. O consumidor era o centro do projeto, e o objetivo era construir um ambiente de pequeno porte e integrar esse ambiente em uma nova geração de máquinas para "pessoas comuns". A especificação da linguagem terminou em agosto de 1991, e a ela deu-se o nome de "Oak" (Carvalho). Por problemas de copyrigh (já existia uma linguagem chamada Oak) o nome foi mudado em 1995 para Java, em homenagem à ilha de Java, de onde vinha o café consumido pela equipe da Sun.

Em 1992, Oak foi utilizada pela primeira vez em um projeto chamado Projeto Green, que tinha por propósito desenvolver uma nova interface de usuário para controlar os aparelhos de uma casa. Em meados de 1993 os projetos propostos não eram economicamente viáveis, e não se via um grande futuro no desenvolvimento de aparelhos que suportassem essa nova linguagem. Justamente nessa época, a World Wide *Web* estava em seu nascimento, trazendo um novo horizonte para a Internet. (É importante lembrar que a Internet já existia muito antes do surgimento da WWW. A WWW nada mais é que um conjunto de protocolos que permite

um acesso mais amigável aos recursos disponíveis na Internet. Dentre esses protocolos, por exemplo, o mais conhecido em geral é o de transferência de hipertexto [http]). Com o lançamento do primeiro browser do mercado, o Mosaic, ocorreu à equipe de desenvolvimento da Sun que uma linguagem independente de plataforma, segura e robusta como a que estava sendo desenvolvida para eletrodomésticos caberia muito bem na Internet. Com esse novo ânimo, a equipe da Sun desenvolveu um browser totalmente escrito em Java, tendo-o terminado no início de 1995 e denominado-o HotJava. O grande diferencial de HotJava para outros browsers da época é que ele permitia a inserção de programas escritos em Java dentro de páginas HTML comuns. No entanto esse software não se popularizou.

A grande aceitação da linguagem Java veio logo a seguir, quando a Netscape anunciou que sua próxima versão do browser Navigator, iria dar suporte a aplicativos Java embutidos em documentos HTML. Em seguida, a Microsoft anunciou o mesmo para o seu Internet Explorer. A partir daí a linguagem tornou-se muito popular.

Há uma certa curiosidade por detrás do nome dado a essa linguagem de programação. Java é o nome de uma ilha do Pacífico, onde se produz uma certa variedade de café homônimo. A inspiração bateu à equipe de desenvolvimento ao saborear esse café em uma lanchonete local. Deram-se conta de como era extremamente apreciado por profissionais da área de software (ao menos nos Estados Unidos), de modo que não foi menos justo fazer-lhe homenagem ao batizar uma nova linguagem de programação. Além disso, a linguagem surgiu num ambiente de certa “rivalidade” com a Microsoft, cujo sistema operacional (Windows) oferecia, na época, uma proteção de tela muito famosa, onde eram exibidas “torradeiras voadoras”, um eletrodoméstico conhecido, mas não mais do que uma xícara de café... (SCHÜTZER & MASSAGO, 1999).

4.4.1.3 Visão Geral

Tendo sido originalmente concebida para o desenvolvimento de pequenos aplicativos e programas de controle de aparelhos eletrodomésticos e eletroeletrônicos, Java mostrou-se ideal para ser usada na rede Internet. O que a torna tão atraente é o fato de programas escritos em Java poderem ser executados virtualmente em qualquer plataforma, mas principalmente

em Unix, Mac e Windows. Em meio a essa pluralidade, Java é um idioma comum, falado por todos (BARRON ESTRADA, 2004). Isto significa que Java é ideal para expressar idéias em forma de programas universalmente aceitos. Soma-se a isso o fato de programas Java poderem ser embutidos em documentos HTML, podendo assim ser divulgados pela rede. Diferente da linguagem C, não é apenas o código fonte que pode ser compartilhado pela rede, mas o próprio código executável compilado, chamado bytecodes.

Em contraste com a letargia de documentos tradicionais, Java acrescenta a interatividade entre o usuário e o documento que está sendo consultado, tornando-o mais expressivo, agradável e surpreendente. Java é ideal para a elaboração de material educacional, pois permite ilustrar claramente os conceitos enquanto possibilita um ensino individualizado.

Conforme SCHÜTZER & MASSAGO (1999), essa linguagem possui estrutura muito semelhante à da linguagem C, da qual descende imediatamente. Java tem em comum com a linguagem C++ o fato de ser orientada a objetos e mantém com esta uma alto grau de semelhança. Esse paradigma de programação consiste de um grau a mais na abstração da programação, em comparação com a programação estruturada, e tem se mostrado extremamente útil na produção de programas cada vez mais sofisticados, em menor tempo e com maior qualidade. A programação orientada a objetos é hoje universalmente adotada como padrão de mercado, e muitas linguagens tradicionais foram aperfeiçoadas para implementar esse paradigma, como C++, Object Pascal, etc.

Atualmente, o site JavaSoft mantém informações atualizadas sobre o desenvolvimento da linguagem Java e suas relações com o mercado, assim como utilitários e ferramentas disponíveis para serem baixados gratuitamente.

4.4.1.4 Como Java é executado

Um programa fonte escrito em linguagem Java é traduzido pelo compilador para os bytecodes, isto é, o código de máquina de um processador virtual, chamado *Java Virtual Machine* (JVM). A JVM é um programa capaz de interpretar os bytecodes produzidos pelo compilador, executando o programa cerca de 20 vezes mais lento do que C. Pode parecer ruim, mas é perfeitamente adequado para a maioria das aplicações. Com isto, um programa Java pode ser executado em qualquer plataforma, desde que esteja dotada de uma JVM. É o

caso dos programas navegadores mais populares, como o Netscape Navigator e o Internet Explorer, que já vêm com uma JVM. A vantagem desta técnica é evidente: garantir uma maior portabilidade para os programas Java em código-fonte e compilados. Porém, as JVM tendem a ser programas extensos que consomem muitos recursos, restringindo assim o tamanho das aplicações escritas em Java.

Atualmente, já existem compiladores capazes de traduzir bytecodes para instruções de máquina nativas, como o Just In Time compiler (ou JIT), tornando os programas ainda mais rápidos. Este compilador requer uma versão específica para cada plataforma onde se pretende que o programa Java seja executado. Em contrapartida à maior velocidade de execução está também uma maior necessidade de memória, pois os bytecodes compilados, em geral, ficam três vezes maiores do que o original. Uma alternativa bem mais interessante, e talvez muito mais viável, é a implementação da JVM em hardware na forma de uma placa ou microchip. A primeira iniciativa neste sentido é da Sun Microelectronics, que está produzindo os chips picoJava ITM, microJava™ e UltraJava™. Estes são capazes executar diretamente bytecodes, acelerando em milhares de vezes a velocidade de execução. Isto permitirá o desenvolvimento viável de aplicativos cada vez mais complexos, abrangentes e funcionais.

4.4.1.5 O que são Applets

As applets são pequenos programas Java que podem ser inseridos dentro de páginas HTML. Com este recurso, uma página torna-se dinâmica, podendo interagir com o usuário que a consulte. Um applet pode ainda executar tarefas complexas, como realizar cálculos e apresentar gráficos, sons e imagens em movimento (SCHÜTZER & MASSAGO, 1999).

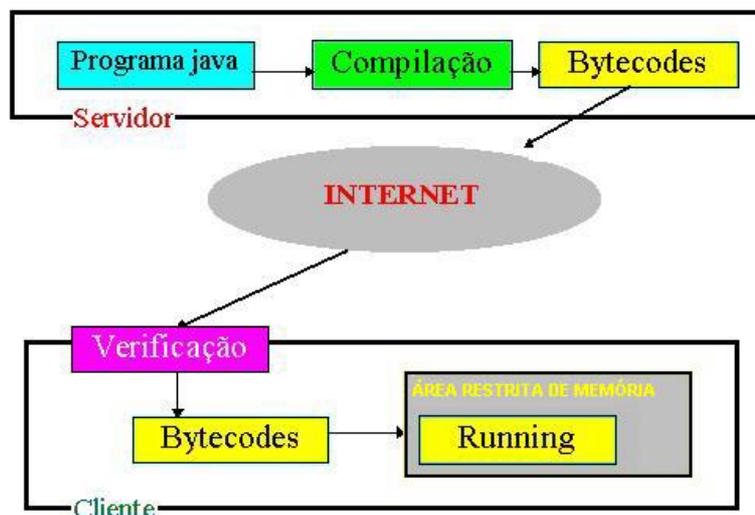


FIGURA 2 - Transmissão de Applets Java

Uma vez que os programas Java são transmitidos como bytecodes, eles podem rodar em qualquer computador sem necessitar uma nova recompilação, independente da plataforma na qual ele será executado. Programas Java sempre são carregados no computador cliente e nele executado. Os Applets são especialmente confeccionados para executarem dentro de uma página HTML. Ao abrir uma página HTML que tenha inserida em si uma applet Java, esta é automaticamente descarregada para o computador cliente e executada.

4.4.2 JavaScript

4.4.2.1 Definição e origem

JavaScript é uma linguagem para páginas da *Web*, desenvolvida pela empresa Netscape. Com essa linguagem, é possível adicionar recursos dinâmicos às páginas HTML, ou seja, é possível ao internauta interagir com a página.

De acordo com ARTÍFICE DA *WEB* (2004), o JavaScript é uma linguagem de programação simples criada para dar mais interatividade e maior funcionalidade às páginas da *Web*. Tendo sido inicialmente desenvolvida pela Netscape, a linguagem JavaScript acabou por dar origem à especificação técnica ECMAScript, que é um padrão oficial reconhecido pela indústria. Os autores SCHÜTZER & MASSAGO (1999), mostram que o JavaScript foi desenvolvido pela Netscape, com o intuito de capacitar a linha de produtos desta empresa (browser e *Web Server*) de uma linguagem básica de scripting. Já BARRON ESTRADA (2004), afirma que JavaScript é baseado na linguagem Java, que por sua vez tem as suas origens nas linguagens de programação C e C++, a sintaxe de programação de JavaScript é semelhante, senão idêntica, às utilizadas por estas linguagens de programação.

ECMA (*European Computer Manufacturers*) é uma associação industrial internacional, fundada em 1961, com sede em Geneve, e se dedica à padronização dos sistemas de informação e tecnologia de comunicação (*Information and Communication Technology - ICT - Systems*).

4.4.2.2 Visão Geral

O código escrito em JavaScript destina-se a ser executado pelo *web* browser quando a página HTML que o contém é visualizada. Ele é uma parte integrante da página e permite que o browser seja capaz de tomar decisões quanto ao modo como o conteúdo é apresentado ao usuário e como pode ser manipulado (BARRON ESTRADA, 2004). Naturalmente, o browser precisa suportar a linguagem. Atualmente, todos os browser suportam JavaScript, com pequenas diferenças quanto às versões.

Pela semelhança dos nomes, não é raro que internautas pensem que o Javascript tem alguma relação com a linguagem de programação Java. Isso não ocorre. Desenvolvida pela Sun, Java é uma linguagem que também possui recursos para *Web*, mas pode ser usada para outros fins e é bem mais complexa.

4.4.2.3 Estrutura do JavaScript

O exemplo à seguir ilustra a estrutura do código JavaScript dentro de uma página HTML:

```
<html>
<head>
  <title></title>
</head>
<body>
<script type="text/javascript">
var nome = "Fulano"
document.write(nome)
document.write("<h1>"+nome+"</h1>")
</script>
  <p>Neste exemplo declara-se uma variável,
    atribui-se-lhe um valor e escreve-se esse valor
    na página.
  </p>
<script type="text/javascript">
var nome = "Fulano"
document.write("<p>O "+nome+" de tal</p>")
</script>

  <p>Uma vez atribuído um valor à variável podemos
    usá-la várias vezes para representar o valor
    que lhe foi atribuído.
  </p>
</body>
</html>
```

O script inicia na tag `<script type="text/javascript">` e termina em `</script>`. A sintaxe é muito semelhante à linguagem C e Java.

4.5 Disseminação de Informações Geográficas na Internet

Uma das linhas de pesquisa e desenvolvimento que vem despertando o interesse da comunidade científica é o acesso a dados geográficos via Internet. Assim como outras linhas

de desenvolvimento recente, existem diversos tipos de enfoque, mas nenhum “vencedor” claro, ou seja, nenhum que se destaque como padrão mais usado ou com vantagens claramente definidas sobre os demais.

De acordo com OSSES (2000), duas alternativas comuns de disseminação de dados geográficos na Internet, são:

1) **Os Servidores Remotos de Mapas**, que, respondendo a pedidos remotos, enviam uma imagem (matriz) de tamanho fixo geralmente em formato GIF ou JPEG. CÂMARA *et al.* (1996) afirmam que essa tipo de alternativa pode usar o browser para apresentar um formulário para preenchimento. Neste formulário são solicitadas informações quanto à região geográfica de interesse (muitas vezes solicitando uma referência explícita a um número de mapa), à composição do mapa (camadas que deverão aparecer) e mesmo alguns elementos de composição visual (cores, espessura de linhas, cores ou hachuras de preenchimento). Quando o usuário termina o preenchimento do formulário, as informações são transmitidas para um servidor, que recupera os dados necessários e converte o mapa final para um formato de imagem, como GIF ou JPEG. Esta imagem é então inserida numa página *Web* criada instantaneamente, e transmitida para o usuário. Este processo é talvez o mais natural do ponto de vista dos browsers, uma vez que lida com a apresentação de imagens, coisa que qualquer browser é capaz de fazer.

De acordo com OSSES (2000), esta solução permite configurar o servidor para responder a diferentes tipos de consulta, sem requerer que todos os dados a ser transmitidos sejam pré-computados. Entretanto, o usuário consegue visualizar apenas as imagens enviadas; qualquer novo pedido é enviado de volta para o servidor, resultando em mais uma transferência pela Internet. Dependendo da velocidade de acesso, esta estratégia pode resultar em longos e sucessivos períodos de espera. O *Internet Map Server* da ESRI é um exemplo. No entanto, esta alternativa apresenta algumas desvantagens. Em primeiro lugar, porque não deixa o usuário “navegar” interativamente pelo mapa. Além disso, a transmissão de imagens é em geral demorada, e realizada de forma repetitiva tende a sobrecarregar os recursos de rede. Por fim, existe o problema de sobrecarga no servidor, que precisa construir o mapa em formato imagem, geralmente a partir de um banco de dados vetorial, e transmiti-lo para o cliente. Note-se que qualquer operação simples, como zoom ou pan, exige a formação de um novo mapa-imagem e nova transmissão.

CÂMARA *et al.* (1996) mostram que outra interface usada é a que apresenta para o usuário um mapa chave, solicitando que indique, com o mouse, qual é a região de seu interesse. Esta abordagem permite um grau um pouco maior de flexibilidade, mas não resolve os problemas principais da alternativa anterior, ou seja, custos de processamento e transmissão, além de não resolver completamente o problema de navegação. Em ambos os casos, o grau de interatividade com o usuário na escolha da região desejada é muito baixo, e o resultado está restrito a um modelo de fragmentação do espaço que foi previamente decidido.

2) **Clientes de Apresentação**, que adotaram como solução a transmissão de todos os dados no formato vetorial para a máquina do cliente, com posterior visualização local. Estes servidores encapsulam a informação em formatos gráficos, que podem ser apresentados por meio de programas adicionais (*plug-ins*) acoplados a *browsers* como o *Netscape* ou o *Internet Explorer* ou por meio de *applets* JAVA. Esta estratégia permite uma maior flexibilidade do lado do cliente, que pode realizar operações locais de visualização e consulta sob os dados transferidos. O tempo de acesso inicial para transferência é maior que no caso anterior, mas muitas das operações posteriores serão realizadas localmente, o que resulta usualmente em um tempo de resposta médio melhor. Exemplos são os produtos *Geomedia Web Map* da INTERGRAPH, *Map Guide* da AUTODESK e *SpringWeb* do INPE. De acordo com CÂMARA *et al.* (1996), a transmissão de objetos geográficos com representação vetorial é mais interessante do que a transmissão de imagens. Desta maneira, o usuário poderia ser livre para decidir a região de interesse, bem como para ativar ou desativar as camadas que deseja. Idealmente, os objetos vetoriais transmitidos seriam armazenados na memória da máquina cliente, para que pudessem ser reaproveitados no caso de operações de zoom ou pan, ganhando tempo para aumentar a interatividade. Outra possibilidade interessante é a aplicação ao mapa vetorial do conceito de “hipermapa”, simulando nos símbolos e objetos vetoriais disponíveis a operação dos links de hipertexto comuns nas páginas da *Web*. Assim, bastaria por exemplo clicar sobre o símbolo de um hospital para consultar seus dados alfanuméricos associados.

No entanto, conforme OSSES (2000), as duas alternativas apresentam problemas. No primeiro caso, todos os dados ficam armazenados no servidor, e no segundo, todos precisam ser transferidos para o cliente. A transmissão de dados geográficos em formato vetorial pela Internet apresenta ainda outro obstáculo: nenhum dos *browsers*, (as ferramentas de navegação na Internet), está preparado para receber e apresentar informações neste formato. Para que isto seja possível, existem duas alternativas. A primeira, que vem sendo adotada por diversos

desenvolvedores de SIG, consiste em criar um plug-in, ou seja, um programa que funciona no computador do usuário, conectado ao browser. Este plug-in reconhece os dados vetoriais à medida em que chegam, geralmente agrupados em um arquivo com extensão padronizada, e os exibe na tela. Esta alternativa tem a desvantagem de exigir a transmissão (download) dos plug-ins a partir do site do desenvolvedor, o que pode ser uma operação demorada (os plug-ins mais comuns têm por volta de 1 Mbyte). Além disso, exige a execução de um procedimento de instalação. Como os plug-ins são específicos para os principais browsers do mercado, que estão em constante evolução, é preciso atualizá-los periodicamente.

Seria mais conveniente dispor de configurações cliente-servidor, que pudessem balancear os pedidos de consulta, permitindo uma apresentação e navegação local em parte dos dados e realizando acessos remotos ao servidor apenas quando estritamente necessário.

De acordo com VASSOLER *et al* (2001), todas essas soluções baseiam-se em componentes de *software* dentro de uma arquitetura do tipo cliente/servidor. Cada tecnologia emprega um ou mais formatos de dados “nativos”. É importante entender que disponibilizar informações utilizando estes componentes significa manipular dados, ou seja, visualizar e algumas vezes até modificar, porém sem permitir que os dados sejam transferidos para a máquina local para posterior análise.

Conforme demonstrado por OSSES (2000), aplicações de Internet utilizam o modelo cliente-servidor que inclui um programa cliente, denominado navegador ou “*browser*”, que se comunica com o servidor através de um protocolo HTTP (*Hyper Text Transfer Protocol*) e faz uma requisição. O servidor processa a requisição e retorna o endereço de uma página HTML através de uma URL (“*Universal Resource Locator*”) que é então enviada ao cliente. Atualmente os navegadores mais comuns no mercado são o *Internet Explorer* e o *Netscape Navigator* e os mesmos fornecedores disponibilizam os seus servidores o *Microsoft Internet Information Server* e o *Netscape Server* respectivamente.

OSSES (2000), afirma ainda que normalmente o modelo utilizado para disseminação de dados geográficos é uma extensão do conceito cliente-servidor conhecido como servidor multi camada (*multi-tiered*). Nesta arquitetura o cliente é normalmente um navegador da Internet ou até mesmo um *software* de SIG completo. O servidor multi-camada consiste de um servidor HTTP e um programa SIG em modo de execução que se comunicam através de um *software* especializado. Quando o usuário (cliente) faz uma requisição ao SIG, a

mensagem é enviada pela Internet ao servidor HTTP. O servidor reconhece esta mensagem repassando-a em seguida ao programa que faz a interface entre ele e o SIG. Este programa converte a mensagem em comandos que são enviados ao SIG que por sua vez retorna os resultados para serem passados de volta ao servidor HTTP. Os dados que o servidor HTTP retornará ao cliente deverão estar num formato padrão Internet ou num padrão que possa ser entendido por um *plug-in* do navegador (programa acoplado), ou um *applet* JAVA (um módulo que estende a funcionalidade de um navegador), conforme a arquitetura do sistema utilizado.

A tecnologia de objetos *ActiveX* do lado cliente e a tecnologia de programação via *JavaScript* são largamente utilizados. VASSOLER *et al.* (2001) apontam que uma vantagem da tecnologia *ActiveX* é o maior poder de processamento que pode ser empregado do lado do cliente, diminuindo assim o tráfego de informações na rede. Em contrapartida, arquiteturas baseadas em *JavaScript* deixam de exigir tal componente específico do lado do cliente. Podem, assim, ser interpretadas por qualquer *browser* que “fale” *JavaScript*.

Em consonância com OSSES (2000), VASSOLER *et al.* (2001) afirmam que de uma maneira geral, do lado do cliente, executa-se uma aplicação em um *browser web*. A aplicação pode ser escrita em JAVA, *JavaScript*, ser um Componente *ActiveX*, um *plugin* ou simplesmente uma página HTML. Por meio desta aplicação, o usuário pode fazer manipulações na forma de exibição dos dados geográficos apresentados, numa maior ou menor interatividade (arrastar, ampliar, etc), dependendo do tipo de tecnologia empregada. Uma camada intermediária é necessária para receber as requisições feitas quando o usuário executa uma ação. Estas requisições chegam a esta camada na forma de uma URL e o servidor de HTTP é quem vai analisá-las e repassá-las para o servidor de mapas apropriado.

Do lado do servidor, executa-se uma aplicação que se conecta à base de dados relacional onde se localizam as informações textuais, multimídia (som, imagens e vídeos), etc. Esta aplicação também se conecta à base de dados espaciais onde estão localizados os mapas. De posse da requisição enviada pelo servidor HTTP, o servidor de mapas fará o processamento necessário e irá gerar uma saída. Esta saída será passada para o servidor HTTP, que a transmitirá para o usuário que fez a requisição. (VASSOLER *et al.* 2001).

5 Desenvolvimento do trabalho

Esse capítulo aborda a metodologia usada para gerar os protótipos propostos. Resumidamente, os passos são:

- Seleção dos mapas para publicação
- Geração do mapa em formato SVG
 - usando ArcView (script em Avenue)
 - usando conversor (JAVA)
- Adequação do código para Protótipo SVG simplificado
- Adequação do código para Protótipo Geoclient modificado

5.1 Seleção dos mapas para publicação

Uma vez caracterizado o problema de como mostrar um mapa digital para usuários que não possuem software SIG, e, mais do que isso, fazê-lo na Internet, cria-se um desafio tecnológico. As instituições de pesquisa precisam enfrentar essa desafio para disseminar as informações transformadas em mapas digitais, que foram geradas em suas ações. Nesta proposta usa-se a linguagem SVG como ferramenta para visualização desses mapas.

O primeiro ponto a ser abordado é a definição de quais mapas digitais existentes na instituição devem ou podem ser disseminados na Internet, partindo da idéia que muitos mapas são gerados apenas para satisfazer uma determinada etapa do processo de pesquisa, e são por isso, desprovidos de elementos que permitam fácil compreensão de seu significado, como legenda clara e concisa, cores adequadas, classes perceptíveis e outros. Normalmente a falta desses elementos não representa um problema para o pesquisador que necessitou gerar o mapa, pois ele conhece profundamente o tema em questão e consegue extrair dele as

informações que precisa. No entanto, se tal mapa fosse apresentado para um leigo, poderia levar à sérios prejuízos na compreensão de seu significado, desqualificando-o, portanto, para apresentação pública. Isto significa que é necessário realizar uma triagem e escolher somente aqueles mapas digitais que apresentem um significado claro por sua natureza e que sejam úteis ao usuário final, pela informação que é capaz de evidenciar. Normalmente esses mapas são produzidos nas últimas etapas da pesquisa, e representam o resultado obtido nela.

É necessário considerar também a escala de apresentação inicial do mapa na tela, ainda que o método aqui proposto permita determinado nível de zoom, para que a generalização seja compatível com o produto que se pretende apresentar. Quanto menor o nível de detalhe, menos vértices existirão nos arquivos eletrônicos que serão transmitidos, o que pode representar significativo ganho de performance (aqui entendido como velocidade na transmissão de dados). Por exemplo, um mapa temático que tenha sido gerado a partir de uma base cartográfica contendo a divisão política do Estado, com escala original 1:250.000, apresentaria um determinado nível de detalhe nas linhas de divisa municipal do mapa, herdado da base cartográfica. Supondo que o mapa temático fosse apresentado na Internet evidenciando um fenômeno qualquer que abrange todo o Estado, e supondo que para o usuário final houvesse interesse apenas em visualizar o fenômeno em todo o Estado ou em macro regiões, então o mapa deveria sofrer generalização, pois o número de vértices seria diminuído, sem prejuízo para a percepção visual.

Outro aspecto importante na seleção dos mapas é a quantidade e qualidade dos atributos associados aos elementos espaciais incluídos em cada tema. Deve-se considerar quais são os atributos textuais que realmente interessam ao usuário, excluindo informações inúteis ou desnecessárias. Um mapa temático sobre potencial de poluição com dejetos de suínos, por exemplo, poderia apresentar diversos pontos que representam as instalações de cada granja de suínos. O software que gerou o mapa, em função das operações de geoprocessamento, poderia ter acrescentado aos atributos de cada ponto o respectivo perímetro e área. Tal perímetro refere-se somente ao pequeno círculo desenhado no mapa (ponto) e não tem nada haver com a instalação física que ele representa. Portanto esses atributos não interessam ao usuário final, e devem ser eliminados, diminuindo o tamanho do banco de dados associado ao tema. As informações realmente relevantes nesse caso, seriam: número de animais, número de matrizes, tipo de criação, volume de dejetos produzido, proprietário, integrador, área da granja e outras informações pertinentes.

5.1.1 Caracterização do tema de trabalho

Uma vez considerados os cuidados na seleção do material a ser exibido na Internet, adotou-se como exemplo para demonstração do método, o Inventário de Terras da Sub-bacia do Rio Ariranha (Oeste de Santa Catarina). Este trabalho foi realizado pela Epagri (Empresa de Pesquisa Agropecuária e Extensão Rural de SC) CIRAM – Centro Integrado de Recursos Ambientais. Trata-se de um levantamento metodológico detalhado das terras da Sub-bacia que visou caracterizá-la quanto à questão ambiental, já que a região é grande produtora de suínos e conseqüentemente geradora de dejetos. Além disso, outros aspectos ligados à agricultura e sócio-econômicos também foram inventariados e agregados ao trabalho, criando uma excelente ferramenta de planejamento regional.

A Sub-bacia do rio Ariranha fica no Oeste do Estado de Santa Catarina, entre 27° e 27°13' de latitude sul e entre 52°12' e 52°28' de longitude oeste, abrangendo parte dos municípios de Seara, Ipumirim, Paial, Xavantina e Arvoredo. A figura a seguir mostra a localização da sub-bacia no Estado de Santa Catarina.

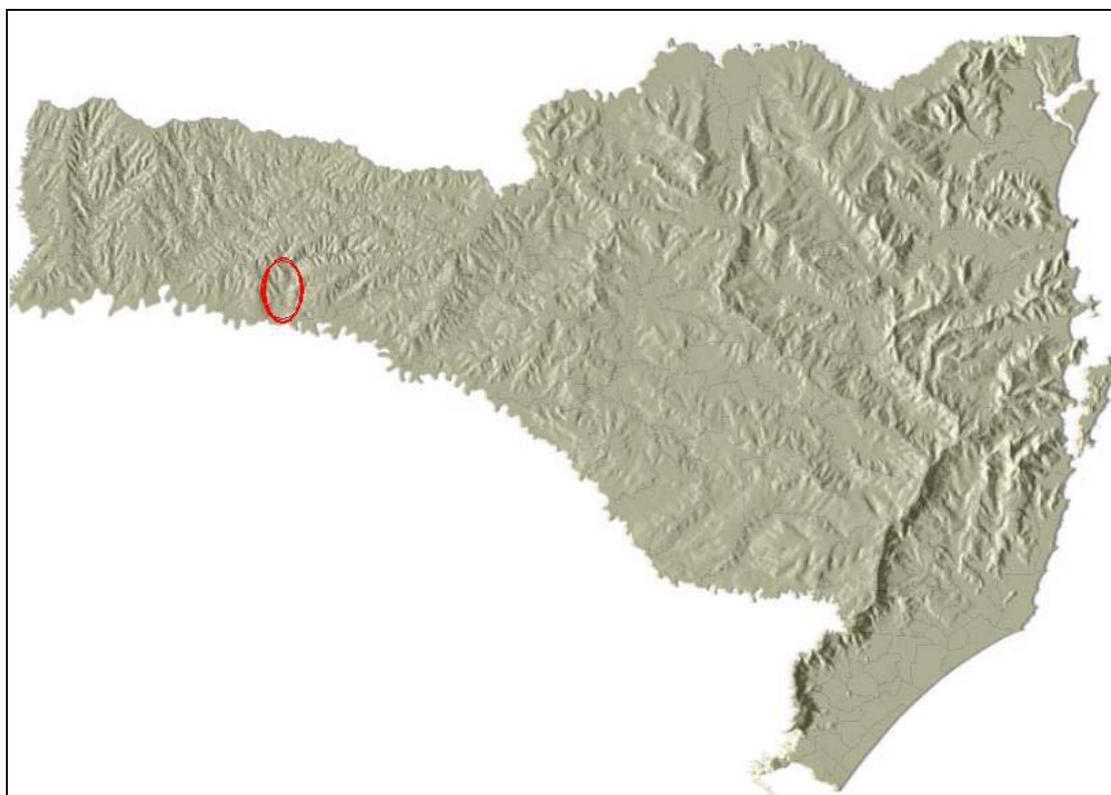


FIGURA 3 – Localização da área de estudo no Estado de Santa Catarina.

Os municípios e a área da microbacia são apresentados na figura a seguir:

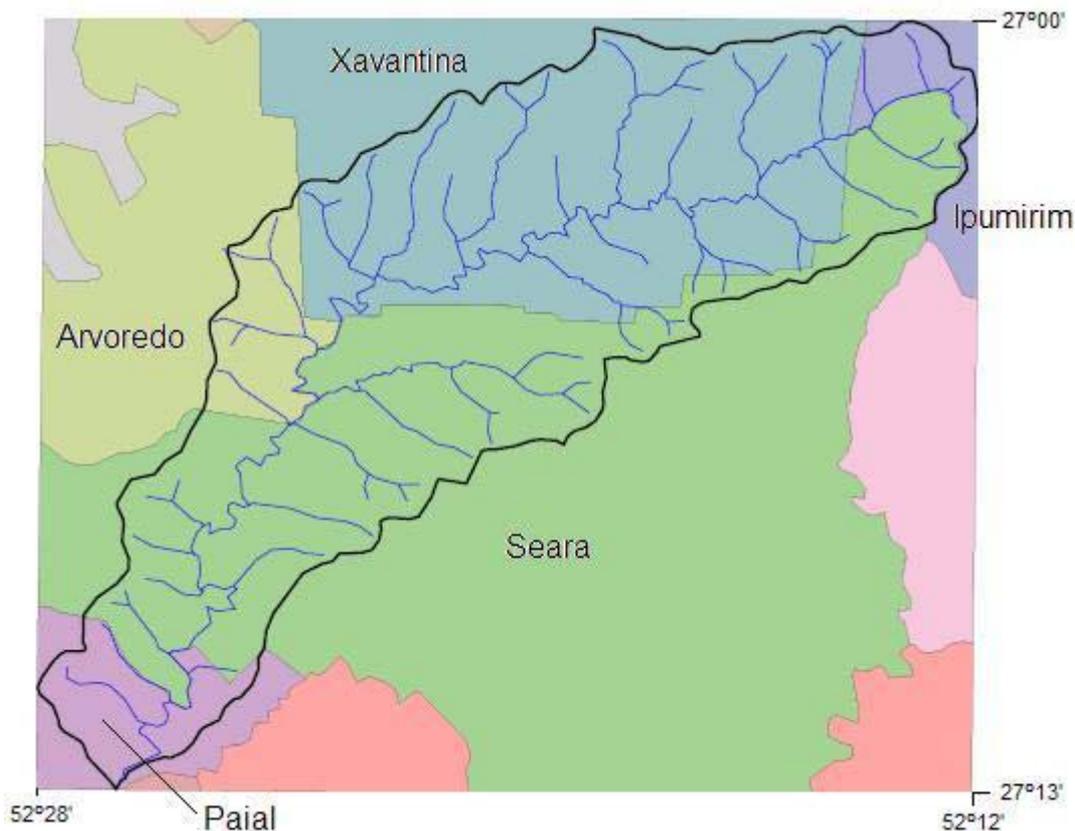


FIGURA 4 – Localização da área de estudo no Oeste de Santa Catarina.

O trabalho de inventário foi bastante amplo, incluindo participação de técnicos no campo e escritório, tendo sido iniciado em 2001 e concluído no final de 2002. Todo o levantamento foi baseado no uso de SIG.

5.1.1.1 Material usado no levantamento

Utilizou-se como material básico aerofotos pancromáticas em escala aproximada 1:25.000 (vôo realizado pela Cruzeiro do Sul - Levantamentos Aerofotogramétricos, de 1977 a 1979), Folha 25, faixa 1 a 7,10 e 13; fotos n.º 29436 a 29443, 26469 a 26477, 26744 a 26755, 26447 a 26458, 26318 a 26323, 26308 a 26312, 26149 a 26144, 32293 a 32298 e 25850 a 25855, respectivamente. Como base cartográfica foi utilizada a folha topográfica de Concórdia (SG-22-Y-D-I) elaborada pelo DSG - Ministério do Exército, 1972, em escala 1:100.000, ampliada para 1:25.000. A imagem de satélite utilizada foi LANDSAT TM7 da

órbita 222.79, bandas 3, 4 e 5 de 24 de setembro de 1999 (EPAGRI, 2003). Os softwares utilizados foram: ILWIS versão 2.23 for Windows, produzido pelo ITC (*International Institute for Aerospace Survey and Earth Sciences*), Enschede, Holanda; Arc Info 7.1.2 for Windows N.T. 4.0 produzido por ESRI (*Environmental Systems Research Institute, Inc.*) Califórnia, U.S.A. e ARCVIEW 3.0 também da ESRI. Os receptores GPS usados para georeferenciamento das sedes das propriedades foram os ETREX SUMMIT, da americana Trimble. As curvas de nível foram obtidas do IBGE, em meio digital no formato SIG e importadas para o ILWIS e atribuídos valores. Localidades, sedes municipais e rodovias foram digitalizadas a partir das cartas topográficas do IBGE (EPAGRI, 2003). Os mapas básicos utilizados neste trabalho foram os mapas planialtimétrico, hidrográfico e rodoviário, compilados através de carta topográfica na escala 1:100.000 e acrescidos de informações complementares obtidas a partir de interpretações de aerofotos, imagens de satélite e atualização a campo.

5.1.1.2 Divulgação dos resultados do Inventário

Originalmente optou-se por divulgar os resultados do Inventário através de CD-ROM, veiculando os mapas em formato digital em conjunto com um software de livre distribuição para visualização dos mapas e acesso às respectivas tabelas de atributos – ArcExplorer da ESRI. Para tanto os mapas foram exportados para o formato “shapefile” e organizados em diretórios de acordo com os temas representados.

Essa forma de distribuição apresentou problemas concentrados em dois aspectos: em primeiro lugar o baixo número de pessoas atingidas em função do número limitado de CDs produzidos (atrelado a esse problema está o custo de produção do material gráfico e mídia envolvidos); em segundo lugar, a dificuldade de uma futura atualização do banco de dados espacial ligado aos mapas, uma vez que seria necessário produzir uma nova remessa de CDs.

A transformação dos mapas para o formato SVG e sua disponibilização na Internet elimina esses problemas, pois permite o acesso a qualquer usuário de Internet e facilita muito o processo de atualização dos dados, o que pode ser feito em uma única operação, diretamente no servidor. Além disso, é possível a criação/adaptação de ferramentas de consulta que permitam manipular e interagir com os mapas, acessando seu banco de dados espacial.

5.1.1.3 Aplicabilidade dos protótipos à outras áreas e temas

Os protótipos apresentados neste trabalho contribuem para que o inventário possa se tornar conhecido pelos técnicos que atuam na área, tanto quanto para os governantes ou agentes tomadores de decisão, permitindo fácil acesso às principais informações para qualquer pessoa, sendo o único requisito a possibilidade de acessar a Internet. Informações como a localização dos principais pontos geradores de poluição por dejetos de suínos, o total de dejetos gerados em cada microbacia, distribuição dos integrados e suas respectivas empresas integradoras, e muitas outras informações podem ser facilmente obtidas e visualizadas espacialmente, acessando o site com os protótipos. Isso permite o planejamento de ações técnicas e administrativas de forma rápida e transparente. Além disso, os próprios moradores da sub-bacia podem visualizar e navegar pelos mapas, o que provavelmente os estimularia a participar do processo de educação ambiental e tomar consciência de sua participação no processo.

Embora o trabalho aqui apresentado tenha tomado como tema o Inventário de Terras da Sub-bacia do Rio Ariranha, é importantíssimo ressaltar que ambos os protótipos são flexíveis e perfeitamente adaptáveis à outras áreas e temas, sem prejuízo de suas características operacionais.

Assim como o trabalho realizado pela Epagri nesse Inventário, existem outros trabalhos de cunho científico, tanto na área de Pesquisa Agropecuária como Extensão Rural cujos resultados podem ser publicados usando os mesmos protótipos aqui apresentados. Da mesma forma, qualquer outra área do Estado ou do País, poderia lançar mão destes protótipos, desde que existam a base cartográfica e os mapas temáticos da área que se deseja incluir no protótipo. Isto representa uma oportunidade de divulgação de resultados de pesquisas com grande capacidade de difusão e usando tecnologia recente e barata, já que trata-se de uma ferramenta aberta e em pleno desenvolvimento.

5.2 Geração do mapa em formato SVG

O primeiro passo para publicação dos mapas selecionados na Internet, envolve a transformação do mapa digital de seu formato original para o formato SVG. Atualmente existem inúmeras ferramentas disponíveis que executam essa tarefa, mas com diferentes graus

de operabilidade e flexibilidade quanto às características do mapa resultante, principalmente em termos de atributos visuais e interatividade. Alguns softwares de edição de imagens, como Adobe Photoshop e CorelDraw já apresentam, em suas versões mais recentes, ferramentas que exportam as imagens para o formato SVG.

A diferença básica entre gerar uma simples imagem a partir de um software de edição de imagem e gerar um “mapa”, está no sistema de coordenadas adotado. O SVG baseia-se num sistema cartesiano para apresentar uma imagem na tela, de forma que qualquer figura apresentada possui coordenadas definidas de cada um de seus vértices. Essas coordenadas serão relativas quando se trata de uma simples figura, e apenas indicam a posição do vértice em relação às dimensões físicas da tela (linhas e colunas). Já um mapa possui geo-referenciamento, e portanto existe uma matriz que correlaciona o sistema de coordenadas adotado (por exemplo UTM ou LatLong - que representa a superfície da terra) e a posição relativa da figura na tela, de acordo com o nível de zoom.

Isto significa que a ferramenta adotada para a transformação do mapa de seu formato digital original para SVG deve considerar o geo-referenciamento, mantendo assim o sistema de coordenadas.

O foco desse trabalho é o mapa em formato SVG e não a transformação envolvida. Mesmo assim, cabe salientar que para o caso específico do Inventário de Terras da Sub-bacia do Rio Ariranha adotou-se duas ferramentas distintas. A primeira trata de um “script” escrito em “*Avenue*”, linguagem interna de programação do software Arcview 3.0, que é instalada como um suplemento e transforma um mapa em formato “shapefile” para SVG usando os recursos do ArcView. A segunda trata de um conversor escrito em JAVA, que necessita apenas do software “JAVA Virtual Machine – JVM” (máquina Java virtual) instalado. Este conversor é capaz de “ler” um arquivo em formato “shapefile” e transformá-lo diretamente para SVG, de forma estática ou interativa.

Ambos os casos estão dentro da filosofia de software livre – que não significa apenas livre distribuição, mas também código aberto, e serão abordados a seguir.

5.2.1 Script em Avenue

Como os mapas originais do Inventário do Ariranha tiveram sua versão final gerada no software ArcView, optou-se pela utilização de um script escrito na linguagem de programação “Avenue”, linguagem de macro nativa específica para este aplicativo. O script é integrante do *GeoClient/GeoServer Project*. Na realidade esse projeto propõe uma solução completa, não só de transformação para SVG, mas também todas as ferramentas necessárias para publicar o mapa transformado na Internet.

Junto com a ferramenta os autores desenvolveram um arquivo de script, que compõe uma “extensão” do ArcView, conforme terminologia usada no software. A partir do ArcView, habilita-se a extensão respectiva, carrega-se e ativa-se os temas desejados e usa-se a ferramenta para exportação, que gera o arquivo SVG conservando as principais características de visualização, como cores dos polígonos, espessura das linhas e ordem de sobreposição dos temas. A Figura 5 ilustra o uso da ferramenta dentro do ambiente do ArcView:

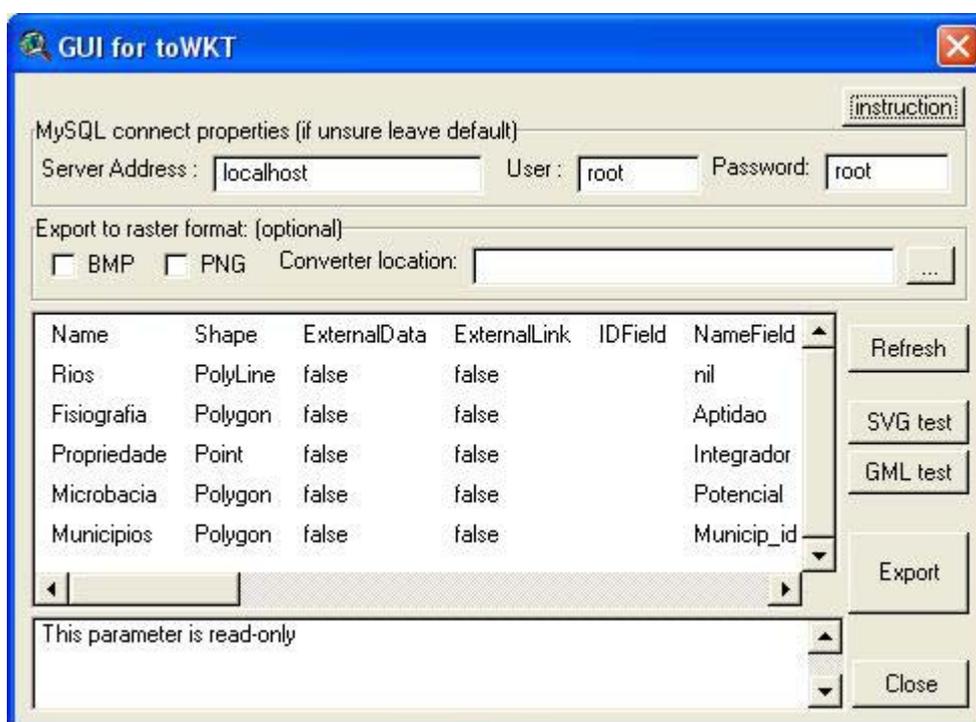


FIGURA 5 - Aspecto da ferramenta para ArcView.

Mais informações à respeito da ferramenta usada podem ser obtidas no site: <http://geoserver.sourceforge.net>.

A maior dificuldade encontrada no uso dessa ferramenta reside no fato de ser necessário fazer alterações manuais no código do arquivo SVG resultante. Isto é necessário porque a ferramenta foi idealizada para gerar um mapa formatado para o projeto GeoClient, conforme proposto pelos autores do script, e por isso o arquivo SVG gerado tem em seu código funções JavaScript e links que interligam-se à outros arquivos componentes do projeto e até conexão com banco de dados, para dar interatividade ao mapa. O código então precisa ser alterado para que o arquivo SVG seja adequado para o uso das ferramentas propostas nesse estudo.

5.2.2 Conversor JAVA

Esta segunda opção de transformação de mapas para o formato SVG foi escolhida por sua simplicidade de operação e por ser um aplicativo de código aberto, o que permite grande flexibilidade de adequação à situações específicas. Por outro lado, exige conhecimento de programação, o que pode ser um fator limitante para seu uso. No entanto, as modificações feitas no código fonte durante os estudos para realização desse trabalho, geraram um aplicativo personalizado, adequado para as situações semelhantes ao do caso tomado aqui como exemplo – Inventário de Terras do Ariranha. Mais informações sobre o código fonte original do conversor podem ser encontradas em “<http://www.mycgiserver.com/~amri/convertter.cocoon.xml>.”

A vantagem de usar um programa escrito em JAVA, reside no fato de existir a possibilidade de rodar em qualquer sistema operacional, já que esta linguagem possui arquitetura neutra, ou seja, é capaz de gerar um código binário para uma JVM (máquina Java virtual), que por sua vez pode ser instalada em diversos sistemas operacionais. Os arquivos de classe Java são compatíveis com qualquer plataforma de hardware que tenha ambiente *runtime* Java. Esse ambiente consiste na JVM, algumas bibliotecas de classe padrão, um verificador de *byte code* (para fins de segurança) e um interpretador de *byte code*. O

interpretador roda os arquivos de classe na JVM sem exigir que o programa tenha seu código fonte alterado para rodar em outro sistema operacional.

Esse aplicativo, doravante denominado de conversor, teve seu código fonte alterado para adequação prática aos propósitos desse estudo. Para tanto, utilizou-se o software NetBeans IDE 3.5.1, distribuído livremente pela Sun Microsystems Inc., que roda em plataforma JAVA 1.4.1_02.

As principais alterações feitas no conversor foram:

a) Tradução para língua portuguesa dos itens do menu, texto de botões, textos de menu “popup”, menus de atalho e barra de status.

b) Alteração no código de transformação de coordenadas, para que, no caso de coordenadas expressas em UTM, o número de casas após a vírgula fosse minimizado, reduzindo o tamanho do arquivo SVG gerado. A alteração desse código tem efeito positivo por dois aspectos: uma forma eficiente de reduzir o tamanho dos arquivos SVG é reduzir o número de casas após a vírgula em cada número que compõe as coordenadas dos vértices, desde que não prejudique o nível de zoom e precisão preconizados para o tipo de mapa em questão; além disso, como o arquivo gerado nada mais é que um arquivo de texto, a edição desses valores após a geração do arquivo é muito mais trabalhosa, portanto, alterar o código do programa que vai gerar o arquivo SVG resulta em menor gasto de tempo e esforço, do que alterar o arquivo gerado.

c) A alteração mais significativa foi aquela que acrescentou ao conversor, a habilidade de gerar pequenos arquivos SVG com os símbolos que compõe a legenda, sendo gerado um arquivo para cada tema (*layer*) incluído no mapa. Além disso, essa alteração faz com que o conversor gere também um arquivo para visualização imediata, escrito em html, que chama o mapa principal em SVG e monta uma legenda simplificada, acrescida de uma caixa de verificação para cada tema, que quando acionada, exhibe ou oculta o respectivo tema.

A seqüência de código à seguir, mostra resumidamente, a parte da função Java introduzida, que realiza essas operações:

```
“...  
public void genJSArray(){
```

```

int totfeat=0;

String[] txtleg;

txtleg = new String[100];

    setBar("", 1, totfeat, data.getTotalFeatures());

ps.print("<html>");

ps.println();

ps.print("<head>");

ps.println();

ps.print("<title>Mapa</title>");

ps.println();

ps.print("<script language=\"javascript\">");

ps.println();

ps.print("themeNames = new Array();");

```

... o código acima mostra como é criada a parte do arquivo SVG que armazena os atributos, na forma de uma matriz (*array*) em JavaScript ...

```

txtleg[layer]="<tr><td><input_type=\"checkbox\"onclick=\"onoff(\"+name+\",this.checked)\"
name=\"chk\"+name+\"\"checked></td><td><EMBEDwidth=\"84px\"height=\"\"+((nclasses*1
2)+5)+\"px\" SRC=\"leg_tema\"+layer+\".svg\" NAME=\"legendSVG\"+name+\"\"
type=\"image/svg+xml\"
pluginspage=\"http://www.adobe.com/svg/viewer/install/\"></td><td>\"+name+\"</td></tr>";

```

... a linha acima mostra parte de um looping, onde são geradas as linhas no arquivo htm que incorporam as legendas no mapa, uma para cada layer existente.

A Figura 6 mostra a interface do aplicativo já alterado:

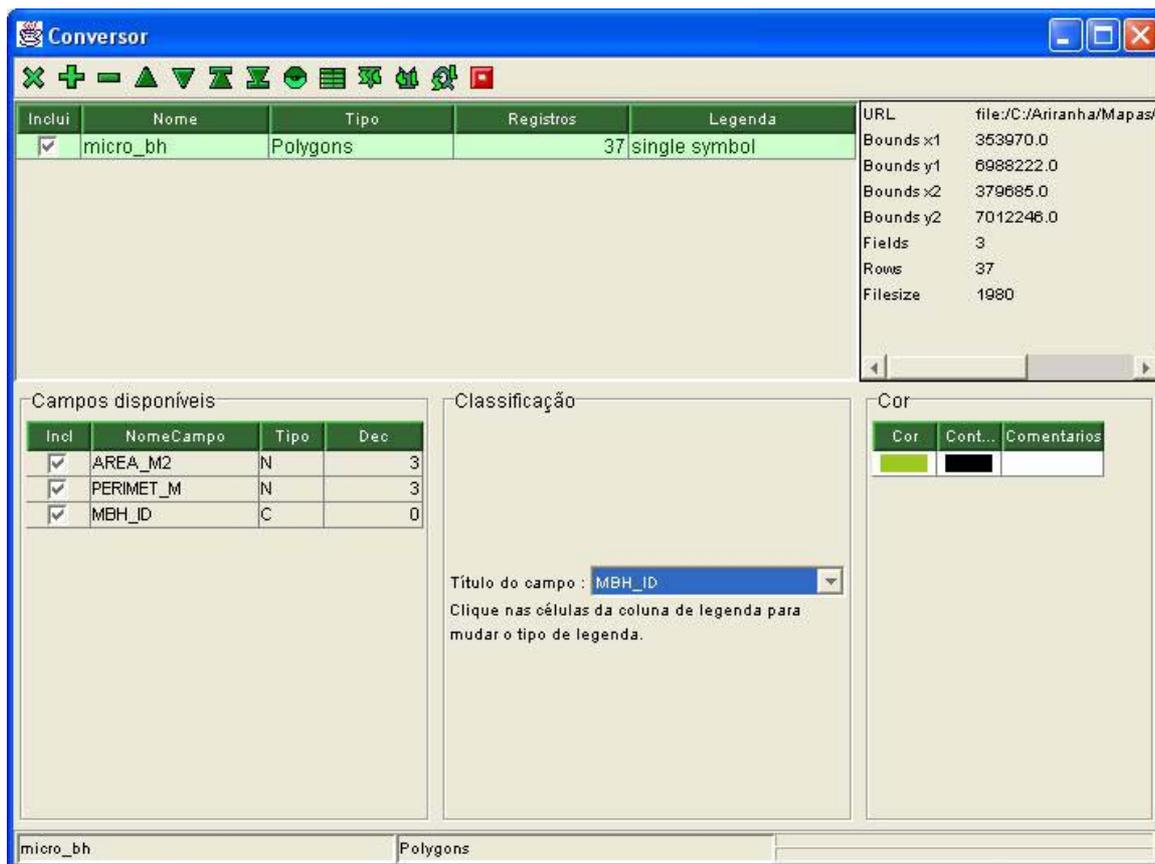


FIGURA 6 – Aspecto do conversor escrito em JAVA.

5.2.2.1 Operação do conversor

O botão com símbolo “+” chama a caixa de diálogo que permite apontar para o arquivo shape que contém o tema desejado. Após confirmação cada arquivo shape incluso aparece na forma de uma linha na tabela superior da tela. O botão “-” suprime o arquivo escolhido. Os quatro botões seguintes (formas de setas triangulares) propiciam troca de ordem entre os temas, para evitar sobreposição. O botão com símbolo do olho permite visualizar previamente o resultado do arquivo SVG à ser gerado. O símbolo de uma pequena tabela permite visualizar os dados da tabela de atributos do arquivo selecionado. Os 3 botões seguintes, abrem as respectivas caixas de diálogo para exportação: para SVG simplificado, onde as alterações desse trabalho cumprem seu efeito; para formato GML, opção já existente e mantida como alternativa; para SQL, opção em desenvolvimento, para conexão futura com

banco de dados. Finalmente o último botão, com símbolo do quadro, traz informações sobre o conversor.

Clicando no tipo de legenda, pode-se escolher entre “single symbol”, “unique value” e “gradual value”. Para cada caso, na janela “Cor” , é possível escolher as cores de preenchimento e borda, assim como escolher por qual campo, dentre os disponíveis na respectiva tabela de atributos, será feita a classificação, na janela “Classificação”. Por fim a janela “Campos disponíveis” mostra os campos da tabela de atributos que serão inclusos no mapa.

É conveniente salientar que para funcionamento desse conversor, é necessário instalar a máquina virtual Java (JVM) no computador.

5.3 Modelo 1 - Protótipo SVG simplificado

Uma vez gerado o mapa no formato SVG, através de qualquer das formas discutidas anteriormente, é necessário fazer adaptações e embutir funções com uso de JavaScript para dar mais interatividade ao produto.

As alterações implementadas no conversor implicam na geração de um arquivo principal em html, com a denominação “mapa_index.htm”. Este arquivo contém as matrizes de dados com os atributos dos temas contidos no mapa. Matrizes são variáveis compostas que podem conter vários dados, armazenados em “n” dimensões, conforme o número de dimensões da matriz. A linguagem JavaScript permite criar e manipular facilmente essas matrizes. O código à seguir mostra parte do arquivo mencionado, com trechos onde se atribui valores às matrizes.

```
<script language="javascript">
themeNames = new Array();
fieldNames = new Array();
values = new Array();
themeNameFields = new Array();
themeNames[0] = "ident";
fieldNames[0] = new Array();
...
```

```

fieldNames[0][6] = "MICROBACIA";
values[0][6] = new Array();
values [0][0][0] = "1";
values [0][0][1] = "10";
values [0][0][2] = "100";
values [0][0][3] = "101";
values [0][0][4] = "102";
values [0][0][5] = "103";
values [0][0][6] = "104";
...
values [0][4][0] = "RIO CLARO";
values [0][4][1] = "SEDE";
values [0][4][2] = "LINHA MOSCONI";
values [0][4][3] = "RIO CLARO";
values [0][4][4] = "QUATRO IRMAOS";
values [0][4][5] = "SEDE";
values [0][4][6] = "SAO MIGUEL";
values [0][4][7] = "DIVISA DAS AGUAS";
values [0][4][8] = "DIVISA DAS AGUAS";
...
</script>

```

Nesse caso, os atributos de cada tema incluso no mapa estão armazenados no próprio arquivo de índice, e não em um banco de dados separado. Esta particularidade implica em mais agilidade na realização de consultas por parte do cliente, pois uma vez carregado o mapa, os dados estão armazenados localmente. Por outro lado, é fácil deduzir que quando a quantidade de dados e/ou elementos no mapa for muito grande, o tempo de transferência dos arquivos para a máquina do cliente poderá ser fator limitante para o uso dessa metodologia.

Este mesmo arquivo também contém uma parte de código html puro que incorpora o mapa principal e a respectiva legenda. Esta código pode ser alterado para criar um ambiente organizado, com tabelas, cores, textos e links adequados. A linha à seguir mostra o link que incorpora o mapa à página htm:

```

...
<EMBED width=\""+largura+"\" height=\""+altura+"\" SRC=\"mapa.svg\"
NAME=\"theSVG\" type=\"image/svg+xml\"
pluginspage=\"http://www.adobe.com/svg/viewer/install/\">";
...

```

O mesmo arquivo de index contém ainda as funções em JavaScript que interagem com a linguagem SVG. A maioria das funções foram alteradas e algumas novas foram criadas para proporcionar maior interatividade. As funções existentes no arquivo e suas respectivas alterações feitas especificamente para este trabalho são apresentadas na Tabela 2:

TABELA 2 - Funções JavaScript incorporadas ao arquivo “mapa_index.htm”

NOME FUNÇÃO	O QUE FAZ	ALTERAÇÃO
Function id	Identifica o elemento sob o mouse, quando clicado	Não
Function write id	Cria a janela com os atributos respectivos do elemento clicado	Alterada para destacar cor do campo de identificação dos demais campos e redimensionar a janela automaticamente em função do número de campos exibidos
Function onoff	Exibe e oculta o tema respectivo	Não
Function over	Troca a cor de preenchimento do elemento sob o mouse	Troca a cor e aplica transparência ao elemento destacado
Function out	Retorna a cor original quando o mouse deixa o elemento	Devolve a cor e retira a transparência
Function overLeg	Quando o mouse passa sobre um item da legenda, destaca no mapa todos os elementos correspondentes	Nova função
Function outLeg	Quando o mouse deixa um item da legenda, volta à cor e transparência normal	Nova função
Function consult	Encontra elementos no mapa, baseado nos critérios oriundos de um formulário na página htm.	Nova função
Function maisa	Acrescenta brilho à imagem de fundo (raster)	Nova função
Function menosb	Diminui brilho da imagem de fundo (raster)	Nova função

As funções “over” e “out” introduzidas no protótipo facilitam a percepção de qual elemento está sob o dispositivo apontador (*mouse*), uma vez que confere transparência ao invés de simplesmente trocar a cor. Além disso, essas funções podem ser facilmente alteradas para criar outros efeitos, como alteração da espessura das linhas de contorno do elemento destacado ou um misto de efeitos.

Dentre as funções novas introduzidas, merece destaque a função “consult”, pois é uma ferramenta particularmente útil para mapas interativos já que identifica determinados elementos do mapa que atendem a um determinado critério, e os destaca, mudando sua cor.

```

...
function consult(){
var svgdoc = document.theSVG.getSVGDocument();
var valor = document.quest.label2.value;
var operador= document.quest.operador.value;
var tema = document.selecao.selet.value;
var ate = 0;
var i = 0;
var cont = 0;
...

```

Identifica o documento principal (SVG) e atribui os valores digitados nos campos do formulário (critérios) às variáveis.

```

...
if (tema == 0) ate = 892;
if (tema == 1) ate = 182;
if (tema == 3) ate = 139;
var themename=themeNames[tema];
...

```

Limita o número de elementos em cada tema.

```

...
for (i;i<ate;i++){
obj=themename+i;
svgobj = svgdoc.getElementById(obj)
svgstyle = svgobj.getStyle()
valora = values[tema][0][i];
...

```

Inicia o looping e determina o estilo do elemento procurado. Determina o valor do atributo do elemento atual.

```

...
switch (operador){
case "==" :
if (valora == valor){
svgstyle.setProperty('fill','rgb(255,255,0)');
cont = cont + 1;
}
}

```

```

        break;
    case ">":
        if (valora > valor){
            svgstyle.setProperty('fill','rgb(255,255,0)');
            cont = cont + 1;
        }
        break;
    case "<":
        if (valora < valor){
            svgstyle.setProperty('fill','rgb(255,255,0)');
            cont = cont + 1;
        }
        break;
    default:
        break;
}
...

```

Determina o tipo de operador e muda a cor dos elementos encontrados para amarelo.

```

...
}
document.quest.label2.value=cont+" encontrado(s) em AMARELO";
if (cont==0){
document.quest.label2.value="Não encontrado!";
}
}
...

```

Finaliza a função, exibindo a mensagem de acordo com o resultado.

O arquivo “mapa.svg” contém os elementos que constituem o mapa propriamente dito. A estrutura de um arquivo SVG é subdividida em instâncias que representam os elementos gráficos, sempre identificados por marcadores do tipo `<g>` descrição do elemento `</g>`. A primeira declaração, de fundamental importância, é a janela de exibição, chamada `ViewBox`. Ela limita a área da tela que conterá os demais elementos gráficos. Nesse caso, as coordenadas iniciais estão no sistema UTM, pois logo após existe uma matriz de conversão que adapta as coordenadas para o número de linhas e colunas disponíveis na tela, através de fatores de multiplicação. O código a seguir demonstra tal situação:

```

<?xml version="1.0"?>
<svg viewBox="353970 6988222 25715 24024">
...

```

Os dois primeiros números representam as coordenadas do canto superior esquerdo da ViewBox, o terceiro número representa a largura da janela, na mesma unidade em que as coordenadas são expressas, e da mesma forma, o último número representa a altura.

```

...
<g id="relevo">
  <image x="353970" y="6988222" width="25715" height="24024"
xlink:href="relevo2.jpg"> <title>relevo</title> </image>
</g>
...

```

A linha acima mostra o primeiro elemento do mapa, que é a imagem de fundo, sendo posicionada em função das coordenadas iniciais e de sua largura e altura. Cabe explicar que, de acordo com a estrutura de um arquivo SVG, o elemento que é declarado no início do arquivo é montado primeiramente e portanto formará a primeira “camada” do mapa. Os demais elementos serão sobrepostos em ordem, logo, o último elemento declarado sobreporá todos os demais. Isto é importante para que se tome o cuidado de dispor as layers de linhas e pontos sobre as layers de polígonos e imagens, para que sua visibilidade não fique obstruída. No entanto, o SVG possui um recurso interessante que permite controlar a opacidade de cada layer, permitindo vários níveis de transparência.

```

...
<g transform ="matrix(1 0 0 -1 0 1.4000468E7)">
...

```

A linha acima mostra a matriz de transformação, que adapta o sistema de coordenadas do mapa à resolução da tela. Caso a ViewBox fosse declarada com valores que representam o número de linhas e colunas da tela, então todas as coordenadas dos elementos inclusos no mapa deveriam seguir o mesmo critério. Como tais coordenadas são oriundas do arquivo shapefile, elas normalmente são expressas em UTM ou coordenadas geográficas. Daí a necessidade de uso da referida matriz de transformação.

Na seqüência do arquivo SVG seguem as demais layers, conforme mostra o código:

```

...
<g id="uso" stroke-width="1.9440124416796266E-5" opacity="0.5">
...

```

A linha acima mostra a declaração do elemento que representa o mapa de uso do solo, especificando espessura da borda e opacidade de 50%. Em seguida, estão declarados cada um dos polígonos que compõe o mapa:

```

...
<path id="uso0" fill="rgb(255,204,51)" stroke="rgb(0,0,0)"
onclick="id(3,0)" onmouseover="over('3','0','fill')"
onmouseout="out('3','0','fill')" d="M373547.49999999994,7008488.999999998
L373557.49999999994,7008468.999999999 373547.49999999994,7008418.999999999
373567.49999999994,7008293.999999999 ... 7011893.999999999 z "/>

<path id="uso1" fill="rgb(246,251,133)" stroke="rgb(0,0,0)"
onclick="id(3,1)" onmouseover="over('3','1','fill')"
onmouseout="out('3','1','fill')" d="M376902.49999999994,7011863.999999998
... 376874.99999999994,7011859.0 376902.49999999994,7011863.999999998 z "/>

<path id="uso2" fill="rgb(246,251,133)" stroke="rgb(0,0,0)"
onclick="id(3,2)" onmouseover="over('3','2','fill')"
onmouseout="out('3','2','fill')" d="M373977.49999999994,7011593.999999999
L374027.49999999994,7011573.999999999 ... z "/>
...

```

Cada linha de código representa um polígono, onde são declarados o identificador, cor de preenchimento, cor da borda e a chamada para as funções “id”, “over” e “out” já descritas na tabela x, que são chamadas em função dos eventos onclick, onmouseover e onmouseout, respectivamente. De forma análoga, são declaradas as demais layers, conforme exemplificado à seguir:

```

...
<g id="ident" stroke-width="7.777561734396268E-6">
<ellipse id="ident0" fill="rgb(204,0,51)" stroke="rgb(0,0,0)"
onclick="id(0,0)" onmouseover="over('0','0','fill')"

```

```

onmouseout="out('0','0','fill')" cx="370131.0" cy="7005302.0" rx="100"
ry="100"/>

<ellipse id="ident1" fill="rgb(204,0,51)" stroke="rgb(0,0,0)"
onclick="id(0,1)" onmouseover="over('0','1','fill')"
onmouseout="out('0','1','fill')" cx="369271.0" cy="7004336.0" rx="100"
ry="100"/>/>
<ellipse id="ident2" ...

...

<ellipse id="ident892" fill="rgb(204,0,51)" stroke="rgb(0,0,0)"
onclick="id(0,892)" onmouseover="over('0','892','fill')"
onmouseout="out('0','892','fill')" cx="365204.0" cy="7006558.0" rx="100"
ry="100"/>/>
</g>
</g>

</svg>

```

Esse modelo de mapa contém uma quantidade razoável de informação e pode ser suficiente para muitas aplicações. Os arquivos assim gerados, incluindo o mapa SVG, o arquivo `mapa_index.htm` e os arquivos de legenda, somam pouco mais de 700 Kb, representando tamanho de arquivo razoável para transmissão pela Internet.

Para apresentação de mapas na *Web* usando esse modelo, cada mapa gerado precisará passar por edições no código, para ajustar as ferramentas ao seu caso específico. Esse esforço é justificável já que o número de pessoas que poderá ter acesso às informações é ilimitado e também devido ao baixo custo envolvido no processo, pois os softwares sugeridos são livres.

A Figura a seguir mostra o aspecto final da página, após todas as implementações:

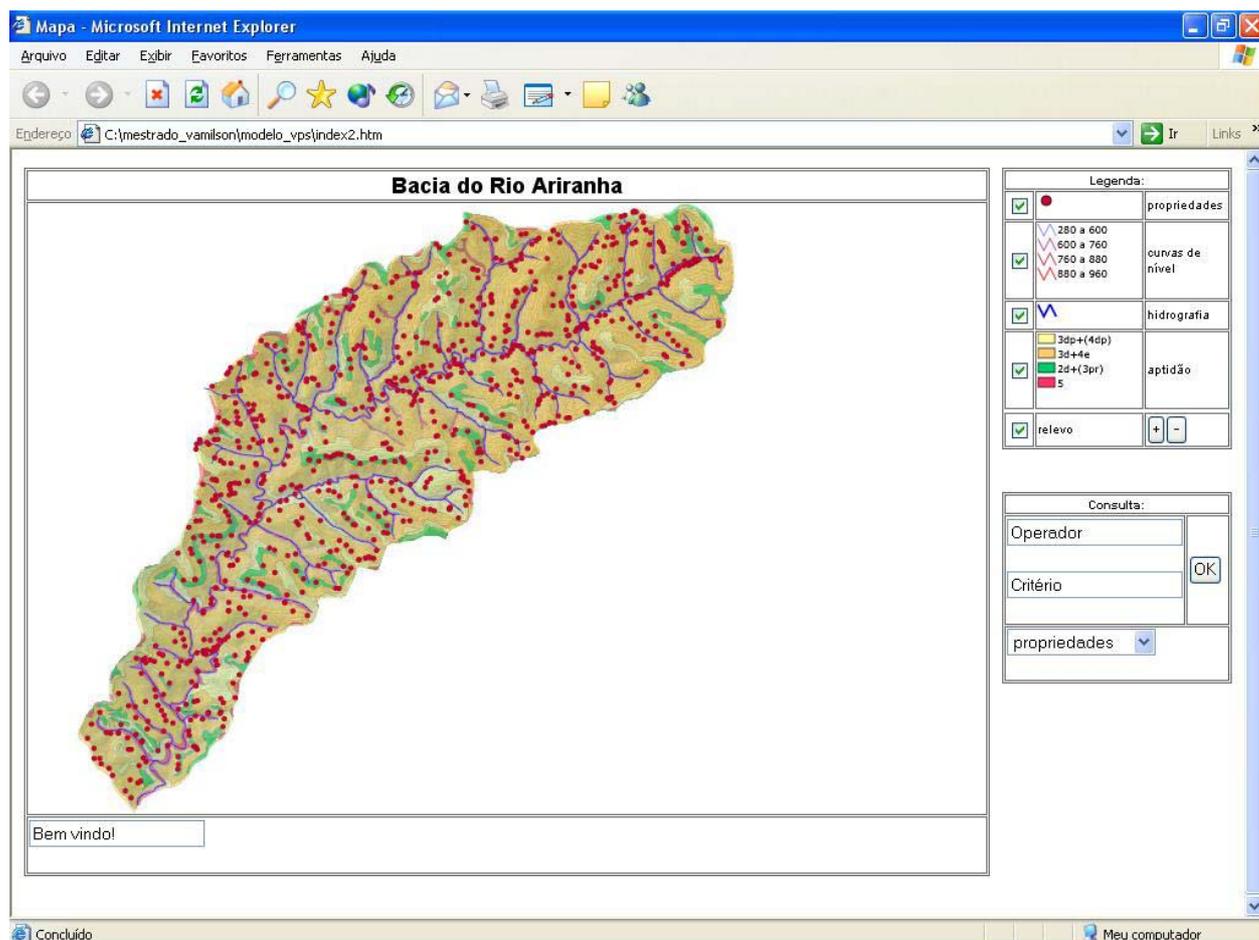


FIGURA 7 - Aspecto da página usando modelo 1.

O modelo 1 pode ser acessado na Internet, para fins de demonstração, através do endereço:

www.paesimoveis.floripa.com.br/mapas/ariranha

5.4 Modelo 2 – Protótipo GeoClient modificado

A ferramenta original que inspirou parte do desenvolvimento deste trabalho foi criada dentro da filosofia de software livre, sendo que o código fonte foi construído com a colaboração de diversos programadores, em especial o indonésio Amri Rosyada e o canadense Nedjo Rogers. O projeto chama-se GeoClient e foi desenvolvido como uma alternativa e

colaboração do projeto GeoServer, que é um projeto maior, em franco desenvolvimento. O projeto GeoServer está sendo construído de acordo com as especificações do OpenGIS Consortium para WFS (*Web Feature Server*) e é também um software livre, disponível sob licença GPL.

O GeoClient propõe uma solução para publicar um mapa no formato SVG na Internet, sem ser, no entanto, um servidor de mapas. Ele apenas exibe no navegador um mapa pré-formatado, sem ocorrer geração dinâmica de mapas em função de requisição do usuário. Escrito em SVG e *ECMAScript/JavaScript*, o aplicativo foi desenhado para rodar em um navegador de Internet, desde que o mesmo suporte SVG. No entanto, é possível utilizá-lo em um navegador sem suporte à SVG usando um *plugin* adequado para leitura desse formato. De acordo com MARISCO (2004), a interatividade com o usuário é toda projetada em script *ECMAScript/JavaScript*, em que são utilizadas funcionalidades do tipo:

- Navegação (pan, zoom, etc);
- Legendas Dinâmicas: símbolos, gradação de cores, gráficos, etc;
- Objetos ativos: símbolos e botões clicáveis;
- Funções para consultas: janelas pop up, caixas de seleções, caixa de textos, etc;
- Modificação dos atributos das feições do mapa: cor, transparência, tamanho dos objetos, etc.

O GeoClient pode ser considerado como um aplicativo composto por diversos arquivos. Um deles contém as definições de estilo que determinam as cores, fontes, preenchimentos e bordas da maioria dos elementos apresentados no mapa. É um arquivo no formato “css” conhecido como folha de estilos. Outro arquivo contém as funções escritas em *ECMAScript/JavaScript* que são funções que interagem com o plugin SVG instalado na máquina do cliente, passando diversos tipos de parâmetros e permitindo a interatividade com o mapa. Um terceiro arquivo contém as definições da legenda do mapa, incluindo o texto e símbolos exibidos na legenda. Por fim, o aplicativo inclui outro arquivo com algumas funções em *JavaScript* desenvolvidas por um outro projeto de software livre, que simplificam a criação de janelas, tabelas e gráficos no formato SVG.

A tabela a seguir mostra os arquivos componentes, suas funções e respectivas alterações feitas especificamente para este trabalho.

TABELA 3 - Arquivos do GeoClient modificado

NOME ARQUIVO	O QUE FAZ	ALTERAÇÃO
estilo.css	Folha de estilos, contendo definição de cores de preenchimento e borda, fonte, tamanho de fonte, espessura de linhas.	Padronização cores de janelas e tipo de fonte.
configura.js	Determina tamanho das janelas, texto nas caixas de diálogo, tamanho e cores da paleta de cores, legenda instantânea e e todas as janelas abertas pelos botões da barra de ferramentas.	Apenas tradução de textos.
ferramentas.js	Contém funções JS que servem para facilitar construção controles que imitam o ambiente windows (botões, rótulos, checkbox, caixas de texto, tabelas, caixa de listagem, caixa de combinação, barra de rolagens, gráfico de colunas, janelas). Este arquivo foi desenvolvido como um projeto a parte, e pode ser usado para qualquer tipo de aplicação SVG.	Sem alterações.
Funções.js	Código JS para atribuir as funções dos botões da barra de ferramentas e das caixas de diálogo – zoom, pan, exibição da legenda, aplicação de rótulos e outros	Adaptação das funções para exibição das informações específicas dos temas usados no estudo de caso.
Legenda.js	Determina lista de layers que compõe a legenda, símbolos, classes, e valores aplicados em cada caso, conforme o tipo de legenda escolhida pelo usuário.	Arredondamento de campos numéricos para melhor exibição no gráfico, adequação do tamanho das barras e diâmetro das pizzas nos gráficos.
mapa.svg	Arquivo SVG, com descrição dos elementos do mapa, chamada para os demais arquivos, carrega funções para os principais eventos, monta a barra de ferramentas, e contém todos os atributos dos temas inclusos no mapa, substituindo a tabela de atributos.	Padronização de tamanho dos botões da barra de ferramentas, cores do gradiente para elementos destacados e desenho dos ícones dos botões.

Todos esses arquivos devem ficar em um único diretório do servidor, junto com o mapa principal em SVG. O arquivo que contém o mapa, deve fazer referência explícita aos demais arquivos mencionados. Assim, quando o usuário acessar o mapa no formato SVG, todos os arquivos que compõem o aplicativo serão descarregados na máquina do cliente em diretório temporário, e a partir daí todas as requisições serão executadas, sem necessidade de atualizações no servidor.

5.4.1 Operação do protótipo GeoClient modificado

Assim como no primeiro caso, para que o protótipo possa ser visualizado em um navegador de Internet, é necessário que o usuário instale anteriormente um plugin apropriado para seu navegador. Existem atualmente alguns plugins de interpretação de SVG disponíveis, mas o mais difundido é o *plugin Adobe SVG Viewer*. Novas funções são incluídas a cada versão do produto, e algumas sofrem modificações, de tal maneira arquivos SVG escritos para serem interpretados por uma determinada versão do plugin, podem ter um comportamento ligeiramente diferente em outras versões, e determinados recursos podem deixar de funcionar. Assim, para o caso aqui estudado, recomenda-se o uso do plugin Adobe SVG Viewer 3.0.

Ao acessar a página, ocorre um pequeno *delay* até que todos os arquivos sejam transferidos para a máquina do usuário. O tempo de espera depende da velocidade de transmissão dos dados da conexão com a Internet e também da capacidade de processamento da máquina, para que o browser interprete os arquivos descarregados. Nesse momento, os dados vetoriais serão transmitidos para o cliente, sem que nenhum arquivo de imagem matricial seja transmitido. Até mesmo as figuras usadas na barra de ferramentas do aplicativo são escritas em SVG.

A Figura a seguir mostra o aspecto geral do protótipo exibindo mapas do Inventário da Sub-bacia do Rio Ariranha.

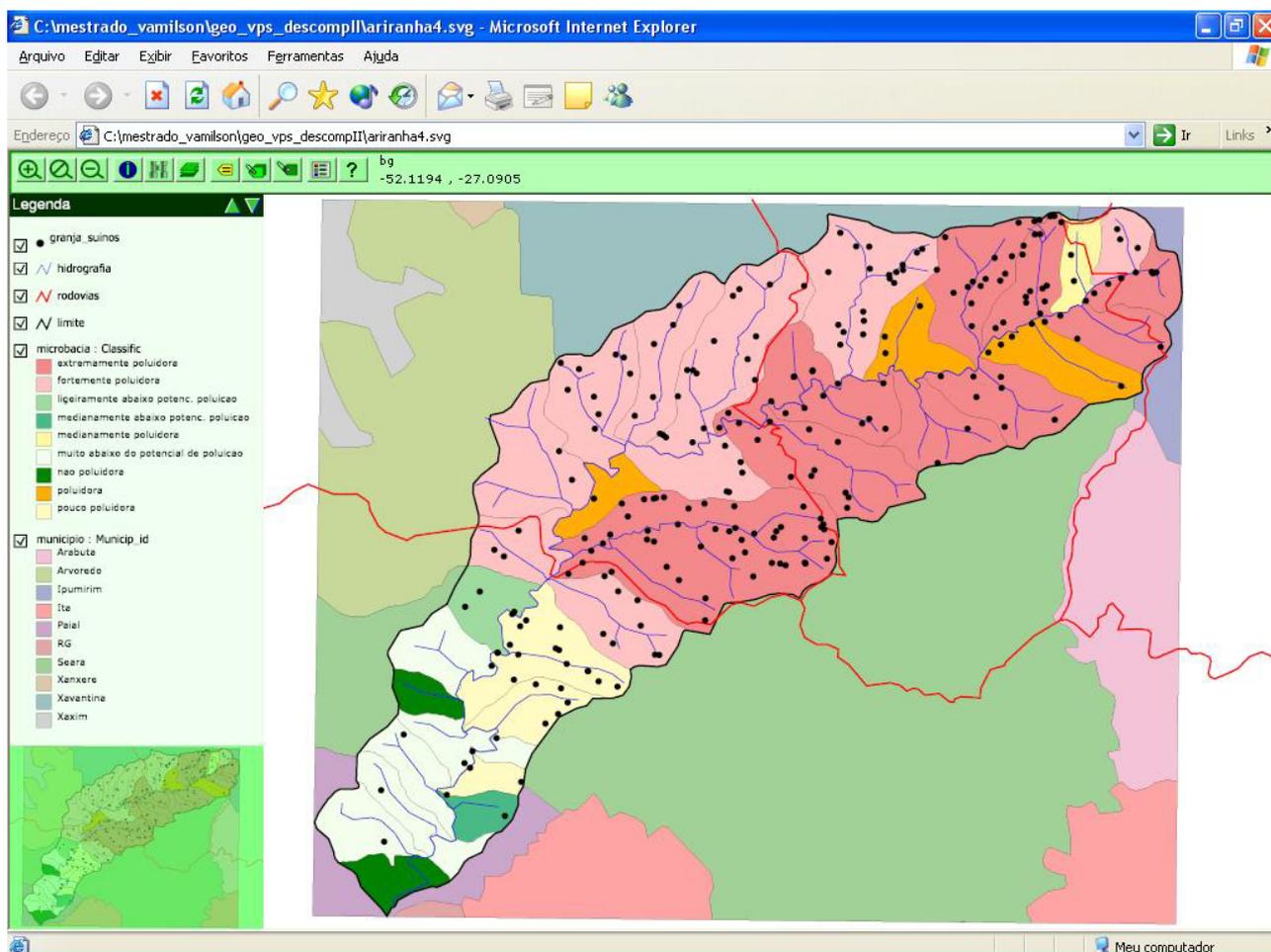


FIGURA 8 - Aspecto da página usando protótipo 2.

Na parte superior da área de navegação aparece uma barra de ferramentas, com as seguintes funcionalidades (da esquerda para direita):

a) Botões de zoom. Ícone: 

Os três primeiros botões da barra de ferramentas apresentam as opções zoom mais, zoom original e zoom menos. Essas funções são alternativas, pois o plugin SVG já contém funções de navegação, disponíveis através do menu de atalho (clique lado direito do mouse) e também através das teclas de atalho: Ctrl+botão direito do mouse para zoom mais; Ctrl + Shift para zoom menos e Alt + botão direito do mouse para pan (simula efeito de arrastar o mapa).

b) Botão informação. Ícone: 

Após clicar nesse botão e em seguida clicar em um elemento do mapa, abre-se uma janela com as classes e atributos relacionados ao elemento ativo no mapa. Nesse protótipo as informações estão no corpo do arquivo SVG, mas poderiam ser acessadas em tempo real através de uma consulta a um banco de dados, o que implicaria em alterações nos arquivos para realizar a conexão com o banco de dados.

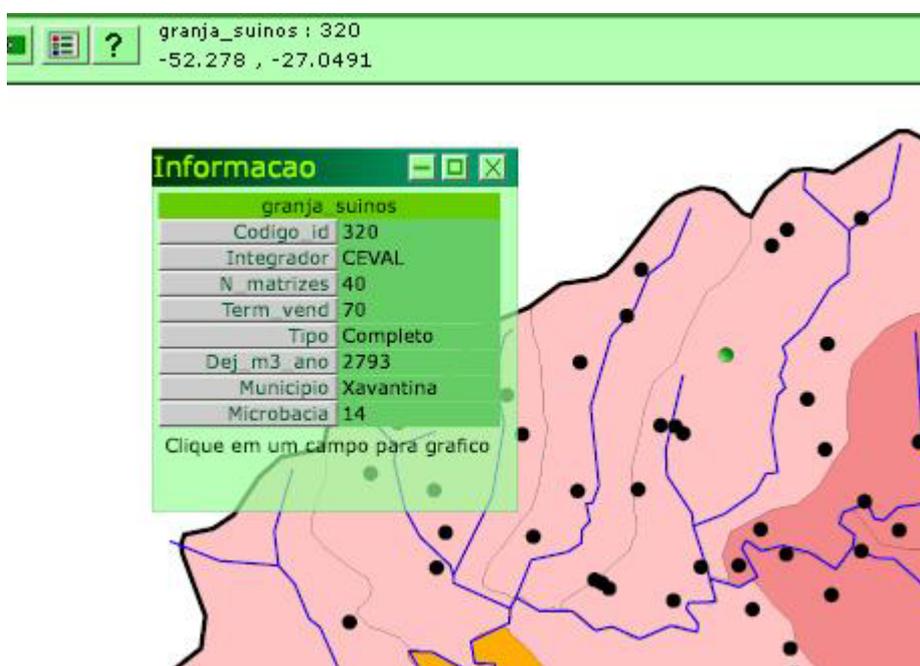


FIGURA 9 - Uso da ferramenta de identificação.

No exemplo, os pontos representam as sedes das granjas de suínos na microbacia. O ponto destacado na cor verde mostra a granja na qual o usuário clicou após selecionar o botão de informação. Surge a janela mostrando os atributos daquela granja.

c) Botão de consulta. Ícone: 

Permite aplicar consultas filtrando determinados atributos dos objetos existentes no mapa. Uma vez executada a consulta, é possível criar um gráfico de barras para qualquer

atributo selecionado, desde que seja um atributo numérico. A Figura 10 ilustra o uso da ferramenta.

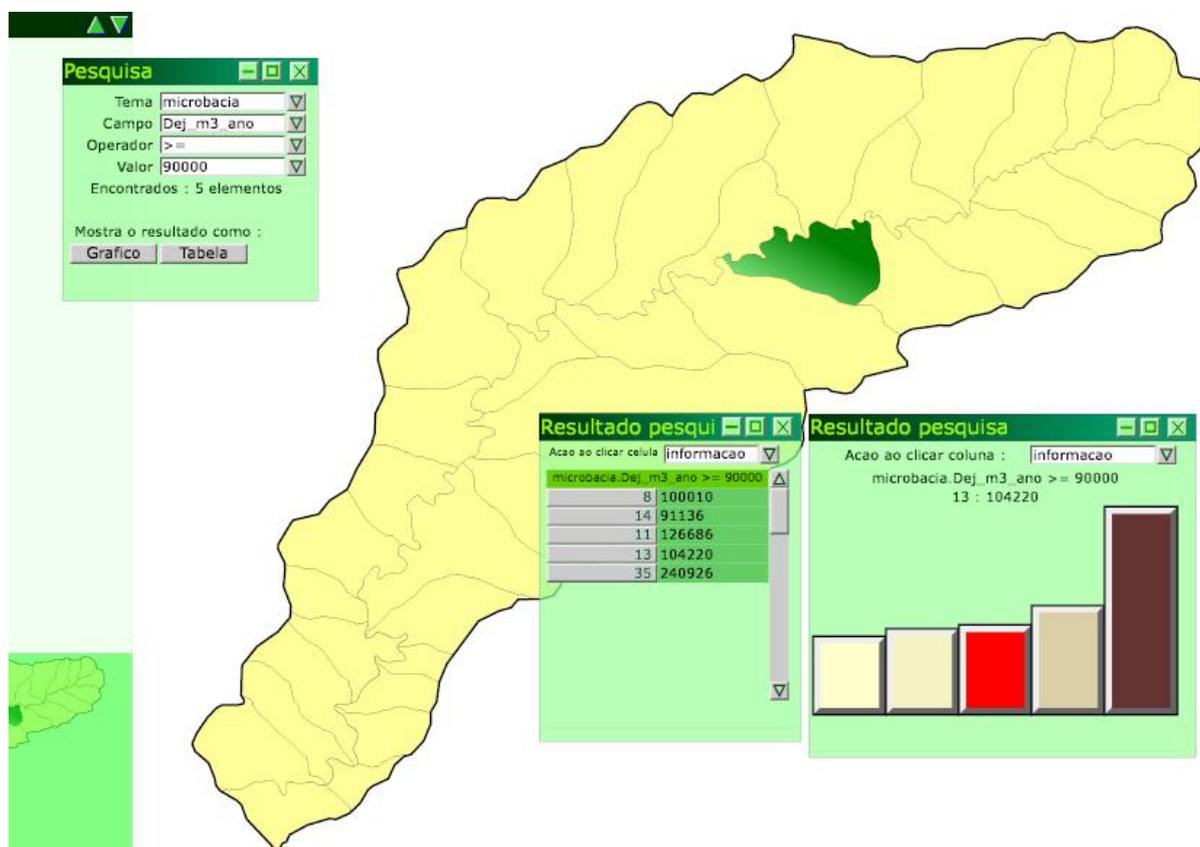


FIGURA 10 - Uso da ferramenta de consulta.

No exemplo da figura acima, ao clicar sobre o botão de consulta, surge a janela como a do lado superior esquerdo, onde o usuário informa o tema em que pretenda realizar a consulta (microbasias), o campo (dejetos m³/ano), operador (>=) e valor (90000). Assim a consulta retornará quais as microbasias que produzem um volume de dejetos maior ou igual à 90000 m³/ano. O resultado pode ser visualizado na forma de tabela (janela inferior esquerda) ou gráfico (direita). Ao passar o mouse sobre a célula da tabela ou sobre a coluna do gráfico, a respectiva microbacia será destacada, como no figura.

d) Botão de configuração da legenda. Ícone: 

Permite classificar cada tema por:

- Símbolo único: uma só cor para todos os elementos do tema;
- Por classes: uma cor para cada classe do tema;
- Cores graduadas: varia os tons de cada classe, a partir de uma cor inicial até uma determinada cor final, e separa as classes por quantis ou por intervalos iguais (só aplicável para campos numéricos).
- Símbolos gráficos: gera colunas ou pizzas (só para campos numéricos).

A figura 11 ilustra o uso da ferramenta de configuração de legenda:

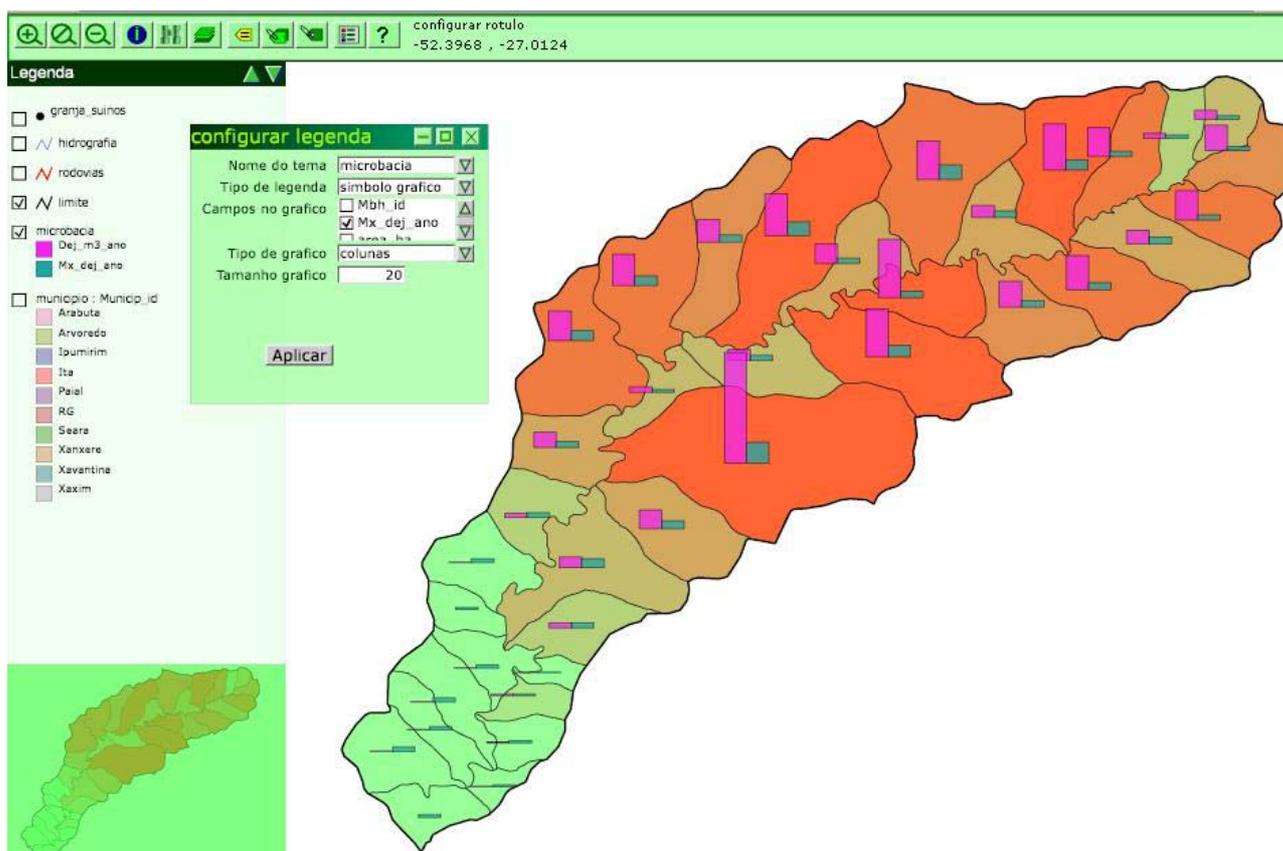


FIGURA 11 - Configuração de legenda.

No exemplo da figura, primeiramente usou-se a classificação por cores graduadas, atribuindo a cor vermelha para as microbacias com maior produção de dejetos e a cor verde para as que não produzem dejetos. Depois, usou-se a classificação por símbolo gráfico, escolhendo colunas verdes para representar o valor máximo de dejetos que a microbacias tem capacidade de absorver, e lilás para o volume de dejetos produzidos.

e) Botão configuração rótulo. Ícone: 

Permite especificar quais temas terão rótulo e qual o tamanho da fonte. Também permite escolher qual o campo fornecerá o texto para exibição. Fonte de tamanho zero não são exibidas.



FIGURA 12 - Configuração do rótulo.

f) Botão de aplicação de rótulo. Ícone: 

Aplica ou desaplica rótulo para todos os temas ativos de acordo com a configuração atual. Na figura a seguir, todos os temas foram omitidos, exceto o tema microbacias, para o qual foi aplicado o rótulo sobre o campo “código de identificação”.

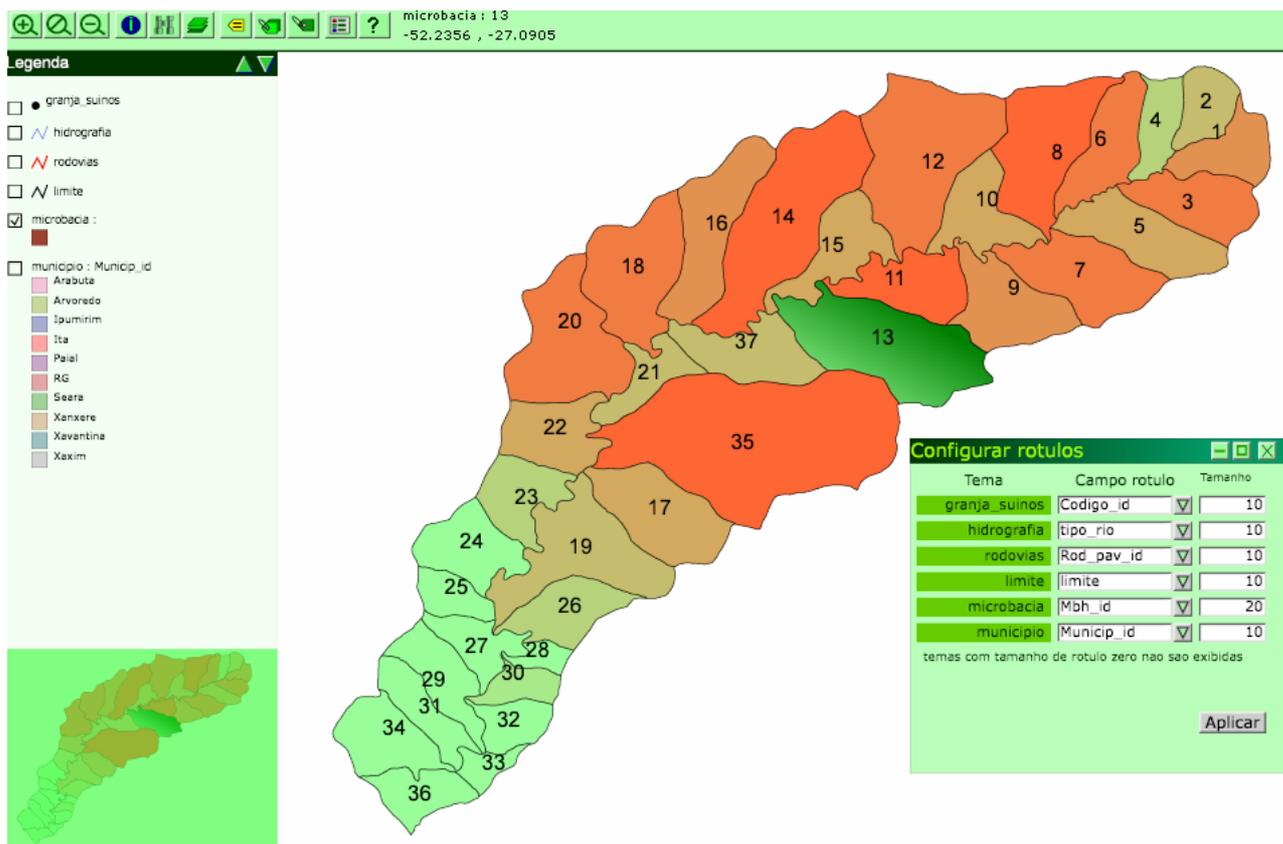


FIGURA 13 - Aplicação de rótulos.

g) Botão aplicação de rótulo manual. Ícone: 

Aplica ou desaplica rótulo somente nos elementos selecionados com o mouse. Isto permite um trabalho mais detalhado, para não sobrepor rótulos quando o polígono é muito pequeno.

h) Botão de preferências. Ícone: 

Permite configurar preferências de visualização gerais. É possível estabelecer a precisão de exibição das coordenadas (casas após a vírgula), desativar o destaque automático de elementos sob o mouse, exibir/ocultar legenda e exibir/ocultar a “visão global”. A visão global é uma área reservada no canto inferior direito da janela que mostra uma miniatura do

mapa. Assim, quando o usuário usa a ferramenta zoom mais, a visão global mostra qual parte do mapa está sendo exibida no momento.

i) Botão sobre. Ícone: 

Exibe informações de autoria sobre o aplicativo em SVG.

A Figura 14 ilustra a flexibilidade do protótipo, gerando um gráfico a partir dos atributos selecionados. No exemplo, o usuário clicou sobre uma microbacia (polígono) com a ferramenta de identificação ativada. Como resposta, surgiu a janela da esquerda mostrando os atributos do elemento espacial selecionado. Ao clicar no campo “Max_dej_ano” que representa a quantidade máxima que cada microbacia pode receber de dejetos de suínos por ano (em m³), surgiu a janela da direita, que mostra um gráfico de barras resumindo o mesmo campo para todas as microbacias do mapa. Ao deslocar o mouse sobre as barras do gráfico, a microbacia correspondente (ou outro elemento do mapa) é destacado.

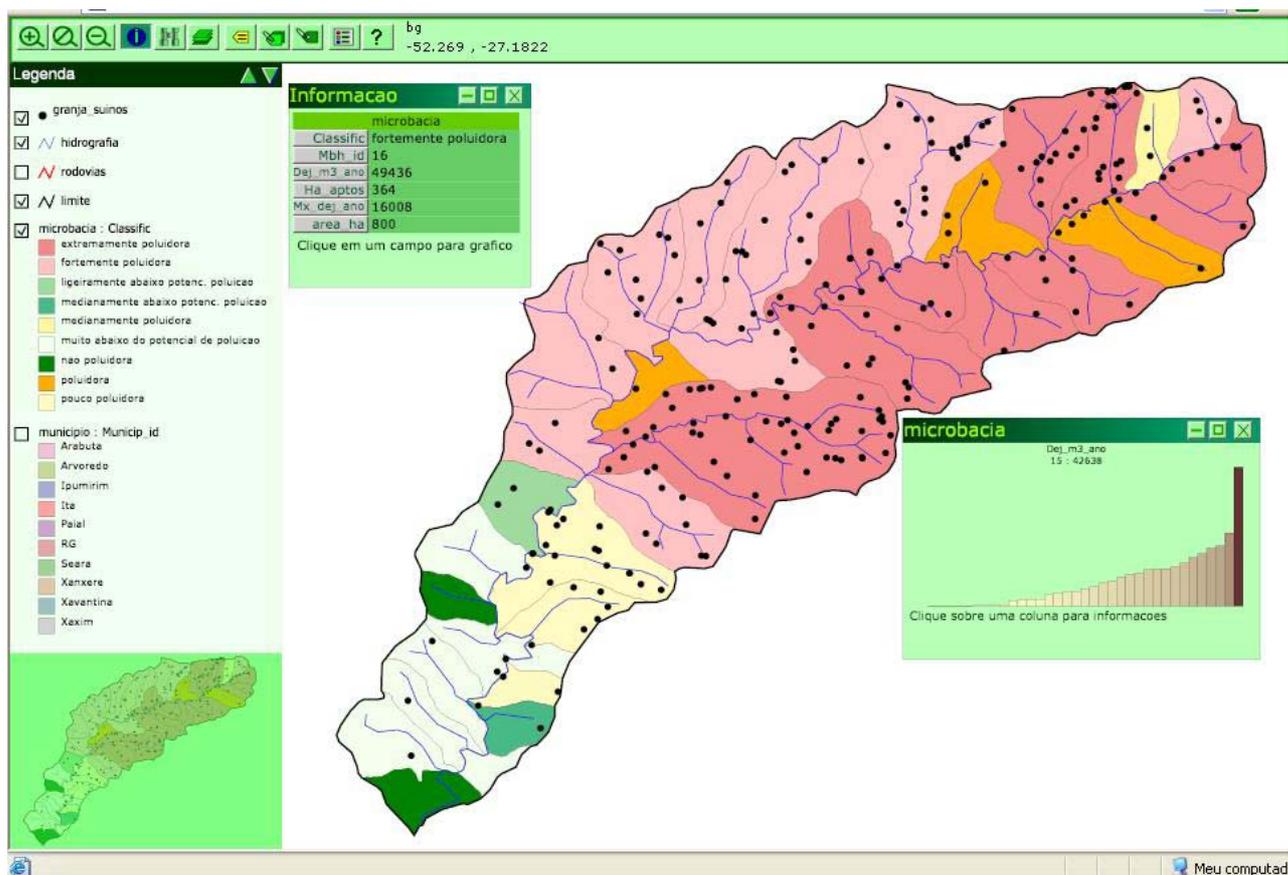


FIGURA 14 - Visualizando atributos e gráficos.

A figura 15 sintetiza o esquema de transformação do mapa original para SVG para cada um dos protótipos.

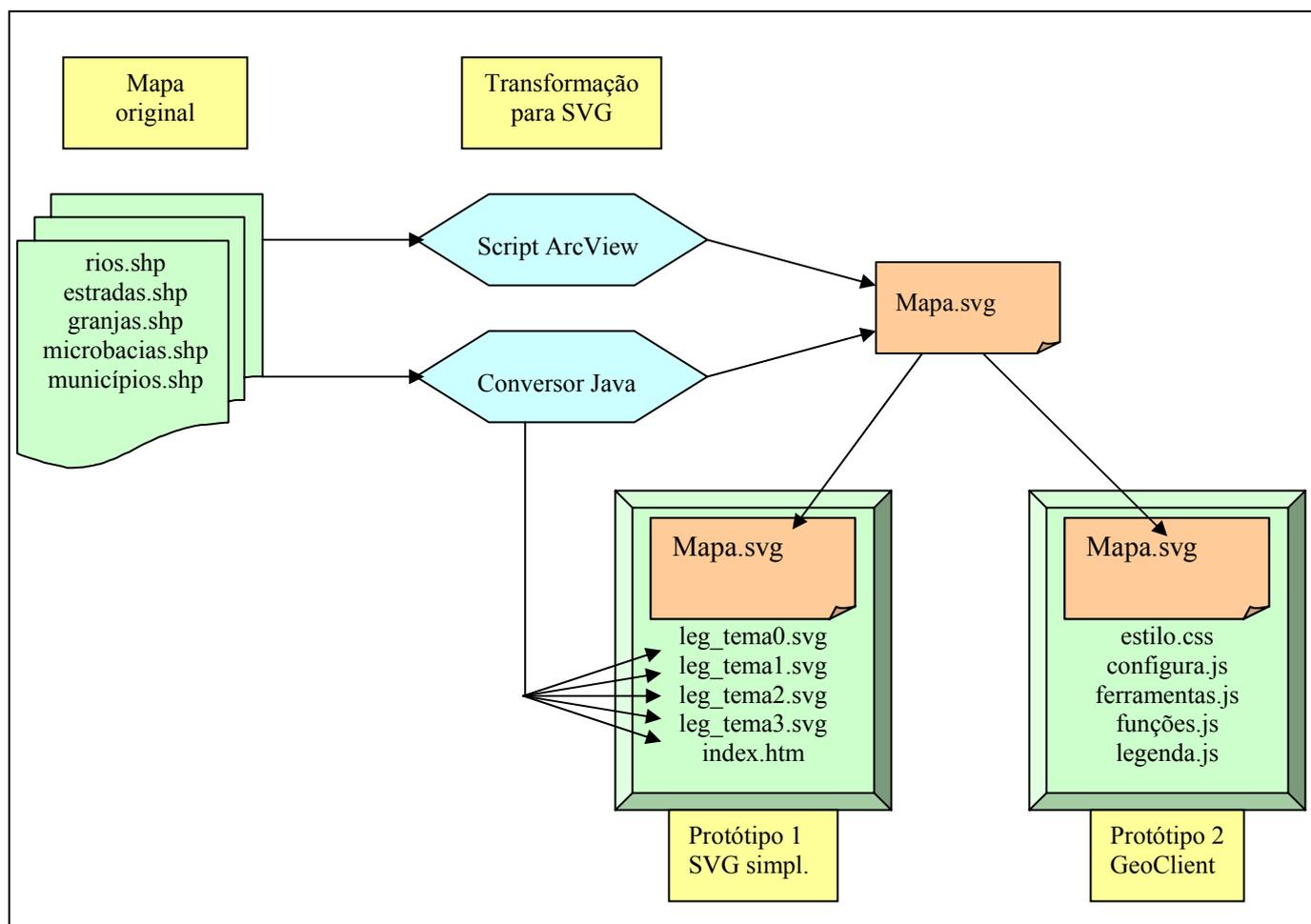


FIGURA 15 - Esquema da metodologia proposta

Protótipo 1

O mapa original é entendido aqui como um mapa composto por vários temas (*layers*), cada um deles correspondendo a um arquivo no formato “shape”. Para o caso do Protótipo 1, o conversor Java deve ser usado, pois as alterações em seu código foram feitas com essa finalidade. O conversor vai gerar então os seguintes arquivos: o mapa, *mapa.svg*, *index.htm*, *leg_tema0.svg*, *leg_tema1.svg*... *leg_temaN.svg*, onde N é o número de temas existentes.

Depois, basta colocar todos os arquivos no mesmo diretório do servidor, para que o mapa possa ser publicado.

Protótipo 2

Uma vez que as alterações no código dos arquivos tenham sido executadas, basta armazenar os arquivos *estilo.css*, *configura.js*, *ferramentas.js*, *funções.js* e *legenda.js* no mesmo diretório do servidor. Nesse caso, a transformação dos mapas originais (shape) para SVG pode ser feita tanto com o conversor Java como com o script do ArcView. Após transformação, basta colocar o arquivo SVG resultante (mapa.svg) no mesmo diretório dos demais arquivos, no servidor.

Ambos os protótipos podem se acessados no site:

www.paesimoveis.floripa.com.br/mapas/ariranha.

6 Conclusões

6.1 A linguagem SVG

SVG apresenta-se como uma linguagem usada cada vez com mais frequência, apontando para um cenário em que, provavelmente, se tornará um padrão muito difundido. Algumas vantagens são claras, mas a mais importante delas está no fato de SVG ser uma linguagem aberta, o que implica em baixo custo e muitas possibilidades de aplicação. O fato dos arquivos SVG serem escritos em XML puro também representa uma vantagem, principalmente pelo fato de permitir edição a partir de uma grande variedade de softwares, inclusive um simples editor de textos. Além disso, XML também representa compatibilidade com muitos sistemas diferentes, inclusive rodando em plataformas distintas. A compatibilidade com a tecnologia Java abre ainda mais o leque de possibilidades de interação com softwares e desenvolvimento de aplicativos, tanto para *Web* como *standalone*.

Além disso, ante a opção de apresentar mapas em forma de imagens matriciais (como JPEG e GIF), nota-se que tais arquivos necessitam ser editados em softwares apropriados, diferente de arquivos SVG que podem ser editados pelo *Notepad*, por exemplo, nativo do Windows. Quanto ao tamanho, arquivos SVG em geral são pequenos e mais comprimidos do que imagens BMP, JPEG ou GIF. Esse detalhe merece atenção especial, principalmente na geração do arquivo SVG, para que se inclua somente os dados necessários, e que se use um nível de generalização adequado, de tal forma que o número de vértices seja reduzido e assim o arquivo se torne menor. Outro aspecto fundamental para publicação de mapas, é que as imagens SVG são escalonáveis, permitindo portanto uso de recursos de zoom. No entanto, os recursos de zoom precisam ser limitados, pois a ampliação não deve ultrapassar o limite de precisão no qual o mapa original foi gerado. Da mesma forma a redução não deve ser exagerada a tal ponto que a densidade das informações no mapa não permitam visualização clara dos fenômenos representados.

As imagens matriciais, para que tenham tamanho pequeno, precisam ter baixa resolução. Isto significa que imagens apropriadas para impressão, com alta resolução, não são

apropriados para Internet, devido ao tamanho do arquivo. Já as imagens vetoriais em SVG podem ser impressas com alta qualidade em qualquer resolução.

Os mecanismos de busca geralmente representam a porta de entrada para o usuário da *Web*, já que é normalmente ali que se inicia uma pesquisa. Nessa situação, se um usuário procurar por uma palavra contida em uma imagem matricial ou em um documento no formato Macromedia Flash por exemplo (que é binário), seria impossível para o mecanismo de busca encontrar tal palavra. O mesmo texto contido em um mapa (ou gráfico) SVG pode ser selecionado e localizado pelos mecanismos de busca tanto do próprio do browser quanto da Internet.

Por outro lado, o maior problema do SVG atualmente é o fato de os principais navegadores (browsers) não suportarem a linguagem, o que obriga o uso de plugins. No entanto, vários sites especializados no assunto apontam uma tendência para que navegadores mais conhecidos, como Mozilla e Internet Explorer da Microsoft suportarem completamente a linguagem em versões futuras. O Adobe GoLive 5 já suporta SVG e o número de softwares para edição de SVG está crescendo. Isto significa que, atualmente, para visualizar arquivos SVG é necessário instalar plugins, como o Adobe SVG Viewer, provavelmente o mais difundido.

A linguagem SVG é relativamente recente, tendo se tornado uma recomendação do W3C em setembro de 2001. Quanto mais recente uma tecnologia, tanto maiores os incrementos adicionados de um ano para outro, já que a medida que o tempo avança a tecnologia vai se tornando mais comum e conseqüentemente mais testada pelos desenvolvedores, revelando “bugs” e principalmente sugerindo alterações, já que se trata uma linguagem aberta. Embora isso represente um rápido avanço, traz consigo o problema da incompatibilidade entre as versões. No início desse trabalho, a versão mais recente do plugin Adobe SVG Viewer era a versão 3.0. Alguns dos recursos estudados aqui foram implementados e testados sob esta versão. No entanto, ao instalar a versão mais recente percebeu-se que algumas funcionalidades foram perdidas, necessitando de alteração no código dos arquivos. Isto significa que usuários que acessem o site com diferentes versões do plugin podem ter resultados diferentes ao acessar o mesmo mapa, gerando a necessidade de adaptar as páginas para que se tornem compatíveis tanto com as versões anteriores como a atual de cada plugin. Isto pode representar um fator decisivo para a escolha desse mecanismo como forma de publicação de mapas na Internet.

Atualmente, o principal competidor do SVG é o software Flash, da Macromedia. Ambos possuem recursos semelhantes. A maior vantagem do SVG sobre o Flash é a sua compatibilidade com outros padrões (como XSL e DOM). Flash é uma tecnologia proprietária e não é código aberto.

6.2 Os protótipos

Embora a proposta desse estudo esteja voltada para visualização de mapas temáticos, do ponto de vista de estrutura de arquivos, a técnica pode ser facilmente adaptada para mostrar outros tipos de mapas, como mapas cadastrais e topográficos. Embora o protótipo seja recomendado para publicação de mapas simplificados, há que ressaltar que a Epagri (Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina) possui atualmente grande quantidade de mapas digitais nas áreas de Agronomia e Ambiental com características muito semelhantes à do projeto tomado como exemplo. Essas informações podem ser publicadas seguindo a metodologia proposta, aumentando assim seu alcance na difusão de informações que podem ser muito valiosas para técnicos, estudantes, governantes e a sociedade como um todo.

Uma vez estabelecida a necessidade de disseminação de dados geográficos na Internet, constata-se que, independente da solução adotada, sempre há necessidade de transferência de arquivos de imagem ou vetores que são utilizados para representação gráfica desses dados. Se estes arquivos forem muito grandes, cria-se uma limitação. Embora a capacidade e velocidade de transmissão de dados da rede Internet apresente claramente uma tendência de aumentar no futuro, resta a questão de procurar soluções para atender a demanda enquanto isso não acontece.

A demora para transmissão de dados na Internet causa impaciência nos usuários, e muitas vezes o site é abandonado por conta dessa demora, deixando de cumprir sua função de transmitir a mensagem e levar a informação ao usuário. Essa situação é válida não só para dados geográficos, como para qualquer outro tipo de informação transmitida na *Web*. OSSES (2000), afirma que mesmo utilizando técnicas de compactação ou organização dos dados, os usuários de aplicações de dados geográficos para Internet sofrem com o tempo de espera dos

arquivos para visualização. Nesse sentido, é válido qualquer esforço para minimizar o tamanho desses arquivos assim como evitar a transmissão desnecessária de arquivo e dados.

Os protótipos aqui estudados levaram esse problema em consideração, procurando uma solução prática para visualização de mapas simplificados, evitando a necessidade de transmissão de arquivos grandes. No entanto, isto limita a quantidade de informações, tanto em termos de quantidade de elementos geográficos representados no mapa, como em quantidade de atributos desses elementos.

O segundo protótipo apresenta certos recursos que auxiliam na configuração do mapa, por parte do usuário, para torná-lo mais adequado para visualização, de acordo com o tipo de legenda escolhida (escolha de cores, rótulos, número de classes e sobreposição de layers). No entanto, o potencial da linguagem SVG no sentido de qualidade de visualização cartográfica não foi suficientemente explorado, senão nos aspectos mencionados. De acordo com CECCONI & GALANDA (2002), muitos dos mapas encontrados na *Web*, assim como em aplicações GIS, pecam na qualidade cartográfica visual. Normalmente as razões são limitações técnicas das linguagens envolvidas, mas muitas vezes, mesmo que as limitações não existam, os princípios básicos da cartografia digital não são respeitados. Nesse sentido, cabe a recomendação de estudos futuros para implementar recursos nos protótipos que sigam tais princípios.

Outro aspecto não explorado nesse estudo, diz respeito à generalização do mapa para diferentes níveis de zoom. As linhas de contorno dos polígonos, rios e estradas precisam ser generalizadas quando o nível de zoom aplicado não exige mais detalhes, assim como o tamanho dos símbolos e rótulos. Isto melhora o aspecto visual do mapa. Existem estudos nesse sentido, como os de CECCONI & GALANDA (2002) e DAHINDEN(2002), mostrando que é possível usar os próprios recursos do SVG para fazê-lo.

6.3 Geral

Um mapa visualizado pela rede Internet pode possuir grande quantidade de informações espaciais, e quando bem estruturado, pode fornecer dados interessantes à tomada de decisões, ao planejamento rural, à organização jurídico-registral, à tributação imobiliária, à instituições de Pesquisa e Desenvolvimento, empresas públicas ou privadas. Para isso é

preciso que todas as informações que nele estejam contidas, sejam referenciadas espacialmente em uma Base Cartográfica, de forma correta, com a maior precisão possível.

Todo planejamento necessita de uma grande quantidade de informações que precisam ser confiáveis e estar ao alcance dos gestores. Nesse sentido o Cadastro Técnico Multifinalitário (CTM) representa a ferramenta primordial para o sucesso do planejamento. Os mapas da área em estudo permitirão a espacialização das informações e sua localização precisa. O mapa é uma peça insubstituível de estudo, não só no âmbito técnico como ecológico e ambiental. Uma parte importante do processo é o acesso que os gestores e pesquisadores terão às informações espaciais geradas pelo CTM. A apresentação de mapas pela rede Internet representa um meio eficiente de divulgação dos produtos obtidos com o uso de geoprocessamento.

Assim, o estudo de meios de apresentação de informações espaciais na Internet, contribui para a divulgação de dados valiosos produzidos pelas instituições de Pesquisa e Desenvolvimento, o que pode servir de instrumento para pesquisas técnicas em diversas áreas de conhecimento, como também pode servir como instrumento de gestão. Fica claro, portanto, que a apresentação de informações espaciais através da rede Internet representa um enorme potencial para administração pública.

7 Referências Bibliográficas

ALMEIDA, Maurício Barcellos. **Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares**. *Ci. Inf.*, maio/ago. 2002, vol.31, no.2, p.5-13. ISSN 0100-1965.

ARONOFF S. **Geographic Information Systems: A Management Perspective**. Ottawa: WDL Publications. 1989.

ARTÍFICE DA WEB. O que é a “Web Semântica”. Extraído de Artífice da Web, artigo “O que é a Web Semântica?”. Nov. 2004. {citado nov. 2004}. Disponível na World Wide Web: <http://www.artifice.web.pt/>.

BAX, Marcello Peixoto. **Introdução às linguagens de marcas**. *Ci. Inf.*, jan./abr. 2001, vol.30, no.1, p.32-38. ISSN 0100-1965.

BARRON ESTRADA, Maria Lucia. **MOOL: An object-oriented programming language with generics and modules**. Florida Institute of Technology, 2004, 203 pages. AAT 3123097.

BURROUGH, P.A. **Principles of Geographical Information System for land resources assessment**. In: "Monographs on Soil And Resources Survey", n. 12, Clarendon Press, Oxford, England, 1986.

BRYAN, M. **An introduction to the Extensible Markup Language (XML)**. [S. l.]: SGML Centre, 1997. [citado 13 mar. 2003]. Disponível na World Wide Web: www.sgml.u-net.com/xmlintro.htm

CÂMARA, Gilberto & MEDEIROS, José Simeão de. **Tendências de evolução do geoprocessamento**. In: Sistemas de informações geográficas: aplicações na agricultura. Coordenadores: Eduardo Delgado Assad e Edson Eyji Sano. 2ª Ed. Embrapa, SPI. Brasília, 1998.434 p.

CÂMARA, G.; MEDEIROS, C.B.; CASANOVA, M.A.; HEMERLY, A.; MAGALHÃES, G. **Anatomia de Sistemas de Informação Geográfica**. Cap. 3 “Arquitetura de sistemas de informação geográfica”. Escola de Computação, SBC, 1996.

CÂMARA, Gilberto; SOUZA, Ricardo Cartaxo Modesto de; PEDROSA, Bianca Maria; VINHAS, Lúbia; MONTEIRO, Antônio Miguel Vieira; PAIVA, João Argemiro;

CARVALHO, Marcelo Tilio de; GATTASS, Marcelo. TerraLib: **Technology in Support of GIS Innovation. II Workshop Brasileiro de Geoinformática, GeoInfo2000**. São Paulo, 2000.

CECCONI, Alesandro & GALANDA, Martin. **Adaptative zooming in web cartography**. University of Zurich. Zurich, Switzerland. 2002. [citado em out 2002] Disponível na World Wide Web: <http://www.svgopen.org/2002/papers/>.

COWEN D.J. **SIG versus CAD versus DBMS: what are the differences?**. In: "Introductory readings in Geographic Information Systems". Londres: Taylor and Francis. 1988.

DAHINDEN, Tobias. **Attractive vectormaps – a call for well arranged webmaps**. Institute of Cartography, Zurich. 2002. [citado em out 2002] Disponível na World Wide Web: <http://www.svgopen.org/2002/papers/>.

GOODCHILD, Michael F., PROCTOR, James D. e WRIGHT, Dawn J. **Demystifying the Persistent Ambiguity of GIS as "Tool" Versus "Science"**. In: "The Annals of the Association of American Geographers. 1997.

IETF — Internet Engineering Task Force. **Hypertext Markup Language - 2.0**. 1995. [citado mar. 2003]. Disponível na World Wide Web: <http://www.ietf.org/rfc/rfc1866.txt>.

LIMA, Paulo. **Intercâmbio de dados espaciais: modelos, formatos e conversores**. Dissertação de Mestrado do Curso de Computação Aplicada. Orientador: Dr. Gilberto Câmara. INPE (Instituto Nacional de Pesquisas Espaciais), São José dos Campos. 2002. 79 p.

MAGON, Ajay & SHUKLA, Reena. **LBS, the ingredients and the alternatives**. In: The Geospatial Resource Portal, Asian GPS Conference, 2001. Noida, UP, India. [citado fev 2004]. Disponível em <http://www.gisdevelopment.net/technology/lbs>.

MARISCO, Nelson. **Web mapas interativos como interface aos dados geoespaciais: uma abordagem utilizando-se tecnologias fontes abertas**. Tese de doutorado para o curso de Engenharia Civil, área de Cadastro Técnico Multifinalitário. Orientador: Prof. Dr.-Ing. Jürgen Philips, Co-orientador: Prof. Dr.-Ing. Norberto Hochheim. Universidade Federal de Santa Catarina. Florianópolis, 2004. 279 p.

MENEGUETTE, A. A. C. **Uma abordagem sócio-construtivista no ensino de cartografia** In: SIMPÓSIO INTERNACIONAL SOBRE MAPAS E GRÁFICOS PARA DEFICIENTES VISUAIS, 1994, São Paulo. Anais do Simpósio Internacional Sobre Mapas e Gráficos para Deficientes Visuais. São Paulo: usp, 1994. v.1. p.21 – 21.

MIRANDA, José Iguelmar. **Publicando Mapas na Web: Servlets, Applets ou CGI?**. Campinas : Embrapa Informática Agropecuária, 2003. 38 p. : il. – (Documentos / Embrapa Informática Agropecuária ; 28).

NEWMANN, A.; WINTER, A. M.. **Vector-based Web Cartography: Enabler SVG**. Disponível em: <<http://www.carto.net/papers/svg/2001>>. Acesso em: 18 fev. 2005.

OGC - OPEN GIS CONSORTIUM. **Geography Markup Language (GML) Implementation Specification, version 2.1.1. 2002**. OpenGIS. [citado 10 set. 2004]. Disponível na World Wide Web: <http://www.opengis.net/gml/02-009/GML2-11.html>.

OSSES, José Roberto, J.Paiva, G.Câmara. **Arquiteturas Cliente-Servidor para Bibliotecas Geográficas Digitais**. II Simpósio Brasileiro de Geoinformática, São Paulo, 2000.

OSSES, José Roberto. **Arquiteturas cliente-servidor para disseminação de dados geográficos**. Dissertação de Mestrado do Curso de Computação Aplicada. Orientadores: Dr. Gilberto Câmara. Dr. João Paiva. INPE (Instituto Nacional de Pesquisas Espaciais), São José dos Campos. 2000. 101 p.

PELKIE, Craig. **Create Dynamic Web Graphics with SVG**. 2001. [citado nov. 2004]. Disponível na World Wide Web: <http://www.web400.com/download/SVG/SVG.htm>.

PRESCOD, Paul. **SVG: A Sure Bet**. 16 de jul. 2003. [citado out. 2004]. Disponível na World Wide Web: <http://www.xml.com/pub/a/2003/07/16/svg-prescod.html>.

ROCHA, Helder Lima Santos da, **Web design e HTML avançado**. 4a. edição. Coleção de textos independentes sobre Web design e HTML. ___ p. IBPINET – São Paulo, SP, Março de 2000.

SANTOS, Ricardo José Leal dos; MEIRELLES, Margareth Simões Penello; MAGALHÃES, Paulo Maurício S. **Utilização da XML/GML, XML/XSLT e XML/SVG no Contexto das Funcionalidades Relacionadas às Estratégias de Visualização Cartográfica: Uma Discussão Introdutória**. *Ci Inf.* [on line] julho de 2001. [citado 11 mar. 2003]. Disponível na World Wide Web: <http://www.ic.unicamp.br/~cmbm/geoinfo/papers/136maggie.pdf>.

SCHÜTZER, Waldeck & MASSAGO, Sadao. **Programação Java**. Departamento de Matemática. Universidade Federal de São Carlos – UFSCa. *Ci Inf.* [on line] 26 dez. 1999. [citado nov. 2004]. Disponível na World Wide Web: <http://www.dm.ufscar.br/~waldeck/curso/java/>.

TEK – SAPO, Revista eletrônica. **Consórcio W3C aprova novas recomendações para gráficos vectoriais**. [on line] outubro 2001. [citado 09 out. 2001]. Disponível na World Wide Web: <http://tek.sapo.pt/4M0/374351.html>

TURNER, Linda. **SGML Users' Group History**. Corporate Communications Avalanche. Boulder. 2003. [citado nov. 2004]. Disponível na World Wide Web: <http://xml.coverpages.org/sgmlhist0.html>.

VASSOLER, Gilmar L.; BRITO, Halisson M. de; BORTOLON, Saulo. **Desenvolvimento de aplicações GIS para Internet: de componentes ActiveX até a GML**. *Ci. Inf.* [on line] junho 2001. [citado mar. 2003]. Disponível na World Wide Web: <http://www.ic.unicamp.br/~cmbm/geoinfo/papers/106gilmar.pdf>.

W3 CONSORTIUM. **Extensible Markup Language (XML): W3C recommendation 6. 2. ed. versão 1.0**. Oct. 2000. [citado 10 fev. 2001]. Disponível no Worl Wide Web: <http://www.w3c.org/>.

W3C CONSORTIUM. **HyperText Markup Language (HTML) Home Page**. set. 2004. [citado nov. 2004]. Disponível na World Wide Web: <http://www.w3.org/MarkUp/>.

W3SCHOOLS. **Introductions to SVG**. jan. 2004. [citado out. 2004]. Disponível na World Wide Web: <http://www.w3schools.com/svg/>.

WIKIPEDIA. **Scalable Vector Graphics**. jun. 2004. Este artigo é licenciado sob as condições de *GNU Free Documentation License*. Extraído de Wikipedia, artigo “Scalable Vector Graphics”. [citado nov. 2004] Disponível na World Wide Web: <http://encyclozine.com/SVG>.

WWW ISO URL. **Platform for Internet Content Selection URL**. 1997. [citado mar. 2003]. Disponível na World Wide Web: <http://www.w3.org/PICS>.

8 – Glossário

Este glossário reúne alguns termos estudados durante a pesquisa e pode servir para dirimir dúvidas do leitor. Cada termo contém uma definição seguida de comentários pertinentes. Os termos foram separados em grupos: instituições, softwares, tipos de software, linguagens e outros termos.

Instituições:

Free Software Foundation (FSF)

Fundação para o Software Livre (FSF). É uma fundação sem fins lucrativos, filantrópica, dedicada à eliminação de restrições sobre a cópia, redistribuição, entendimento e modificação de programas de computadores.

Comentários: Atua promovendo o desenvolvimento e o uso de software livre em todas as áreas da computação, mas particularmente, ajudando a desenvolver o sistema operacional GNU. Muitas organizações distribuem qualquer software livre que por acaso esteja disponível. Em contraste, a Fundação para o Software Livre se concentra no desenvolvimento de novos softwares livres - e em tornar este software um sistema coerente que possa eliminar a necessidade de se utilizar software proprietário.

Além de desenvolver o GNU, a FSF distribui cópias do software GNU e manuais por uma taxa de distribuição, e aceita doações dedutíveis sobre os impostos para suportar o desenvolvimento do GNU. A maior parte dos fundos da FSF vem deste serviço de distribuição.

W3C (World Wide *Web* Consortium)

Trata-se de um consórcio de âmbito mundial (pessoa jurídica legalmente constituída), com participação de diversas instituições e colaboradores que objetiva levar a *World Wide Web* a atingir seu potencial máximo através do desenvolvimento de protocolos comuns que promovam sua evolução e garantam sua interoperabilidade.

Comentários: Atualmente a W3C tem mais de 450 Membros (empresas, instituições, fundações e outras formas de sociedade legalmente constituídas) e um quadro de aproximadamente 70 pessoas em tempo integral a nível global que contribuem para o desenvolvimento de especificações de W3C e software.

Em outubro de 1994, Tim Berners-Lee, considerado o inventor da *Web*, fundou o W3C dentro do *Massachusetts Institute of Technology, Laboratory for Computer Science* [MIT/LCS] em colaboração com o *European Organisation for Nuclear Research* [CERN], local onde a *Web* se originou, com suporte da *Defense Advanced Research Project Agency*

[DARPA] e da *European Commission*. Em abril de 1995 o INRIA (*Institut National de Recherche en Informatique et Automatique*) tornou-se o primeiro host europeu da W3C, seguido pelo *Keio University of Japan (Shonan Fujisawa Campus)* na Ásia em 1996. Hoje existem diversos escritórios da W3C em diversos países, onde o trabalho é efetivado.

Na prática, a W3C organiza o trabalho necessário para o desenvolvimento da tecnologia usada na *Web*, em suas atividades. Cada atividade tem sua própria estrutura, mas uma atividade tipicamente consiste de um grupo de trabalho, um grupo de interesse e um grupo de coordenação. Conhecendo a estrutura dessa atividade, esses grupos geralmente produzem diversos tipos de relatórios técnicos, chamados de “Recomendações” assim como exemplos de códigos. É altamente recomendável que qualquer desenvolvedor de tecnologia para *Web* (inclusive desenvolvedores de sites) conheçam as recomendações da W3C.

Softwares:

GNU

É um projeto de desenvolvimento de um sistema operacional livre.

Comentários: A palavra GNU é um acrônimo recursivo para “*GNU’s not Unix*” ou “GNU não é Unix” que se pronuncia "guh-NEW" ou "guniw". A palavra também designa um antílope com cabeça e chifres semelhantes aos do búfalo, nativo do continente Africano, por isso o desenho de tal animal é usado algumas vezes como ilustração associada ao software. O projeto GNU foi lançado em 1984 para desenvolver um sistema operacional completo e livre, similar ao Unix. Variações do sistema GNU, que utilizam o núcleo Linux, são hoje largamente utilizadas; apesar desses sistemas serem normalmente chamados simplesmente de “Linux”, segundo a FSF eles seriam mais precisamente designados de GNU/Linux.

PostGIS

Software que habilita suporte para objetos geográficos no banco de dados objeto-relacional PostgreSQL.

Comentários: O PostGIS permite que um servidor PostgreSQL seja usado como um banco de dados espacial para sistemas de informações geográficas (SIG), de forma semelhante à extensão SDE da ESRI ou à extensão Oracle’s Spatial, da Oracle. O PostGIS segue as especificações propostas pelo OpenGIS *Simple Features Specifications for SQL*. Foi desenvolvido inicialmente pela *Refractions Research Inc.* como um projeto de pesquisa de banco de dados espaciais com fonte aberta (*open source*). É distribuído de acordo com os termos de GNU – General Public License.

PostgreSQL

É um sofisticados sistema de banco de dados objeto-relacional, enquadrando-se na definição de Software Livre.

Comentários: O banco de dados PostgreSQL foi desenvolvido na Universidade de Berkeley, nos anos 80, como um projeto acadêmico e se encontra na versão 7.3, sendo um projeto mantido pela comunidade *Open Source*. A coordenação do desenvolvimento do PostgreSQL é executada pelo *Global Development Group*, este é formado por um amplo

grupo de desenvolvedores ao redor do mundo, o que faz o PostgreSQL ter uma constante evolução no que se refere à correção de bugs e implementação de novas funcionalidades.

SIG

Acrônimo de “Sistema de Informações Geográficas”. Igual à GIS, do inglês *Geographical Information System*.

- "Qualquer conjunto de procedimentos manuais ou baseados em computador destinados a armazenar e manipular dados referenciados geograficamente" (ARONOFF, 1989);

- " SIGs constituem um conjunto de ferramentas para coleta, armazenamento, recuperação, transformação e exibição de dados espaciais do mundo real para um conjunto particular de propósitos " (BURROUGH, 1986);

- "Um sistema de apoio à decisão que envolve a integração de dados espacialmente referenciados, em um ambiente para resolução de problemas" (COWEN, 1988);

- "Um sistema de informações baseado em computador que permite a captura, modelagem, manipulação, recuperação, análise e apresentação de dados georreferenciados" (WORBOIS, 1995).

Comentários:

Por tratar-se de uma tecnologia em franco processo de desenvolvimento, torna-se difícil chegar a uma definição de SIG que satisfaça ao mesmo tempo os envolvidos com seu uso, seu desenvolvimento, e até mesmo àqueles que fazem seu marketing. De acordo com GOODCHILD, 1997, há inclusive os que chegam a considerar SIG como uma ciência, e não como uma ferramenta.

Os Sistemas de Informações Geográficas têm uma gama muito grande de aplicações, havendo inclusive sistemas que, com frequência, usam as mesmas ferramentas de SIG, mas aplicam-nas em situações muito diferentes. Assim, os diferentes grupos de usuários chegam à sua própria definição de SIG.

Nota: não confundir com SIG de “Sistema de Informações Gerenciais”.

Autores:

ARONOFF S. (1989) *Geographic Information Systems: A Management Perspective*. Ottawa: WDL Publications.

BURROUGH P.A. (1986) *Principles of Geographic Information Systems for Land Resources Assessment*, em "Monographs on Soil And Resources Survey", n. 12, Oxford: Clarendon Press.

COWEN D.J. (1988) SIG versus CAD versus DBMS: what are the differences?, em "Introductory readings in Geographic Information Systems". Londres: Taylor and Francis.

GOODCHILD, Michael F., PROCTOR, James D. e WRIGHT, Dawn J. (1997) Demystifying the Persistent Ambiguity of GIS as "Tool" Versus "Science", em "The Annals of the Association of American Geographers.

WORBOYS, Michael F. (1995) *GIS: A Computing Perspective*. Londres: Taylor and Francis.

Tipos de Software:

Código Aberto (Open Source)

O termo é usado para designar a mesma idéia definida aqui para software livre (veja Software Livre).

Comentários: Apesar da similaridade com Software Livre, a FSF destaca uma diferença de “filosofia” entre os dois termos, explicando que em 1998, alguns desenvolvedores participantes da comunidade de Software Livre (*Free Software community*) começaram a usar o termo Código Aberto (*Open Source*) ao invés de Software Livre para descrever mais precisamente o trabalho que faziam. Assim o termo Código Aberto rapidamente foi associado à uma abordagem diferente, com filosofia própria, valores diferentes, principalmente nos critérios usados para licenciamento de programas. Ainda de acordo com a FSF, hoje, o movimento de Software Livre e o movimento de Código Aberto possuem objetivos e metas diferentes. A diferença fundamental entre os dois movimentos está nos seus valores, seus pontos de vista à respeito do assunto. Para o movimento Código Aberto, a decisão sobre qual software deve ser de código aberto ou não, é uma questão técnica

e não ética. Poderia-se dizer que Código Aberto é uma metodologia de desenvolvimento de software, enquanto Software Livre é um movimento social. Para o movimento Código Aberto um software não-livre é uma solução sub-otimizada. Para o movimento Software Livre, um software não-livre é um problema social e o software livre é a solução.

Freeware

O termo “*freeware*” não possui uma definição clara e aceita, mas é muito usada para pacotes que permitem redistribuição mas não modificação, são geralmente gratuitos (e seu código fonte não está disponível). Estes pacotes portanto *não* são software livre.

Shareware

Shareware é um software com permissão para redistribuir cópias, mas impõe que qualquer um que continue usando uma cópia *deve* pagar por uma licença.

Comentários: Normalmente pode-se usar o programa por um período experimental e findo esse período é necessário registrar-se, pagando uma licença. Não é Software Livre.

Software de Domínio Público (Public Domain Software)

Software no domínio público é software não protegido por copyright., o que significa que algumas cópias ou versões modificadas podem não ser livres.

Comentários: Algumas vezes o termo “domínio público” é usado de forma imprecisa para se referir à Software Livre ou ainda para “disponível gratuitamente”. No entanto, “domínio público” é um termo legal e significa, precisamente, “sem copyright”.

Software Livre (Free Software)

Refere-se a qualquer software que venha com permissão para cópia, uso e distribuição, com ou sem modificações, gratuitamente ou não. Em particular, isso significa que o código fonte deve estar disponível, mas não necessariamente será gratuito.

Comentários: Na língua inglesa a palavra *free* pode significar tanto *liberdade* como *de graça*, o que gerou certa confusão.

De acordo com a FSF, software livre é uma questão de liberdade, não preço [em inglês: "free software is a matter of freedom, not price"]. Mas empresas de software proprietários às vezes usam o termo "software livre" (*free software*) para referir-se ao preço. Algumas vezes a intenção é mostrar que é possível obter uma cópia binária sem nenhum custo; outras vezes significa que uma cópia está incluída no computador que está sendo comprado. De qualquer forma isso não se relaciona com a idéia de Software Livre proposta pela FSF no projeto GNU.

Software Proprietário (Proprietary Software)

É todo aquele programa de computador que não é livre ou apresenta algum tipo de restrição. Seu uso, redistribuição ou modificação é proibido legalmente, ou requer permissão, ou é restrito de tal forma que não se possa efetivamente alterá-lo ou distribuí-lo livremente.

Linguagens

SGML

Sigla de *Standard Generalized Markup Language*. É um sistema para definição de linguagens de marcação. Também pode ser compreendido como uma linguagem de marcas para a representação de informações em texto.

Comentários: Os autores introduzem marcas nos documentos que representam informações estruturais, de apresentação e semânticas ao longo do texto. HTML é um

exemplo desse tipo de linguagem de marcação. A linguagem foi reconhecida como um padrão ISO (8879) em 1986. SGML não é um conjunto predeterminado de marcas, e sim uma linguagem para se definirem quaisquer conjuntos de marcas, uma linguagem *autodescritiva*; cada documento SGML carrega consigo sua própria especificação formal, conhecido como *Data Type Document* (DTD).

HTML

Sigla de *HyperText Markup Language*. É um tipo de linguagem de marcação considerada como uma *lingua franca* para publicar hipertexto na *World Wide Web*. É um formato não proprietário baseado em SGML, e pode ser criado e processado com uma grande quantidade de ferramentas, desde simples processadores de texto até sofisticadas ferramentas de autoria do tipo WYSIWYG. **Comentários:** HTML usa marcações do tipo `<h1>` e `</h1>` para estruturar o texto em cabeçalhos, parágrafos, listas hiperlinks e outros.

XML

Sigla de *Extensible Markup Language*. É um formato de texto derivado de SGML, simples e muito flexível. É um formato universal de estrutura de documentos na *Web*. Ele permite definir seus próprios formatos de marcações, principalmente naqueles pontos em o HTML não consegue suprir. Vem sendo usado para representar uma grande variedade de dados incluindo gráficos e formatos de metadados.

XHTML

Acrônimo de *Extensible HyperText Markup Language* (XHTML™). Compreende uma família de tipos de documentos e módulos que reproduzem, reagrupam e estendem a capacidade do HTML. É uma reformulação do HTML baseado em XML. XHTML é o sucessor do HTML.

É uma linguagem para descrever gráficos bi-dimensionais e aplicações gráficas escritas em XML.

Outros termos:

Copyright

Direito autoral. Direito exclusivo de imprimir, reproduzir ou vender obra literária, científica ou artística. É uma proteção legal que cobre trabalhos literários, científicos e artísticos publicados ou não, ou qualquer forma de expressão, desde que exista algo tangível ou material.

Comentários: Se for um ensaio, um jogo, uma música, uma fotografia, código de HTML ou um gráfico do computador que possa ser impresso, ser gravado em fita ou ser salvo em disco rígido (ou qualquer meio magnético), então pode ser protegido. As leis de copyright concedem ao criador o direito exclusivo de reproduzir, fazer alterações, distribuir publicamente, executar e indicar o trabalho. O termo *exclusivo* significa que somente o criador de tal trabalho tem esse direito, e não qualquer um que tenha acesso à ele.

Esta definição básica à respeito do copyright baseia-se no que foi definido na **convenção de Berne para a proteção da propriedade literária e artística (convenção de Berne)** e também na **Universal Copyright Convention (UCC)** e não consulta as leis de nenhum país. Entretanto, as leis de copyright variam do país para país mas em regra geral não ferem nem fornecem menos proteção do copyright do que a convenção de Berne.

GNU – GPL

Sigla de Licença Pública Geral do GNU ou GPL - *General Public License*.

Comentários: As licenças de muitos softwares são desenvolvidas para restringir sua liberdade de compartilhá-lo e mudá-lo. Contrária a isso, a Licença Pública Geral GNU pretende garantir sua liberdade de compartilhar e alterar software livres, garantindo que o

software será livre e gratuito para os seus usuários. Esta Licença Pública Geral aplica-se à maioria dos softwares da *Free Software Foundation* e a qualquer outro programa cujo autor decida aplicá-la. Pode ser aplicada à qualquer programa livre.

Dado espacial

O termo “dado espacial” denota qualquer tipo de dado que descreva fenômenos aos quais esteja associada alguma dimensão espacial como, por exemplo, estruturas moleculares de um composto químico.

Comentários: Assim os entes espaciais representados em um mapa, como casa, ponte, igreja, rio, etc., são dados espaciais. Quando esta dimensão espacial é a superfície terrestre, tem-se um tipo especial de dados espaciais: os dados geo-referenciados ou dados geográficos.

Dado geográfico

É um dado espacial geo-referenciado, isto é, possui uma *localização* expressa como coordenadas de um mapa, com sistema de projeção definido e que representa um objeto real na superfície da terra.

Comentários: os dados geográficos podem conter *atributos descritivos*, que por sua vez podem estar representados num banco de dados convencional. Esses atributos são características dos dados geográficos, e podem ser divididos em característica não-espacial, que representa os atributos textuais do fenômeno, ou seja, nome e atributos específicos do mesmo; característica espacial, que é o próprio georeferenciamento; e as características gráficas, que são as representações pictóricas dos dados.

WebMapping

É o processo de construção e design de mapas para visualização na Internet.

Comentários: este processo baseia-se em vários princípios e modelos. Os mapas podem ser diferentes, de acordo com a finalidade, as suas características e os níveis de interatividade desejados. Os mapas encontrados na *Web* podem servir apenas para mostrar a localização de uma entidade qualquer, ou incorporar um motor de pesquisa para consulta de tudo o que ele representa.

Determinados tipos de mapas devem permitir o uso do pan/zoom, bem como mostrar ou esconder determinados níveis (camadas ou layers), seja porque o usuário só tem interesse em visualizar parte da informação, ou porque deixa de fazer sentido, a uma determinada escala, ver todos os pormenores do mapa.