

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE CIÊNCIAS RURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM GEOMÁTICA**

**SISTEMA MULTIUSUÁRIO PARA PUBLICAÇÃO DE
INFORMAÇÕES GEORREFERENCIADAS COM BASE
EM FERRAMENTAS DE CÓDIGO FONTE ABERTO**

DISSERTAÇÃO DE MESTRADO

Henrique Schetinger Filho

Santa Maria, RS, Brasil

2005

S327s

Schetinger Filho, Henrique,

Sistema multiusuário para publicação de informações georreferenciadas com base em ferramentas de código fonte aberto / por Henrique Schetinger Filho ; orientador Rudiney Soares Pereira. – Santa Maria, 2005.
52 f. : il.

Dissertação (mestrado) – Universidade Federal de Santa Maria, 2005.

1. Geomática 2. Sistema de informação geográfica 3. SIG 4. Geoprocessamento 5. *Webmaps* 6. Código Fonte Aberto I. Pereira, Rudiney Soares, orient. II. Título

CDU: 528.7:004.4

Ficha catalográfica elaborada por
Luiz Marchiotti Fernandes CRB-10/1160
Biblioteca Setorial do Centro de Ciências Rurais/UFSM

SISTEMA MULTIUSUÁRIO PARA PUBLICAÇÃO DE INFORMAÇÕES GEORREFERENCIADAS COM BASE EM FERRAMENTAS DE CÓDIGO FONTE ABERTO

por

Henrique Schetinger Filho

Dissertação apresentada ao Curso de Mestrado do
Programa de Pós-Graduação em Geomática,
Área de Sensoriamento Remoto, da Universidade Federal de Santa
Maria, como requisito parcial para obtenção do grau de
Mestre em Geomática.

Orientador: Prof. Rudiney Soares Pereira

Santa Maria, RS, Brasil

2005

**Universidade Federal de Santa Maria
Centro de Ciências Rurais
Programa de Pós-Graduação em Geomática**

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

**SISTEMA MULTIUSUÁRIO PARA PUBLICAÇÃO DE
INFORMAÇÕES GEORREFERENCIADAS COM BASE
EM FERRAMENTAS DE CÓDIGO FONTE ABERTO**

elaborada por
Henrique Schetinger Filho

como requisito parcial para obtenção do grau de
Mestre em Geomática

COMISSÃO EXAMINADORA:

Rudiney Soares Pereira, Dr. (UFSM)
(Presidente/Orientador)

Enio Giotto, Dr. (UFSM)

José Américo de Mello Filho, Dr. (UFSM)

Santa Maria, 26 de abril de 2005.

AGRADECIMENTOS

À Universidade Federal de Santa Maria, pela oportunidade de participar do Programa de Pós-Graduação em Geomática.

Ao Prof. Rudiney Soares Pereira, por todo o seu apoio, disponibilidade, amizade e conhecimentos que me ajudaram a chegar até aqui.

A todos professores do PPGG que colaboraram com sua dedicação e conhecimento.

Aos funcionários do curso que contribuíram com sua disponibilidade e auxílio em tarefas administrativas.

À Rosa, Victor, Christina, Luísa e Isabela, pelo incentivo, carinho, compreensão e todo o apoio. Amo vocês.

A Eugênio, Catarina, Pedro e João por todo o apoio.

Aos colegas do Curso de Geomática pela amizade e companheirismo.

A Elvis pelo auxílio técnico, apoio e amizade.

A Cristhiano por todo o apoio e amizade.

A todos os colegas do NDIGe.

A todas as pessoas que de alguma forma contribuíram neste trabalho.

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Geomática
Universidade Federal de Santa Maria

SISTEMA MULTIUSUÁRIO PARA PUBLICAÇÃO DE INFORMAÇÕES GEORREFERENCIADAS COM BASE EM FERRAMENTAS DE CÓDIGO FONTE ABERTO

AUTOR: HENRIQUE SCHETINGER FILHO

ORIENTADOR: RUDINEY SOARES PEREIRA

Data e Local da Defesa: Santa Maria, 26 de abril de 2005.

Os sistemas de informações geográficas (SIG) são ferramentas de grande potencial para apoiar a gestão de organizações. A implantação e uso desses sistemas requerem, como requisitos fundamentais, recursos computacionais (hardware e software) e recursos humanos (pessoal qualificado). Devido ao custo elevado de alguns sistemas e envolvimento de muitos profissionais na área de geoprocessamento, a implantação destes sistemas pode se tornar onerosa. Uma possível solução para reduzir custos é o desenvolvimento de sistemas de informação geográfica capazes de serem administrados remotamente, através de interface *web* e da utilização de ferramentas de software livre e de código fonte aberto. Este trabalho tem por objetivo o desenvolvimento de um sistema multiusuário para publicação de informações georreferenciadas para apoio à tomada de decisões. O sistema criado está baseado em: interface de visualização para *web*, construído a partir de ferramentas de software livre e código fonte aberto; mecanismo para criação de novas consultas através da *web*; mecanismo de administração dos dados através da *web*; independência de plataforma no cliente com o uso de navegador *web*; sistema de gerenciamento de banco de dados e ferramenta para geração de mapas através da Internet (*webmaps*).

Palavras-chave: Sistemas de Informação Geográfica, Geoprocessamento, *Webmaps*, Código Fonte Aberto.

ABSTRACT

Master Dissertation
Master Course in Geomatics
Universidade Federal de Santa Maria

**MULTIUSER SYSTEM TO PUBLISH
GEOREFERENCED INFORMATION BASED ON
OPEN SOURCE CODE TOOLS**

AUTHOR: HENRIQUE SCHETINGER FILHO

ADVISOR: RUDINEY SOARES PEREIRA

Santa Maria, april 26, 2005.

The geographic information systems (GIS) are great potential tools to support organizations management. The requirements for implantation and use of this system are computational (the hardware and software) and human resources. Due to the high cost of some systems and involvement of many professionals in the geoprocessing area, the implantation of these systems can become onerous. A possible solution to reduce costs is the development of geographic information systems capable to be remotely managed, through web interface and using free software and open source code tools. This work has for its objective the development of a multiusing system to publish georeferenced information to support management tasks. The system is based on a web visualization interface, made by free software and open source code tools; tools for data administration through web; platform independence in the client with web browser; database management system; tools to generate web maps to the InterNet and ways to create new queries through web.

Key words: Geographic Information Systems, Geoprocessing, Webmaps, Open Source Code.

LISTA DE FIGURAS

Figura 1:	Arquitetura cliente/servidor para publicação de mapas na web.....	9
Figura 2:	Arquitetura do <i>Geomedia Web Map</i> da Intergraph.....	10
Figura 3:	Arquitetura do ARCIMS.....	11
Figura 4:	Arquitetura da tecnologia <i>Chameleon</i>	23
Figura 5:	Arquitetura geral do sistema proposto.....	28
Figura 6:	Papéis dos usuários do Sistema <i>web</i>	31
Figura 7:	Passos para publicação de um <i>webmap</i>	32
Figura 8:	Esquema de apresentação da tela de consulta do mapa.....	33
Figura 9:	Interface de entrada(<i>login</i>) no sistema.....	37
Figura 10:	Interface de edição de projetos de mapas.....	38
Figura 11:	Interface de edição de planos de informação.....	39
Figura 12:	Tela de consulta de um <i>webmap</i>	40
Figura 13:	Exemplo de aplicação de um <i>webmap</i> com dados do Campus da UFSM.....	43

LISTA DE SIGLAS

- API** – *Application Programming Interface*
- CGI** – *Common Graphics Interface*
- CWC** – *Chameleon Web Mapping Components*
- ESRI** – *Environmental Systems Research Institute*
- GIF** – *Graphics Interchange Format*
- GIS** – *Geographical Information System*
- GPL** – *Gnu Public License*
- GRASS** - *Geographic Resources Analysis Support System*
- HTML** – *Hypertext Markup Language*
- HTTP** – *Hipertext Transfer Protocol*
- INPE** – *Instituto Nacional de Pesquisas Espaciais*
- PDF** – *Portable Document Format*
- PNG** – *Portable Network Graphics*
- RAM** – *Random Access Memory*
- RGB** – *Red Green Blue*
- SGBD** – *Sistema de Gerenciamento de Banco de Dados*
- SIG** – *Sistema de Informações Geográficas*
- TIFF** – *Tagged Image File Format*
- SQL** – *Structured Query Language*
- SSL** – *Secure Sockets Layers*
- UMN** – *Universidade de Minnesota*

LISTA DE TABELAS

Tabela 1 – Gerações de SIG.....	7
Tabela 2 – Estrutura da tabela de Usuários.....	41
Tabela 3 – Estrutura da tabela de Planos de Informação.....	41
Tabela 4 – Estrutura da tabela de Parâmetros do Sistema.....	41
Tabela 5 – Estrutura da tabela de Projetos.....	42
Tabela 6 – Descrição das funções da barra de ferramentas.....	44

SUMÁRIO

RESUMO.....	iv
ABSTRACT.....	v
LISTA DE FIGURAS.....	vi
LISTA DE SIGLAS.....	vii
LISTA DE TABELAS.....	viii
1. INTRODUÇÃO.....	1
2. REVISÃO DE LITERATURA.....	4
2.1 Software Livre.....	4
2.2 Software Proprietário.....	4
2.3 Sistemas de Informações Geográficas.....	5
2.4 Tipos de dados em Geoprocessamento.....	6
2.5 Representação computacional de mapas.....	7
2.6 Aspectos da publicação de mapas na web.....	8
2.6.1 Ferramentas Comerciais para publicação de mapas na web....	10
2.6.2 Ferramentas de software livre para publicação de mapas na web.....	13
2.7 Armazenamento de Dados em SIG.....	15
2.8 SGBD e Licenças de uso.....	16
2.9 SGBD comerciais.....	17
2.10 SGBD de licença de software livre.....	18
2.10.1 Postgresql.....	18
2.10.2 Mysql.....	18
2.11 Extensões espaciais para SGBD.....	19
2.12 O Padrão OGC para interoperabilidade em SIG.....	20
2.13 Sistemas Operacionais.....	21
2.14 Sistema de Moldes Chameleon para UMN Mapserver.....	22
2.15 Linguagem de programação PHP.....	25
3. MATERIAL E METODOLOGIA.....	27
3.1 Material.....	27
3.1.1 Hardware.....	27
3.1.2 Software.....	27
3.2 Metodologia.....	27

3.2.1 Arquitetura do Sistema.....	28
3.2.2 Caracterização dos usuários do sistema.....	29
3.2.3 Função dos usuários no sistema.....	30
3.2.4 Funcionalidade do sistema.....	32
3.2.5 Tabelas do sistema.....	34
4. RESULTADOS E DISCUSSÃO.....	35
4.1 Conceitos básicos para publicação das informações no sistema proposto.....	35
4.1.1 Projeto.....	35
4.1.2 Planos de informação.....	36
4.2 A Interface Homem-Computador.....	37
4.3 Tabelas do Banco de Dados.....	40
4.4 Exemplo de uso do sistema proposto: mapa do Campus da UFSM.....	42
4.5 Aspectos importantes do uso do sistema.....	44
5. CONCLUSÃO.....	46
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	48

1. INTRODUÇÃO

O desafio de administrar um município envolve um conjunto de fatores, tais como o planejamento, avaliação de recursos, estudo de dados regionais, controle de recursos financeiros e outros.

Os sistemas de informações geográficas (SIG) são ferramentas de grande potencial para apoiar a gestão de municípios. Pode-se citar vários exemplos de uso destes sistemas neste escopo: saúde pública, agricultura, tributação, saneamento, controle de tráfego, controle de propriedades urbanas e rurais, entre outros.

Os requisitos para implantação e uso desse sistema envolvem: recursos computacionais (hardware e software) e recursos humanos: pessoal qualificado.

Atualmente municípios de grande porte no Brasil já fazem uso de SIG para apoiar projetos na gestão municipal.

Devido ao custo elevado de alguns sistemas e envolvimento de muitos profissionais na área de geoprocessamento, a implantação destes sistemas pode se tornar onerosa.

Uma parcela grande de municípios em todo Brasil possui poucos recursos financeiros para aplicar na implantação e uso de SIG. Sofrem por falta de planejamento e de apoio. O uso de SIG para gestão desses municípios poderá auxiliar muito o seu desenvolvimento. Para tanto, é necessário facilitar o acesso desta tecnologia propondo soluções para a redução de custo na implantação e uso destes sistemas.

Uma possível solução para esta carência de recursos é o desenvolvimento de sistemas de informação geográfica de custos reduzidos, capazes de serem administrados remotamente, através de interface Web. Para redução de custos é possível utilizar ferramentas de software livre e código fonte aberto. O SIG pode também assumir uma interface *web* para publicação de mapas, o que hoje é conhecido como *webmap*. Para isto seria possível que pessoas qualificadas em Universidades e centros de pesquisas pudessem auxiliar, através de parcerias e convênios firmados, a configuração e manipulação de SIG remotamente, bem como implantar estes sistemas em vários municípios a custos viáveis. O apoio poderia ser dado, tanto em termos de recursos computacionais, com a criação de bancos de dados remotos sediados, por exemplo, em Universidades, como apoio tecnológico, através de consultorias de profissionais qualificados no treinamento as equipes de

gestores municipais.

Este trabalho tem por objetivo o desenvolvimento de um sistema multiusuário para publicação de informações georreferenciadas para apoio à tomada de decisões que poderá ser usado, por exemplo, em apoio à administração a nível municipal e em outras organizações . Os objetivos específicos são: projeto da arquitetura do sistema, implementação de um protótipo do sistema, estudo de caso com a publicação de dados da região centro do estado.

Este sistema possui as seguintes características básicas:

- interface visualização para Web,
- construído a partir de ferramentas de software livre e código fonte aberto
- mecanismo para criação de novas consultas através da *web*
- mecanismo de administração dos dados através da *web*
- independência de plataforma no cliente (através de uso de navegador *web*)
- uso de sistema de gerenciamento de banco de dados objeto-relacional
- uso de ferramenta para geração de mapas através da *web*.

O uso da tecnologia de código fonte aberto para o desenvolvimento do sistema proposto traz as seguintes vantagens:

- 1) Por se utilizar software de código fonte aberto é possível melhorar a qualidade do SIG sendo este objeto de pesquisa e aprimoramento constante. Isto não é possível em casos de licença proprietária, pois o código fonte é secreto e propriedade industrial de determinada empresa. Normalmente, as empresas não permitiriam, por exemplo, que o seu segredo industrial fosse divulgado amplamente para universidades ou grupos de pesquisadores com objetivo de aprimoramento do software.
- 2) A utilização de software de código aberto permite reduzir consideravelmente o custo do software, facilitando o uso de SIG em instituições com poucos recursos financeiros.
- 3) A utilização de SIG baseado em tecnologia de software livre permite possíveis adaptações a equipamentos de hardware de custos mais acessíveis. Isto é devido à facilidade que certas ferramentas de software livre possuem de adaptação a sistemas operacionais que, em alguns casos, não exigirem recursos de hardware de grande desempenho. Esta situação permite o uso de SIG adaptado a equipamentos de hardware menos onerosos financeiramente, ou equipamentos considerados “obsoletos” para os aplicativos de software

proprietário.

O sistema construído poderá auxiliar atividades de projeto e aplicações de recursos baseando-se em informações regionais inserindo-as dentro do sistema e efetuando consultas personalizadas. Estas informações poderão resultar de consultas sobre o sistema a partir do cruzamento de bases de dados disponíveis gratuitamente, como, por exemplo, dados do Instituto Brasileiro de Geografia e Estatística (IBGE), de fundações governamentais de economia, de institutos governamentais de pesquisa, informações de satélites, e outras fontes de dados gratuitas.

A seguir, apresenta-se uma revisão de literatura onde são apresentados conceitos básicos relacionados com o desenvolvimento do sistema *webmap*. Posteriormente apresenta-se as seções de material e metodologia, resultados e discussão, conclusões e finalmente as referências bibliográficas.

2. REVISÃO DE LITERATURA

2.1 Software Livre

Conforme explicado em Projeto GNU (2003), o software livre é o software que tem a permissão para que qualquer pessoa o use, o copie e o distribua, com ou sem modificações, tanto gratuitamente quanto cobrando. Isto significa que o código fonte deve ser acessível, ou seja, aberto para poder ser alterado. Muitos idiomas, entre elas o espanhol, têm duas palavras diferentes para livre e gratuito. O inglês tem um palavra correspondente a grátis que se refere inequivocamente ao preço, mas não tem um adjetivo referido especificamente à liberdade, de forma que o termo "*free*" emprega-se tanto para referir-se ao preço como à liberdade. Esta ambigüidade arrasta-se a outras línguas, dada a predominância do inglês nos assuntos de informática. Devido a más traduções, é freqüente o emprego de software livre como sinônimo de software gratuito, ou mesmo tenta-se reduzir o software livre a mero software gratuito. Há muitas maneiras de fazer com que um programa seja livre, muitos detalhes podem se resolver de diferentes formas, mas nem por isso o resultado deixa de ser livre.

O *Geographic Resources Analysis Support System* (GRASS) é um SIG configurável desenvolvido pelo Exército dos Estados Unidos com o objetivo de fazer a análise espacial do território. GRASS é um SIG gratuito e de código fonte aberto, capaz de processar imagens *raster* e vetoriais e com funcionalidades gráficas disponíveis em interfaces humano-computador para diferentes sistemas operacionais de software livre (MITASOVA e NETELER, 2002). Está registrado sob a licença GPL (*Gnu Public License*) (*Free Software Foundation*, 1991).

2.2 Software Proprietário

Conforme explicado em Projeto GNU (2003), o software proprietário ou de licença comercial é aquele que não é nem livre nem semi-livre. O seu uso, modificação ou redistribuição está proibida, ou requer a permissão de uso do seu fabricante ou responsável, ou está restringido de tal forma que é praticamente impossível fazê-lo livremente. O uso deste software exige o pagamento de licenças comerciais.

2.3 Sistemas de Informações Geográficas

Segundo Wetherbe (1987), um sistema é uma coleção ou arranjo de entidades relacionadas ou conectadas de tal modo que formam uma unidade ou um todo. Estas entidades são geralmente arranjadas de modo que possam interagir para colcluir um ou mais objetivos. Os sistemas de informação podem ser entendidos como um conjunto de *software*, *hardware*, e recursos humanos que interagem para atingir objetivos ou resultados específicos.

Segundo Câmara (1996) SIG (Sistemas de Informações Geográficas) do inglês *Geographic Information Systems* (GIS) “são sistemas automatizados usados para armazenar, avaliar e manipular dados geográficos, ou seja, dados que representam objetos e fenômenos em que a localização geográfica é uma característica inerente à informação e indispensável para analisá-la”.

Segundo Strieder (2002), um SIG pode coletar e integrar dados de fontes heterogêneas, selecionar e fornecer informações conforme o interesse do usuário. Estas fontes são, por exemplo, dados cartográficos, dados de censo, cadastro urbano e rural, imagens de satélite e MNT (Modelo Numérico de Terreno). Estes dados são manipulados pelos componentes do SIG.

A composição de um SIG abrange tanto *hardware* quanto *software*.

Segundo Câmara et. al. (1996), existem três gerações de SIG, conforme mostra a tabela 1.

Tabela 1 – Gerações de SIG. (Fonte: Câmara, 1996).

	1^a geração (1980 – 1990)	2^a geração (1990 – 1997)	3^a geração (1997 - ?)
Tecnologia	CAD, cartografia	BD, imagens	Sistemas distribuídos
Uso principal	Desenho de mapas	Análise espacial	Centro de dados
Ambiente	Projetos isolados	Cliente – servidor	Multi – servidor
Sistemas	Pacotes separados	Sistema Integrado	Interoperabilidade

Segundo Bolfe (2001), SIG é um conjunto de ferramentas especializadas em adquirir, armazenar, recuperar, transformar e emitir informações espaciais. Bolfe (2001) também explica que sistemas de informações geográficas são aplicativos constituídos de cinco módulos. Cada módulo é um subsistema que permite as operações de entrada e verificação de dados, o armazenamento e gerenciamento de banco de dados, a apresentação e saída de dados, a transformação de dados e a interação com o usuário.

Câmara (1996) afirma que o termo Sistemas de Informação Geográfica (SIG) aplica-se a sistemas que realizam o tratamento computacional de dados geográficos e recuperam informações não apenas com base em suas características alfanuméricas, mas também através de sua localização espacial; oferecem ao administrador (urbanista, planejador, engenheiro) uma visão especial de seu ambiente de trabalho, onde todas as informações disponíveis sobre um determinado assunto estão ao seu alcance, interrelacionadas com a localização geográfica.

O autor ressalta que há pelo menos três grandes maneiras de utilizar um SIG:

- como ferramenta para produção de mapas;
- como suporte para análise espacial de fenômenos;
- como um banco de dados geográficos, com funções de armazenamento e recuperação de informação espacial.

2.4 Tipos de dados em Geoprocessamento

Assad (1998) lista os principais tipos de dados utilizados em SIG:

- Mapas temáticos: descrevem de forma qualitativa a distribuição espacial de uma grandeza geográfica, por exemplo a aptidão agrícola de uma região;
- Mapas cadastrais: onde cada elemento é considerado um objeto geográfico dotado de atributos e podendo estar associado a várias representações gráficas. Por exemplo, os lotes de uma cidade são elementos do espaço geográfico que possuem atributos, tais como o nome do dono, a localização, o valor venal, o valor do imposto municipal devido. Estes atributos ficam armazenados num sistema gerenciador de banco de dados.
- Redes: que são informações associadas a serviços de utilidade pública, como água, luz e telefone, e também redes de drenagem (bacias hidrográficas), rodovias;

- Imagens: podem ser obtidas por satélites, fotografias aéreas, que representam formas de captura indireta de informação espacial. São armazenadas como matrizes onde cada elemento da imagem (chamado de *pixel*) tem um valor proporcional à energia eletromagnética refletida ou emitida pela área da superfície terrestre correspondente;
- Modelos numéricos de terreno: são utilizados para representar quantitativamente uma grandeza que varia continuamente no espaço. Muitas vezes são associados à altimetria, também podem representar unidades geológicas, como teor de minerais, ou propriedades do solo ou subsolo.

2.5 Representação computacional de mapas

De acordo com Assad (1998), existem duas formas principais de representações computacionais de mapas: vetoriais e matriciais. Na forma vetorial, busca-se reproduzir o objeto geográfico o mais exatamente possível, usando-se três formas básicas: pontos, linhas e áreas ou polígonos. Na forma matricial, também chamada de *raster*, o espaço é representado como uma matriz $P(m, n)$ composta de m colunas e n linhas, onde cada célula possui um número de linha, um número de coluna e um valor correspondente ao atributo estudado, onde cada célula é acessada de forma individual através de suas coordenadas.

Um exemplo de utilização da forma vetorial em SIG foi desenvolvido pela empresa ESRI, chamado de arquivo *shapefile*, utilizado pelos produtos da empresa, como por exemplo o *ArcView*. Os dados são organizados em arquivos binários com extensão *.SHP* com a geometria, arquivos com extensão *.SHX* com índices e extensão *.DBF* com atributos. Programas podem ser criados para ler ou escrever arquivos *.SHP* (*shapefile*) usando sua especificação técnica, e bibliotecas como a *Shapelib*, a qual é um *software* que fornece suporte para lidar com arquivos no formato em nível de programação. O formato inclui ponto, linha e polígono e tipos para o caso de entidades representadas por mais de um polígono ou polígonos com vários anéis (Câmara, 2004).

2.6 Aspectos da publicação de mapas na web

Diferentes tecnologias são necessárias para desenvolvimento de SIG: dentre elas pode-se citar os SGBD, linguagens de programação, ferramentas para publicação de mapas na Internet.

Câmara (2004) aborda a importância da disseminação de dados geográficos pela Internet. Devido ao alto custo de produção dos dados geográficos, essa disseminação é de grande importância para os seus diversos usuários tais como: órgãos ambientais de planejamento e fiscalização, estudantes, pesquisadores e empresas com propósito de determinar melhores rotas para distribuição de seus produtos. Até mesmo turistas procurando se localizar em uma cidade desconhecida ou planejar uma viagem são usuários em potencial. A publicação de mapas na Internet, especificamente na *world wide web*, permite uma disseminação da informação espacial para um grande número de pessoas de forma facilitada.

Os aplicativos existentes para publicar mapas na *web* não têm as mesmas características que um SIG convencional, porém possuem um conjunto de funcionalidades para uma interação com o mapa (Miranda, 2003).

Limp (1999) chama de servidores de mapas os aplicativos que disponibilizam mapas na Internet.

O processo de publicação de mapas pela *web* exige que se tenham ferramentas de auxílio junto ao servidor do protocolo *web* para processamento de imagens dos mapas.

Duas abordagens para publicar mapas são possíveis em termos do processamento das informações. Tomando-se o conjunto formado por servidor *web*, meio de transmissão, e navegador *web*, o processamento das operações pode ser estruturada sob duas abordagens. No servidor *web* por meio de aplicativos servidores ou no navegador *web* por meio de *plugins*. Os *plugins* são programas que se acoplam ao navegador para incluir outras funcionalidades que por padrão este não possui.

Segundo Miranda (2003), quando a interação é feita pelo computador servidor, uma imagem ou o próprio mapa é gerado e enviado para o navegador, no computador cliente.

Com o uso do navegador, podem existir operações para o usuário interagir com o mapa, como a mudança de escala a partir de seleção de uma região do mapa. Neste instante, o navegador comunica-se com o servidor, que responde

à solicitação redesenhando a região correspondente à seleção da imagem no navegador.

A interação do usuário final com o mapa exige um conjunto básico de operações como: zoom, aproximação da imagem a ser visualizada, amostragem de coordenadas de determinada área.

Existem atualmente duas alternativas de aplicativos para publicação de mapas na Web: os produtos comerciais e os produtos livres ou sob a licença de software livre e código fonte aberto.

Segundo Vassoler (2001), nos últimos anos, diversas empresas passaram a apresentar soluções alternativas para se disponibilizar mapas na Internet. Miranda (2002) lista algumas soluções de licença comercial destas alternativas: *MapGuide* (Autodesk, 2005) ; *Geomedia Web Map* (Intergraph, 2005); *ArcIMS* (Esri, 2005). O mesmo autor cita também os servidores de mapas não-proprietários, disponíveis gratuitamente na Internet: *UMN Mapserver*, *Mapit* e o *ALOV Map* . Todas essas soluções baseiam-se em componentes de software dentro de uma arquitetura do tipo cliente/servidor. A figura 1 mostra uma representação genérica deste tipo de arquitetura.

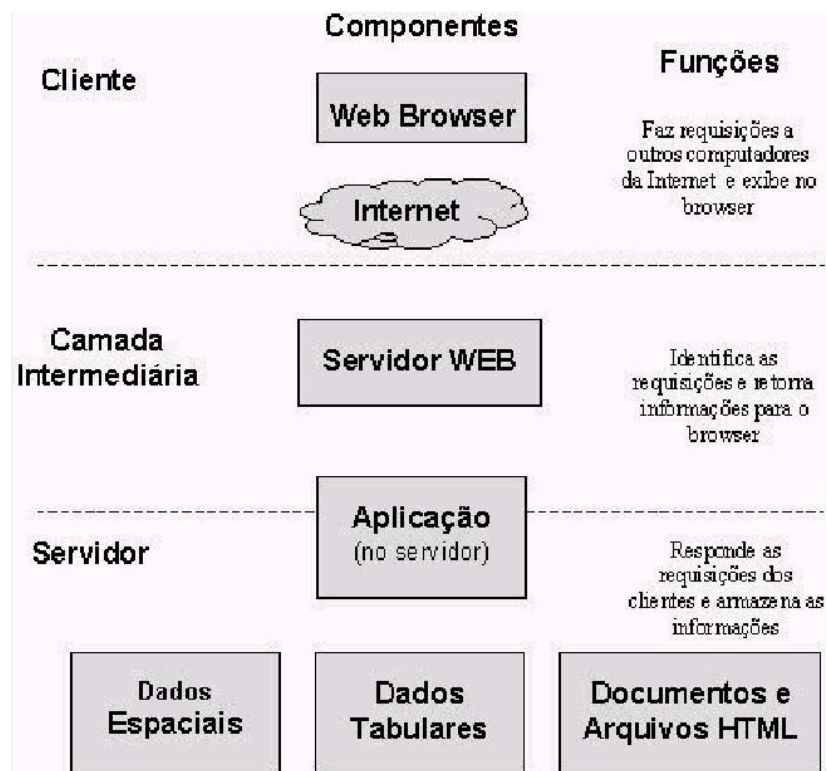


Figura 1: Arquitetura cliente/servidor para publicação de mapas na web.

(Fonte: Vassoler, 2001)

2.6.1 Ferramentas Comerciais para publicação de mapas na web

Geomedia Web Map é um exemplo de software comercial desenvolvido pela empresa Intergraph, capaz de publicar mapas na Internet usando a transferência de dados entre servidor *web* e navegador através de um formato vetorial próprio. A sua arquitetura interna, mostrada na figura 2, é composta de três camadas: de apresentação, de lógica de negócios e de dados. A camada de apresentação representa as diversas formas de visualização dos dados geográficos: navegadores *web*, computadores portáteis, chamados de *handhelds*, sistemas SIG desktop. Os dados são enviados para esta camada através da tecnologia OLE/COM da Microsoft Corporation e do padrão *Extended Markup Language* (XML). A camada de lógica de negócio é responsável pela funções geoespaciais de tratamento de dados e está desenvolvida na tecnologia OLE/COM da Microsoft, podendo interagir com componentes criados na linguagem de programação C++ e *Visual Basic* da Microsoft. A camada de dados é responsável pela conexão com fontes de dados de vários formatos, por exemplo, oriundas de SGBD relacionais como Oracle e Microsoft Sql Server além de dados oriundos de SIG de outros fabricantes como ESRI-Shapefiles (INTERGRAPH, 2004).

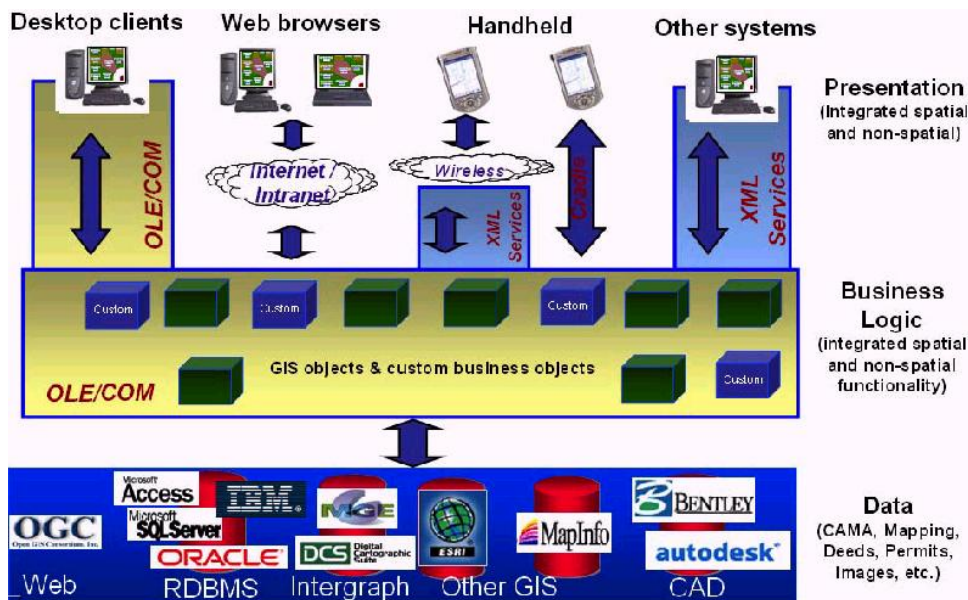


Figura 2: Arquitetura do *Geomedia Web Map* da Intergraph.

(Fonte: INTERGRAPH, 2004).

O ARCIMS é um outro exemplo de software comercial, desenvolvido pela empresa ESRI, que tem por finalidade a publicação de informações geográficas na Internet. É composto por uma arquitetura em camadas mostrado na figura 3 ESRI(2005).

Uma visão geral desta arquitetura foi apresentada em ESRI (2002). A camada de apresentação (*presentation tier*) é formada por softwares clientes para acessar, visualizar e analisar dados geográficos. Estes clientes podem ser personalizados, usando-se uma ferramenta chamada ArcIMS Designer. A camada de lógica de negócios (*business logic tier*) gerencia os pedidos e administra o *site* servidor do ArcIMS. A camada de dados (*data tier*) é formada pelas fontes de dados usadas com o ArcIMS. Diferentes tipos de formatos de dados podem ser acessados nesta camada: arquivos vetoriais, arquivos *raster*, conexões com bancos de dados relacionais.

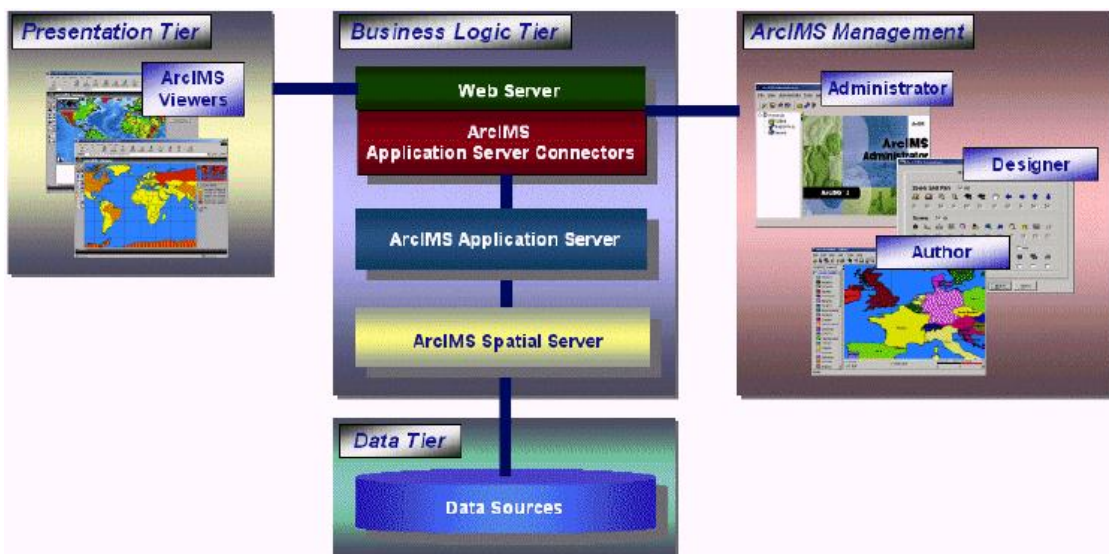


Figura 3. Arquitetura do ARCIMS. (Fonte: ESRI, 2002).

Na figura 3, a camada de apresentação (*presentation tier*) representa os programas para visualização dos dados geográficos e mapas gerados, tais como navegadores *web*.

Como visto em ESRI (2002), a camada de lógica de negócios (*business logic tier*) é composta pelos seguintes itens:

- servidor *web* (*web server*) para controlar o envio e recebimento de requisições do navegador *web*, usando o protocolo *Hipertext Transfer Protocol* (HTTP), ou protocolo de transferência de hipertexto;
- conectores de servidor de aplicação (*Application Server Connectors*) é responsável pela comunicação entre o servidor *web* e o servidor de aplicações (*application server*).
- servidor de aplicações (*application server*) é um aplicativo que fica em execução constante no sistema operacional e faz o controle de execução de pedidos de processamento de atividades solicitadas pelo módulo de visualização, redirecionando as solicitações para os devidos módulos do servidor espacial.
- servidor espacial (*spatial server*) é o núcleo das operações geográficas. Este item proporciona as capacidades funcionais para acessar mapas e dados dentro de um formato apropriado antes de enviá-los para a visualização. Dentre as funções deste item tem-se: tratamento de imagens *raster* e vetoriais; servidor de consultas; cálculo de coordenadas para localização de pontos nos mapas; servidor de metadados ou de descritores de dados.
- camada de dados (*Data Tier*) é o módulo que faz interface com fontes de dados, como por exemplo conexões com sistemas de gerenciamento de banco de dados (exemplo: Oracle, MS Sql Server), imagens *raster*, imagens vetoriais (ESRI Shapefiles).

Conforme ESRI (2002), vê-se na figura 3 que o módulo de gerenciamento do ArcIMS (ArcIMS Management) é formado por três aplicativos que o desenvolvedor usa para criação dos mapas para web ou webmap. Estes aplicativos são: *Author*, para criação de projetos de mapas e arquivos no formato shape, onde serão definidos os planos de informação dos mapas; *Designer*, para criar os leiautes de apresentação das páginas web para visualização dos mapas nos navegadores; *Administrator*, que controla as funções de administração dos serviços de mapas no servidor espacial (*spatial server*), como, por exemplo, iniciar e parar os serviços do servidor web, definir o endereço URL do mapa a ser publicado e manipulação de arquivos de configuração.

2.6.2 Ferramentas de software livre para publicação de mapas na web

Um exemplo de ferramenta de software livre para publicação de mapas na web é o UMN Mapserver.

UMN Mapserver foi estruturado em conformidade com o "OpenGIS Consortium Web Mapping Testbed (WMT)" e foi desenvolvido originalmente pela Universidade de Minnesota (UMN) nos Estados Unidos, em cooperação com a NASA e o Departamento dos Recursos Naturais de Minnesota (MNDNR). Ajustes adicionais foram feitos pelo MNDNR e o Centro de Informação e Administração do solo de Minnesota (LMIC). O desenvolvimento atual é financiado pelo projeto TerraSIP, patrocinado pela NASA, pela UMN e pelo consórcio de interesses da administração do solo (Mapserver, 2005).

MapServer é um ambiente de desenvolvimento, com código fonte aberto, para construção de aplicações geoespaciais voltadas para Internet. Tem a habilidade de gerar mapas e exibi-los em um navegador *web* através da Internet. Tem recursos para publicar dados no formato vetorial (exemplo: arquivos no formato ESRI), no formato Raster (exemplo: arquivos nos formatos PNG, GIF, TIFF), e condições de personalizar a exportação de dados através de um sistema de configuração chamado de *Mapscript* (Mapserver, 2005).

Herzman (2002) mostra a estrutura básica do Mapserver composta por:

- um arquivo executável que trabalha no padrão cgi-bin para geração de páginas web dinâmicas;
- um arquivo de configuração de mapa, chamado de *mapfile*, que armazena o esquema e as fontes de dados da aplicação SIG;
- um arquivo *template*, ou molde, no html para a modelar a apresentação da interface da aplicação a ser mostrada no navegador web;
- um arquivo *template* de consultas para modelar os resultados das consultas feitas pelo usuário.

O Mapserver foi desenvolvido usando-se a linguagem de programação "C". Muitos sistemas operacionais de código aberto, por exemplo o Linux e o NetBSD, utilizam esta linguagem para a instalação e a adaptação de programas. Portanto, isto possibilita que o Mapserver seja instalado em vários outros sistemas operacionais.

Segundo Mapserver (2003) uma imagem qualquer pode ser apresentada com

recursos de georreferenciamento em um navegador *web* usando-se este servidor de mapas.

Com o Mapserver é possível criar mapas com mais de uma camada, aplicar operações em uma das camadas para destacar atributos específicos, e ainda rotular regiões do mapa.

O Mapserver pode, tanto tratar camadas *raster*, quanto vetoriais. Pode importar e exportar vários formatos raster. Atualmente suporta imagens nos padrões de cores RGB e multiespectrais (multi-camadas), e possibilita salvar as imagens em formatos padronizados, por exemplo, arquivos do tipo pdf e tiff.

Mapas dinâmicos podem ser gerados caso se queira aplicar, por exemplo, um *zoom* em uma determinada região do mapa. Para isto, existe a funcionalidade de se gerar mapas clicando-se em algum ponto do mapa mostrado.

Ao criar aplicações *web* para processar mapas, dentro da arquitetura de SIG, é comum a intenção de se gerar mapas que podem ser modificados pelo usuário da aplicação de forma interativa. Ou seja, o usuário pode modificar o conteúdo do mapa ou das informações contidas nele. Para proporcionar esta interatividade, o software Mapserver utiliza moldes ou *templates* em linguagem HTML. Estes moldes são arquivos em formato HTML com diretivas de comandos de programação específicos, utilizados para formatar a interface no navegador com a qual o usuário irá interagir.

São exemplos de alguns recursos dos *templates* HTML: gerar *zoom* e *pan* de um mapa, inserir legendas, controlar camadas (*layers*) do mapa inserindo-as ou retirando-as de uma visualização; adicionar uma barra de escala (*scalebar*), incluir um mapa de referência.

Um dos vários exemplos de SIG que utilizam o Mapserver é “Iowa Tornado Database” descrito em Herzman (2002).

Outro exemplo de ferramenta de software livre para publicação de mapas na web, conforme Miranda (2002), é o *Mapit*, um programa de tecnologia CGI (Common Gateway Interface) do tipo script, escrito em linguagem *Python*, para uso com dados espaciais matriciais. Pode ser usado nos sistemas operacionais Linux e Microsoft Windows. A arquitetura do *Mapit* é do tipo cliente/servidor, conforme visto na figura 1. Como o *Mapit* foi escrito em linguagem *Python*, é necessário ter no servidor web o módulo de software para esta linguagem (Mapit, 2002).

Outro exemplo de ferramenta semelhante ao Mapit, citada por Miranda (2002) é o ALOV Map. Esta ferramenta consiste em uma aplicação em linguagem Java que permite publicar mapas vetoriais e matriciais na Internet. Ele pode ser usado de

duas maneiras: somente cliente ou cliente/servidor. A versão cliente é um *applet* (módulo Java processado no navegador) para visualização de dados no formato SHAPE da ESRI. Como é característica de todo *applet*, cada vez que ele for inicializado, o servidor *web* irá transferir todos os arquivos de mapas para o computador cliente. Isto requer cuidado com o tamanho dos arquivos, pois o tempo de transmissão pode se tornar longo. A versão cliente/servidor do ALOV Map, um *servlet* (módulo Java processado no servidor *web*), é mais flexível e permite transferir dados de forma incremental. O servidor de mapas envia dados vetoriais para o cliente de forma mais eficiente, através da técnica específica, reduzindo o tráfego de rede. Os atributos espaciais são armazenados em um sistema de gerenciamento de banco de dados relacional de acordo com padrões SFS da OpenGIS (ALOV, 2005).

2.7 Armazenamento de Dados em SIG

Os Sistemas de Gerenciamento de Banco de Dados (SGBD) desempenham atualmente um papel fundamental na arquitetura de SIG, já que estão dando o suporte às atividades de armazenamento e recuperação dos dados do SIG. Câmara (1996) indica que estas atividades fazem parte da estrutura geral de um SIG, formada pelos seguintes componentes: interface com o usuário; entrada e integração de dados; funções de consulta e análise espacial; visualização e plotagem; armazenamento e recuperação de dados (organizados sob a forma de um banco de dados geográficos).

Um SGBD consiste em um conjunto de aplicações dedicadas à gestão de dados com uma ou várias interfaces de utilização, nos quais o utilizador pode executar um conjunto de funções de carácter genérico ou específico. O mercado de sistemas de gerenciamento de bancos de dados tem aumentado as funcionalidades destes sistemas. Por exemplo, muitas das funções necessárias para o processamento dos dados dizem respeito à gestão realizada em SGBD, contribuindo para uma considerável proteção aos dados e facilitando a disponibilidade de informação. Na gestão de dados em SGBD, são ocultados determinados detalhes sobre a forma de armazenamento e manutenção desses dados (Silberschatz et al., 1997).

O autor Assad (1998) explica que um SGBD é um software de carácter geral para manipulação eficiente de grandes coleções de informações estruturadas e

armazenadas de uma forma consistente e integrada. Ressalta-se ainda que os principais objetivos de um SGBD são:

- tornar disponíveis dados integrados para uma grande variedade de usuários através de interfaces amigáveis;
- garantir a privacidade dos dados através de segurança dentro do sistema;
- permitir o compartilhamento de dados de forma organizada, atuando como mediador entre as aplicações e o banco de dados, garantindo assim o controle e a redução do nível de redundância e administrando acessos concorrentes;
- controlar a administração dos dados com garantia de manutenção de padrões de qualidade e integridade; e
- possibilitar a independência dos dados no sentido de poupar ao usuário detalhes físicos de organização e armazenamento.

Dentre as tecnologias de SGBD mais usadas em SIG destaca-se a dos SGBD relacionais, baseados no modelo entidade-relacionamento.

Segundo Silberschatz et al (1997), esse modelo busca representar os dados em um conjunto de objetos básicos chamados entidades e nos relacionamentos entre esses objetos.

Segundo Heuser (2000), uma entidade é um conjunto de objetos, da realidade modelada, sobre os quais deseja-se manter informações no banco de dados, e relacionamento é um conjunto de associações entre entidades.

Por exemplo, um banco de dados relacional poderia armazenar um conjunto de dados relativos aos vendedores e clientes de uma loja. As entidades são cliente e vendedor. Os atributos dessas entidades poderiam ser nome, telefone, endereço do cliente, e nome e comissão do vendedor. Um relacionamento pode ser visto como a ocorrência de vendas firmadas entre estas duas entidades. A forma com que estes dados são representados e armazenados é o que caracteriza os SGBD relacionais.

2.8 SGBD e Licenças de uso

Os Sistemas de Gerenciamento de Banco de Dados desenvolvidos atualmente estão em duas possíveis situações: comerciais ou sob licença proprietária e não-comerciais ou sob a licença de software livre.

São exemplos de SGBD de licença proprietária: Oracle (Oracle, 2005), IBM DB2, (IBM 2005), MS SqlServer (Microsoft, 2005). São exemplos de SGBD sob licença de software livre: Postgresql (Postgresql,2005) e MySQL (Silva, 2000).

2.9 SGBD comerciais

O SGBD Oracle é um SGBD cliente/servidor objeto-relacional (ORDBMS) de alto desempenho produzido pela Oracle Corporation (ORACLE, 2005). Foi projetado para ser executado em sistemas operacionais UNIX, Linux e Microsoft Windows NT. Sua tecnologia integra recursos capazes de gerenciar grande volume de dados em ambientes multiusuário. Isso garante o acesso simultâneo dos usuários ao banco de dados, sem comprometer o desempenho do sistema. O Oracle fornece um potente mecanismo de integridade referencial, possuindo também suporte de gatilhos e procedimentos armazenados. Para prover facilidade na administração dos recursos, as versões mais recentes do Oracle dispõem de uma ferramenta conhecida como *Oracle Enterprise Manager*. Essa ferramenta integra vários utilitários de administração e manipulação de elementos de banco de dados. Outro mecanismo importante a ser destacado é o *Data Manager*, que oferece recursos que permitem importar e exportar objetos de um banco de dados para outros, facilitando as tarefas de migração e atualização (Oracle, 2005).

Outro exemplo de SGBD comercial é o SQL Server, desenvolvido pela Microsoft Corporation. É um SGBD cliente/servidor relacional, projetado para oferecer alto desempenho e suporte a processamento de alto volume de dados. Sua plataforma é baseada em sistemas operacionais da Microsoft nas versões Windows NT Server ou Workstation, Windows 95/98/ME e em plataformas Windows 2000. Tem como linguagem SQL nativa a *Transact-SQL*, que incorpora vários recursos como otimização de consultas, construção de programação e procedimentos armazenados. Na sua versão 7, o SQL Server possui um ambiente integrado de desenvolvimento, conhecido como *Enterprise Manager*, onde permite administrar todos os recursos de qualquer servidor de banco de dados SQL acessível. Trata-se de uma interface que apresenta uma estrutura de árvore, onde é possível visualizar todos os detalhes de um servidor de banco de dados, incluindo, além dos bancos de dados, todas as ferramentas de gerência necessárias. Outra ferramenta importante é o *Query Analyzer*, que permite executar instruções SQL e visualizar os resultados através de uma interface simples (Microsoft, 2005).

IBM-DB2 é um SGBD que segue também o modelo relacional e foi desenvolvido pela empresa IBM. Está disponível para sistemas operacionais Unix, Linux, IBM-AIX, Solaris, MS Windows 2000, Windows NT. É dotado de ferramentas adicionais para administração de bancos de dados, tais como o aplicativo

denominado Centro de Administração Gráfica (IBM, 2005), que facilita a criação e manutenção de objetos tais como tabelas, bancos de dados, relacionamentos.

2.10 SGBD de licença de software livre

2.10.1 Postgresql

Segundo Biazus (2003), o Postgresql é um SGBD objeto-relacional de código aberto, ou de licença de software livre, com mais de 15 anos de desenvolvimento. É robusto e confiável, além de ser extremamente flexível e rico em recursos. Ele é considerado sob o modelo objeto-relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos, como herança e tipos personalizados.

Alguns recursos presentes na versão mais recente são: Sub-consultas; controle de concorrência multi-versão; Integridade Referencial; Funções armazenadas (Stored Procedures), que podem ser escritas em várias linguagens de programação (PL/PgSQL, Perl, Python, Ruby, e outras), gatilhos, tipos definidos pelo usuário, esquemas de banco de dados, conexões seguras (BIAZUS, 2003).

Um importante fator para o uso Postgresql em projetos de sistemas de informações geográficas é a extensão Postgis, citada e descrita por Blasby (2003), a qual foi integrada ao SGBD para processar junto à consultas do padrão SQL os recursos para tratar dados georreferenciados. O Postgis adiciona o suporte para objetos geográficos ao SGBD Postgresql.

Em Leyh (2003) temos um exemplo de utilização do Postgresql e sua extensão Postgis no desenvolvimento de SIG para a área da saúde pública.

2.10.2 Mysql

Mysql é um SGBD de código fonte aberto que tem se destacado atualmente como solução para gerenciamento de dados em uma série de projetos das mais variadas áreas. Em Perez (2003) tem-se um exemplo de sua aplicação em um SIG para a área do turismo. Atualmente o Mysql não suporta operações geográficas em suas consultas. Pode-se compor a arquitetura de SIG usando ferramentas para processamento de dados geográficos e o Mysql para construir sistemas de qualidade considerável.

Segundo Silva (2000), o Mysql é um servidor de banco de dados SQL, que opera em múltiplas plataformas, abrangendo todas as variedades Linux, AIX, FreeBSD, Irix e até Windows 95/98/NT. Para plataformas Unix e OS/2, o Mysql é gratuito. Entre suas características principais, destacam-se:

- bom desempenho e estabilidade;
- reconhece vários tipos de dados, por exemplo os tipos: caracteres variáveis, texto, arquivos binários, data e hora, conjuntos e enumerações;
- possui interfaces com as linguagens de programação C, C++, Java, Perl, PHP, Python e TCL;
- interface para os padrões de acesso a banco de dados ODBC e JDBC;
- sistema de privilégio de acesso robusto e seguro, com controle de acesso por computador, por usuário, por objetos dentro do banco de dados, como tabelas e colunas.

Como limitações, pode-se citar a ausência de capacidade de replicação, controle de transações, integridade referencial, cursores no servidor e procedimentos armazenados ou *stored procedures*.

2.11 Extensões espaciais para SGBD

As necessidades inerentes ao tratamento de dados geográficos conduziram ao surgimento de módulos ou *extensões*, um incremento de funcionalidades tradicionais em SGBD. Como principal característica destas *extensões*, destaca-se a possibilidade de armazenar informação geográfica em estrutura de tabelas: os arquivos geográficos são convertidos em unidades cartesianas (pares de coordenadas X e Y) e armazenados em tabelas dentro do banco de dados (Pereira, 2002).

Alguns SGBD atuais incorporam capacidades adicionais, chamadas de extensões espaciais, para tratamento de dados geográficos e assim são denominados SGBD objeto-relacionais (Miranda, 2003).

Ferreira (2003) cita que atualmente os principais SGBD com estas extensões são: Oracle Spatial Extender, IBM DB2 Spatial Extender dentro da categoria de software comercial.

Como alternativas de software livre tem-se o Postgresql, com sua extensão espacial Postgis Ramsey (2005), e o Mysql com uma implementação recente de funções espaciais Silva (2004), (Mysql, 2005).

Em Marisco (2004) tem-se um exemplo de utilização do Postgresql e do Postgis para a construção de mapas interativos para a web. Esse autor afirma: “A utilização do Postgresql e sua extensão espacial PostGIS fornece todo o suporte ao armazenamento dos dados gráficos e não gráficos utilizados ao longo do desenvolvimento deste trabalho.”

2.12 O Padrão OGC para interoperabilidade em SIG

O Consórcio OPEN GIS ou OpenGIS Consortium (OGC) é uma organização sem fins lucrativos com o objetivo de tornar os sistemas de geoprocessamento abertos (OGC, 2003). O OGC almeja a integração completa de dados geográficos e recursos de geoprocessamento através do uso de sistemas de informação geográfica interoperáveis.

De acordo com Thomé (1998) o comitê técnico do consórcio estava trabalhando para estabelecer uma especificação que definisse uma arquitetura de software abrangente para sistemas abertos de geoprocessamento. Os sistemas construídos baseados nesta especificação seriam capazes de praticar a interoperação entre aplicações em uma rede local, e também de navegar sobre um ambiente heterogêneo e distribuído, como a Internet, acessando dados geográficos heterogêneos e recursos de geoprocessamento. Para a criação dessa especificação, o consórcio OPEN GIS utiliza-se de um processo de consenso entre seus membros. Tal consenso é traduzido para uma especificação abstrata e uma especificação de implementação para cada um dos componentes de software relacionados aos Sistemas de Informação Geográfica.

Algumas das especificações definidas pelo OGC são: *Geography Markup Language* (GML), *Simple Features-SQL*(SFS), *Web Coverage Service*(WCS), *Web Feature Service* (WFS), *Web Map Context Documents*(WMC), *Web Map Service*(WMS) (OGC,2005).

A GML é uma codificação no formato XML (*Extended Markup Language*) para o transporte e armazenamento de informações geográficas, incluindo também geometrias e propriedades de características geográficas.

A SFS é uma especificação para uma interface de programação de aplicação

ou application programming interface (API) que proporciona a publicação, armazenamento, acesso e operações simples para formatos como pontos, linhas, polígonos, multipontos, extendendo a linguagem SQL, que é um padrão para sistemas de gerenciamento de bancos de dados relacionais.

WFS propõe a definição de uma interface que descreve operações de manipulação de dados no OpenGis Simple Features tal que aplicativos servidores e clientes possam se “comunicar” em um mesmo nível.

WCS é uma especificação que estende a interface para o Web Map Server ou servidor de mapas para *web*, permitindo acesso às coberturas geoespaciais que representam valores de propriedades de localidades geográficas.

WMS proporciona três protocolos de operação (*GetCapabilities*, *GetMap* e *GetFeatureInfo*) para suportar a criação e exibição de visões de mapas que podem vir de fontes remotas e heterogêneas.

A especificação denominada Web Map Context Documents (WMC) define como criar, armazenar e usar informações de estado de um cliente de aplicação do tipo WMS (Web Map Service).

2.13 Sistemas Operacionais

O uso de software aplicativo exige um pré-requisito fundamental que é o software básico. O sistema operacional do computador é chamado de software básico, e se caracteriza por ser o sistema responsável pelo gerenciamento dos dispositivos de hardware, tais como discos magnéticos, periféricos, memória RAM, e atividades como o processamento de tarefas, manipulação de interface homem-computador, o gerenciamento de arquivos, o gerenciamento de usuários, entre outros. Os sistemas operacionais de software livre foram usados neste trabalho para servir de base para estudo e o desenvolvimento futuro de ferramentas para SIG.

Projetos de SIG têm usado sistemas operacionais de código aberto e licença de software livre, como base para arquiteturas dos SIG distribuídas e interface *web*. São exemplos de sistemas operacionais utilizados em SIG o Linux e o FreeBSD. Usando Linux temos, em Tydac (2003), um SIG que gera mapas do território suíço. Outro caso de uso do Linux é descrito em Vieira (2001) e Anatec (2000), um sistema de mapas da cidade de Porto Alegre-RS.

O sistema operacional Linux é usado como base para instalação de servidor

web, a exemplo o servidor Apache, descrito em Apache (2003), podendo-se utilizar módulos cliente dos sistemas através de um navegador *web* em qualquer outro sistema operacional, por exemplo Microsoft Windows e Machintosh.

O sistema operacional FreeBSD é outro exemplo que serve de base para o funcionamento de ferramentas para arquiteturas de SIG. Em Minnesota (2001) temos exemplos de projetos SIG utilizando o sistema FreeBSD.

2.14 Sistema de moldes Chameleon para UMN Mapserver

A empresa canadense DM Solutions desenvolveu um projeto de código aberto para apoio ao uso do UMN Mapserver, na forma de um sistema de moldes apoiado em tecnologia PHP e XML denominado de Chameleon (DM SOLUTIONS, 2005a). Este sistema de moldes é um conjunto de códigos em linguagem de programação capaz de ser reutilizado para vários fins. O Chameleon uma tecnologia completa para criação de aplicativos de mapas para *web* ou *webmaps*. A estrutura do Chameleon está composta por 3 itens: *Chameleon Web Mapping Components* (CWC), Chameleon Server, UMN MapServer.

Os CWC são componentes chamados de *widgets* que manipulam um subconjunto de funcionalidades dentro de um *webmap*. Eles são definidos através de comandos CWC dentro de páginas HTML. Por exemplo, um mapa ou uma ferramenta de zoom podem ser definidas dentro de uma página *web* para identificar uma localização específica na página, seu tamanho, e outros parâmetros necessários para este componente. Um *webmap* pode ser criado agrupando-se em uma página HTML um conjunto de componentes CWC, como, um componente para a tela de apresentação de mapa, um para um quadro de legenda, um para uma barra de escala.

O Chameleon Server recebe um arquivo HTML e processa cada um dos componentes CWC individuais. Ele gerencia a aplicação *webmap* e se comunica com o UMN Mapserver para gerar o as informações de conteúdo do mapa consultado.

O UMN Mapserver é o mecanismo que gera as imagens do mapa, gerencia os dados dos planos de informação e manipula todo o processamento geográfico. Estas três camadas da arquitetura permitem aos usuários da tecnologia Chameleon trabalhar com um nível mais simplificado de complexidade. Estes usuários dispõe de uma coleção de CWC para desenvolver e personalizar, de forma facilitada,

aplicações de mapas para web.

O Chameleon suporta os padrões do OpenGis Consortium (OGC) para interoperabilidade de dados geográficos de diversas fontes.

A figura 4 mostra esta arquitetura de 3 camadas.

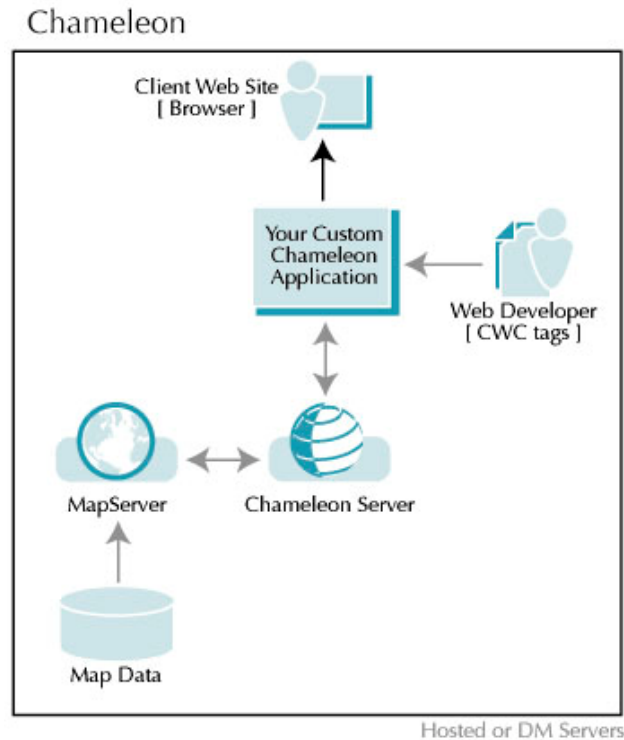


Figura 4. Arquitetura da tecnologia Chameleon.
(Fonte: DM Solutions, 2005a)

Conforme DM SOLUTIONS (2005b) são abordados alguns conceitos básicos do Chameleon e que são citados a seguir:

- Web Mapping Service (WMS) é um padrão do consórcio OGC que define um serviço capaz de proporcionar acesso a informação geográfica através da Internet usando um protocolo padrão. WMS suporta uma variedade de solicitações para acesso a imagens e metadados.

- Initialization Script é um arquivo de parâmetros que cria uma instância do Chameleon e que contém comandos CWC. Este arquivo é passado para um Servidor Chameleon o qual o comunica-se com a aplicação *webmap* atual.

- Mapfile é um arquivo básico que armazena informações sobre o mapa que está sendo mostrado com a aplicação Chameleon. Este arquivo é um mecanismo de configuração do servidor de mapas UMN Mapserver.

- Contexto é uma descrição em XML de dados geográficos. Uma instância do

Chameleon Service usa um Contexto para determinar que imagens solicitar de qual servidor WMS para mostrar um mapa para o usuário.

A *widget* é o elemento básico de uma aplicação Chameleon. As *widgets* são representadas por diretivas em formato XML em um template Chameleon. Cada *widget* representa uma parte específica da funcionalidade que vai ser apresentada na tela do *webmap*. As *Widgets* são representadas em um arquivo molde HTML. Um exemplo de *widget* pode ser:

```
<CWC2 TYPE="nomedaWidget">
  <TEST value="1" />
  <TEST value="2"/>
</CWC2>
```

Para a criação de aplicações Chameleon deve-se criar um arquivo de molde ou template como mostrado abaixo:

```
<html>
<head>
  <title>Widgets</title>
  <link href="cwc2.css" rel="stylesheet" type="text/css">
</head>
<body onload="CWC2OnLoadFunction()">
  <form></form>
</body>
</html>
```

Uma *widget* básica para qualquer aplicação é a que apresenta a imagem do mapa na tela, chamada de *MapDHTML*, para a qual se define tipo de apresentação, largura e altura.

Alguns exemplos de *widgets* disponíveis no Chameleon para personalização de webmaps são: *KeyMap*: mostra um mapa de referência na tela; *LegendTemplate*: manipula um molde para apresentação de uma legenda do mapa; *Mapsize*: permite ao usuário do navegador *web* alterar o tamanho de amostragem do mapa na tela; *Scalebar*: mostra a imagem de uma barra de escala na tela.

2.15 Linguagem de programação PHP

A linguagem PHP é uma tecnologia alternativa para construção de páginas *web* dinâmicas. O PHP sucede de um produto mais antigo, chamado PHP/FI. PHP/FI foi criado por Rasmus Lerdorf em 1995, inicialmente como simples scripts em linguagem *Perl* como estatísticas de acesso para seu currículo online. Ele nomeou esta série de script de '*Personal Home Page Tools*'. Como mais funcionalidades foram requeridas, Rasmus escreveu uma implementação em linguagem C muito maior, que era capaz de comunicar-se com base de dados, e possibilitava à usuários desenvolver simples aplicativos dinâmicos para Web. Rasmus resolveu tornar disponível o código fonte do PHP/FI para que todos pudessem ver, e também utilizá-lo, bem como corrigir erros e melhorar o código (PHP,2005).

PHP/FI, que significa Personal Home Page / Forms Interpreter, incluía algumas funcionalidades básicas do PHP que estão disponíveis atualmente. Ele usava definição de variáveis no estilo da linguagem *Perl*, interpretação automática de variáveis vindas de formulário e sintaxe embutida no arquivo HTML. A sua própria sintaxe era similar a do *Perl*, porém muito mais limitada, simples, e um pouco inconsistente (PHP,2005).

Em 1997, PHP/FI 2.0, a segunda versão da implementação em linguagem C, obteve milhares de usuários ao redor do mundo (estimado), com aproximadamente 50,000 domínios reportando que tinha PHP/FI 2.0 instalado, representando aproximadamente 1% dos domínios da Internet da época. Enquanto isto havia milhares de pessoas contribuindo com pequenos códigos para o projeto (PHP, 2005).

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes Unix (incluindo HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS. O PHP também é suportado pela maioria dos servidores *web* atuais, incluindo Apache e Microsoft Internet Information Server (PHP, 2005).

PHP é uma linguagem de programação na forma de script que fica acoplada a um servidor *web*. Foi desenvolvida sob licença de software livre e código fonte aberto.

Um programa criado em PHP permite gerar código em linguagem HTML de forma dinâmica que será enviada para um navegador web. Desta forma, o código

escrito em PHP permite controlar o que aparece na janela do navegador, tais como formulários, texto, imagens. PHP permite também a conexão com sistemas de gerenciamento de banco de dados. Devido à sua popularidade tornou-se uma linguagem com material didático disponível gratuitamente, além de uma grande variedade de trechos de código fonte para reutilização são encontrados na Internet atualmente.

A linguagem PHP é uma importante ferramenta para construção de SIG para *web*, pois apresenta a facilidade de gerar comunicação com o usuário através de um navegador *web*. Também apresenta recursos para acesso aos SGBD. Desta forma é muito utilizado atualmente em projetos de SIG. Um caso de uso desta linguagem é mostrado em Perez et al (2003) para o desenvolvimento de um SIG para a área de turismo.

3. MATERIAL E METODOLOGIA

3.1 Material

Neste trabalho foram utilizados como materiais os recursos computacionais constituídos por *hardware* e *software*.

3.1.1 Hardware

Como hardware para o sistema foi utilizado um computador que desempenhou o papel de servidor do sistema. Este computador é composto por um processador do fabricante Intel, modelo Pentium II, com memória RAM de 128 Megabytes e disco rígido de 40 Gigabytes, drive de CDROM, teclado, monitor de vídeo e mouse.

3.1.2 Software

Para o desenvolvimento do sistema proposto foram utilizadas as seguintes ferramentas de software:

- sistema operacional Linux, de distribuição Debian, versão *Testing*;
- servidor web Apache versão 1.3;
- linguagem de programação PHP versão 4.3.10-3, para Linux;
- Navegadores web: Mozilla Firefox versão 1.0 para MS Windows e Konqueror para Linux;
- Sistema de Gerenciamento de Banco de Dados Postgresql versão 7.4.6 para Linux;
- Servidor de Mapas UMN Mapserver versão 4.01 para Linux;
- Sistema de moldes Chameleon para UMN Mapserver versão 2.0.

3.2 Metodologia

Através do estudo de ferramentas para publicação de mapas na web baseadas em software livre, foi proposto um modelo de arquitetura para o sistema. Posteriormente foi feita a implementação de um protótipo do sistema, que está

dotado de um conjunto básico de funções que, em trabalhos futuros, terá possibilidade de ser expandido. Uma aplicação exemplo foi feita utilizando-se o sistema desenvolvido com dados georreferenciados do município de Santa Maria-RS.

3.2.1 Arquitetura do Sistema

O sistema proposto está composto pelas seguintes ferramentas: servidor web Apache, linguagem PHP, servidor de mapas UMN Mapserver, sistema de moldes Chameleon, SGBD Postgresql.

O servidor *web* e a linguagem PHP fazem a comunicação com o navegador *web* do usuário e são responsáveis pela transferência de dados através da Internet. O UMN Mapserver e o sistema de moldes Chameleon encarregam-se da geração das imagens em formato de mapas, bem como um conjunto de recursos de georreferenciamento dentro do *webmap*. O SGBD Postgresql armazena em tabelas relacionais os dados relativos aos planos de informação, projetos de *webmap*, e configurações do sistema. Na figura 5 apresenta-se uma ilustração destas ferramentas em forma esquemática para compreensão da relação de comunicação entre elas.

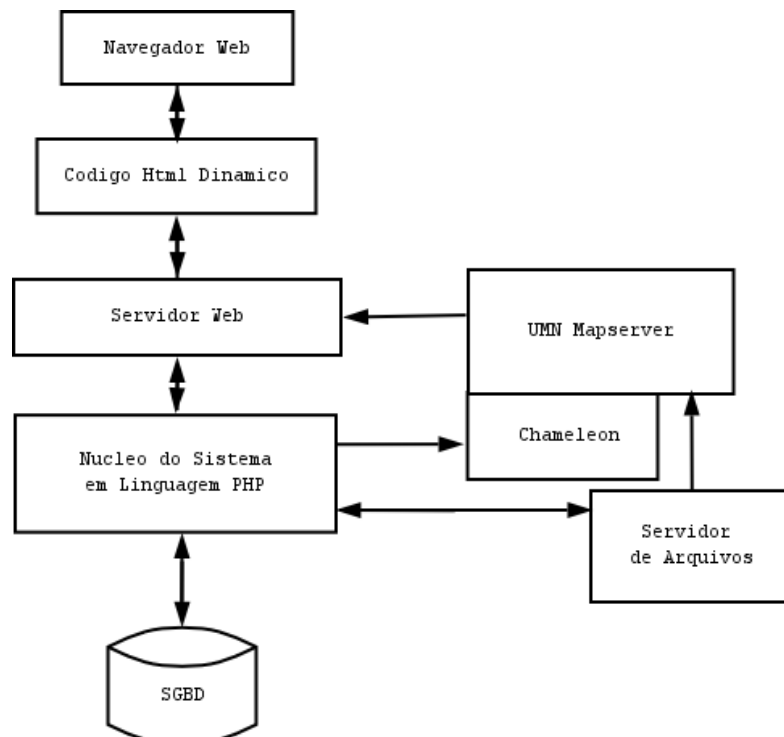


Figura 5: Arquitetura geral do sistema proposto.

3.2.2 Caracterização dos usuários do sistema.

Três papéis são desempenhados pelos usuários do sistema: usuários de consulta, projetistas de mapas, administradores do sistema proposto. A seguir são descritas as atribuições de cada um destes.

O administrador do sistema é um usuário que tem um perfil técnico, tem conhecimento de instalação e configuração das ferramentas que compõe o sistema tais como servidor de banco de dados, servidor *web*, comandos do sistema operacional. Este usuário tem acesso total e irrestrito ao sistema para que possa cuidar da instalação e funcionamento correto de todo o sistema. Caso um projeto esteja mal configurado ele pode através do seu acesso corrigir o problema.

O projetista de mapas é um usuário com conhecimentos de cartografia e de tratamento de imagens. Ele é quem vai dar um direcionamento para o uso do sistema proposto, pois ele que irá “povoar” com dados o sistema. O enfoque destes dados poderá ser, por exemplo, na área ambiental, urbana, de turismo, de trânsito, ou outro, dependendo da necessidade deste usuário em publicar informações georreferenciadas. O projetista de mapas prepara previamente seus dados através de um programa de geoprocessamento, por exemplo SPRING (INPE, 2005), ou GRASS (MITASOVA e NETELER, 2002). Nestes softwares ele vai fazer a criação dos planos de informação que irão compor o mapa. Após esta etapa, faz-se acesso ao sistema para criar o projeto do *webmap* e para inserir os planos de informações, com seus respectivos arquivos de imagem no sistema. O acesso deste usuário é feito por meio de identificação por senhas, e , desta forma, está restrito a manipular informações referentes aos parâmetros de projeto de mapas e seus planos de informação, não tendo acesso a outras áreas do sistema. Também não poderá acessar outros projetos de propriedade de outros projetistas de mapas. Cada projetista publica seus dados independente da interferência dos demais. Estas restrições de acesso permitem que os dados a serem publicados tenham sua qualidade e veracidade garantidas.

Os usuários de consulta não têm autorização para acessar os parâmetros de definição de projetos e de planos de informação do sistema. A eles compete a tarefa de consulta do mapa publicado através de um programa cliente através do uso de navegador web apontado para o endereço HTTP do projeto específico. Estes usuários vão usufruir das informações georreferenciadas publicadas através de cada projeto disponível no sistema. Por exemplo, é possível utilizar o sistema para fazer

consultas de coordenadas, mudar escalas de visualização, calcular distâncias entre pontos do mapa, tornar visível ou não determinados planos de informação, e demais tarefas possíveis na tela de consulta do sistema.

3.2.3 Função dos usuários no sistema

A instalação dos softwares que compõem o módulo servidor do sistema e as configurações básicas são executadas pelos administradores do sistema. Estes tem a função de cadastrar usuários projetistas de mapas que tem direito a entrada ou “login” na interface web do sistema através de senha. Outra função destes é construir e disponibilizar novos modelos de mapas que serão utilizados pelos projetistas. Estes novos modelos são escritos no formato de parâmetros do UMN Mapserver, do sistema de moldes Chameleon e de diretivas em linguagem HTML e PHP , por meio de programação específica.

Após a instalação dos componentes servidores, os administradores podem configurar o sistema remotamente através de um navegador *web*.

Os projetistas de mapas utilizam um navegador *web*, registram-se com um nome de usuário e senha, fazem a criação de mapas dotados de planos de informação e selecionam modelos de visualização *webmaps* disponíveis no sistema.

Os passos básicos para a criação de um projeto de mapa são: definição de coordenadas geográficas do mapa base do projeto, definição de planos de informação, envio dos arquivos de imagens *raster* e vetoriais para o servidor do sistema e escolha de um modelo apresentação. Em uma próxima etapa, o projetista executa a operação de publicação do projeto. Isto faz com que o seu projeto esteja disponível na *web* para os usuários de consulta através de uma URL única do projeto. Estas etapas podem ser vistas de forma esquemática na figura 7.

Os usuários de consulta acessam a página do mapa através de um navegador e podem efetuar consultas nos mapas criados pelos projetistas. Estes usuários não necessitam de autorização de senha no sistema já que a página do mapa fica disponível com acesso público. Várias páginas de mapas podem ser sediadas em um mesmo servidor do sistema. A figura 6 mostra de forma esquemática estes papéis.

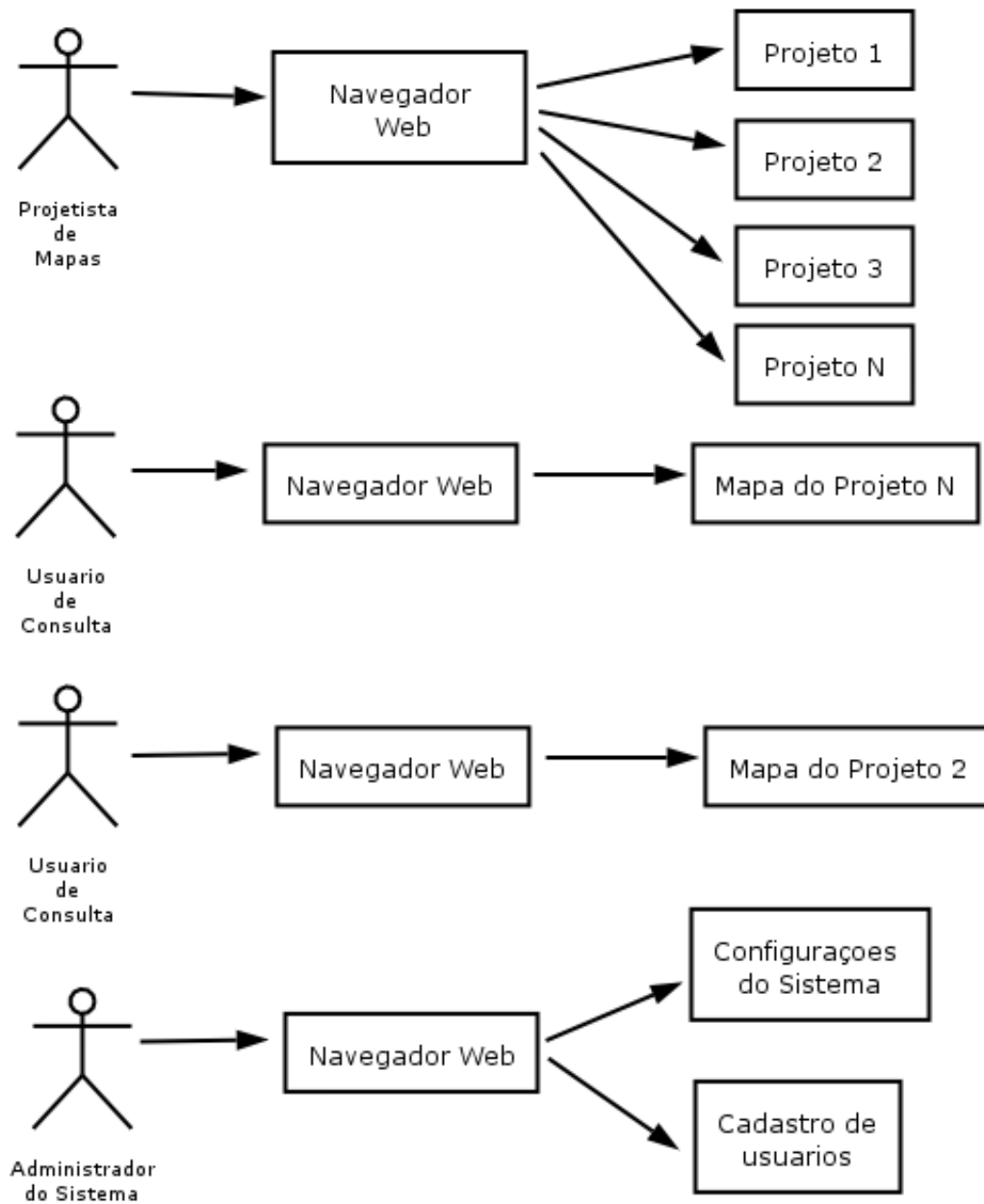


Figura 6: Papéis dos usuários do Sistema.

3.2.4 Funcionalidade do sistema

Para a publicação de um projeto são necessários a definição de parâmetros básicos para o funcionamento do UMN Mapserver e o envio de arquivos que constituem os planos de informação. Estes parâmetros são armazenados no SGBD Postgresql e os arquivos dos planos de informação, nos formatos raster (.gif, .png, .tif e outros) e vetoriais (ESRI Shapefiles) enviados por “upload” para um servidor de arquivos.

Durante a operação de “publicação” do projeto, o sistema cria automaticamente um arquivo de configuração para o UMN Mapserver que irá preparar o mapa para consultas através da web. Este arquivo de configuração utiliza os parâmetros do projeto armazenados em tabelas do SGBD Postgresql. Após qualquer alteração de dados do projeto o projetista deve efetuar novamente a operação de publicação para atualizar este arquivo de configuração.

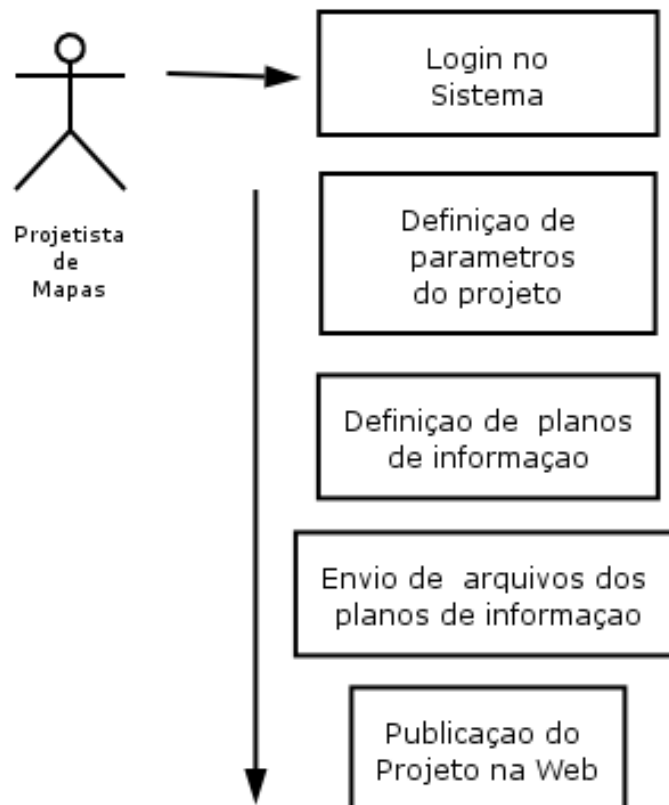


Figura 7: Passos para publicação de um webmap.

Após a publicação do webmap, uma página web do mapa fica disponível para consultas sobre os planos de informação. A figura 8 mostra um esquema de apresentação do mapa para o usuário que faz acesso às informações por meio de uma página em um navegador web.

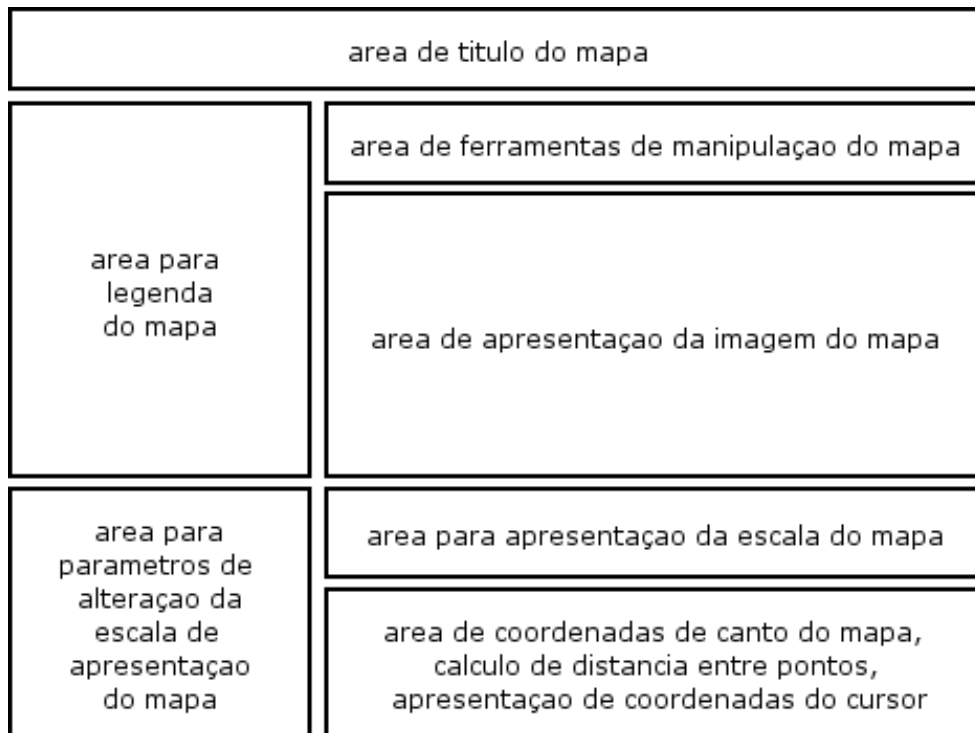


Figura 8: esquema de apresentação da tela de consulta do mapa.

A seguir lista-se a função das ferramentas do sistema:

- Servidor web Apache: responsável pela publicação das páginas web dos mapas através do protocolo HTTP.
- PHP: linguagem de programação executada pelo servidor web capaz de disponibilizar páginas em linguagem HTML geradas dinamicamente. O núcleo do sistema é um programa em linguagem PHP que consulta o servidor de banco de dados para disparar o funcionamento do servidor de mapas UMN Mapserver para a geração dinâmica de uma página web de mapas.
- SGBD Postgresql: sistema de gerenciamento de banco de dados, responsável pelo armazenamento de dados em tabelas relacionais que contem os parâmetros para publicação da estruturas dos mapas.

d) UMN Mapserver: responsável pela publicação das imagens gráficas finais dos mapas na *web*.

e) Sistema de moldes Chameleon: é uma ferramenta desenvolvida em linguagem PHP para construção de páginas web personalizadas utilizando os recursos do UMN Mapserver. Em resumo, é um conjunto de pré-configurações para uso do servidor de mapas UMN Mapserver, com funcionalidades de interface adicionais, tais como botões de ferramentas, legendas automáticas, e outros controles que possibilitam a reutilização deste código para aumentar a funcionalidade do UMN Mapserver. Através do Chameleon pode-se criar com facilidade, usando funções já prontas, páginas dotadas de mapas mais complexos do que se fizesse o uso diretamente do UMN Mapserver.

3.2.5 Tabelas do sistema

O sistema possui um banco de dados que está composto por quatro tabelas a citar: tabela de usuários; tabela de projetos; tabela de planos de informação e por último a tabela de parâmetros do sistema. A seguir descreve-se estas tabelas.

a) Tabela de Usuários: composta pelos dados de login, nome de usuário, senha e categoria dos usuários do sistema, onde a categoria é “administrador” ou “projetista de mapas”;

b) Tabela de Projetos: composta pelos dados de projetos de mapas, nome do projeto, descrição do projeto, largura do mapa na tela, altura do mapa na tela, coordenadas de canto do mapa(ponto superior direito e ponto inferior esquerdo), unidade das coordenadas(graus decimais ou métricas), modelo de apresentação do mapa, configurações do mapa de referência: coordenadas de canto, imagem, largura e altura, status (visível ou não visível na tela).

c) Tabela de Planos de Informação: armazena os planos de informação de cada projeto. Cada plano possui um arquivo *shapefile* ou *raster* que fica armazenado no servidor de arquivos do sistema onde cada projeto possui um ou mais planos de informação a ele vinculado;

d) Tabela de Parâmetros do Sistema: armazena informações de configuração do sistema, como a localização dos arquivos dos projetos no servidor, URL do sistema e outros parâmetros.

4. RESULTADOS E DISCUSSÃO

O presente trabalho teve como resultado a construção de um sistema multiusuário para divulgação de dados geográficos na *web*, com mecanismos para facilitar o processo de publicação destes dados. Através de um processo simples os usuários publicam informações geográficas em formato de planos de informação criando um *webmap* particular que pode ser acessado pela *web* para consultas. A manutenção destas informações é feita à distância, através de um navegador *web*. Estes dados ficam centralizados em um computador servidor, o qual pode armazenar vários *webmaps* simultaneamente. Cada *webmap* foi denominado como um novo “projeto”. O formato de publicação deste *webmap* é configurável através de “modelos” disponíveis para publicação. Estes modelos foram criados pelos administradores facilitando o trabalho dos usuários de publicação. Os usuários que publicam os planos de informação não necessitam conhecimentos avançados sobre tecnologias de Internet para publicação de seus dados na *web*. A seguir está descrita a estrutura e funcionamento do sistema.

4.1 Conceitos básicos para publicação das informações no sistema proposto

A publicação de informações através do sistema está baseada na apresentação final de um mapa, com recursos de georreferenciamento, disponível em uma página *web*. Para isto, é necessária a compreensão de conceitos básicos: os projetos de *webmaps* ou simplesmente projetos, planos de informação, atributos do sistema.

4.1.1 Projeto

Dentro do sistema, a criação de projeto é a forma para cadastrar uma nova página *webmap* independente. Ele é descrito no sistema por meio da interface de edição de projetos, mostrada na figura 10. Para cada projeto é necessário definir os seguintes parâmetros:

- coordenadas de canto da imagem: em graus decimais ou em metros, sendo definido o canto inferior esquerdo e o canto superior direito da imagem;
- nome do projeto;

- coordenadas das imagens do mapa de referência, idem às do canto da imagem;
- largura e altura da janela do mapa expressa em pixels;
- modelo de apresentação do *webmap*.

4.1.2 Planos de informação

Após a declaração dos parâmetros do projeto definiu-se os planos de informação vinculados ao *webmap*. Um projeto pode possuir um ou mais planos de informação a ele vinculados. Cada plano de informação é apresentado na tela de consulta do *webmap* do projeto podendo estes serem alterados durante a fase de consulta para ficarem visíveis ou não. No sistema são permitidos dois tipos de dados: *raster* e vetoriais. Os dados vetoriais são inseridos no sistema na forma de arquivos do formato *shapefile*. Cada plano de informação foi descrito no sistema através da interface de edição de planos de informação. Esta interface é mostrada na figura 11. Para cada plano é necessário definir os seguintes parâmetros:

- nome do plano de informação;
- nome do arquivo de dados;
- para arquivos vetoriais deve-se definir outros dois arquivos auxiliares, um com extensão *.dbf* e outro com extensão *.shx*);
- para vetores, devem ser definidas informando-se o nome das classes e as cores de contorno e de preenchimento do polígono no sistema RGB, com valores descritos para o grau de vermelho, verde e azul.
- estado do plano de informação se está inicialmente visível ou invisível.
- o índice de transparência do plano de informação que varia de 0 a 100 %.

Para cada plano de informação, após a definição dos seus parâmetros executa-se a operação de envio dos arquivos de imagens para o servidor de arquivos.

Após concluir a definição dos planos de informação, o projetista de mapas retorna para a interface de edição de projetos para executar a operação de “publicação” do *webmap*. Após esta operação, os dados já podem ser visualizados em um navegador *web* no formato de um *webmap* através de um endereço de URL específico do projeto criado. Este endereço é de acesso público, não necessitando de senha para visualização.

4.2 A Interface Homem-Computador

A interface homem-computador do sistema proposto é composta por objetos mostrados através de um navegador *web*. Os módulos desta interface estão listados a seguir:

- 1) *Login* ou entrada do sistema
- 2) Edição de Projetos
- 3) Edição de Planos de Informação
- 4) Edição de parâmetros do sistema
- 5) Tela de consulta de um *webmap*

As interfaces citadas anteriormente são acessíveis através de um navegador *web* apontado para um endereço URL do servidor do sistema. Nas figuras 9 a 11 são mostradas as interfaces principais dos módulos da IHC:

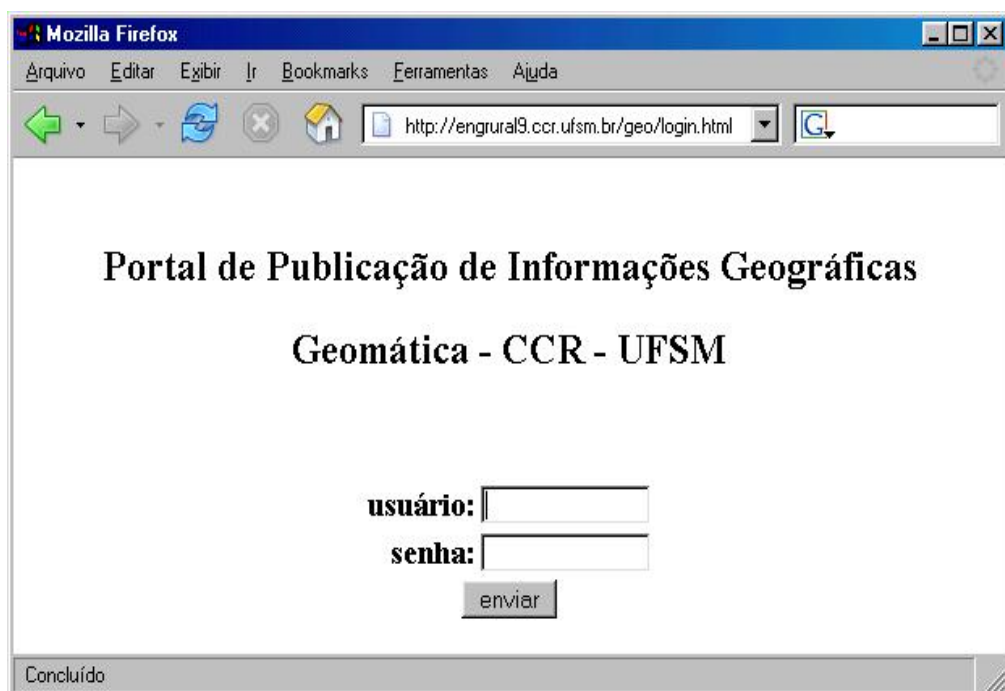


Figura 9: interface de entrada(login) no sistema.

Mozilla Firefox _ □ ×

Arquivo Editar Exibir Ir Bookmarks Ferramentas Ajuda

← → ↻ × 🏠 ↵

Edição de Projetos

id projeto : 1

Nome do projeto :

Descricao do projeto :

Largura do mapa :

Altura do mapa :

Unidade das coordenadas : ▾

Extensão do mapa :

Ponto inferior esquerdo :

Min X:

Min Y:

Ponto superior direito :

Max X:

Max Y:

Coordenadas do mapa de referência :

Ponto inferior esquerdo :

Min X:

Min Y:

Ponto superior direito :

Max X:

Max Y:

Imagem do mapa de referência :

Status do mapa de referência : ▾

Largura do mapa de referência :

Altura do mapa de referência :

Modelo de apresentação do mapa : ▾

Operação : ▾

- Salvar
- Novo
- Excluir
- Publicar
- Layers
- Pesquisar

Concluído

Figura 10: interface de edição de projetos de mapas.

Edição de Planos de Informação

projeto :1

Nome do plano de informação: areas_campus

Tipo de imagem: Shapefile

Status do plano de informação: ON

Nome da classe: areas_campus

Cor de preenchimento do poligono: 100 150 130

Cor do contorno do poligono: 0 0 0

Índice de Transparencia: 30

Arquivo principal : areas_campus.shp

Alterar arquivo principal: Arquivo...

Somente para arquivos Shapefile:

Alterar arquivo dbf: areas_campus.dbf Arquivo...

Alterar arquivo shx: areas_campus.shx Arquivo...

Operação: Enviar

- Salvar
- Novo
- Excluir
- Pesquisar

Concluído

Figura 11: Interface de edição de planos de informação.

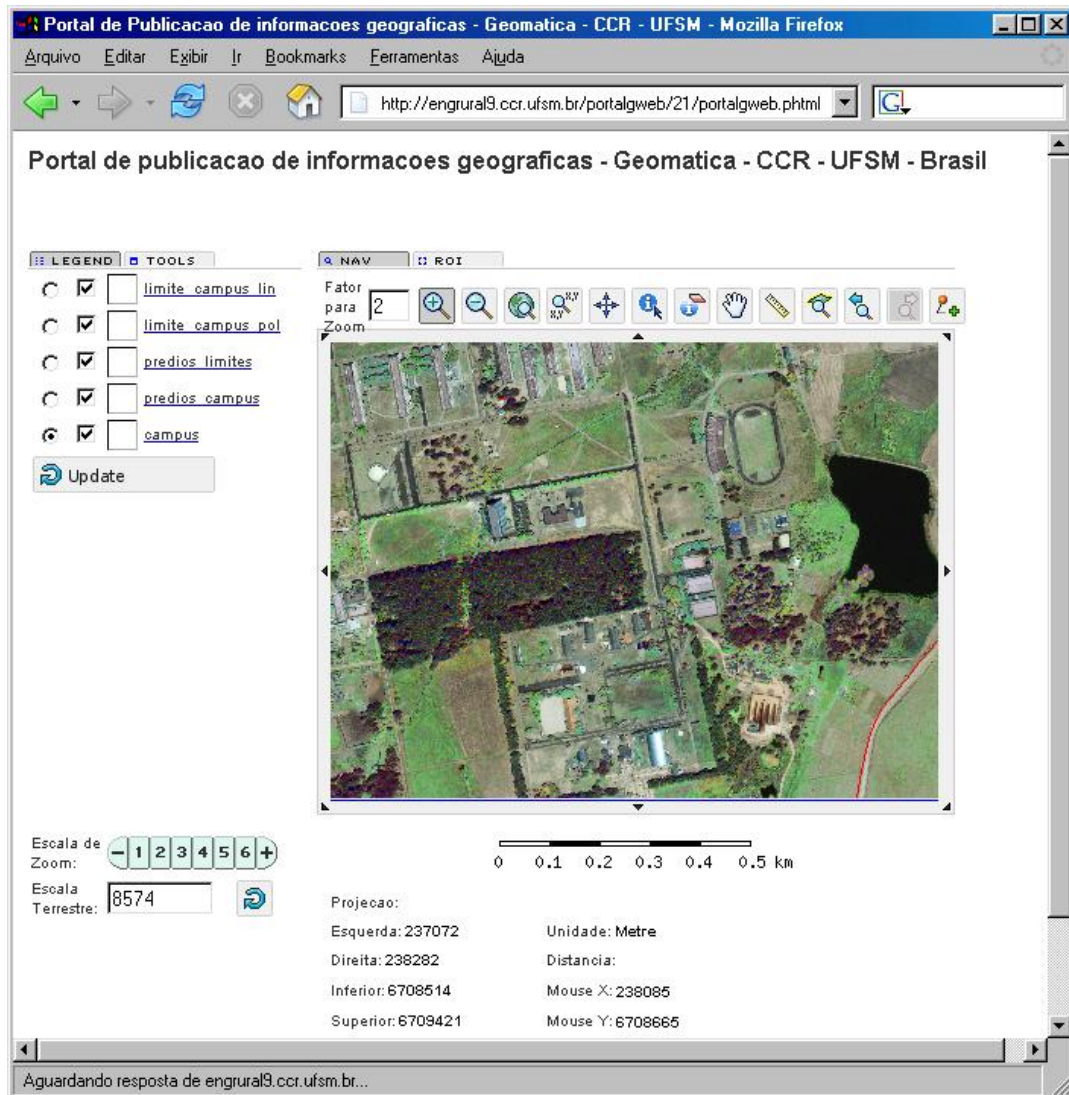


Figura 12: Tela de consulta de um webmap.

4.3 Tabelas do Banco de Dados

Os dados lançados nas interfaces gráficas mostradas na seção 4.2 foram armazenados em tabelas do banco de dados Postgresql. A seguir nas tabelas 2 a 5 são apresentados os tipos de armazenamento específicos do SGBD Postgresql e a descrição das colunas da estrutura interna deste banco de dados.

Tabela 2: Estrutura da tabela de Usuários.

Coluna	Tipo de dados	Descrição
idusuario	integer	Identificador do usuário (chave primária da tabela)
Login	character(12)	Nome reduzido para tela de entrada no sistema
Nome	character varying	Nome completo do usuário
Classe	character(3)	Classe do usuário (Administrador ou projetista de mapas)
Senha	character varying(15)	Senha de identificação para entrada no sistema

Tabela 3: Estrutura da tabela de Planos de Informação.

Coluna	Tipo de dados	Descrição
idlayer	integer	Identificador do plano de informação (chave primária da tabela)
name	character varying	Nome do plano de informação
idprojeto	integer	Identificador do projeto a que o plano está vinculado
tipo	character(12)	Tipo do plano (raster ou shape)
status	character(3)	Estado do plano (visível ou não visível)
nomearquivo	character varying	Nome do arquivo de dados do respectivo plano
classname	character varying(30)	Nome da classe do polígono caso tipo seja shape
colorclass	character varying(15)	Cor de preenchimento do polígono no padrão RGB
outlinecolor	character varying(15)	Cor de contorno do polígono
symbol	integer	Símbolo da legenda do plano de informação
transparency	integer	Índice de transparência do plano (0 a 100 %)

Tabela 4: Estrutura da tabela de Parâmetros do Sistema.

Coluna	Tipo de dados	Descrição
idusuario	integer	Identificador do usuário (chave primária da tabela)
login	character(12)	Nome reduzido para tela de entrada no sistema
nome	character varying	Nome completo do usuário
classe	character(3)	Classe do usuário (Administrador ou projetista de mapas)
senha	character varying(15)	Senha de identificação para entrada no sistema

Tabela 5: Estrutura da tabela de Projetos.

Coluna	Tipo de dados	Descrição
idprojeto	integer	Identificador do projeto (chave primária da tabela)
idusuario	integer	Identificador do Usuário a que criou o projeto
nome	character varying(100)	Nome do projeto
descricao	character varying(200)	Descrição do projeto
sizemap_largura	integer	Largura do mapa em pixels
sizemap_altura	integer	Altura do mapa em pixels
minx	double precision	Coordenada x do canto inferior direito do mapa
miny	double precision	Coordenada y do canto inferior direito do mapa
maxx	double precision	Coordenada x do canto superior esquerdo do mapa
maxy	double precision	Coordenada y do canto superior esquerdo do mapa
units	character(10)	Unidade das coordenadas (graus decimais ou metros)
referenceminx	double precision	Coordenada x do canto inferior direito do mapa de referência
referenceminy	double precision	Coordenada y do canto inferior direito do mapa de referência
referencemaxx	double precision	Coordenada x do canto superior esquerdo do mapa de referência
referencemaxy	double precision	Coordenada y do canto superior esquerdo do mapa de referência
statusreference	character(3)	Indicador do mapa de referência (visível ou não visível)
referencelargura	integer	Largura do mapa de referência
referencealtura	integer	Altura do mapa de referência
modelomapa	integer	Modelo do mapa do projeto (numerado por seqüência-1,2,3)
referenceimage	character varying	Arquivo de dados do mapa de referência

4.4 Exemplo de uso do sistema proposto: mapa do Campus da UFSM

Um exemplo de uso do sistema foi construído para demonstrar o funcionamento do sistema. Foram utilizados os seguintes planos de informação:

- imagem *raster* em formato geotiff com foto do satélite Ikonos do Campus da UFSM;
- digitalização de prédios do campus em dois formatos: linhas de contorno dos prédios e polígonos do prédio;
- Polígono que mostra o perímetro e área de propriedade da UFSM no bairro Camobi na cidade de Santa Maria.

O cadastro destas informações no sistema *web* geraram um mapa de consulta dos dados do campus que pode ser visto na figura 13.

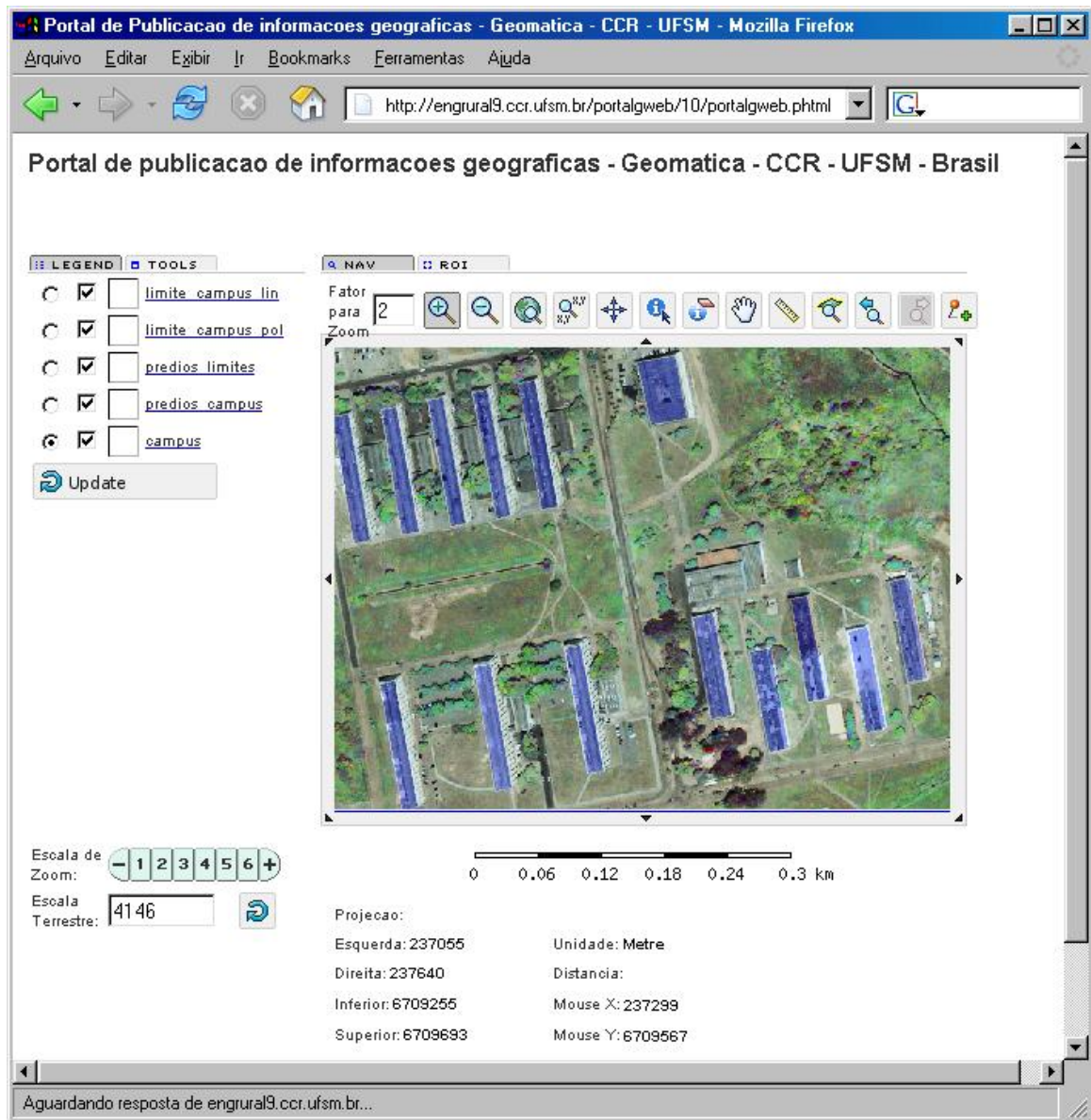












FIGURA 13: exemplo de aplicação de um *webmap* com dados do Campus da UFSM em Santa Maria-RS.

A tabela 6 apresenta a descrição dos principais botões da barra de ferramentas da interface apresentada na figura 13 e suas funções.

Tabela 6 – Descrição das funções da barra de ferramentas.

Fonte: Schetinger (2005).

Botão	Ação
	Executa zoom in
	Executa zoom out
	Faz zoom para o tamanho inicial da imagem
	Zoom para um quadro de coordenadas específicas
	Centraliza um ponto da imagem na tela
	Arrasta a imagem com uso do cursor
	Mede distância entre dois pontos
	Retorna para visualização anterior
	Mostra a visualização posterior
	Insere notas no mapa

4.4 Aspectos importantes do uso do sistema

O sistema proposto neste trabalho foi desenvolvido com ferramentas de software livre e possui como características os seguintes itens:

- gratuito, ou seja, não tem uma licença de uso do software com custos, e sim está sob a licença GPL;
- código fonte aberto que permite a sua manutenção e extensão e modificação sem necessidade de pagamento de licenças ou taxas para os desenvolvedores do sistema;
- a criação e publicação de mapas pode ser feita por usuários que não tenham conhecimento de linguagens de programação nem de instalação de ferramentas *web*. Muitas das ferramentas atuais de publicação de mapas exigem a programação em linguagens “script”. Com o uso do sistema os usuários necessitam apenas o conhecimento de criação de planos de informação e de coordenadas;
- o servidor do sistema está baseado em sistema operacional Linux, que permite o uso de computadores de custo mais acessível mesmo para utilização de serviços *web* que sustentem vários usuários conectados simultaneamente;

- um servidor pode sediar várias páginas de mapas web simultaneas. Isto permite reduzir a mão-de-obra de manutenção do servidor a uma pequena equipe de administradores;
- a instalação e configuração do sistema servidor é feita com facilidade;
- a configuração e manutenção do Sistema pode ser feita à distância;
- os projetistas de mapas, após cadastrados no sistema, podem criar e manter projetos à distância através de um navegador;
- através do uso do sistema usuários qualificados tais como profissionais da área de geomática, por exemplo, sediados em Universidades, podem auxiliar remotamente outros usuários a cadastrar seus dados e finalmente publicar webmaps. Pode-se formar assim uma rede de cooperação interdisciplinar entre Universidades e órgãos governamentais(por exemplo: prefeituras) para auxílio à distância usando navegadores web e o servidor do sistema instalado em qualquer das instituições.

5. CONCLUSÃO

No presente trabalho, utilizaram-se as seguintes ferramentas de código fonte aberto: sistema operacional Linux, linguagem de programação PHP, Sistema de Gerenciamento de Banco de Postgresql, servidor de mapas UMN Mapserver, sistema de moldes Chameleon.

A integração destas ferramentas possibilitou a criação de um sistema multiusuário para publicação de informações georreferenciadas.

O uso do sistema criado facilitou a publicação de informações na *web*, sem necessidade de conhecimentos de linguagens script ou de programação para *web* por parte dos usuários finais.

A tarefa de publicação de informações georreferenciadas foi transformada em um processo auxiliado por um sistema multiusuário dotado de um banco de dados, deixando o projetista de mapas se concentrar na preparação dos seus dados.

Através da construção de um modelo de dados e a criação de um programa em linguagem PHP foi possível integrar as ferramentas para a criação de uma estrutura que possibilite a publicação de dados geográficos na *web* com facilidade.

Muitas páginas de informações geográficas podem ser sediadas em um mesmo equipamento de hardware.

Usuários leigos podem através do uso do sistema publicar seus dados georreferenciados usando uma senha e um navegador *web*. Estes usuários não necessitam de conhecimento em tecnologia *web* para criação e publicação de dados no formato de um mapa interativo para *web*, também chamado de *webmap*.

A construção de um protótipo da ferramenta na forma de servidor de mapas para *web* foi construído para teste do sistema. Estas ferramentas juntas e funcionando corretamente demonstraram a facilidade da publicação de dados georreferenciados sem a necessidade de construção de programas complexos para a tarefa de publicação.

A arquitetura do sistema proposto permite a inclusão de recursos adicionais tendo a vantagem de substituição das ferramentas que compõe o servidor do sistema por versões mais atualizadas, já que todas são baseadas em licença de software livre.

A instalação da estrutura do servidor do sistema proposto está baseado em ferramentas gratuitas e de código fonte aberto, e isto evita gastos com pagamento de licenças de software.

A possibilidade de integrar profissionais através do uso de navegadores *web* pode facilitar o auxílio à organizações governamentais (por exemplo: administração municipal) reduzindo o custos e facilitando a tarefa de publicação de *webmaps* que visam auxiliar tarefas de gestão com tecnologia de dados georreferenciados.

Como recomendação para trabalhos futuros tem-se os seguintes itens como relevantes:

- construção de novos modelos para apresentação do sistema;
- construção de um instalador automático do servidor do sistema;
- inclusão de dados geográficos no banco de dados armazenando objetos geográficos diretamente no SGBD com a ferramenta PostGIS;
- teste de desempenho do servidor do sistema.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ALOV. **ALOV Map. Free Java GIS**. Disponível em < <http://alov.org/index.html>>.

Acesso em abril de 2005.

AUTODESK INC. **AutoDesk MapGuide**. Disponível em:

<<http://www.autodesk.com/prods/mapguide/>>. Acesso em abril de 2005.

ANATEC. **Site da cidadania**. Revista Cidades do Brasil. Edição de fevereiro de 2000. Disponível em:

< <http://www.cidadesdobrasil.com.br/tecnologia/tecnologia06.htm> > .

Acesso em outubro de 2003.

ASSAD, E. D. **Sistema de Informações Geográficas. Aplicações na Agricultura**. 2.ed., rev. e ampl. Brasília: Embrapa-SPI / Embrapa-CPAC, 1998. 434 p.

APACHE. **Apache Web Server**. Disponível em: <<http://www.apache.org>>. Acesso em setembro de 2003.

BOLFE, E. L. **Geoprocessamento aplicado a análise de recursos florestais estudo de caso : Folha SH. 22-Y-A / DSG**. Dissertação em Engenharia Agrícola - Universidade Federal de Santa Maria, 2001.

BIAZUS, D.O. **Um breve histórico do Postgresql**. Projeto Postgresql-br. 2003.

Disponível em: <<http://pgsqlbr.querencialivre.rs.gov.br/index.php?corpo=historico>> .

Acesso em outubro de 2003.

BLASBY, D. **Developing Applications with PostGIS**. Mapserver User Meeting. University of Minnesota. Junho de 2003.

CÂMARA, G., Davis, C. **Bancos de Dados Geográficos. Volume III da Coleção "Geoprocessamento: Teoria e Aplicações"**. São José dos Campos: INPE, 2004.

CÂMARA, G. et al. **Anatomia de Sistemas de Informações Geográficas**. Campinas: Unicamp. 1996.

DM SOLUTIONS. **Chameleon page**. Disponível em: <<http://www.dmsolutions.ca/technology/chameleon.html>>. Acesso em abril de 2005. (2005a).

DM SOLUTIONS. **Chameleon Help**. Disponível em: <<http://chameleon.maptools.org/help/viewer.phtml>> Acesso em abril de 2005. (2005b)

ESRI. **ArcIMS 4 Architecture and Funcionality**. Abril de 2002 Disponível em: <<http://www.esri-portugal.pt/produtos/documents/arcims4architecture.pdf>>. Acesso em abril de 2005.

ESRI. **ArcIMS**. Disponível em: <<http://www.esri.com/software/arcims/index.html>> . Acesso em abril de 2005.

FERREIRA, K. R. **Interface para operações espaciais em bancos de dados geográficos**. Dissertação em computação aplicada. INPE, 2003.

FREE SOFTWARE FOUNDATION. **Licença Pública GNU**. 1991. Disponível em <<http://members.tripod.com.br/RamonFlores/GNU/gpl.html>>. Acesso em outubro de 2003.

HEUSER, C. A. **Projeto de Banco de Dados**. Porto Alegre: Sagra Luzzatto, 2000.

HERZMAN, D. **Serving Iowa Mesonet Data with University Minnesota's Mapserver**. Iowa Mesonet Environmental. 2002. Disponível em: <<http://www.gis.ucar.edu/02workshop/presentations/herzmann.pdf>> . Acesso em outubro de 2003.

INPE. **Spring: Sistema de Processamento de Informações Georreferenciadas**. Disponível em <<http://www.dpi.inpe.br/spring/>>. Acesso em abril de 2005.

INTERGRAPH CORPORATION. **GeoMedia Web Map**. Disponível em: <<http://www.intergraph.com/gis/gmwm/>> . Acesso em abril de 2005.

INTERGRAPH. **Assessment Solutions That Make Business Sense**. Janeiro de 2004. Disponível em <<http://imgs.intergraph.com/imgsfiles/wp/WP1038B.pdf>>. Acesso em abril de 2005.

LIMP, W. F. **Mapping hits warp speed on the World Wide Web!**. 1999. Disponível em <<http://www.geoplace.com/gw/1999/0999/999tec.asp>>. Acesso em: abril de 2005.

LEYH, F.C.C.W. Monitoramento geográfico da programação pactuada e integrada (PPI/PDR) de Serviços e Investimentos em Saúde Pública na Internet com um “Servidor de Mapas”. CONGRESSO BRASILEIRO DE SAÚDE COLETIVA, 7., 2003. Brasília. **Anais...** Brasília: ABRASCO , 2003.

MAPSERVER PROJECT. **MapServer 4.0 Tutorial**, disponível em <<http://terrasip.gis.umn.edu/projects/tutorial/>> . Acesso em outubro de 2003.

MARISCO, N. PHILIPS, J. PEREIRA, H.R. Protótipo de Mapa para Web Interativo: uma abordagem utilizando código aberto. **Revista Brasileira de Cartografia**, v.56, n. 1, 2004

MINNESOTA DEPARTMENT OF NATURAL RESOURCES. **ForNet Applications**. Disponível em: <<http://www.ra.dnr.state.mn.us/>>. Conteúdo editado em 2001. Acesso em outubro de 2003.

MIRANDA, J.I. **Diretivas para disponibilizar mapas na Internet**. Campinas : Embrapa Informática Agropecuária, 2002. 29 p. — (Documentos / Embrapa Informática Agropecuária ; 14)

MIRANDA, J. I., SOUZA, K. X. S. Como Publicar Mapas na Web. XI SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO, 11., 2003, Belo Horizonte, **Anais...** BH: INPE, p. 349-355

MITASOVA, H. e NETELER, M. Freedom in geoinformation science and software development: A GRASS GIS contribution. THE OPEN SOURCE FREE SOFTWARE

GIS – GRASS USER CONFERENCE, 2002, Trento, **Proceedings...** Università di Trento, 2002

OPEN GIS CONSORTIUM (OGC). **The importance of going “Open”**. Disponível em : <<http://www.opengis.org/press/?page=papers>> . Acesso em outubro de 2003.

OPEN GIS CONSORTIUM (OGC) - **Open GIS Consortium**. Disponível em: <<http://www.opengis.org/techno/specs.htm>>. Acesso em abril de 2005.

PEREIRA, E.T.L. **Uso de SIG na análise da distribuição espacial de escolas. Estudo de caso: município de Formigueiro, RS**. Dissertação de Mestrado em Geomática – Universidade Federal de Santa Maria, 2001

PEREZ, A.L.F; FOGAÇA, E.T.; POZZEBON, E., Sistema de informações da serra catarinense. **Revista do CCEI-URCAMP**, v. 12, 2003.

PHP DOCUMENTATION GROUP. **Manual do PHP**. 2005. Disponível em: <http://www.php.net/manual/pt_BR/>. Acesso em abril de 2005.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. **Tutorial do Postgresql 7.3.2**. 2002. Disponível em: <<http://pgsqlbr.querencialivre.rs.gov.br/downloads/tutorial.pdf>>. Acesso em: outubro de 2003.

PROJETO GNU. **Categorias de software livre e não livre**. Disponível em <<http://members.tripod.com.br/RamonFlores/GNU/gpl.html>>. Acesso em outubro de 2003.

SILBERSCHATZ,A., et al **Sistema de banco de dados**. McGraw-Hill: MAKRON Books, 1997.

SILVA, F.A.M. e MARIZ FILHO, F.G. Mysql server: velocidade. **Revista do Linux**, n. 5, 2000.

STRIEDER, I.T. **Elaboração e análise do mapa de conflitos ambientais do Município de Salvador da Missões – RS**, Monografia em Geomática, Universidade Federal de Santa Maria, 2002.

TYDAC INC. **Ortsplan Mapserver - Swiss on-line Maps based on Tele Atlas and Raster Tiles**. Disponível em <<http://www.ortsplan.ch/>>. Acesso em outubro de 2003.

VASSOLER, G. , BRITO, H.M., BORTOLON, S. Desenvolvimento de Aplicações GIS para Internet: de componentes ActiveX até a GML. WORKSHOP BRASILEIRO DE GEOINFORMÁTICA, 3., 2001, RJ. **Anais...** RJ: IME, 2001

VIEIRA, C. E. & MOTTA, J. **SIGPOA-Informativo do Sistema de Informações Geográficas da Prefeitura de Porto Alegre**. Porto Alegre: SPM, 2001.

WETHERBE, J. C. **Análise de Sistemas de informação por computador**. 3 ed. Rio de Janeiro: Campus, 1987.