



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Antonio Ignácio Glória

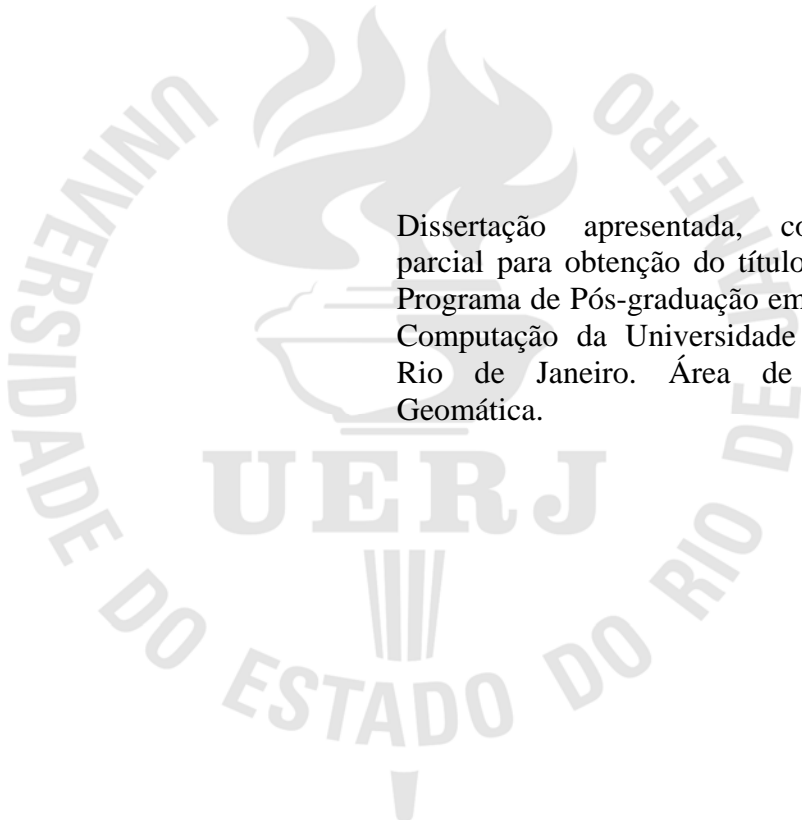
**SAQUA - sistema baseado em software livre para análise do
comportamento da qualidade das águas fluviais:
aplicado à bacia do rio Paraíba do Sul.**

Rio de Janeiro

2010

Antonio Ignácio Glória

**SAQUA - sistema baseado em software livre para análise do
comportamento da qualidade das águas fluviais:
aplicado à bacia do rio Paraíba do Sul**



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-graduação em Engenharia de Computação da Universidade do Estado do Rio de Janeiro. Área de concentração: Geomática.

Orientador: Prof. Dr. João Araújo Ribeiro

Coorientadora: Prof.^a Dr.^a Luciene Pimentel da Silva

Rio de Janeiro

2010

CATALOGAÇÃO NA FONTE

UERJ / REDE SIRIUS / CTC/B

G562 Glória, Antônio Ignácio.
Saqua - sistema baseado em software livre para análise do comportamento da qualidade das águas fluviais: aplicado à bacia do rio paraíba do sul./ Antônio Ignácio Glória. – 2010.
150 f.

Orientador: João Araujo Ribeiro.
Coorientador: Luciene Pimentel da Silva.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Sistemas de computador - Teses. 2. Água qualidade - Teses. 3. Linguagem de programação - Teses. 4. Engenharia de Computação. I. Ribeiro, João Araújo. II. Silva, Luciene Pimentel da. III. Universidade do Estado do Rio de Janeiro. IV. Título.

CDU 37.018.3

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta dissertação, desde que citada a fonte.

Assinatura

Data

Antonio Ignácio Glória

**SAQUA - sistema baseado em software livre para análise do
comportamento da qualidade das águas fluviais:
aplicado à bacia do rio Paraíba do Sul**

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-graduação em Engenharia de Computação da Universidade do Estado do Rio de Janeiro. Área de concentração: Geomática.

Aprovado em: 17 de março de 2010.

Banca Examinadora:

Prof. Dr. João Araújo Ribeiro (Orientador)
Faculdade de Engenharia - UERJ

Prof.^a Dr.^a Luciene Pimentel da Silva (Coorientadora)
Faculdade de Engenharia - UERJ

Prof. Dr. Orlando Bernardo Filho
Faculdade de Engenharia - UERJ

Prof.^a Dr.^a Rosa Maria Formiga Johnsson
Faculdade de Engenharia - UERJ

Prof.^a Dr.^a Mônica de Aquino Galeano Massera da Hora
Universidade Federal Fluminense - UFF

Rio de Janeiro

2010

DEDICATÓRIA

À família que formei durante este período, À participação e o apoio de minha esposa em todos estes momentos, Ao milagre de um filho em nossas vidas.

AGRADECIMENTOS

Aos orientadores e professores do curso de Geomática da UERJ pelas lições recebidas e a compreensão dos demais profissionais envolvidos.

À Fundação IBGE pelo incentivo aos seus funcionários para o aperfeiçoamento de seus conhecimentos.

RESUMO

GLÓRIA, Antônio Ignácio. *Saqua - sistema baseado em software livre para análise do comportamento da qualidade das águas fluviais: aplicado à bacia do rio paraíba do sul*. 2010. 150 f. Dissertação (Mestrado em Engenharia de Computação) – Faculdade de Engenharia, Universidade Estadual do Rio de Janeiro, Rio de Janeiro, 2010.

Nesta dissertação foi desenvolvido o sistema SAQUA (Sistema para Análise da Qualidade das Águas Fluviais), que permite o acompanhamento dos dados de séries históricas de parâmetros físico-químicos para análise da qualidade de águas fluviais. A alimentação do sistema SAQUA se dá a partir do arquivo tipo texto gerado no Hidroweb, sistema de banco de dados hidrológicos da ANA (Agência Nacional de Águas), disponibilizado na internet. O SAQUA constitui uma interface que permite a análise espaço-temporal de parâmetros de qualidade da água específicos definidos pelo usuário. A interface foi construída utilizando o servidor de mapas Mapserver, as linguagens HTML e PHP, além de consultas SQL e o uso do servidor Web Apache. A utilização de uma linguagem dinâmica como o PHP permitiu usar recursos internos do Mapserver por meio de funções que interagem de forma mais flexível com códigos presentes e futuros, além de interagir com o código HTML. O Sistema apresenta como resultado a representação gráfica da série histórica por parâmetro e, em mapa, a localização das estações “em análise” também definidas pelo usuário, geralmente associadas a uma determinada região hidrográfica. Tanto na representação gráfica da série temporal quanto em mapa, são destacados a partir de código de cores a estação de monitoramento e a observação em que os limites estabelecidos na resolução CONAMA 357/05 não foi atendido. A classe de uso da resolução CONAMA que será usada na análise também pode ser definida pelo usuário. Como caso de estudo e demonstração das funções do SAQUA foi escolhida a bacia hidrográfica do rio Paraíba do Sul, localizada na região hidrográfica Atlântico Sudeste do Brasil. A aplicação do sistema demonstrou ótimos resultados e o potencial da ferramenta computacional como apoio ao planejamento e à gestão dos recursos hídricos. Ressalta-se ainda, que todo o sistema foi desenvolvido a partir de softwares disponibilizados segundo a licença GPL de software livre, ou seja, sem custo na aquisição de licenças, demonstrando o potencial da aplicação destas ferramentas no campo dos recursos hídricos.

Palavras-chave: Software livre. Qualidade das águas. Servidor de mapas. Hidroweb.

ABSTRACT

This dissertation involved the development of SAQUA System (for Fluvial Water Quality Analysis). SAQUA allows graphical analysis of physical-chemical parameters time series for assessing river water quality. SAQUA takes input data from the generated text file in Hidroweb, ANA's (Brazilian Water Agency) hydrological database, available on its Internet site. SAQUA works as an interface that allows the spatio-temporal analysis of user-defined water quality parameters. The interface was built using MapServer, HTML and PHP, SQL queries and using Apache Web server. Applying a dynamic language like PHP allowed the use of internal resources through Mapserver functions that interact more flexibly with present and future codes, besides interacting with the HTML code itself. The system output is a graphic representation of the time series for each parameter and a location map showing stations under analysis, which are user defined, generally associated to a river basin. Both for time series graphical representation as for map format, there is a color code showing stations for which eventually parameters are out of established limits in the Brazilian CONAMA 357/05 resolution. CONAMA resolution use-class which should be applied on the analysis is also user defined. A case study involving the Paraíba do Sul River Basin, southeastern Brazil was taken. The implemented system showed excellent results and potential as a computational tool on supporting water resources planning and management. It is noteworthy that the entire system has been developed from software released under the GPL free software. These also demonstrate the potential of GPL free software to develop tools in the field of water resources.

Keywords: Free software. Water quality. Map Server. Hidroweb.

LISTA DE FIGURAS

Figura 1 - Estações fluviométricas no sistema HIDRO.....	27
Figura 2 - Diagrama geral do Sistema SAQUA	29
Figura 3 - Arquitetura lógica de rede cliente/servidor em 3 camadas	32
Figura 4 - Diagrama ER para o SAQUA.....	38
Figura 5 - Diagrama de fluxo de dados para o SAQUA.....	39
Figura 6 - Diagrama de Transição de Estados ao carregar dados.....	41
Figura 7 - Diagrama de casos de uso do Sistema SAQUA	43
Figura 8 - Diagrama de sequência: consultas de locais, períodos e parâmetros.....	44
Figura 9 - Diagrama de sequência: consultas específicas e carga de dados	45
Figura 10 - Ambiente inicial da interface implementada	46
Figura 11 - Dados em <i>hidrografia5.dbf</i>	50
Figura 12 - Dados em <i>agua_qualidade5.dbf</i>	51
Figura 13 - Detalhe cartográfico do intervalo de coordenadas iniciais e limites definidos.....	53
Figura 14 - Localização de arquivos e pastas no servidor.....	59
Figura 15 - Arquivo que relaciona as pastas para o Servidor Apache.....	62
Figura 16 - Estratégia simples de uso.....	65
Figura 17 - Tela carregando parte dos dados da Estação Queluz.....	68
Figura 18 - Série histórica do parâmetro de Oxigênio Dissolvido pelo Sistema Hidro.	69
Figura 19 - Tela do SAQUA mostrando gráfico da Estação Cataguases	70
Figura 20 - Tela mostrando a ampliação do gráfico da Estação Cataguases.....	71
Figura 21 - Tela do SAQUA com dados da Estação Santo Antonio de Pádua II.....	72
Figura 22 - Tela do SAQUA, em maior escala, da Estação Juiz de Fora-Jusante.....	73
Figura 23 - Imagem de satélite capturada da Estação Santa Branca (dez./2007).....	75
Figura 24 - Imagem da Estação Santa Branca na API do Google Maps (out./2009).....	76

Figura 25 - Imagem de satélite da Estação Santa Branca (out./2009).....	77
Figura 26 - Imagem de uma estação (ANA, 2002, p.23).....	77
Figura 27 - Comandos no PostgreSQL para criar parâmetro de Cromo Total.....	79
Figura 28 - Tela mostrando o final da carga dos dados de todas estações	80
Figura 29 - Tela do SAQUA mostrando dados de parâmetros de Cromo Total	80
Figura 30 - Comandos no PostgreSQL para o parâmetro de Cromo Total	81
Figura 31 - Tela do windows mostrando a exclusão diária de arquivos temporários	90

LISTA DE TABELAS

Tabela 1 - Classes e respectivos usos da água conforme a Resolução CONAMA 357	22
Tabela 2 - Região Hidrográfica e sua correlação com as sub-bacias correspondentes	24
Tabela 3 - Funcionalidades presentes na construção do SAQUA pelos programas livres.....	42
Tabela 4 - Atualizações no arquivo <i>agua_qualidade5.dbf</i>	52
Tabela 5 - Atributos da entidade <i>Alerta</i> , implementados em PostgreSQL.....	55
Tabela 6 - Atributos da entidade <i>Classificação</i> , implementados em PostgreSQL.....	55
Tabela 7 - Atributos da entidade <i>Critério</i> , implementados em PostgreSQL.....	55
Tabela 8 - Atributos da entidade <i>Estação</i> , implementados em PostgreSQL.....	56
Tabela 9 - Atributos da entidade <i>Parâmetro</i> , implementados em PostgreSQL	56
Tabela 10 - Atributos da entidade <i>Qualidade de Água</i> , obtidos em arquivos texto	56
Tabela 11 - Atributos da entidade <i>Situação na Estação</i> , em formato dbase.....	57
Tabela 12 - Classes de tipo de rede segundo Sistema Hidro.....	58
Tabela 13 - Conversão das coordenadas geográficas para coordenadas em decimais	66
Tabela 14 - Limites de parâmetros cadastrados no SAQUA segundo o CONAMA.....	78
Tabela 15 - Parâmetros disponíveis nos sistemas HIDRO e SAQUA	87

LISTA DE ABREVIATURAS E SIGLAS

ANA	Agência Nacional de Águas
API	Interface de programação de aplicativos (Application Program Interface)
CONAMA	Comissão Nacional de Meio Ambiente
CNRH	Conselho Nacional de Recursos Hídricos
DNAEE	Departamento Nacional de Águas e Energia Elétrica
DNS	Servidor do domínio de nomes (Domain Name Server)
GIS	O mesmo que SIG (Geografic Information System)
GUI	Interface Gráficas do Usuário (Graphical User Interface)
HTML	Linguagem para manipular “hipertexto” (HiperText Markup Language)
HIDROWEB	Sistemas de Informações Hidrológicas disponibilizado no sitio da ANA
IBGE	Fundação Instituto Brasileiro de Geografia e Estatística
IP	Protocolo de Internet (Internet Protocol)
PHP	Linguagem ferramental para páginas pessoais (Personal Home Page Tools)
SAQUA	Sistema Baseado em Software Livre para Análise do Comportamento da Qualidade das Águas Fluviais: aplicado à bacia do Rio Paraíba do Sul
SGBD	Sistema gerenciador de banco de dados
SIG	Sistema de Informações Geográficas
SQL	Linguagem estruturada de consultas (Structured Query Language)
UML	Linguagem de Modelagem Unificada (Unified Modeling Language)
URL	Localizador Uniformizado de Recursos (Universal Resource Locator), possui endereço de um recurso
Web	Rede de alcance mundial, também conhecida como WWW (World Wide Web)

SUMÁRIO

	INTRODUÇÃO	13
1	MATERIAIS E MÉTODOS	19
1.2	Base legal para a gestão de recursos hídricos	19
1.2	Sistema HIDROWEB	23
1.3	Aplicação Hidro	25
1.3.1	<u>Feições de Mapas</u>	27
1.4	Proposta do Sistema SAQUA	29
1.4.1	<u>Software Livre e software gratuito</u>	30
1.4.2	<u>Arquitetura cliente/servidor</u>	31
1.4.3	<u>Interfaces GUI e API</u>	33
1.4.4	<u>Servidor Web</u>	34
1.4.5	<u>Servidor de Mapas</u>	34
1.4.6	<u>Servidor de Banco de Dados</u>	35
1.5	Diagrama Entidade Relacionamento do SAQUA a partir do HIDROWEB	36
1.6	Diagrama de Transição de Estados HIDROWEB – SAQUA	40
1.7	Conjunto de Programas Usados	42
2	APLICAÇÃO DO SISTEMA SAQUA	46
2.1	Aspectos gerais da implementação	46
2.2	Os alertas nas feições	48
2.3	Atributos e campos utilizados pelo SAQUA	54
2.4	Organização de pastas e arquivos	58
2.5	Instalações e configurações para o SAQUA	61
2.6	A API do <i>Google Maps</i>	63
3	ANÁLISE DOS RESULTADOS	67
3.1	A carga das séries históricas no SAQUA	67
3.2	Dados exibidos no SAQUA	69
3.3	Imagens de satélite das estações	74
3.4	Cadastro de demais parâmetros	78
	CONCLUSÕES E SUGESTÕES	82
	REFERÊNCIAS	84
	APÊNDICE A – Lista de sítios utilizados neste trabalho	86

APÊNDICE B – Relação de parâmetros de qualidade de água.....	87
APÊNDICE C – Remoção periódica de arquivos temporários.....	90
APÊNDICE D – Códigos	91

INTRODUÇÃO

Problemática e Justificativa

Várias das atividades envolvidas no desenvolvimento humano apresentam impactos ambientais associados. Entre os recursos naturais impactados, destaca-se a questão da qualidade das águas fluviáteis. O Sistema HIDROWEB é um sistema de banco de dados baseado na *web* no qual são disponibilizados os dados hidrológicos de toda a rede de monitoramento brasileira. A gestão do sistema é feita pela ANA (Agência Nacional de Águas). O acesso ao sistema se dá a partir do sítio da ANA¹ ou diretamente pelo endereço de seu portal: “<http://hidroweb.ana.gov.br>”.

No Sistema HIDROWEB, entre outros, são armazenados dados pluviométricos e fluviométricos (relacionados ao estado dos cursos d'água e rios brasileiros). Entre os dados fluviométricos que caracterizam os escoamentos, estão os níveis d'água, perfil de seção transversal, medições de descarga, séries de vazões; além de parâmetros físico-químicos, Tucci (1988, p. 543-544) que caracterizam a qualidade das águas. No Brasil são adotados pela ANA o monitoramento de 135 parâmetros de qualidade das águas, listados no APÊNDICE B.

Ao todo no Sistema HIDROWEB estão armazenadas atualmente as informações relativas a 9840 postos fluviométricos, com diferentes códigos cadastrados, distribuídos nas 12 regiões hidrográficas brasileiras, definidas na Resolução nº 32, de 15 de outubro de 2003, do CNRH (Conselho Nacional de Recursos Hídricos). No entanto, observa-se que nem todos os postos possuem monitoramento de qualidade das águas e ainda, em muitos deles observam-se falhas nas séries temporais, como também a ausência de alguns dos parâmetros em determinadas campanhas de monitoramento². A ANA busca pelo menos quatro campanhas anuais em todos os postos da rede nacional. Em Magalhães Jr. (2000, p. 113-115) é apresentado um retrato do monitoramento das águas no Brasil. Estes dados são fundamentais para os estudos da Engenharia de Recursos

¹ <http://www.ana.gov.br>

² A coleta sistemática de dados, "*in situ*", é realizada através de campanhas de campo anuais nos principais rios, como descrito no projeto HIBAM disponível em: <http://www.ana.gov.br/hibam/metodologia.asp>

Hídricos e dimensionamento de obras hidráulicas. Destaca-se ainda nesse contexto a gestão de recursos hídricos.

A Lei 9433/97 define o Sistema (organizacional) e a Política Nacional para o Gerenciamento dos Recursos Hídricos, Brasil (1997). No contexto da Política Nacional de Recursos Hídricos, em seus fundamentos, é previsto que a bacia hidrográfica é a unidade territorial para implementação da Política Nacional de Recursos Hídricos e atuação do Sistema Nacional de Gerenciamento. Nos fundamentos é previsto ainda que a gestão dos recursos hídricos deve ser descentralizada e contar com a participação do poder público, dos usuários e das comunidades.

A Resolução CONAMA 357/05 dispõe sobre a classificação e diretrizes ambientais para o enquadramento dos corpos de águas superficiais, bem como estabelece as condições e padrões de lançamento de efluentes, revogando a resolução CONAMA 20/86.

O enquadramento dos corpos de água em classes, pela Lei 3239/99 do Estado do Rio de Janeiro no art. 16 utilizando como base a legislação ambiental, segundo os usos preponderantes destes, visa assegurar às águas qualidade compatível com os usos prioritários destinados, a diminuir custos de combate à poluição através de ações preventivas permanentes e a estabelecer metas de qualidade de água. Esta lei veio complementar o art. 9 da lei federal 9433/97.

A classificação das águas, essencial à proteção da qualidade das mesmas, deve ser avaliada por condições e padrões específicos e definida em função dos usos preponderantes atuais e futuros. Os corpos de água classificam-se em 13 (treze) classes, sendo 4 (quatro) classes destinadas às águas doces.

Também é levada em consideração na legislação a “necessidade de se criar instrumentos para avaliar a evolução da qualidade das águas, em relação às classes estabelecidas no enquadramento, de forma a facilitar a fixação e controle de metas visando atingir gradativamente os objetivos propostos” (CONAMA, 2005).

A Agência Nacional de Águas, ANA, integra o Sistema Nacional de Gerenciamento de Recursos Hídricos, Brasil (1997), e possui um conjunto de dados e informações de estações fluviométricas e pluviométricas (Rede Hidrometeorológica Nacional). Estas informações são disponibilizadas por meio do Sistema de Informações Hidrológicas, o Sistema HIDROWEB, componente do Sistema Nacional de Informações sobre Recursos Hídricos, SNIRH.

No passado recente, observam-se iniciativas no sentido de operacionalizar e automatizar a aplicação dos dados armazenados no Sistema HIDRO em benefício da gestão dos recursos hídricos³, inclusive, da qualidade das águas. Observa-se, no entanto, um potencial para desenvolvimento de ferramentas computacionais facilitadoras para análise dos dados, assim como na determinação de parâmetros e valores de referência determinados a partir das séries temporais observadas. Uma ferramenta que utiliza dados do sistema HIDRO, por exemplo, para estes fins é o Sistema Generalizado para apoio à Decisão na Gestão de Recursos Hídricos em seu módulo “SisInfo” desenvolvido no Projeto Macacu (DA HORA; MARQUES, 2010, p.9).

Por outro lado, existe hoje a facilidade de encontrar ferramentas computacionais de uso livre, por software livre⁴ ou software gratuito⁵, que dispõem de um grande número de recursos visuais e permitem construir diferentes tipos de interfaces, inclusive utilizando diferentes composições, como por exemplo, uma arquitetura de redes cliente-servidor.

As possibilidades de aplicação destas ferramentas desenvolvidas em software livre são muito amplas, contemplando desde as publicações *web* com manipulação de mapas até consultas a grandes bancos de dados empresariais efetuadas com boa confiabilidade.

Nesta dissertação buscou-se demonstrar a possibilidade de aplicação de ferramentas computacionais de uso livre no desenvolvimento de sistemas de análise de dados hidrológicos de forma espaço-temporal, levando em conta a adequação das mesmas e os serviços oferecidos.

Desta forma, foi desenvolvido o sistema SAQUA (Sistema baseado em software livre para análise do comportamento da qualidade das águas fluviais), que acoplado às informações geradas e disponíveis no Sistema HIDROWEB possibilita a análise dos parâmetros e das séries temporais dos parâmetros físico-químicos de qualidade das águas. Para demonstração, o SAQUA foi aplicado à bacia do Rio Paraíba do Sul.

³ <http://www.hidro.ufrj.br> e <http://www.agevap.org.br>

⁴ programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído.

⁵ não implica no pagamento de licenças de uso ou patentes, porém sem as obrigatoriedades do software livre.

Objetivos Gerais e Específicos

O propósito geral desse trabalho é a construção de um sistema baseado em software livre para a análise do comportamento da qualidade das águas fluviais, o SAQUA.

Através do SAQUA pretende-se:

- Facilitar a análise da evolução histórica de parâmetros físico-químicos de qualidade de água;
- Exibir, armazenar e manipular dados geo-referenciados a fim de obter melhor conhecimento regional;
- Fornecer uma ferramenta que, através da análise temporal e do conhecimento regional, contribua na gestão de recursos hídricos.

Natureza da Metodologia

O SAQUA foi criado a partir de dados fornecidos pelo Sistema HIDROWEB, utilizando técnicas de modelagem em uma arquitetura de rede cliente/servidor. Além disso foram desenvolvidos programas para construir páginas dinâmicas em ambiente *web* que incorporem uma interface de comunicação com servidor de mapas e com o servidor de banco de dados.

O Sistema HIDROWEB é um ambiente *web* disponibilizado pela ANA, e necessita apenas de uma interface *web* para prover o acesso aos dados e informações hidrometeorológicas da aplicação HIDRO.

A aplicação HIDRO é uma aplicação cliente/servidor projetada especificamente para o ambiente gráfico do Windows. Permite a gerência da base de dados hidrometeorológica em um banco de dados relacional e o cálculo de funções hidrometeorológicas básicas, além da visualização dos dados.

A aplicação HIDRO é fornecida com um manual do usuário e um dicionário de dados em seu apêndice. Isto facilitou a modelagem na qual foi utilizado um Diagrama Entidade Relacionamento (DER), que é um modelo conceitual dos dados do mundo real, e ajuda a evitar redundâncias na implementação do projeto.

A arquitetura de rede cliente/servidor de 3 camadas permite separar as tarefas do Cliente com as do Servidor de Aplicativos e as do Servidor de Banco de Dados. Geralmente isto reduz o custo de manutenção, pois as mudanças nas camadas de aplicação e dados geralmente não necessitam de novas instalações no Cliente.

Nesta arquitetura de rede, trabalham de forma integrada no mesmo equipamento o Servidor *Web*, que fornece os recursos de navegação e disponibiliza a página *web* ao Cliente, e o Servidor de Mapas, que permite a interação com os mapas dentro da página *web*. Estes dois servidores atuam como um Servidor de Aplicativos.

O Servidor de Banco de Dados opera de forma independente, podendo ou não estar no mesmo equipamento do servidor de aplicativos. Este servidor utiliza um Sistema Gerenciador de Banco de Dados, cujo objetivo central é o de gerenciar o acesso e a correta manutenção dos dados armazenados.

O tratamento de imagens para a localização geo-referenciada de alertas não faz parte da solução proposta. Da mesma forma, os programas utilizados não possuem esta funcionalidade, uma vez que os dados geográficos já estão definidos pela localização das estações e as imagens de satélite utilizadas pelo SAQUA apenas mostram as proximidades das estações.

Organização da Dissertação

No segundo capítulo é feito um levantamento da base legal para a gestão de recursos hídricos e é apresentado o Sistema HIDROWEB enfocando suas operações e os atributos que foram aplicados ao SAQUA através da modelagem de dados. Também são descritas as funcionalidades do SAQUA e as ferramentas de uso livre utilizadas dentro da arquitetura proposta.

O terceiro capítulo mostra a descrição, implementação, e utilização do SAQUA, explicando vários aspectos envolvidos no seu código como, por exemplo, a adaptação da região escolhida, o uso da identificação cartográfica no tratamento da informação e a criação de arquivos temporários para solução de algumas situações diferenciadas. Neste capítulo também é feito o estudo de caso para a bacia do Rio Paraíba do Sul, com os resultados obtidos no SAQUA.

O quarto capítulo mostra a análise dos resultados, buscando comparar dados obtidos no SAQUA com a realidade existente por eventos conhecidos.

O quinto capítulo finaliza o estudo, apresentando conclusões e sugestões para trabalhos futuros.

1 MATERIAIS E MÉTODOS

1.1 Base legal para a gestão de recursos hídricos

O Conselho Nacional do Meio Ambiente (CONAMA) é um órgão consultivo e deliberativo que possui a finalidade de assessorar, estudar e propor ao Conselho de Governo, diretrizes de políticas governamentais para o meio ambiente e os recursos naturais, assim como deliberar, no âmbito de sua competência, sobre normas e padrões compatíveis com o meio ambiente ecologicamente equilibrado e essencial à sadia qualidade de vida, competências que lhe são aferidas pela Lei número 6938, de 31 de agosto de 1981 (BRASIL,1981).

A política nacional da gestão de recursos hídricos, em âmbito nacional, é regida pela Lei número 9.433, de 8 de janeiro de 1997, Brasil (1997), que define seus fundamentos, objetivos, diretrizes, instrumentos e organização geral através de seções.

Quanto aos fundamentos:

- A água é um recurso natural limitado, um bem público, dotado de valor econômico e seu uso se torna prioritário para o consumo humano e a dessedentação dos animais, em situações em que ocorra a sua escassez;
- A bacia hidrográfica é a unidade territorial para implementação desta política e atuação do Sistema Nacional de Gerenciamento de Recursos Hídricos;
- A gestão de recursos hídricos deve ser descentralizada, ter a participação do Poder Público, dos usuários e das comunidades e proporcionar sempre o uso múltiplo das águas.

Quanto aos objetivos desta política, são os seguintes:

- Assegurar às gerações, atual e às futuras, a necessária disponibilidade de água em padrões de qualidade adequado aos respectivos usos;
- Com vistas ao desenvolvimento sustentável, utilizar de forma racional e integrada os recursos hídricos, incluindo o transporte aquaviário;
- Prevenir e defender contra eventos hidrológicos críticos de origem natural ou decorrente do uso inadequado dos recursos naturais.

Quanto às diretrizes gerais de ação é previsto que:

- Não haverá dissociação dos aspectos de quantidade e qualidade na gestão sistemática;
- Adequação da gestão de recursos hídricos às diversas regiões do Brasil, respeitando suas peculiaridades;
- Integração da gestão ambiental com a gestão de recursos hídricos;
- Integração da gestão das bacias hidrográficas com a gestões das zonas costeiras e a dos sistemas estuarinos;
- Articulação do planejamento de recursos hídricos com os planejamentos regional, estadual e nacional e também com o dos usuários;
- Articulação da gestão destes recursos com a gestão do uso do solo,
- Articulação da União com os Estados visando o gerenciamento dos recursos hídricos de interesse comum.

São consideradas instrumentos da Política Nacional de Recursos Hídricos:

- Os Planos de Recursos Hídricos;
- O enquadramento dos corpos de água em classes;
- A outorga dos direitos de uso de recursos hídricos;
- A cobrança pelo uso de recursos hídricos;
- A compensação a municípios (vetado);
- O Sistema de Informações sobre Recursos Hídricos.

A gestão sistemática dos recursos hídricos, sem dissociação dos aspectos de quantidade e qualidade se destaca entre as diretrizes gerais de ação, sendo que as metas de racionalização de uso, aumento de quantidade e melhoria da qualidade dos recursos hídricos disponíveis fazem parte do conteúdo mínimo dos planos de recursos hídricos.

Integram o Sistema Nacional de Gerenciamento de Recursos Hídricos, conforme disposto no artigo 33 da lei número 9984, de 17 de julho de 2000 (BRASIL, 2000):

- O Conselho Nacional de Recursos Hídricos;
- A Agência Nacional de Águas;
- Os Conselhos de Recursos Hídricos dos Estados e do Distrito Federal;
- Os Comitês de Bacia Hidrográfica;
- Os órgãos dos poderes públicos federal, estaduais, do Distrito Federal e municipais cujas competências se relacionem com a gestão de recursos hídricos;

- As Agências de Água.

O Conselho Nacional de Recursos Hídricos é gerido por um presidente, que é o ministro titular do Ministério de Meio Ambiente, dos Recursos Hídricos e da Amazônia Legal, e por um secretário executivo que será o titular do órgão integrante da estrutura do mesmo ministério, responsável pela gestão dos recursos hídricos.

Por esta mesma lei foi criada a Agência Nacional de Águas - ANA, autarquia sob regime especial, com autonomia administrativa e financeira, vinculada ao Ministério do Meio Ambiente, com a finalidade de implementar, em sua esfera de atribuições, a Política Nacional de Recursos Hídricos. Sua atuação obedece aos fundamentos, objetivos, diretrizes e instrumentos da Política Nacional de Recursos Hídricos e será desenvolvida em articulação com órgãos e entidades públicas e privadas integrantes do Sistema Nacional de Gerenciamento de Recursos Hídricos.

A ANA disponibiliza em seu sítio leis estaduais⁶, entre elas a Lei número 3.239 de 2 de agosto de 1999, RJ (1999), que institui a Política Estadual de Recursos Hídricos e cria o Sistema Estadual de Gerenciamento de Recursos Hídricos no Estado do Rio de Janeiro, que segue as políticas adotadas especificando características locais.

O enquadramento dos corpos hídricos nas respectivas classes de uso, segundo a Lei 3239/99 do Estado do Rio de Janeiro no art. 17, serão feitos, na forma da lei, pelos Comitês de Bacia Hidrográfica (CBH's) e homologados pelo Conselho Estadual de Recursos Hídricos (CERHI) após avaliação técnica pelo órgão competente do Poder Executivo.

Em todo o território nacional o enquadramento é proposto, pela agência local de água, ao respectivo Comitê de Bacia Hidrográfica para encaminhamento ao CNRH ou conselhos estaduais de recursos hídricos, de acordo com o domínio destes, conforme o art. 44 da Lei 9433/97.

A resolução CONAMA 357, de 15 de março de 2005, revoga a resolução CONAMA 20 e define no capítulo 5 as diretrizes ambientais para o enquadramento dos corpos hídricos, CONAMA (2005, p. 23). Nestas diretrizes é definido que este enquadramento será definido pelos usos preponderantes mais restritivos da água, conforme mostrado na **Tabela 1**.

⁶ <http://www.ana.gov.br/Institucional/ASPAR/legislacaoEstadosDF.asp>

Tabela 1 - Classes e respectivos usos da água conforme a Resolução CONAMA 357

CLASSES	USOS
ESPECIAL	<ul style="list-style-type: none"> - abastecimento para consumo humano, com desinfecção; - preservação do equilíbrio natural das comunidades aquáticas; - preservação dos ambientes aquáticos em unidades de conservação de proteção integral.
1	<ul style="list-style-type: none"> - abastecimento para consumo humano, após tratamento simplificado; - proteção das comunidades aquáticas; - recreação de contato primário, tais como natação, esqui aquático e mergulho, conforme Resolução CONAMA n. 274, de 2000; - irrigação de hortaliças que são consumidas cruas e de frutas que se desenvolvam rentes ao solo e que sejam ingeridas cruas sem remoção de película; e - proteção das comunidades aquáticas em Terras Indígenas.
ÁGUAS DOCES 2	<ul style="list-style-type: none"> - abastecimento para consumo humano, após tratamento convencional; - proteção das comunidades aquáticas; - recreação de contato primário, tais como natação, esqui aquático e mergulho, conforme Resolução CONAMA n. 274, de 2000; - irrigação de hortaliças, plantas frutíferas e de parques, jardins, campos de esporte e lazer, com os quais o público possa vir a ter contato direto; e - aqüicultura e à atividade de pesca
3	<ul style="list-style-type: none"> - abastecimento para consumo humano, após tratamento convencional ou avançado; - irrigação de culturas arbóreas, cerealíferas e forrageiras; - pesca amadora; - recreação de contato secundário; e - dessedentação de animais
4	<ul style="list-style-type: none"> - navegação; - harmonia paisagística.

Panorama do Enquadramento dos Corpos d'Água

⁷ http://www.ana.gov.br/acoesadministrativas/cdoc/Catalogo_Publicacoes/5_volume_5_ANA.pdf

A resolução CONAMA 357 prevê que no desacordo com os usos pretendidos serão estabelecidas metas obrigatórias de melhoria da qualidade de água para efetivação dos enquadramentos, excetuando parâmetros que excedam limites devido às condições naturais.

Também são definidas que as ações de gestão referentes ao uso dos recursos hídricos ou referentes à gestão ambiental deverão basear-se nas metas progressivas intermediárias e final aprovadas pelo órgão competente. Todas estas metas devem ser atingidas em regime de vazão de referência para as quais deverão ser elaborados estudos específicos sobre a dispersão e assimilação de poluentes em meio hídrico.

Em corpos de água intermitentes ou que apresentam diferença sazonal significativa, as metas obrigatórias poderão variar ao longo do ano, enquanto que nos utilizados para o abastecimento da população, o enquadramento e o licenciamento ambiental preservarão as condições de consumo.

Através da análise dos valores de parâmetros específicos da qualidade das águas, é possível identificar impactos ambientais, respeitando os limites pré-estabelecidos de aceitação nas respectivas classes de uso da água.

1.2 Sistema **HIDROWEB**

O Sistema de Informações Hidrológicas da Agência Nacional de Águas, denominado Sistema HIDROWEB, é um sistema de banco de dados baseado na *web* que está disponibilizado no sítio da ANA⁸.

As estações hidrológicas foram codificadas de acordo com sua localização pela divisão hidrográfica estabelecida pelo antigo Departamento Nacional de Águas e Energia Elétrica (DNAEE). Nesta codificação o País é dividido em 9 (nove) bacias ou regiões hidrográficas, cada uma delas divididas em 10 (dez) sub-bacias, sendo que a Bacia 9 corresponde à área de drenagem de qualquer bacia hidrográfica da América do Sul, sem a interferência da rede potamográfica brasileira.

⁸ <http://hidroweb.ana.gov.br/>

Porém o Conselho Nacional de Recursos Hídricos (CNRH), por meio da resolução número 32, de 15 de outubro de 2003, instituiu a Divisão Hidrográfica Nacional em 12 (doze) regiões hidrográficas, a fim de orientar, fundamentar e implementar o Plano Nacional de Recursos Hídricos. O campo descrito com “RH” na **Tabela 2** correlaciona estas regiões hidrográficas.

As feições das sub-bacias são fornecidas no Sistema HIDROWEB em 8 arquivos compactados, de acordo com a codificação das estações fluviométricas, cujo nome é iniciado pelo prefixo bacia seguido do código das bacias definidas pelo DNAEE, correspondendo à rede potamográfica nacional. A junção destas sub-bacias compõem as regiões hidrográficas, conforme mostra a **Tabela 2**.

Tabela 2 - Região Hidrográfica e sua correlação com as sub-bacias correspondentes

Número Tabela (RH)	Região Hidrográfica (Resolução CNRH n° 32/2003)	Sub Bacias Correspondentes Utilizadas na Codificação das Estações (DNAEE)
1	AMAZÔNICA	10 A 19 E 30
2	TOCANTINS	20 A 29
3	ATLÂNTICO NORDESTE OCIDENTAL	31 A 33
4	DO PARNAÍBA	34
5	ATLÂNTICO NORDESTE ORIENTAL	35 A 39
6	DO SÃO FRANCISCO	40 A 49
7	ATLÂNTICO LESTE	50 A 55
8	ATLÂNTICO SUDESTE	56 A 59
9	PARANÁ	60 A 65
10	PARAGUAI	66 A 67
11	URUGUAI	70 A 79
12	ATLÂNTICO SUL	80 A 88

Inventário das estações fluviométricas⁹

Além das feições das sub-bacias, o Sistema HIDROWEB disponibiliza dados referentes às estações fluviométricas e pluviométricas. As medições destas estações são fornecidas através de séries históricas que podem ser consultadas através da aplicação HIDRO.

⁹ <http://www.ana.gov.br/GestaoRecHidricos/InfoHidrologicas/Inventarios/default.asp?acao=flu>

Juntamente à aplicação HIDRO, a ANA fornece um manual do usuário, com sua operacionalização e, inclusive, o seu dicionário de dados, ANA (2002). As atualizações da aplicação HIDRO, tanto em versão do aplicativo como no cadastro das estações e séries históricas, são disponibilizadas no Sistema HIDROWEB.

Os dicionários de dados consistem em “repositórios de definições de todos os elementos e conceitos utilizados e manipulados pela organização e respectivos sistemas de informação e que incluem entre outros os dados, ficheiros, processos e entidades” (SILVA, 2001, p.79).

A partir do conteúdo do dicionário de dados da aplicação HIDRO, iniciou-se uma pesquisa em relação ao que deveria ser utilizado, incorporando informações relevantes encontradas neste dicionário de dados ao banco de dados do SAQUA, utilizando um SGBD, ou Sistema Gerenciador de Banco de Dados.

Os sistemas de banco de dados relacional padrão “armazenam informações sobre bancos de dados, tabelas, colunas etc., no que é comumente conhecido por catálogos do sistema (Alguns sistemas chamam de dicionário de dados). Os catálogos são vistos pelo usuário como qualquer outra tabela, mas o SGBD armazena suas informações internas nestas tabelas.” (POSTGRESQL, 2007, p.480).

A partir do estudo da aplicação Hidro, inclusive com o dicionário de dados fornecido, e as feições de mapas obtidas no Sistema HIDROWEB, teve início a modelagem do sistema SAQUA.

1.3 Aplicação Hidro

A aplicação HIDRO é um programa gratuito fornecido pela ANA, do tipo cliente/servidor¹⁰, que requer um microcomputador com poucos recursos (166 MHz de CPU, 64Mb de RAM e 20Mb de disco), projetado para o ambiente gráfico Windows 32 bits (95/98/Me/NT4/2000) e superiores.

Os principais objetivos da aplicação HIDRO, junto à base de dados hidrometeorológica, centralizada no Sistema HIDROWEB, são o seu gerenciamento, a inclusão de dados por entidades

¹⁰ Arquitetura onde a estação cliente busca informações do equipamento servidor

que operam a rede hidrometeorológica, o cálculo de funções hidrometeorológicas básicas e a visualização de dados (gráficos, imagens etc.).

Sua origem vem desde o início da década de 70, quando o DNAEE promoveu o desenvolvimento de um sistema de banco de dados destinado a gerenciar as informações coletadas em toda a rede hidrometeorológica nacional. No final da década de 80, promoveu o desenvolvimento de um novo sistema, dessa vez para microcomputadores da linha PC/MS-DOS.

No final de 1998, a Superintendência de Estudos e Informações Hidrológicas da ANEEL (Agência Nacional de Energia Elétrica, que herdou atribuições do antigo DNAEE) constatou a necessidade de promover o desenvolvimento de um novo sistema. Iniciou-se então o desenvolvimento do sistema Hidro e, no final de 2001, a atribuição de manter e desenvolver novas versões deste sistema passou a ser da então criada Agência Nacional de Águas.

O HIDRO é uma aplicação de banco de dados capaz de acessar tanto um banco de dados local quanto remoto. No caso do acesso local utiliza-se o banco de dados *Access*, enquanto que no acesso remoto utiliza-se o Servidor de banco de dados SQL ou Oracle, porém este acesso é efetuado por organizações ou usuários externos autorizados.

Neste trabalho foi feito acesso local, obtendo os bancos de dados da Internet, através do Sistema HIDROWEB. Após instalar, executa-se o aplicativo HIDRO, cuja tela de autenticação permite acesso ao usuário “hidro” e senha “master”. Em seguida, deve-se fazer as devidas atualizações de versão e inventário.

Desejando-se, por exemplo, consultar a quantidade de estações fluviométricas cadastradas, abre-se, dentro do Banco de Dados HIDRO.MDB, as pastas de “Registros Permanentes”, em seguida, “Inventário” e seleciona-se finalmente “Estações”. Depois, no menu superior, seleciona-se a opção “Registros” e a seguir “Consulta”.

Na janela de consulta deve-se selecionar restrições para o dado “Tipo de Estação”, neste caso “Fluviométrica”. Após clicar no botão “ok” na janela de consulta tem-se o resultado que expressa a quantidade de 9840 estações listadas, conforme mostra a **Figura 1**.

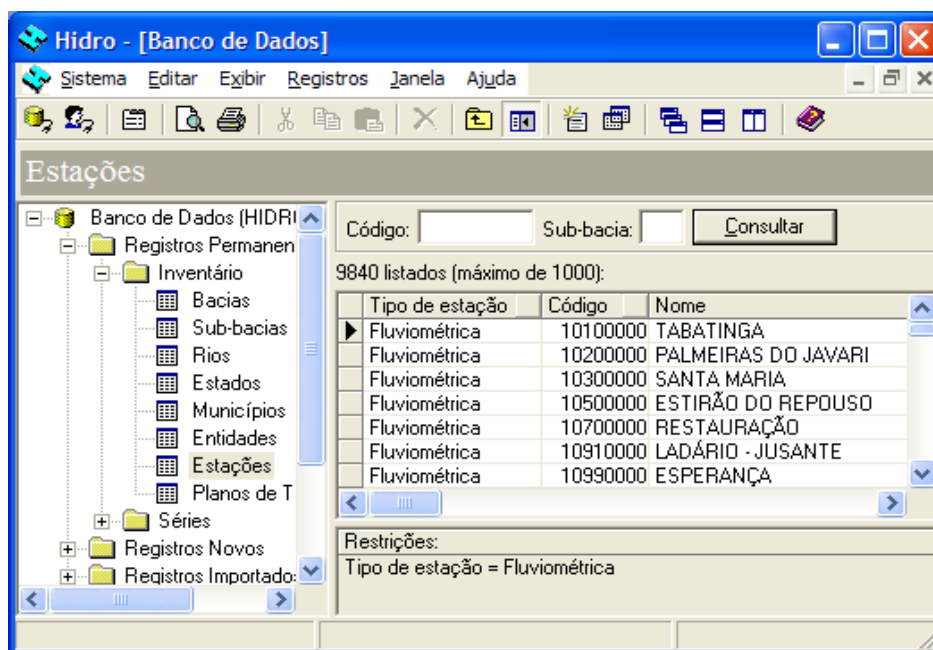


Figura 1 - Estações fluviométricas no sistema HIDRO.

Notou-se, ao longo do trabalho, que algumas destas estações diferem apenas em relação ao código cadastrado, situando-se fisicamente nas mesmas coordenadas geográficas cadastradas e por vezes alterando outros dados cadastrais. Este é o caso quando, eventualmente, o órgão operador decide pela mudança de localização do posto para uma seção próxima à anterior.

No SAQUA foram feitas seleções observando as estações para o caso específico da bacia do Rio Paraíba do Sul, enquanto que para se poder aplicá-lo em outras sub-bacias, ou em mais estações, deve-se conferir as feições existentes e outras considerações sobre o histórico das estações que se pretende monitorar.

1.3.1 Feições de Mapas

As feições de mapas fornecidas pela ANA estão compactadas em 8 (oito) arquivos, um para cada bacia hidrográfica nacional, conforme divisão da DNAEE, excetuando a área de drenagem (bacia 9), no formato ESRI Shapefile, e representam em suas camadas informações diferenciadas nos conjuntos temáticos dos mapas fornecidos abaixo relacionados:

- Estações pluviométricas (ANEEL - 1999) ;
- Estações fluviométricas (ANEEL - 1999) ;
- Estações de qualidade da água (ANEEL - 1999) ;

- Estações telemétricas (ANEEL - 1999) ;
- Usinas hidrelétricas (ANEEL - 1999) ;
- Curvas de nível de 100 metros (ANEEL - 1999) ;
- Hipsometria (ANEEL - 1999) ;
- Hidrografia (ANEEL - 1999) ;
- Sistema viário (ANEEL - 1999) ;
- Malha municipal (IBGE - 1997) ;
- Sedes municipais (IBGE - 1997) ;
- Limites das bacias e sub-bacias hidrográficas (ANEEL - 1999) ;
- Reservas indígenas (FUNAI - 1999) ;
- Solos (USGS - 1992) ;
- Vegetação (USGS - 1992).

As estações de qualidade de água visualizadas foram as cadastradas pela ANEEL em 1999, possuindo um histórico que engloba datas anteriores a uma década, e possivelmente até quatro décadas, levando em consideração o início da implementação do sistema HIDRO.

Estas feições foram utilizadas na construção do SAQUA e obtidas através da opção “Mapas/Baixar Bacias” existentes no Sistema HIDROWEB¹¹. Após obter e descomprimir as feições, foram conservadas apenas as informações referentes às estações de qualidade de água na bacia do Rio Paraíba do Sul, que será, como descrito no item “2.2Os alertas nas feições”, adotada como caso de estudo. Também foram conservadas as feições gerais para composição do mapa hidrográfico nacional.

¹¹ <http://hidroweb.ana.gov.br/>

1.4 Proposta do Sistema SAQUA

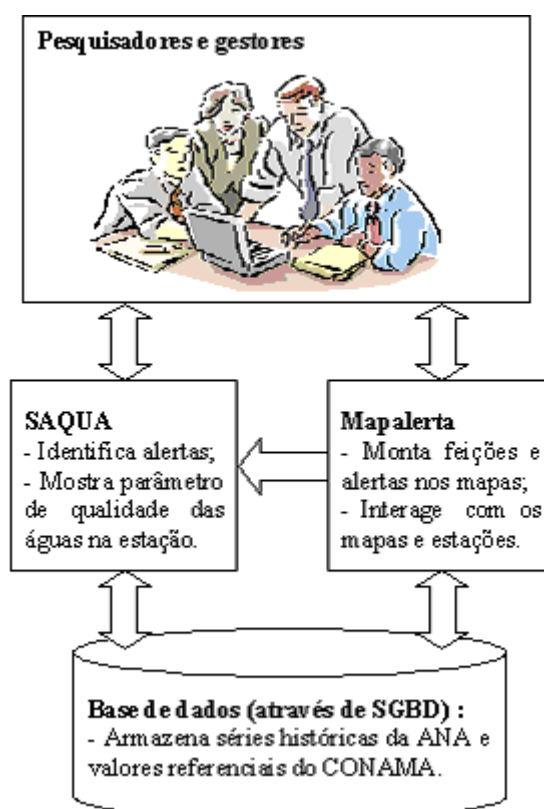


Figura 2 - Diagrama geral do Sistema SAQUA

Através do Sistema SAQUA, conforme proposto no diagrama da **Figura 2**, pesquisadores e gestores podem interagir diretamente com as interfaces do aplicativo, na qual foi prevista uma interface também denominada SAQUA, que organiza os dados da base em alertas de acordo com os respectivos valores, permitindo o acompanhamento da qualidade de água nas estações.

O termo “alerta”¹² foi utilizado neste trabalho por expressar uma situação problema através da ocorrência de eventos, podendo desta forma alertar uma pessoa ou aplicativo de problemas ocorridos, embora não se justifique como um sinal ou aviso para estar vigilante, por não utilizar dados fornecidos em tempo real, ainda possui o sentido de precaução e vigilância, uma vez que estes “alertas” podem ser identificados em determinados períodos históricos.

Pela outra interface, denominada Mapalerta, o pesquisador poderá interagir com o mapa, visualizando-o de acordo com as feições disponíveis como, por exemplo, traçados de rios, e também clicar em alguma estação para obter mais informações sobre a mesma.

¹² <http://michaelis.uol.com.br/moderno/portugues/index.php?lingua=portugues-portugues&palavra=alerta>

Ambas as interfaces interagem com a base de dados na qual são armazenadas além das séries históricas, os valores referenciais do CONAMA para geração de alertas. Ao utilizar os valores de referência para cada parâmetro de qualidade este sistema permite que se acompanhe mensalmente a evolução destes parâmetros expostos através de mapas.

Para a utilização dos mapas foi implementada a interface do aplicativo Mapalerta que, a partir do servidor de mapas Mapserver, irá utilizar os dados disponibilizados pelo SAQUA no SGBD, adotando para isto o banco de dados PostgreSQL.

Quando for solicitada uma consulta pela interface desejada, serão apresentadas informações contidas no banco de dados, utilizando sua localização espaço-temporal com alertas cujos critérios utilizam valores de parâmetros de qualidade de água definidos pelo CONAMA.

1.4.1 Software Livre e software gratuito

Software livre, segundo a definição criada pela *Free Software Foundation*, é qualquer programa de computador que pode ser usado, copiado, estudado, modificado e redistribuído sem nenhuma restrição. Entre eles podemos citar o servidor Apache, a linguagem PHP, o servidor Mapserver e o SGBD PostgreSQL.

O Apache foi o servidor *Web* escolhido por ser a solução de software livre mais utilizada mundialmente e completa para esta finalidade. Já a linguagem PHP, entre outros motivos, foi adotada por permitir integração com o servidor Mapserver e fornecer os recursos de programação necessários.

O Mapserver foi utilizado como o servidor de mapas uma vez que cria mapas de informações espaciais armazenadas em meio digital. Ele suporta tanto dados vetoriais como matriciais (raster), trabalhando com mais de 20 (vinte) diferentes dados de formato vetoriais, incluindo *shapefiles*.

O MapServer pode operar em 2 (dois) modos diferentes: “CGI” e “MapScript”. No modo “CGI”, as funções dele são vistas pelo Servidor *Web* como uma tarefa em “CGI script”. Esta forma é mais fácil e rápida para atualizar e produzir uma aplicação de forma direta (KROPLA, 2005).

O PostgreSQL é derivado do pacote POSTGRES escrito na Universidade da Califórnia em Berkeley. Com mais de uma década de desenvolvimento, o PostgreSQL é atualmente o mais avançado banco de dados de código aberto disponível sendo, portanto, bastante confiável (POSTGRESQL, 2007, p.xv).

O PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional (SGBDOR), isto é, incorpora as tecnologias de orientação a objeto e relacional, e utiliza o modelo cliente-servidor. Além dos recursos de um banco de dados relacional, ele possui um modelo de banco de dados orientado a objetos, onde objetos, classes e herança são suportados diretamente nos esquemas do banco de dados e na linguagem de consulta, além de fornecer uma extensão do modelo de dados com a personalização de tipos de dados e métodos.

Software gratuito, conforme o próprio termo indica, implica no não pagamento de licenças de uso ou patentes. Este tipo de software não possui obrigatoriamente código aberto, nem redistribuição gratuita, ou ainda licenças restritivas quanto ao uso e a redistribuição, entre outros aspectos.

O software gratuito para uso disponibilizado na Internet pela ANA no final de maio de 2006, denominado Sistema Hidro, permite atualizar dados de estações de suas séries históricas, assim como efetuar diversas consultas.

Até bem pouco tempo, quem quisesse implementar um sistema com acesso via *web* a mapas interativos tinha que se preocupar com todos os detalhes desta implementação. Utilizando a API do *Google Maps*, este trabalho é simplificado e o implementador pode apenas chamar suas funções. Com estas funções, em áreas já mapeadas do globo terrestre, obtêm-se facilmente localizações por endereço através de ferramentas de busca, manipulação e navegação.

1.4.2 Arquitetura cliente/servidor

O SAQUA pode ser entendido com um sistema cliente/servidor aplicado em uma arquitetura lógica de rede cliente/servidor em 3 (três) camadas para aplicação *web*. Aqui, *web* é entendido como um sistema de documentos em hipermídia interligados e executados na Internet. A **Figura 3** expressa a arquitetura lógica de rede cliente/servidor em três camadas, tendo a atuação do SAQUA como aplicativo no Servidor de Páginas Web e cliente do SGBD PostgreSQL.

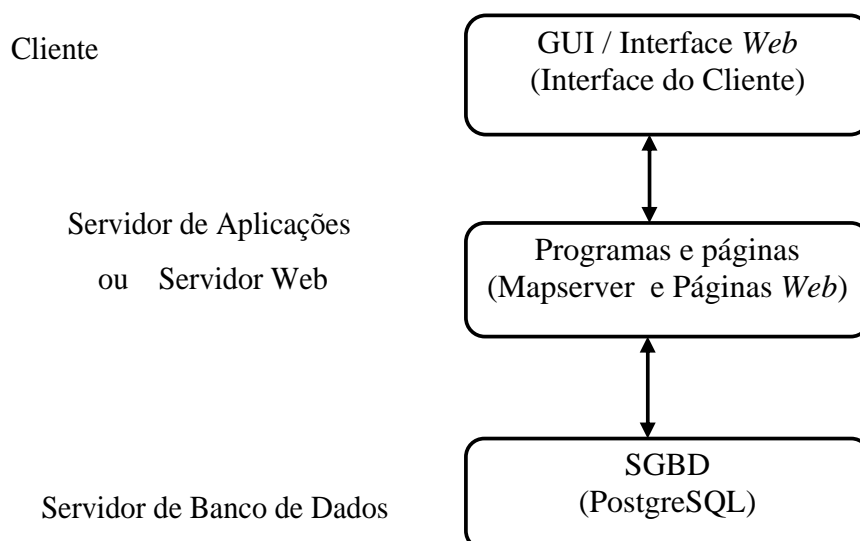


Figura 3 - Arquitetura lógica de rede cliente/servidor em 3 camadas

Nesta arquitetura cliente-servidor (ELMASRI e NAVATHE, 2005, p.31), o cliente é aquele quem busca, do Servidor de Aplicações, informações através de uma GUI (Interface Gráfica do Usuário, do inglês, Graphical User Interface), como por exemplo um navegador *web*. Este Servidor de Aplicações, por sua vez consulta um Servidor de Banco de Dados.

O servidor de Aplicações disponibiliza seus serviços para o cliente, o qual pode visualizar as aplicações de acordo com os produtos que executam. No caso do Servidor *Web*, o servidor de Aplicação pode interpretar o código HTML, de *HiperText Markup Language* (Linguagem para manipular “hipertexto”), e visualizar as figuras incorporadas através do navegador *Web*.

A construção do código HTML pode envolver ou não o uso de linguagens dinâmicas, além de ter figuras incorporadas por meio de bibliotecas gráficas ou de um Servidor de Mapas, que forneça uma interface interativa aos mapas existentes.

No caso específico do SAQUA é utilizado um Servidor de Banco de Dados, pois sua gerência é feita através de uma aplicação que respeita suas características internas de controle e segurança, e existe a possibilidade de acesso aos dados independente do local de execução da aplicação.

O servidor *Web* é o programa responsável pela disponibilização de páginas HTML na Internet, podendo incorporar linguagens de programação que tornam estas páginas dinâmicas, ou seja, páginas que podem alterar seu conteúdo a partir de parâmetros recebidos, requerendo um processamento extra por parte do servidor.

O Software cliente utilizado consiste em uma ferramenta de navegação que também pode ser executada no servidor como, por exemplo, o Mozilla Firefox que é uma solução em software livre ou o Internet Explorer, que é parte integrante do sistema operacional Windows.

No Servidor *Web* foi utilizada a linguagem de programação dinâmica PHP, de *Personal Home Page Tools* (Linguagem ferramental para páginas pessoais), com uso de bibliotecas específicas que permitem a integração desta ferramenta com o Mapserver e consultas à base de dados no PostgreSQL e nas feições do mapa, além de bibliotecas gráficas.

É possível expandir este conceito para uma arquitetura de “n” camadas, uma vez que o Sistema SAQUA permite utilizar imagens da API do *Google Map* incorporando seus objetos através de outro servidor *Web*, o qual por sua vez consulta a base de objetos existente no sítio do Google. Optou-se por afirmar que este acesso pode ser feito implicitamente através do Servidor *Web* através de imagens capturadas.

1.4.3 Interfaces GUI e API.

As interfaces gráficas para usuários, ou GUI de *Graphical User Interfaces* (Interface Gráfica para Usuários), são aquelas que fornecem aos clientes menus, formulários e capacitam gerar interfaces *Web GUI* para um banco de dados, ou capacitam um banco de dados para a *Web* (ELMASRI; NAVATHE, 2005).

A API, de *Application Programming Interface* (Interface de Programação de Aplicativos), é uma interface entre um componente ou sistema de suporte. Uma API é formada pelo conjunto de protótipos de funções, dados, tipos, regras de uso e especificações que permitem um componente servidor tornar disponível seus recursos para os programas clientes (STAA, 2000, p.111).

Ambas interfaces gráficas, CGI ou API, acessadas no lado cliente de uma arquitetura cliente-servidor dependem dos aplicativos processando no lado servidor, sejam através de diferentes serviços, como *Web* e SGBD, ou recursos, como funções e especificações.

1.4.4 Servidor Web

O Servidor *Web*, parte integrante do Servidor de Aplicações, é a ferramenta que envia os códigos em *HTML* para as *GUI*. Este servidor permite utilizar aplicações escritas em diferentes linguagens que fazem chamadas às funções de servidores através de uma API, como o “MapScript” definido como “uma API que segue comandos escritos em diversas linguagens para fazer chamadas às funções do Mapserver”¹³ (KROPLA, 2005, p.155, tradução nossa).

No caso específico do sistema SAQUA foi utilizado o Servidor Apache, versão 2.2.4, com linguagem de programação PHP, versão 5.2.3, sobre o ambiente operacional Windows XP, Home Edition versão 2002. Foram utilizadas, além da biblioteca padrão, a biblioteca “php_mapscript.dll” para uso com o Mapserver, a biblioteca “php_dbase.dll” para exibição e atualização das feições pelo aplicativo SAQUA, e a biblioteca “php_ogr.dll” apenas para exibição dos dados de feições.

Os comandos em linguagem PHP são processados no Servidor *Web* retornando código em HTML, o qual é processado e exibido pelo navegador da estação cliente, e permitem a construção dinâmica de referências, imagens e consultas aos servidores utilizados.

1.4.5 Servidor de Mapas

O Servidor de Mapas é a ferramenta que constrói os mapas digitais, conforme configurado, e exibe as imagens gráficas. Um pequeno incremento na complexidade do código, acima da aplicação “MapScript”, através da interface desenvolvida “possibilitará mostrar dados dinâmicos no mapa e receber atributos associados com estas feições dinâmicas”¹⁴ (KROPLA, 2005, p. 232, tradução nossa).

No modo “MapScript”, a API do Mapserver é acessada por uma linguagem de programação (Perl, Python ou PHP). Esta interface permite uma aplicação flexível com mais recursos.

¹³ O texto em língua estrangeira é: “MapScript is an API (application program interface) that allows applications written in several language to make calls to MapServer functions ...”

¹⁴ O texto em língua estrangeira é: “... will provide the ability to display dynamic data on the map and retrieve attributes associated with these dynamic features.”

Segmentos de códigos em “HTML” são gerados pela linguagem de programação PHP, o que permite associá-los com imagens gráficas geradas pelo Mapserver ou outras imagens PNG (Portable Network Graphics) através da biblioteca gráfica GD, instalada entre outras com o Mapserver para windows, além de incorporar consultas ao SGBD PostgreSQL.

As imagens em formato PNG permitem uma compressão sem perda de qualidade e a retirada do fundo de imagens com o uso do canal alfa, pois o canal alfa permite o uso das transparências que é importante para sobreposição das imagens. Neste trabalho, as imagens, originadas a partir de *shapefiles*, são gravadas em pastas temporárias de acordo com interações nas imagens, pelo Servidor Mapserver, para serem exibidas na interface do Cliente, pelo Servidor *Web*.

Em outras palavras, a interface do Cliente acessa o Servidor *Web*, mas é o Servidor de Mapas que prepara as imagens dos mapas a serem exibidos e que permite a interação com estas imagens.

1.4.6 Servidor de Banco de Dados

O PostgreSQL é um servidor de banco de dados altamente sofisticado, com alta performance, estável e capacitado para lidar com grandes volumes de dados.

Como é típico em aplicações cliente-servidor, o cliente e o servidor podem estar em equipamentos diferentes, se comunicando por meio de uma conexão de rede TCP/IP utilizando uma porta de acesso no servidor, que por padrão é a “5432” para o serviço do banco de dados PostgreSQL (POSTGRESQL, 2007, p.258).

Uma vez preparada a estrutura do banco de dados, pelo modelo lógico definido no diagrama entidade relacionamento (DER), e inserindo seus valores, com o auxílio da linguagem PHP e da base de dados da ANA, o cliente pode consultar estes dados de forma transparente utilizando o SAQUA.

Para acessar este banco de dados foram cadastrados um usuário para gerência do banco chamado “postgres”, e outro usuário chamado “saqua” para efetuar as consultas e inclusões de dados. Ambos efetuam comandos SQL dentro do código em PHP, sendo isto tudo transparente ao usuário pesquisador ou gestor.

Enfim, a solução obtida contempla a arquitetura cliente/servidor de três camadas para a aplicação *web*. Entretanto, para que a re-utilização de códigos se torne menos complexa foi elaborada uma interface que permite exibir alertas, prevendo que as atualizações destes alertas sejam feitas diretamente nos dados da feição.

1.5 Diagrama Entidade Relacionamento do SAQUA a partir do HIDROWEB

A modelagem conceitual, que representa conceitos e características observadas no ambiente, ignorando particularidades da implementação, faz parte da modelagem de dados, que é a atividade de especificação das estruturas de dados, e pode ser representada através do Diagrama Entidade Relacionamento (DER).

O DER também é chamado de diagrama entidade-associação e “(..) apresenta a visão estática do sistema, identificando as entidades sobre as quais interessa guardar informação, bem como o respectivo relacionamento.” (SILVA, 2001, p. 81).

Todos os relacionamentos foram criados com cardinalidade, valor que indica quantas ocorrências de uma Entidade participam no mínimo e no máximo de cada relacionamento, que não alcançam o valor de N para N. Ou seja, o número máximo de ocorrências no lado de uma entidade pode atingir o valor de N ocorrências, entretanto do outro lado do relacionamento não deve ultrapassar o valor máximo de uma ocorrência na outra entidade com que se relaciona.

Quando isto ocorre não é preciso definir mais entidades em um modelo lógico, modelo este que respeita a regras de derivação e de restrição, permitindo que se construa o modelo físico, o qual inclui a análise das características e recursos necessários para armazenamento e manipulação das estruturas de dados, através de atributos e campos como os definidos no item “2.3 Atributos e campos utilizados pelo SAQUA”.

No DER foram representados os critérios estabelecidos, segundo a resolução CONAMA 357/05, adaptando a base de dados do Sistema HIDROWEB, com os parâmetros de qualidade de água, para que sejam identificados alertas nas medições das estações. Este modelo pode ser visto na **Figura 4**.

A resolução do CONAMA permite definir a entidade *Critério*, que busca as informações disponíveis nas entidades *Parâmetro* e *Classificação*, e usa padrões para estabelecer os alertas na entidade *Alerta*. Os parâmetros de qualidade de água pertencentes à entidade *Parâmetro*, seguem os mesmos existentes no Sistema HIDROWEB, conforme exibido no APÊNDICE B.

A entidade *Alerta*, por sua vez, consulta parâmetros de qualidade de água medidos nos arquivos das estações de qualidade de água, através da entidade *Qualidade de Água*. Sendo que cada instância desta entidade gera, para os parâmetros monitorados de cada estação, os seus alertas.

A entidade *Situação na Estação*, foi obtida de arquivos de dados das feições fornecidas pelo Sistema HIDROWEB, entre eles os arquivos *agua_qualidade5.dbf* e *hidrografia5.dbf* que sofreram alterações com a inclusão e preenchimento de campos para identificar o tipo de alerta obtido na estação de qualidade de água e para selecionar as feições de rios presentes na hidrografia da sub-bacia selecionada. Com isto, por meio do atributo *Alerta*, incluído no arquivo *agua_qualidade5.dbf*, podem ser mostrados no mapa os resultados dos alertas solicitados e presentes na entidade *Alerta*.

A entidade *Estação* foi criada para especificar as estações que efetivamente são monitoradas, enquadrando-as em classes quanto ao tipo de rede em que se efetua a coleta, existente na entidade *Classificação*, e referências das medições presente na entidade *Qualidade de Água*.

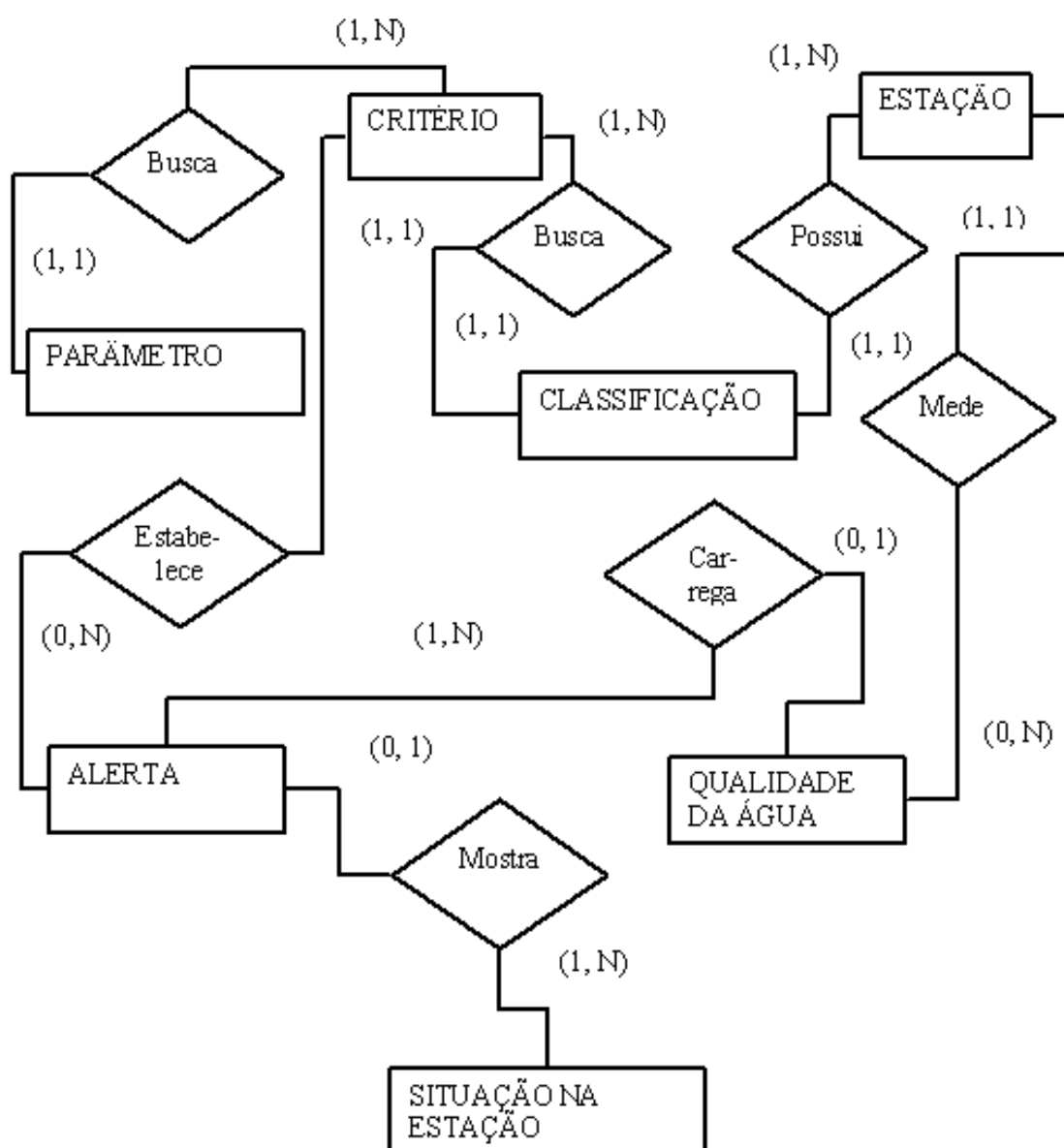


Figura 4 - Diagrama ER para o SAQUA

Após a modelagem de dados pelo DER (COUGO, 1997), utilizando técnicas de análise estruturada (PAGE-JONES, 1988), é mostrado o Diagrama de Fluxo de Dados para o SAQUA na **Figura 5**.

Para a modelagem funcional do SAQUA foi utilizada a linguagem UML. UML é “a abreviação de Linguagem de Modelagem Unificada (Unified Modeling Language), uma notação (principalmente diagramática) para modelagem de sistemas usando conceitos orientados a objetos” (LARMAN, 2000, p.28).

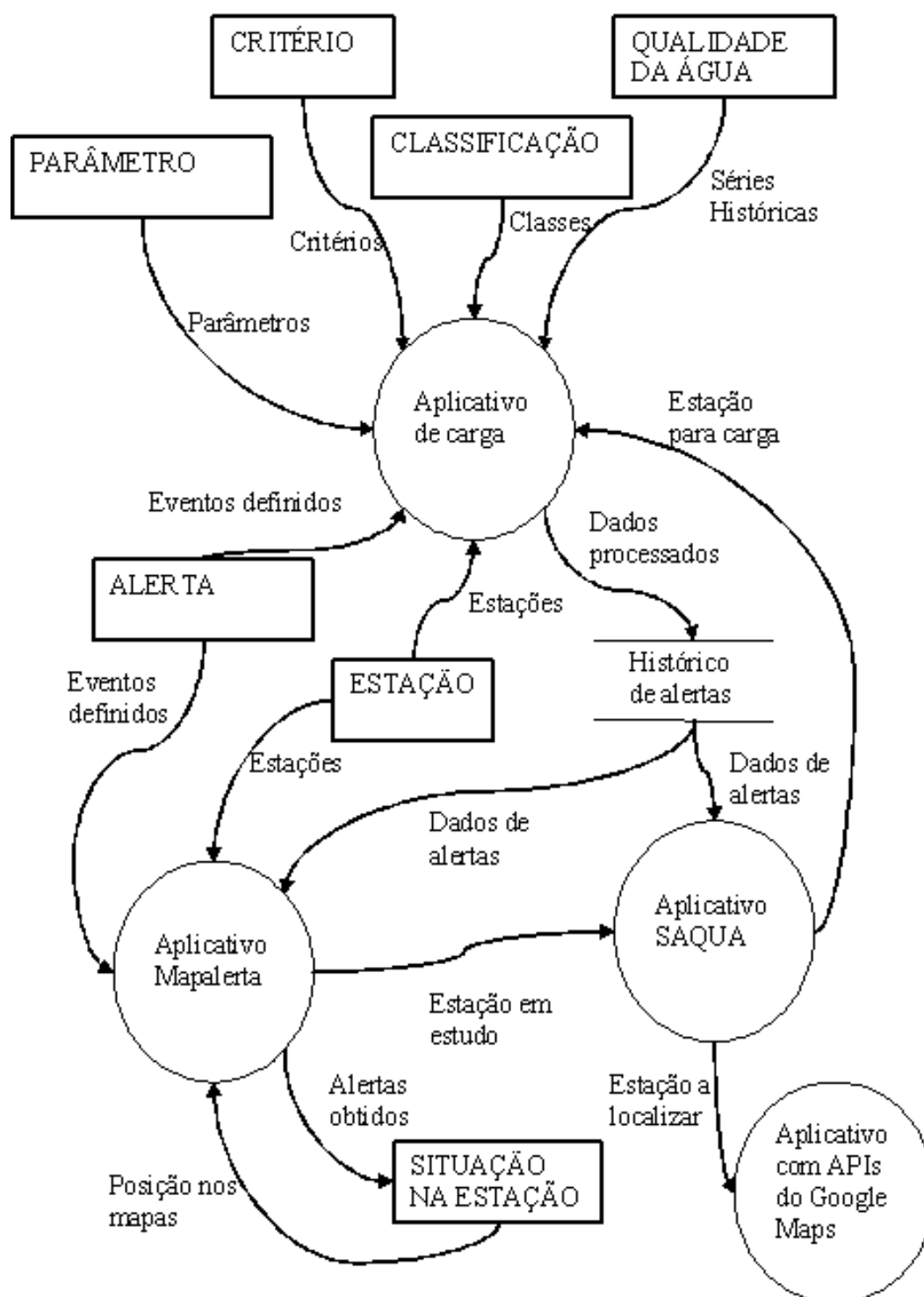


Figura 5 - Diagrama de fluxo de dados para o SAQUA

As funcionalidades do SAQUA serão mostradas inicialmente por um Diagrama de Transição de Estados que mostra a lógica na carga de dados do Sistema HIDROWEB para o SAQUA e, dentro de um contexto sobre o software aplicado. Em seguida, será apresentado um Diagrama de Casos de Uso e 2 (dois) Diagramas de Sequência.

1.6 Diagrama de Transição de Estados HIDROWEB - SAQUA

O Diagrama de Transição de Estados mostra a lógica da implementação utilizada na cópia dos dados históricos fornecidos pelo Sistema HIDROWEB para as entidades do SAQUA.

Algumas funções são responsáveis pelo armazenamento das informações, são elas as funções “Alterar” e “Incluir”. A função “Alterar” altera os valores de dados de parâmetro, de acordo com as estatísticas, sem criar novo registro; e a função “Incluir” inclui os valores dos dados para cada parâmetro, criando as estatísticas e o novo registro.

Alguns estados são posições intermediárias para efetuar determinada ação e seguir para o estado seguinte. O estado “Estação mesclada” efetua a ação de “Permanecer no mês” e segue para o estado “meses seguintes”. O estado “Término mês seguinte” efetua a ação “Finalizar” e segue para o estado “Termino mês inicial”. O estado efetua a ação “Finalizar” e segue para o estado “Estação Seleccionada”, terminando o ciclo.

Quanto às variáveis “T” de inicial e “S” de seguinte, elas são usadas para manipular os dados e estatísticas obtidas. A carga das séries históricas segue a ordem cronológica dos dados fornecidos, logo o armazenamento começa quando existem mais de duas (2) informações para cada dado de parâmetro, por isto a **Figura 6** mostra duas variáveis (“S” e “T”).

O estado de “Estação Mesclada” foi criado para casos em que existam 2 (dois) códigos iguais de estação cadastradas na tabela “estacao”, a fim de dar continuidade a uma série anterior coletada no mesmo local. Por isto, neste estado, após a exclusão dos dados cadastrados cuja data seja superior ao dado cuja data seja inferior ao mês do dado recebido e possua alguma ocorrência, o qual será considerado como “dado junção”, será feita seu repasse para o dado inicial, variável “T”, colocado o dado recebido no segundo dado “S”, e seguindo para o estado dos “Meses Seguintes”.

O diagrama de Transição de Estados, mostrado na **Figura 6**, expressa esta lógica implementada no arquivo *carga.php*.

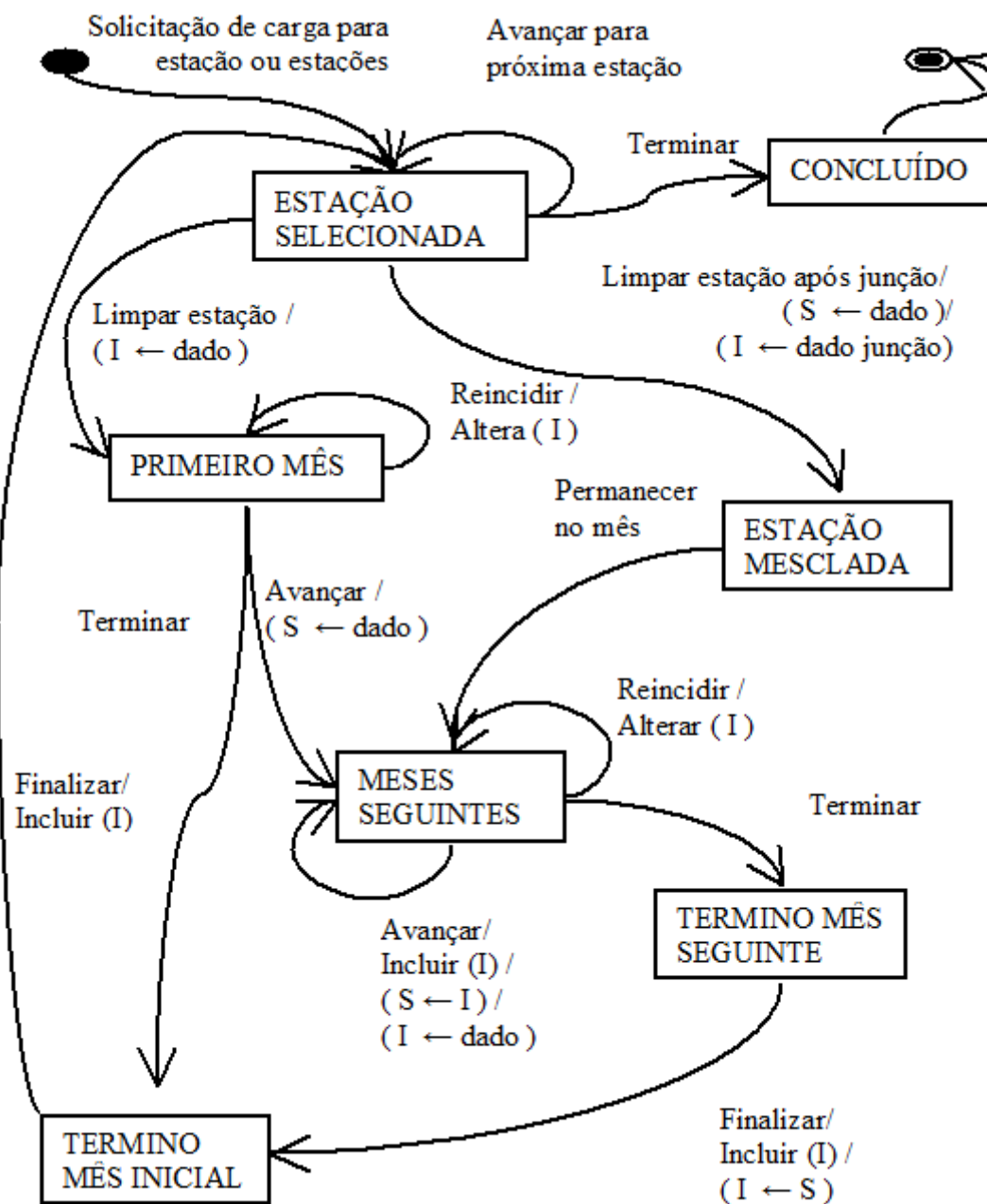


Figura 6 - Diagrama de Transição de Estados ao carregar dados

Por exemplo, uma mesma estação identificada pelas mesmas coordenadas foi cadastrada e duas séries históricas foram fornecidas em arquivos distintos. Com isto ambas foram cadastradas no SAQUA com o mesmo código na tabela “estacao”, inicialmente a mais antiga e depois a mais recente. Porém, pode acontecer que em algum momento os dados se sobreponham e com isto prevalecerá os dados da estação mais recente.

Nesta etapa, o dado é cadastrado na tabela “alerta”, ou seja, nesta parte do programa é feita a classificação dos dados segundo o critério cadastrado, gerando os alertas e preenchendo meses em que não existam ocorrência da coleta com zero a fim de facilitar a exposição gráfica destes dados no SAQUA.

1.7 Conjunto de Programas Usados

Na construção do SAQUA foram utilizados programas desenvolvidos como software livre a fim de obter algumas funcionalidades conforme descrito na **Tabela 3**.

Tabela 3 - Funcionalidades presentes na construção do SAQUA pelos programas livres

Software Livre	Recursos	Funcionalidades
Servidor Apache, com a linguagem PHP aplicada e algumas bibliotecas necessárias	Disponibiliza ao usuário soluções em interface <i>web</i> , permite integração aos demais serviços e o uso de CGI e API	O usuário Pesquisador ou gestor pode escolher opções no SAQUA interagindo com todos os aplicativos através de uma interface única
Servidor Mapserver com recursos de bibliotecas na linguagem PHP	Apresenta feições em mapas permitindo manipulação em mapas através de arquivos temporários e recursos em linguagem de programação	Manipular de forma transparente alertas em mapas geo-referenciados através de visualizações e seleções sobre as coordenadas das agências onde os dados foram coletados
SGBD PostgreSQL com consultas e atualizações em linguagem PHP	Fornecer um sistema de gerência de banco de dados seguro e portátil	Através de um usuário específico manipula de forma transparente os dados dos parâmetros de qualidade de água em uma base de dados confiável

Larman (2000, p.72-73) esclarece que “As funções do sistema, identificadas durante as especificações de requisitos precedentes, deveriam ser todas alocadas a casos de uso.”... “Um diagrama de caso de uso ilustra um conjunto de casos de uso para um sistema, os atores e os casos de uso”

Foi elaborado, então, o diagrama de casos do uso do SAQUA, conforme mostrado na **Figura 7**.

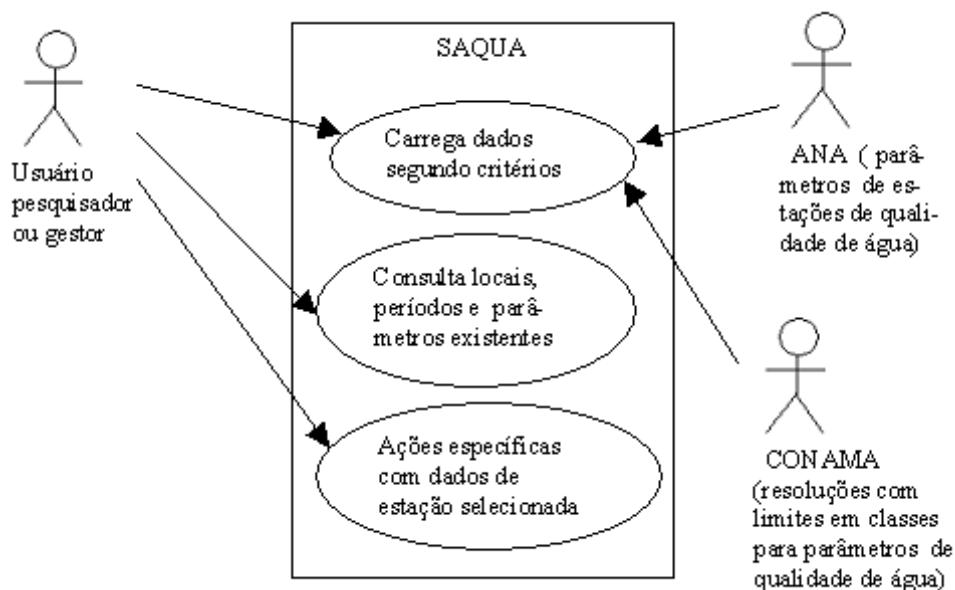


Figura 7 - Diagrama de casos de uso do Sistema SAQUA

Este diagrama mostra os seguintes casos de uso:

O primeiro indica a carga de dados numa estrutura resultante, conforme o DER para o SAQUA, pela intervenção do usuário pesquisador, que faz uma solicitação conforme descrito no Diagrama de Transição de Estados.

Larman (2000, p.143) afirma que “Um diagrama de sequência do sistema é uma figura que mostra, para o particular cenário de um caso de uso, os eventos que os atores externos geram, sua ordem e os eventos entre sistemas”.

Portanto, tendo em vista as especificações do sistema construído, os casos de uso foram divididos em 2 (dois) diagramas de sequência, os quais são empregados fora de seu uso comum uma vez que não foi utilizada linguagem orientada à objeto, mas sim linguagem estruturada, e estão presentes na **Figura 8** e na **Figura 9**.

A carga dos dados foi efetuada através do aplicativo “carga.php”, porém ela iniciou através de consultas específicas a dados das estações de qualidade de água, sendo feito primeiro uma consulta de local sobre a estação, mostrada na **Figura 8**, para se ter acesso a atualização dos dados da estação através da seleção de “carga”, mostrada na **Figura 9**.

O segundo caso de uso se refere à consulta de locais, períodos e parâmetros existentes, partindo de uma seleção inicial, e pode ser compreendida através do diagrama de sequência de seleções, na **Figura 8**.

Outras consultas específicas podem ser encontradas no Diagrama de sequência de ações específicas e carga de dados, ilustradas na **Figura 9**.

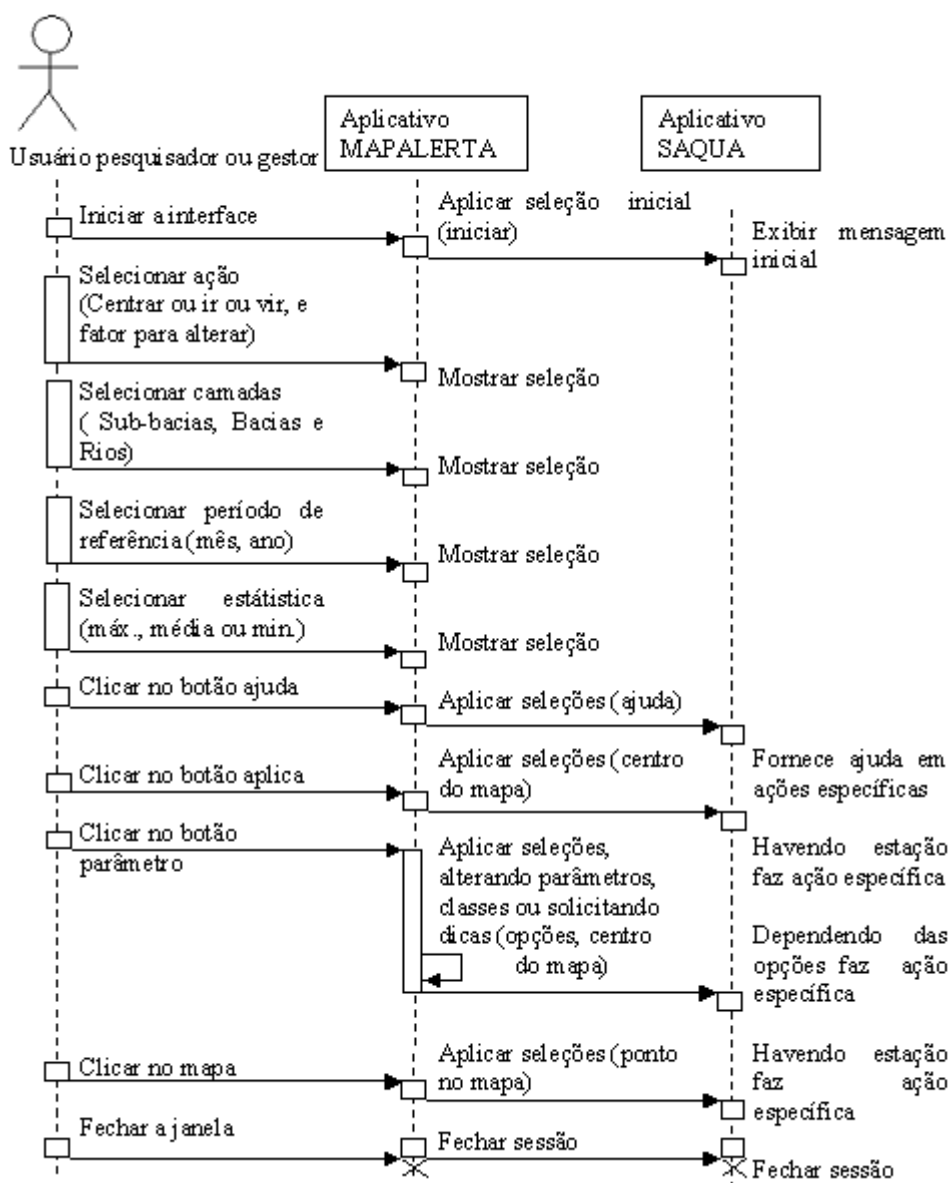


Figura 8 - Diagrama de sequência: consultas de locais, períodos e parâmetros

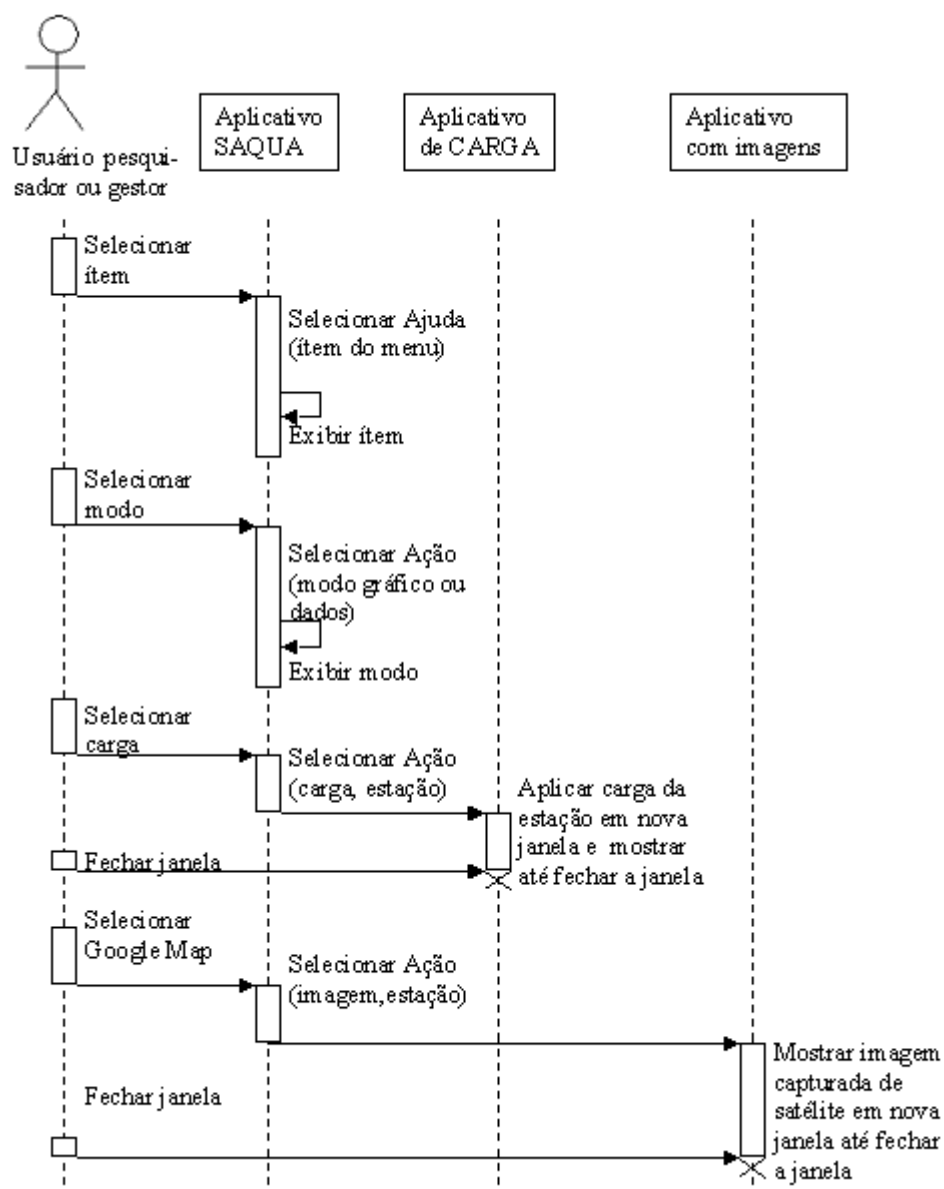


Figura 9 - Diagrama de sequência: consultas específicas e carga de dados

2 APLICAÇÃO DO SISTEMA SAQUA

2.1 Aspectos gerais da implementação

Além dos software livre citados, a implementação do SAQUA também utilizou a ferramenta OWTChart. Esta ferramenta possui código aberto e é baseada na biblioteca “GDChart”, voltada à plataforma “Windows”, e que permite apresentar gráficos a partir de dados sequenciais.

Embora o estudo tenha focado a bacia do Rio Paraíba do Sul, este sistema poderá ser aplicado em qualquer sub-bacia nacional, havendo disponibilidade de dados da ANA, utilizando os limites dos parâmetros de qualidade de água que sejam cadastrados.

A interface da implementação do SAQUA, feita em HTML, com linguagem PHP (Achour, 2007), se divide inicialmente em 2 (dois) ambientes (*frames*): superior e inferior na mesma janela de apresentação. O ambiente inicial desta interface encontra-se na **Figura 10**.

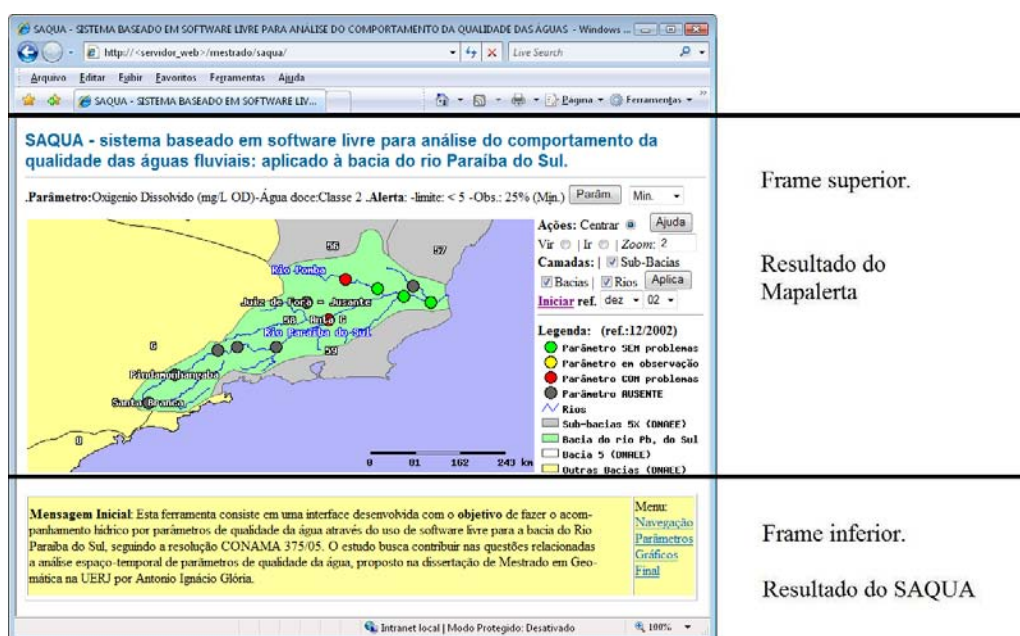


Figura 10 - Ambiente inicial da interface implementada

O ambiente superior é integrado ao Mapserver através de 2 (dois) arquivos:

- O arquivo *mapalerta.php.map*, que faz as definições de comportamento do mapa, mostra os alertas na imagem, sendo constantemente copiado em pasta temporária para interagir com o Mapserver. Nele estão indicadas as pastas utilizadas, as especificações de escalas e feições;

- O arquivo *mapalerta_saqua.php*, que alterna os alertas segundo os parâmetros e período de tempo monitorado, além de interagir com o mapa definindo camadas e formas de interação, podendo chamar o ambiente inferior com informações específicas de estação ou menu de ajuda. Os alertas são alternados através de atualizações no arquivo de dados das feições pelas informações consultadas na base de dados do Servidor PostgreSQL.

Uma primeira versão para esta interface, sem o uso de arquivo temporário para definição de mapas, foi um “Servidor *Web* para acompanhamento da evolução de parâmetros físico químicos de qualidade das águas”, Glória; Ribeiro; Pimentel da Silva (2007) utilizável para um monitoramento onde exista um parâmetro específico de qualidade de água a ser monitorado em tempo real. Esta primeira versão ficou documentada pelos arquivos *alerta.php* e *mapalerta.php*.

O ambiente inferior é composto pelo aplicativo SAQUA, consistindo no arquivo de integração *saqua.php*, que recebe as solicitações do ambiente superior, ou aplicativo Mapalerta, e exibe as informações solicitadas, consultando quando necessário a base de dados do Servidor PostgreSQL, além de poder alternar sua apresentação através de sub-menus.

Neste ambiente inferior também é possível chamar outros aplicativos, como para a carga da estação selecionada, por meio do arquivo *carga.php*, ou para mostrar a imagem de satélite fornecida pela “API do Google Map” Glória; Campelo; Ribeiro (2007), localizando as coordenadas da estação e suas proximidades, através do arquivo *google_estac_off.php*.

A “API do Google Map” é um software gratuito e foi utilizada para obter imagens de satélite, o qual possui algumas restrições de uso, como por exemplo a presença de seu logotipo nas imagens. O arquivo *google_estacoes.php* foi criado para interagir com o arquivo *google_estac_off.php* e permitir acessos em tempo real.

A instalação do SAQUA, após configurações do software livre “Mapserver” para Windows, na pasta *ms4w*, foi preparada para ser feita de forma transparente ao usuário, após a colocação de todos seus arquivos, seja por um processo de cópia ou descompactação, seguido pela execução do arquivo de lote *instala.bat*.

Todos os códigos fontes implementados e alterações em arquivos de configuração efetuadas estão documentados no APÊNDICE D, sua distribuição é expressa no item “2.4 Organização de pastas e arquivos”. No caso de pretender instalar uma nova versão de software livre, ou utilizar o mesmo produto com as alterações necessárias deve-se seguir o item “2.5 Instalações e configurações para o SAQUA”.

O próximo item mostra as alterações feitas nas feições para a identificação dos alertas, as quais serão levadas juntas com o código fonte para o funcionamento do SAQUA. Inclusive este item permite observar o tratamento que foi dado à bacia do Rio Paraíba do Sul, e com alguns parâmetros disponibilizados.

2.2 Os alertas nas feições

Após descompactar as feições de bacias obtidas no sítio do Sistema HIDROWEB foram separados os arquivos a serem utilizados correspondentes às feições “agua_qualidade5”, “hidrografia5”, “subbacias5” e as de todas as bacias utilizando o nome no formato “baciaX”, onde “X” é o número da bacia correspondente à divisão original do DNAEE.

Foi escolhida uma amostra de 12 estações, o que facilitou identificar os alertas nos mapas, ocultando estações sem dados atuais e reduzindo a quantidade de dados carregados, pois o número de estações disponíveis no sítio da ANA é bastante elevado, acima de 80 nesta bacia.

Para identificar os alertas nos mapas, decorrentes do histórico de parâmetros com valores com problemas, foram definidos círculos que dependendo de sua cor expressam 4 (quatro) situações: - Parâmetro sem problema pela cor verde; - Parâmetro em observação pela cor amarela; - Parâmetro com problema pela cor vermelha; - Parâmetro ausente pela cor cinza.

A escolha para o tamanho dos círculos depende não apenas da quantidade de estações no mapa e da escala utilizada como também das proximidades das estações, sendo obtido o resultado para uma tela com resolução de 800x600 pixels. No caso de utilizar uma resolução gráfica diferente o efeito pode ser compensado através de aproximações (*zoom*) na interface *web*.

Uma primeira versão utilizando estas feições foi implementada conforme descrita no artigo “Servidor *web* para acompanhamento da evolução de parâmetros físico químicos de qualidade das

águas” Glória; Ribeiro; Pimentel da Silva (2007, p.1848), apresentado no XXIII Congresso Brasileiro de Cartografia como uma solução para mostrar alertas de forma geo-referenciada.

Esta primeira implementação seria uma solução para monitorar um determinado parâmetro, onde o estado de alerta é atualizado diretamente da feição utilizada, e se pode consultar cada valor do parâmetro nas estações em tempo real. Nesta solução, os círculos de alertas não alteram de tamanho e dependendo da escala utilizada pode ocorrer sobreposição destes, para evitar isto, basta limitar a extensão máxima do mapa para uma região mais próxima à sub-bacia onde foram definidos os círculos.

Esta versão também não previa alterações de alertas conforme seleção de parâmetros e períodos referentes aos dados históricos de qualidade de água. Estes itens foram incorporados na construção do SAQUA, onde foi utilizado o código destas interfaces, permitindo fazer alterações nos dados das feições por meio de arquivos temporários e preencher os alertas.

O re-uso é o aproveitamento de um mesmo artefato em vários lugares de um único sistema ou de diversos sistemas diferentes e o elemento é copiado e alterado em cada um dos locais em que é reutilizado (STAA, 2000, p.6).

Nas feições estão cadastrados, entre outros itens, os traçados dos rios por segmentos, sendo selecionados apenas os pertencentes à bacia do Rio Paraíba do Sul, e os nomes das estações com seus códigos. Para estas foram selecionados apenas uma amostra com seus dados atualizados de acordo com mudanças ocorridas nas localidades.

Esta seleção ocorreu através da inclusão e preenchimento do campo de código da sub-bacia no arquivo *hidrografia5.dbf*, para os segmentos de rios; e da inclusão e preenchimento do campo de alertas no arquivo *agua_qualidade5.dbf*, para a seleção de amostra, atualização de nomes e códigos das estações e exposição dos círculos de alertas.

A partir de um monitoramento inicial, o SAQUA permite alterações de períodos mensais, tipos de estatísticas, parâmetros e simulações de classe, que afetam os valores dos alertas que serão gravados nos dados da feição de acordo com o conteúdo arquivo *agua_qualidade5.dbf*.

A criação destes campos com o preenchimento e alteração dos valores iniciais foi feita utilizando a planilha *Calc* do software livre *OpenOffice*, disponível tanto para plataforma Linux como para plataforma Windows, conforme apresentado na **Figura 11**.

	A	B	C
285		0	
286	Rio Pomba	58600000	58
287	Rio Paraíba do Sul		0 58
288	Rio Paraíba do Sul	58001000	58
289			0 58
290			0 58
291			0 58
292			0 58
293			0 58
294	Rio Paraíba do Sul	58001000	58
295	Rio Paraíba do Sul		0 58
296	Rio Paraíba do Sul	58001000	58
297			0 58

Figura 11 - Dados em *hidrografia5.dbf*

Com a criação do campo “sub-bacia” foi possível selecionar dentro da feição hídrica fornecida para a Bacia 5, ou Bacia Atlântica trecho leste, apenas os rios da Sub-bacia 58, ou bacia do Rio Paraíba do Sul, permitindo desenhar esta camada no mapa sem os rios das demais sub-bacias da Bacia 5.

Assim, acrescentou-se o campo *sub-bacia* ao arquivo *hidrografia5.dbf* preenchendo a célula “C1” da planilha com “SUB_BACIA,C,2”, e salvando-o junto com seu preenchimento e segundo o seguinte critério:

- Nas linhas cujo código dos rios, coluna “B”, iniciam por padrão com valor 5 (cinco) seguido de 8 (oito), além de algumas outras linhas que seguem com código de rios 0 (zero), abaixo das linhas com o referido padrão no código de rios, e que não pertenciam à sub-bacia 59, a qual não possuía código de rios identificado neste arquivo, foram preenchidas com o número 58 (cinquenta e oito) no campo criado para “sub-bacia”.

Para identificar os alertas nas feições das estações de qualidade de água criou-se o campo “alerta” no arquivo de dados *agua_qualidade5.dbf*, similar ao que foi feito no arquivo *hidrografia5.dbf* com o campo “sub-bacia”, conforme ilustrado na **Figura 12**.

	A	B	C	E	G	H	I	J	K	M
58	56995500	17	-19,43	-40,67	Ponte do Pancas	Pancas	ES	BH	FDSQ	
59	58099000	2	-23,37	-45,9	Santa Branca	Paraíba do Sul	SP	SP	FDSQT	0
60	58183000	2	-22,92	-45,47	Pindamonhangaba	Paraíba do Sul	SP	SP	FDSQT	0
61	58235100	2	-22,53	-44,77	Queluz	Paraíba do Sul	SP	SP	FDSQ	0
62	58250000	2	-22,47	-44,45	Resende	Paraíba do Sul	RJ	SP	FDQT	0
63	58300000	2	-22,53	-44,18	Barra Mansa - Jusante	Paraíba do Sul	RJ	SP	FDSQT	
64	58321000	6	-22,47	-43,83	Barra do Pirai	Paraíba do Sul	RJ	SP	FDSQT	0
65	58335000	2	-22,85	-44,2	Lidice	Pirai	RJ	SP	FDQ	
66	58480500	5	-21,75	-43,33	Juiz de Fora - Jusante	Paraibuna Mineiro	MG	SP	FDSQ	0
67	58630002	2	-22,03	-42,98	Anta G	Paraíba do Sul	RJ	SP	FRDSQT	0
68	58770000	7	-21,38	-42,7	Cataguases	Pomba	MG	SP	FDSQT	2
69	58790002	7	-21,53	-42,18	Santo Antonio de Padua II	Pomba	RJ	SP	FDSQT	1
70	58880001	8	-21,65	-41,75	Sao Fidelis	Paraíba do Sul	RJ	SP	FDSQT	0
71	58960000	8	-21,48	-41,6	Cardoso Moreira - RV	Muriae	RJ	SP	FDSQ	0
72	58974000	8	-21,75	-41,32	Campos - Ponte Municipal	Paraíba do Sul	RJ	SP	FRDSQ	0

Figura 12 - Dados em *agua_qualidade5.dbf*

Embora o SAQUA utilize este arquivo como referência para preencher os alertas nos mapas, o preenchimento inicial destes alertas efetua a seleção da amostra. Após feito este preenchimento inicial dos alertas, ainda que o SAQUA não possua dados sobre situação de alertas de uma estação da amostra, ou que os mesmos não tenham sido cadastrados, será utilizado o valor 3 (três) que correspondente à estação sem dados, mantendo assim a seleção das amostras.

Pode-se observar, na **Figura 12**, que foi criado o campo *alerta* através do preenchimento da célula “M1” com o conteúdo “ALERTA,C,1”, cujos dados foram preenchidos para as estações da amostra seguido os seguintes critérios: - Valor 2 (dois), para parâmetro com problema; - Valor 1 (um), para parâmetro em observação; - Valor 0 (zero), para parâmetro sem problema.

Como não foi considerada estação selecionada sem dados de parâmetros, para um período inicial arbitrado, não foi usado o valor 3 (três), utilizado para parâmetro ausente. As estações de qualidade de água que não pertencem a esta bacia, e as que não pertenciam à amostra selecionada, como as estações de Ponte do Pancas, de Barra Mansa e de Lídice, não receberam valores definidos para o campo *alerta* a fim de serem excluídas do monitoramento pelo SAQUA.

A **Tabela 4** mostra os dados anteriores do arquivo *agua_qualidade5.dbf* e os dados que foram atualizados nas linhas conforme mostrado na **Figura 12**.

Tabela 4 - Atualizações no arquivo *agua_qualidade5.dbf*

Linha ou registro mais 1(um)	Dados anteriores dos campos		Dados atualizados dos campos	
	Código da Estação	Estação	Código da Estação	Estação
61	58235000	Queluz	58235100	Queluz
67	58630000	Anta F	58630002	Anta G
69	58790000	Santo Antônio de Pádua	58790002	Santo Antônio de Pádua II

A situação inicial destes alertas foi arbitrada segundo consulta pelo SAQUA para o período do mês de dezembro de 2002, verificando os valores de oxigênio dissolvido segundo o padrão do CONAMA utilizado para a classe 2, que é a classe de enquadramento destas estações.

Nesta situação inicial considerou-se a média mensal como estatística e situação de observações para valores superiores à 25%, ou seja, neste caso em que a condição de situação de alerta é ser inferior à 5 mg/L O₂ a condição de observação é ser inferior à 6,25 mg/L O₂.

Após os ajustes dos dados nas feições foi definido um tamanho inicial para o círculo do alerta, que se situa no centro da posição geográfica da estação de qualidade de água, definido pelo comando “SIZE 15” dentro do arquivo *alerta.map*.

Este código foi re-escrito no arquivo *mapalerta.map*, com o cuidado de localizar cada linha de ocorrência destes comandos para permitir redimensionar os círculos de acordo com as cores definidas nas situações de alertas.

Da mesma forma foi localizada a linha de ocorrência do comando “DATA agua_qualidade5” no arquivo *mapalerta.map* a fim de substituí-la por feições criadas em arquivos temporários, pois define o arquivo de feições utilizado na exposição dos alertas.

Na representação da situação na estação pelo círculo de alertas, estes permitem interação e redimensionamento de acordo com os mapas, observando o seguinte:

- O intervalo de longitude escolhido no mapa inicial ocupa 5,5 graus (entre -46,5° e -41°), logo o raio do círculo precisa ser ajustado na figura de mapa. Considerando um círculo completo de 360 graus, uma constante (c) é utilizada para o ajuste e identificação do círculo selecionado no mapa (na função “EstacaoProx” dos arquivos *alerta.php* e *mapalerta.php*);

- No uso da interface, o raio do círculo é atualizado pelo fator de aproximação ou afastamento solicitado, sendo que no caso de aproximações maiores que a região inicial dos mapas o círculo de alerta não altera de tamanho, embora a escala do mapa diminua, pois resultaria em representações grandes e desnecessárias para uma boa identificação;

- Quando atingido o valor máximo de afastamento definido para o mapa representativo das bacias hidrográficas brasileiras, ocupando 55° no intervalo de longitude (entre -70° e -15°), o raio do círculo é reduzido em 10 vezes o seu tamanho inicial para identificação no mapa, através de um multiplicador, para que atinja seu tamanho mínimo no mapa.

Ou seja, o tamanho do círculo de alertas trabalhou com diferentes variáveis prevendo a alteração na escala dos mapas e a mudança no seu tamanho para evitar sobreposições.

No valor máximo de afastamento pode-se notar que o intervalo de latitude aumentou em 10 vezes o intervalo inicial, englobando todo o mapa de bacias brasileiras, com intervalo de 41 graus de latitude (entre -35° e $+6^\circ$), partindo de um intervalo inicial de 4,1 graus (entre $20,4^\circ$ e $24,5^\circ$).

Como o Mapserver trabalha com graus estipulados em decimais, e a bacia do Rio Paraíba do Sul localiza-se na Região Sudeste entre os paralelos $20^\circ 26'$ e $23^\circ 00'$ e os meridianos $41^\circ 00'$ e $46^\circ 30'$ oeste de Greenwich, ANA; AGEVAP (2006, p.2), valores que em decimais correspondem aos paralelos em $-20,4^\circ$ e -23° , e aos meridianos em -41° e $-46,5^\circ$ respectivamente, e que estão englobados nos intervalos iniciais previsto no mapa para latitude entre $-20,4^\circ$ e $-24,5^\circ$ e para longitude entre -41° e $-46,5^\circ$, nota-se que os valores utilizados podem partir de uma conversão de minutos para decimais, e que a diferença na latitude foi usada para englobar toda bacia nacional na ampliação máxima.

Como resultado obteve-se uma variação de escalas, conforme mostra a **Figura 13**.

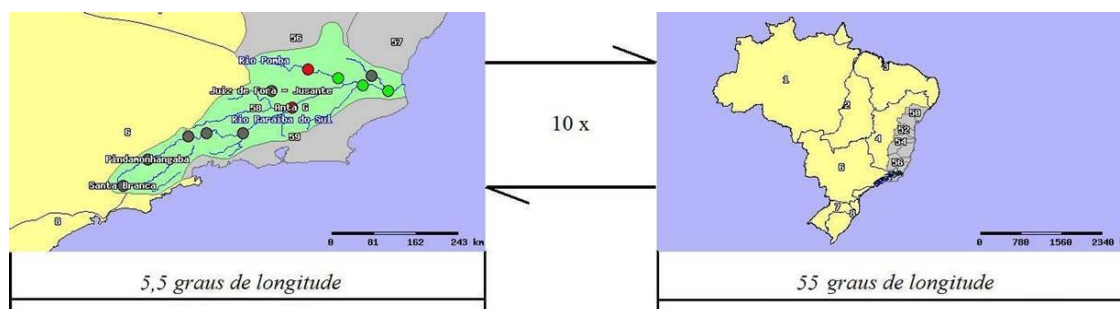


Figura 13 - Detalhe cartográfico do intervalo de coordenadas iniciais e limites definidos

O tamanho do círculo de alertas trabalhou com diferentes variáveis prevendo a alteração na escala dos mapas e a mudança no seu tamanho para evitar sobreposições.

Caso se deseje incluir novas estações cujas coordenadas não estejam presentes na feição “agua_qualidade5”, além de verificar se o tamanho dos círculos de alertas atendem à nova situação, será necessário utilizar uma ferramenta para alterá-la. Existem ferramentas disponíveis no mercado com o próprio ArcView da ESRI, como também em software livre como o QuantumGIS¹⁵, que é um Sistema de Informações Geográficas multiplataforma, compatível com a plataforma windows, e suporta formatos matriciais e vetoriais, incluindo o formato “shapefile”.

Com o propósito de validar o sistema proposto, foi escolhida uma amostra onde existem estações em operação por um grande período de tempo, e outras estações que foram instaladas em regiões onde havia históricos anteriores, conservando as mesmas coordenadas geográficas, e que tiveram seus dados incorporados aos da estação atual, precisando em alguns casos atualizar a descrição atual da estação e seus códigos.

Porém, conhecer apenas os dados das feições utilizadas, que consistem após estas alterações na entidade *Situação na Estação* não é o suficiente para entender como o SAQUA identifica e armazena os alertas para colocá-los nestas feições, portanto no próximo tópico veremos toda a estrutura do banco de dados utilizado expresso através de tabelas.

2.3 Atributos e campos utilizados pelo SAQUA

As entidades, assim como as informações de banco de dados necessárias para um Dicionário de Dados, podem ser encontradas nas tabelas de atributos utilizados pelo SAQUA, podendo acrescer informações do Dicionário de Dados do sistema Hidro, presente no Apêndice A do Manual do Usuário, ANA (2002), com mais detalhes dos valores permitidos de alguns campos.

As tabelas de atributos de entidades seguem por ordem alfabética, iniciando pelas implementadas no banco de dados do PostgreSQL, especificamente criadas para o SAQUA.

¹⁵Sítio oficial do QuantumGIS: <http://qgis.org/>

A **Tabela 5** mostra os atributos da Entidade *Alerta*, onde ficam armazenados os históricos dos parâmetros selecionados por classe e estações definidas, avaliados pelos critérios estabelecidos, com os alertas obtidos nas estatísticas para a situação de média, máxima e mínima.

Tabela 5 - Atributos da entidade *Alerta*, implementados em PostgreSQL

Atributo da entidade <i>Alerta</i>	Campo da tabela <i>alerta</i>	Tipo	Tamanho
Código da estação	cod_estac	Número	Int
Código do parâmetro	cod_param	Número	Int
Código da classe	cod_class	Número	Int
Mês	Mês	Número	Int
Ano	Ano	Número	Int
Média	media	Número	Real
Máximo	maximo	Número	Real
Mínimo	minimo	Número	Real
Ocorrências	ocorre	Número	Int
Situação na média	situa_med	Texto	1
Situação no máximo	situa_max	Texto	1
Situação no mínimo	situa_min	Texto	1

A **Tabela 6** mostra os atributos da Entidade *Classificação*, onde fica armazenada a descrição do tipo de rede pela utilização da água doce, seguindo o mesmo código de tipo de rede utilizado pelo sistema Hidro, conforme mostrado na **Tabela 12**.

Tabela 6 - Atributos da entidade *Classificação*, implementados em PostgreSQL

Atributo da entidade <i>Classificação</i>	Campo da tabela <i>classes</i>	Tipo	Tamanho
Código da classe	cod_class	Número	Int
Nome da classe	classe	Texto	20

A **Tabela 7** mostra os atributos da Entidade *Critério*, onde ficam armazenados os critérios para os parâmetros de qualidade de água previstos.

Tabela 7 - Atributos da entidade *Critério*, implementados em PostgreSQL

Atributo da entidade <i>Critério</i>	Campo da tabela <i>critério</i>	Tipo	Tamanho
Código do parâmetro	cod_param	Número	Int
Código da classe	cod_class	Número	int
Condição	cond	Texto	2
Limite	limite	Número	real
Aviso	aviso	Número	real

A **Tabela 8** mostra os atributos da Entidade *Estação*, onde ficam armazenadas a localização dos arquivos integrantes da Entidade *Qualidade da Água* que efetivamente são utilizadas. O nome da estação é descrito no atributo “Estação” da Entidade *Situação na Estação*, na **Tabela 11**.

Tabela 8 - Atributos da entidade *Estação*, implementados em PostgreSQL

Atributo da entidade Estação	Campo da tabela estacao	Tipo	Tamanho
Código da estação	cod_estac	Número	Int
Código da classe	cod_class	Número	Int
Arquivo de origem	arq_orig	Texto	80

A **Tabela 9** mostra os atributos da Entidade *Parâmetro*, onde ficam armazenados os parâmetros que são efetivamente utilizados.

Tabela 9 - Atributos da entidade *Parâmetro*, implementados em PostgreSQL

Atributo da entidade Parâmetro	Campo da tabela param	Tipo	Tamanho
Código do parâmetro	cod_param	Número	Int
Unidade	unidade	Texto	14
Nome do parâmetro	parametro	Texto	30

A **Tabela 10** foi derivada dos arquivos *QualAgua.txt* fornecidos pelo Sistema HIDROWEB em formato CSV, ou seja, na prática o tamanho dos campos está limitado à posição do separador ponto e vírgula e por isto não foram expressos.

Tabela 10 - Atributos da entidade *Qualidade de Água*, obtidos em arquivos texto

Atributo da entidade Qualidade de Água	Campo do arquivo QualAgua.txt	Tipo	Tamanho
Código da estação	EstacaoCodigo	Número	-
Nível de consistência	NivelConsistencia	Número	-
Data	Data	Data	-
Hora	Hora	Hora	-
Número da medição	NumMedicao	Número	-
Posição horizontal da coleta	PosHorizColeta	Número	-
Posição vertical da coleta	PosVertColeta	Número	-
Choveu	Choveu	Número	-
Profundidade	Profundidade	Número	-
Temperatura do ar	TempAr	Número	-
Temperatura da amostra	TempAmostra	Número	-
* 135 parâmetros (11 ao 145)	de pH à Amoniac	Número [1..135]	-
Status dos parâmetros de qualidade da água	QualAgua001..137Status	Número [1..137]	-

(*) Os atributos dos 135 parâmetros de qualidade de água encontram-se no APÊNDICE B

Esta tabela mostra os atributos da Entidade *Qualidade de Água*, que consiste em vários arquivos texto, existindo pelo menos um para cada estação cadastrada, onde conserva-se não apenas todos os atributos e campos, como também o nome da entidade definida no Sistema Hidro.

Também podem ser encontrados os nomes dos campos e tipos dos 135 parâmetros no Apêndice A do Manual do Usuário, ANA (2002, p.57-62), os quais não foram necessários na implementação por considerar valor de código numérico, sendo código zero (0) para a “EstacaoCodigo” e assim progressivamente, passando pelos códigos 11 (onze) para o “pH” e 145 (cento e quarenta e cinco) para “Amoníaco”. Os códigos dos parâmetros encontram-se no APÊNDICE D.

A **Tabela 11** mostra os atributos da Entidade *Situação na Estação*, que consiste em partes das feições fornecidas cujo acréscimo ocorreu apenas em relação aos atributos “Alerta” no arquivo *agua_qualidade5.dbf* e “Código da sub-bacia” no arquivo *hidrografia5.dbf*. Os nomes dos atributos foram definidos por afinidade aos nomes dos campos armazenados.

Tabela 11 - Atributos da entidade *Situação na Estação*, em formato dbase

Atributo da entidade Situação na Estação	Arquivo Dbase	Campo	Tipo	Tamanho
Código da estação	agua_qualidade5.dbf	codigo	Número	9,0
Roteiro	agua_qualidade5.dbf	roteiro	Número	7,0
Latitude	agua_qualidade5.dbf	lat	Número	7,2
Módulo de Latitude	agua_qualidade5.dbf	latitude	Número	9,2
Longitude	agua_qualidade5.dbf	long	Número	9,2
Módulo de Longitude	agua_qualidade5.dbf	Longitude	Número	9,2
Estação	agua_qualidade5.dbf	estacao	Texto	28
Curso d’água	agua_qualidade5.dbf	curso_dagu	Texto	20
Unidade da federação	agua_qualidade5.dbf	Uf	Texto	7
Operadora	agua_qualidade5.dbf	operadora	Texto	10
Tipos de coleta	agua_qualidade5.dbf	tipo_colet	Texto	10
Observações	agua_qualidade5.dbf	observacao	Texto	15
Alerta	agua_qualidade5.dbf	alerta	Texto	1
Código da bacia	bacia1.dbf à bacia8.dbf e sub_bacias5.dbf	bacia	Número	2,0
Código da sub-bacia	bacia1.dbf à bacia8.dbf , sub_bacias5.dbf e hidrografia5.dbf	sub_bacia	Número	2,0
Nome do rio	hidrografia5.dbf	nome	Texto	50
Código do rio	hidrografia5.dbf	codigo	Número	16,0

Os dados fornecidos junto às feições nos diversos arquivos utilizados, foram unidos nesta tabela conforme foram fornecidos pelo Sistema HIDROWEB, com os poucos campos acrescentados para utilização do SAQUA, no formato “dbase”.

Estes dados são utilizados na construção das feições nos mapas fazem parte da entidade *Situação da Estação* e operam em conjunto com arquivos de formato “shapefile”. Não foram

alterados os campos originais dos arquivos fornecidos pelo Sistema HIDROWEB, apenas acrescido o campo “Alerta” no arquivo *agua_qualidade5.dbf*.

Sabendo que alguns arquivos que representam a entidade *Situação na Estação* serão duplicados e alterados em área temporária, entende-se melhor como será mostrada a relação com a entidade *Alerta*, onde determinado alerta pode ser mostrado em vários locais no mapa por meio de arquivos distintos.

A **Tabela 12** mostra valores utilizados no código da Entidade *Classificação*, cujos atributos foram mostrados na **Tabela 6**, onde também acrescentou-se 2 (dois) outros códigos para unir as quatro classes de água doce que possuem mesmos limites para avaliação do parâmetro Ph, uma para quando este estiver alto, código 11, e outra quando o mesmo estiver baixo, código 12.

Tabela 12 - Classes de tipo de rede segundo Sistema Hidro

Código	Tipo de rede	Código	Tipo de rede
0	Não	6	Água salgada (1)
1	Especial	7	Água salgada (2)
2	Água doce classe 1	8	Água salobra (1)
3	Água doce classe 2	9	Água salobra (2)
4	Água doce classe 3	10	Sem classificação
5	Água doce classe 4		

Com exceção dos arquivos *dbase* presentes na **Tabela 11**, que ficam na pasta “/mestrado/mapalerta”, e dos arquivos em formato texto presentes na **Tabela 10**, que ficam na pasta “/mestrado/saqua/estações”, toda a base de dados administrada pelo Servidor PostgreSQL, seguindo a instalação proposta, ficará dentro da estrutura de pastas deste Servidor “/postgresql” .

A localização dos arquivos é importante não apenas para que se encontre a base de dados, mas também para entender o aplicativo, sabendo onde será feita a carga de dados, além de saber onde serão armazenados os arquivos temporários e outros, o que é o assunto do próximo tópico.

2.4 Organização de pastas e arquivos

O SAQUA é composto por uma estrutura hierárquica de pastas e arquivos. A pasta *mestrado* deve ser colocada na raiz da unidade onde será instalado o *ms4w* (Mapserver para

Windows). Na pasta *mestrado* existe um arquivo para chamada dos aplicativos, *index.php*, e um arquivo em lote para exclusão de arquivos temporários, *limpa_tmp.bat*.

A pasta *mestrado* conforme mostra a **Figura 14**, possui 4 (quatro) sub-pastas, que serão explanadas como pastas, na seguinte ordem: - Pasta *mapalerta*; - Pasta *saqua*; - Pasta *tmp*; - Pasta *instala*.

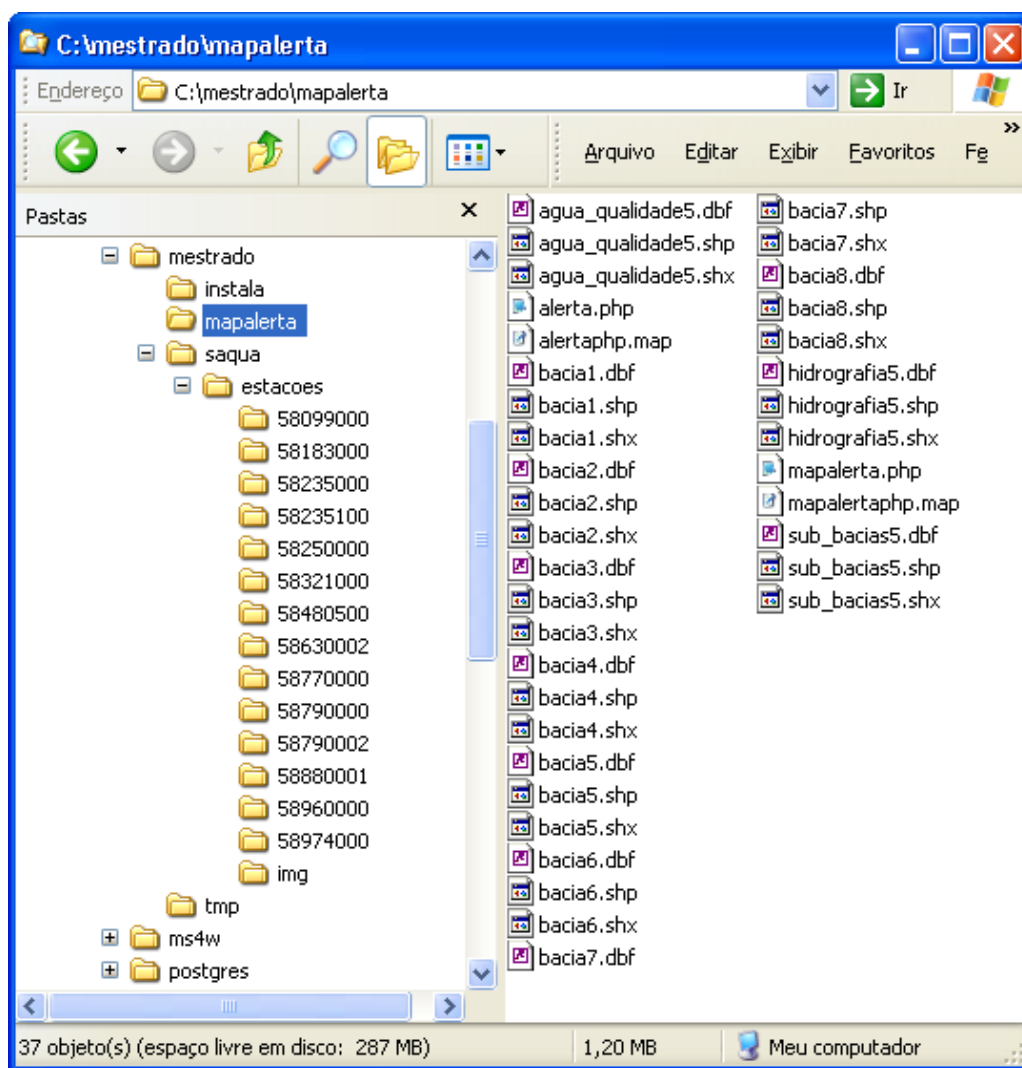


Figura 14 - Localização de arquivos e pastas no servidor

Na pasta *mapalerta* foi colocado inicialmente as feições selecionadas nos 8 (oito) arquivos compactados obtidos no sitio do Sistema HIDROWEB: *Bacia1.zip*; *Bacia2.zip*; *Bacia3.zip*; *Bacia4.zip*; *Bacia5.zip*; *Bacia6.zip*; *Bacia7.zip* e *Bacia8.zip* . Em cada arquivo compactado foram utilizados os arquivos das feições de *bacia*, com o nome de arquivo *bacia* acrescido do número destes arquivos e extensões: *.dbf*; *.shp* e *.shx*.

Do arquivo compactado *Bacia5.zip* também foram utilizadas as feições *hidrografia5*, *subbacia5* e *agua_qualidade5*, com as mesmas extensões. Assim, com 8 arquivos de feições de bacias mais 3 arquivos específicos de feições na sub-bacia 5, multiplicado por 3 tipos de extensões de feição, tem-se 33 (trinta e três) arquivos copiados para esta pasta.

Juntando a estes 33 arquivos os 2 arquivos da interface inicial, *alerta.php* e *alerta.php.map*, e os 2 arquivos reconstruídos para o SAQUA, *mapalerta.php* e *malapertaph.map*, obtém-se os 37 (trinta e sete) arquivos presentes na sub-pasta *mapalerta*, conforme mostra a **Figura 14**.

A pasta *saqua* possui 7 (sete) arquivos com códigos implementados: *carga.php*, *saqua.php*, *index.html*, *mensagem2.html*, *google_estacoes.html*, *google_estacoes.php* e *google_estac_off.php*. Esta pasta também possui a sub-pasta *estacoes*, que por sua vez possui sub-pastas com os códigos das estações selecionadas, cada qual com um arquivo texto, *qualagua.txt*, obtido no sítio do Sistema HIDROWEB em formato compactado, consistindo nas séries históricas utilizadas.

Ainda na sub-pasta *estacoes* existe a sub-pasta *img* onde foram capturadas telas, através da API do Google Map, utilizando o código do arquivo *google_estacoes.html*, para cada estação selecionada, totalizando os 12 (doze) seguintes arquivos: *anta_g.png*; *barra_do_pirai.png*; *campos.png*; *cardoso_moreira.png*; *cataguases.png*; *juiz_de_fora.png*; *pindamanhangaba.png*; *queluz.png*; *resende.png*; *santa_branca.png*; *são_fidelis.png* e *sto_antonio_padua_ii.png*.

A sub-pasta *tmp* é destinada à gravação de arquivos temporários para construção de mapas, tanto da interface inicial como da interface utilizada pelo SAQUA.

Na interface inicial, o nome escolhido para os arquivos de mapas e legendas foram respectivamente *mapalerta* e *legalerta*, seguidos de um valor numérico aleatório de 6 (seis) dígitos, para cada solicitação, e da extensão de formato da figura: *.png*. Esta flexibilidade de poder escolher os nomes dos arquivos temporários deve-se ao fato de utilizar o modo “Mapscript” em vez do modo “CGI” (KROPLA, 2005, p. xxvi).

Na interface utilizada pelo SAQUA para mapas continua existindo a geração destes 2 (dois) arquivos, conservando o mesmo valor numérico aleatório para: 1 (um) arquivo de configuração de mapas, iniciado por *mapalerta* seguido do valor numérico e a extensão *.map*, e 3 (três) arquivos que reconstróem a feição *agua_qualidade5* em arquivos iniciados por *agua_q* seguidos do valor numérico e as 3 (três) extensões de origem: *.dbf*; *.shp* e *.shx*.

O esvaziamento da pasta temporária deve ser feito periodicamente removendo os arquivos antigos. Para remoção básica de todos arquivos foi criado o arquivo *limpa_tmp.bat* com uma única linha de comando: “del c:\mestrado\tmp*.*/S /Q”. Um exemplo de execução periódica do mesmo é mostrado no APÊNDICE C. Restando, desta forma no APÊNDICE D, entre os arquivos da pasta *mestrado*, apenas o conteúdo do arquivo *index.html*.

A pasta *instala* possui os arquivos com sequências de comando, ou arquivos em lote, *instala.bat* e *desinstala.bat*, além do arquivo instalador *postgresql-8.2-int.msi* e do arquivo *cria_e_preenche_banco.php* para criação e preenchimento do banco de dados no PostgreSQL.

Os arquivos implementados estão presentes no APÊNDICE D e podem ser utilizados seguindo as instalações e configurações orientadas no próximo item.

2.5 Instalações e configurações para o SAQUA

Para a execução do código utilizado nas interfaces é necessário que esteja instalado como Servidor Web o Apache à partir da versão 2.0, o PHP a partir da 5.0, o PHP/Mapscript a partir da versão 4.6, o OWTChart a partir da 1.2 o PHP/OGR a partir da versão 1.0. Estes recursos são encontrados no pacote de software livre Mapserver para Windows versão 2.2.4.

Pode-se copiar por mídia o pacote do Mapserver descrito acima, da pasta */ms4w*, e todos os demais arquivos utilizados neste trabalho, na pasta */mestrado*. Estas pastas devem ser colocadas na raiz da mesma unidade de disco e quando executar o arquivo *instala.bat* na sub-pasta */mestrado/instala/*, as demais configurações serão efetuadas e os dados carregados.

Para facilitar o uso de futuras versões, e permitir comparações em diferentes ambientes operacionais, será descrito desde a obtenção de software livre até as configurações necessárias para sua utilização com o SAQUA.

O pacote Mapserver para Windows foi obtido no sítio “<http://maptools.org/ms4w/>” como um arquivo compactado. Este foi descompactado na raiz da unidade do disco escolhido, criando uma estrutura sob a pasta *ms4w*. Para seu funcionamento, ainda são necessárias algumas alterações.

Algumas bibliotecas do PHP, versão 5, não são configuradas na instalação padrão e devem ser habilitadas: a que permite executar funções do Mapserver (mapscript), a que permite executar funções da biblioteca OGR (php_ogr) e a que possui funções de biblioteca dBase (php_dbase).

Isto significa que será necessário acrescentar, ou retirar de comentários, as seguintes linhas no arquivo de carga inicial do php, situado na sub-pasta */ms4w/Apache/cgi-bin*, arquivo **php.ini** :

```
extension=php_mapscript_4.10.0.dll
extension=php_ogr.dll
extension= php_dbase.dll
```

Outra linha alterada foi a referente ao tempo utilizado para execução, que é necessário quando faz a carga dos dados das estações da ANA no SAQUA, assim aumentou-se o valor padrão de 30 segundos para o valor máximo de 30 minutos (1800 segundos), ou seja, o valor da linha padrão “max_execution_time=30” foi alterado para “max_execution_time = 1800”.

Utilizou-se um arquivo de configuração dinâmica do servidor Apache, disponível a partir da versão 2.0 do Apache, o arquivo *httpd_mestrado.conf*, cujo conteúdo é mostrado na **Figura 15** através do editor bloco de notas do windows, que colocado na sub-pasta */ms4w/httpd*, após reiniciar o serviço, identifica a estrutura de pastas criadas para o SAQUA no Apache.

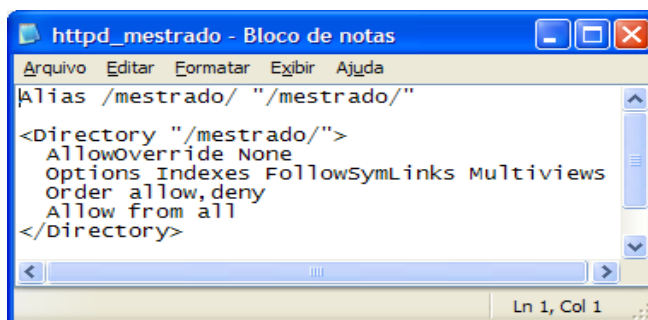


Figura 15 - Arquivo que relaciona as pastas para o Servidor Apache

O arquivo de configuração do Apache, situado na pasta */ms4w/Apache/conf*, arquivo *httpd.conf*, deve estar preparado para executar arquivos com extensão *.php*. Para isto deve possuir ativas as linhas de comandos: “Add handler php-script .php”, “Action php-script /cgi-bin/php.exe” e “Add-Type php-script”.

Terminado estas configurações, pelo menos para a versão utilizada neste trabalho, basta executar o arquivo *apache-install.bat*, que se situa na sub-pasta */ms4w*, para que seja instalado o Servidor Web.

O Sistema Gerenciador de Banco de Dados objeto relacional PostgreSQL foi instalado, opcionalmente no mesmo equipamento, com estrutura de banco de dados para atender a este trabalho.

Para instalar o Servidor PostgreSQL foi obtida uma versão no formato do *Microsoft Windows Installer*, extensão *.msi*, indicada no sítio oficial do PostgreSQL¹⁶, neste caso o arquivo *postgresql-8.2-int.msi*, situado na sub-pasta */mestrado/instala*.

Fez-se a instalação informando o nome do computador local que será utilizado para a criação de uma conta de serviço do usuário “saqua”, assim como haverá outra conta para o superusuário padrão (“postgres”) e a estrutura de seus diretórios será criada a partir da pasta “c:\postgres”. A sintaxe utilizada está contida no arquivo *instalar.bat*, porém esta interface possui interações que permitem fazer a configuração padrão com estes dados.

Para finalizar, é executado o comando que cria o banco e preenche os dados, exceto os de séries históricas: “\ms4w\Apache\cgi-bin\php.exe -f \mestrado\instala\cria_e_preenche_banco.php”.

O preenchimento dos dados de séries históricas de todas pode ser feito ao chamar a página “http://<endereço_do_servidor_web>/mestrado/saqua/carga.php” .

Esta preparação do ambiente obedece à sequência de comandos presente no arquivo *instalar.bat*, na sub-pasta */mestrado/instala*, que após instalar o Apache, já pré-configurado, e o servidor PostgreSQL, criando os usuários “postgres” e “saqua”, executa o arquivo *cria_e_preenche_banco.php* e o arquivo *carga.php*, sem interação do usuário.

No caso de remoção existe também uma sequência de comando para remoção do sistema em “C:/mestrado/instala/desinstala.bat” que desinstala os aplicativos e remove as pastas.

2.6 A API do *Google Maps*

Conhecer as imediações do local de coleta pode ser um diferencial para buscar a fonte de um problema ambiental encontrado. Neste trabalho foram mostradas as imagens de satélites

¹⁶ <http://www.postgresql.org/>

próximas às medições pelas coordenadas geográficas, IBGE (1988), das estações de qualidade de água fornecidas.

Desta forma, são disponibilizadas imagens de satélite fornecidas através da API do *Google Maps*, onde é possível navegar em imagens mistas de satélites e mapas, partindo do ponto central das estações cadastrados, com as coordenadas convertidas para o formato decimal.

A API do *Google Maps* permite usar *JavaScript* para incorporar o Google Maps na página da *web*, fornecendo diversos utilitários para manipular mapas e adicionar conteúdo ao mapa através de diversos serviços.

A API do *Google Maps* é um serviço gratuito, atualmente em fase de testes, disponível para qualquer sítio que seja gratuito, ressaltando que sua utilização deve seguir os termos de uso presente no sítio do *Google Maps*¹⁷.

A divulgação da página também depende da chave de validação do mapa que estará amarrada na URL (“*Universal Resource Locator*”) definida na sua solicitação, a qual deverá estar direcionada para o endereço IP (“*Internet Protocol*”) do Servidor *Web*.

A **Figura 16** mostra uma estratégia simples de uso, com possíveis fluxos de acesso à informação da API do *Google Maps*, visando o uso de mapas e imagens de satélites dentro do Servidor *Web*, o qual também pode permitir o acesso direto do cliente, uma vez que este possua a chave de acesso.

A consulta ao Servidor de *Google Maps* (**Figura 16** – item 1) permite acessos que utilizem a API através da chave cadastrada na URL, encaminhando o mapa processado neste Servidor para o Servidor *Web* (**Figura 16** – item 2), que retorna seu código ao cliente (**Figura 16** – item 3).

Entretanto, isto não impede que o cliente execute os mesmos arquivos localmente em seu equipamento, em acesso direto (**Figura 16** – item 4). Porém ele não conseguirá disponibilizá-lo a menos que se torne um Servidor *Web*.

¹⁷ <http://code.google.com/intl/pt-BR/apis/maps/terms.html>

Estratégia simples de uso para acesso à API do Google Map.

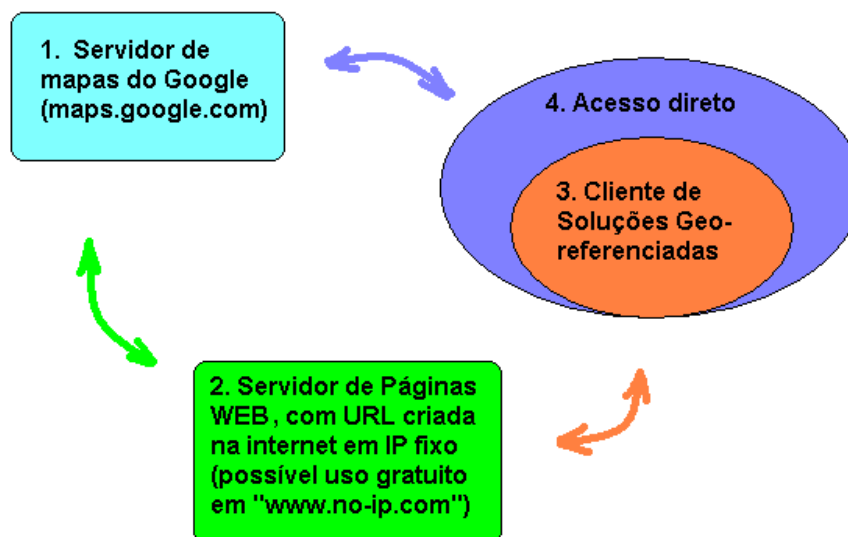


Figura 16 - Estratégia simples de uso

Na consulta proposta, o Cliente de Soluções Geo-referenciadas consulta as páginas WEB com a API do *Google Maps*, obtendo imagens processadas no Servidor de mapas do *Google Maps*, porém o código está no Servidor Web, ou seja, quando o Servidor Web é o próprio cliente o mapa não é direcionado, e vai direto ao cliente, desde que tenha informado uma chave válida.

A chave válida pode ser obtida através da URL <http://www.google.com/apis/maps>, onde será necessário associar um nome de servidor válido para a Internet, pela URL.

A estratégia proposta na **Figura 16** prevê que em instalações futuras outra chave válida possa ser criada apontando para o Servidor Web, cuja URL pode ser obtida gratuitamente no sítio www.no-ip.com, desde que se possua um endereço válido para Internet no servidor Web.

A chave utilizada no código deve ser substituída pela nova chave obtida, para que os clientes possam acessar, via servidor Web, os serviços do *Google Maps*. Foi criada uma interface integrada ao SAQUA, com imagens capturadas das estações, arquivo *google_estac_off.php*, que permite qualquer usuário visualizá-las, independente do acesso à Internet. Esta interface teve suas imagens capturadas por uma interface executada no cliente, com o arquivo *google_estacoes.html*.

Após acessar a interface com imagens capturadas e clicar sobre esta, é que se tem acesso à interface que utiliza a API do *Google Maps*, com o arquivo *google_estacoes.php*, identificando a imagem atual da estação e permitindo a interação. Esta interface funcionará sem a atualização da chave utilizada quando o cliente também for o Servidor Web.

Estes aplicativos foram criados, com as linguagens “PHP” e “HTML”, para selecionar imagens das estações capturadas, recebendo como parâmetro do SAQUA o código da estação e enviando suas coordenadas para acessar as imagens através da API do *Google Maps*.

Deve-se lembrar também que as imagens são constantemente atualizadas neste sítio, assim como também suas bibliotecas podem sofrer alterações, surgindo melhores formas de apresentação. No artigo “Como utilizar as API’s do Google Map”, Glória; Campelo; Ribeiro (2007) foi descrito como adaptar exemplos e assim utilizar novas ferramentas.

As coordenadas das estações são fornecidas no formato “grau:minuto:segundo” e precisam ser transformadas em decimais para o uso na API do *Google Maps*, através da conversão em decimais conforme mostra a **Tabela 13**, mantendo o sinal e aplicando a seguinte transformação:

$$/\text{grau em decimal}/ = / \text{grau}/ + \text{minuto}/60 + \text{segundo}/3600$$

Tabela 13 - Conversão das coordenadas geográficas para coordenadas em decimais

Estações de qualidade de água	Coordenadas geográficas (grau:minuto:segundo)		Coordenadas em decimais	
	Latitude	Longitude	Latitude	Longitude
Santa Branca	-23:22:08	-45:54:00	-23,368889	-45,900000
Pindamonhangaba	-22:54:40	-45:28:13	-22,911111	-45,470278
Queluz	-22:32:24	-44:46:22	-22,540000	-44,772778
Resende	-22:28:00	-44:26:43	-22,466667	-44,445278
Barra do Pirai	-22:26:58	-43:47:56	-22,449444	-43,798889
Juiz de Fora - Jusante	-21:46:42	-43:19:31	-21,778333	-43,325278
Anta G	-22:02:07	-42:59:27	-22,035278	-42,990833
Cataguases	-21:23:22	-42:,41:47	-21,389444	-42,696389
Santo.Antônio de Pádua II	-22:32:32	-42:10:50	-22,542222	-42,180556
S. Fidelis	-21:38:43	-41:45:08	-21,645278	-41,752222
Cardoso Moreira - RV	-21:29:14	-41:37:00	-21,487222	-41,616667
Campos - Ponte municipal	-21:45:12	-41:18:01	-21,753333	-41,300278

As coordenadas das estações em decimais são as utilizadas para centralizar as estações nas imagens obtidas na API do *Google Maps*, com o fator de aproximação que mais se adequar à imagem existente que será exibida.

3 ANÁLISE DOS RESULTADOS

3.1 A carga das séries históricas no SAQUA

O aplicativo *carga.php* efetua a carga das séries históricas, armazenadas nos arquivos *qualagua.txt*, para a base de dados identificando os alertas. Este aplicativo pode fazer uma carga coletiva de todas as estações ou uma carga individual pelo código da estação, excluindo os dados anteriores correspondentes ao tipo de carga solicitada.

Para fazer a carga coletiva de todas as estações, basta acessar o Servidor *Web* pelo endereço `http://<servidor_web>/mestrado/saqua/carga.php`, onde `<servidor_web>` é o nome ou o endereço IP do servidor. Esta carga também pode ser feita chamando este aplicativo por um interpretador PHP e direcionando a saída para um arquivo texto, conforme feito pelo arquivo *instala.bat*.

A carga coletiva de todas as estações retorna um histórico de acompanhamento muito extenso, por isto é preferível refazer a carga individual da estação para avaliar este histórico.

A carga individual da estação pode ser solicitada quando a estação estiver selecionada no SAQUA e escolher o link “Carga” disponível para a mesma. Será aberta uma nova janela, a qual mostrará a quantidade de registros excluídos, os arquivos e parâmetros que serão carregados, sua identificação a cada linha e as ações resultantes.

Um pequeno exemplo de carga individual foi gerado ao selecionar a estação Juiz de Fora, código “58480500”, conforme ilustrado na **Figura 17**, através do endereço: `“http://<servidor_web>/mestrado/saqua/carga.php?cod_estac=58480500”`.

Observando a **Figura 17**, a primeira linha desta janela mostra a ocorrência de dados levantados às 10h e 15min do dia 19/08/2002 no arquivo *qualagua.txt*, onde foram encontrados dados para os parâmetros de Oxigênio Dissolvido (código 19), Ph da água (código 11) e Demanda Bioquímica de Oxigênio (código 18), o que causou inclusões de registros de alertas, de ocorrências anteriores, elevando o sub-total de alertas para 611.

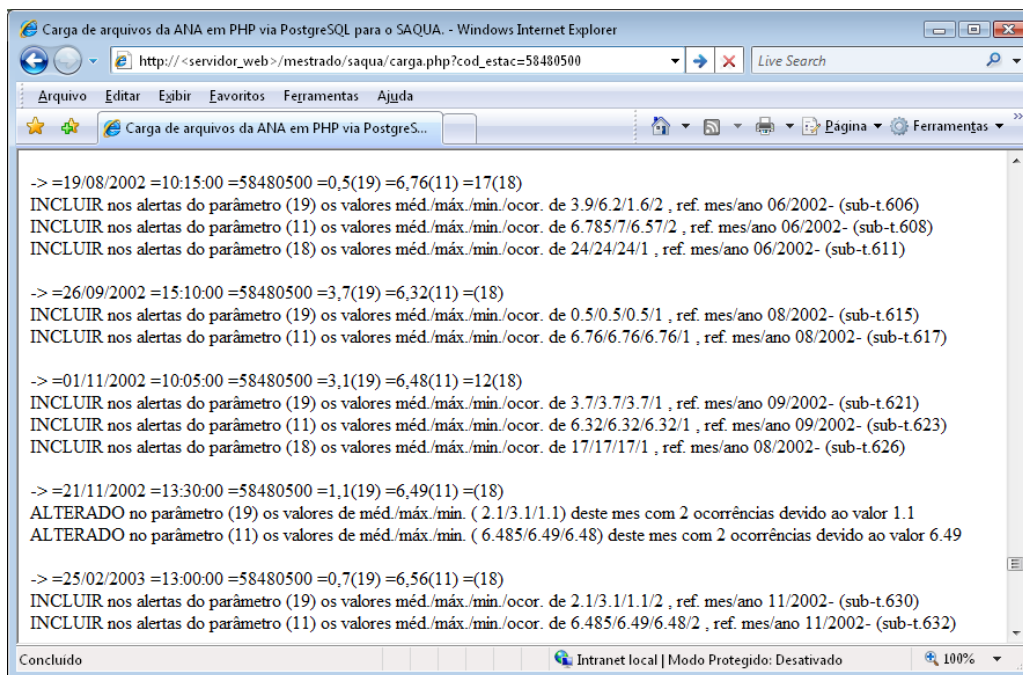


Figura 17 - Tela carregando parte dos dados da Estação Queluz

Seguindo para a próxima linha de ocorrência de dados levantados no dia 26/09/2002, foram encontrados dados para Oxigênio Dissolvido, com 4 (quatro) classes cadastradas, e Ph da água, com 2 (duas) classes cadastradas, o que fez que os dados cadastrados na primeira linha para estes parâmetros fornecessem 6 (seis) inclusões e o sub-total de alertas chegasse a 617.

Na próxima linha de ocorrência, em 01/11/2002, além de ambos parâmetros da linha anterior que forneceram 6 (seis) inclusões, surgiu dados para o parâmetro de Demanda Bioquímica de Oxigênio, com 3 (três) classes cadastradas, elevando o sub-total de alertas para 626, incluindo desta forma alertas nos meses de setembro e agosto de 2002.

Porém na linha seguinte de ocorrência, em 21/11/2002 não houve inclusão de alertas, pois os dados preenchidos no mesmo mês alterou apenas os valores para as estatísticas de alertas dos parâmetros encontrados, que são incluídos na linha seguinte de ocorrência, em 25/03/2003, referente ao mês da ocorrência.

Este processo de carga inicia removendo os registros de alertas das estações a serem carregadas, iniciando o subtotal com valor 0 (zero), prosseguindo conforme relatado acima, e segue até finalizar cada arquivo de carga, quando inclui os alertas não encontrados.

No caso da estação possuir dados mesclados, o início dos dados do próximo arquivo é que serão considerados, removendo qualquer dado desta estação em data igual ou superior, junto com o

dado do lançamento imediatamente anterior de cada parâmetro, o qual é incorporado e re-lançado de acordo com as ocorrências encontradas.

Esta exposição da carga dos dados também pode ser armazenada em um arquivo temporário, ao invés de ir direto para a tela, como ocorre no procedimento do arquivo *instala.bat*, isto torna o processamento mais rápido por não exibir linha a linha, porém na tela o usuário pode interromper o processo, forçando uma situação problema, e acompanhar diretamente a carga.

3.2 Dados exibidos no SAQUA

Após carregar as séries históricas de qualidade de água no Sistema HIDRO para as estações Cataguases e Santo Antonio de Pádua II e efetuar consultas gráficas sobre o parâmetro de oxigênio dissolvido, utilizando o seu Manual do Usuário, ANA (2002), chega-se na **Figura 18**.

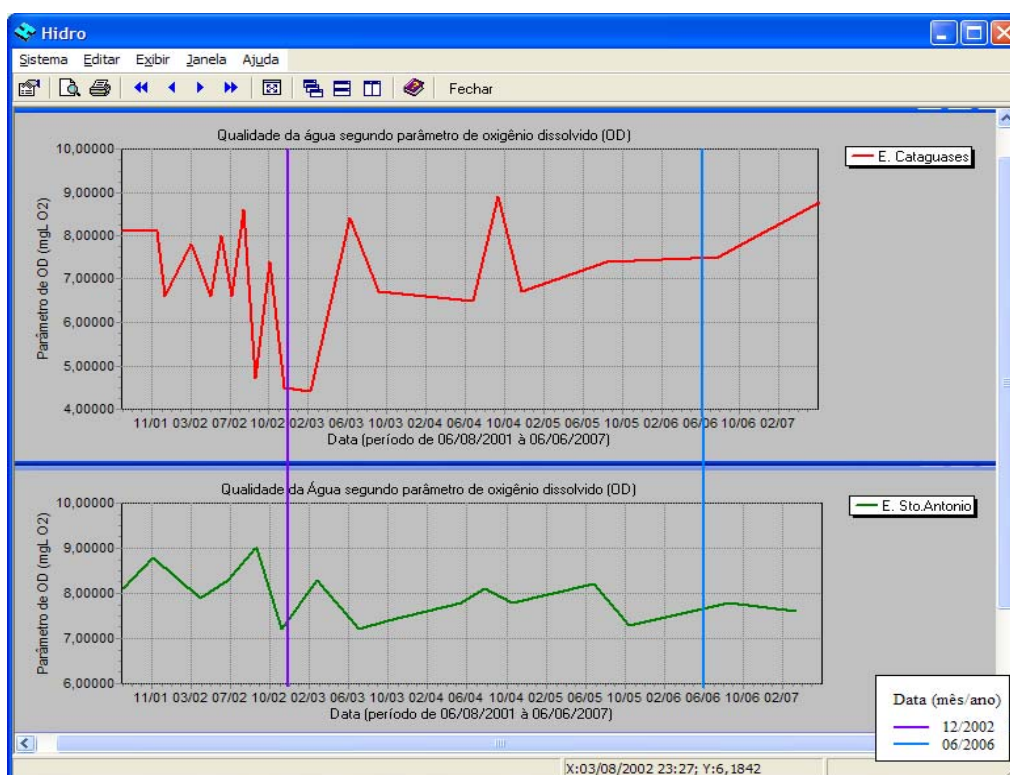


Figura 18 - Série histórica do parâmetro de Oxigênio Dissolvido pelo Sistema Hidro.

A tela inicial do aplicativo SAQUA foi apontada para o parâmetro de oxigênio dissolvido, quando clicado sobre a estação Cataguases, primeira região em vermelho vista no mapa de cima

para baixo, o aplicativo *saqua.php* expõe parte do gráfico para esta estação, conforme a **Figura 19**, o qual pode ser comparada à **Figura 18**, antes do corte de dezembro de 2002.

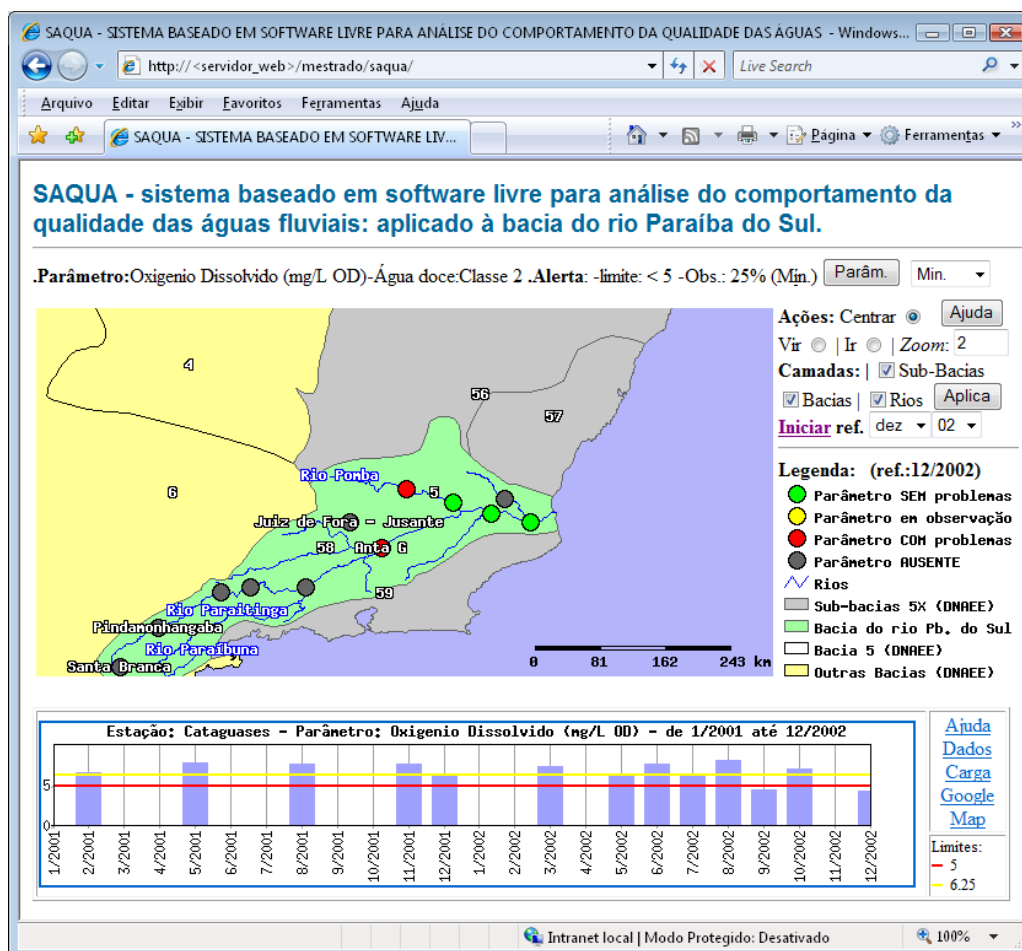


Figura 19 - Tela do SAQUA mostrando gráfico da Estação Cataguases

Embora o gráfico do SAQUA mostre o período de 2 (dois) anos anteriores incluindo o mês de referência, ele permite alterar o mês de referência, ressaltando que as cores colocadas no gráfico do Sistema Hidro foram selecionadas e não identificadas como alertas, o que ocorre no SAQUA. Existe a opção de alterar a estatística, para que se tenha 1 (um) valor que identifique o alerta no mês de referência.

No caso exemplificado na **Figura 19** foi utilizada a estatística de valores mínimos (Min.) para cada mês, e que geram alertas para o parâmetro de Oxigênio Dissolvido quando o valor é inferior a 5 mg/L OD e entram em estado de observação quando este valor for igual ou acima deste e inferior a 6,25 mg/L OD (25% acima do valor de alerta). Também existe um link para ampliação do gráfico em barras, ou seja, clicando sobre o mesmo será criado em uma nova janela este mesmo gráfico para permitir reconhecer melhor os valores de alertas e avisos, conforme mostrado na **Figura 20**.

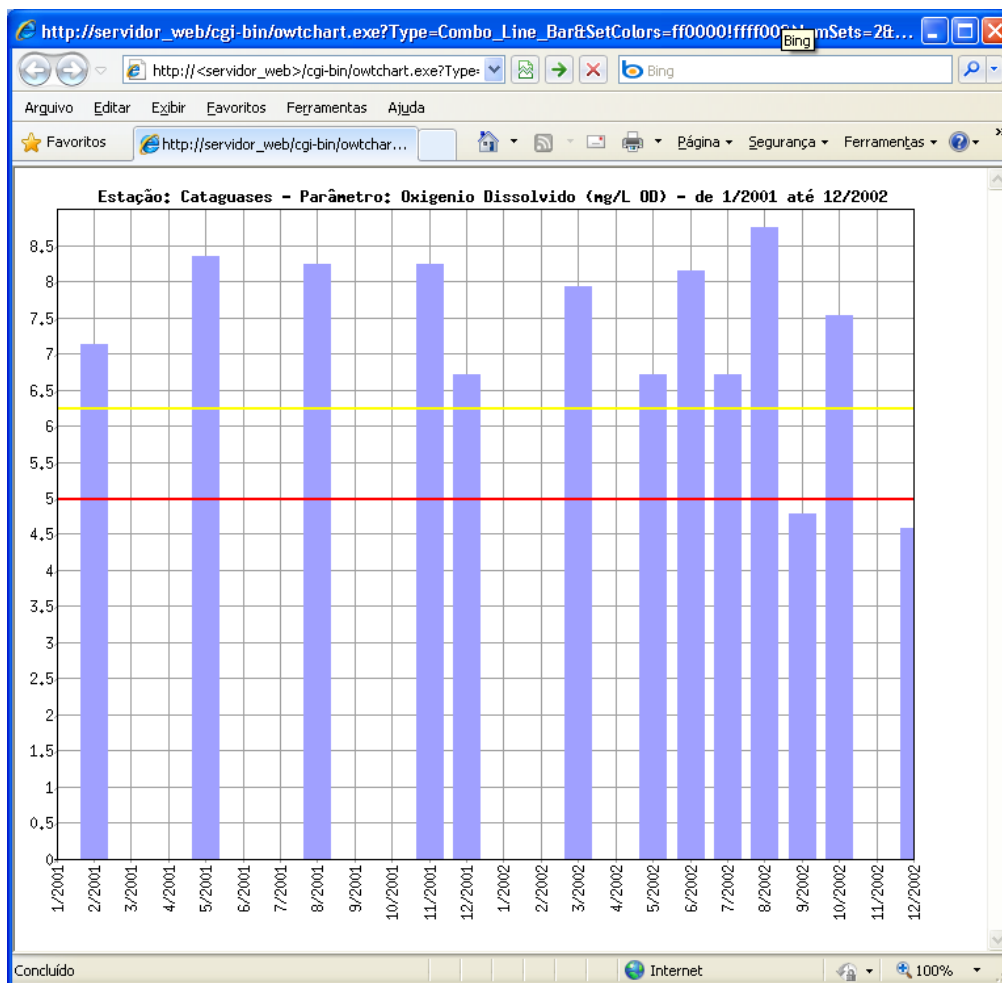


Figura 20 - Tela mostrando a ampliação do gráfico da Estação Cataguases

A fim de explorar um pouco mais os recursos construídos para identificar o histórico de alertas, será exibido o parâmetro “Ph da água”, utilizando uma classe cadastrada como “V.Baixo”, uma vez que foram cadastradas as classes “V.Baixo” e “V.Alto”, pois o valor Ph é comum para todas classes e se torna crítico quando alto ou baixo.

Para isto foram alterados o parâmetro, a classe e a estação escolhida, ao clicar no botão “Parâm”, escolher o parâmetro “Ph da água”, em seguida clicar no botão classe, escolher a classe “V.Baixo” e clicar no mapa sobre o círculo de alerta da estação Santo Antonio de Pádua II, que é a próxima estação à jusante da estação Cataguases no mapa. Após isto surgirá o gráfico desta estação e ao clicar no link “Dados”, existente na lateral do gráfico, chega-se na tela da **Figura 21**.

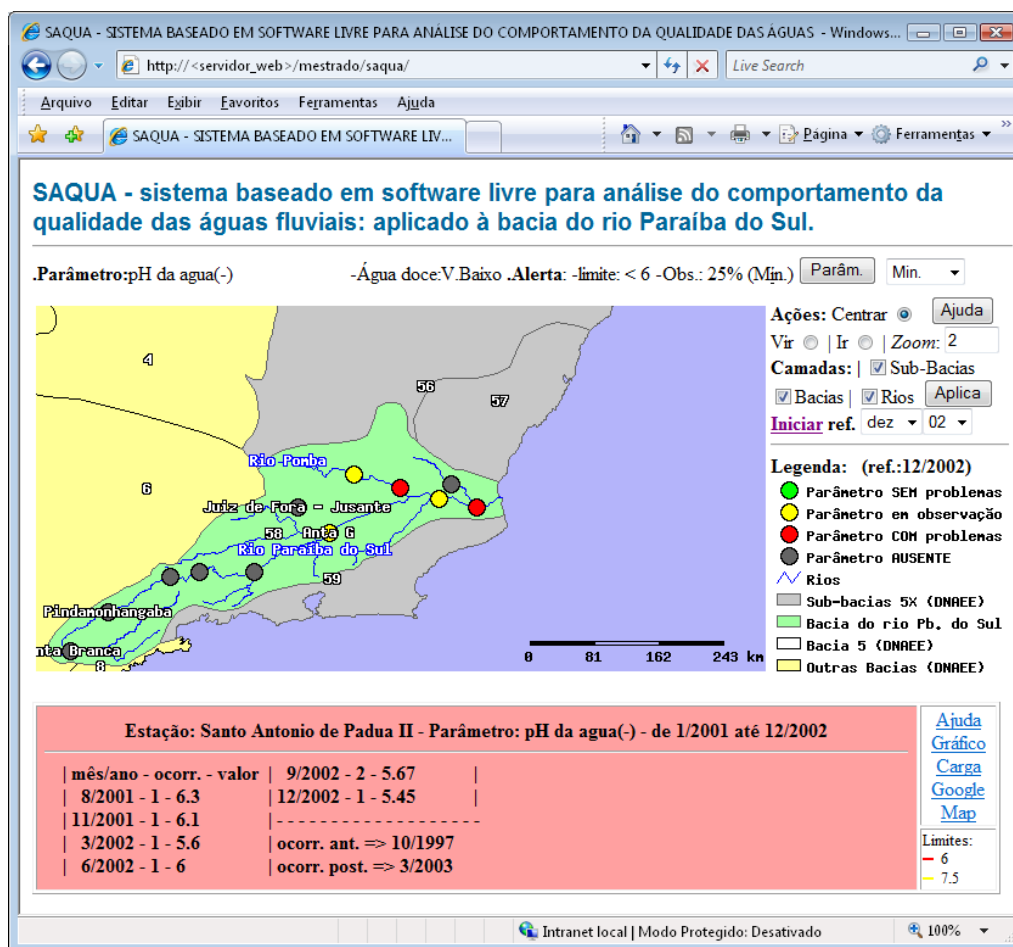


Figura 21 - Tela do SAQUA com dados da Estação Santo Antonio de Pádua II

Na **Figura 21** observa-se o período de 2 (dois) anos antes da data de referência com o número de ocorrência e o valor da estatística selecionada em cada mês. Quando o valor de ocorrência é 1 (um), os valores para as estatísticas de mínima, média e máxima são iguais ao valor encontrado. Também é identificado o mês onde foi encontrada uma ocorrência anterior à primeira ocorrência (“ocorr.ant.”), e o mês que foi encontrado uma ocorrência posterior à última ocorrência (“ocorr.post”), para se conhecer a distância do dado mais próximo à este período de 2 anos.

Na tela da **Figura 21** verifica-se que tanto as estações Santo Antônio de Pádua II como a estação Campos-Ponte Municipal tiveram problemas significativos quanto ao baixo valor do Ph da água, e não haviam apresentado níveis críticos quanto ao parâmetro de Oxigênio Dissolvido, como a estação Cataguases, conforme mostrado na **Figura 19**.

Ocorrências como estas podem observadas com o auxílio do SAQUA. Entretanto, o interesse aqui é mostrar o potencial e as limitações desta ferramenta. As principais limitações são os dados escassos de alguns parâmetros e o potencial de uso pode ser ampliado para outras regiões.

Em um período próximo ao exemplo anterior, foi utilizado o parâmetro de Demanda Bioquímica de Oxigênio (DBO), mudando o mês de referência de dezembro para novembro, a estatística máximo e explorando um pouco mais dos recursos gráficos disponíveis, conforme explicado a seguir, chegou-se na **Figura 22**.

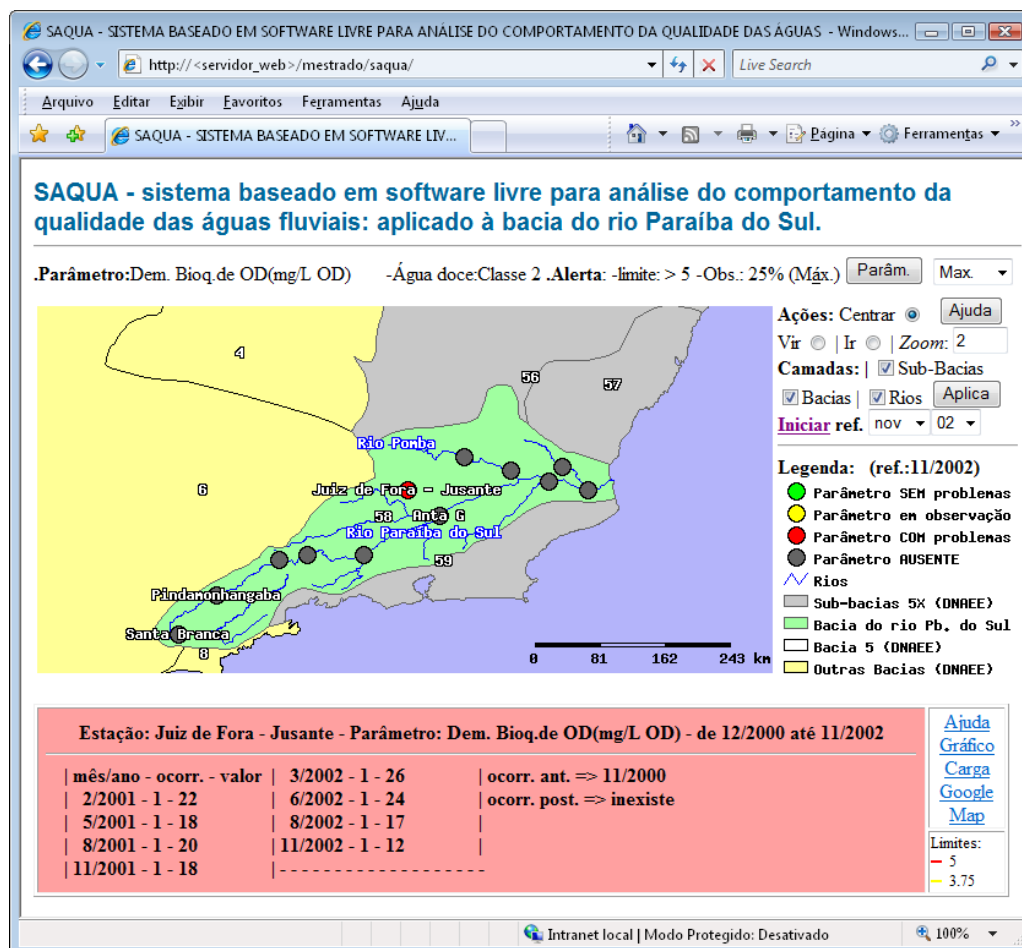


Figura 22 - Tela do SAQUA, em maior escala, da Estação Juiz de Fora-Jusante

A tela da **Figura 22** foi obtida partindo da tela criada ao clicar no link Iniciar. Ao clicar no botão "Parâm", escolher o parâmetro "Demanda Bioq. de O₂", em seguida clicar no botão "Classe" que surge após o parâmetro, escolher "Classe 2", depois selecionar a estatística para "Max", uma vez que este limite se refere a valores maiores do que 5 (cinco) e deseja-se consultar o pior caso, selecionar o botão "Ir", colocar o valor 4 (quatro) no campo do "Zoom", trocar a referência (ref.) para o mês de novembro (nov) de 2002 (02), a seguir clicar sobre a estação "Juiz de Fora - Jusante" Santa Branca, esperar o surgimento do gráfico, e clicar sobre o link "Dados" que surge após o gráfico.

Nota-se a escassez de dados sobre este último parâmetro em relação às estações selecionadas, principalmente em datas mais recentes. Na **Figura 22**, sabe-se que o último valor

cadastrado foi neste mês de novembro de 2002, e não existe ocorrência posterior cadastrada até a data atual. Ao selecionar cada uma das estações selecionadas e verificar o link “Dados” das mesmas verifica-se que esta mesma situação de ausência de dados deste parâmetro para datas posteriores a novembro de 2002, até pelo menos as séries históricas fornecidas na atual data.

3.3 Imagens de satélite das estações

As imagens de satélite utilizadas no Google Maps são construídas a partir de fotografias de satélite obtidas de fontes diversas, imagens aéreas (fotografadas de aeronaves) e Sistemas de Informações Geográficas em 3 (três) Dimensões. A maior parte da superfície da terra, exibido em serviços de mapeamento *web*, como o Google Maps, é baseado na melhoria e na cor equilibrada de imagens do satélite Landsat 7, lançado em 15 de abril de 1999.

As imagens das estações fluviométricas são apresentadas através de consultas dos aplicativos do SAQUA pela API do Google Maps. Para isto é centrada as coordenadas da estação nas imagens de satélite.

Foi feita uma captura individual de cada imagem das estações, no formato híbrido, para que, quando não houver acesso à Internet, tenha-se esta imagem para consulta. Não havendo problemas com acesso à Internet e o Servidor *Web* estando devidamente configurado, basta clicar sobre a imagem capturada para obter os recursos da API do *Google Maps*.

Ao selecionar no SAQUA a estação desejada, por exemplo a estação Santa Branca, clicando sobre a mesma na tela inicial, e desejar saber onde ela fica por uma imagem de satélite basta clicar sobre a opção “G.Map”, e surgirá outra janela para navegação, com a imagem capturada da estação, conforme mostrado na

Figura 23.



Figura 23 - Imagem de satélite capturada da Estação Santa Branca (dez./2007)

Outra forma de obter esta imagem na API, sem necessitar clicar na imagem capturada, é acessar uma interface inicial independente do SAQUA, mostrada na **Figura 24**, acessando o endereço “http://<servidor_web>/mestrado/saqua/google_estacoes.html”, e clicar em “Santa Branca”.

No decorrer do desenvolvimento deste trabalho houve uma melhoria na qualidade da imagem disponibilizada de algumas estações, como por exemplo a imagem da estação Santa Branca, fato este que pode ser observado ao se comparar a

Figura 23 com a **Figura 24**.

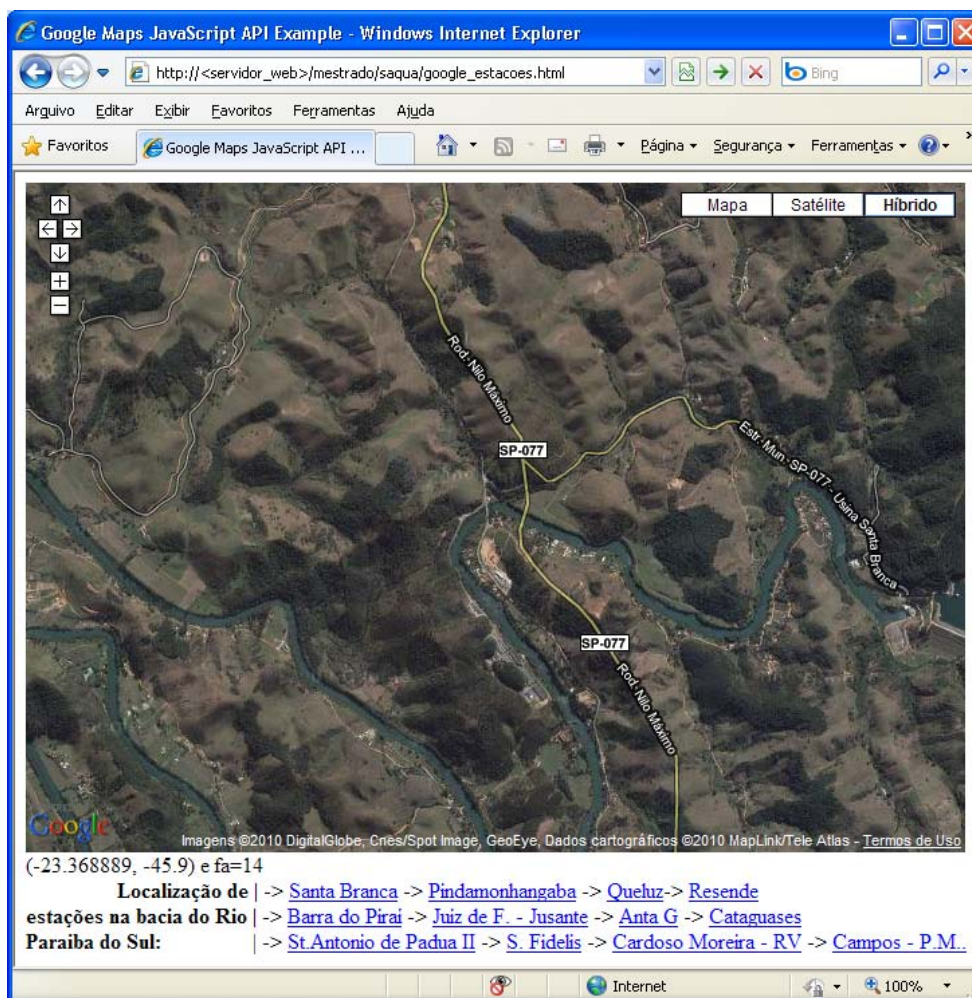


Figura 24 - Imagem da Estação Santa Branca na API do Google Maps (out./2009)

Com isto foi possível utilizar um fator de aproximação maior, mudando de 14 para 18 vezes, ao clicar 4 (quatro) vezes no sinal de mais (+) mostrado na tela da **Figura 24**, obtendo, desta forma, uma tela com uma riqueza maior de detalhes mostrada na **Figura 25**, obtida no endereço “http://<servidor_web>/mestrado/saqua/google_estacoes.php”, chamado a partir do SAQUA.

A API do *Google Maps* permite combinar as imagens de satélite e o mapa da região, através da opção “Híbrido”, usada nas imagens anteriores, fornecendo também opções de aproximação e afastamento, além de setas para navegação, como também os impressos de fontes e termos de uso¹⁸.

A estação Santa Branca foi escolhida, como exemplo, por ser facilmente identificada mais ao sul da amostra e ter ocorrido melhoria na qualidade da imagem. As imagens das 3(três) últimas figuras validam também os aplicativos desenvolvidos nos arquivos *google_estac_off.php*, *google_estacoes.html* e *google_estacoes.php*, respectivamente.

¹⁸ <http://code.google.com/intl/pt-BR/apis/maps/terms.html>



Figura 25 - Imagem de satélite da Estação Santa Branca (out./2009)

Para que ocorram medições, as estações fluviométricas devem estar localizadas próximas de rios, conforme ilustra a **Figura 26**, e também em alguns casos próximos a pontes, como na maioria das estações cadastradas e visualizadas no SAQUA.

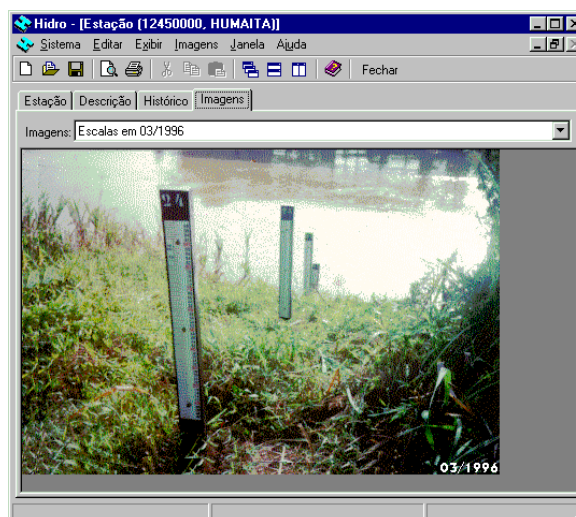


Figura 26 - Imagem de uma estação (ANA, 2002, p.23)

3.4 Cadastro de demais parâmetros

Quando o SAQUA foi instalado os parâmetros OD, Ph e DBO foram cadastrados conforme mostra a **Tabela 14**, ou seja, foram colocados no bando de dados estes parâmetros, através de seus códigos, os códigos da classe de enquadramento em água doce pertencente e outras para simulação, as condições de estado crítico com seu valor limite de alerta segundo a resolução CONAMA 356 e o limite porcentual para o estado em observação em relação ao valor limite de alerta.

Tabela 14 – Limites de parâmetros cadastrados no SAQUA segundo o CONAMA

Código do Parâmetro	Código da Classe	Condição	Valor limite de alerta	Limite % em observação
19 (Oxigênio dissolvido)	3 (Classe 2)	< (menor que)	5 (5 mg/L O ₂)	25 (25%)
19 (Oxigênio dissolvido)	2 (Classe 1)	< (menor que)	6 (6 mg/L O ₂)	25 (25%)
19 (Oxigênio dissolvido)	4 (Classe 3)	< (menor que)	4 (4 mg/L O ₂)	25 (25%)
19 (Oxigênio dissolvido)	5 (Classe 4)	< (menor que)	2 (2 mg/L O ₂)	25 (25%)
18 (Dem. Bioq. de O ₂)	3 (Classe 2)	> (maior que)	5 (5 mg/L O ₂)	25 (25%)
18 (Dem. Bioq. de O ₂)	2 (Classe 1)	> (maior que)	3 (3 mg/L O ₂)	25 (25%)
18 (Dem. Bioq. de O ₂)	4 (Classe 3)	> (maior que)	10 (10 mg/L O ₂)	25 (25%)
11 (Ph da água)	11 (V.baixo)	< (menor que)	6 (6)	25 (25%)
11 (Ph da água)	12 (V.alto)	> (maior que)	9 (9)	25 (25%)

Para efetuar o cadastro de outros parâmetros com as condições estabelecidas pelo CONAMA, para a classe que desejar consultar, podem ser utilizados comandos do PostgreSQL, através de sua interface, assim como também removê-los, conforme exemplificado a seguir.

Desejando também acompanhar no SAQUA o parâmetro de cromo total, código 125, cuja condição estabelecida pelo CONAMA para crítico na classe 2 é ser maior do que 0,05 mg/L Cr, pode-se cadastra-lo da seguinte forma no Servidor PostgreSQL:

Partindo do botão “Iniciar” do Windows, opção “Todos Programas”, na pasta da versão do PostgreSQL instalado, no caso “PostgreSQL 8.2”, selecionar “psql to ‘postgres’”. Com isto será pedido a autenticação do usuário postgres, e efetua-se as linhas de comando que inserem estes dados, conforme mostrado na **Figura 27**.


```

C:\> psql to 'postgres'
Password for user postgres:
Welcome to psql 8.2.4, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

Warning: Console code page (850) differs from Windows code page (1252)
        8-bit characters may not work correctly. See psql reference
        page "Notes for Windows users" for details.

postgres=# \c saqua
You are now connected to database "saqua".
saqua=# \d param
          Table "public.param"
  Column  | Type          | Modifiers
-----+-----+-----
 cod_param | integer       |
 unidade  | character varying(7) |
 parametro | character varying(20) |

saqua=# \d criterio
          Table "public.criterio"
  Column  | Type          | Modifiers
-----+-----+-----
 cod_param | integer       |
 cod_class | integer       |
 cond      | character varying(2) |
 limite    | real          |
 aviso     | real          |

saqua=# insert into param values ( 125, 'mg/L Cr', 'Cromo Total');
INSERT 0 1
saqua=# insert into criterio values ( 125, 3 , '>', 0.05, 25);
INSERT 0 1
saqua=# \q

```

Figura 27 - Comandos no PostgreSQL para criar parâmetro de Cromo Total

Após conectar a base de dados “saqua” foram exibidos os campos da tabela “param” como “cod_param”, “unidade” e “parametro” e os campos da tabela “critério” como “cod_param”, “cod_class”, “cond”, “limite” e “aviso”, respectivamente. Esta ordem foi utilizada para a inclusão dos dados conforme pode ser notado pelos comandos.

O acesso a base de dados ocorreu através do comando psql “\c saqua”, assim como a exibição dos campos das tabelas “param” e “critério” pelos comandos “\d param” e “\d critério” respectivamente. Os valores foram incluídos nestas tabelas, respeitando a sequência dos campos, através dos seguintes comandos SQL: “insert into param values (125, 'mg/L Cr', 'Cromo Total');” e; “insert into criterio values (125, 3 , '>', 0.05, 25);”. Esta seção é encerrada ao executar o comando psql “\q”.

A próxima etapa será a de efetuar a carga dos dados de todas estações a fim de atualizar os dados cadastrados do monitoramento deste parâmetro. Isto pode ser feito diretamente acessando a carga pelo endereço “http://<servidor_web>/mestrado/saqua/carga.php/” e verificado seu termino ao percorrer a barra de rolamentos até o final, conforme ilustra a **Figura 28**.

Em qualquer caso de carga de dados no SAQUA é desejável avisar aos demais usuários para que dois ou mais usuários não efetuem simultaneamente carga de dados na mesma estação, pois uma carga iria interromper a outra, o que seria solucionado com uma nova carga de dados.

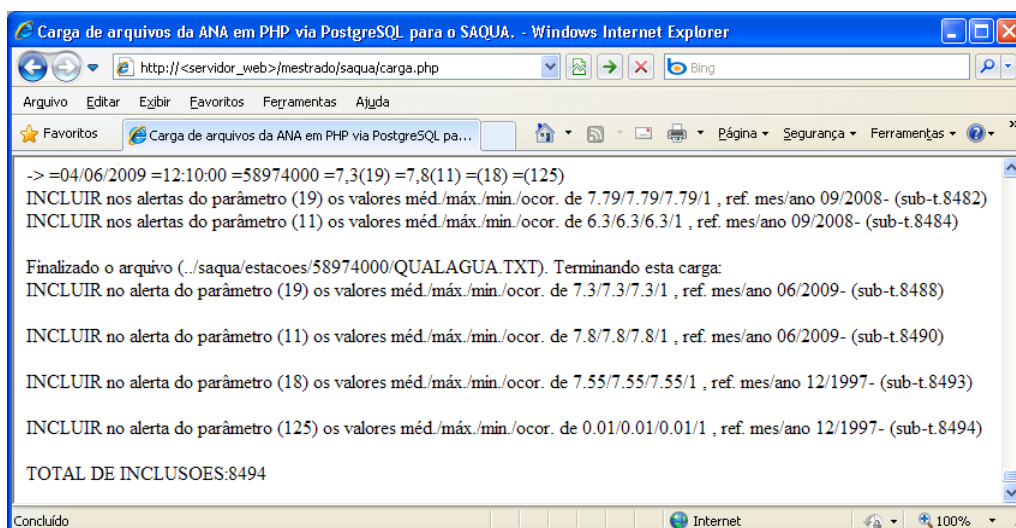


Figura 28 - Tela mostrando o final da carga dos dados de todas estações

Para mostrar a inclusão efetuada, pode-se através do SAQUA, clicar no botão “Param” e selecionar o parâmetro “Cromo Total”, a seguir selecionar o ano de referência para o ano de 1995, mantendo o mês de dezembro, e clicar sobre a estação AntaG, obtendo a **Figura 29**.

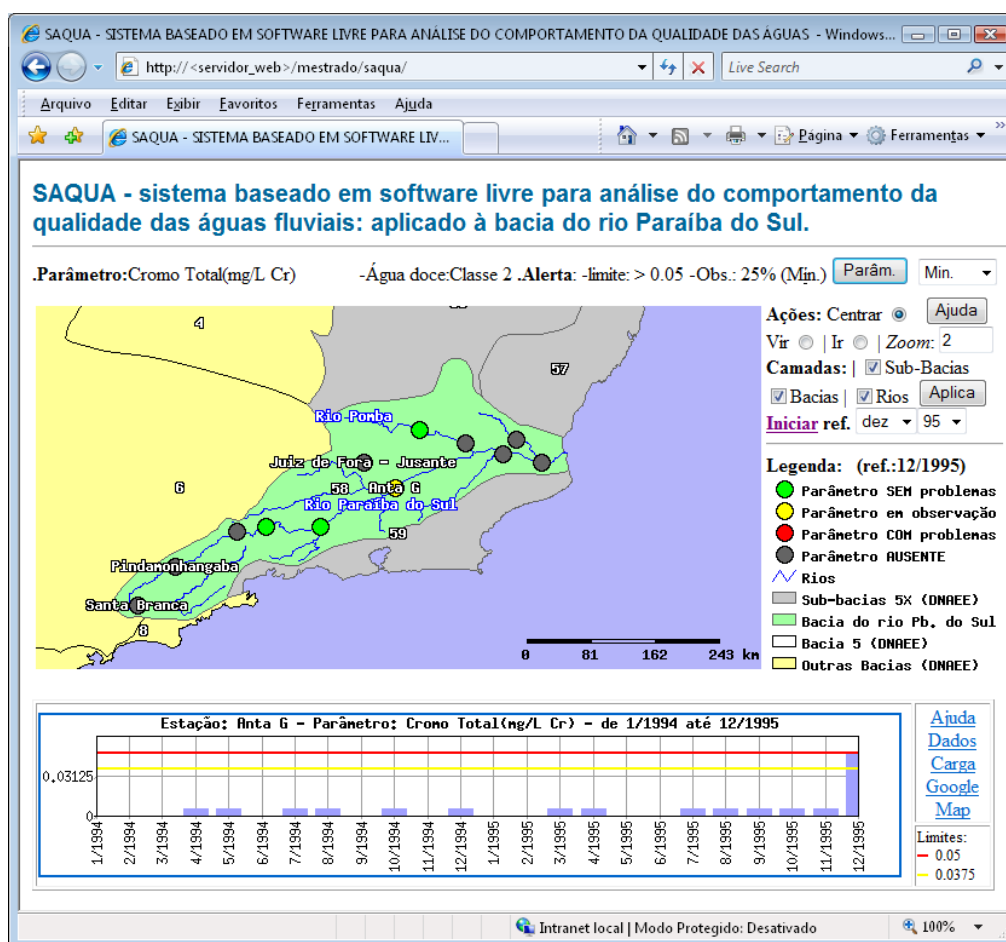
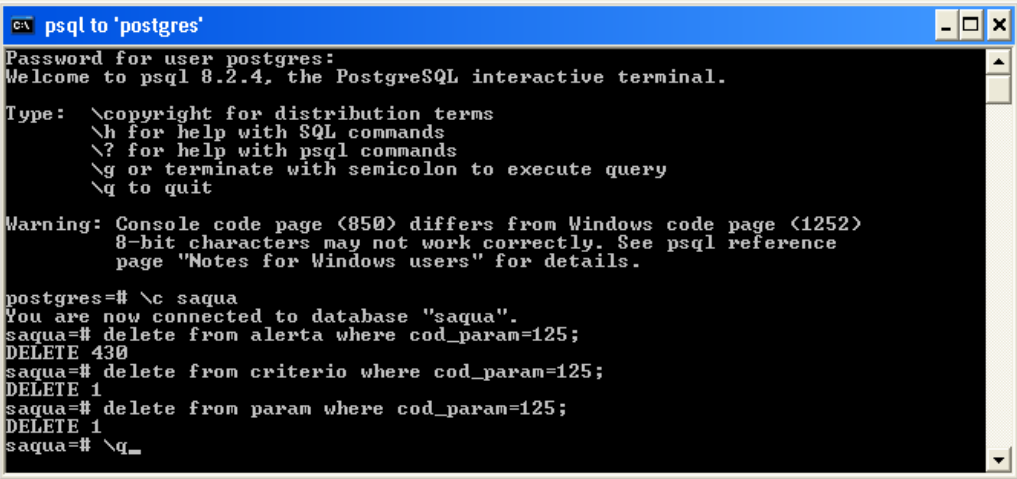


Figura 29 - Tela do SAQUA mostrando dados de parâmetros de Cromo Total

Ao alterar o período para um mês acima de março de 1998 não se encontrou mais valores disponíveis, ou seja, todas as estações informam não terem dados disponíveis até a data atual. Caso se deseje excluir parâmetros sem valores atuais como o de Cromo Total, basta conectar a base de dados do SAQUA e eliminar o parâmetro de Cromo Total, identificado pelo código 125, nos registros gerados nas tabelas “alerta”, “criterio” e “param”, conforme mostrado na **Figura 30**.



```
psql to 'postgres'
Password for user postgres:
Welcome to psql 8.2.4, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

Warning: Console code page (850) differs from Windows code page (1252)
        8-bit characters may not work correctly. See psql reference
        page "Notes for Windows users" for details.

postgres=# \c saqua
You are now connected to database "saqua".
saqua=# delete from alerta where cod_param=125;
DELETE 430
saqua=# delete from criterio where cod_param=125;
DELETE 1
saqua=# delete from param where cod_param=125;
DELETE 1
saqua=# \q
```

Figura 30 - Comandos no PostgreSQL para o parâmetro de Cromo Total

Novamente, utilizou-se a interface do PostgreSQL, através do botão “Iniciar” do Windows, opção “Todos Programas” e pasta da versão do PostgreSQL instalado selecionar “psql to ‘postgres’”. A seguir autentica-se o usuário postgres, faz-se o acesso a base de dados, através do comando “\c saqua”, e exclui-se os registros através dos seguintes comandos: “delete from alerta where cod_param=125;”, “delete from criterio where cod_param=125 ;” e “delete from param where cod_param=125;”. Esta seção é encerrada ao executar o comando “\q”.

Desta forma é possível, com o conhecimento do SAQUA, selecionar os parâmetros que se deseja monitorar e verificar se os mesmos poderão atender as expectativas desejadas.

Conforme o exemplo aqui colocado, pode-se facilmente incluir qualquer parâmetro existente nas séries históricas de qualidade de água, com o cuidado de verificar as características de outras séries históricas, reutilizar o aplicativo para efetuar consultas a parâmetros que possam ser delimitados por outras referências legais.

4 CONCLUSÕES E SUGESTÕES

Através de recursos existentes em software livre, dados e ferramentas disponibilizadas gratuitamente na internet, o SAQUA reduz a escassez no que se refere a aplicativos para análise de séries temporais de qualidade de água, disponibilizando uma interface amigável ao usuário.

Ao disponibilizar um comparativo da evolução histórica de parâmetros físico-químicos de qualidade de água com valores esperados pelo CONAMA, o SAQUA permite exibir, armazenar e confrontar os valores mensais desses parâmetros com os limites das classes de uso de água, com dados geo-referenciados das estações fluviométricas de qualidade de água.

Espera-se que esta interface possa ser útil a pesquisadores e órgãos gestores de recursos hídricos, uma vez que não há custos significativos envolvidos e existe uma contribuição com a atividade de acompanhamento de parâmetros históricos de qualidade de água.

Não foi apenas através dos códigos do SAQUA que este trabalho procurou contribuir, mas também através de novas idéias, utilizando antigos conceitos de reutilização de software e tendências de aplicações, visando atender um público alvo de pesquisadores e pessoas interessadas em acompanhar o impacto dos parâmetros de qualidade de água pela atuação de diversas gestões.

O SAQUA possui suas funcionalidades em software livre, e acompanha a tendência de outros aplicativos que também utilizam a interface do Mapserver para Windows e integrações com o Google Map, como por exemplo o “i3geo”, lançado em setembro de 2007, que é a “Interface Integrada de Ferramentas de Geoprocessamento para Internet” do Ministério do Meio Ambiente.

Com a intenção de contribuir ainda mais na gestão de recursos hídricos, pode-se dar continuidade ao SAQUA, ampliando seu uso e estendendo o seu estudo a qualquer sub-bacia, com melhorias na sua codificação, como por exemplo o uso de linguagem de estilo, feições trabalhadas e interface para parâmetro com a carga seletiva destes.

No caso deste trabalho ser utilizado para atender vários usuários a atualização das séries históricas deveria ser efetuada por um administrador, que coloque as séries históricas da ANA na base do SAQUA e execute um segmento de código para efetuar a carga e armazenar este resultado, alterando a opção de carga no SAQUA para visualização do resultado desta carga, eliminando situações problemas existentes em se efetuar cargas simultâneas.

A disponibilidade de dados, que são obtidos no sítio da ANA se refere às coletas divulgadas por diversas empresas, onde os dados são muitas vezes escassos ou inexistentes para determinados parâmetros. Uma forma para facilitar a atualização destes dados seria ter um acesso cliente-servidor à base de dados do Sistema HIDRO, onde outras funcionalidades também poderiam ser construídas, porém envolveria interesse e compromisso de ambas as partes.

Pode-se estender o estudo a qualquer sub-bacia, ou até mesmo em mais estações desta bacia que não faça parte das feições fornecidas, através da reconstrução da feição utilizada para expor os alertas, a feição “agua_qualidade5”, por meio de uma ferramenta GIS.

Melhorias gráficas também podem ser implementadas, pois uma alteração da resolução gráfica, ou até mesmo a compensação através de aproximações na interface *web*, pode interferir no tamanho dos caracteres uma vez que não foi utilizada uma linguagem de estilo, como a CSS (*Cascading Style Sheets*), para definir a apresentação em HTML.

No próprio banco de dados e na interface com o usuário existe a possibilidade de se fazer melhorias, por exemplo, através da inclusão de campos chaves que facilitem a crítica na inclusão e exclusão de parâmetros e limites, assim como na revisão da atual carga para que faça atualização apenas de parâmetros alterados, lembrando que já são filtrados apenas os parâmetros que estão cadastrados na base de dados do SAQUA.

É interessante utilizar ferramentas de software livre com boa aceitação e que futuramente possam ser integrados a outros sistemas, como por exemplo, na construção de um sistema que encontre históricos de inundações e estude seu efeito sobre a saúde da população, com dados de estações pluviométricas e fluviométricas, agregando informações de regiões inundadas e de possíveis doenças as quais possam ser associadas a alterações de parâmetros de qualidade de água.

Os parâmetros de qualidade de água, individualmente para a avaliação de uma situação em si não são tão considerados como os índices gerados através da composição dos mesmos, porém muitas investigações podem ser feitas através desta avaliação individual, e para reforçar esta forma de avaliação quantitativa é que foi criado e desenvolvido o SAQUA.

REFERÊNCIAS

ACHOUR, M. at al Manual do PHP ed by Philip Olson, 2007. Disponível em: <<http://www.php.net/docs.php>> Acesso em: 04/11/07.

BRASIL, Lei número 9.433, de 8 de janeiro de 1997 . Disponível em: <<http://www.aneel.gov.br/cedoc/blei19979433.pdf>> Acesso em: 23/11/2009.

BRASIL, Lei número 6.938, de 31 de agosto de 1981 . Disponível em: <<http://www.planalto.gov.br/CCIVIL/LEIS/L6938org.htm>> Acesso em: 07/12/2009.

BRASIL, Lei número 9.984, de 17 de julho de 2000 . Disponível em: <http://www.e.gov.br/defaultCab.asp?idservinfo=19403&url=http://www.cnrh.gov.br/sitio/index.php?option=com_content&view=article&id=17:lei-no-9984-de-17-de-julho-de-2000&catid=18:leis&Itemid=8> Acesso em: 23/11/2009.

AGÊNCIA NACIONAL DE ÁGUAS, 2002. HIDRO - Manual do Usuário. Disponível em: <<http://SistemaHIDROWEB.ana.gov.br/SistemaHIDROWEB.asp?TocItem=6010>> Acesso em: 18/07/07.

AGÊNCIA NACIONAL DE ÁGUAS/AGEVAP, 2006. A cobrança pelo uso da água nas bacias dos rios Paraíba do Sul e PCJ em 2006 - Avaliação e propostas de aperfeiçoamento. Disponível em <www.ana.gov.br/GestaoRecHidricos/CobrancaUso/_ARQS-Estudos/BPS/Cobranca_agua_PBSul.pdf> Acesso em: 06/01/2008

COUGO, P. Modelagem Conceitual e Projeto de Banco de Dados. Rio de Janeiro: ed. Abdr, 1997.

DA HORA, M.; MARQUES, E. SAD-RH: Sistema Generalizado de Apoio à Decisão na Gestão de Recursos Hídricos. Niterói, RJ: UFF/FEC, 2010.

ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. São Paulo: Peason Addison Wesley, 2005.

GLÓRIA, A. I.; CAMPELO, S. S.; RIBEIRO, J. A. Como utilizar as API's do Google Map. In: Simpósio Brasileiro de Sensoriamento Remoto (SBSR), 13., 2007, Florianópolis. Anais... São José dos Campos: INPE, 2007, p. 5787-5794. CD-ROM. ISBN 978-85-17-00031-7

GLÓRIA, A. I.; RIBEIRO, J. A.; PIMENTEL DA SILVA, L. Servidor Web para acompanhamento da evolução de parâmetros físico-químicos de qualidade das águas. In: XXIII Congresso Brasileiro de Cartografia, 2007, Rio de Janeiro. XXIII CBC, 2007, p.1848-1857.

IBGE, Noções Básicas de Cartografia, 1998 . Disponível em: <<ftp://geoftp.ibge.gov.br/documentos/cartografia/>> Acesso em: 14/11/08.

KROPLA B. Beginning MapServer: Open Source GIS Development, Springer-Verlag New York Inc., New York, Estados Unidos, 2005. Disponível em:
<<http://www.apress.com/book/bookDisplay.html?bID=443>> Acesso em: 18/07/07.

LARMAN, C. Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos. Porto Alegre, ed.Bookman, 2000.

MAGALHÃES JR, A.P. A situação do monitoramento das águas no Brasil – Instituições e Iniciativas. RBHR – Revisa Brasileira de Recursos Hídricos. Vol. 5, no 3, Jul/Set 2000, p. 113-115. Porto Alegre/RS : ABRH,2000.

CONAMA, 2005. Resolução nº 357 - Conselho Nacional do Meio Ambiente. Disponível em:
<<http://www.crq4.org.br/downloads/resolucao357.pdf>> Acesso em: 18/07/07

RIO DE JANEIRO, Lei 3239, de 02 de agosto de 1999. Disponível em:
<<http://www.ana.gov.br/Institucional/ASPAR/LegislacaoEstadosDF/RIODEJANEIRO.doc>>
Acesso em: 07/12/2009.

PAGE-JONES, M., Projeto Estruturado de Sistemas. São Paulo: McGraw-Hill, 1988.

SILVA, A. M. R. e VIDEIRA C. A. E ., UML, Metodologia e Ferramentas CASE , 1ª ed. Centro Atlântico,Lda, 2001 Porto , Lisboa , Portugal Disponível em:
<<http://www.centroatlantico.pt>> Acesso em: 18/07/07.

STAA, A. Programação modular: desenvolvendo programas complexos de forma organizada e segura, Rio de Janeiro, ed. Campus, 2000.

POSTGRESQL, Global Development Group. Documentação do PostgreSQL 8.0.0 / Projeto de Tradução para o Português do Brasil. University of Califórnia, 2007. Disponível em:
<<http://sourceforge.net/projects/pgdocptbr/>> Acesso em: 08/11/08

TUCCI, Carlos E. M. Modelos hidrológicos. Porto Alegre, ed. Universidade/UFRGS/Associação Brasileira de Recursos Hídricos, 1998.

APÊNDICE A – Lista de sítios utilizados neste trabalho

Sítio da ANA : <<http://www.ana.gov.br/>>

Sítio do Google Map: <<http://www.google.com./apis/maps/>>

Sítio do Mapserver : <<http://mapserver.gis.umn.edu/>>

Sítio do PostgreSQL : <<http://www.postgresql.org/>>

Sítio do QuantumGIS (QGIS): <<http://qgis.org/>>

APÊNDICE B – Relação de parâmetros de qualidade de água

Tabela 15 - Parâmetros disponíveis nos sistemas HIDRO e SAQUA

Código	Parâmetro	Unidade	Código	Parâmetro	Unidade
1	Nível de consistência	-	75	1,1 Dicloroetano	mg/L
5	Posição horizontal da coleta	-	76	1,2 Dicloroetano	mg/L
6	Posição vertical da coleta	-	77	Pentaclorofenol	mg/L
7	Choveu	-	78	Tetracloroetano	mg/L
8	Profundidade	-	79	Tricloroetano	mg/L
9	Temperatura da ar	°C	80	Tetracloroeto de carbono	mg/L
10	Temperatura da amostra	°C	81	2,4,6 triclorofenol	mg/L
11	PH	-	82	Aldrin	mg/L
12	Cor	mg Pt/Co	83	Clordano	mg/L
13	Turbidez	FTU	84	DDT	mg/L
14	Condutividade elétrica	uS/cm a 20°C	85	Dieldrin	mg/L
15	Dureza total	mg/L CaCO ₃	86	Endrin	mg/L
16	Dureza	mg/L	87	Endossulfan	mg/L
17	DQO	mg/L O ₂	88	Epóxido de heptacloro	mg/L
18	DBO	mg/L O ₂	89	Heptacloro	mg/L
19	OD	mg/L O ₂	90	Lindano	mg/L
20	Sólidos totais	mg/L	91	Metoxicloro	mg/L
21	Sólidos fixos	mg/L	92	Dodecacloro + Nonacloro	mg/L
22	Sólidos voláteis	mg/L	93	Bifenilas policloradas (PCBs)	mg/L
23	Sólidos em suspensão totais	mg/L	94	Toxafeno	mg/L
24	Sólidos em suspensão fixos	mg/L	95	Demeton	mg/L
25	Sólidos em suspensão voláteis	mg/L	96	Gution	mg/L
26	Sólidos dissolvidos totais	mg/L	97	Malation	mg/L
27	Sólidos dissolvidos fixos	mg/L a 180°C	98	Paration	mg/L
28	Sólidos dissolvidos voláteis	mg/L	99	Carbaril	mg/L
29	Sólidos sedimentáveis	mg/L	100	Ácido 2,4 Diclorofenoxiacético	mg/L
30	Detergentes	mg/L LAS	101	2,4,5 - TP	mg/L
31	Alcalinidade CO ₃	mg/L	102	2,4,5 - T	mg/L
32	Alcalinidade HCO ₃	mg/L	103	BHC	mg/L

Tabela 15 - Parâmetros disponíveis nos sistemas HIDRO e SAQUA

Código	Parâmetro	Unidade	Código	Parâmetro	Unidade
33	Alcalinidade OH	mg/L	104	Ethion	mg/L
34	Cloretos	mg/L Cl	105	Dy Syston / Disulfton	mg/L
35	Sulfatos	mg/L SO4	106	Phosdrin (mevinphos)	mg/L
36	Sulfetos	mg/L S	107	DDE PP	mg/L
37	Fluoretos	mg/L F	108	Azinfos etil	mg/L
38	Fosfato total	mg/L P	109	Diazinon	mg/L
39	Cianetos	mg/L CN	110	Estreptococos fecais	NMP/100 ml
40	Nitrogênio total	mg/L N	111	Pseudomonas	NMP/100 ml
41	Amônia não ionizável	mg/L N	112	Salmonelas	NMP/100 ml
42	Nitrogênio amoniacal	mg/L N	113	Colifagos	NMP/100 ml
43	Nitratos	mg/L N	114	Bactérias heterotróficas	UFC/ml
44	Nitritos	mg/L N	115	Protozoários	N° org/ml
45	Compostos organoclorados	mg/L	116	Fungos	UFC/ml
46	Compostos organofosforados	mg/L	117	Algas	N° UPA/ml
47	Alumínio	mg/L Al	118	Contagem de bactérias em placa	UFC/ml
48	Arsênio	mg/L As	119	Clorofila	mg/L feofitina
49	Cádmio	mg/L Cd	120	Óleos e graxas	mg/L
50	Chumbo	mg/L Pb	121	Alcalinidade total	mg/L CaCO3
51	Cobre	mg/L Cu	122	Carbono orgânico total	mg/L C
52	Cromo trivalente	mg/L Cr	123	Hidrocarbonetos	mg/L
53	Cromo hexavalente	mg/L Cr	124	Ortofosfato total	mg/L PO4
54	Ferro solúvel	mg/L Fe	125	Cromo total	mg/L Cr
55	Manganês	mg/L Mn	126	Metil - Paration	mg/L
56	Mercúrio	mg/L Hg	127	Nitrogênio orgânico	mg/L
57	Níquel	mg/L Ni	128	Oxigênio consumido em meio ácido	mg/L O2
58	Zinco	mg/L Zn	129	Sódio total	mg/L Na
59	Índice de fenóis	mg/L C6H5OH	130	Magnésio total	mg/L Mg
60	Coliformes totais	NMP/100 ml	131	Sílica dissolvida	mg/L SiO2
61	Coliformes fecais	NMP/100 ml	132	Potássio total	mg/L K
62	Cloro residual	mg/L Cl	133	Cálcio total	mg/L Ca
63	Bário	mg/L Ba	134	Ferro total	mg/L Fe
64	Berílio	mg/L Be	135	Descarga líquida	m3/s
65	Boro	mg/L B	136	Fósforo total	mg/L P

Tabela 15 - Parâmetros disponíveis nos sistemas HIDRO e SAQUA

Código	Parâmetro	Unidade	Código	Parâmetro	Unidade
66	Cobalto	mg/L Co	137	Bismuto total	mg/L Bi
67	Estanho	mg/L Sn	138	Acidez (pH = 4,5)	-
68	Lítio	mg/L Li	139	Fitoplâncton total	N° org/ml
69	Prata	mg/L Ag	140	Nitrogênio total	mg/L N
70	Selênio	mg/L Se	141	Nitrogênio albuminoide	mg/L
71	Urânio total	mg/L U	142	Transparência	-
72	Vanádio	mg/L V	143	Entero-bactérias patogênicas	N° org/ml
73	Benzeno	mg/L	144	Zooplâncton total	N° org/ml
74	Benzo-a-Pireno	mg/L	145	Amoníaco	mg/L N

Observações:

- Os campos que correspondem aos códigos 0, 2, 3 e 4 se referem ao código da estação, data que foram realizadas as medições, hora inicial das mesmas e número da medição, respectivamente. Como estes códigos foram orientações para os sistemas não constam desta tabela.

- Os campos que correspondem aos códigos 1,5,6 e 7 são códigos numéricos que representam diferentes situações. Para o código 1 (Nível de consistência) o número: 0 = Bruto; 1 = Consistido . Para código 5 (Posição horizontal da coleta) o número: 1 = Margem direita ; 2 = Centro; 3 = Margem esquerda ; 4 = Composta; 5 = Extra; 6 = Indefinida; 7 = Superfície; 8 = Integrada; 9 = 40%; 10 = 50%. Para código 6 (Posição vetical da coleta) o número: 1 = Indefinida ; 2 = Superficial; 3 = Integrada; 4 = Fundo; 5 = 20%; 6 = 40%; 7 = 50%; 8 = 60%; 9 = 70%; 10 = 80%. Para o código 7 (Choveu) o número: 0 = Não; 1 = Sim.

APÊNDICE C – Remoção periódica de arquivos temporários

Para remoção de todos arquivos temporários do SAQUA foi criado o arquivo *limpa_tmp.bat* com uma única linha de comando: “del c:\mestrado\tmp*.* /S /Q” executado, por exemplo com periodicidade diária, conforme mostra a **Figura 31**.

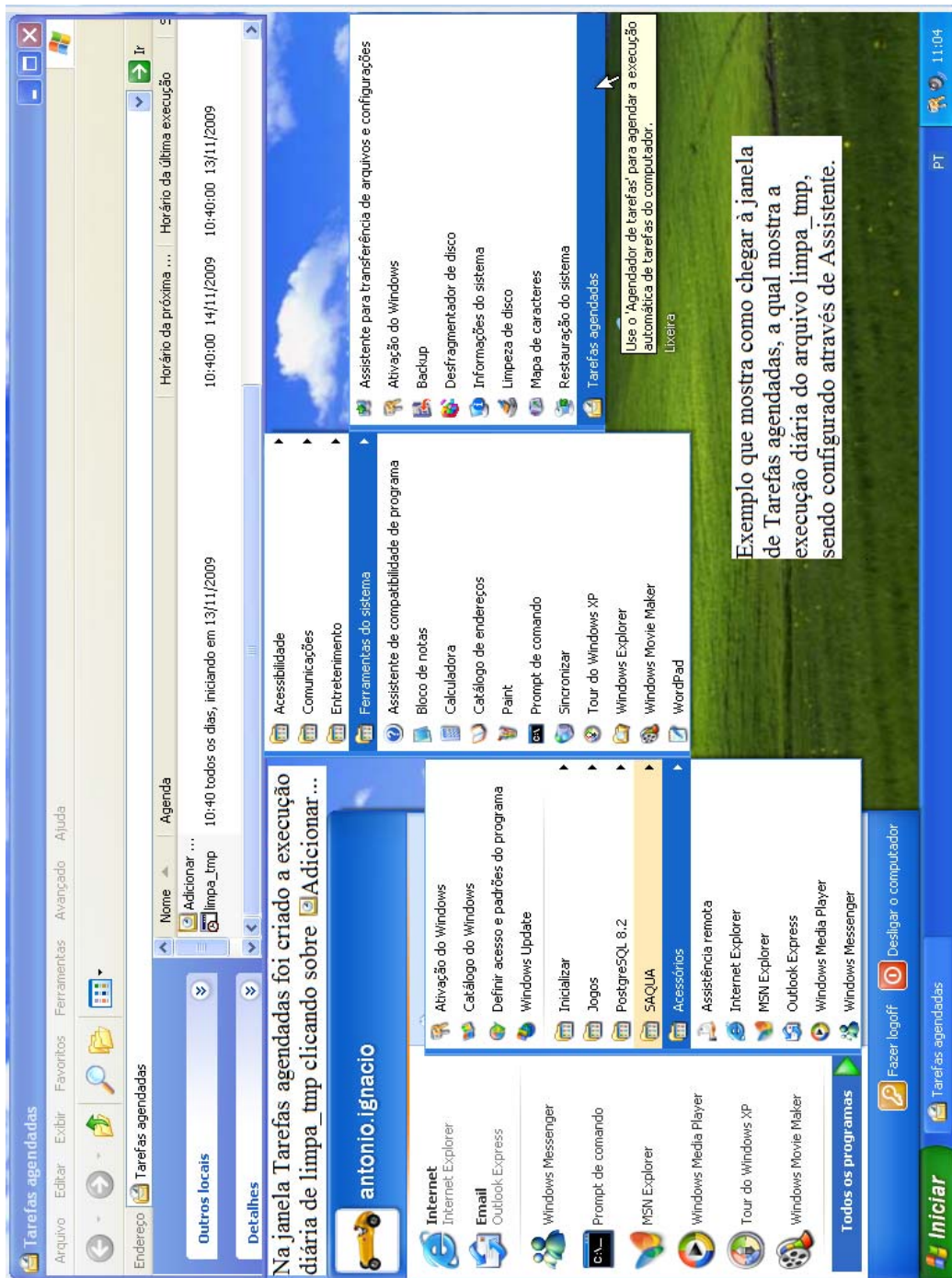


Figura 31 – Tela do windows mostrando a exclusão diária de arquivos temporários

APÊNDICE D – CÓDIGOS

Segue um sumário dos códigos implementados, neste anexo, com exceção do arquivo “limpa_tmp.bat”, na pasta *mestrado*, que possui uma única linha de comando “del c:\mestrado\tmp*.* /S /Q”, seguindo a ordem de sub-pastas abaixo da pasta “mestrado”:

Código do arquivo <i>index.html</i> na pasta <i>mestrado</i>	92
Código do arquivo <i>alerta.php</i> na sub-pasta <i>mapalerta</i>	94
Código do arquivo <i>alerta.php.map</i> na sub-pasta <i>mapalerta</i>	101
Código do arquivo <i>mapalerta.php</i> na sub-pasta <i>mapalerta</i>	109
Código do arquivo <i>mapalerta.php.map</i> na sub-pasta <i>mapalerta</i>	120
Código do arquivo <i>instala.bat</i> na sub-pasta <i>instala</i>	128
Código do arquivo <i>cria_e_preenche_banco.php</i> na sub-pasta <i>instala</i>	128
Código do arquivo <i>desinstala.bat</i> na sub-pasta <i>instala</i>	131
Código do arquivo <i>index.html</i> na sub-pasta <i>saqua</i>	132
Código do arquivo <i>mensagem2.html</i> na sub-pasta <i>saqua</i>	132
Código do arquivo <i>saqua.php</i> na sub-pasta <i>saqua</i>	132
Código do arquivo <i>carga.php</i> na sub-pasta <i>saqua</i>	139
Código do arquivo <i>google_estacoes.html</i> na sub-pasta <i>saqua</i>	144
Código do arquivo <i>google_estacoes.php</i> na sub-pasta <i>saqua</i>	146
Código do arquivo <i>google_estac_off.php</i> na sub-pasta <i>saqua</i>	148

Todos estes arquivos fazem parte da implementação do SAQUA que em conjunto com os software livre a serem instalados e configurados, os arquivos de dados das séries históricas, fornecidos pela ANA através do HIDROWEB, e os padrões adotados pela CONAMA, armazenados para os parâmetros selecionados, permitem a utilização do mesmo.

Código do arquivo *alerta.php* na sub-pasta *mapalerta*

```

<?php
/--Arquivo em PHP para interagir diretamente com funcoes
/--do mapserver usando definicoes em alertaphp.map-----
//-----
/-- Funcoes extras utilizadas:
/--EstacaoProx - retorna o numero do registro menos um
function EstacaoProx($point,$map,$radius) {
    // Seleciona a camada da consulta
    $qlayer = $map->getLayerByName('estacao');
    $qlayer->set("tolerance",$radius);
    // $radius e um valor ajustado no navegador. Em
    // queryByPoint a unidade de tolerancia e ignorada
    // usando as unidades nativa do mapa. A longitude inicial
    // neste caso varia em 5 graus (-46 ate -41), no globo
    // terrestre (360 graus), e utiliza-se uma constante (c)
    // para ajustar as dimensoes no mapa. O $radius sera
    // trocado quando a unidade de tolerancia, o mapa ou
    // as escalas de unidade forem trocados.
    $c = 1.927;
    @$qlayer->queryByPoint($point, MS_SINGLE,
        $radius*(5/(360*$c)));
    $numResults = $qlayer->getNumResults();
    // No resultado capturado, o numero do identificador
    // armazenado, ou o registro do arquivo menos um.
    if ($numResults != 0) {
        $query_result = $qlayer->getResult(0);
        $StoreList[0] = $query_result->shapeindex;
    } else {
        $StoreList = "";
    }
    // sem resultados
    }
    return $StoreList;
}
/--Voltameta - retorna o metadado do campo numerico
function Voltameta($arquivo,$campo) {
    OGRRegisterAll();
    $hSFDriver=NULL;
    $strfilename = "\mestrado\mapalerta\\"$arquivo" ;
    $meta = "";
    $hDatasource1 = OGROpen($strfilename,
        0, $hSFDriver);
    $hlayer1 = OGR_DS_GetLayer($hDatasource1,0);
    $hLayerDefn = OGR_L_GetLayerDefn( $hlayer1 );
    $numFields = OGR_FD_GetFieldCount(
        $hLayerDefn );
    if ($campo <= $numFields) {
        $hFeature1 = OGR_L_GetNextFeature( $hlayer1);
    }
}

```



```

    $hFeatureDefn = OGR_F_GetDefnRef($hFeature1);
    $hFieldDefn = OGR_FD_GetFieldDefn(
        $hFeatureDefn, $campo);
    $meta = OGR_FLD_GetNameRef($hFieldDefn);
    OGR_F_Destroy( $hFeature1 );
}
return $meta;
}
}
//--Voltareg - retorna o registro do campo numerico
function Voltareg($arquivo,$campo,$reg) {
    OGRRegisterAll();
    $hSFDriver=NULL;
    $strfilename = "\mestrado\mapalerta\\$arquivo" ;
    $registro = "";
    $hDatasource1 = OGROpen($strfilename,
        0, $hSFDriver);
    $hlayer1 = OGR_DS_GetLayer($hDatasource1,0);
    $hLayerDefn = OGR_L_GetLayerDefn( $hlayer1 );
    $numFields = OGR_FD_GetFieldCount(
        $hLayerDefn );
    if ($campo <= $numFields) {
        $hFeature1 = OGR_L_GetNextFeature( $hlayer1);
        for($count= (integer)$reg[0]; $count>0 ; $count--){
            if (($hFeature1 = OGR_L_GetNextFeature(
                $hlayer1 )) == NULL ) $count=0; }
        $hFeatureDefn = OGR_F_GetDefnRef($hFeature1);
        $hFieldDefn = OGR_FD_GetFieldDefn(
            $hFeatureDefn, $campo);
        if( OGR_F_IsFieldSet( $hFeature1, $campo ) )
            $registro = OGR_F_GetFieldAsString(
                $hFeature1, $campo );
        $meta = OGR_FLD_GetNameRef($hFieldDefn);
        OGR_F_Destroy( $hFeature1 );
    }
    return $registro;
}
}
// img2map - Converte coord. da imagem para o mapa
function img2map($width,$height,$point,$ext) {
    $minx = $ext->minx;
    $miny = $ext->miny;
    $maxx = $ext->maxx;
    $maxy = $ext->maxy;
    if ($point->x && $point->y){
        $x = $point->x;
        $y = $point->y;
        $dpp_x = ($maxx-$minx)/$width;
        $dpp_y = ($maxy-$miny)/$height;
        $x = $minx + $dpp_x*$x;
        $y = $maxy - $dpp_y*$y;
    }
}

```

```

    }
    $pt[0] = $x;
    $pt[1] = $y;
    return $pt;
}
//-----
// ----- Inicio de codigo: - Parte em PHP
// Valores padrao
$script_name = "alerta.php"; //este arquivo
// locais padrao (arbitrados segundo instalacao)
$map_path = "/mestrado/mapalerta/";
$img_path = "/mestrado/tmp/";
$tmp_path = "../tmp/";
$map_file = "alertaphp.map";
// padroes iniciais mostrados no Navegador
$zoomsize=2;
$pan="CHECKED";
$zoomout="";
$zoomin="";
// Niveis mostrados por padrao
$bacia="CHECKED";
$sub_bac="CHECKED";
$rrios="CHECKED";
// ponto padrao do clique
$clickx = 300;
$clicky = 150;
$clkpoint = ms_newPointObj();
$sold_extent = ms_newRectObj();
// posicao do mapa apresentado
$extent = array(-46.5, -24.5, -41, -20.4);
$max_extent = ms_newRectObj();
$max_extent->setextent(-70, -35, -15, 6);
$radius = 15; // ajuste inicial da tolerancia em km
// Criando o mapa inicial
$map = ms_newMapObj($map_path.$map_file);
// Passando as variaveis iniciais
if (( $_POST['img_x'] and $_POST['img_y'] )
    or $_POST['refresh'] ) {
    if ( $_POST['refresh'] ) {
        $clickx = 300;
        $clicky = 150;
    } else {
        $clickx = $_POST['img_x'];
        $clicky = $_POST['img_y'];
    }
}
// Atualizando o clique do mouse (para acoes)
$clkpoint->setXY($clickx,$clicky);
// Pegando o valor de tolerancia
$radius = abs( $_POST['radius'] );

```

```

// Atualizando se houve trocas nose niveis selecionados
if ( $_POST['layer'] ) {
    $layers = join(" ", $_POST['layer']);
} else {
    $layers = "";
}
$this_layer = 0;
if ( $_POST['layer'] ) {
    $layers = join(" ", $_POST['layer']);
} else {
    $layers = "";
}
$this_layer = 0;
if (preg_match("/sub_bac/", $layers)){
    $sub_bac = "CHECKED";
    $this_layer = $map->getLayerByName('sub_bac');
    $this_layer->set('status', MS_ON);
} else {
    $sub_bac = "";
    $this_layer = $map->getLayerByName('sub_bac');
    $this_layer->set('status', MS_OFF);
}
if (preg_match("/bacia/", $layers)){
    $bacia = "CHECKED";
    $this_layer = $map->getLayerByName('bacia1');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia2');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia3');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia4');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia5');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia6');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia7');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia8');
    $this_layer->set('status', MS_ON);
} else {
    $bacia = "";
    $this_layer = $map->getLayerByName('bacia1');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia2');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia3');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia4');
}

```

```

    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia5');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia6');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia7');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia8');
    $this_layer->set('status', MS_OFF);
}
if (preg_match("/rios/", $layers)){
    $rios = "CHECKED";
    $this_layer = $map->getLayerByName('rios');
    $this_layer->set('status', MS_ON);
} else {
    $rios = "";
    $this_layer = $map->getLayerByName('rios');
    $this_layer->set('status', MS_OFF);
}
// Pega extensoes do mapa solicitado
if ( $_POST['extent'] ) {
    $extent = split(" ", $_POST['extent']);
}
// Atualiza o mapa para a extensao do formulario
$map->setExtent($extent[0],$extent[1],
                $extent[2],$extent[3]);
// Salva esta extensao como um objeto de acoes
$old_extent->setextent($extent[0],$extent[1],
                    $extent[2],$extent[3]);
// Calcula o fator de zoom para o metodo zoomPoint
// Considerando a variavel zoomfactor = +/- N
// Se N > 0 aproxima - N < 0 afasta - N = 0 centraliza.
// Altera proporcionalmente a tolerancia ( radius
$zoom_factor = $_POST['zoom']*$_POST['zsize'];
if ($zoom_factor == 0) {
    $zoom_factor = 1;
    $pan = "CHECKED";
    $zoomout = "";
    $zoomin = "";
} elseif ($zoom_factor < 0) {
    $pan = "";
    $zoomout = "CHECKED";
    $zoomin = "";
    if (($radius*(-$zoom_factor)) < 150)
        {$radius = $radius*(-$zoom_factor);}
    else $radius=150 ; // Para 50 graus (max_extent)
} else {
    $pan = "";
    $zoomout = "";
}

```

```

$zoomin = "CHECKED";
$radius = $radius/$zoom_factor ;
}
$zoomsize = abs( $_POST['zsize'] );
// Aproximando (ou afastando) para o ponto clicado
$map->zoomPoint($zoom_factor,$clkpoint,$map->width,
               $map->height,$old_extent,$max_extent);
}
// Atualiza nomes das imagens dos mapas e legenda
$map_id = sprintf("%0.6d",rand(0,999999));
$image_name = "mapalerta".$map_id.".png";
$image_url=$tmp_path.$image_name;
$leg_name = "legalerta".$map_id.".png";
$leg_url=$tmp_path.$leg_name;
// Desenha e salva a imagem do mapa
$image=$map->draw();
$image->saveImage($img_path.$image_name);
// Desenha e salva a imagem da legenda
$leg = $map->drawLegend();
$leg->saveImage($img_path.$leg_name);
// Salva as extensoes depois de centralizar e aproximar
// uma variavel do formulario pelo valor delimitado
$new_extent = sprintf("%3.6f",$map->extent->minx)." "
               .sprintf("%3.6f",$map->extent->miny)." "
               .sprintf("%3.6f",$map->extent->maxx)." "
               .sprintf("%3.6f",$map->extent->maxy);
// Formata a escala da imagem para exibir
$scale = sprintf("%10d",$map->scale);
// Converte coordenadas selecionadas no mapa
list($x,$y) = img2map($map->width,
                    $map->height,$clkpoint,$old_extent);
$x_str = sprintf("%3.6f",$x);
$y_str = sprintf("%3.6f",$y);
$qpoint = ms_newPointObj(); // cria e atualiza parametro
$qpoint->setXY($x,$y);
$nearby = EstacaoProx($qpoint,$map,$radius);
$arqgest = "agua_qualidade5.dbf";
?>
<HTML>
<!-- Codigo em HTML com chamadas em PHP ----- ->
<HEAD><TITLE>Análise Espacial Temporal de
Parâmetros de Qualidade das Águas na
Sub-bacia 58</TITLE></HEAD>
<BODY font face="Arial,Helvetica, sans-serif" style="font-size: 10px">
<table width="798" height="12" border="0" align="center"><tr><td width="796"
colspan="2"><tr><font color="#006699" face="Arial, Helvetica, sans-serif"
style="font-size: 20px; font-weight: bold;">
Análise Espacial Temporal de Parâmetros de Qualidade
das Águas na Sub-bacia 58.<br><HR>

```

```

</tr></font></td><td width="600"> <form method=post
  action="<?php echo $script_name;?>">
  <input name="img" type="image" src="<?php echo $image_url;?>"
  width="600" height="300" border=0></div>
  <td width="196"><b>A&ccedil;&otilde;es:</b> Centrar <INPUT type=radio
  name="zoom" value=0 <?php echo $span;?>>
  &nbsp;&nbsp;&nbsp;<INPUT type="submit" name="ajuda" value="Ajuda"><br>
  Vir <INPUT type=radio name="zoom" value=1 <?php echo $zoomin;?>>
  | Ir <INPUT type=radio name="zoom" value=-1 <?php echo $zoomout;?>>
  | Fator: <INPUT type=text name="zsize" value="<?php echo $zoomsize;?>"
  size=3><br> <B>Camadas: |</B>
  <input type="checkbox" name="layer[]" value="sub_bac"
  <?php echo $sub_bac;?>>Sub-Bacias<br> <input type="checkbox"
  name="layer[]" value="bacia" <?php echo $bacia;?> >Bacias |
  <input type="checkbox" name="layer[]" value="rios" <?php echo $rios;
  ?> >Rios&nbsp;&nbsp;&nbsp;<input type="submit" name="refresh" value="Aplica">
  <B><a href=" ../mapalerta/alerta.php">Iniciar</a></b><HR> </div>
  <B>Legenda:</B> <BR> </p>
  <INPUT type="hidden" name="extent" value="<?php echo $new_extent;?>">
  <INPUT type="hidden" name="radius" value="<?php echo $radius;?>">
</td></table>
<table border="1" width="796" heigth="8" border="0" align="center">
<?php
if ( $_POST['ajuda'] ) {
  echo "<td bgColor=#9999FF width='85%' colspan='2'>" ;
  $ajuda[1]="Sobre a manipula&ccedil;&atilde;o do mapa -
  <b>A&ccedil;&otilde;es</b> permitidas:";
  $ajuda[2]="Centrar -> Utiliza o ponto que clicar no mapa para o centro,
  ou pode-se aplicar o centro atual.";
  $ajuda[3]="Ir -> Com o Fator escolhido aproxima-se centrando | Vir ->
  Com o Fator escolhido afasta-se centrando";
  $ajuda[4]="Fator -> Valor utilizado nas acoes de ir e vir linearmente
  | <b>Camadas</b>: -> Oculta ou exhibe camadas opcionais";
  $ajuda[5]="<b>Bot&otilde;es</b>: Aplica -> Corresponde a clicar
  exatamente no centro do mapa | Ajuda -> Exibe este menu";
  for($i=1 ; $i<6 ; $i++) { echo $ajuda[$i]."<br>\n"; }
  echo "<td bgColor=#9999FF> O link <U>Iniciar</U><br>reinicia a
  p&aacute;gina<p> Autor:<br>Antonio I.Gl&oacute;ria.</tr><tr>";
  } else {
  if ($nearby) $situac=Voltareg($arqest,12,$nearby);
  if ( $nearby and (strlen($situac) > 0) ) { // Se esta proximo e com alerta
  if( $situac == 1) { $colore="#ffffa0";} else { if( $situac == 2) {
  $colore="#ffa0a0";} else { $colore="#a0ffa0";}}
  echo "\n<tr><td bgColor=#c0c0c0 width='50%' colspan='2'>
  <B>Arquivo</B>: ".$arqest." - <B>Registro:</B> ".$( $nearby[0]+1);
  // preenche campos impares com metadados e pares com dados da estacao
  for($i=0 ; $i<13 ; $i++) {
  $campo[(1+2*$i)]=Voltameta($arqest,$i);
  $campo[(2+2*$i)]=Voltareg($arqest,$i,$nearby);

```

```

}
echo "&nbsp;<td bgColor=$colore width='50%' colspan='2'><B>".
    $campo[13].":</B> ".$campo[14]."</tr>\n<tr>";
for($i=0 ; $i<6 ; $i++) {
echo "<td bgColor=$colore width='20%'><B>".$campo[(1+2*$i)].
    "</B>: ".$campo[(2+2*$i)];
echo "<td bgColor=$colore width='30%'><B>".$campo[(15+2*$i)].
    "</B>: ".$campo[(16+2*$i)]."";
if ( ($i%2) != 0 ){ echo "</tr>\n<tr>";}
}
}
} ?>
<td width='100%' colspan='4'>
<B>Coordenada Centro:</B> (X,Y) = (
<?php echo $x_str.", ".$y_str." ) - Servidor em";
$ab="(".PHP_OS;
if ( strpos($ab,"WIN") !=0 ) {echo " Windows $ab "}; else {
echo " Linux $ab" };
$ver=(string) getenv(SERVER_SOFTWARE); echo "&nbsp;c/ $ver";?>
</tr></table>
</FORM></BODY></HTML>

```

Código do arquivo *alertaphp.map* na sub-pasta *mapalerta*

```

# Mapfile com camada para bacias hidricas disponiveis no site da ANA.
MAP                # Início do objeto MAP
# Define tamanho do mapa em tela
SIZE 600 300
# Define posição no mapa apresentado
EXTENT -46.5 -24.5 -41 -20.4 # centralizando para sub_bacia 58
IMAGECOLOR 180 180 250 # escolhendo cor do que não for feição
IMAGETYPE PNG24
SYMBOL # standard circular brush
NAME 'circle'
TYPE ELLIPSE
POINTS 1 1 END
FILLED TRUE
END
# Determina saída para web
WEB
METADATA
"wms_title" "Alertas hidricos c/Mapserver"
"wms_srs" "EPSG:4326"
END
END
# Coloca Legenda no mapa
LEGEND
STATUS ON

```

```
POSITION LR
TRANSPARENT TRUE
END
UNITS DD
# Define a projeção do mapa
PROJECTION
  "proj=latlong"
  "ellps=WGS84"
END
# Coloca Escala no mapa
SCALEBAR
STATUS EMBED
POSITION LR
UNITS KILOMETERS
INTERVALS 3
TRANSPARENT TRUE
OUTLINECOLOR 0 0 0
END
# As camadas das 8 bacias
LAYER
NAME bacia1
TYPE POLYGON
STATUS ON
DATA bacia1
LABELITEM 'BACIA'
CLASS
NAME 'Outras Bacias (DNAEE)'
STYLE
COLOR 255 255 150
OUTLINECOLOR 0 0 0
SYMBOL 0
END
LABEL
COLOR 255 255 255
OUTLINECOLOR 0 0 0
POSITION CC
END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
LAYER
NAME bacia2
TYPE POLYGON
STATUS ON
DATA bacia2
LABELITEM 'BACIA'
```



```
CLASS
  STYLE
    COLOR 255 255 150
    OUTLINECOLOR 0 0 0
    SYMBOL 0
  END
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
LAYER
  NAME bacia3
  TYPE POLYGON
  STATUS ON
  DATA bacia3
  LABELITEM 'BACIA'
  CLASS
    STYLE
      COLOR 255 255 150
      OUTLINECOLOR 0 0 0
      SYMBOL 0
    END
    LABEL
      COLOR 255 255 255
      OUTLINECOLOR 0 0 0
      POSITION CC
    END
  END
  PROJECTION
    "proj=latlong"
    "proj=WGS84"
  END
END
LAYER
  NAME bacia4
  TYPE POLYGON
  STATUS ON
  DATA bacia4
  LABELITEM 'BACIA'
  CLASS
    STYLE
      COLOR 255 255 150
```

```
    OUTLINECOLOR 0 0 0
    SYMBOL 0
  END
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
LAYER
  NAME bacia5
  TYPE POLYGON
  STATUS ON
  DATA bacia5
  LABELITEM 'BACIA'
  CLASS
    NAME 'Bacia 5 (DNAEE)'
    STYLE
      COLOR 255 255 255
      OUTLINECOLOR 0 0 0
      SYMBOL 0
    END
    LABEL
      COLOR 255 255 255
      OUTLINECOLOR 0 0 0
      POSITION CC
    END
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
LAYER
  NAME bacia6
  TYPE POLYGON
  STATUS ON
  DATA bacia6
  LABELITEM 'BACIA'
  CLASS
    STYLE
      COLOR 255 255 150
      OUTLINECOLOR 0 0 0
      SYMBOL 0
```

```
END
LABEL
  COLOR 255 255 255
  OUTLINECOLOR 0 0 0
  POSITION CC
END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
LAYER
  NAME bacia7
  TYPE POLYGON
  STATUS ON
  DATA bacia7
  LABELITEM 'BACIA'
  CLASS
  STYLE
    COLOR 255 255 150
    OUTLINECOLOR 0 0 0
    SYMBOL 0
  END
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
LAYER
  NAME bacia8
  TYPE POLYGON
  STATUS ON
  DATA bacia8
  LABELITEM 'BACIA'
  CLASS
  STYLE
    COLOR 255 255 150
    OUTLINECOLOR 0 0 0
    SYMBOL 0
  END
  LABEL
    COLOR 255 255 255
```

```

    OUTLINECOLOR 0 0 0
    POSITION CC
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
# A camada da bacia do Rio Par.do Sul
LAYER
  NAME fixo
  TYPE POLYGON
  STATUS DEFAULT # Sempre aparece
  DATA sub_bacias5
  LABELITEM 'SUB_BACIA'
  CLASSITEM 'SUB_BACIA'
  CLASS
    EXPRESSION '58'
    NAME 'Bacia do rio Pb. do Sul'
    STYLE
      OUTLINECOLOR 100 100 100
      COLOR 160 255 160
    END
    LABEL
      COLOR 255 255 255
      OUTLINECOLOR 0 0 0
      POSITION CC
    END
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
# Para as demais sub-bacias
LAYER
  NAME sub_bac
  TYPE POLYGON
  STATUS ON
  DATA sub_bacias5
  LABELITEM 'SUB_BACIA'
  CLASSITEM 'SUB_BACIA'
  CLASS
    EXPRESSION ('[SUB_BACIA]' ne '58')
    NAME 'Sub-bacias 5X (DNAEE)'
    STYLE
      OUTLINECOLOR 100 100 100
      COLOR 200 200 200
    END
  END

```

```
END # Fim do objeto STYLE
LABEL
  COLOR 255 255 255
  OUTLINECOLOR 0 0 0
  POSITION CC
END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
# A camada de rios da bacia 5 para a sub-bacia 58
LAYER
  NAME rios
  TYPE LINE
  STATUS ON
  DATA hidrografia5
  LABELITEM 'NOME'
  CLASSITEM 'SUB_BACIA'
  CLASS
  EXPRESSION /5/
  NAME 'Rios'
  STYLE
  COLOR 0 0 255
  OUTLINECOLOR 0 0 0
  SYMBOL 0
  END
  LABEL
  COLOR 255 255 255
  OUTLINECOLOR 0 0 255
  POSITION CC
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
# A Camada de alertas para estacoes de qualidade da agua
LAYER
  NAME estacao
  TYPE POINT
  STATUS DEFAULT
  LABELCACHE on
  TOLERANCEUNITS miles
  DATA agua_qualidade5
  LABELITEM 'ESTACAO'
  CLASSITEM 'ALERTA'
```

```
TEMPLATE err_alerta.php
CLASS
  EXPRESSION '0'
  NAME 'Estações SEM Alertas'
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
  STYLE
    COLOR 0 255 0
    OUTLINECOLOR 0 0 0
    SYMBOL 'circle'
    SIZE 15
  END
END
CLASS
  EXPRESSION '1'
  NAME 'Estações em Observação'
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
  STYLE
    COLOR 255 255 0
    OUTLINECOLOR 0 0 0
    SYMBOL 'circle'
    SIZE 15
  END
END
CLASS
  EXPRESSION '2'
  NAME 'Estações COM Alertas'
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
  STYLE
    COLOR 255 0 0
    OUTLINECOLOR 0 0 0
    SYMBOL 'circle'
    SIZE 15
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
```

```

END
END
#-----
END # Fim do objeto MAP e do mapfile

```

Código do arquivo *mapalerta.php* na sub-pasta *mapalerta*

```

<?php
/--Arquivo em PHP para interagir diretamente com funcoes
/--do mapserver usando definicoes em mapalertaphp.map-----
//-----
/-- Funcoes extras utilizadas:
/-- Alteralerta - Atualiza arquivos de alertas no temporario
function Alteralerta($conex1,$map_id,$cod_par,$cod_cla,$mes,$ano,$estat,$mult) {
  if ($estat=="A") {
    $result2 = pg_exec ($conex1, "select cod_estac,situa_max from alerta where
      cod_param=$cod_par and cod_class=$cod_cla and mes=$mes and ano=$ano;");
    $tot_tuplas2 = pg_numrows ($result2);
  } elseif ($estat=="I") {
    $result2 = pg_exec ($conex1, "select cod_estac,situa_min from alerta where
      cod_param=$cod_par and cod_class=$cod_cla and mes=$mes and ano=$ano;");
    $tot_tuplas2 = pg_numrows ($result2);
  } else { // caso em que $estat=="M"
    $result2 = pg_exec ($conex1, "select cod_estac,situa_med from alerta where
      cod_param=$cod_par and cod_class=$cod_cla and mes=$mes and ano=$ano;");
    $tot_tuplas2 = pg_numrows ($result2);
  }
  for ($stupla = 0; $stupla < $tot_tuplas2; $stupla ++){
    $cod_clas[$stupla]=pg_result ($result2, $stupla, 0);
    $alerta[$stupla]=pg_result ($result2, $stupla, 1);
  }
  $arqdb="./tmp/agua_q".$map_id.".dbf";
  $arqsx="./tmp/agua_q".$map_id.".shx";
  $arqsp="./tmp/agua_q".$map_id.".shp";
  $arqmp="./tmp/mapalerta".$map_id.".map";
  $db = dbase_open('./mapalerta/agua_qualidade5.dbf', 2);
  if ($db) {
    $define= array(
      array("CODIGO", "N", 9 , 0),
      array("ROTEIRO", "N", 7 , 0),
      array("LAT", "N", 7 , 2),
      array("LATITUDE", "N", 9 , 2),
      array("LONG", "N", 9 , 2),
      array("LONGITUDE", "N", 9 , 2),
      array("ESTACAO", "C", 28 , 0),
      array("CURSO_DAGU", "C", 20 , 0),
      array("UF", "C", 7 , 0),
      array("OPERADORA", "C", 10 , 0),
      array("TIPO_COLET", "C", 10 , 0),

```

```

array("OBSERVACOE", "C", 15 , 0),
array("ALERTA", "C", 1 , 0));
$dest=$arqdb;
if (dbase_create( $dest, $define)) {
    // Carrega os dados do arquivo original em uma matriz
    $record_numbers = dbase_numrecords($db);
    for ($recno = 1; $recno <= $record_numbers; $recno ++){
        $rec = dbase_get_record($db, $recno);
        $matriz[$recno]=$rec;
        $nf = dbase_numfields($db);
    }
    $record_numbers = dbase_numrecords($db);
    dbase_close($db);
}
}
if ($dest) {
    $db2 = dbase_open($dest, 2);
    for ($i = 1; $i <= $record_numbers; $i++) {
        if ($matriz[$i][12] != " "){
            // O codigo da estacao $i esta em $matriz[$i][0]
            // O estado da estacao $i esta em $matriz[$i][12]
            // Todos valores iniciados sem dados, ou alerta "3"
            $matriz[$i][12]="3";
            for ($stupla = 0; $stupla < $tot_tuplas2; $stupla ++){
                if ($matriz[$i][0]==$cod_clas[$stupla]) {
                    $matriz[$i][12]=$alerta[$stupla];}
            }
        }
    }
    dbase_add_record($db2,
        array(
            $matriz[$i][0] ,
            $matriz[$i][1] ,
            $matriz[$i][2] ,
            $matriz[$i][3] ,
            $matriz[$i][4] ,
            $matriz[$i][5] ,
            $matriz[$i][6] ,
            $matriz[$i][7] ,
            $matriz[$i][8] ,
            $matriz[$i][9] ,
            $matriz[$i][10] ,
            $matriz[$i][11] ,
            $matriz[$i][12]
        )
    );
} // nesta linha fim do for

dbase_close($db2);
}

```



```

    @$qlayer->queryByPoint($point, MS_SINGLE,
        $radius*(5.5/(360*$c)));
    $numResults = $qlayer->getNumResults();
    // No resultado capturado, o numero do identificador
    // armazenado, ou o registro do arquivo menos um.
    if ($numResults != 0) {
        $query_result = $qlayer->getResult(0);
        $StoreList[0] = $query_result->shapeindex;
    } else {
        $StoreList = "";
    }
    // sem resultados
    }
    return $StoreList;
}
// img2map - Converte coord. da imagem para o mapa
function img2map($width,$height,$point,$ext) {
    $minx = $ext->minx;
    $miny = $ext->miny;
    $maxx = $ext->maxx;
    $maxy = $ext->maxy;
    if ($point->x && $point->y){
        $x = $point->x;
        $y = $point->y;
        $dpp_x = ($maxx-$minx)/$width;
        $dpp_y = ($maxy-$miny)/$height;
        $x = $minx + $dpp_x*$x;
        $y = $maxy - $dpp_y*$y;
    }
    $pt[0] = $x;
    $pt[1] = $y;
    return $pt;
}
//-----
// ----- Inicio de codigo: - Parte em PHP
// Valores padrao
// Conectando ao banco de dados e buscando valores cadastrados de parametros
$conex1 = pg_connect ("host = localhost
port = 5432
dbname = saqua
user = saqua
password = mestrado")
or die ("Falha na conexão!".pg_last_error());
// Preenche vetor de parametros cadastrados (cod_par e param) e matrizes
// com classes e seus nomes, condicoes, limites e obs (aviso)
$result1 = pg_exec ($conex1, "select cod_param, parametro,unidade from param");
$tot_tuplas1 = pg_numrows ($result1);
for ($tupla = 0; $tupla < $tot_tuplas1; $tupla ++){
    $cod_param[$tupla]=pg_result ($result1, $tupla, 0);
    $nom_param[$tupla]=pg_result ($result1, $tupla, 1).'.'.

```

```

        pg_result ($result1, $tupla, 2).');
$result2 = pg_exec ($conex1, "select cod_class, cond, limite, aviso
        from criterio where cod_param=$cod_param[$tupla] ");
$tot_tuplas2 = pg_numrows ($result2);
for ($tupla2 = 0; $tupla2 < $tot_tuplas2; $tupla2 ++){
    $cod_par_clas[$tupla][$tupla2]=pg_result ($result2, $tupla2, 0);
    $cod_par_cond[$tupla][$tupla2]=pg_result ($result2, $tupla2, 1);
    $cod_par_lim[$tupla][$tupla2]=pg_result ($result2, $tupla2, 2);
    $cod_par_avi[$tupla][$tupla2]=pg_result ($result2, $tupla2, 3);
    $cod_cl=$cod_par_clas[$tupla][$tupla2];
    $result3 = pg_exec ($conex1,
        "select classe from classes where cod_class=$cod_cl ");
    $cod_par_nomclas[$tupla][$tupla2]=pg_result ($result3, 0, 0);
}
}
// No caso de inexistencia de parametro sera utilizado o primeiro cadastrado
// Idem para a classe, no caso o primeiro cadastrado no banco foi OD-classe 2
if ( $_POST['item_par'] ) { $item_par=$_POST['item_par'];
} else { $item_par=0; }
if ( $_POST['item_class'] ) { $item_class=$_POST['item_class'];
} else { $item_class=0; }
// consultar na tabela -- Preencher
$cod_par=$cod_param[$item_par];
$parametro= $nom_param[$item_par];
$cod_class=$cod_par_clas[$item_par][$item_class];
$classe= $cod_par_nomclas[$item_par][$item_class];
$padrao= $cod_par_nomclas[$item_par][$item_class];
$limobs= $cod_par_avi[$item_par][$item_class];
$limite=$cod_par_lim[$item_par][$item_class];
$cond=$cod_par_cond[$item_par][$item_class];
//-----
$script_name = "mapalerta.php"; //este arquivo
// locais padrao (arbitrados segundo instalacao)
$map_path = "/mestrado/mapalerta/";
$img_path = "/mestrado/tmp/";
$tmp_path="/mestrado/tmp/";
// Valor de referencia para temporarios (map_id)
$map_id = sprintf("%0.6d",rand(0,999999));
$map_file = "../tmp/mapalerta".$map_id.".map";
$zoomsize=2;
$pan="CHECKED";
$zoomout="";
$zoomin="";
// Niveis mostrados por padrao
$bacia="CHECKED";
$sub_bac="CHECKED";
$rrios="CHECKED";
// ponto padrao do clique
$clickx = 300;

```

```

$clicky = 150;
$clkpoint = ms_newPointObj();
$sold_extent = ms_newRectObj();
// posicao do mapa apresentado
$extent = array(-46.5, -24.5, -41, -20.4);
$max_extent = ms_newRectObj();
//$max_extent->setextent(-70, -40, -20, 10);
$max_extent->setextent(-70, -35, -15, 6);
// Pegando o valor de tolerancia
if( $_POST['radius']) {$radius = abs( $_POST['radius'] ); }
else { $radius = 15; }; // ajuste inicial da tolerancia em km
if ( $_POST['mes'] and $_POST['ano'] ) {$mes=$_POST['mes'];$ano=$_POST['ano'];}
else { $mes="12"; $ano="2002"; };
if ( $_POST['estat'] ) {
    $estat=$_POST['estat'];
} else {
    $estat="I"; // I=minima M=media A=maxima
}
// Calcula o fator de zoom para o metodo zoomPoint
// Considerando a variavel zoomfactor = +/- N
// Se N > 0 aproxima - N < 0 afasta - N = 0 centraliza.
// Altera proporcionalmente a tolerancia ( radius
$zoom_factor = $_POST['zoom']*$_POST['zsize'];
if ($zoom_factor == 0) {
    $zoom_factor = 1;
    $span = "CHECKED";
    $zoomout = "";
    $zoomin = "";
} elseif ($zoom_factor < 0) {
    $span = "";
    $zoomout = "CHECKED";
    $zoomin = "";
    if (($radius*(-$zoom_factor)) < 150)
        {$radius = $radius*(-$zoom_factor);}
    else {$radius=150 ;} // Para 50 graus (max_extent)
} else {
    $span = "";
    $zoomout = "";
    $zoomin = "CHECKED";
    $radius = $radius/$zoom_factor ; }
$muilt=1; // supondo que nao ha necessidade de reduzir tamanho de circulo
if ($radius>15) $muilt=sqrt(15/$radius); // reduzir area do circulo com mult.
// Criar copias e preenche arquivos em pasta temporaria para alertas no mapa
Alteralerta($conex1,$map_id,$cod_par,$cod_class,$mes,$ano,$estat,$muilt);
// Cria ponteiro do mapa
$map = ms_newMapObj($map_path.$map_file);
// Passando as variaveis iniciais
if (( $_POST['img_x'] and $_POST['img_y'] )
    or $_POST['refresh'] or $_POST['altera'] ) {

```

```

if ( $_POST['refresh'] or $_POST['altera']) {
    $clickx = 300;
    $clicky = 150;
} else {
    $clickx = $_POST['img_x'];
    $clicky = $_POST['img_y'];
}
// Atualizando o clique do mouse (para acoes)
$clkpoint->setXY($clickx,$clicky);
// Atualizando se houve trocas nose niveis selecionados
if ( $_POST['layer'] ) {
    $layers = join(" ",$_POST['layer']);
} else {
    $layers = "";
}
$this_layer = 0;
if ( $_POST['layer'] ) {
    $layers = join(" ",$_POST['layer']);
} else {
    $layers = "";
}
$this_layer = 0;
if (preg_match("/sub_bac/", $layers)){
    $sub_bac = "CHECKED";
    $this_layer = $map->getLayerByName('sub_bac');
    $this_layer->set('status', MS_ON);
} else {
    $sub_bac = "";
    $this_layer = $map->getLayerByName('sub_bac');
    $this_layer->set('status', MS_OFF);
}
if (preg_match("/bacia/", $layers)){
    $bacia = "CHECKED";
    $this_layer = $map->getLayerByName('bacia1');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia2');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia3');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia4');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia5');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia6');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia7');
    $this_layer->set('status', MS_ON);
    $this_layer = $map->getLayerByName('bacia8');
    $this_layer->set('status', MS_ON);
}

```

```

} else {
    $bacia = "";
    $this_layer = $map->getLayerByName('bacia1');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia2');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia3');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia4');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia5');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia6');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia7');
    $this_layer->set('status', MS_OFF);
    $this_layer = $map->getLayerByName('bacia8');
    $this_layer->set('status', MS_OFF);
}
if (preg_match("/rios/", $layers)){
    $rios = "CHECKED";
    $this_layer = $map->getLayerByName('rios');
    $this_layer->set('status', MS_ON);
} else {
    $rios = "";
    $this_layer = $map->getLayerByName('rios');
    $this_layer->set('status', MS_OFF);
}
// Pega extensoes do mapa solicitado
if ( $_POST['extent'] ) {
    $extent = split(" ", $_POST['extent']);
}
// Atualiza o mapa para a extensao do formulario
$map->setExtent($extent[0],$extent[1],$extent[2],$extent[3]);
// Salva esta extensao como um objeto de acoes
$old_extent->setextent($extent[0],$extent[1],$extent[2],$extent[3]);
$zoomsize = abs( $_POST['zsize'] );
// Aproximando (ou afastando) para o ponto clicado
$map->zoomPoint($zoom_factor,$clickpoint,$map->width,
    $map->height,$old_extent,$max_extent);
}
// Atualiza nomes das imagens dos mapas e legenda
// As variaveis $map_id e $map foram criadas no inicio do codigo,
// antes de chamar a funcao Alteralerta que gera arquivos usados.
$image_name = "mapalerta".$map_id.".png";
$image_url=$tmp_path.$image_name;
$leg_name = "legalerta".$map_id.".png";
$leg_url=$tmp_path.$leg_name;
$map_file = "../tmp/mapalerta".$map_id.".map";

```

```

// Desenha e salva a imagem do mapa
$image=$map->draw();
$image->saveImage($img_path.$image_name);
// Desenha e salva a imagem da legenda
$leg = $map->drawLegend();
$leg->saveImage($img_path.$leg_name);
// Salva as extensoes depois de centralizar e aproximar
// uma variavel do formulario pelo valor delimitado
$new_extent = sprintf("%3.6f",$map->extent->minx)." "
    .sprintf("%3.6f",$map->extent->miny)." "
    .sprintf("%3.6f",$map->extent->maxx)." "
    .sprintf("%3.6f",$map->extent->maxy);
// Formata a escala da imagem para exibir
$scale = sprintf("%10d",$map->scale);
// Converte coordenadas selecionadas no mapa
list($x,$y) = img2map($map->width,
    $map->height,$clkpoint,$old_extent);
$x_str = sprintf("%3.6f",$x);
$y_str = sprintf("%3.6f",$y);
$qpoint = ms_newPointObj(); // cria e atualiza parametro
$qpoint->setXY($x,$y);
$nearby = EstacaoProx($qpoint,$map,$radius,$mult);
$arqgest = "agua_qualidade5.dbf";
?>

<HTML>
<script language='javascript'>
    // Gerenciamento de janelas usando JavaScript para aguardar processamento
    window.open('/mestrado/saqua/mensagem2.html','saqua');
</script>
<!-- Codigo em HTML com chamadas em PHP ----- ->
<HEAD><TITLE>SAQUA/mapalerta</TITLE></HEAD>
<BODY font face="Arial,Helvetica, sans-serif" style="font-size: 10px">
<table width="798" height="12" border="0" align="center"><td width="796"
    colspan="2"><tr><font color="#006699" face="Arial, Helvetica, sans-serif"
    style="font-size: 20px; font-weight: bold;">
    <form method=post action="<?php echo $script_name;?>"><alc>
    SAQUA - sistema baseado em software livre para an&acut;lise do comportamento
    da<br> qualidade das &acut;guas fluviais: aplicado &agrave; bacia
    do rio Para&acut;ba do Sul.</alc><br><HR></font>
<?php if( $_POST["altera"] OR preg_match("/i/",$_POST["ajuda"]) ) {
if(preg_match("/Classe/",$_POST["altera"]) OR preg_match("/Aviso/",
    $_POST["ajuda"])){ // Selecao de classe
echo "Par&acirc;metro: ".str_replace("O2","OD",$parametro).
    "\n -&Aacute;gua doce: <SELECT size=1 name=item_class >";
for ($stupla = 0; $stupla < count($cod_par_clas[$item_par]); $stupla++){
echo "\n<OPTION ";
if($stupla==$item_class){echo " selected ";}
echo "value=".$stupla.">".$cod_par_nomclas[$item_par][$stupla]." - (condi".

```

```

" &ccedil;&atilde;o) limite: (".$cod_par_cond[$item_par][$stupla.]") ".
$cod_par_lim[$item_par][$stupla.] - Obs: ".
$cod_par_avi[$item_par][$stupla.]"%";
}
echo "</SELECT>\n";
echo '<INPUT type="hidden" name="item_par" value=".$item_par." >';
echo "\n <INPUT type='submit' name='ajuda' value='Aviso'>&nbsp;";
} else {
// Selecao de parametro
echo 'Par&acirc;metro: <SELECT size=1 name=item_par >';
for ($stupla = 0; $stupla < count($cod_param) ; $stupla ++){
echo "\n<OPTION ";
if($stupla==$item_par){echo " selected ";}
echo "value=".$stupla.">".$cod_param[$stupla.] - ".$nom_param[$stupla];
}
echo "</SELECT>\n";
echo '<INPUT type="hidden" name="item_class" value=".$item_class.'" >';
echo "\n -&Aacute;gua doce: ".$padrao." - limite: ".$cond." ".$limite.
" - Obs.: ".$limobs."%";
echo "\n <INPUT type='submit' name='altera' value='Classe'>";
echo "\n <INPUT type='submit' name='ajuda' value='Dica'>&nbsp;";
}} else {
echo
"<b>.Par&acirc;metro:</b>".str_replace("O2","OD",$parametro).str_repeat("&nbsp;","3/2*(2
9-strlen(
$parametro)))."-&Aacute;gua doce: ".$padrao."\n<b>.Alerta</b>: -".
"limite: ".$cond." ".$limite." -Obs.: ".$limobs."% (";
if ($stat=="A") echo "M<u>&aacute;</u>x."; elseif ($stat=="I")
echo "M<u>&iacute;</u>n."; else echo "<u>M</u>&eacute;dia";
echo ")\n";
echo '<INPUT type="hidden" name="item_par" value=".$item_par.'" >';
echo '<INPUT type="hidden" name="item_class" value=".$item_class.'" >';
echo '<INPUT type="submit" name="altera" value="Par&acirc;m.">&nbsp;';
} ?>
<SELECT size=1 name="estat" >
<OPTION <?php if(preg_match("/A/", $stat)){echo selected;} ?> value="A">Max.
<OPTION <?php if(preg_match("/M/", $stat)){echo selected;} ?> value="M">Media
<OPTION <?php if(preg_match("/I/", $stat)){echo selected;} ?> value="I">Min.
</SELECT></b>
</tr></td><td width="600">
<input name="img" type="image" src="<?php echo $image_url;?>"
width="600" height="300" border=0></div>
<td width="196"><b>A&ccedil;&otilde;es:</b> Centrar <INPUT type=radio
name="zoom" value=0 <?php echo $pan;?>>
&nbsp;&nbsp;<INPUT type="submit" name="ajuda" value="Ajuda"><br>
Vir <INPUT type=radio name="zoom" value=1 <?php echo $zoomin;?>>
| Ir <INPUT type=radio name="zoom" value=-1 <?php echo $zoomout;?>>
| <i>Zoom</i>: <INPUT type=text name="zsize" value="<?php echo $zoomsize;?>"
size=3><br> <B>Camadas: |</B>
<input type="checkbox" name="layer[]" value="sub_bac"

```



```

"$map_id\&mes=$mes\&ano=$ano\&cod_par=$cod_par\&param="
str_replace("O2","OD",$parametro).
"\&cod_cla=$cod_class\&classe=$classe\&estat=$estat', 'saqua');"
echo "</script>";
} else {
// Envia ao saqua o pedido de exibir ajuda padrao
echo "<script language='javascript'>";
echo "window.open('/mestrado/saqua/saqua.php?ajuda=inicial','saqua');"
echo "</script>";
}
pg_close ($conex1);?>
</tr></table>
</FORM></BODY></HTML>

```

Código do arquivo *mapalertaphp.map* na sub-pasta *mapalerta*

```

# Mapfile com camada para bacias hidricas disponiveis no site da ANA.
MAP # Início do objeto MAP
# Define tamanho do mapa em tela
SIZE 600 300
# Define posição no mapa apresentado
EXTENT -46.5 -24.5 -41 -20.4 # centralizando para sub_bacia 58
IMAGECOLOR 180 180 250 # escolhendo cor do que não for feição
IMAGETYPE PNG24
SYMBOL # standard circular brush
NAME 'circle'
TYPE ELLIPSE
POINTS 1 1 END
FILLED TRUE
END
# Determina saída para web
WEB
METADATA
"wms_title" "Alertas hidricos c/Mapserver"
"wms_srs" "EPSG:4326"
END
END
# Coloca Legenda no mapa
LEGEND
STATUS ON
POSITION LR
TRANSPARENT TRUE
END
UNITS DD
# Define a projeção do mapa
PROJECTION
"proj=latlong"
"ellps=WGS84"

```

```
END
# Coloca Escala no mapa
SCALEBAR
  STATUS EMBED
  POSITION LR
  UNITS KILOMETERS
  INTERVALS 3
  TRANSPARENT TRUE
  OUTLINECOLOR 0 0 0
END
LAYER # As camadas das 8 bacias
  NAME bacia1
  TYPE POLYGON
  STATUS ON
  DATA ../mapalerta/bacia1
  LABELITEM 'BACIA'
  CLASS
    NAME 'Outras Bacias (DNAEE)'
    STYLE
      COLOR 255 255 150
      OUTLINECOLOR 0 0 0
      SYMBOL 0
    END
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
LAYER
  NAME bacia2
  TYPE POLYGON
  STATUS ON
  DATA ../mapalerta/bacia2
  LABELITEM 'BACIA'
  CLASS
    STYLE
      COLOR 255 255 150
      OUTLINECOLOR 0 0 0
      SYMBOL 0
    END
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
```

```
    POSITION CC
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
LAYER
  NAME bacia3
  TYPE POLYGON
  STATUS ON
  DATA ../mapalerta/bacia3
  LABELITEM 'BACIA'
  CLASS
    STYLE
      COLOR 255 255 150
      OUTLINECOLOR 0 0 0
      SYMBOL 0
    END
    LABEL
      COLOR 255 255 255
      OUTLINECOLOR 0 0 0
      POSITION CC
    END
  END
  PROJECTION
    "proj=latlong"
    "proj=WGS84"
  END
END
LAYER
  NAME bacia4
  TYPE POLYGON
  STATUS ON
  DATA ../mapalerta/bacia4
  LABELITEM 'BACIA'
  CLASS
    STYLE
      COLOR 255 255 150
      OUTLINECOLOR 0 0 0
      SYMBOL 0
    END
    LABEL
      COLOR 255 255 255
      OUTLINECOLOR 0 0 0
      POSITION CC
    END
  END
END
```

```
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
LAYER
  NAME bacia5
  TYPE POLYGON
  STATUS ON
  DATA ../mapalerta/bacia5
  LABELITEM 'BACIA'
  CLASS
    NAME 'Bacia 5 (DNAEE)'
    STYLE
      COLOR 255 255 255
      OUTLINECOLOR 0 0 0
      SYMBOL 0
    END
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
LAYER
  NAME bacia6
  TYPE POLYGON
  STATUS ON
  DATA ../mapalerta/bacia6
  LABELITEM 'BACIA'
  CLASS
    STYLE
      COLOR 255 255 150
      OUTLINECOLOR 0 0 0
      SYMBOL 0
    END
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
END
PROJECTION
  "proj=latlong"
```

```
"proj=WGS84"  
END  
END  
LAYER  
NAME bacia7  
TYPE POLYGON  
STATUS ON  
DATA ../mapalerta/bacia7  
LABELITEM 'BACIA'  
CLASS  
STYLE  
  COLOR 255 255 150  
  OUTLINECOLOR 0 0 0  
  SYMBOL 0  
END  
LABEL  
  COLOR 255 255 255  
  OUTLINECOLOR 0 0 0  
  POSITION CC  
END  
END  
PROJECTION  
  "proj=latlong"  
  "proj=WGS84"  
END  
END  
LAYER  
NAME bacia8  
TYPE POLYGON  
STATUS ON  
DATA ../mapalerta/bacia8  
LABELITEM 'BACIA'  
CLASS  
STYLE  
  COLOR 255 255 150  
  OUTLINECOLOR 0 0 0  
  SYMBOL 0  
END  
LABEL  
  COLOR 255 255 255  
  OUTLINECOLOR 0 0 0  
  POSITION CC  
END  
END  
PROJECTION  
  "proj=latlong"  
  "proj=WGS84"  
END  
END
```

```
LAYER # A camada da bacia rio Pb.do Sul
NAME fixo
TYPE POLYGON
STATUS DEFAULT # Sempre aparece
DATA ../mapalerta/sub_bacias5
LABELITEM 'SUB_BACIA'
CLASSITEM 'SUB_BACIA'
CLASS
  EXPRESSION '58'
  NAME 'Bacia do rio Pb. do Sul'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 160 255 160
  END
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
LAYER # As demais sub-bacias
NAME sub_bac
TYPE POLYGON
STATUS ON
DATA ../mapalerta/sub_bacias5
LABELITEM 'SUB_BACIA'
CLASSITEM 'SUB_BACIA'
CLASS
  EXPRESSION ('[SUB_BACIA]' ne '58')
  NAME 'Sub-bacias 5X (DNAEE)'
  STYLE
    OUTLINECOLOR 100 100 100
    COLOR 200 200 200
  END # Fim do objeto STYLE
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
```

```
END
LAYER # Os rios da sub-bacia 58
NAME rios
TYPE LINE
STATUS ON
DATA ../mapalerta/hidrografia5
LABELITEM 'NOME'
CLASSITEM 'SUB_BACIA'
CLASS
EXPRESSION /5/
NAME 'Rios'
STYLE
  COLOR 0 0 255
  OUTLINECOLOR 0 0 0
  SYMBOL 0
END
LABEL
  COLOR 255 255 255
  OUTLINECOLOR 0 0 255
  POSITION CC
END
END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
# Camada de alertas para estacoes de qualidade da agua
LAYER
NAME estacao
TYPE POINT
STATUS DEFAULT
LABELCACHE on
TOLERANCEUNITS miles
# Abaixo linha 313 - alteravel para temporarios (AIG)
DATA ../mapalerta/agua_qualidade5
LABELITEM 'ESTACAO'
CLASSITEM 'ALERTA'
TEMPLATE err_mapalerta.php
CLASS
EXPRESSION '0'
NAME 'Parâmetro SEM problemas'
LABEL
  COLOR 255 255 255
  OUTLINECOLOR 0 0 0
  POSITION CC
END
STYLE
  COLOR 0 255 0
```



```
    OUTLINECOLOR 0 0 0
    SYMBOL 'circle'
    SIZE 15
  END
END
CLASS
  EXPRESSION '1'
  NAME 'Parâmetro em observação'
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
  STYLE
    COLOR 255 255 0
    OUTLINECOLOR 0 0 0
    SYMBOL 'circle'
    SIZE 15
  END
END
CLASS
  EXPRESSION '2'
  NAME 'Parâmetro COM problemas'
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
  STYLE
    COLOR 255 0 0
    OUTLINECOLOR 0 0 0
    SYMBOL 'circle'
    SIZE 15
  END
END
CLASS
  EXPRESSION '3'
  NAME 'Parâmetro AUSENTE'
  LABEL
    COLOR 255 255 255
    OUTLINECOLOR 0 0 0
    POSITION CC
  END
  STYLE
    COLOR 100 100 100
    OUTLINECOLOR 0 0 0
    SYMBOL 'circle'
    SIZE 15
  END
END
```

```

END
PROJECTION
  "proj=latlong"
  "proj=WGS84"
END
END
#-----
END # Fim do objeto MAP e do mapfile

```

Código do arquivo *instala.bat* na sub-pasta *instala*

```

@ECHO OFF
REM Este arquivo de lote instala o Mapserver for Windows copiado no
REM disco C:, por descompactacao junto com as configuracoes do SAQUA.
ECHO ----- Executa o instalador do Apache/Mapserver -----
c:
cd \ms4w
REM call c:\ms4w\apache-uninstall.bat
call c:\ms4w\apache-install.bat
ECHO ----- Executa o instalador do PostgreSQL -----
ECHO A seguir abrira a janela PostgreSQL-8, apos intalar retornara para esta janela...
cd \mestrado\instala
rem A linha abaixo evita iteracoes, porem a criacao carga teria de ser depois em bat
if not exist "%USERPROFILE%\..\saqua" (
  msixexec /i postgresql-8.2-int.msi /qb INTERNALLAUNCH=1 ADDLOCAL=server,psql
  SERVICEDOMAIN="%COMPUTERNAME%" SERVICEACCOUNT="saqua"
  CREATESERVICEUSER=1 SERVICEPASSWORD="mestrado"
  SUPERPASSWORD="mestrado" BASEDIR="c:\postgres" ) else (
  msixexec /i postgresql-8.2-int.msi /qb INTERNALLAUNCH=1 ADDLOCAL=server,psql
  SERVICEDOMAIN="%COMPUTERNAME%" SERVICEACCOUNT="saqua"
  SERVICEPASSWORD="mestrado" SUPERPASSWORD="mestrado"
  BASEDIR="c:\postgres" )
ECHO Caso nao tenha ocorrido problemas foram criados o superusuario "postgresql" e
ECHO o usuario para gerencia de serviço "saqua", ambos com senha inicial "mestrado".
ECHO ----- CRIANDO E PREENCHENDO O BANCO DE DADOS -----
call \ms4w\Apache\cgi-bin\php-cgi.exe -f \mestrado\instala\cria_e_preenche_banco.php
ECHO Atualizando o banco de dados com as series historicas do HIDROWEB e gerando
alertas
ECHO O andamento esta sendo direcionado para o arquivo "\mestrado\tmp\carga.html".
ECHO Se desejar acompanhe o crescimento deste arquivo, ou simplesmente aguarde ...
call \ms4w\Apache\cgi-bin\php-cgi.exe -f \mestrado\saqua\carga.php
>\mestrado\tmp\carga.html
ECHO "A INSTALACAO TERMINOU - VERIFIQUE SE NAO HOUE PROBLEMAS"

```

Código do arquivo *cria_e_preenche_banco.php* na sub-pasta *instala*

<?php

```

echo "Assumindo que foi instalado o Pgsq1 com usuario Postgres e senha mestrado,\n";
echo "a qual pode ser alterada no futuro, sera criado o usuario 'saqua' com\n";
echo "senha mestrado e o banco do 'saqua', onde ficam as tabelas do SAQUA.\n";
echo "\n===== \n";
echo "\n ---- PARA O SISTEMA DE ACOMPANHAMENTO DE QUALIDADE DE AGUA
- SAQUA ---- \n";
$conex1 = pg_connect ("host = localhost
port = 5432
user = postgres
password = mestrado")
or die ("Falha na conexão!".pg_last_error());
$result1 = pg_query ($conex1, "CREATE USER saqua WITH NOSUPERUSER
CREATEDB CREATEROLE
        INHERIT PASSWORD 'mestrado'; ");
if($result1) echo "\nUsuario 'saqua' criado com sucesso";
else echo "\nUsuario 'saqua' pode ter sido criado anteriormente. Verifique !";
$result2 = pg_query ($conex1, "CREATE DATABASE saqua WITH OWNER=saqua; ");
if($result2) echo "\nNome do Banco 'saqua' criado com sucesso";
else echo "\nNome do Banco 'saqua' pode ter sido criado anteriormente. Verifique !";
echo "\n----- \n";
pg_close ($conex1);
$conex1 = pg_connect ("host = localhost
port = 5432
user = saqua
password = mestrado")
or die ("\nFalha na conexão!".pg_last_error());
echo "Criando Tabelas com alguns dados:\n";
$result1 = pg_query ($conex1, "
CREATE TABLE alerta (
        cod_estac    int,
        cod_param    int,
        cod_class    int,
        mes          int,
        ano          int,
        media        real,
        maximo       real,
        minimo       real,
        ocorre       int,
        situa_med    varchar(1),
        situa_max    varchar(1),
        situa_min    varchar(1)
);");
if($result1) echo "Tabela Alerta criada com sucesso\n";
$result1 = pg_query ($conex1, "
CREATE TABLE classes (
        cod_class    int,
        classe       varchar(8)
);");
if($result1) { echo "Tabela Classes criada com sucesso\n";

```

```

$result2 = pg_query ($conex1, "insert into classes values ( 3, 'Classe 2' ); ");
$result2 = pg_query ($conex1, "insert into classes values ( 2, 'Classe 1' ); ");
$result2 = pg_query ($conex1, "insert into classes values ( 4, 'Classe 3' ); ");
$result2 = pg_query ($conex1, "insert into classes values ( 5, 'Classe 4' ); ");
$result2 = pg_query ($conex1, "insert into classes values ( 11, 'V.Baixo' ); ");
$result2 = pg_query ($conex1, "insert into classes values ( 12, 'V.Alto' ); ");
$result1 = pg_query ($conex1, "
CREATE TABLE criterio (
    cod_param    int,
    cod_class    int,
    cond         varchar(2),
    limite      real,
    aviso       real
);");
if($result1) { echo "Tabela Criterio criada com sucesso\n";
$result2 = pg_query ($conex1, "insert into criterio values ( 19, 3, '<', 5, 25 ); ");
$result2 = pg_query ($conex1, "insert into criterio values ( 19, 2, '<', 6, 25 ); ");
$result2 = pg_query ($conex1, "insert into criterio values ( 19, 4, '<', 4, 25 ); ");
$result2 = pg_query ($conex1, "insert into criterio values ( 19, 5, '<', 2, 25 ); ");
$result2 = pg_query ($conex1, "insert into criterio values ( 18, 3, '>', 5, 25 ); ");
$result2 = pg_query ($conex1, "insert into criterio values ( 18, 2, '>', 3, 25 ); ");
$result2 = pg_query ($conex1, "insert into criterio values ( 18, 4, '>', 10, 25 ); ");
$result2 = pg_query ($conex1, "insert into criterio values ( 11, 11, '<', 6, 25 ); ");
$result2 = pg_query ($conex1, "insert into criterio values ( 11, 12, '>', 9, 25 ); ");}
$result1 = pg_query ($conex1, "
CREATE TABLE estacao (
    cod_estac    int,
    cod_class    int,
    arq_orig     varchar(80)
);");
if($result1) { echo "Tabela Estacao criada com sucesso\n";
$result2 = pg_query ($conex1, "insert into estacao values (
58099000,3,'../sagua/estacoes/58099000/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58183000,3,'../sagua/estacoes/58183000/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58235100,3,'../sagua/estacoes/58235000/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58235100,3,'../sagua/estacoes/58235100/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58250000,3,'../sagua/estacoes/58250000/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58321000,3,'../sagua/estacoes/58321000/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58480500,3,'../sagua/estacoes/58480500/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58630002,3,'../sagua/estacoes/58630002/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58770000,3,'../sagua/estacoes/58770000/QUALAGUA.TXT');");

```

```

$result2 = pg_query ($conex1, "insert into estacao values (
58790002,3,'../saqua/estacoes/58790000/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58790002,3,'../saqua/estacoes/58790002/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58880001,3,'../saqua/estacoes/58880001/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58960000,3,'../saqua/estacoes/58960000/QUALAGUA.TXT');");
$result2 = pg_query ($conex1, "insert into estacao values (
58974000,3,'../saqua/estacoes/58974000/QUALAGUA.TXT');");}
$result1 = pg_query ($conex1, "
CREATE TABLE param (
cod_param int,
unidade varchar(7),
parametro varchar(20)
);");
if($result1) { echo "Tabela Param criada com exito\n";
$result2 = pg_query ($conex1, "insert into param values ( 19, 'mg/L O2',
'Oxigenio Dissolvido ');");
$result2 = pg_query ($conex1, "insert into param values ( 11, '-', 'pH da agua');");
$result2 = pg_query ($conex1, "insert into param values ( 18, 'mg/L O2',
'Dem. Bioq.de O2');");}
echo "\n===== \n";
?>

```

Código do arquivo *desinstala.bat* na sub-pasta *instala*

```

@ECHO OFF
REM Este arquivo de lote desinstala o Mapserver for Windows copiado no
REM disco C:, e o Postgresql, apagando seus dados e tambem o SAQUA.
ECHO ----- Executa o desinstalador do Apache/Mapserver -----
c:
cd \ms4w
call c:\ms4w\apache-uninstall.bat
cd \
ECHO ----- Executa o desinstalador do PostgreSQL -----
cd \mestrado\instala
net stop "PostgreSQL Database Server 8.2"
ECHO A seguir abrira a janela PostgreSQL-8, apos desintalar retornara para esta janela...
msiexec /x postgresql-8.2-int.msi /qb
ECHO ----- Apaga todos arquivos intalados e o SAQUA-----
cd \
if EXIST "%ALLUSERSPROFILE%\Menu Iniciar\Programas\SAQUA\" (RD /S /Q
"%ALLUSERSPROFILE%\Menu Iniciar\Programas\SAQUA")
rmdir c:\ms4w /S /Q
rmdir c:\postgres /S /Q
ECHO ----- SAQUA TERMINANDO DE SER REMOVIDO -----
REM A ultima linha apaga os arquivos finais do SAQUA junto com este script.

```

rmdir c:\mestrado /S /Q

Código do arquivo *index.html* na sub-pasta *saqua*

```
<html><head>
<title>SAQUA - SISTEMA BASEADO EM SOFTWARE LIVRE PARA AN&Aacute;LISE
DO
  COMPORTAMENTO DA QUALIDADE DAS &Aacute;GUAS FLUVIAS:
  APLICADO &Agrave; BACIA DO RIO PARA&Iacute;BA DO SUL.</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<frameset rows="430,*" framespacing="0" frameborder="NO" border="0" >
  <frame name="mapalerta" scrolling="NO" noresize
    src="/mestrado/mapalerta/mapalerta.php">
  <frame name="saqua" src="/mestrado/saqua/mensagem2.html">
</frameset><noframes><body></body></noframes></html>
```

Código do arquivo *mensagem2.html* na sub-pasta *saqua*

```
<html><head></head><body>
<p align="center"><i><b><font size="4">Aguarde - processando...</font></b>
<p align="center">Caso o SAQUA esteja
corretamente instalado <p align="center"> e todos os servi&ccedil;os
necess&aacute;rios configurados e inicializados não ocorrerá erros.</i></html>
```

Código do arquivo *saqua.php* na sub-pasta *saqua*

```
<?php
/--Arquivo em PHP para interagir diretamente com a interface
/--de servidor web p/alertas hidricos (arquivo mapalerta.php)
//-----
/-- Funcao utilizada:
/-- Voltaregdb - retorna o registro do campo numerico
function Voltaregdb($arquivo,$campo,$reg) {
  $strfilename = "../tmp/".$arquivo ;
  $reg_campo = "";
  $db=dbase_open($strfilename, 0);
  if ($db) {
    $rec = dbase_get_record($db, $reg[0]+1);
    $reg_campo = $rec[$campo];
    dbase_close($db);
  }
  return $reg_campo;
}
//-----
// ----- Inicio de codigo: - Parte em PHP
```

```
// Todo codigo php, fora funcoes esta incorporado ao codigo HTML
?>
<HTML>
<!-- Codigo em HTML com chamadas em PHP ----- ->
<HEAD><TITLE>SAQUA</TITLE></HEAD>
<BODY font face="Arial,Helvetica, sans-serif" stylet="font-size: 10px">
<table border="1" width="796" heigth="84" align="center">
<?php
if ( $_GET['ajuda'] ) {
    $ajuda=$_GET['ajuda'] ;
    if (preg_match("/mapalerta/", $ajuda)){
        echo "<td bgColor=#ffffa0 width='92%'> Sobre a <b>navega&ccedil;&atilde;o".
            "</b> no mapa: - O link <U>Iniciar</U> reinicia a p&aacute;gina como ".
            "pela primeira vez - <b>A&ccedil;&otilde;es</b>: \n<br> Centrar -> ".
            "Utiliza o ponto que clicar no mapa para o centro, ou pode-se aplicar".
            " o centro atual.\n<br> Vir -> Com o Fator escolhido aproxima-se ".
            "centrando | Ir -> Com o Fator escolhido afasta-se centrando \n<br>".
            "Fator -> Valor utilizado nas acoes de ir e vir linearmente ".
            "<b>Camadas</b>: -> Oculta ou exhibe camadas ao selecionar <br>".
            "Bot&otilde;es: Aplica -> Corresponde a clicar exatamente no centro ".
            "do mapa | Ajuda -> Exibe este menu";
        echo "\n<td bgColor=#ffffa0> Menu:<br>".
            "<a href='/mestrado/saqua/saqua.php?ajuda=inicial'>Objetivos</a><br>\n".
            "<a href='/mestrado/saqua/saqua.php?ajuda=parametro'>Par&acirc;metros</a>".
            "<br>\n<a href='/mestrado/saqua/saqua.php?nearby=1&ajudalocal=1'>".
            "Gr&aacute;ficos</a><br>\n<a href='/mestrado/saqua/saqua.php?ajuda=" .
            "finais'>Final</a><p></tr><tr>\n";
    } elseif (preg_match("/parametro/", $ajuda)){
        echo "<td bgColor=#ffffa0 width='92%'> A escolha dos <b>par&acirc;metros".
            "</b> de acompanhamento h&iacute;drico: - Através dos bot&otilde;es ".
            "de sele&ccedil;&atilde;o visualizados: <br>- Estat&iacute;stica -> ".
            "Ao final da primeira linha (M&aacute;ximo, M&eacute;dia ou M".
            "&iacute;nimo); - Mês e ano -> Após o indicador <b>ref.</b><br> ".
            "- Através dos bot&otilde;es de sele&ccedil;&atilde;o visualizados ".
            "ap&oacute;s selecionar o bot&atilde;o <b>Par&acirc;m.</b> ou <b>".
            "Classe</b> (dentro de Par&acirc;m.): <br> - Par&acirc;metro -> ".
            "C&oacute;digo e nome do par&acirc;metro; - Classe -> De &aacute;gua ".
            "doce junto a condi&ccedil;&ocirc;es e limites adotados.<br>Bot".
            "&otilde;es: Aplica -> Atualiza os dados, utilizando a navega&ccedil;".
            "&atilde;o no mapa | Dica (em Par&acirc;m.) -> Exibe este menu";
        echo "\n<td bgColor=#ffffa0> Menu:<br>".
            "<a href='/mestrado/saqua/saqua.php?ajuda=inicial'>Objetivos</a><br>\n".
            "<a href='/mestrado/saqua/saqua.php?ajuda=mapalerta'>Navega&ccedil;&atil".
            "de;o</a><br>\n<a href='/mestrado/saqua/saqua.php?nearby=1&ajudalocal=1".
            ">Gr&aacute;ficos</a><br>\n<a href='/mestrado/saqua/saqua.php?ajuda=" .
            "finais'>Final</a><p></tr><tr>\n";
    } elseif (preg_match("/inicial/", $ajuda)) { // Aqui trata a ajuda inicial
        echo "<td bgColor=#ffffa0 width='92%'><b>Mensagem Inicial</b>: Esta ferramen".
            "ta consiste em uma interface desenvolvida com o <b>objetivo</b> de "
```

```

" fazer o acompanhamento hídrico por parâmetros de qualidade da água ".
" através do uso de software livre para a bacia do Rio Paraíba do
" Sul, seguindo a resolução CONAMA 375/05. O estudo busca
" contribuir nas questões relacionadas à análise espacial ".
" o temporal de parâmetros de qualidade da água,
" proposto na dissertação de Mestrado em
" Geomática na UERJ por Antonio Ignacio Gloria. ";
echo "\n<td bgcolor=#ffffa0> Menu:<br>".
" <a href='/mestrado/saqua/saqua.php?ajuda=mapalerta'>Navegar;"
" o</a><br>\n<a href='/mestrado/saqua/saqua.php?ajuda=parametro'>Parâmetros</a><br>\n<a href='/mestrado/saqua/saqua.php?nearby=1&ajudalocal="
" 1'>Gráficos</a><br>\n<a href='/mestrado/saqua/saqua.php?ajuda="
" finais'>Final</a><p></tr><tr>\n";
} elseif (preg_match("/finais/", $ajuda)) { // Aqui trata a ajuda inicial
echo "\n<td bgcolor=#ffffa0 width='92%'><b>Mensagem Final</b>: Embora esta fer-
" ramenta permita alterar as classes de água doce dos parâmetros;"
" metros, cumpre lembrar que as estações selecionadas
" das pertencem a Classe 2, possibilitando fazer simulações
" de uso. As opções de escolha por seleção
" foram cadastradas no banco de dados em sua criação, não
" cessitando que os mesmos sejam carregados, conforme descrito na dis-
" erta, onde está documentada esta aplicação. |
" Aviso (em Classe) -> Exibe este menu | Um bom uso na utilização;"
" o desta interface e que seja bastante agradável! ";
echo "\n<td bgcolor=#ffffa0>Menu:<br>".
" <a href='/mestrado/saqua/saqua.php?ajuda=inicial'>Objetivos</a><br>\n".
" <a href='/mestrado/saqua/saqua.php?ajuda=mapalerta'>Navegar;"
" o</a><br>\n<a href='/mestrado/saqua/saqua.php?ajuda=parametro'>Parâmetros</a><br>\n<a href='/mestrado/saqua/saqua.php?nearby=1&ajudalocal="
" 1'>Gráficos</a><p></tr><tr>";
}} elseif ($_GET['nearby']) { // Aqui trata seleção da estação
if (!$_GET['ajudalocal']){
$conex2 = pg_connect ("host = localhost
port = 5432
dbname = saqua
user = saqua
password = mestrado")
or die ("Falha na conexão!".pg_last_error());
$map_id = $_GET['map_id'];
$arqest = "agua_q".$map_id.".dbf";
$nearby[0] = $_GET['nearby'];
$situac = Voltaregdb($arqest,12,$nearby);
$cod_est= Voltaregdb($arqest,0,$nearby);
$estacao= Voltaregdb($arqest,6,$nearby);
if( $situac != " "){
if( $situac == 0) { $colore="#a0ffa0"; $coloreg="00ff00"; } else {
if( $situac == 1) {
$colore="#ffffa0"; $coloreg="ffcc00";} else { if( $situac == 2) {
$colore="#ffa0a0"; $coloreg="ff0000";} else {$colore="#a0a0a0";}}}

```



```

$mes=intval($_GET['mes']);
$ano=intval($_GET['ano']);
$cod_par=$_GET['cod_par'];
$param=$_GET['param'];
$cod_cla=$_GET['cod_cla'];
$classe=$_GET['classe'];
$estat=$_GET['estat'];
if( $situac <= 3) { //trata a estação selecionada
  $nmeses=24; // arbitra historico 2 anos antecedendo o periodo selecionado
  if( $nmeses>$mes ) {
    $mes_ini=12-($nmeses-$mes-1)%12;} else $mes_ini=$mes-$nmeses+1;
  $ano_ini=$ano-intval(($nmeses+11-$mes)/12);
  $mes_fim = $mes ; $ano_fim = $ano;
  // Consulta historicos de alerta - se estat="M" h_val=media
  if ($estat=="A") {
    $result1 = pg_exec ($conex2, "select mes, ano, maximo, ocorre from
      alerta where cod_estac=$cod_est and cod_class=$cod_cla and
      cod_param=$cod_par and (ano*100+mes>=$ano_ini*100+$mes_ini) and
      (ano*100+mes<=$ano_fim*100+$mes_fim);");
  } elseif ($estat=="I") {
    $result1 = pg_exec ($conex2, "select mes, ano, minimo, ocorre from
      alerta where cod_estac=$cod_est and cod_class=$cod_cla and
      cod_param=$cod_par and (ano*100+mes>=$ano_ini*100+$mes_ini) and
      (ano*100+mes<=$ano_fim*100+$mes_fim);");
  } else { // Condicao de $estat="M"
    $result1 = pg_exec ($conex2, "select mes, ano, media, ocorre from
      alerta where cod_estac=$cod_est and cod_class=$cod_cla and
      cod_param=$cod_par and (ano*100+mes>=$ano_ini*100+$mes_ini) and
      (ano*100+mes<=$ano_fim*100+$mes_fim);");
  } //abaixo procura proxima data e anterior
  $result2 = pg_exec ($conex2, "select mes,ano from alerta where
    cod_estac=$cod_est and cod_class=$cod_cla and cod_param=$cod_par
    and ocorre>0 and (ano*100+mes>$ano_fim*100+$mes_fim) limit 1;");
  if ( pg_numrows ($result2) ) {
    $proxocor=pg_result ($result2, 0, 0)."/".pg_result ($result2, 0, 1);
  } else $proxocor="inexiste";// Busca de valores limites
  $result2 = pg_query ($conex2, "select cond,limite,aviso from
    criterio where cod_param=$cod_par and cod_class=$cod_cla");
  $cond=pg_result ($result2, 0, 0);
  $limite=pg_result ($result2, 0, 1);
  $aviso=pg_result ($result2, 0, 2);
  if (preg_match("/</", $cond)) {
    $prod_av=$aviso/100+1;} else {$prod_av=-$aviso/100+1;};
  $val_aviso=$limite*$prod_av;
  $result2 = pg_exec ($conex2, "select mes,ano from alerta where
    cod_estac=$cod_est and cod_class=$cod_cla and cod_param=$cod_par
    and ocorre>0 and (ano*100+mes<$ano_ini*100+$mes_ini) order by
    ano*100+mes desc limit 1;");
  if ( pg_numrows ($result2) ) {

```

```

    Santocor=pg_result ($result2, 0, 0)."/".pg_result ($result2, 0, 1);
} else $antocor="inexiste";
$tot_tuplas1 = pg_numrows ($result1);
$vals="";
$labels="";
$estac=$estacao;
$inicio=$mes_ini."/".substr($ano_ini,0,4);
$fim=$mes_fim."/".substr($ano_fim,0,4);
if($tot_tuplas1>0) {
    $h_mes=pg_result ($result1, 0 , 0);
    $h_ano=pg_result ($result1, 0 , 1);
    $h_val=pg_result ($result1, 0 , 2);
    $dif_mes = ($h_mes-$mes_ini)+12*($h_ano-$ano_ini);
} else {
    $h_mes=$mes_ini;
    $h_ano=$ano_ini;
    $hval=0;
    $dif_mes=$nmeses;
}
$zera_mes=0; // Zera dados no grafico ate ocorrencia
if ( $dif_mes>0 ) {
    for ( $i = 0 ; $i < $dif_mes ; $i ++ ) {
        $vals=$vals."0"; // Colocando dado zerado no grafico
        if (!(($i==$dif_mes-1 and $tot_tuplas1==0)) $vals=$vals."!" ;
        $labels=$labels.(1+($mes_ini+$i-1)%12)."/".
        ($ano_ini+intval((($mes_ini+$i-1)/12)));
        if (!(($i==$dif_mes-1 and $tot_tuplas1==0)) $labels=$labels.";" ;
    }; // Fim do "for" para preenchimento de zeros antes dos dados
    $zera_mes=$zera_mes+$dif_mes;
} // Abaixo cria linhas para apresentacao de dados
$dado_lin[1]=str_repeat("&nbsp;" ,4).
"\n| m&ecirc;s/ano - ocorr. - valor &nbsp;|&nbsp;";
$dado_lin[2]=str_repeat("&nbsp;" ,4)."\n|&nbsp;";
$dado_lin[3]=str_repeat("&nbsp;" ,4)."\n|&nbsp;";
$dado_lin[4]=str_repeat("&nbsp;" ,4)."\n|&nbsp;";
$dado_lin[5]=str_repeat("&nbsp;" ,4)."\n|&nbsp;";
$col=1; // preenche meses seguintes em graficos e tabela
// Volta 1 mes do ano para dar continuidade
$h_mes=($h_mes-1)%12; $h_ano=$h_ano-intval((($h_mes)/12);
for ($tupla = 0; $tupla < $nmeses-$zera_mes; $tupla++){
    $h_mes_ant=$h_mes; $h_ano_ant=$h_ano;
    if ($tupla<$tot_tuplas1) {
        $h_mes=pg_result ($result1, $tupla, 0);
        $h_ano=pg_result ($result1, $tupla, 1);
        $h_val=pg_result ($result1, $tupla, 2);
        $h_oco=pg_result ($result1, $tupla, 3);
    } else { if ($tupla>0){ // Valores nulos no final
        $h_mes=1+$h_mes%12;if ($h_mes==1) $h_ano=$h_ano+1;
        $h_val=0; $h_oco=0;}}
}

```

```

// Variaveis de troca para preencher zeros intermediarios ===
$dif_mes = ($h_mes-$h_mes_ant)+12*($h_ano-$h_ano_ant);
$h_val2=$h_val; $h_oco2=$h_oco;
$h_mes2=$h_mes; $h_ano2=$h_ano;
for ($i = 0; $i < $dif_mes; $i++) { // Preenche dados e zeros
  if ($i+1==$dif_mes) {$h_val=$h_val2; $h_oco=$h_oco2;
    $h_mes=$h_mes2; $h_ano=$h_ano2;
  } else { $zera_mes++; // Trata zeros intermediarios
    $h_mes=1+($h_mes_ant+$i)%12; $h_val=0; $h_oco=0;
    $h_ano=$h_ano_ant+intval(($h_mes_ant+$i)/12);}
  $vals=$vals.$h_val;
  if ($tupla < $nmeses-1-$zera_mes+$dif_mes-$i-1) $vals=$vals."!";
  if ($h_val>0) { // preenche dados e grafico
    $dado_lin[($col%5)+1]=$dado_lin[($col%5)+1].
    str_repeat("&nbsp;",4-2*strlen($h_mes)).$h_mes."/".$h_ano." - ".
    $h_oco." - ".$h_val.str_repeat("&nbsp;",18-2*strlen($h_val));
    if ($h_val>intval($h_val)) $dado_lin[($col%5)+1]=
      $dado_lin[($col%5)+1]."&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;"; else
      $dado_lin[($col%5)+1]=$dado_lin[($col%5)+1]."&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;";
    $col++; $labels=$labels.$h_mes."/".substr($h_ano,0,4);
    if ($tupla < $nmeses-1-$zera_mes) $labels=$labels.";";
    } else { //Preenche graficos sem ocorrencia cadastrada
      $labels=$labels.$h_mes."/".substr($h_ano,0,4);
      if ($tupla < $nmeses-1-$zera_mes+$dif_mes-$i-1)
        $labels=$labels.";";
    } // Fim do "for" para preenchimento de dados e zeros para cada alerta
  } // Fim do "for" para preenchimento de todos alertas no período de n meses
  $dado_lin[($col%5)+1]=$dado_lin[($col%5)+1].str_repeat("- ",19); $col++;
  $dado_lin[($col%5)+1]=$dado_lin[($col%5)+1].ocorr. ant. => ".$santocor; $col++;
  $dado_lin[($col%5)+1]=$dado_lin[($col%5)+1].ocorr. post. => ".$sproxocor; $col++;
  if (!$_GET['dados']) {
    echo "<td width='92%' rowspan='2' >";
    echo '<a href=" ../cgi-bin/owtchart.exe?Type=Combo_Line_Bar'.
      '&SetColors=ff0000!ffff00&NumSets=2&YAxis2=0'.
      '&W=710&H=560&NumPts='.$nmeses.'&Vals='.str_repeat($limite.'!', $nmeses).
      str_repeat($sval_avisos.'!', $nmeses).$vals.str_repeat('!0', $nmeses).&Title='.
      'Esta&ccedil;&atilde;o: '.trim($stac).' - Par&acirc;metro: '.
      $param.' - de '$inicio.' at&eacute; '$fim.'&XLabels='.$labels.
      '&StackType=BESIDE" target="_blank">
    <img src = " ../cgi-bin/owtchart.exe?Type=Combo_Line_Bar'.
      '&SetColors=ff0000!ffff00&NumSets=2&YAxis2=0'.
      '&W=710&H=132&NumPts='.$nmeses.'&Vals='.str_repeat($limite.'!', $nmeses).
      str_repeat($sval_avisos.'!', $nmeses).$vals.str_repeat('!0', $nmeses).&Title='.
      'Esta&ccedil;&atilde;o: '.trim($stac).' - Par&acirc;metro: '.
      $param.' - de '$inicio.' at&eacute; '$fim.'&XLabels='.$labels.
      '&StackType=BESIDE"></a>';
  } else {
    echo '<td width="92%" rowspan="2" bgColor='.$scolore.'><p align="center"><b>'.
      'Esta&ccedil;&atilde;o: '$stac.' - <b>Par&acirc;metro:</b>'.

```

```

    $param.' - de '.$inicio.' at&eacute; '.$fim;
    echo '<HR width="710" height="2">';
    echo $dado_lin[1]."<br>".$dado_lin[2]."<br>".$dado_lin[3]."<br>".
    $dado_lin[4]."<br>".$dado_lin[5]."<br>";
    echo '</td>';
}
} else {
    echo '<td width="92%" rowspan="2" bgColor="$colore">';
    echo "<p>N&Atilde;O H&Aacute; tipo cadastrado de alerta > 3 - Verifique banco";
} //fim do else e do if
pg_close ($conex2);}
else { // A ajuda local pode ser chamada de fora habilitando o nearby
    echo '<td bgColor=#ffffa0 width="92%" rowspan="2">';
    echo "Os <b>gr&aacute;ficos</b> s&atilde;o obtidos ao clicar sobre ".
        "o c&iacute;rculo de localiza&ccedil;&atilde;o das esta&ccedil;".
        "&otilde;es possuem o formato de data no eixo horizontal no forma".
        "to 'mes/ano' quando o numero de ocorr&ecirc;ncias for maior que ".
        "zero, e 'mes|ano' quando o valor for calculado por n&atilde;e en".
        "contrar ocorr&ecirc;ncias. Caso n&atilde;o haja dado anterior ca".
        "dastrado repete a primeira data. Os links laterais: <U>Dados</U>".
        "-> Mostra os valores obtidos; <U>Gr&aacute;fico</U>-> Retorna ".
        "ao gr&aacute;fico; <U>Carga</U>-> Abre a recarga para a esta".
        "&ccedil;&atilde;o; <U>Google Map</U>-> Abre o Google Map cen".
        "trado na esta&ccedil;&atilde;o; <U>Ajuda</U>-> Esta ".
        "Ajuda"; }
if ($_GET['map_id']) { // para chamar apenas ajuda local nao informe map_id
    echo '<td align="center">';
    if (!$_GET['ajudalocal']) {
        echo '<a href="saqua.php?nearby='.$_GET['nearby'].'&map_id='.
            $_GET['map_id'].'&mes='.$_GET['mes'].'&ano='.$_GET['ano'].
            '&cod_par='.$_GET['cod_par'].'&param='.$_GET['param'].'&cod_cla='.
            $_GET['cod_cla'].'&classe='.$_GET['classe'].'&estat='.$_GET['estat'].
            '&ajudalocal=1';
        if ($_GET['dados']) { echo '&dados=1">';} else echo "">';
        echo 'Ajuda</a><br>';}
    echo '<a href="saqua.php?nearby='.$_GET['nearby'].'&map_id='.$_GET['map_id'].
        '&mes='.$_GET['mes'].'&ano='.$_GET['ano'].'&cod_par='.$_GET['cod_par'].
        '&param='.$_GET['param'].'&cod_cla='.$_GET['cod_cla'].'&classe='.
        $_GET['classe'].'&estat='.$_GET['estat'];
    if (!$_GET['dados']) { echo '&dados=1">Dados</a>';}
    else echo "">Gr&aacute;fico</a>';
    if ($_GET['ajudalocal']) {
        echo '<br><a href="saqua.php?nearby='.$_GET['nearby'].
            '&map_id='.$_GET['map_id'].'&mes='.$_GET['mes'].'&ano='.$_GET['ano'].
            '&cod_par='.$_GET['cod_par'].'&param='.$_GET['param'].'&cod_cla='.
            $_GET['cod_cla'].'&classe='.$_GET['classe'].'&estat='.$_GET['estat'];
        if ($_GET['dados']) { echo '&dados=1">Dados</a>';} else
            echo "">Gr&aacute;fico</a>';}
    echo '<br><a href="carga.php?cod_estac='.$cod_est.'" target="_blank">'.

```

```

' Carga</a><br><a href="google_estac_off.php?cod=!'$_GET['nearby']'.
'" target="_blank"> Google<br>Map</a></tr><tr></td>';
} else {
echo "\n<td bgColor=#ffffa0>Menu:<br>".
" <a href='/mestrado/saqua/saqua.php?ajuda=inicial'>Objetivos</a><br>\n".
" <a href='/mestrado/saqua/saqua.php?ajuda=mapalerta'>Navega&ccedil;&atil".
" de;o</a><br><a href='/mestrado/saqua/saqua.php?ajuda=parametro'>Par",
" &acirc;metros</a><br>\n<a href='/mestrado/saqua/saqua.php?ajuda="
" finais'>Final</a><p></tr><tr>"; }
if (!$_GET['ajudalocal']) {
echo '<td><table border="0" cellpadding="0" cellspacing="0" width="100%">'.
'<tr><td width="100%" colspan="2"><font size=2>'.
' Limites:</td></tr><tr><td width="25%" rowspan="2">'.
'<hr align="left" width="9" noshade size="2" color="#FF0000">'.
'<hr align="left" width="9" noshade size="2" color="#FFFF00"></td>'.
'<td width="77%"><font size=2>$.limite.</td></tr><tr><td width="77%"><font
size=2>$.val_avisos.</td></tr></table>';}
}??>
</tr></table>
</FORM></BODY></HTML>

```

Código do arquivo *carga.php* na sub-pasta *saqua*

```

<?php
/--Arquivo em PHP para carregar alertas hidricos no SAQUA
//-----
/-- Funcoes utilizadas: Incluir e Gerar
/--Incluir - Inclui alerta no banco e retorna contador incrementado - recebe:
/-- contador,codigo est.,parametro,media,maximo,minimo,ocorrencias,mes,ano (9)
function Incluir($cont,$cod_estac,$param,$media,$max,$min,$ocor,$mes,$ano) {
// buscar limites para parametros
$conex1 = pg_connect ("host = localhost
port = 5432
dbname = saqua
user = saqua
password = mestrado")
or die ("Falha na conexão!".pg_last_error());
$result2 = pg_query ($conex1, "select cod_class,cond,limite,aviso from
criterio where cod_param=$param");
$tot_tuplas2 = pg_numrows ($result2);
for ($tupla = 0; $tupla < $tot_tuplas2; $tupla ++)
{
$classe=pg_result ($result2, $tupla, 0);
$cond=pg_result ($result2, $tupla, 1);
$limite=pg_result ($result2, $tupla, 2);
$aviso=pg_result ($result2, $tupla, 3);
$situ_med='0'; $situ_max='0'; $situ_min='0';
//Dependendo da condicao o percentual do aviso cresce ou decresce

```

```

if (preg_match("/</", $cond)) {
    $prod_av=$aviso/100+1;} else {$prod_av=-$aviso/100+1;};
// Considera a estacao sem alerta e avalia conforme expressoes abaixo:
$expr_med="if(".$media.$cond.$limite.") {\$situ_med = '2';} elseif(".$media.
    $cond."(".$limite.*".$prod_av.")) {\$situ_med = '1';}";
$expr_max="if(".$max.$cond.$limite.") {\$situ_max = '2';} elseif(".$max.
    $cond."(".$limite.*".$prod_av.")) {\$situ_max = '1';}";
$expr_min="if(".$min.$cond.$limite.") {\$situ_min = '2';} elseif(".$min.
    $cond."(".$limite.*".$prod_av.")) {\$situ_min = '1';}";
eval($expr_med); eval($expr_max); eval($expr_min);
$result5 = pg_query ($conex1, "insert into alerta values ($cod_estac, ".
    "$param, $classe, $mes, $ano, $media, $max, $min, ".
    "$socor, $situ_med, $situ_max, $situ_min);");
}
return $cont+$tot_tuplas2;
}
//-----
// --- Inicio de codigo: - Programa principal em PHP incorporado ao codigo HTML
?>
<html>
<head><TITLE>Carga de arquivos da ANA em PHP via PostgreSQL para o
SAQUA.</TITLE>
</head><body>Programa para carregar dados da ANA no SAQUA => carga.php<br>
Obs: Ao carregar sem o codigo da estacao (cod_estac) busca todas estacoes.<p>
<?php
// O cabeçalho com metadados do arquivo texto na linha 18 e dados apos linha 19
// -----
// Abrindo banco de dados - Buscando campos de consultas:
$cont=0; // Inicializa contador de registros incluídos
$conex1 = pg_connect ("host = localhost
port = 5432
dbname = saqua
user = saqua
password = mestrado")
or die ("Falha na conexão!".pg_last_error());
$result2 = pg_query ($conex1, "select cod_param from param");
if ( $_GET['cod_estac'] ){
    $result1 = pg_query ($conex1, "select * from estacao where cod_estac=" .
        $_GET['cod_estac']);
    $tot_tuplas = pg_numrows ($result1);
    // Apaga todos os alertas anteriores da estacao no banco, pois criara novos
    $result3 = pg_query ($conex1, "delete from alerta where cod_estac=" .
        $_GET['cod_estac']);}
else {
    $result1 = pg_query ($conex1, "select * from estacao");
    $result3 = pg_query ($conex1, "delete from alerta ");
    $tot_tuplas = pg_numrows ($result1);}
echo "EXCLUÍDO OS DADOS ANTERIORES do arquivo de alerta referentes a carga, ".
    "no total de: ".pg_affected_rows($result3)." REGISTROS<br>";

```

```

echo "Rela&ccedil;&atilde;o de arquivos a serem carregados:<br>";
for ($stupla = 0; $stupla < $tot_tuplas; $stupla ++){
    $arquivo[$stupla+1] = pg_result ($result1, $stupla, 2);
    echo ($stupla+1)." - Arquivo: ".$arquivo[$stupla+1]."<br>";
}
$narquivos=count($arquivo);
$tot_tuplas = pg_numrows ($result2);
echo "Rela&ccedil;&atilde;o de c&oacute;digos de par&acirc;metros a serem "
    "carregados:<br>";
for ($stupla = 0; $stupla < $tot_tuplas; $stupla ++){
    $pegacampo[$stupla+1] = pg_result ($result2, $stupla, 0);
    echo ($stupla+1)." - c&oacute;digo de par&acirc;metro: ".$pegacampo[$stupla+1].
        "<br>";
}
$npegas=count($pegacampo);
for( $narq=1 ; $narq<=$narquivos ; $narq ++) {
    //+ Estacao(oes) selecionada(as)
    echo "<p> ARQUIVO: ".$arquivo[$narq]."( ".$narq." de ".$narquivos." )<br>";
    // Cria outras cadeias (arrays) paralelas para acompanhamento a cada arquivo
    for( $ncampo=1 ; $ncampo<=$npegas ; $ncampo ++) {
        $pegamedant[$ncampo]=0;
        $pegamaxant[$ncampo]=0;
        $pegaminant[$ncampo]=0;
        $pegamesant[$ncampo]=0;
        $pegaanoant[$ncampo]=0;
        $peganmeses[$ncampo]=0;
        $peganocorr[$ncampo]=0; // Ocorrencia ou n para a media
    }
    $estacao=0;
    $abraArq = fopen("$arquivo[$narq]", "r");
    if (!$abraArq){
        echo ("<p><b>Arquivo n&atilde;o encontrado: ".$arquivo[$narq]."</b></p>");
    }else{
        $lin=0;
        $mens=0;
        $mesant=0;
        $anoant=0;
        while ($campo = fgetcsv ($abraArq, 2048, ";")) {
            $lin=$lin+1; // limite abaixo ate a linha 33 para exibicao em tela
            if ($lin>20) {
                $estacao=$campo[0];
                // -- Selecao do criterio de Inclusao
                if ($lin==21) { //Apenas na primeira vez
                    $mesclada=false; //- $continua=false;
                    // Testa, se nao for continuidade o codigo da estacao sera diferente
                    $result3 = pg_query ($conex1, "SELECT cod_estac from estacao WHERE "
                        "arq_orig = '$arquivo[$narq]'");
                    $tot_tuplas = pg_numrows ($result1);
                    if ($tot_tuplas=0){

```

```

// Em codigo nao cadastrado preenche o encontrado
$result4 = pg_query ($conex1, "UPDATE estacao SET cod_estac = ".
    "$stacao WHERE arq_orig = '$arquivo[$narq]'");
} else { // O estado de mesclado e identificado apos a carga que continua
    $cod_est=pg_result ($result3, 0, 0);
    if ( $stacao==$cod_est ) { //Por convencao, e a estacao atual,
        if ($continua) $mesclada=true; $continua=false;//sem mais continuidade
    } else {
        echo "<br>Possui continuidade ( $stacao <> $cod_est )<br>";
        if ($continua) $mesclada=true;$continua=true;//deve ter mais parte
    } } if(!$cod_est){$cod_est = $stac;}
$mes=Substr($campo[2],3,2); $ano=Substr($campo[2],6,4);
echo "<p>-> =" . $campo[2]. " = " . Substr($campo[3],11,8). " = " . $stacao;
$pega=0;
if ($mesclada) {
    $mesclada=false;
    for( $ncampo=1 ; $ncampo<=$npegas ; $ncampo ++ ) {
        $result4= pg_query ($conex1, "select mes,ano,media,maximo,minimo,".
            "ocorre from alerta where cod_estac=$stacao and cod_param=" .
            "$pegacampo[$ncampo] and ocorre>0 and ano*100+mes < $ano*100+" .
            "$mes order by ano*100+mes desc limit 1");
        $sha_dado = pg_numrows ($result4);
        if ($sha_dado){
            for( $ncbusca=0 ; $ncbusca<6 ; $ncbusca ++ ){ // Valor anterior
                $pegamedant[$ncampo]=pg_result($result4,0,2);
                $pegamaxant[$ncampo]=pg_result($result4,0,3);
                $pegaminant[$ncampo]=pg_result($result4,0,4);
                $pegamesant[$ncampo]=pg_result($result4,0,0);
                $pegaanoant[$ncampo]=pg_result($result4,0,1);
                $peganocorr[$ncampo]=pg_result($result4,0,5); }
            $result3= pg_query ($conex1, "delete from alerta where ".
                "cod_estac=" . $stacao. " and cod_param=" . $pegacampo[$ncampo].
                " and ano>=$pegaanoant[$ncampo] and mes>=$pegamesant[$ncampo]");
            $apagado=pg_affected_rows($result3);
            $cont=$cont-$apagado; $mens=$mens+1;
            $mensagem[$mens]="MESCLADO o parametro ($pegacampo[$ncampo]).
                ") pelos valores de m&eacute;d./m&aacute;x./min./ocorr. ".
                "($pegamedant[$ncampo]/$pegamaxant[$ncampo]/$pegaminant[$ncampo].".
                "$peganocorr[$ncampo]) de $pegamesant[$ncampo]/$pegaanoant[$ncampo], ".
                "sendo $apagado registros removidos<br>";
        } else{
            $result3= pg_query ($conex1, "delete from alerta where ".
                "cod_estac=" . $stacao. " and cod_param=" . $pegacampo[$ncampo].
                " and ano>=$ano and mes>=$mes");
            $apagado=pg_affected_rows($result3);
            $cont=$cont-$apagado; $mens=$mens+1;
            $mensagem[$mens]="REMOVER $apagado registros do parametro
                ($pegacampo[$ncampo]).
                ") por nao dispor de dados mesclados antes de $mes/$ano<br>";
        }
    }
}

```



```

    }elseif ($campo[$pegacampo[$pega]]) { $pegamesant[$pega]=$mes;
      $pegaanoant[$pega]=$ano;
      $valcampo=Str_replace(",",".",$campo[$pegacampo[$pega]]);
      $peganocorr[$pega]=1; $pegamedant[$pega]=$valcampo;
      $pegaminant[$pega]=$valcampo;$pegamaxant[$pega]=$valcampo;}
    echo "=". $campo[$pegacampo[$pega]].".".$pegacampo[$pega]."); ";
    if ($pega=$npegas) { echo "<br>";for( $msg=1; $msg <= $mens ; $msg ++ ) {
      echo $mensagem[$msg];}; $mens=0;}
    } elseif ( $lin==18 ) {
      echo "<br> -> Dados Exibidos [=Parametros(codpar)]: =Data =Hora ".
        "=EstacaoCodigo <br> "; $pega=0;
      while ($pega<$npegas) { $pega=$pega+1;
        echo "=". $campo[$pegacampo[$pega]].".".$pegacampo[$pega].");";
      };
    }
  }
// Fim do Arquivo - inclusao de ultimos campos
echo "<br> Finalizado o arquivo (".$arquivo[$narq]."). Terminando esta carga:";
for( $pega=1 ; $pega<=$npegas ; $pega ++ ) {
  if ( $pegaanoant[$pega] ) {
    echo "<br> INCLUIR no alerta do par&acirc;metro (".$pegacampo[$pega].
      " os valores m&eacute;d./m&aacute;x./min./ocor. de ".
      $pegamedant[$pega]."/".$pegamaxant[$pega]."/".$pegaminant[$pega]."/".
      $peganocorr[$pega]." , ref. mes/ano ".$pegamesant[$pega]."/".
      $pegaanoant[$pega];
    $cont=incluir($cont,$cod_est,$pegacampo[$pega],$pegamedant[$pega],
      $pegamaxant[$pega],$pegaminant[$pega],$peganocorr[$pega],
      $pegamesant[$pega],$pegaanoant[$pega]);
    echo "- (sub-t.".$cont.)<br>";    }
  };
  fclose($abraArq);
} // --< fim do senao houve falha de abertura de arquivo
} // ---< fim do loop de arquivos
pg_close ($conex1);
echo "<p> TOTAL DE INCLUSOES:". $cont;?>
</body></html>

```

Código do arquivo **google_estacoes.html** na sub-pasta *saqua*

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<!-- Programacao em HTML, com uso de JavaScript e API's do Google Map-->
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
  <title>Google Maps JavaScript API Example</title>
  <script src="http://maps.google.com/maps?file=api&v=2.x&key=
ABQIAAAA9CA42OhiF49HvMeyC8P4ZhQFYHsB1A9lVTeRSIrdGqDHRRWmCRRM
  bvDh6Y9yImYo-G21cpS9Vw_YA"

```



```

<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
<title>Localizacao por coordenadas das Estacoes de Qualidade de Agua pelo
  Google Maps</title>
</head>
<body>
<?php
// O codigo (cod) corresponde ao numero de registro menos 1
if ( $_GET['cod'] ) { if ( $_GET['cod'] == 66 )
  echo '<a href="/google_estacoes.php?lat=-21.389444&long=-42.696389&fa=17'.
    '&cod='.$_GET['cod'].'"><br></a>';
  elseif ( $_GET['cod'] == 67 )
    echo '<a href="/google_estacoes.php?lat=-21.542222&long=-42.180556&fa=14'.
      '&cod='.$_GET['cod'].'"><br></a>';
  elseif ( $_GET['cod'] == 57 )
    echo '<a href="/google_estacoes.php?lat=-23.368889&long=-45.900000&fa=18'.
      '&cod='.$_GET['cod'].'"><br></a>';
  elseif ( $_GET['cod'] == 58 )
    echo '<a href="/google_estacoes.php?lat=-22.911111&long=-45.470278&fa=18'.
      '&cod='.$_GET['cod'].'"><br></a>';
  elseif ( $_GET['cod'] == 59 )
    echo '<a href="/google_estacoes.php?lat=-22.540000&long=-44.772778&fa=18'.
      '&cod='.$_GET['cod'].'"><br></a>';
  elseif ( $_GET['cod'] == 60 )
    echo '<a href="/google_estacoes.php?lat=-22.466667&long=-44.445272&fa=18'.
      '&cod='.$_GET['cod'].'"><br></a>';
  elseif ( $_GET['cod'] == 62 )
    echo '<a href="/google_estacoes.php?lat=-22.449444&long=-43.798889&fa=17'.
      '&cod='.$_GET['cod'].'"><br></a>';
  elseif ( $_GET['cod'] == 65 )
    echo '<a href="/google_estacoes.php?lat=-22.035278&long=-42.990833&fa=14'.
      '&cod='.$_GET['cod'].'"><br></a>';
  elseif ( $_GET['cod'] == 64 )
    echo '<a href="/google_estacoes.php?lat=-21.778333&long=-43.325278&fa=18'.
      '&cod='.$_GET['cod'].'"><br></a>';
  elseif ( $_GET['cod'] == 68 )
    echo '<a href="/google_estacoes.php?lat=-21.645278&long=-41.752222&fa=14'.
      '&cod='.$_GET['cod'].'"><br></a>';
  elseif ( $_GET['cod'] == 69 )
    echo '<a href="/google_estacoes.php?lat=-21.487222&long=-41.616667&fa=14'.

```

