

MAFWeb: o SIG Defesa

Marcelo Corsino Ferreira, Pier-Giovanni Taranti, Leandro Ouriques Mendes de Carvalho e Aldo Dolorico Balbi
Centro de Análise de Sistema Navais (CASNAV), Praça Barão de Ladário s/n, Ilha das Cobras, ed. 8 - 3º Andar - AMRJ
Centro - Rio de Janeiro - RJ - Brasil - 20091-000

Resumo — Em um Sistema Militar de Comando e Controle, o atributo fundamental para obtenção da Consciência Situacional é a visualização do Teatro de Operações que permite o acompanhamento das Forças. Normalmente, utiliza-se uma interface gráfica que exhibe o posicionamento, as características e os inter-relacionamentos dos meios aéreos, navais e terrestres constituídos em operações de paz, crise ou guerra. No âmbito do Ministério da Defesa, o principal braço de Comando e Controle atualmente utilizado é o Sistema de Planejamento Operacional Militar (SIPLM). O Módulo de Acompanhamento de Forças (MAFWeb), integrante do SIPLM, tornou-se o SIG do MD, utilizado frequentemente nas Operações Conjuntas.

Palavras-chaves — SISM², Consciência Situacional, MAPSERVER.

I. INTRODUÇÃO

Nos últimos dez anos houve uma grande mudança de paradigma no planejamento e execução de operações militares. Com a realidade imposta pela Era da Informação e a associação entre as demandas da sociedade e ao aparato tecnológico disponível, a informação passou a ser um fator decisivo, não somente no setor comercial, mas também no complexo e exigente setor militar.

Um Sistema Militar de Comando e Controle (SISM²) possui diversas características que vão auxiliar o comandante na tomada de decisão em um Teatro de Operações. Trata-se de um conjunto de instalações, equipamentos, comunicações, doutrinas, procedimentos e pessoal, essenciais para o comando, em nível nacional, das crises e dos conflitos [1]. O Sistema de Planejamento Operacional Militar (SIPLM) é o principal braço de Comando e Controle (C²) do Ministério da Defesa. O SIPLM é um sistema de informação computacional, de arquitetura modular, que tem como objetivo, apoiar o Comandante Supremo em seus processos de tomada de decisão. A interoperabilidade entre o MD e as Forças Armadas (FA) foi possível, devido à sua instalação nos diversos centros de C² das FA, aumentando a capacidade de intercambiar serviços ou informações entre eles.

Os módulos do SIPLM têm como objetivos armazenar, replicar e apresentar diferentes tipos de dados, de forma rápida e segura, utilizando os recursos de uma *intranet*.

Dentre os vários módulos do SIPLM, a principal ferramenta de acompanhamento dos meios aéreos, navais e terrestres é o Módulo de Acompanhamento de Forças Web (MAFWeb). O MAFWeb apresenta os meios georreferenciados e foi projetado para adequar-se à praticidade de utilização segundo um navegador *web*. Tem por objetivo o acompanhamento de operações em andamento, possibilitando a identificação de feições geográficas de interesse, apresentando o quadro estratégico-operacional, conhecido atualmente como “Consciência Situacional” (CS). Permite ainda que, dependendo do grau de CS, o ciclo de comando e controle (Observar, Orientar-se, Decidir e Agir - OODA) possa ser

efetivado, tornando-o uma excelente ferramenta de apoio à decisão. O MAFWeb apresenta as posições georreferenciadas atualizadas de navios, tropas e aeronaves, sobre uma ou mais camadas de informações geográficas de interesse (mapas e cartas). Contudo, o MAFWeb torna possível estender a CS a um número maior de utilizadores, já que funciona em um navegador *web* (*browser*), no lado cliente.

Inicialmente, o MAFWeb foi implementado a partir do I3GEO [2]. O I3GEO é um software para *Internet* baseado em um conjunto de diversos softwares livres, principalmente uma ferramenta de *webmapping* [3] conhecida por MapServer [4,5]. O foco principal do I3GEO é a disponibilização de dados ao público, aliados a um conjunto de ferramentas de navegação, geração de análises, compartilhamento e geração de mapas sob demanda, sendo uma ferramenta *web* de código aberto desenvolvida pelo Ministério do Meio Ambiente (MMA). A decisão em utilizar o I3GEO foi viável no início do desenvolvimento do MAFWeb, uma vez que algumas funcionalidades básicas de um Sistema de Informações Geográficas (SIG), tais como *pan*, *zoom* e *pick* (seleção) já estavam presentes, o que fez diminuir o tempo de implementação. Foi elaborado um estudo do I3GEO para adaptá-lo e otimizá-lo de acordo com as necessidades provenientes do levantamento de requisitos do sistema. Para isso, foi necessário compreendê-lo em um baixo nível de código-fonte. Durante as customizações advindas dos requisitos do cliente, o I3GEO portava-se como um sistema de pouca flexibilidade devido ao alto nível de acoplamento entre os componentes do software, fazendo com que pequenas modificações resultassem em muitas outras mudanças no código. A complexidade inerente ao sistema aumentou, fazendo com que as funcionalidades presentes no I3GEO, não mais atendessem às necessidades. Entretanto, um grande passo já havia sido dado, pois foram adquiridos conhecimentos importantes para elaboração de um SIG. Desde então, decidiu-se que para a nova versão do MAFWeb seria desenvolvido um novo núcleo para o sistema, de modo que as funcionalidades consideradas eficientes do MAFWeb/I3GEO fossem incorporadas. A versão atual do MAFWeb é totalmente independente do *framework* I3GEO.

Este artigo está organizado em cinco seções. A primeira apresenta uma introdução com os conceitos e histórico do sistema. A seção 2 descreve as características do MapServer, *engine* responsável pela criação dos mapas no servidor. A seção 3 apresenta as linguagens de programação e os bancos de dados utilizados no desenvolvimento do sistema. A seção 4 ilustra as principais funcionalidades do MAFWeb. Finalmente, a seção 5 apresenta as considerações finais e trabalhos futuros.

II. MAPSERVER

O MapServer, que surgiu de um projeto de código aberto, tem como finalidade exibir mapas dinâmicos através de um

navegador *web*. A sua arquitetura é representada na Fig. 1. Em sua forma mais básica, o MapServer é um programa *Common Gateway Interface* (CGI) que fica inativo no servidor *web*. Quando uma requisição é enviada ao MapServer, ele usa as informações transmitidas pelo *Universal Resource Locator* (URL) e um arquivo (*mapfile*) contendo a descrição do mapa para criar uma imagem do mapa solicitado. O MapServer também pode ser utilizado independentemente do modo CGI através do *MapScript* [6].

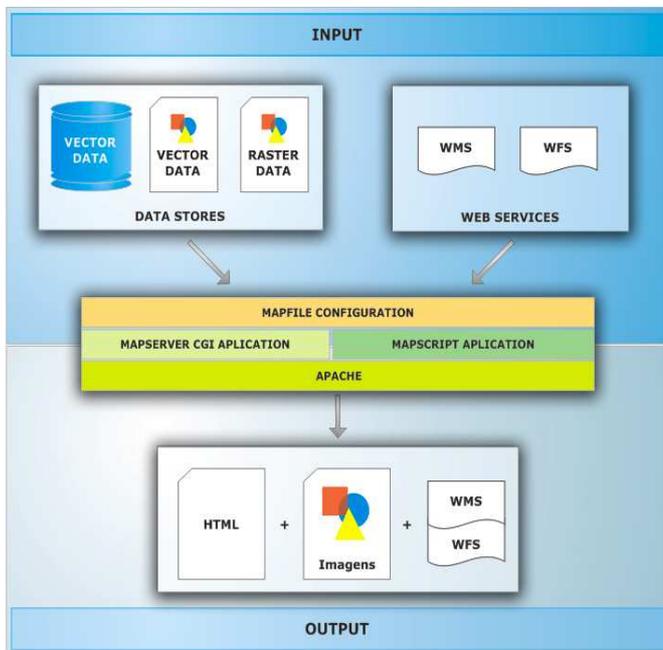


Fig. 1. Arquitetura do MapServer

O *Mapscript* é uma *Application Programming Interface* (API) orientada a objeto que fornece uma interface de *scripts* para o MapServer, a fim de construir aplicações *web* e *stand-alone*. Atualmente, ele oferece suporte às seguintes linguagens: PHP, Perl, Python, Ruby, Java e .NET. Ele permite a criação e atualização dinâmica do arquivo do mapa (*mapfile*). O *mapfile* é um texto estruturado que contém a configuração do mapa que será utilizado no sistema. Nele, define-se o tamanho e a extensão da área do mapa, indica-se a imagem que será gerada pelo seu processamento, lista-se suas respectivas camadas, incluindo a fonte de dados, projeções, simbologia, entre outros.

O MapServer “interpreta” um arquivo que contém as especificações do mapa (*mapfile*). Um mapa é constituído pela superposição de um conjunto de camadas e cada camada faz referência a uma fonte de dados. O MapServer suporta uma grande variedade de formatos de arquivos de entrada que podem ser divididos em dois grupos: arquivos com dados vetoriais e arquivos com dados *raster* ou matriciais.

Os dados vetoriais são representados em forma de coordenadas pelas unidades básicas de informação espacial: pontos, linhas e polígonos. Alguns dos formatos atualmente suportados são: os *shapefiles*, que correspondem a uma coleção de arquivos com extensões: SHP, SHX, DBF, PRJ[7]; DGN, formato vetorial gerado por ferramentas CAD e GPX, arquivo do tipo XML que representa o formato padrão para o intercâmbio de dados GPS entre aplicações e serviços *web* na *Internet* [8]. Os dados *raster* ou matriciais representam imagens digitais.

As camadas do tipo *raster* contêm a referência à imagem que ela irá exibir no mapa. Entretanto, a imagem não contém nenhuma informação que indique onde o MapServer deve posicioná-la no mapa. A informação de georreferenciamento está inserida em um arquivo com extensão *.wld* (*world*). A geração desse arquivo é realizada de forma automática [9].

Os arquivos de imagens nos formatos BMP, JPG, PNG, KAP e TIFF são exibidos por intermédio da biblioteca GDAL [10]. Esta biblioteca é compilada e linkeditada em conjunto com o Mapserver. A camada que referencia uma imagem é processada pela biblioteca GDAL. Outro formato de arquivo suportado pelo MAFWeb é o GeoTIFF que já possui definido internamente o georreferenciamento da imagem. O arquivo de imagem KAP é definido pelo padrão BSB, formato utilizado para reproduções de mapas topográficos e cartas náuticas [11]. Ele consiste de três arquivos principais de cartas: o arquivo de documentação (BSB), o arquivo de imagem (KAP) e um tipo de arquivo que informa as correções e atualizações (PTC).

III. ARQUITETURA DO SISTEMA

O MAFWeb é um sistema *web*. O processamento que ocorre no servidor para criar os mapas é transparente ao usuário final. O cliente recebe o código HTML dinâmico resultante do processamento. O sistema foi desenvolvido seguindo os padrões definidos pelo W3C [12], consórcio de empresas de tecnologia que regulamenta e desenvolve padrões para a criação e a interpretação dos conteúdos na *web*. O código HTML [12, 13] que descreve o *layout* do sistema, bem como as classes de estilo descritas em *Cascade Style Sheet* (CSS) [12, 13] foram escritos seguindo esse padrão.

Grande parte dos navegadores não está padronizado conforme o W3C. Dessa forma, eles podem interpretar comandos de maneira incorreta e, conseqüentemente, a interface pode sofrer distorções em navegadores distintos. O sistema foi desenvolvido para ser utilizado no navegador Firefox 3 [14], mas também tem se apresentado estável no Safari 4 [15], Opera [16] e Chrome 1 [17].

O PHP [18] é a linguagem interpretada responsável por gerar o código HTML dinâmico no servidor. Com o PHP é possível se conectar aos bancos de dados e a outras fontes que contêm as informações do sistema, além de processar essas informações e gerar as páginas *web* dinamicamente para o cliente.

O SIPLM faz acesso a duas bases de dados: uma em Firebird [19] e outra em PostgreSQL [20]. Este último foi agregado ao sistema devido, principalmente, às suas funcionalidades geoespaciais. No PostgreSQL são armazenadas as informações georreferenciadas dos acompanhamentos uma vez que o Firebird não possui suporte a dados espaciais. Os dados espaciais são gerenciados no PostGIS que é uma extensão do PostgreSQL o qual permite que objetos geográficos sejam armazenados no banco de dados [21]. O PostGIS permite definir novos tipos de dados: ponto, linha, polígono e também possui um conjunto de funções que realizam os cálculos geométricos e analisam as relações entre os objetos geográficos. Estas funções são definidas no *Open GIS Consortium* (OGC) [22]. Algumas dessas funções foram aplicadas no MAFWeb para determinar os objetos que estariam localizados dentro de uma área definida na consulta, seja esta área circular ou poligonal. A linguagem PHP possui uma grande quantidade de bibliotecas

que executam funções específicas: manipulação de arquivos ZIP, XML e PDF. Neste projeto, a principal biblioteca utilizada foi o *MapScript* [6] que fornece uma interface de *scripts* para o MapServer. Suas classes instanciam objetos que gerenciam as características dos elementos descritos no *mapfile*. Alguns exemplos e classes são: *mapObj*, *layerObj*, *classObj*, *labelObj*, *styleObj* e *colorObj*. Também foram utilizadas as bibliotecas de conexão ao Firebird e PostgreSQL.

No MAFWeb, o *Javascript* [13] também foi bastante utilizado, sobretudo para aperfeiçoar a interatividade do usuário com o aplicativo. Isto pode ser notado em todos os aparatos de navegação, tais como menus, a árvore de operações, as janelas, e as ferramentas de manipulação e interrogação do mapa e dos meios apresentados. É uma linguagem *script* a qual é interpretada no lado cliente, isto é, pelo navegador em que está sendo executado o aplicativo *web*. Ela segue o padrão DOM [12,13] estabelecido pelo W3C, que permite alterar e editar a estrutura, conteúdo e estilo de um documento eletrônico.

Além disso, o uso metodológico do *Javascript* com o XML [12,13] permite que toda a navegação do usuário aconteça praticamente dentro de um único documento, que é modificado dinamicamente a cada comando. Não existe troca de páginas e sim a manipulação de dados e alteração de conteúdo. Com o objeto XMLHttpRequest é possível conectar-se de forma assíncrona (independente da linearidade da execução do *script*) ao servidor e recuperar somente os dados requisitados. Apenas o que é de interesse do usuário trafega entre o cliente e o servidor, não havendo mais necessidade de recarregar todo o conteúdo, o que significa um grande ganho de desempenho do aplicativo. Ressalta-se aqui, a utilização do protocolo HTTPS, garantindo segurança ao tráfego de informações do sistema.

Outra novidade no aprimoramento do modo de utilização do *Javascript* foi o *Javascript Object Notation* (JSON) [23], que é um formato para intercâmbio de dados computacionais. Através dele o *Javascript* pode ser construído de forma orientada a objeto. Tanto os dados trocados com o servidor, como os manipulados no lado cliente seriam organizados em objetos. Outras bibliotecas que utilizam a mesma metodologia JSON de estruturação do código foram incorporadas ao MAFWeb no intuito de aperfeiçoar o *design* e a navegabilidade, tais como: a biblioteca *Jquery* com o *plugin* "Scroll to", para a navegação da árvore; a biblioteca *Prototype* [24] com os *plugins* "Scriptaculous" [25] para a reorganização da ordem de listas e imagens, o "Colorpicker" para seleção de cores; e a biblioteca *Wz_jsgraphics* [26] para desenhar formas geométricas (linhas, círculos, elipses, polígonos) de maneira dinâmica.

A interação entre esses recursos computacionais utilizados para desenvolver o MAFWeb é ilustrada na Fig. 2. No lado servidor, ela apresenta o servidor *web apache*, através do qual as páginas são acessadas pelo cliente, que se comunica com o interpretador do PHP. Este acessa as bases de dados e a *engine* do MapServer para gerar uma imagem do mapa e montar a página *web*. Este conteúdo é enviado para o lado cliente que o acessa através de um navegador (*browser*).

A interface do MAFWeb foi desenvolvida seguindo os princípios de usabilidade [27].

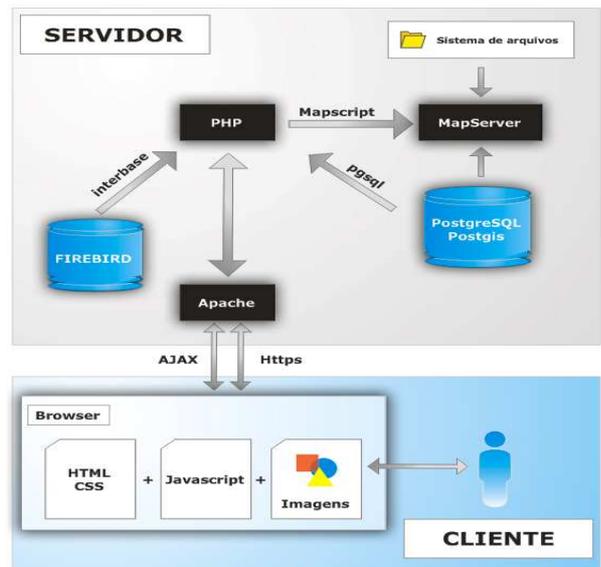


Fig. 2. Arquitetura do MAFWeb

IV. MAFWEB

O MAFWeb (Fig 3) possui um menu superior que provê acesso às funcionalidades que definem os meios e camadas que serão exibidos no mapa. Essas funcionalidades permitem salvar, carregar e remover camadas e cenários, além de prover a gerência das configurações do usuário. A barra de ferramentas, também localizada na parte superior, contém funções que realizam operações de *pan* e *zoom*, consultas espaciais, cálculo de distâncias, inserção de textos e interrogação dos meios e mapas.

No lado esquerdo é carregada a Árvore de Operações (AOp) que apresenta os acompanhamentos dentro de uma hierarquia por País, Força Armada, Natureza das Operações e Operação. Existe também um quadro que apresenta informações básicas do acompanhamento selecionado na AOp. O mapa ocupa a maior parte da área da interface que pode ser expandida quando oculta-se a AOp e o quadro de informações básicas (Fig 4). O rodapé da interface apresenta a data de referência dos dados carregados, a escala do mapa, a posição referente ao cursor do mouse no mapa em coordenadas geográficas (latitute e longitude) e coordenadas UTM.

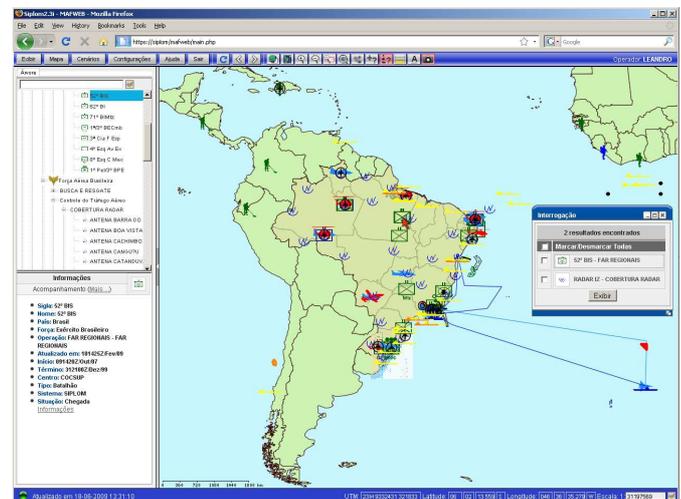


Fig. 3. Interface do MAFWeb

Para verificar o comportamento do MAFWeb, diversos testes de carga foram realizados. Cerca de 50 camadas foram carregadas, utilizando diversos tipos de arquivos de mapas, entre eles o SHP, DGN, GeoTiff e KAP. Dados provenientes do PostGIS também foram testados. Referências de imagens de até 70 MB de tamanho foram utilizadas na interface. Durante a fase de implantação do software no MD, os testes foram acompanhados pelos operadores do Centro de Operações do Comando Supremo (COCS). Esse grupo de usuários qualificou a interface nos testes, visto que o acesso às informações georreferenciadas foi considerado bastante satisfatório, conforme métrica utilizada pelos operadores. Desde a implantação da primeira versão do MAFWeb, foi possível visualizar a Consciência Situacional nas Operações Conjuntas (Combinadas) PORAQUÊ (2008) e ATLÂNTICO (2008). Na Fig. 4 é possível observar a interface do MAFWeb com alguns tipos de mapas e cartas nos formatos SHP, DGN e KAP.

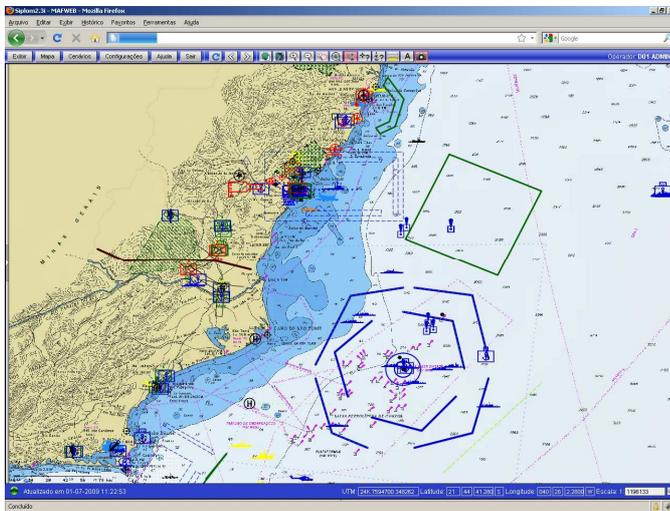


Fig. 4. Camadas de Informações agregadas ao MAFWeb

V. OBSERVAÇÕES FINAIS

O Ministério da Defesa necessita de ferramentas computacionais que apoiem o Comandante Supremo na tomada de decisão, seja em eventos em situações de paz, crise ou guerra. Atualmente, como parte de seu SISMC², o SIPLOM disponibiliza diversas funcionalidades que vão permitir a visualização da Consciência Situacional, conforme solicitações advindas dos níveis político-estratégicos. O MAFWeb, ferramenta principal de visualização, utiliza o MapServer, permitindo que o sistema suporte diversos formatos de arquivos de imagens. Isto contribuiu para que o sistema tivesse uma integração mais eficiente com as tarefas diárias das três Forças. O software processa imagens vetoriais geradas em ferramentas CAD que são utilizadas, tanto pelo Exército Brasileiro, como pela Força Aérea Brasileira, além de abrir cartas eletrônicas de navegação, disponibilizadas e utilizadas pela Marinha do Brasil.

A consolidação do MAFWeb gerou novas necessidades. Estão sendo desenvolvidas novas funcionalidades que aproveitam os recursos de georreferenciamento disponíveis no MapServer para que se permita interrogar os trajetos, gerar um trajeto a partir da ferramenta de medir distância no mapa; calcular a área e o perímetro das áreas circulares e

poligonais desenhadas no mapa; calcular o ponto de máxima aproximação e ponto de cruzamento entre acompanhamentos. Avanços na utilização do MapServer despertaram interesses em novos estudos. Pretende-se que o mapa seja gerado utilizando o modo *tile* para otimizar sua navegação e diminuir o tempo de carga de suas respectivas imagens. Serão realizados estudos para disponibilizar a alteração dinâmica da projeção do mapa e de suas camadas (*on the fly*). Já foi confirmada a viabilidade para promover a integração do MapServer com o *Google Maps* e *Google Earth*, uma vez que é possível gerar o arquivo KML de leitura destes aplicativos a partir do *mapfile*. Serão utilizados os recursos de *web services* do MapServer (WMS e WFS) para permitir carregar novas camadas no mapa. Estas novas camadas serão provenientes de outros sistemas, solidificando o conceito de interoperabilidade no MAFWeb. Além disso, o sistema deverá prover interoperabilidade com outros dispositivos eletrônicos, destacando-se o desenvolvimento de uma versão *Mobile*.

REFERÊNCIAS

- [1] Doutrina Militar de Comando e Controle, Ministério da Defesa, Ref. MD31-D-03, Ed. 2006.
- [2] I3GEO: Portal do Software Público Brasileiro. Disponível em: <http://www.softwarepublico.gov.br>
- [3] Tyle Mitchell, "Web Mapping Illustrated Using Open Source GIS Toolkits", O'Reilly, Junho 2005.
- [4] Mapserver. Disponível em: <http://www.mapserver.org>
- [5] Kropa, B., "Begging MapServer: Open Souce GIS Development", Editora Apress.
- [6] McKenna, J., "PHP/Mapscript Class Reference", versão r7813 (16/07/2008).
- [7] ESRI Shapefile Technical Description - An ESRI White Paper, July1998. Disponível em: <http://www.esri.com/library/hitepapers/pdfs/shapefile.pdf>.
- [8] GPX. Disponível em: <http://www.topografix.com/gpx.asp>.
- [9] Moretti, R. B. e Vencovsky, V. P., "Tutorial de Georreferenciamento de Imagens". Disponível em <http://www.gismaps.com.br>. Acessado em 26/09/2008.
- [10] GDAL. Disponível em: <http://www.gdal.org>.
- [11] Arraes, K. A., 2006, *Uma Ferramenta para Visualização e Simulação de Exercícios Navais em Cenários Marítimos Reais*. Tese de M.Sc., Instituto de Computação, Programa de Pós-Graduação em Computação, UFF. Rio de Janeiro.
- [12] W3C – Web Standards. Disponível em: <http://www.w3.org/>
- [13] W3Schools Online Web tutorials. Disponível em: <http://www.w3schools.com>.
- [14] Firefox 3. Disponível em: <http://br.mozdev.org/>
- [15] Apple - Safari. Disponível em: <http://www.apple.com/safari/>
- [16] Opera Browser. Disponível em: <http://www.opera.com>
- [17] Google Chrome. Disponível em: <http://www.google.com/chrome/>
- [18] PHP – Hypertext Preprocessor. Disponível em: <http://www.php.net/>
- [19] Firebird. Disponível em: www.firebirdsql.org.
- [20] PostgreSQL. Disponível em: <http://www.postgresql.org>.
- [21] Santili, S., Leslie, M., Hodgs, C. *et al.*, "PostGIS Manual" versão 1.3.3, (12/04/2008).
- [22] Open GIS Consortium, Inc., "OpenGIS Simple Features Specification for SQL" Revision 1.1. (05/05/1999).
- [23] JSON (JavaScript Object Notation). Disponível em: <http://www.json.org/>
- [24] Prototype JS Framework. Disponível em: <http://www.prototypejs.org>
- [25] Scriptaculous JS Framework. Disponível em: <http://script.aculo.us>.
- [26] Javascript Vectorgraphics Library. Disponível em: http://www.walterzorn.com/jsgraphics/jsgraphics_e.htm.
- [27] Santos, R. L. G., 2006, *Usabilidade de Interfaces para Sistemas de Recuperação de Informação na Web*. Tese de D. Sc., Pontifícia Universidade Católica do Rio de Janeiro – PUC-RJ, Rio de Janeiro.