



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE TECNOLOGIA E RECURSOS NATURAIS
UNIDADE ACADÊMICA DE ENGENHARIA CIVIL
PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL E AMBIENTAL
ÁREA DE CONCENTRAÇÃO: ENGENHARIA DE RECURSOS HÍDRICOS E SANITÁRIA

**ARQUITETURA GENÉRICA PARA UM SISTEMA DE SUPORTE À
DECISÃO EM RECURSOS HÍDRICOS**

ARTUR MOISES GONÇALVES LOURENÇO

CAMPINA GRANDE - PB

2015

ARTUR MOISES GONÇALVES LOURENÇO

**ARQUITETURA GENÉRICA PARA UM SISTEMA DE SUPORTE À
DECISÃO EM RECURSOS HÍDRICOS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil e Ambiental da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Mestre em Engenharia Civil e Ambiental.

Orientador: Prof.º PhD Wilson Fadlo Curi

CAMPINA GRANDE - PB

2015

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

- L892a Lourenço, Artur Moises Gonçalves.
Arquitetura genérica para um sistema de suporte à decisão em recursos hídricos / Artur Moises Gonçalves Lourenço. – Campina Grande, 2015.
157 f. : il. color.
- Dissertação (Mestrado em Engenharia Civil e Ambiental) – Universidade Federal de Campina Grande, Centro de Tecnologia e Recursos Naturais, 2015.
- "Orientação: Prof.º PhD.Wilson Fadlo Curi".
Referências.
1. Sistema de Informação. 2. Banco de Dados. 3. Metrologia.
4. Modelagem. I. Curi, Wilson Fadlo. II. Título.

CDU 624:004.65(043)

ARTUR MOISES GONÇALVES LOURENÇO

ARQUITETURA GENÉRICA PARA UM SISTEMA DE SUPORTE À DECISÃO EM RECURSOS HÍDRICOS

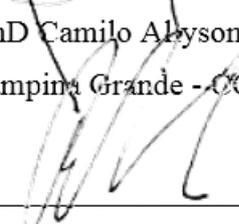
Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil e Ambiental da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Mestre em Engenharia Civil e Ambiental.

Aprovado em 13/04/2015

BANCA EXAMINADORA:


Orientador - Prof.º PhD Wilson Fadlo Curi
(Universidade Federal de Campina Grande - CT - UAF)


Examinador Interno - Prof.º PhD Camilo Alyson Simões de Farias
(Universidade Federal de Campina Grande - OCTA - UACTA)


Examinador Externo - Prof.º PhD José Roberto Gonçalves de Azevedo
(Universidade Federal de Pernambuco - CTG - DEC)

Dedicatória

A todos que considero minha família.

Agradecimentos

À algo supremo, criador da vida, que não poderá ser explicado ou negado seja quaisquer o nível de conhecimento que o homem um dia virá a ter;

À minha família por me apoiar e possibilitar mais um passo neste caminho tortuoso que é ser pesquisador e professor no Brasil;

À minha amada Raphaele pelo suporte e por suportar minhas ausências;

Aos meus colegas de mestrado em especial a turma da Área de Engenharia de Recursos Hídricos pelo convívio e amizades durante essa trajetória;

Ao Prof.º Wilson, meu orientador, que me fez sair da zona de conforto, fazendo-me superar vários limites, possibilitando um real ganho de conhecimento no mestrado e maturidade para a vida acadêmica;

Aos Professores do programa de pós-graduação o qual esta dissertação está inserida, pela transmissão dos conhecimentos e ensinamentos;

Aos servidores lotados no Laboratório de Hidráulica I, nas pessoas de Aurezinha, Ismael, Vera e Júlio, pela eficiência, atenção e amizade com todos os alunos;

À Universidade Federal de Campina Grande pela oportunidade;

Ao CNPq pela concessão da bolsa de estudos;

À todas outras pessoas não citadas que contribuíram de alguma forma para que eu chegasse até aqui.

Muito Obrigado!

*“Pouco conhecimento faz com que as pessoas se sintam orgulhosas.
Muito conhecimento, que se sintam humildes.”*

Leonardo da Vinci

Resumo

O objetivo principal deste trabalho é propor uma arquitetura para um Sistema de Suporte a Decisão-SSD em recursos hídricos genérico, aberto, expansível e adaptativo às características da unidade de planejamento bacia hidrográfica, e às necessidades atuais de seus planejadores. Utilizou-se a abordagem orientada a objetos para a modelagem dos componentes da bacia hidrográfica. A arquitetura do sistema definido foi a cliente-servidor multicamadas, que permite realizar processos separados que podem ser executados em diferentes processadores. Para o subsistema SIG (Sistema de Informações Geográficas) foi utilizado o *framework* SIG baseado na web, o i3Geo. As variáveis que podem ser armazenadas no banco de dados foram estruturadas utilizando-se de conceitos de metrologia, o que permite uma melhor caracterização das mesmas e proporciona sua adaptabilidade aos requerimentos do sistema. O modelo SMAPModel, desenvolvido em ambiente R, foi integrado ao subsistema de modelos do SSD como forma de mostrar deste ter uma boa interface com outros modelos e sistemas. Também foram desenvolvidos outros aplicativos de acesso a dados externos ao sistema. O sistema se mostrou estável, apresentando interface amigável para o usuário e fluidez na integração de seus componentes. Espera-se que o sistema proposto possa contribuir nos processos decisórios podendo evoluir e ser customizado conforme a necessidade dos usuários e instituições, como em agências de recursos hídricos.

Palavras-chave: Sistemas de Informação, Banco de Dados, Metrologia, Modelagem, SIG.

Abstract

The main objective of this work is to propose a generic architecture and structure for a water resources Decision Support System-DSS, which is open, scalable and adaptive for the planning of a watershed unit, with its characteristics and the current needs of its planners. An object-oriented approach for modeling the components of the watershed was used. The system's architecture was defined as a multi-client server to perform separate processes that can run on different processors. For the subsystem GIS was used a GIS framework based on the web i3Geo software. Variables that can be stored in the developed database were structured using metrology concepts, which allows its better characterization as well as provides their adaptability to the systems' requirements. The SMAPModel, which was developed in R environment, was integrated to the SSD's models subsystem to show its capacity of having good interface with other models and systems. Other applications have also been developed to access external data to the system. The system proved to be stable and to have friendly user interface and fluidity in the integration of its components. It is expected that the proposed system can support decision-making processes and evolve to be customized according to the needs of users and institutions, such as water resources state agencies.

Key-words: Information Systems, Database, Metrology, Modeling, GIS.

Lista de Abreviaturas e Siglas

AESA	Agência Executiva de Gestão das Águas do Estado da Paraíba.
ANA	Agência Nacional da Águas.
API	<i>Application Programming Interface.</i>
CAD	<i>Computer Aided Design.</i>
CASE	<i>Computer Aided Software Engineering.</i>
CISDERGO	<i>Cropping and Irrigation System Design with Reservoir and Groundwater Operation.</i>
CRUD	<i>Create, Read, Update e Delete.</i>
DAO	<i>Data Access Object.</i>
DBA	<i>Database Administrator.</i>
DBMS	<i>Database Management System.</i>
DBO	Demanda Bioquímica de Oxigênio.
DIKW	<i>Data- Information- Knowledge- Wisdom.</i>
EER	<i>Enhanced Entity- Relationship.</i>
GIRH	Gestão Integrada de Recursos Hídricos
GIS	<i>Geographic Information System.</i>
GOTA	Grupo de Otimização Total da Água.
GPL	<i>General Public Licence.</i>
GUI	<i>Graphic User Interface.</i>
GWP	<i>Global Water Partnership.</i>
HTML	<i>Hyper Text Markup Language.</i>
i3Geo	Interface Integrada para Internet de Ferramentas de Geoprocessamento.
IDE	<i>Integrated Development Environment.</i>
INDE	Infraestrutura Nacional de Dados Espaciais.
IWRM	<i>Integrated Water Resources Management.</i>
JSON	<i>JavaScript Object Notation.</i>
KML	<i>Keyhole Markup Language.</i>
LAN	<i>Local Area Network.</i>
MMA	Ministério do Meio Ambiente.
MP	Ministério do Planejamento.
MVC	<i>Model-View-Controller.</i>
ODBC	<i>Open Database Connectivity.</i>
OGC	<i>Open Geospatial Consortium.</i>
OGR/GDAL	<i>OpenGIS Simple Features Reference/ Geospatial Data Abstraction Library.</i>
OOP	<i>Object Oriented Programming.</i>
ORM	<i>Object Relational Mapping.</i>
ORNAP	<i>Optimal Reservoir Network Analysis Program.</i>
OSGeo	<i>Open Source Geospatial Foundation.</i>
PMA	Parceria Mundial da Água.
PNRH	Política Nacional de Recursos Hídricos.
POO	Programação Orientada a Objetos.

PSO	<i>Particle Swarm Optimization.</i>
PSPB	Portal do Software Público Brasileiro.
REST	<i>Representational State Transfer.</i>
RIOSS	<i>Riverbasin Information and Operation Support System.</i>
RTTI	<i>Run- Time Type Information.</i>
SA	Sistema de Análise.
SGBD	Sistema de Gerenciamento de Banco de Dados.
SI	Sistema de Informação.
SI ²	Sistema Internacional de Medidas.
SIGs	Sistemas de Informação Geográfica.
SINGREH	Sistema Nacional de Gerenciamento de Recursos Hídricos.
SIRH	Sistemas de Informações sobre Recursos Hídricos.
SMAP	<i>Soil Moisture Accounting Procedure.</i>
SNIRH	Sistema Nacional de Informações sobre os Recursos Hídrico.
SOAP	<i>Simple Object Access Protocol.</i>
SQL	<i>Structured Query Language.</i>
SSD	Sistema de Suporte a Decisão.
TGS	Teoria Geral dos Sistemas.
TI	Tecnologia da Informação.
UML	<i>Unified Modeling Language.</i>
UTM	<i>Universal Transversa de Mercator.</i>
W3C	<i>World Wide Web Consortium.</i>
WAN	<i>Wide Area Network.</i>
WSDL	<i>Web Services Description Language.</i>
WWW	<i>World Wide Web.</i>
XML	<i>Extensible Markup Language.</i>

Lista de Tabelas

Tabela 2.1 - Principais conceitos da modelagem orientada a objetos.	37
Tabela 2.2 - Softwares para aplicações em recursos hídricos.	50

Lista de Figuras

Figura 2.1 - Hierarquia DIKW.	23
Figura 2.2 - Atividades de um Sistema.	25
Figura 2.3 - Interações entre os subsistemas e entre eles e seus ambientes.	27
Figura 2.4 - Diagramas ilustrando os modelos de dados.	30
Figura 2.5 - Diagrama de um Sistema de Banco de Dados.	32
Figura 2.6 - Arquitetura cliente-servidor em duas camadas.	34
Figura 2.7 - Arquitetura cliente-servidor três camadas.	35
Figura 2.8 - Exemplo de um Diagrama de Classe.	39
Figura 2.9 - Componente de um Sistema de Informações.	41
Figura 2.10 - Modelos geométricos matricial e vetorial.	43
Figura 2.11 - Arquitetura de um SIG Relacional.	45
Figura 2.12 - Componentes de um SSD.	48
Figura 3.1 - Fluxo de processamento de operações do SSD.	54
Figura 3.2 - Arquitetura do SSD.	56
Figura 3.3 - Etapas no processo de prototipagem.	58
Figura 3.4 - IDE do MySQL Workbench.	59
Figura 3.5 - Estrutura do banco de dados e componentes do SSD.	62
Figura 3.6 - Modelagem do banco de dados utilizando diagramas EER no MySQL Workbench.	63
Figura 3.7 - Diagramas EER do grupo de tabelas referentes a autenticação e <i>logger</i>	64
Figura 3.8 - Diagramas EER do grupo de tabelas referentes a projetos e cenários.	65
Figura 3.9 - Diagramas EER do grupo de tabelas referentes aos clientes, instituições e documentos.	66
Figura 3.10 - Diagramas EER do grupo de tabelas referentes as localidades.	67
Figura 3.11 - Estrutura de armazenamento de variáveis SSD seguindo conceitos de metrologia.	69
Figura 3.12 - Diagramas EER do grupo de tabelas referentes a unidade de medida e grandeza.	70
Figura 3.13 - Diagramas EER do grupo de tabelas referentes as caracterização das variáveis.	71
Figura 3.14 - Diagramas EER do grupo de tabelas referentes ao sistema de medição.	72
Figura 3.15 - Diagramas EER do grupo de tabelas referentes a bacias hidrográfica e rio.	73

Figura 3.16 - Conjunto de tabelas responsáveis por armazenar atributos e geometria de dados espaciais.....	75
Figura 3.17 - Arquitetura básica de aplicativos MapServer.	78
Figura 3.18 - Arquitetura do i3Geo.	80
Figura 3.19 - Interface básica do i3Geo.	81
Figura 3.20 - Principais <i>units</i> do servidor de aplicação.....	83
Figura 3.21 - Diagrama de classe da classe Ttbbh.	84
Figura 3.22 - Diagrama de classe da classe TGenericDAO.	85
Figura 3.23 - Principais <i>units</i> do lado cliente.	86
Figura 3.24 - Principais <i>units</i> que constroem a interface com o usuário.....	87
Figura 3.25 - Interface do <i>form</i> genérico de cadastro.....	88
Figura 3.26 - Interface web para o RIOSS.	89
Figura 3.27 - Etapas do processamento e modelagem do SMAPmodel.....	92
Figura 3.28 - Diagrama de classe da classe TSMAPModel.	93
Figura 3.29 - Diagramas EER do grupo de tabelas referente ao SMAPModel.	93
Figura 3.30 - <i>Units</i> que compõe o ANA App.....	95
Figura 3.31 - Diagrama de classe das classes presentes na <i>unit</i> uMMA.	96
Figura 3.32 - Diagrama de classe da classe TFrmAESADatImport.	97
Figura 4.1 - Interface inicial do SIG.....	98
Figura 4.2 - Barra de botões e menu ferramentas do SIG.	100
Figura 4.3 - Menu Aplicativos do SIG.	101
Figura 4.4 - Menu arquivo do SIG.	102
Figura 4.5 - Menu análise do SIG.	103
Figura 4.6 - Cálculo de geometrias.....	104
Figura 4.7 - Geração de gráficos.	105
Figura 4.8 - Guia sanfona do SIG.....	105
Figura 4.9 - Interface do servidor de aplicações.....	107
Figura 4.10 - Servidor de aplicações em execução.	108
Figura 4.11 - Tela de abertura do SSD.	109
Figura 4.12 - Tela de login e recuperação de senha do SSD.....	109
Figura 4.13 - Tela principal do SSD.....	110
Figura 4.14 - Menu Sistema.	111
Figura 4.15 - Menu Arquivo.....	112

Figura 4.16 - Criando novo projeto.	113
Figura 4.17 – Salvando projeto.....	114
Figura 4.18 - Item Estados no menu localidades.....	115
Figura 4.19 - Item Tipos de Locais no menu localidades.....	116
Figura 4.20 - Item Locais no menu localidades.....	117
Figura 4.21 - Tela de definição da localização espacial.	118
Figura 4.22 - Item Instituições no menu institucional.	119
Figura 4.23 - Item Clientes no menu institucional.	120
Figura 4.24 - Item Usuários no menu institucional.	121
Figura 4.25 - Item Grandezas no menu sistema de metrologia.	122
Figura 4.26 - Item Unidades de Medida no menu sistema de metrologia.	123
Figura 4.27 - Item Grupo de Variáveis no menu sistema de metrologia.....	124
Figura 4.28 - Item Variáveis no menu sistema de metrologia.....	125
Figura 4.29 - Cadastro Função Utilidade no item variáveis.....	126
Figura 4.30 - Cadastro Classe de Variáveis no item variáveis.	127
Figura 4.31 - Cadastro Valores de Classe.....	128
Figura 4.32 - Cadastro Valores de Classe com dois intervalos.	129
Figura 4.33 - Item Estações de Medição no menu sistema de medições.....	130
Figura 4.34 - Item Variáveis Medidas no menu Estações de Medição.	131
Figura 4.35 - Cadastro Valores de Variáveis.....	132
Figura 4.36 - Tela para escolha dos valores de classe.....	133
Figura 4.37 - Valores de classe para série escolhida.	134
Figura 4.38 - Exemplo de série de dados de DBO cadastrada.	135
Figura 4.39 - Utilização da ferramenta Função Utilidade.	136
Figura 4.40 - Item Grupo de Atividades no menu atividades.....	137
Figura 4.41 - Item Atividade no menu atividades.	138
Figura 4.42 - SIG em funcionamento no SSD.....	139
Figura 4.43 - Interface do aplicativo de acesso ao banco dados da ANA.	140
Figura 4.44 - Consulta no aplicativo.	141
Figura 4.45 - Interface do aplicativo de acesso aos dados da AESA.	142
Figura 4.46 - Análise e importação dos dados.....	143
Figura 4.47 - Interface criada para o SMAPModel.	144
Figura 4.48 - Simulação no SMAPModel.	145

Figura 4.49 - Proposta de interface para o modelo ORNAP. 147

Sumário

<i>Capítulo 1</i>	17
1 INTRODUÇÃO	17
1.1 Objetivo Geral	20
1.2 Objetivos Específicos	20
1.3 Justificativa.....	20
<i>Capítulo 2</i>	22
2 REVISÃO BIBLIOGRÁFICA	22
2.1. De Dado à Sabedoria	22
2.2. Sistemas e seus Componentes	23
2.3. Modelagem de Sistemas	25
2.4. Análise e Modelagem de Sistemas Hídricos	25
2.5. Sistemas de Banco de Dados	29
2.5.1 Modelos de Dados	29
2.5.2. Sistemas de Gerenciamento de Banco de Dados.....	30
2.5.3. Arquitetura para os SGDBs	32
2.6. Sistemas Orientados a Objetos	35
2.7. Sistema de Informações.....	40
2.8 Sistema de Informações Geográficas	41
2.9 Sistema de Suporte à Decisão.....	46
2.9.1 Aplicações em Recursos Hídricos	47
<i>Capítulo 3</i>	52
3 METODOLOGIA	52
3.1 Arquitetura do Sistema	52
3.2 Processo de Desenvolvimento do Sistema	57
3.2 Sistema de Gerenciamento para o Banco de Dados	58
3.3 Modelagem do SSD.....	60

3.3.1 Técnicas de Programação	60
3.3.2 Modelagem do Banco de Dados	61
3.3.3 Visualização de Dados Espaciais	75
3.3.4 Desenvolvimento Lado Servidor	81
3.3.5 Desenvolvimento Lado Cliente Desktop	86
3.3.6 Lado Cliente Web	88
3.3.6 Sistema de Modelos	89
3.3.7 Acesso a Dados Externos	94
<i>Capítulo 4</i>	98
4 RESULTADOS E DISCUSSÃO	98
4.1 Sistema de Informações Geográficas do SSD	98
4.2 Servidor de Aplicações	106
4.3 Cliente Desktop	108
4.3.1 Localidades	114
4.3.2 Institucional	118
4.3.3 Sistema de Metrologia	121
4.3.4 Sistema de Medições	129
4.3.5 Atividades	137
4.3.6 SIG	138
4.4 Aplicativos de Acesso a Dados Externos	139
4.5 Sistema de Modelos	143
<i>Capítulo 5</i>	148
5 CONCLUSÕES	148
6 Recomendações para Trabalhos Futuros	151
Referências Bibliográficas	152

Capítulo 1

1 Introdução

O aumento populacional e a ineficácia ou ausência da gestão e planejamento do uso dos diversos recursos naturais disponíveis no planeta tem configurado a crise mundial atual, distanciando cada vez mais o futuro da humanidade dos ideais de desenvolvimento discutidos nas assembleias realizadas sobre meio ambiente e desenvolvimento sustentável.

A recessão econômica, crise alimentar, fragilidade da matriz energética, aumento da pobreza, guerras e fenômenos ambientais extremos são reflexos do modelo de desenvolvimento atual. Contudo, paralelamente o desenvolvimento exponencial da tecnologia surge como ferramenta aliada para tentativa da mitigação destes efeitos e suporte na gestão dos recursos naturais.

Não diferente dos demais recursos naturais, a água, a qual é essencial para a vida e desenvolvimento das diversas atividades produtivas, tem sofrido continua pressão pelo homem fazendo com que sua gestão e gerenciamento necessitem cada vez mais de ferramentas de suporte nas tomadas de decisão, abrangendo os diversos componentes que constituem um sistema de recursos hídricos.

Os sistemas de recursos hídricos tratam, primordialmente, de aspectos qualitativos da água em termos de compatibilização entre oferta e demanda de seus múltiplos usos e os impactos nas dimensões socioeconômicas, ambientais e técnico-operacionais. São sistemas complexos e mutáveis onde seu planejamento envolve questões que vão desde a governança até as intervenções físicas, portanto, uma visão holística do sistema é necessária para a adequação de cada possível intervenção.

No Brasil, a Lei 9.433/97 instituiu a Política Nacional de Recursos Hídricos (PNRH) e cria o Sistema Nacional de Gerenciamento de Recursos Hídricos (SINGREH). As novas diretrizes estabelecidas pela lei trazem inovações e avanços no tocante da gestão dos recursos hídricos, em especial a definição da bacia hidrográfica como unidade de planejamento e a gestão participativa, esta última incorporando o controle social na gestão das águas.

Também foram estabelecidos na política nacional instrumentos para a sua aplicação a cobrança e outorga da água, o enquadramento dos recursos hídricos e a criação dos sistemas informações e planos de recursos hídricos.

Em especial, os Sistemas de Informações sobre Recursos Hídricos (SIRH), segundo a PNRH, tem o objetivo de coletar, tratar, armazenar e recuperar informações intervenientes da gestão das águas.

Este instrumento tem papel crucial na integração e cruzamento dos dados obtidos durante a implementação dos demais instrumentos. Além de dar suporte para o desenvolvimento dos planos, os sistemas de informação devem possibilitar o processamento de dados brutos transformando em informações para criação métricas de avaliação da gestão que está sendo implementada na bacia. Dessa forma, além do mesmo ser um instrumento de coleta, armazenamento e disseminação de dados, também deve-se considerá-lo como forte instrumento de suporte nos processos decisórios em sistemas de recursos hídricos.

A Agência Nacional das Águas (ANA) tem conduzido as atividades de desenvolvimento e suporte do Sistema de Informações Hidrológicas Nacional, o HIDRO. O sistema tem o objetivo de centralizar séries históricas armazenadas por diversas instituições, e não considera o desenvolvimento de uma nova arquitetura e estrutura de armazenamento que possa atender as necessidades atuais do planejamento e gerenciamento dos recursos hídricos.

Além disso, a estrutura criada pela ANA tem como alvo o Sistema Nacional de Informações sobre os Recursos Hídrico (SNIRH), não possuindo assim escala adequada para que seja utilizada em nível estadual, a exemplo de, atualmente, existir uma grande demanda para dar apoio ao desenvolvimento e implementação dos planos diretores das bacias hidrográficas inseridas nos estados. Pode-se ressaltar, também, que cada sistema hídrico possui uma estrutura própria e características peculiares distintas em termos de aspectos socioeconômicos e ambientais, assim como a implantação de novas intervenções hídricas, mudança de regras de operação ou de usos da água afetam estes aspectos.

Dentro deste contexto, há uma necessidade de incorporar estes dados, dentro de um sistema apropriado, dinâmico, para que possam ser usados para estabelecer ou gerar índices e indicadores socioeconômicos, ambientais e operacionais, obtidos via modelos que venham atuar sobre os dados disponíveis, gerando relatórios e formas de apresentação apropriadas constituindo assim um sistema mais abrangente, de suporte à decisão.

Diante do exposto, este trabalho tem como objetivo propor uma arquitetura genérica, bem como desenvolvimento e implementação, de um Sistema de Suporte à Decisão (SSD) para sistemas de recursos hídricos, o qual seja aberto, expansível e adaptativo as características da unidade de planejamento, a bacia hidrográfica, e requerimentos de seus planejadores.

1.1 Objetivo Geral

Propor uma arquitetura genérica, bem como desenvolvimento e implementação, de um sistema de suporte à decisão em recursos hídricos o qual seja aberto, expansível e adaptativo às características da unidade de planejamento, a bacia hidrográfica, e possa atender os requerimentos de seus planejadores.

1.2 Objetivos Específicos

1. Definir uma arquitetura para o SSD;
2. Definir tecnologias a serem utilizadas no sistema (tipo de banco de dados, linguagens de programação, visualização espacial, etc.);
3. Implementar e adaptar, quando necessário, a estrutura para o banco de dados existente que melhor permita modelar e caracterizar os sistemas de recursos hídricos;
4. Definir um Sistema de Informações Geográficas para visualização e análise dos dados espaciais;
5. Desenvolver interface do sistema, que seja amigável para o usuário;
6. Permitir ao sistema acesso aos bancos de dados externos sobre recursos hídricos;
7. Verificar a integração do sistema com outras ferramentas utilizadas na área de recursos hídricos;
8. Validar o sistema por meio de uma aplicação.

1.3 Justificativa

Esta dissertação está inserida na linha de pesquisa Gestão de Recursos Hídricos deste Programa de Pós-Graduação na área de Engenharia de Recursos Hídricos e Sanitária. Esta dissertação se insere no contexto da necessidade de desenvolvimento de sistemas computacionais de suporte à decisão em recursos hídricos, que sejam adaptativos as características da bacia hidrográfica, às necessidades dos planejadores e que possam integrar novas ferramentas disponíveis, modelos e outros componentes físicos, socioeconômicos, ambientais e técnico-operacionais.

Com a crescente evolução tecnológica, faz mister pensar na incorporação de novas tecnologias nos SSD, provendo-os com novas ferramentas metodológicas para melhorar a gestão de recursos hídricos. Por outro lado, na nova era da informação *Big Data* a facilidade da instrumentação e medições, barateamento de microcontroladores e sensores têm gerado um

grande volume de dados que precisam ser armazenados em bancos de dados de forma dinâmica e que possam prover informações para tomada de decisões em um SSD.

Os SSDs sobre recursos hídricos têm sido objeto de estudo de diversos pesquisadores e de demanda por instituições de gestão de recursos hídricos. O Grupo de Otimização Total da Água (GOTA) tem desenvolvido modelos de otimização e simulação baseados em métodos multiobjetivos e multicriteriais para dar suporte às tomadas de decisão em recursos hídricos. Além disso, tem procurado desenvolver e aperfeiçoar modelos de banco de dados dinâmico que atendam às necessidades multiobjectivos de um sistema e que possua interface amigável para serem usados por usuários finais, permitindo a inclusão dos tomadores de decisão nos processos de modelagem e planejamento.

Capítulo 2

2 Revisão Bibliográfica

Neste capítulo são abordados os vários temas necessários que formam a base de conhecimento para a concepção de um Sistema de Suporte à Decisão (SSD). São abordados assuntos da área de Análise de Sistemas, Gestão da Informação, Engenharia de Software, entre outros e suas interfaces com a Análise de Sistemas de Recursos Hídricos. Também são apresentados temas mais conceituais sobre a base da construção da informação, tópicos técnicos como a arquitetura de sistemas e o modelo de análise para a construção do software utilizado, o modelo orientado a objetos. Ao final é realizado um levantamento de algumas aplicações de SSDs em recursos hídricos que permitem a incorporação de experiências para o aperfeiçoamento do desenvolvimento do SSD proposto.

2.1. De Dado à Sabedoria

A conceituação de dado, informação, conhecimento e sabedoria, e seus entendimentos, torna-se essencial para gestores e analistas de recursos hídricos caracterizar, interpretar e tomar decisões em sistemas de recursos hídricos. Na ciência e gestão da informação, a hierarquia DIKW (do inglês, *Data-Information-Knowledge-Wisdom*) estrutura as etapas genéricas para a apropriação do saber a partir da coleta de dados.

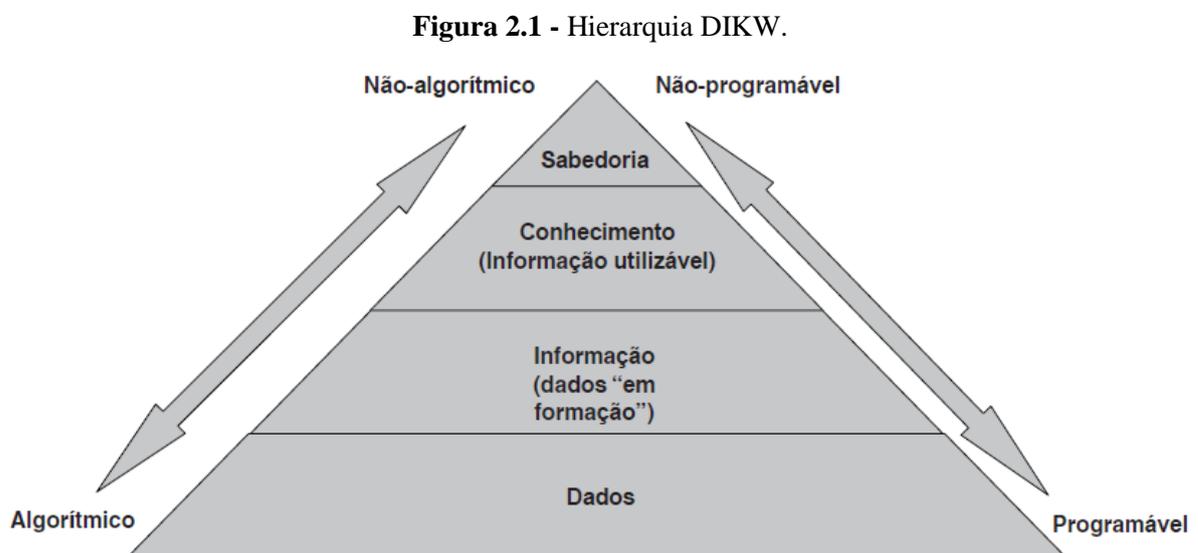
Para Davenport e Prusak (1998), os dados descrevem apenas parte daquilo que aconteceu, não fornecem julgamento nem interpretação e nem qualquer base sustentável para a tomada de decisão. Embora a matéria-prima do processo decisório possa incluir dados, eles não podem dizer o que fazer e são matéria-prima essencial para a criação da informação. Um hidrometrista ao realizar leituras de cotas de um determinado rio em uma régua limnimétrica não obtém nenhuma informação em si, apenas dados sobre uma variável medida.

A informação pode ser considerada o processamento dos dados para algum propósito. Os dados são coletados, organizados, ordenados, e atribuídos significados e contexto, trazendo assim alguma informação (McGee e Prusak, 1997). Portanto, a informação é o dado que sofreu alguma análise e permite a tomada de decisão. O hidrólogo de posse de uma série histórica de cotas num determinado trecho de rio pode confeccionar a curva-chave do mesmo, produzindo informações úteis a serem utilizadas em projetos hidráulicos, como a vazão máxima.

Conhecimento está associado a capacidade de interpretação e expertise do analista. O conhecimento é a combinação de dados e informações, à qual se acrescenta a opinião de especialistas, habilidades e experiência, para resultar em um bem valioso que pode ser usado para auxiliar à tomada de decisão (Rowley, 2007). Um hidrólogo experiente de posse da curva-chave de um determinado rio pode inferir sobre algumas informações devido sua experiência na região, como a construção de um barramento a montante do trecho medido a qual influenciou os resultados dos parâmetros da equação na construção da curva.

Sabedoria é a capacidade de agir de forma crítica ou prática em qualquer situação. Baseia-se em juízo ético relacionado ao sistema de crenças de um indivíduo (Jashapara, 2005). Quando se obtém o saber é possível validar o entendimento sobre algo. Outro hidrólogo, por exemplo, pode ir mais além e verificar os processos erosivos da bacia, a partir de estudos do uso do solo da mesma, e verificar a situação de assoreamento da calha do rio, concluindo a necessidade de uma nova curva-chave para a seção, ou ainda, pode ser mais prático e utilizar metodologias de regionalização de vazão.

A Figura 2.1 apresenta a hierarquia do conhecimento relacionada aos níveis de desenvolvimento de algoritmos e programação de um sistema para dar suporte a decisões.



2.2. Sistemas e seus Componentes

Com base na Teoria Geral dos Sistemas (TGS) por Bertalanffy (1968), pesquisadores de diversas áreas tem desenvolvido o tema que hoje atende às mais diversas ciências. Segundo

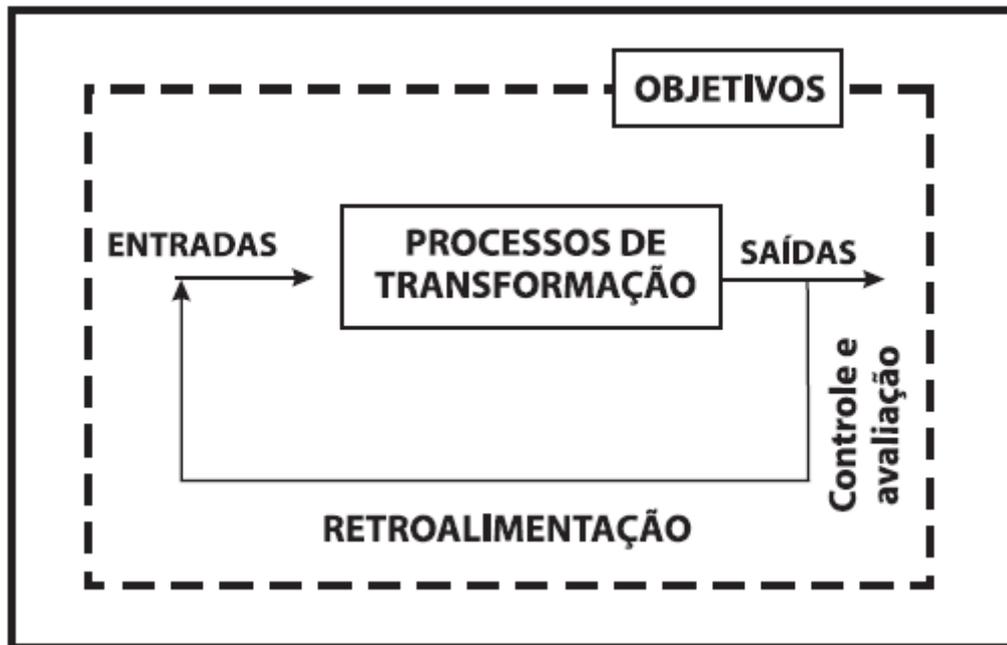
Dias (1985), sistema é um conjunto de partes coordenadas, que concorrem para a realização de um conjunto de objetivos.

Para Torres (1995) sistema é um conjunto de componentes e processos que visam transformar determinadas entradas em saídas, um conceito que se aproxima ao de Stair (2010), onde um sistema é um conjunto de elementos ou componentes que interagem para se atingir um ou mais objetivos.

Pode-se notar que os conceitos apresentados são bastante genéricos e podem ser aplicados a diversas análises no mundo real. Todos os sistemas são compostos por diferentes componentes, elementos, partes ou entidades interconectados que interagem entre si e geram algum tipo de saída, portanto, a relação que se estabelece entre eles é que determina como o sistema funciona e quais serão seus resultados.

Assim, as atividades de um sistema dependem da entrada, processamento, saída e realimentação por algum tipo de insumo. A Bacia Hidrográfica pode ser considerada um sistema natural, onde, a entrada é um evento de precipitação, os processos do ciclo hidrológico são o processamento, a vazão em um ponto de controle gerada pelo escoamento é a saída e, finalmente, outro evento de chuva realimenta o sistema, que pode se comportar de forma diferente. A Figura 2.2 apresenta as atividades desenvolvidas dentro de um sistema.

Figura 2.2 - Atividades de um Sistema.



Fonte: Mülber e Ayres, 2005.

2.3. Modelagem de Sistemas

Um modelo é a simplificação de um sistema, é uma forma de abstrair algumas características do mundo real afim de facilitar a análise e simulação de processos inerentes ao sistema. Os modelos devem ter linguagem, forma de uso e acesso fáceis. Quanto mais complexos são os sistemas mais é necessário seu processo de modelagem para análise.

Para Stair (2010), os modelos podem ser tipificados em: narrativos (são modelos descritos através de palavras, como exemplo: relatórios, conversações a respeito de um dado sistema, etc.); físicos (são modelos reduzidos – protótipos ou mesmo simulados em computador); esquemáticos (são modelos representados por gráficos, diagramas, figuras, mapas, etc.); e matemáticos (são representações aritméticas da realidade).

2.4. Análise e Modelagem de Sistemas Hídricos

Como mencionado anteriormente, a TGS foi aplicada em várias áreas do conhecimento, não sendo diferente na análise em recursos hídricos. Dooge (1973) faz um apanhado dos conceitos e definições utilizados em recursos hídricos e conceitua sistema como qualquer estrutura, esquema ou procedimento, real ou abstrato, que num dado tempo referência inter-relaciona-se com uma entrada, causa ou estímulo de energia ou informação, e uma saída, efeito ou resposta de energia ou informação.

Tucci (2005) sintetiza sistema como qualquer engenho que responde, por meio de uma saída, a uma entrada. Cada sistema possui suas próprias características e pode, em função de cada caso, ser subdividido em subsistemas onde certas características são mais homogêneas.

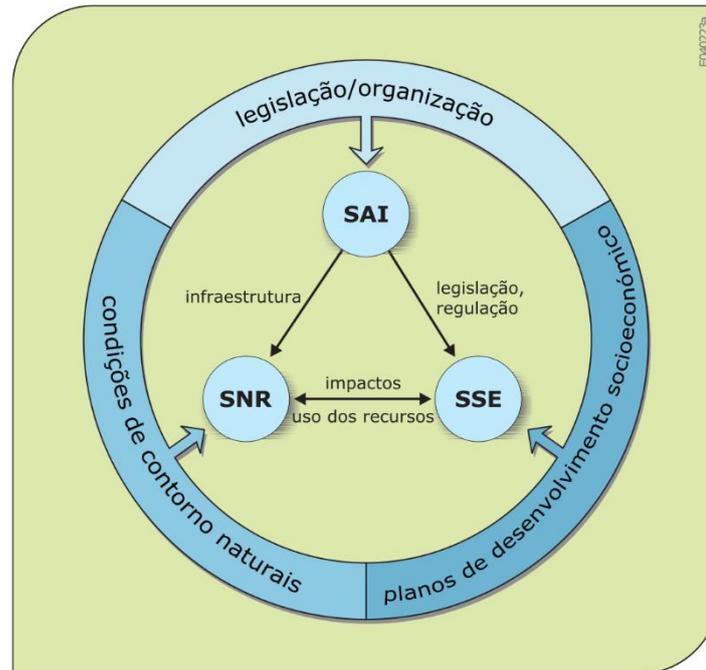
A Parceria Mundial da Água (PMA, do inglês, GWP – *Global Water Partnership*) define a Gestão Integrada de Recursos Hídricos (GIRH, do inglês, IWRM - *Integrated Water Resources Management*) com uma abordagem que promove o desenvolvimento e gestão coordenada da água, solo e recursos relacionados, a fim de maximizar o bem-estar econômico e social de uma forma equitativa sem comprometer a sustentabilidade de ecossistemas vitais.

Nessa abordagem holística deve-se considerar a influência e interação de três subsistemas interdependentes, são eles:

- O subsistema natural do rio, onde ocorrem os processos físicos, químicos e biológicos;
- O subsistema socioeconômico, que inclui as atividades humanas relacionadas com a utilização do sistema natural do rio;
- O subsistema administrativo e institucional de administração, legislação e regulamentação, onde os processos de decisão, de planejamento e gestão ocorrem.

A Figura 2.3 apresenta as interações entre os subsistemas e entre eles e seus ambientes.

Figura 2.3 - Interações entre os subsistemas e entre eles e seus ambientes.



Fonte: Adaptado de Loucks e van Beek, 2005 (tradução).

Segundo Braga, Barbo e Nakayama (1998), a análise de sistemas de recursos hídricos é um enfoque sistêmico através do qual os componentes do sistema de recursos hídricos e suas interações são descritas em termos quantitativos - devendo-se acrescentar aqui termos qualitativos - por meio de equações matemáticas e funções lógicas.

Na análise de sistemas de recursos hídricos técnicas de modelagem de simulação e otimização têm sido aplicadas no planejamento de recursos hídricos desde a década de 1960s (Loucks e van Beek, 2005; Yeh, 1985; Labadie, 2004 e Wurbs, 2005). Em modelos de otimização, o problema é representado através de funções matemáticas, que contem parâmetros e variáveis de decisão, a serem maximizadas ou minimizadas. Já em modelos de simulação não é objetivo encontrar a máxima sinergia entre os componentes do sistema hídrico, mas sim analisar cenários e o seu comportamento, possibilitando maior interação com o mesmo.

Apesar de frequentemente utilizados no planejamento e gerenciamento de recursos hídricos, os modelos citados anteriormente apresentaram algumas dificuldades como: tamanho dos programas na solução de problemas complexos, falta de comunicação entre tomadores de decisão e analistas, consideração de múltiplos objetivos, inclusão de aspectos não quantitativos no processo de decisão, entre outros, fizeram com que a utilização destas técnicas ficasse prejudicada em aplicações práticas de engenharia (Loucks e van Beek, 2005; Mota *et. al.*, 2011).

O avanço tecnológico crescente dos computadores em termos de armazenamento de dados e velocidade de processamento, o desenvolvimento das linguagens de programação de alto nível¹, a popularização da tecnologia e o compartilhamento e colaboração dos desenvolvedores tem mudado o paradigma da aplicação de modelos na análise em recursos hídricos.

Estas mudanças trouxeram facilidades para uso dos modelos, e o mais importante, a comunicação entre os analistas/modeladores e os tomadores de decisão. Assim novos conceitos de modelagem aberta e modelos de visão compartilhada, onde todos envolvidos no planejamento e gestão dos recursos hídricos participam do processo de modelagem e simulação, foi estabelecido.

Loucks e van Beek (2005) citam as principais características dos softwares e modelos atuais que proporcionaram essa quebra de paradigma, são:

- Abordagens interativas para operação do modelo que põe os usuários mais no controle de seus computadores, modelos e dados;
- Gráficos computacionais que facilitam a entrada de dados, edição, visualização e compreensão;
- Sistemas de Informação Geográfica que fornecem aperfeiçoamento na análise espacial e na capacidade de visualização;
- Sistemas especialistas que pode ajudar o usuário a entender melhor como problemas de decisão complexo poderia ser resolvido, e ao mesmo tempo explicar aos usuários por uma decisão particular pode ser melhor do que outra;
- E-mail e Internet, os quais deixam os analistas, planejadores e gestores se comunicarem e compartilhem dados e informações com outras pessoas em todo o mundo, utilizar modelos que estão localizados e mantidos em locais distantes;
- Sistemas multimídia que permitem o uso de som e animações de vídeo em análises, tudo com objetivo de melhorar a comunicação e compreensão.

¹ Linguagem de programação de alto nível é como se chama, na Ciência da Computação de linguagens de programação, uma linguagem com um nível de abstração relativamente elevado, longe do código de máquina e mais próximo à linguagem humana, exemplos: C/C++, Matlab, R, Python, Java, PHP, Object Pascal, etc.

2.5. Sistemas de Banco de Dados

Atualmente, é possível afirmar que os bancos de dados têm papel providencial para o funcionamento de qualquer sistema computacional. Em diversas áreas, seja na engenharia, economia ou genética, os bancos de dados atuam com a finalidade de armazenar dados relacionados de algum sistema ou aplicação.

A Agência Nacional da Águas (ANA), por exemplo, necessita de um banco de dados para armazenar informações coletadas pelas estações hidrométricas da rede nacional, estes dados podem ser armazenados em planilhas eletrônicas, arquivos de texto delimitados ou até em formatos não digitais considerados bancos de dados não computadorizados. Portanto, um banco de dados é uma coleção de dados operacionais armazenados, sendo usados pelos sistemas de aplicação de uma determinada organização (Date, 2004).

2.5.1 Modelos de Dados

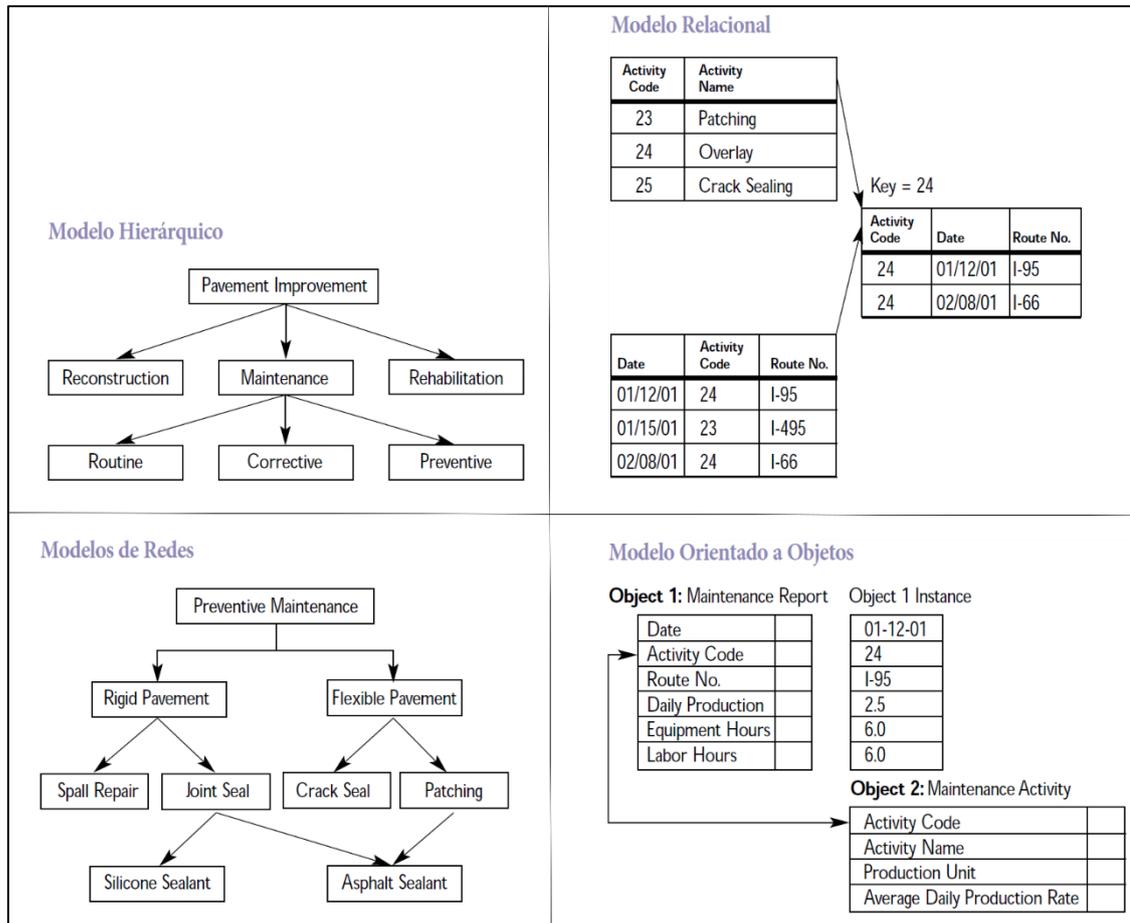
Uma característica importante em um banco de dados é o modelo de dados que será armazenado. A estrutura do banco de dados está diretamente relacionada com seu modelo de dados. Elmasri e Navathe (2011) definem modelos de dados como uma coleção de conceitos que podem ser usados para descrever a estrutura de um banco de dados. Os mesmos autores classificam os modelos como:

- Modelo de dados de alto nível ou conceitual: são modelos de dados estruturados de forma próxima de como usuário final os entende no mundo real. Entidades, atributos e relacionamentos são alguns dos conceitos utilizados nesse tipo de modelo. Modelo Entidade-Relacionamento e Modelo de Objetos são exemplos deste tipo;
- Modelo de dados representativo ou de implementação: são modelos que também oferecem conceitos fáceis de serem entendidos por um usuário final, porém não está muito longe da forma de como os dados são armazenados. Modelos de Dados Relacional, Hierárquico e de Rede são exemplos deste modelo, sendo o primeiro o mais utilizado atualmente e os dois últimos considerados modelos legados²;
- Modelo de dados de baixo nível ou físico: fornece conceitos que descrevem os detalhes de como os dados são armazenados na mídia de armazenamento do computador.

² Termo utilizado para modelos ou sistemas que foram bastante utilizados no passado, mas devido o desenvolvimento tecnológico entraram em desuso.

A Figura 2.4 apresenta diagramas ilustrando alguns tipos de modelos de dados.

Figura 2.4 - Diagramas ilustrando os modelos de dados.



Fonte: Adaptado de US DOT, 2001 (tradução).

2.5.2. Sistemas de Gerenciamento de Banco de Dados

Para realizar o gerenciamento de banco de dados se utiliza um Sistema de Gerenciamento de Banco de Dados (SGBD, do inglês, DBMS - *Database Management System*), que é uma coleção de programas para criar, operar e manter o banco de dados. O SGBD é um sistema de software de uso geral que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre vários usuários e aplicações (Elmasri e Navathe, 2011).

No processo de definição é especificado o tipo, estrutura e restrições dos dados que serão armazenados. Essas informações descritivas são armazenadas no SGDB e se chamam metadados. Na construção, o banco de dados é armazenado em um local controlado pelo SGDB.

A manipulação dos dados acontece por meio de funções de consulta ou *query*, onde pode-se criar, ler, atualizar e deletar algum registro, na linguagem informal o conjunto dessas operações se chama CRUD (*Create, Read, Update e Delete*, ou seja, Criar, Ler, Atualizar e Apagar, respectivamente). Por último, o compartilhamento permite o acesso do banco de dados por diversos usuários e aplicativos.

Os sistemas desenvolvidos ou programas de aplicação acessam o banco de dados através de consultas ou requisições de dados para o SGBD. A consulta/*query* normalmente faz com que alguns dados sejam recuperados, uma transação pode causar que alguns dados sejam lidos e outros sejam gravados no banco de dados (Elmasri e Navathe, 2011).

Para a ANA disponibilizar os dados do seu banco interno para diversas instituições e usuários um SGBD é utilizado, outra função do SGBD é manter a segurança dos dados contra defeitos ou falhas e contra acessos não-autorizados ou maliciosos. Mota (2011) destaca vantagens da abordagem da utilização de uma SGBD, são eles:

- Compartilhamento de dados;
- Rapidez na manipulação e no acesso aos dados;
- Disponibilidade dos dados em tempo hábil;
- Reduzir o esforço humano em seu desenvolvimento de softwares;
- Controle integrado de informações distribuídas fisicamente;
- Aplicação automática de restrições de segurança;
- Redução de problemas de integridade dos dados.

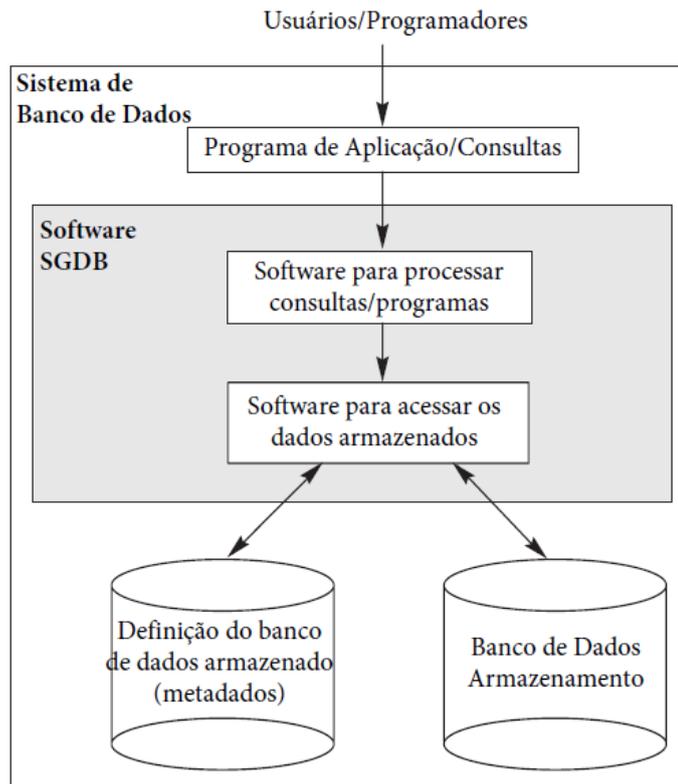
Portanto, um Sistema de Banco de Dados pode ser definido como a junção do banco de dados e o SGBD. Outras partes integrantes de um sistema de banco de dados são a parte física, o hardware, a comunicação, e os usuários que se utilizam do sistema. O hardware compreende os elementos físicos que compõe o sistema de banco de dados, como as mídias de armazenamento, os canais de entrada/saída, redes, entre outros. Quanto aos usuários pode-se destacar três categorias, segundo Date (2004), são elas:

- Programador de aplicação: desenvolve programas sobre o banco de dados, ou seja, cria aplicações que acessarão o sistema de banco de dados;
- Usuário final: público que consulta e atualiza o banco de dados utilizando-se, geralmente, das aplicações desenvolvidas pelos programadores de aplicações. Pode ter conhecimentos da área de Tecnologia da Informação (TI);

- Administrador de Banco de Dados (DBA, do inglês, *Database Administrator*): responsáveis por gerir o SGDB.

A Figura 2.5 apresenta um diagrama simplificado de um ambiente de sistema de banco de dados.

Figura 2.5 - Diagrama de um Sistema de Banco de Dados.



Fonte: Adaptado de Elmasri e Navathe, 2011 (tradução).

2.5.3. Arquitetura para os SGDBs

A arquitetura dos SGDBs tem acompanhado a evolução tecnológica dos sistemas computacionais em geral.

Antigamente computadores centrais com bom poder de processamento, os chamados *mainframes*, tinham que processar todas as funções de programas de aplicação, interface com usuários e as funcionalidades dos SGBDs. Assim, o SGDB era centralizado e tudo ocorria em apenas uma máquina. Com o passar do tempo os SGBDs começam a explorar o poder de processamento disponível do lado do usuário (PCs e estações de trabalho), gerando as arquiteturas cliente-servidor para os SGDBs (Elmasri e Navathe, 2011).

Na arquitetura cliente-servidor a máquina lado cliente possui um programa de aplicação que requisita algum serviço a máquina servidora, então quando um programa de aplicação no lado cliente requer acesso a alguma funcionalidade adicional, como o acesso ao banco de dados, ele se conecta a um servidor que oferece a funcionalidade como resposta.

Um servidor pode oferecer vários serviços ao cliente como acesso à arquivos, impressão e acesso a banco de dados. Em alguns casos em um mesmo computador pode-se ter instalado o software cliente e servidor. Porém, é mais comum que o software cliente e servidor sejam executados em máquinas separadas (Elmasri e Navathe, 2011). O Servidor é composto de equipamentos de maior poder computacional em termos de processamento e armazenamento. Já os clientes são geralmente computadores pessoais, menos robustos e conectados em rede.

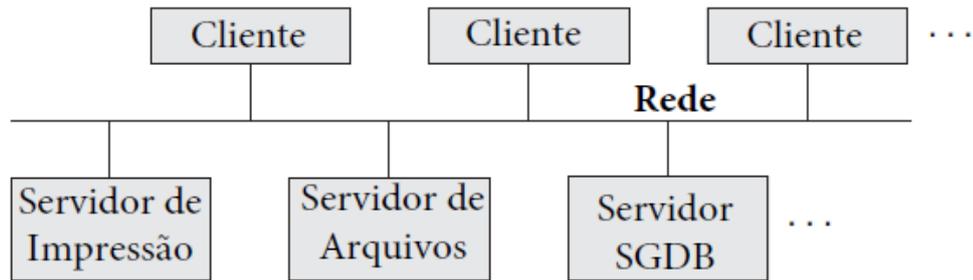
Existem três tipos principais de arquiteturas para sistemas de software na abordagem SGBD e estrutura cliente/servidor: duas camadas, três camadas e n -camadas.

Na arquitetura duas camadas tanto o programa de interface com o usuário como o programa de aplicação são executados no lado do cliente. Quando é necessário acessar o SGBD, se estabelece uma conexão com o SGBD (que está no lado do servidor) e então o programa cliente pode se comunicar com o SGBD.

Um padrão denominado Conectividade de Banco de Dados Aberta (ODBC - *Open Database Connectivity*) oferece uma Interface de Programação de Aplicações (API - *Application Programming Interface*), que permite que os programas do cliente chamem o SGBD, desde que as máquinas cliente e servidor tenham o software necessário instalado (Elmasri e Navathe, 2011).

A Figura 2.6 apresenta a arquitetura em duas camadas, representando vários clientes acessando diversos serviços em servidores dedicados.

Figura 2.6 - Arquitetura cliente-servidor em duas camadas.



Fonte: Adaptado de Elmasri e Navathe, 2011 (tradução).

Atualmente novas necessidades e demandas em termos de eficiência de processamento fizeram que novas aplicações utilizem uma arquitetura chamada arquitetura de três camadas, que adiciona uma nova camada intermediária entre o cliente e o servidor.

Essa camada intermediária é chamada de servidor de aplicação, *App Server* ou servidor web, dependendo da aplicação. A função dessa camada é mediar a execução de programas de aplicação que possuem as regras ou lógica de negócios (procedimentos, funções ou restrições), que são usadas para acessar os dados do servidor de banco de dados, processá-los e retorná-los para camada de interface com o usuário ou de apresentação. Outra função desta camada é melhorar a segurança do banco de dados, verificando as credenciais de um cliente antes de encaminhar uma solicitação ao servidor de banco de dados.

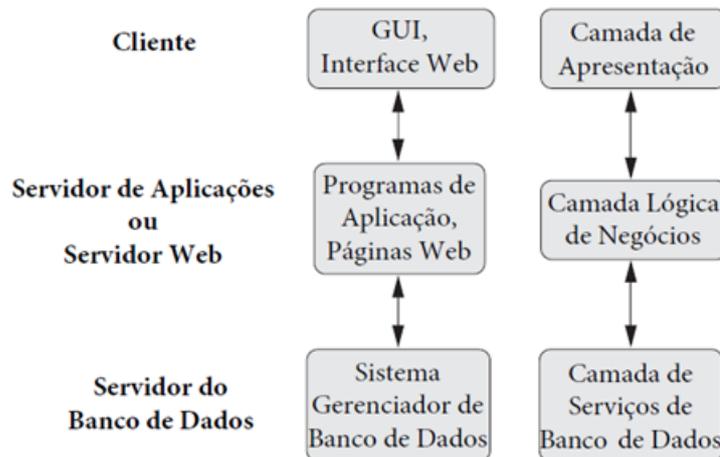
Nesta arquitetura os clientes possuem apenas as interfaces GUI (*Graphic User Interface*) e algumas regras de negócios adicionais, quando necessárias.

O servidor intermediário aceita e processa solicitações do cliente, e envia consultas e comandos do banco de dados ao servidor de banco de dados, e depois atua como um canal para passar (parcialmente) dados processados do servidor de banco de dados aos clientes, onde podem ainda ser processados e filtrados para serem apresentados aos usuários no formato da GUI. Assim, a interface com o usuário, as regras da aplicação e o acesso aos dados atuam como três camadas (Elmasri e Navathe, 2011).

A Figura 2.7 apresenta a arquitetura cliente-servidor em três camadas. A camada de apresentação exibe informações ao usuário e permite a entrada de dados. A camada de lógica de negócios cuida das regras e restrições intermediárias antes dos dados serem passados para o usuário ou ao SGBD. A camada inferior inclui todos os serviços de gerenciamento de dados. A camada do meio também pode atuar como um servidor web, que recupera resultados das

consultas do servidor de banco de dados e os formata para as páginas web dinâmicas, que são vistas pelo navegador web no lado do cliente (Elmasri e Navathe, 2011).

Figura 2.7 - Arquitetura cliente-servidor três camadas.



Fonte: Adaptado de Elmasri e Navathe, 2011 (tradução).

Na arquitetura cliente-servidor n -camadas ou múltiplas camadas, ou ainda em inglês *multi-tier*, geralmente a camada da lógica de negócios é dividida em mais camadas. O servidor de aplicações e o servidor web, e até outros mais, podem coexistir caso a aplicação possua diversas camadas de apresentação como desktop, mobile e web.

Além de distribuir a programação e os dados pela rede, as aplicações de n camadas têm a vantagem de que qualquer camada poder ser executada em um processador ou plataforma de sistema operacional adequado e ser tratada independentemente, ou seja, você pode ter uma camada de lógica de negócios que pode estar ligada a diferentes camadas de apresentação de vários usuários e sistemas diferentes (Elmasri e Navathe, 2011).

2.6. Sistemas Orientados a Objetos

Orientação a objetos, também conhecida como Programação Orientada a Objetos (POO) ou ainda em inglês *Object-Oriented Programming* (OOP) é um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos (Reisswitz, 2009). A programação orientada a objetos é caracterizada pelo uso de diversos mecanismos como encapsulamento, herança e métodos.

Em diversas situações o termo modelagem orientada ao objeto, em vez de programação, é utilizado. A modelagem orientada a objetos nasceu na necessidade de modelar sistemas complexos do mundo real, como sistemas de recursos hídricos, que possuem diversos componentes e suas interações. Geralmente na modelagem de problemas de engenharia, em especial de recursos hídricos, os modeladores e planejadores definem a área de estudo e o problema a ser resolvido (domínio da análise), os sistemas hídricos possuem diversos componentes (objetos) que se relacionam (mensagens) e produzem resultados (estados) diferentes a partir das intervenções realizadas. Essa modelagem é representada matematicamente por algoritmos.

Para Viegas e Lanna (2003) na modelagem convencional, o modelador faz a separação entre os atributos e os processos envolvidos, programando apenas a implementação computacional desses últimos e considerando os primeiros tão somente como elementos de entrada e saída.

Ainda segundo Viegas e Lanna (2003), já na modelagem orientada a objetos busca-se tratar de modo integrado, em todas as suas fases, os dados e os processos envolvidos, utilizando-se para tal das características introduzidas pelas modernas linguagens orientadas a objetos, fazendo com que o modelador tenha sempre, ao longo do processo, a visualização integral do modelo associado com o contexto que ele busca representar e, além disso, que outros modeladores possam fazê-lo, através da criação de bibliotecas de uso amplo, facilitando o trabalho em equipe e a associação de grupos mais abrangentes, como é o caso das redes de pesquisa.

A modelagem orientada a objetos tem como estratégia eliminar o chamado “gap semântico”, dificuldade recorrente no processo de modelar o mundo real, domínio do problema, em um conjunto de componentes de software que seja o mais fiel na sua representação deste domínio (Reisswitz, 2009). Ou seja, esta abordagem facilita a comunicação do profissional modelador e do usuário da área alvo, pois deixa mais óbvia e natural a simbologia e conceitos abstratos do mundo real da ferramenta de modelagem.

Este novo paradigma proporcionou avanços para o desenvolvimento de modelos e para análise de sistemas na área de recursos hídricos, contribuindo para a comunicação entre os analistas/modeladores e os tomadores de decisão. Possibilitou, ainda, a criação de modelos de visão compartilhada conforme citado por Loucks e van Beek (2005).

A Tabela 2.1 apresenta um resumo dos principais conceitos da modelagem orientada a objetos fazendo uma analogia aos componentes dos sistemas de recursos hídricos.

Tabela 2.1 - Principais conceitos da modelagem orientada a objetos.

Conceito	Descrição	Exemplo
Classe	Representa o conjunto de objetos semelhantes que tem características afins. É a entidade do mundo real.	A caracterização de uma Bacia Hidrográfica pelo analista resultando a classe Bacia.
Objeto	E a instância de uma classe, um elemento do mundo real objetivo de análise com seus atributos.	Bacia Hidrográfica do Rio Paraíba e suas peculiaridades.
Atributo	Refere-se as características particular de um objeto. Basicamente a estrutura de dados que vai representar a classe.	Classe Bacia possui área, perímetro, ordem, comprimento do rio principal, talvegue etc.
Método	Operação lógica presente em uma classe para reproduzir um comportamento. As habilidades do objeto.	Cálculo da sinuosidade da Bacia pertencente a classe Bacia invocada pelo objeto Bacia do Rio Paraíba.
Encapsulamento	Consiste na separação de aspectos internos e externos de um objeto, relacionando seus atributos e métodos, encapsulando detalhes.	Atributos: Comprimento do rio principal, comprimento do talvegue. Método: Cálculo da sinuosidade.
Herança	Mecanismo pelo qual uma classe (subclasse) pode estender outra classe (superclasse), aproveitando seus métodos atributos.	A subclasse Sub-Bacia compartilha atributos e métodos da superclasse Bacia.
Subclasse	Uma nova classe que herda características de sua(s) classe(s) ancestral(is).	Classe Bacia tem as subclasses Sub-Bacia e Micro-Bacia.
Mensagem	Requisição entre objetos para solicitar/invocar algum método.	Calcular a curva hipsométrica da bacia x.
Polimorfismo	Habilidade de usar a mesma mensagem para invocar comportamentos diferentes do objeto.	Chama o método: Calcular Vazão Máxima. Invoca as derivações Método Racional, I-Pai-Wu, SCS, etc.

Fonte: Autor.

As análises e projetos orientados a objetos têm como meta identificar o melhor conjunto de objetos para descrever um sistema. O funcionamento deste sistema se dá através do relacionamento e troca de mensagens entre estes objetos. Na programação orientada a objetos, implementa-se um conjunto de classes que definem os objetos presentes no sistema de software. Cada classe determina o comportamento (definido nos métodos) e estados possíveis (atributos) de seus objetos, assim como o relacionamento com outros objetos (Reisswitz, 2009).

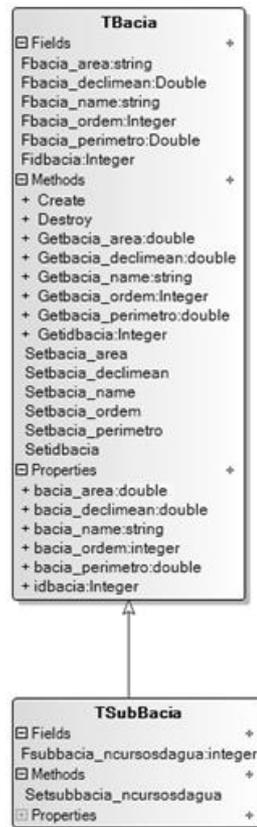
São exemplos de linguagens de programação orientadas a objetos: C++, C#, VB.NET, Java, Object Pascal, Objective C, Python e Ruby. Javascript, R, MatLab, PHP, Perl e Visual Basic são exemplos de linguagens de programação com suporte a orientação a objetos.

Na área de Engenharia de Software, a Linguagem de Modelagem Unificada (do inglês, UML - *Unified Modeling Language*) é uma linguagem de modelagem que permite representar um sistema de forma padronizada, ela auxilia a visualização da comunicação entre os objetos.

Basicamente, a UML permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados. Junto com uma notação gráfica, a UML também especifica significados/semântica, fazendo uma analogia a plantas arquitetônicas na engenharia civil desenvolvidas em programas CAD (*Computer Aided Design*), a UML fornece os diagramas para construção das “plantas” dos softwares na engenharia de software. Ferramentas CASE (*Computer Aided Software Engineering*) são utilizadas para realizar o processo de modelagem do software, na verdade elas abrangem todas ferramentas baseadas em computadores que auxiliam atividades de engenharia de software, desde análise de requisitos e modelagem até programação e testes.

A Figura 2.8 apresenta um diagrama de classe, que é um dos métodos gráficos utilizados na UML, de uma classe denominada Bacia com seus atributos e métodos que são herdados para a classe SubBacia.

Figura 2.8 - Exemplo de um Diagrama de Classe.



Fonte: Autor.

2.7. Sistema de Informações

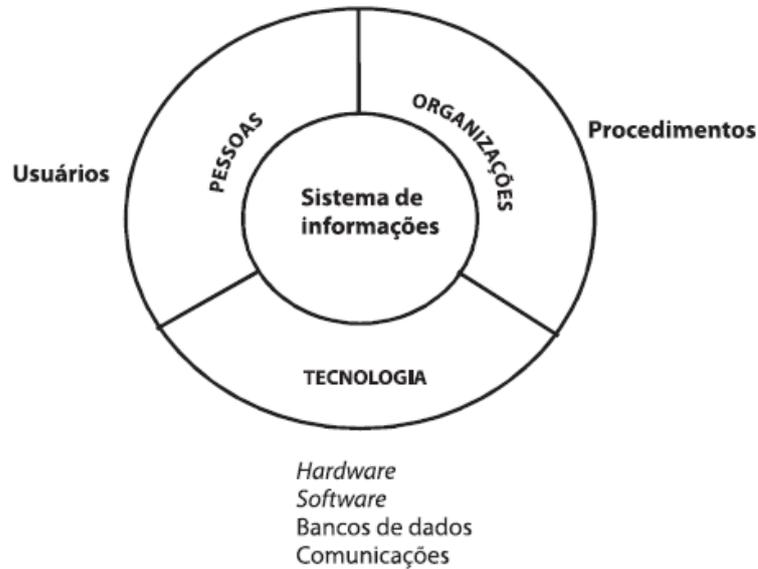
Um Sistema de Informações (SI) pode ser definido como um conjunto de componentes inter-relacionados e integrados, que tem o objetivo de coletar, armazenar, disseminar e processar dados, produzindo informação para facilitar a coordenação, o controle, a análise, a visualização e o processo decisório das organizações e indivíduos (Laudon e Laudon, 2012). Este conceito vai um pouco além do conceito generalista de sistema visto anteriormente, pois inclui as ações de um SI e quem ele dá suporte.

Pode-se associar as atividades descritas anteriormente realizadas em um sistema às realizadas em um SI, são elas:

- **Entrada:** a entrada é a atividade de coleta e captura dos dados brutos. Em um SI sobre recursos hídricos, para se produzir um hidrograma relativos as vazões de uma certa seção de um rio, há a necessidade de coletar os dados previamente por meio de uma estação fluviométrica. Independentemente do sistema envolvido, o tipo de entrada é determinado pela saída desejada do sistema;
- **Processamento:** processamento significa converter ou transformar dados em saídas úteis. O processamento pode envolver fazer cálculos, comparar dados, tomar ações alternativas e armazenamento de dados para uso futuro. Uma agência de recursos hídricos processa os pedidos de outorga de uso de água utilizando o código atribuído a um usuário, relaciona a demanda com a oferta, pode tomar alguma ação e, finalmente, armazenar o registro para uso futuro;
- **Saída:** a saída envolve a produção de informações úteis, geralmente sob a forma de documentos e relatórios. Um relatório mensal com a situação de oferta d'água de um determinado reservatório realizado a partir de séries dos múltiplos usos do mesmo, é uma informação útil para tomada de decisões naquele sistema;
- **Feedback:** é a informação do sistema que é usado para fazer mudanças nas atividades de entrada ou de processamento. O cancelamento de outorgas e início do racionamento d'água, entre outras situações, são mudanças que irão afetar o sistema produzindo novas entradas (novos valores de demandas) e mudanças no processamento das informações.

Basicamente um SI é um produto de três componentes: a tecnologia, as organizações e as pessoas. A Figura 2.9 ilustra os componentes de um sistema de informações.

Figura 2.9 - Componente de um Sistema de Informações.



Fonte: Mülber e Ayres, 2005.

Analisando a Figura 2.9 percebe-se que dentro de um SI existem outros sistemas, como por exemplo, o Sistema de Banco de Dados que integra componente tecnologia.

As organizações equivalem a coleção de processos operacionais e administrativos. Nestes processos incluem estratégias, políticas, métodos e regras. Regras operacionais de um reservatório podem ser incorporadas em um SI, por exemplo.

As pessoas podem ser os elementos mais importantes na maioria dos sistemas de informação. Elas fazem a diferença entre sucesso e fracasso para a maioria das organizações. Ao pessoal de sistemas de informação está incluída todas as pessoas que gerenciam, executam, programam, e mantem o sistema (Stair e Reynolds, 2014). Uma agência de gestão de recursos hídricos necessita de pessoas capacitadas e com condições de trabalho para fazer o uso efetivo de um SI.

2.8 Sistema de Informações Geográficas

Atualmente os Sistemas de Informações Geográficas (SIGs, no singular SIG, ou ainda em inglês, GIS - *Geographic Information System*) tem aplicações em praticamente todas as áreas do conhecimento e durante as três últimas décadas têm sido utilizados constantemente no planejamento de recursos hídricos. A maioria dos componentes de um sistema hídrico tem características multidimensionais, ou seja, no espaço e no tempo, é nesse aspecto que os SIGs

tem se mostrado um aliado muito importante para análises que auxiliam a tomada de decisão dos planejadores.

Um SIG pode ser definido como uma aplicação de computador capaz de realizar praticamente qualquer operação concebível em informação geográfica, desde a aquisição e compilação através da visualização, consulta e análise para modelagem, compartilhamento e arquivamento (Longley *et al.*, 2010). Um SIG possui os mesmos componentes citados anteriormente em um SI, porém com facilidades para trabalhar os dados espaciais.

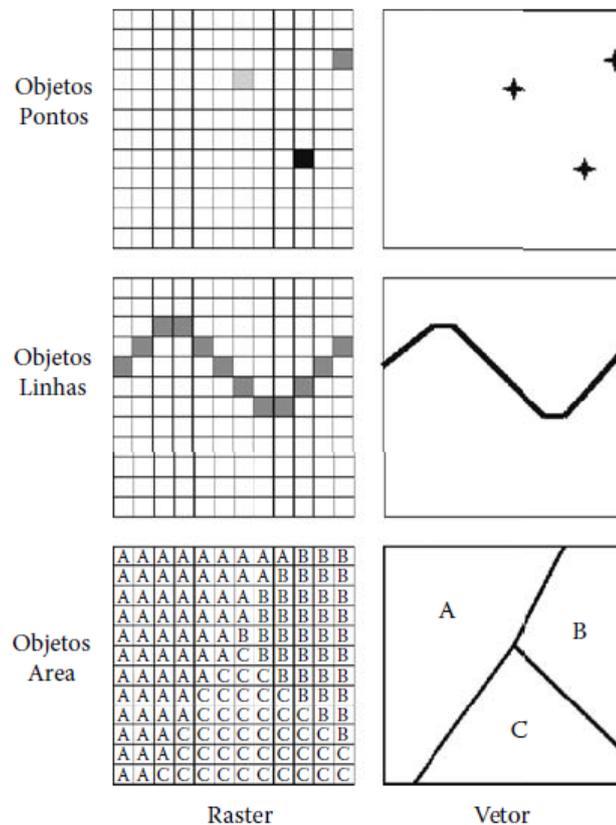
Os SIGs nasceram junto com a evolução tecnológica, havia necessidade de armazenar e analisar o crescente número de dados espaciais de forma rápida. Assim, foram criados os bancos de dados espaciais, com seus modelos de dados e suas arquiteturas.

A difusão tecnológica tem auxiliado no desenvolvimento dos SIGs proporcionando a criação de novos modelos de dados e arquiteturas, padronização de formatos, funcionalidades de análise avançadas (modelagem), interoperabilidade de dados, sistemas livres e novas formas de apresentação colaborativas (*Web Mapping*). O *Open Geospatial Consortium* (OGC) tem trabalho no desenvolvimento e na implementação de padrões para conteúdo e serviços geomáticos³, SIG, processamento de dados e compartilhamento.

Basicamente nos modelos de dados geográficos os objetos espaciais são constituídos dos atributos e da geometria. Os atributos são caracterizados por estruturas temáticas ou semânticas, dados alfanuméricos, enquanto a geometria é representada por estruturas geométricas-topológicas, dados gráficos. A Figura 2.10 apresenta os dois modelos de dados geométricos mais utilizados.

³ A Geomática integra todos os meios utilizados para a aquisição e gerenciamento de dados espaciais, dentre as atividades relacionadas aos serviços geomáticos estão: a cartografia, apoios topográficos, mapeamentos digitais, sensoriamento remoto, sistemas de informações geográficas, hidrografia, entre outros.

Figura 2.10 - Modelos geométricos matricial e vetorial.



Fonte: Adaptado de Johnson, 2009 (tradução).

A geometria contém informações espaciais dos objetos geográficos que podem estar associados a dados não-espaciais alfanuméricos, os seus atributos. Por exemplo, um elemento vetorial do tipo ponto possui suas coordenadas espaciais em um sistema de coordenadas pré-definido, ele pode representar um poço e ter em seus atributos dados tabulares de vazão, qualidade d'água, etc.

Câmara, Davis e Monteiro (2001), citam quatro principais tecnologias que derivam a grande variedade de SIGs disponíveis no mercado, são elas:

- Os SIGs *desktop*: sistemas herdeiros da tradição de Cartografia, com suporte de bancos de dados limitado e cujo paradigma típico de trabalho é o mapa (chamado de “cobertura” ou de “plano de informação”). Desenvolvidos a partir do início da década de 80 para ambientes da classe VAX e, a partir de 1985, para sistemas PC/DOS, esta classe de sistemas é utilizada principalmente em projetos isolados, sem a preocupação de gerar arquivos digitais de dados. Esta geração também pode ser caracterizada como sistemas orientados a projeto (*project oriented GIS*).

- A segunda geração de SIGs, os bancos de dados geográficos: chegaram ao mercado no início da década de 90 e caracteriza-se por ser concebida para uso em ambientes cliente-servidor, acoplado a SGDBs relacionais e com pacotes adicionais para processamento de imagens. Esta geração também pode ser vista como sistemas para suporte às instituições (“*enterprise oriented GIS*”);
- A terceira geração de SIGs, as bibliotecas geográficas digitais ou centros de dados geográficos: são caracterizadas pelo gerenciamento de grandes bases de dados geográficos, com acesso através de redes locais e remotas, com interface via *World Wide Web* (WWW). E para esta terceira geração, com o crescimento dos bancos de dados espaciais e a necessidade de seu compartilhamento com outras instituições, requer tecnologias como bancos de dados distribuídos e federativos. Estes sistemas deverão seguir os requisitos de interoperabilidade, de maneira a permitir o acesso de informações espaciais por SIGs distintos. A terceira geração de SIG pode ainda ser vista como o desenvolvimento de sistemas orientados para troca de informações entre uma instituição e os demais componentes da sociedade (“*society-oriented GIS*”).

Câmara, Davis e Monteiro (2001) ainda ressaltam que um aspecto fundamental destas diferentes tecnologias é sua complementaridade: os SIGs desktop podem utilizar SGDBs, que podem estar ligados a servidores web, e os usuários destes dados podem ter interfaces personalizadas, construídas a partir de componentes SIG⁴.

Como citado anteriormente na segunda geração de SIGs, os SGDBs, principalmente os relacionais, começaram a gerenciar os bancos de dados dos SIGs, primeiramente apenas os atributos dos objetos espaciais, porém, atualmente devido à grande aplicabilidade e o interesse pelo uso dos SIGs em ambiente corporativo fez com que os SGDBs comerciais incluíssem o armazenam tanto dos atributos dos objetos como de sua geometria. As principais vantagens desta estratégia são:

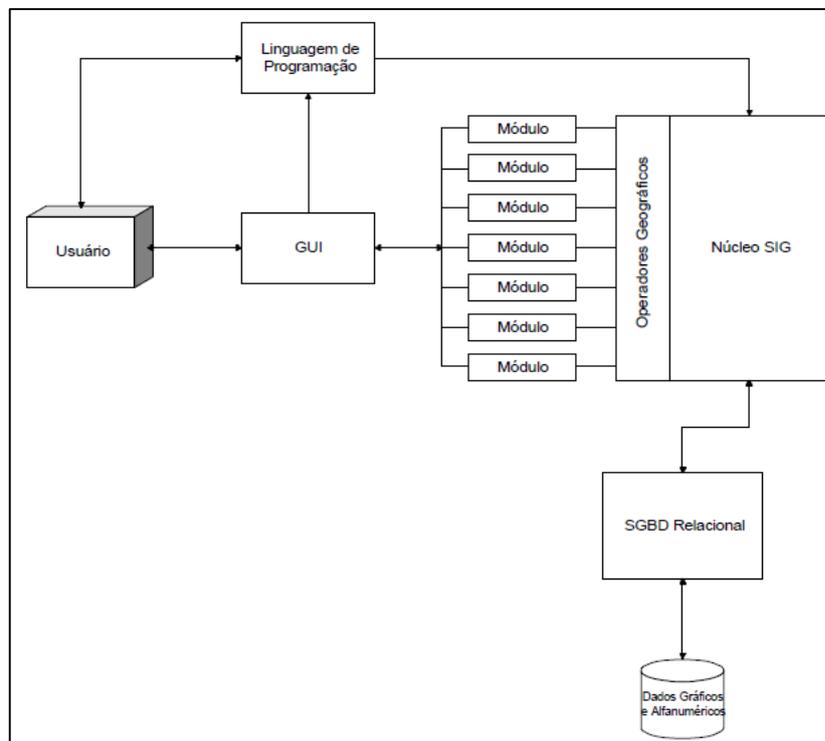
- Evitar os problemas de controle de integridade típicos do ambiente “desktop”, permitindo o acesso concorrente aos dados;
- Facilitar a integração com as bases corporativas já existentes, como sistemas legados, que já utilizam SGBDs relacionais.

⁴ São conjunto de classes para se trabalhar com objetos espaciais que são utilizadas no desenvolvimento de softwares que permitem visualização de dados espaciais ou para construção de SIGs.

Dentre os SGBDs relacionais com capacidades de armazenamento de dados geográficos, estão o ORACLE, PostgreSQL e MySQL. Dessa forma nascem os SIGs com arquitetura relacional ou SIGs Relacionais.

Nesta arquitetura, em três camadas, os dados gráficos são organizados em tabelas, de forma semelhante aos dados alfanuméricos. Um sistema de chaves é utilizado para relacionar estas tabelas, formando um esquema relacional cuja integridade é garantida pelo SGBD. A Figura 2.11 apresenta um diagrama da arquitetura de um SIG relacional.

Figura 2.11 - Arquitetura de um SIG Relacional.



Fonte: Câmara, Davis e Monteiro, 2001.

A interação do usuário com o sistema é através da GUI, um conjunto integrado de menus, barras de ferramentas e outros controles (camada de apresentação). A GUI permite o acesso às ferramentas nos vários módulos do SIG que realizam as operações geográficas. Este conjunto de ferramentas define os recursos ou funções que o software GIS tem disponível para realizar operações com dados geográficos (camada de lógica de negócios). Os dados são armazenados no banco de dados mantido pelo SGBD (camadas de servidor de dados).

Algumas características e vantagens desta arquitetura são:

- Dados gráficos e alfanuméricos armazenados de forma integrada no banco de dados relacional, externo ao SIG, evitando inconsistências nos bancos de dados geográficos;
- Implementação de recursos de geoprocessamento (operadores espaciais, ferramentas de análise, etc.) apoiadas no gerenciador relacional, e apresentadas como extensões ou complementações ao modelo relacional;
- Grande robustez de implementação, devido às garantias de integridade do esquema relacional;
- Grande estabilidade, devido ao avançado grau de desenvolvimento dos SGBD relacionais;
- Maior escalabilidade do sistema, possibilitando a realizações de tarefas com uso dos servidores em paralelo.

2.9 Sistema de Suporte à Decisão

Um Sistema de Suporte à Decisão (SSD) pode ser considerado como a integração todos os temas discutidos nos tópicos anteriores presentes neste capítulo. Viu-se que, em análise de sistemas o sistema do mundo real, o domínio da análise, é decomposto em subsistemas, componentes, elementos, partes ou entidades para facilitar a análise, modelagem e simulação.

No design e implementação de um SSD ocorre uma reintegração de componentes, os sistemas computacionais, os bancos de dados, as pessoas e organizações são integrados com a finalidade de validar conhecimentos adquiridos, auxiliando a tomada de decisão. É importante salientar que a integração e funcionamento destes componentes pode ser uma tarefa complexa, porém é essencial que a facilidade do uso desse sistema e sua fluidez sejam suas principais características.

O termo SSD teve origem no mundo dos negócios, foi utilizado para descrever uma nova classe de softwares computacionais que davam suporte aos executivos em tarefas não rotineiras e problemas não estruturados. Problemas não estruturados são aqueles para os quais não existem soluções através de algoritmos bem definidos e ocasiona não serem facilmente tratáveis por computador (Vieira, 2007). Os SSDs quebraram esse paradigma, pois possibilitam uma estreita interação entre homem e máquina.

Há algumas definições para um SSD na literatura. Ele pode ser definido como um tipo de SI que é interativo e auxilia na tomada de decisão a respeito de uma determinada questão ou problema não estruturado (Stair, 1998; Turban, 1995; Turban e Aronson, 2001). Alguns autores

o considera um sistema computacional para auxílio na tomada de decisão (Grigg, 1996; Finlay, 1994; Porto e Azevedo, 2002; Adelman, 1992).

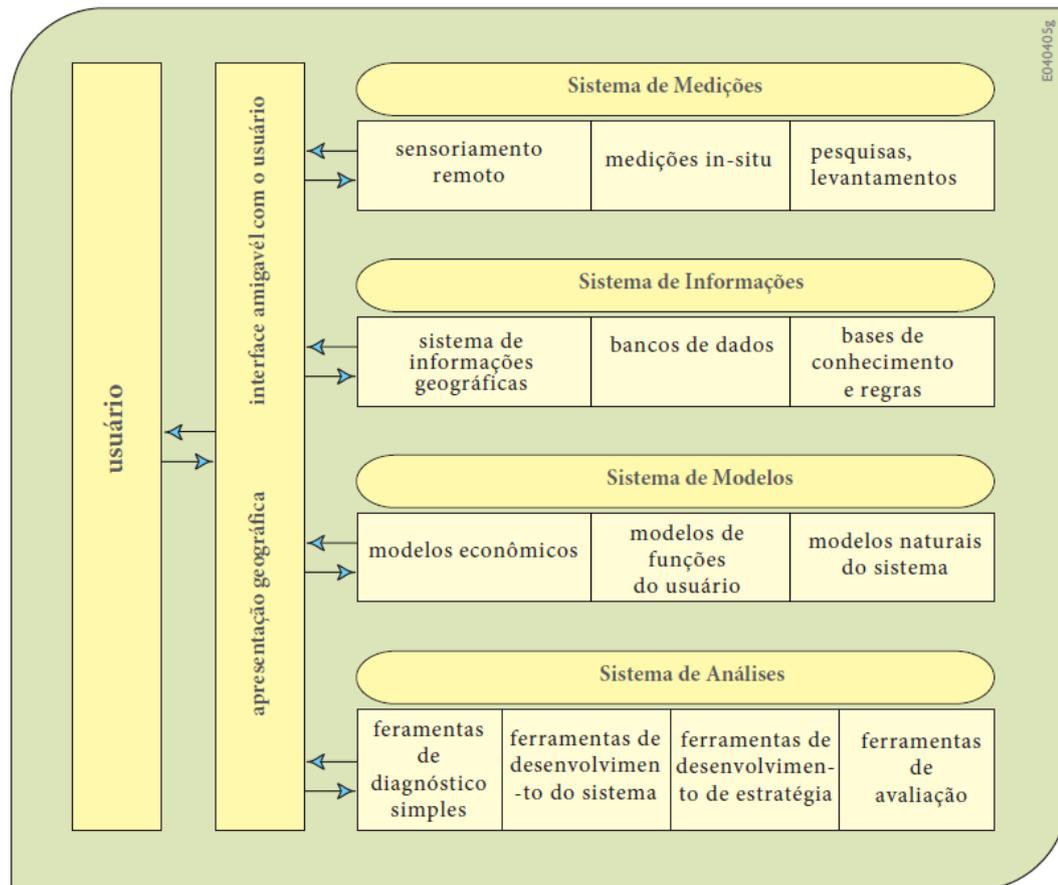
Mais recentemente alguns autores trazem uma visão mais abrangente de um SSD, como uma coleção ou conjunto de pessoas (usuários, analistas, modeladores e tomadores de decisão), organizações, softwares computacionais, bancos de dados e outros dispositivos, organizados e funcionando integrados, de forma a dar suporte a tomada de decisão específica de um dado problema (Kersten, Mikolajuk e Yeh, 2000; Loucks e van Beek, 2005; Stair e Reynolds, 2014).

Essa definição holística que considera todos os componentes e agentes envolvidos em um sistema, ainda sim abstraindo características mais importantes do mundo real, tem sido o modelo mais recentemente indicado para suporte nas tomadas de decisão, podendo ser considerado como SSDs orientados ao desenvolvimento sustentável, pois incorporam aspectos sociais, econômicos e ambientais.

2.9.1 Aplicações em Recursos Hídricos

Os SSDs tiveram suas primeiras aplicações em recursos hídricos a partir dos anos 1980s, Loucks, French e Taylor (1985) discutiram o design e implementação de um SSD por meio de modelos hidrológicos interativos. Desde então é verificado diversas aplicações do uso de SSDs para auxiliar a resolução de problemas no planejamento e gerenciamento de recursos hídricos (Loucks e da Costa, 1991; Watkins e McKinney, 1995; Loucks e Gladwell, 1999; Simonovic, 2000). A Figura 2.12 apresenta os componentes geralmente presentes em um SSD na gestão de recursos hídricos.

Figura 2.12 - Componentes de um SSD.



Fonte: Adaptado de Loucks e van Beek, 2005 (tradução).

Koutsoyiannis *et al.* (2003) desenvolveram um SSD para o gerenciamento de recursos hídricos na Grécia, o sistema se mostrou leve no aspecto de processamento quando utilizado o módulo de modelos estocásticos de simulação aplicado a um complexo sistema hídrico em Atenas. O sistema foi desenvolvido na linguagem *Object Pascal*, utilizou componentes GIS da ESRI e banco de dados Oracle.

Dawoud (2010) apresenta o desenvolvimento de um sistema de informações para gestão integrada de recursos hídricos com foco em regiões áridas, o AWRIS, desenvolvido para Abu Dhabi. O software tem seu núcleo baseado em SIG e interface web para visualização de dados.

Um sistema foi proposto por Abramson *et al.* (2014) para dar suporte na tomada de decisões relacionadas ao acesso de água a comunidades remotas e com baixo poder aquisitivo.

Magiera e Froelich (2014) propuseram um sistema de suporte para aperfeiçoar o uso da água de forma racional em municípios com residências familiares urbanas. Arsene, Gabrys e

Al-Dabass (2012) desenvolveram um sistema de apoio à decisão para os sistemas de distribuição de água baseado em redes neurais artificiais e teoria dos grafos com foco na detecção de perdas.

Zeng *et al.* (2012) desenvolveram um sistema para dar suporte a gestão integrada de recursos hídricos baseado na web, conhecido como *Web-based Decision Support System*. O sistema possui diversos módulos integrados de modelos matemáticos e probabilísticos, SIG, entre outros.

Giupponi e Sgobbi (2013) relatam experiências no uso de SSD em países em desenvolvimento, em especial a África, sendo constatado que maiores investimentos e esforços devem ser aplicados não nos aspectos tecnológicos do sistema, mas no sistema institucional-administrativo daquele país.

Uma gama de softwares e pacotes comerciais também tem sido apresentados por diversos autores para diversos segmentos na gestão de recurso hídricos (Ahmad e Simonovic, 2001; Paredes, Andreu e Solera, 2010; Pierleoni *et al.*, 2014).

De forma geral na literatura os SSDs não possuem todos os componentes apresentados na Figura 2.12, principalmente o componente SIG, isso se deu principalmente pela questão abordada anteriormente sobre os SGBDs não armazenarem inicialmente em seu banco os atributos e geometria dos dados espaciais, porém a interface visual e amigável com os usuários, ou tentativa dela, está sempre presente nos SSDs.

Departamentos de diversas instituições também tem desenvolvido SSDs, alguns com licença livre, outros proprietários e ainda como extensões para softwares livres e proprietários. O modelo SWAT (*Soil & Water Assessment Tool*) de avaliação ambiental de bacias hidrográficas pode ser acoplado ao software proprietário ArcGIS por meio da extensão ArcSWAT, incorporando assim facilidades de um SIG. A Tabela 2.2 traz os principais softwares para aplicações em recursos hídricos com uma breve descrição e seu tipo de licença.

Tabela 2.2 - Softwares para aplicações em recursos hídricos.

Nome	Descrição	Tipo de Licença
MODSIM	MODSIM-DSS é estruturado como um Sistema de Apoio à Decisão, com uma interface gráfica do usuário (GUI), que permite aos usuários criar qualquer topologia de sistema de bacias hidrográficas.	Livre
WaterWare	WaterWare é implementado em uma arquitetura aberta orientada a objeto cliente-servidor, totalmente habilitado para web e baseado na Internet, apoiando a integração de bancos de dados, GIS, modelos de simulação e otimização e ferramentas analíticas de fácil utilização.	Proprietário
WEAP	É uma ferramenta de software de fácil utilização que faz uma abordagem integrada de planejamento de recursos hídricos.	Proprietário
AQUARIUS	Aquírio é um conjunto de ferramentas de análise, em vez de um único modelo dedicado para alocação de água. O modelo foi implementado utilizando uma programação orientada a objetos (OOP) linguagem (C++).	Livre
AcquaNet	AcquaNet é um modelo integrado para análise de sistemas complexos em recursos hídricos. É constituído por um módulo base, responsável pelo traçado e pela integração entre os seguintes módulos: (1) alocação de água; (2) qualidade da água; (3) irrigação; (4) produção de energia; (5) análise econômica para alocação e (6) CAR - curvas de aversão a risco.	Livre
SWMM	SWMM é um modelo de simulação da qualidade da água dinâmico hidráulico-hidrológico. Ele é usado para um evento único ou a longo prazo de simulação (contínua) da quantidade de escoamento superficial e qualidade das áreas essencialmente urbanas.	Livre
MIKE	Conjunto de produtos para aplicações de recursos hídricos. Os produtos trabalham individualmente ou em conjunto para resolver problemas de recursos hídricos em aplicações de sistemas complexos e integrados.	Proprietário
Aquaveo	Estado da arte do software de modelagem ambiental especializado em águas subterrâneas, bacias hidrográficas, e modelagem de águas superficiais.	Proprietário

Fonte: Autor.

No Brasil há algumas iniciativas de desenvolvimento e alguns softwares bem utilizados como é o caso do AcquaNet desenvolvido pelo LabSid - USP (Laboratório de Sistemas de Suporte a Decisão – Universidade de São Paulo).

O Grupo de Otimização Total da Água (GOTA) vem realizando pesquisas relativas ao desenvolvimento de ferramentas para otimizar a alocação da água para múltiplos usos, em termos quantitativos e qualitativos. Foram desenvolvidas ferramentas computacionais no ambiente MatLab, o *Optimal Reservoir Network Analysis Program* (ORNAP) (Curi e Curi, 2001b; Santos *et al.*, 2011) e o *Cropping and Irrigation System Design with Reservoir and Groundwater Operation* (CISDERGO) (Curi e Curi, 2001a; Alencar, 2009).

A integração destes modelos de simulação e otimização, constituindo um subsistema de modelos, aos outros subsistemas que compõe um SSD passará a se chamar de *Riverbasin Information and Operation Support System* (RIOSS). Além disso, estudos sobre outorga de água, baseado em métodos multiobjetivos e multicriteriais, para incluir aspectos sócio-econômicos, físicos, ambientais e técnico-operacionais, também foram desenvolvidos (Rodrigues, 2007; Santos *et. al*, 2011).

Mota *et. al* (2011) iniciaram o desenvolvimento da interface para acesso ao RIOSS via web com alguns módulos, a qual vem sido aperfeiçoada (Curi e Leal, 2014). O RIOSS necessita de interface desktop para acesso em rede e um servidor de aplicações que possa compartilhar serviços para serem utilizados pelas interfaces web e desktop como acesso aos modelos, visualização espacial e aos sistemas de medições e informações. A arquitetura e estrutura do SSD proposto, o qual o desenvolvimento será abordado no próximo capítulo, também tem como objetivo abarcar essas necessidades.

Capítulo 3

3 Metodologia

Neste capítulo serão descritas as etapas metodológicas para o desenvolvimento do SSD, como o modelo de desenvolvimento de software escolhido (prototipagem), linguagens de programação utilizadas e a arquitetura do sistema, esta última de grande importância na estrutura do sistema, pois ditará muitas etapas no processo de modelagem do SSD. Foram também descritos a abordagem utilizada para a construção do sistema de modelos e o processo de desenvolvimento de aplicativos para acesso a dados e bancos de dados externos. Esta etapa de desenvolvimento, que também incluiu o planejamento e definição dos requisitos do sistema, certamente é o estágio mais importante e oneroso nas etapas de construção do SSD.

3.1 Arquitetura do Sistema

A escolha da arquitetura ou ainda o padrão de arquitetura do sistema, ocorre na fase do processo de criação e design, a qual se decide a organização do sistema de software para atender a requisitos funcionais e não funcionais do sistema modelado do domínio do problema. O processo de desenvolvimento de um sistema software é fortemente afetado por alguma mudança em sua arquitetura.

A arquitetura de um sistema de software pode ser baseada num padrão ou estilo particular de arquitetura (Sommerville, 2010). Cada padrão de arquitetura tem sua utilização indicada e de certa forma foi criada para resolver problemas e atender novas necessidades.

Para o sistema proposto pensou-se numa arquitetura genérica que pudesse atender a diversos tipos de usuários, instituições, necessidades e ao mesmo tempo incorporar novas facilidades tecnológicas. A arquitetura que atende essas necessidades é a cliente-servidor múltiplas camadas ou em inglês *n-tier/multi-tier client-server*. Nesta arquitetura, detalhada no capítulo anterior, as diferentes camadas do sistema, ou seja, apresentação, lógica de negócios (processamento das aplicações), gerenciamento de dados, são processos separados que podem executar em diferentes processadores.

Outro padrão de arquitetura que foi utilizado no desenvolvimento do SSD foi o MVC (*Model-View-Controller*). No padrão MVC diferentes partes do código do sistema são responsáveis por representar o *Model* (Modelo), *View* (Visão) e *Controller* (Controlador) da

aplicação, que equivalem ao gerenciamento de dados, apresentação dos dados e lógica de funcionamento do sistema, respectivamente (McGovern, 2004; Lobo, 2008).

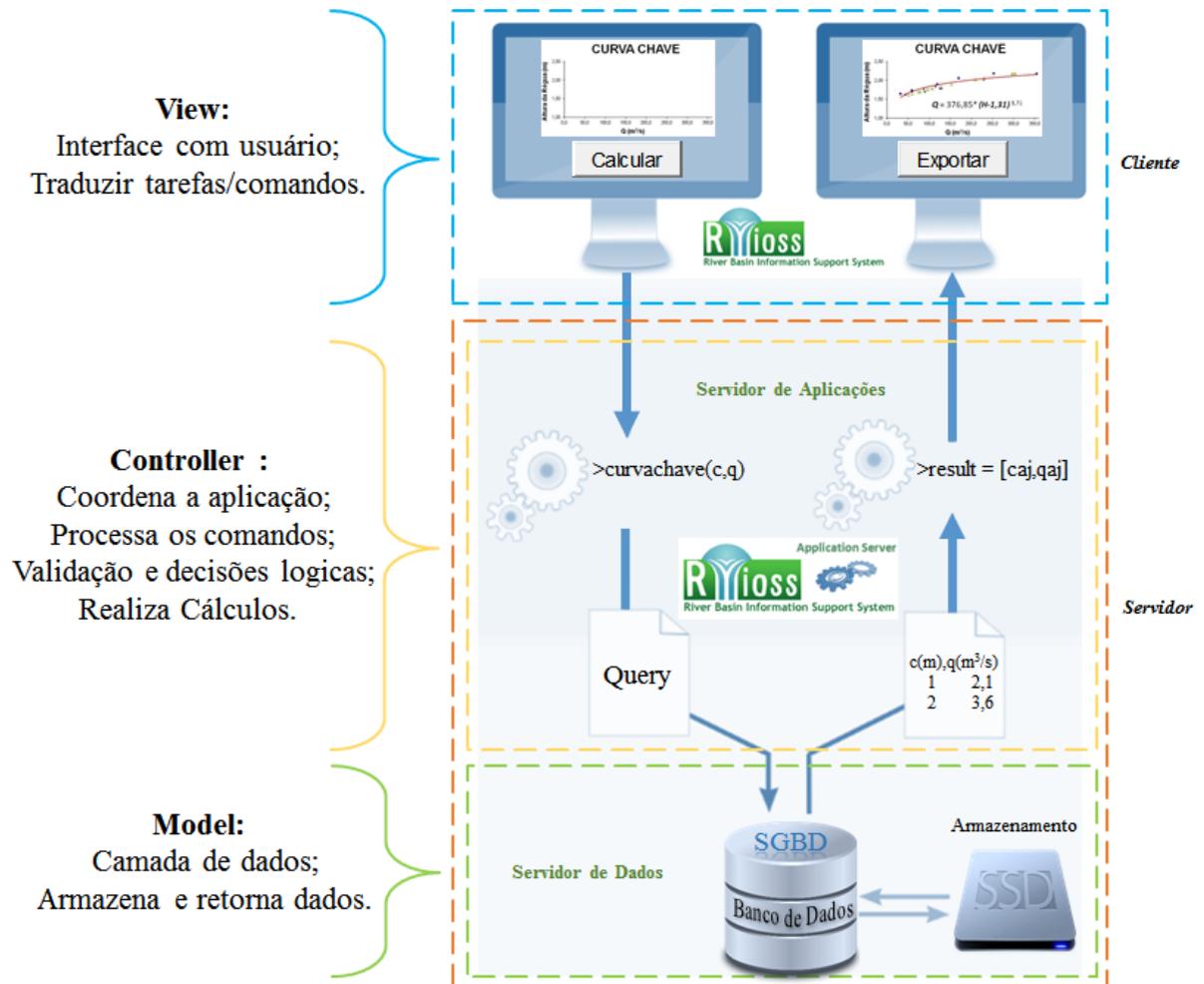
O conceito do padrão MVC à primeira vista pode se assemelhar muito ao padrão *n*-camadas, porém ele diz respeito a distribuição lógica e conceitual do sistema. Padrão *n*-camadas conceitua a separação física e de comunicação do sistema. No padrão *n*-camadas a camada cliente nunca acessa diretamente a camada de dados sempre passando pela camada de lógica de negócios, ou as vezes chamada de *middleware*. Dessa forma, o fluxo de mensagens se dá de forma unidirecional entre as camadas.

No padrão MVC o *View* pode receber atualizações diretamente do *Model* fazendo o fluxo de mensagens se tornar triangular, isso pode ocorrer lançando mão de um padrão de projeto chamado *observer*. Existem algumas variações da arquitetura MVC (Otero, 2012).

De forma geral os dois padrões têm como princípio o conceito de *separation of concerns* (separação de conceitos), dar as respectivas funções a cada camada, e são utilizados de forma conjunta e em harmonia constantemente no desenvolvimento em camadas.

A Figura 3.1 ilustra o fluxo de processamento de operações do sistema.

Figura 3.1 - Fluxo de processamento de operações do SSD.



Fonte: Autor.

Na Figura 3.1 o usuário interage com o sistema por meio da camada de apresentação (*View*), ao acionar no botão “Calcular” ele envia uma mensagem para a camada onde está a lógica de negócios (*Controller*) no lado *Server* e neste caso no servidor de aplicação. A mensagem é processada através do método da classe responsável, que é invocado. Neste é realizada uma *Query* (consulta) por meio da linguagem SQL solicitando, por exemplo, dados de cota e vazão de uma série armazenada no banco de dados para realização dos cálculos para gerar a curva-chave.

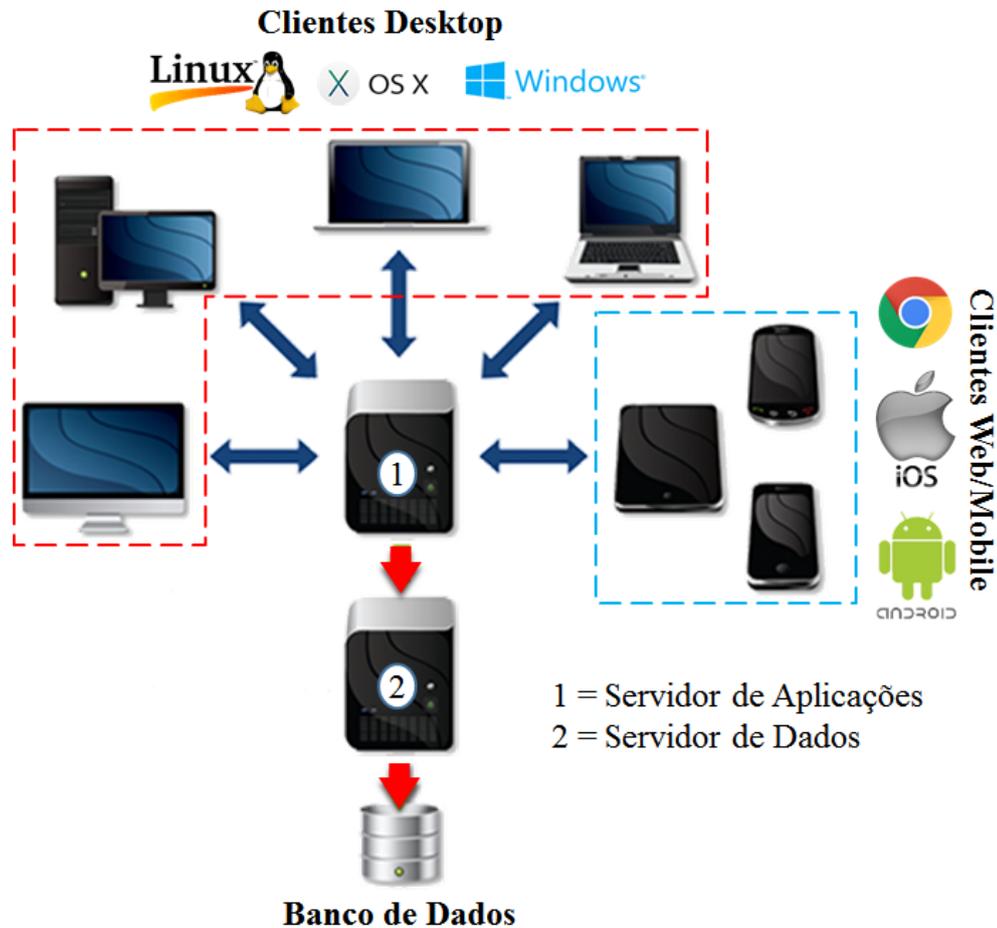
A consulta é recebida pela camada de gerenciamento de dados (*Model*) localizada fisicamente no servidor de dados. O SGDB trata o código, recupera os dados de cota e vazão solicitados e os retorna para a camada de negócios, aonde acontece o processamento gerando os valores de cota e vazão ajustados para produção do gráfico que é visualizado pelo usuário na camada de apresentação.

O SSD proposto tem camadas de apresentação desktop e web e acesso em rede via LAN (*Local Area Network*) e WAN (*Wide Area Network*). Sistemas baseados na web tradicionalmente funcionam com arquitetura em camadas. Como foi visto nesta arquitetura a camada de lógica de negócios pode estar ligada a diferentes camadas de apresentação de vários usuários (desktop, web ou mobile) que podem possuir sistemas operacionais diferentes (Windows, Linux ou OS X). Outra vantagem dessa arquitetura é a escalabilidade, pode-se adicionar servidores dedicados conforme a necessidade, como por exemplo um servidor para web, um para aplicações e um de mapas geográficos.

A separação em camadas ainda possibilita que vários usuários interajam com o sistema sem ter necessidade de ter todas as camadas em suas máquinas, tornando o software mais flexível, possibilitando que cada camada seja acessada e modificada individualmente sem ter que modificar os outros módulos do software. É importante salientar que todas as camadas podem estar localizadas no computador cliente sendo o servidor neste caso o próprio computador cliente (*localhost*), o que é comum na fase de desenvolvimento do sistema.

A Figura 3.2 apresenta a estrutura do SSD na arquitetura cliente-servidor múltiplas camadas.

Figura 3.2 - Arquitetura do SSD.



Fonte: Autor.

Na Figura 3.2 os diferentes clientes acessam o servidor de aplicações, que inclui o servidor web, de mapas e os modelos. O servidor de aplicações se conecta ao servidor de dados onde está o SGDB. O servidor de aplicações pode ser particionado fisicamente, o que deve acontecer quando o sistema estiver em seu funcionamento pleno, onde os modelos podem estar em um servidor dedicado e mais robusto possibilitando melhorar a performance no uso de modelos de otimização, por exemplo.

Como visto anteriormente, os aplicativos que estão no lado cliente contendo apenas a camada apresentação podem ser programados para diferentes sistemas operacionais o que é uma necessidade atual. Instituições federais, por exemplo, há alguns anos estão em constante transição para o uso de sistemas operacionais livres (Ubuntu, Mint, etc.). Também ocorre o crescimento de outras marcas no mercado brasileiro como a Apple com o sistema operacional OS X, mas que ainda é dominado pelo sistema da Microsoft, o Windows.

Os clientes desktop também podem acessar o SSD por meio da web, como também é o caso dos clientes mobile, porém pensando no sistema de medições do SSD, os dispositivos portáteis também podem conter uma aplicação para ser utilizada como dispositivo de medição. Em alguns casos em lugares remotos onde não possui acesso à internet, eles podem armazenar dados temporariamente para serem enviados ao banco de dados quando entrarem em área com acesso à rede. Neste trabalho será abordado o desenvolvimento do lado cliente desktop, porém o lado servidor foi desenvolvido pensando no acesso nos dois outros tipos de clientes citados.

3.2 Processo de Desenvolvimento do Sistema

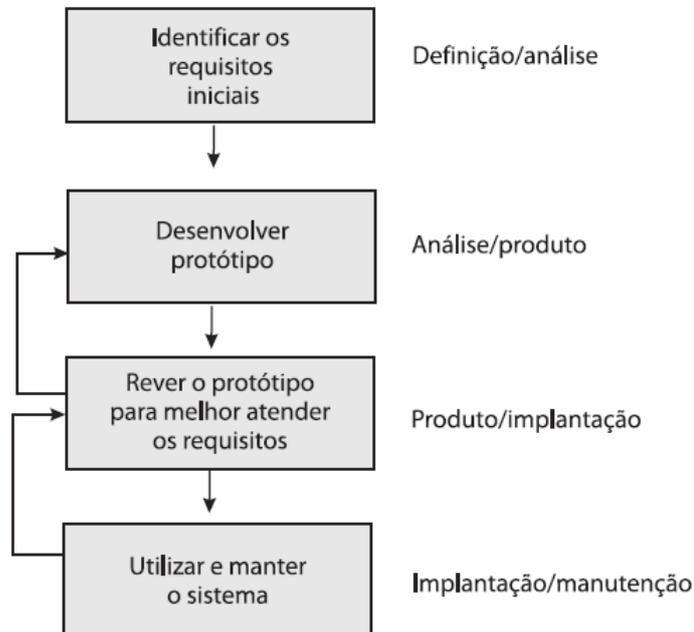
Devido à natureza semiestruturada ou não estruturada de problemas abordados pelos sistemas de apoio à decisão, é bastante improvável que os gerentes e desenvolvedores de um SSD tenham um entendimento completo do problema de decisão (Turban, Aronson e Liang, 2007). Este fato é visível em sistema de recursos hídricos onde constantemente ocorrem alterações, sejam elas estruturais, políticas ou ambientais. Dessa forma um SSD para sistemas de recursos hídricos deve primar pelas características de ser aberto, expansível e adaptativo.

Estas alterações influenciam no comportamento do sistema, incorporando novas variáveis ao processo de modelagem e novas regras de operação, fazendo analogia ao software, novas variáveis podem equivaler a alterações no banco de dados e novas regras de operação a modificação na lógica/regras de negócio, assim é necessário utilizar-se de metodologias de desenvolvimento de sistemas de software adequadas. Ressalta-se a importância mais uma vez da escolha da arquitetura do sistema (em camadas) e do método de análise (orientado a objetos).

Assim, a maioria dos SSDs são desenvolvidos através do processo de prototipagem, também conhecido como projeto iterativo ou desenvolvimento evolutivo (Turban, Aronson e Liang, 2007). Prototipagem, segundo O'Brien (2004), é o rápido desenvolvimento e teste de modelos de funcionamento ou protótipos de novas aplicações em um processo interativo e repetitivo que envolve tanto especialistas técnicos como usuários finais. O sistema protótipo é repetidamente aperfeiçoado até se tornar adequado.

Essa forma de desenvolvimento vai de encontro com a visão holística dos SSDs que é conhecida como abordagem sociotécnica. A Figura 3.3 apresenta as etapas do desenvolvimento do SSD por meio do processo de prototipagem.

Figura 3.3 - Etapas no processo de prototipagem.



Fonte: Mülber e Ayres, 2005.

3.2 Sistema de Gerenciamento para o Banco de Dados

O banco de dados do SSD proposto terá que armazenar vários tipos de dados, arquivos em geral (pdf, imagens, vídeos, etc.), dados alfanuméricos e dados espaciais. Dessa forma, é necessário um SGBD que suporte todos estes modelos de dados. Ainda é necessário um SGDB que tenha boa performance para sistemas desktop e web, grande portabilidade e compatibilidade.

O sistema de gerenciamento de banco de dados MySQL supre estas necessidades e é atualmente um dos SGBDs relacionais mais utilizados no mundo. O MySQL era mantido pela empresa Sun Microsystems até ser comprado pela empresa Oracle que disponibiliza-o em versões comercial e gratuita.

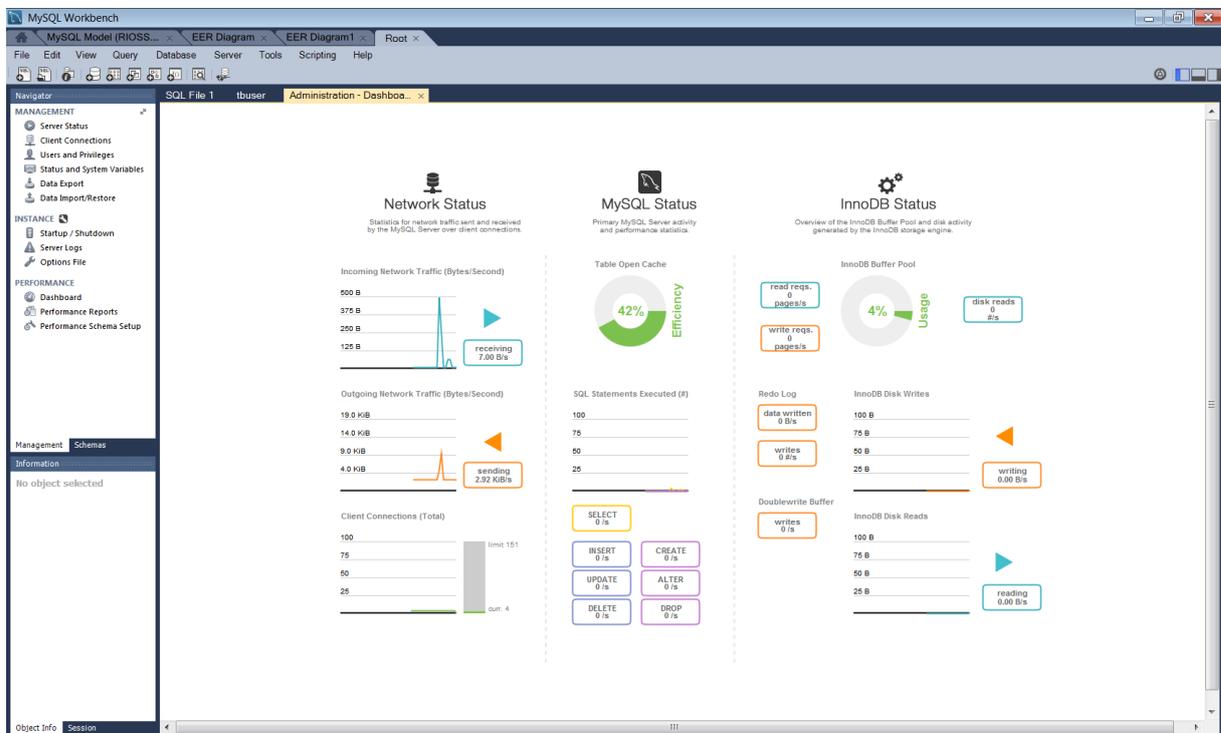
A Oracle disponibiliza drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, C#, Visual Basic, Python, Perl, PHP, ASP e Ruby, além de ser multi-plataforma suportado por praticamente todos os sistemas operacionais atuais.

A versão gratuita do SGDB é a MySQL *Community Edition* que é mantida pela licença GPL (*General Public Licence*). Ela atualmente supre as necessidades de sistemas de grande a pequeno porte. Sua compatibilidade com dados espaciais possibilita a integridade dos mesmos

centralizando o banco de dados, sendo este um fator muito positivo, o qual foi comentado no capítulo anterior com relação aos SIGs relacionais.

O pacote MySQL on Windows versão 5.6.23 fornece um conjunto de ferramentas para desenvolvimento e gerenciamento de aplicações baseadas em MySQL no Windows. Uma dessas ferramentas é o MySQL Workbench que fornece um Ambiente Integrado Desenvolvimento (IDE - *Integrated Development Environment*) com ferramentas para design de banco de dados, administração, desenvolvimento e migração de banco de dados SQL. A Figura 3.4 apresenta uma imagem da IDE do MySQL Workbench.

Figura 3.4 - IDE do MySQL Workbench.



Fonte: Captura de tela em sistema Windows.

Apesar do SSD proposto ser modelado no paradigma orientado a objetos, que será descrito no tópico seguinte, um SGBD relacional foi escolhido. Um SGBD relacional trabalha com dados estruturados de forma de tabelas com atributos e relacionamentos, a manipulação dos dados se dá a partir de comandos da linguagem SQL (*Structured Query Language*), já um sistema orientado a objetos trabalha com classes, objetos e seus comportamentos. Isto causa uma diferença que é conhecida como impedância objeto-relacional.

Algumas razões motivaram a não escolha de um SGBD orientado a objetos, são elas:

- Poucos recursos de ferramentas gráficas para desenvolvimento;

- Linguagens para consultas de objetos são difíceis e pouco padronizadas;
- Pouco explorado e pouca documentação, pois é uma tecnologia relativamente nova.

Existem algumas técnicas e abordagens para preservar as características da orientação a objetos no desenvolvimento de um sistema orientado a objetos que se utiliza de um SGBD relacional, alguns destas técnicas foram testadas e utilizadas durante o desenvolvimento do SSD, as quais serão descritas nos próximos tópicos.

3.3 Modelagem do SSD

3.3.1 Técnicas de Programação

A técnica de análise e desenvolvimento utilizada para modelagem do SSD proposto foi a orientada a objetos. Entre algumas vantagens, citadas no capítulo anterior, do uso da orientação a objetos, a reutilização de código é uma das fortes características na Programação Orientada a Objetos (POO).

Outra grande vantagem do desenvolvimento orientado a objetos diz respeito a manutenção de código. Como a representação do sistema se aproxima do mundo real, o entendimento do sistema de software como um todo e seus componentes individualmente é mais simples, fazendo, por exemplo, que o trabalho em equipe se torne mais fácil. As bibliotecas estão disponíveis em forma de classes, o que também facilita a reutilização.

Nos dias atuais dificilmente se programa um sistema utilizando apenas uma linguagem de programação, em especial sistemas que permitem acesso via internet. Um SSD se utiliza de diversas tecnologias para suprir necessidades dos seus diversos componentes e para resolver problemas específicos que em uma determinada linguagem é mais viável.

O núcleo do SSD proposto foi desenvolvido na linguagem orientada a objetos conhecida antigamente como Object Pascal, que após ser mantida, reformulada e atualizada por diversas empresas hoje é conhecida oficialmente como Delphi que também é o nome da IDE mais utilizada desta linguagem.

Nos últimos anos a Embarcadero, empresa que mantém a linguagem atualmente, tem feito melhorias na linguagem incorporando o que há de mais moderno das linguagens orientadas a objetos como RTTI (*Run-Time Type Information*), *Custom Attributes* e *Generics*. Outro grande avanço incorporado no Delphi foi a possibilidade de compilar com uma mesma

linguagem aplicativos para diferentes dispositivos com diferentes plataformas. Porém, a IDE do Delphi só suporta a plataforma Windows. Existem ainda outras IDEs como FreePascal e Kylix (descontinuada) livres que suportam plataformas baseadas no Linux e compilam aplicativos para o sistema Linux.

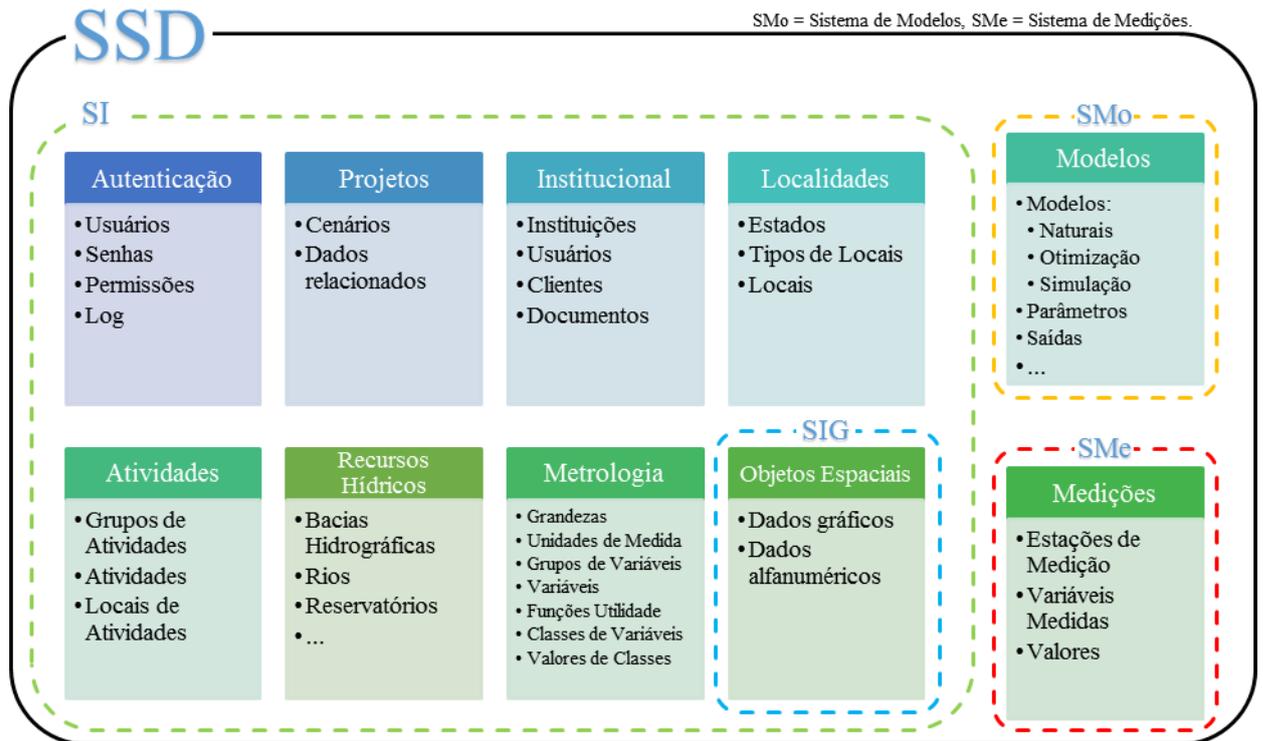
Foram ainda utilizadas linguagens de marcação como HTML (*Hyper Text Markup Language*) e XML (*Extensible Markup Language*). O CSS (*Cascading Style Sheets*) foi utilizado para definir a apresentação de documentos escritos nas linguagens de marcação.

Outras tecnologias usadas principalmente no SIG foram: JavaScript, que permite rodar scripts que ficam embutidos nos documentos HTML e pode fornecer níveis de interatividade para páginas Web que não são acessíveis com um HTML simples; e PHP que é uma linguagem de script interpretada do lado do servidor de código aberto, que atualmente é amplamente utilizado no desenvolvimento web.

3.3.2 Modelagem do Banco de Dados

Procurou-se modelar o banco de dados considerando o máximo de aspectos inerentes aos subsistemas presentes em um sistema de recursos hídricos de forma a possibilitar a boa caracterização dos problemas geralmente envolvidos. Da mesma forma buscou-se compatibilizar aspectos relacionados aos modelos utilizados em recursos hídricos, tanto os dados de entrada como armazenar suas saídas. A Figura 3.5 apresenta a estrutura do banco de dados e componentes do SSD.

Figura 3.5 - Estrutura do banco de dados e componentes do SSD.



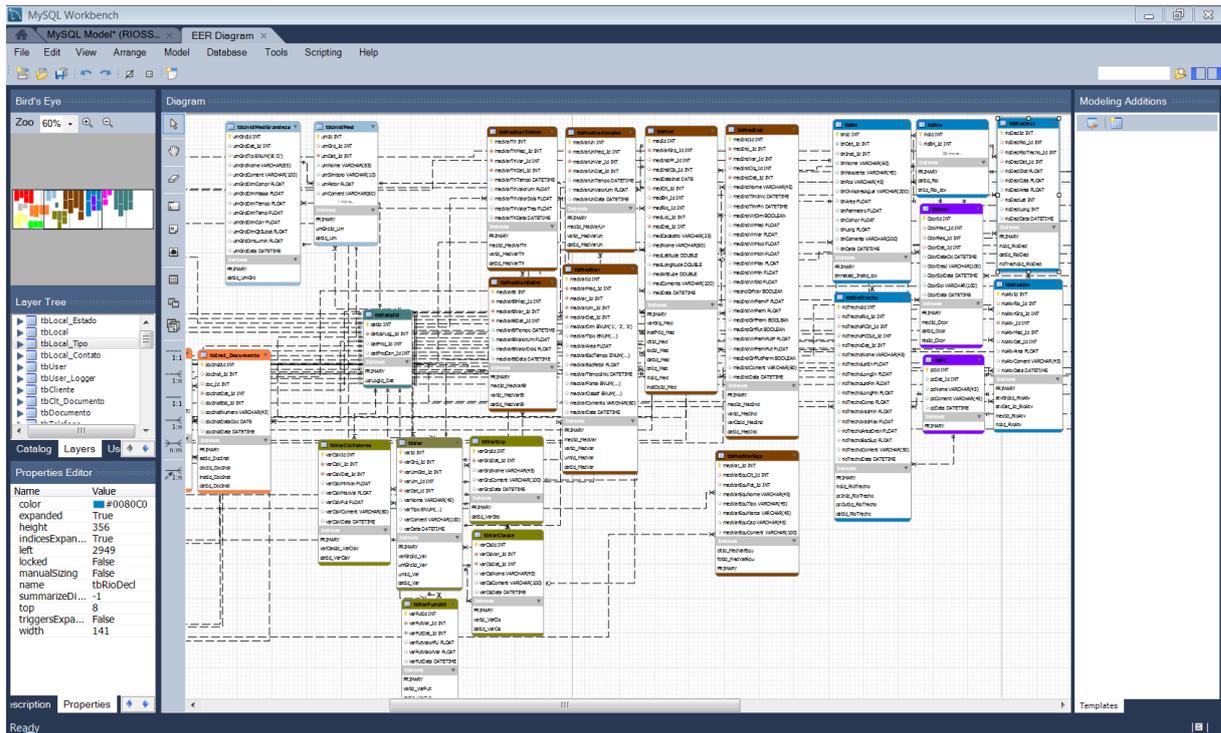
Fonte: Autor.

Na Figura 3.5 cada caixa denominada representa um conjunto de tabelas relacionadas que compõem cada componente ou subsistema do SSD, tabelas estas não só apenas relacionadas dentro do componente que as pertence, mas com relacionamentos com os demais componentes. Neste trabalho será abordado o desenvolvimento dos quatro subsistemas apresentados na Figura 3.5, que serão base para construção do subsistema de análise (SA) do SSD que fornecerá em si informações para tomada de decisão.

A modelagem e design do banco de dados foi realizada no MySQL Workbench. O programa permite a criação de modelos por meio de diagramas EER (*Enhanced Entity–Relationship*) que posteriormente cria os *schemas* físicos no banco de dados utilizando a função *Forward Engineer*. O MySQL Workbench também possui função de engenharia reversa, a *Reverse Engineer*, que permitir realizar o processo inverso possibilitando a importação de bancos de dados já criados.

A Figura 3.6 apresenta o ambiente onde é criado o modelo de banco de dados no MySQL Workbench.

Figura 3.6 - Modelagem do banco de dados utilizando diagramas EER no MySQL Workbench.



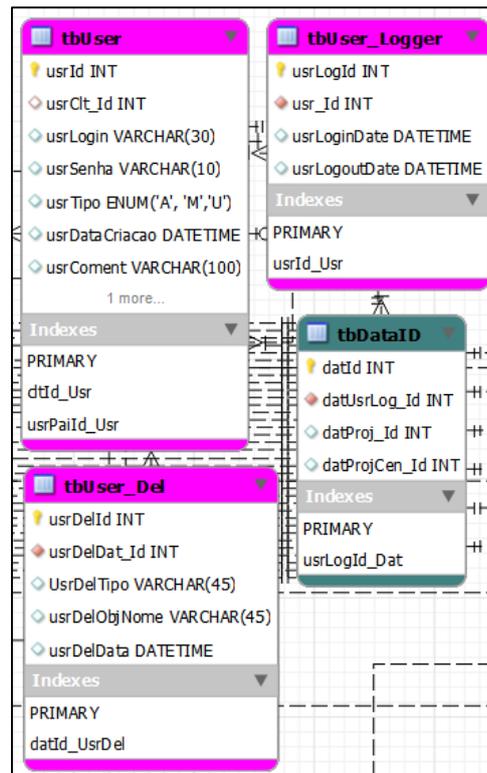
Fonte: Captura de tela em sistema Windows.

O processo de modelagem de banco de dados é um dos processos, se não o mais, oneroso no desenvolvimento de um sistema, é necessário criar um banco consistente obedecendo as regras dos bancos de dados relacionais e, ao mesmo tempo, visar o processo de implementação. Também é preciso ter um bom conhecimento do sistema modelado, portanto, a curva de desenvolvimento do banco de dados se tornou menos íngreme à medida que foram incorporados experiências anteriores nos mais de 10 anos de pesquisa realizadas pelo grupo GOTA.

Ao final da primeira modelagem o banco de dados totalizou 50 tabelas, porém esse número aumentou na medida do desenvolvimento do SSD. Foram incluídas novas tabelas para atender novos requisitos, o que foi possível por meio do processo iterativo de prototipagem. No decorrer do ciclo de vida de um sistema pode haver novas inclusões de tabelas e talvez mudanças de relacionamento.

A Figura 3.7 apresenta diagramas EER do grupo de tabelas responsáveis pelo armazenamento de dados referentes ao sistema de autenticação e *logger* do SSD.

Figura 3.7 - Diagramas EER do grupo de tabelas referentes a autenticação e *logger*.



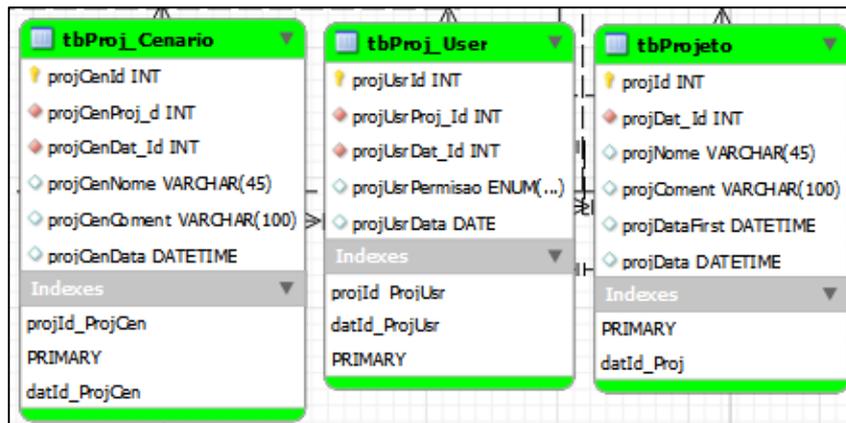
Fonte: Autor.

O grupo de tabelas apresentadas na Figura 3.7 é responsável por armazenar dados referentes ao sistema de autenticação e *logger* do SSD. O sistema tem uma conta padrão (RIOSS) que possui todos os privilégios para controle do sistema, essa conta pode criar outros usuários e escolher quais serão seus privilégios, privilégios estes que estão relacionados com o tipo de usuário que é representado pelo campo enumerado *usrTipo* (Administrador, Manutenção e Usuário Comum).

As tabelas *tbUser_Logger* e *tbUser_Del* armazenam dados referentes as operações realizadas pelo usuário que está conectado ao sistema, como: hora de entrada e saída e quais registros que foram deletados durante a sessão, se ocorrerem. A tabela *tbDataID* é responsável por relacionar cada inserção de dado no sistema a um identificador comum, ou seja, todas as tabelas do sistema possuem uma chave estrangeira para ela, assim cada novo registro possui o identificador da mesma da sessão em andamento no sistema. A mesma também possui os campos *datProj_Id* e *datProjCen_Id* para que possa associar os dados ao devido projeto e cenário vinculado.

A Figura 3.8 apresenta diagramas EER do grupo de tabelas responsáveis pelo armazenamento de dados referentes aos projetos e cenários para cadastro no SSD.

Figura 3.8 - Diagramas EER do grupo de tabelas referentes a projetos e cenários.

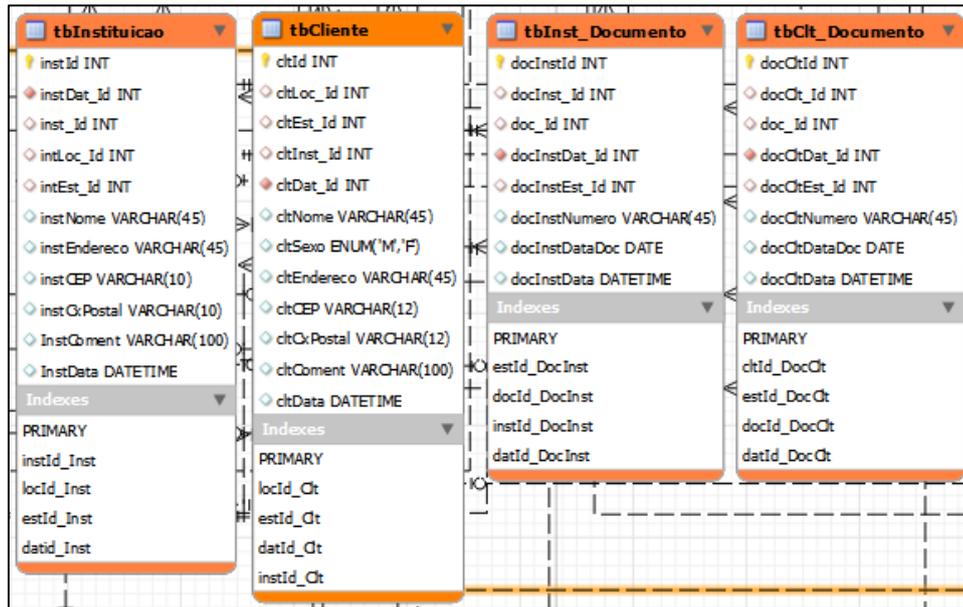


Fonte: Autor.

O grupo de tabelas apresentado na Figura 3.8 é responsável por armazenar dados sobre os projetos que podem ser criados no SSD, bem como os usuários que estão no projeto, seu nome e as permissões de cada usuário dentro do projeto. Pode-se criar Cenários dentro de um projeto para serem usados por modelos para simular condições do sistema, os cenários são armazenados na tbProj_Cenario. A tabela tbProjeto é o ponto de partida para iniciar as atividades no sistema, a partir do seu identificador será possível realizar a associação do projeto e outros dados referentes ao projeto, bem como carregar o projeto salvo a cada nova sessão no sistema.

A Figura 3.9 apresenta diagramas EER do grupo de tabelas responsáveis pelo armazenamento de dados referentes aos clientes, instituições e seus respectivos documentos para cadastro no SSD.

Figura 3.9 - Diagramas EER do grupo de tabelas referentes aos clientes, instituições e documentos.

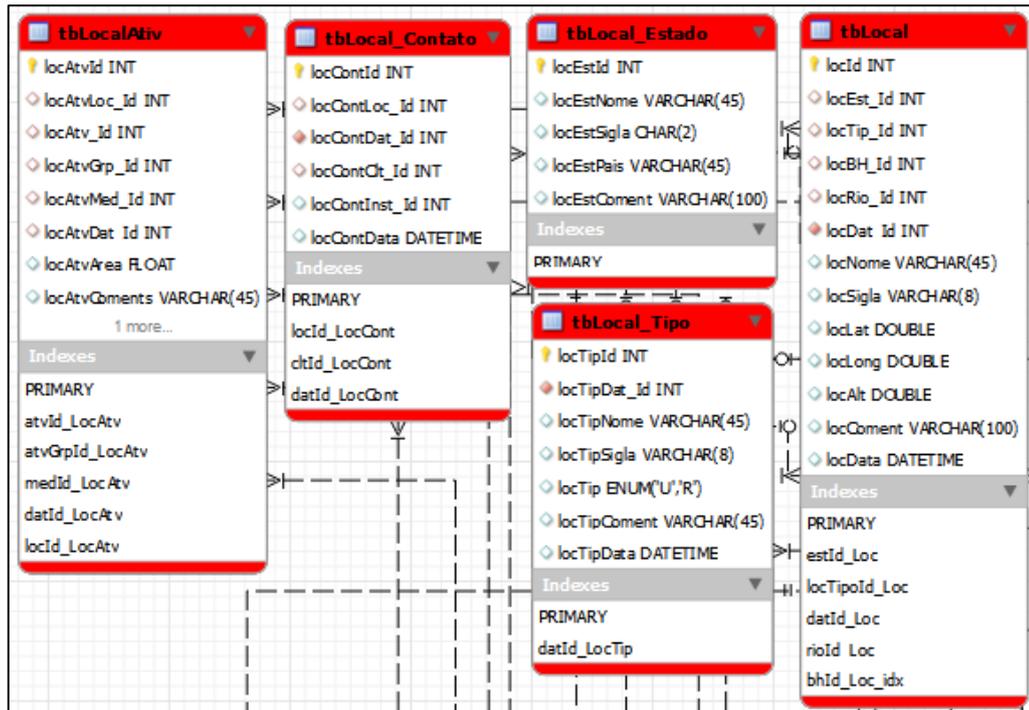


Fonte: Autor.

A tabela tbInstituicao apresentada na Figura 3.9 armazena dados de alguma instituição que faça parte do sistema hídrico ou de algum projeto cadastrado no SSD (universidades, agências de recursos hídricos, ONGs, etc). Já a tabela tbCliente refere-se ao representante institucional responsável aonde o SSD está sendo utilizado. As tabelas tbInst_Documento e tbClt_Documento referem-se aos respectivos documentos das instituições e clientes.

A Figura 3.10 apresenta diagramas EER do grupo de tabelas responsáveis pelo armazenamento de dados referentes as localidades para cadastro no SSD.

Figura 3.10 - Diagramas EER do grupo de tabelas referentes as localidades.



Fonte: Autor.

O conjunto de tabelas apresentadas na Figura 3.10 representa as possíveis localidades que podem ser cadastradas no SSD dentro de um projeto. Um local pode ser uma cidade ou outro lugar mais específico.

A tabela `tbLocal` tem diversos relacionamentos com outras tabelas, podendo armazenar dados como: em qual bacia hidrográfica aquele local está inserido, sua localização espacial, estado a que pertencente, entre outros. Já a tabela `tbLocalAtiv` armazena dados sobre atividades que estão ocorrendo em um determinado local possuindo um campo com chave estrangeira para tabela `tbLocal`.

As variáveis armazenadas no banco de dados serão estruturadas se utilizando de conceitos de metrologia. Está é uma característica muito importante do SSD, pois possibilitará o controle da inserção de dados e torna-o maleável para inserção e criação de novas variáveis. Novas variáveis poderão ser inseridas no sistema à medida que o planejador se depara com novos problemas sem a necessidade de programadores especializados e mudanças no sistema, mas apenas aproveitando-se da estrutura montada para o banco de dados.

Metrologia é a ciência que se preocupa com a criação, reprodução, conversão e transferência de unidades de medidas e as suas normas (Bewoor e Kulkarni, 2009). Em sistemas

de recursos hídricos geralmente as medições ocorrem por meio da instrumentação (hidrometria) de diversos componentes naturais, mensurando variáveis de diferentes grandezas que podem ser expressas em diversas unidades de medida. Há certas variáveis que não possuem grandezas já estabelecidas, são as grandezas derivadas, e são incluídas nos processos de modelagem à medida que o planejador conhece o problema mais detalhadamente ou quando novos ocorrem.

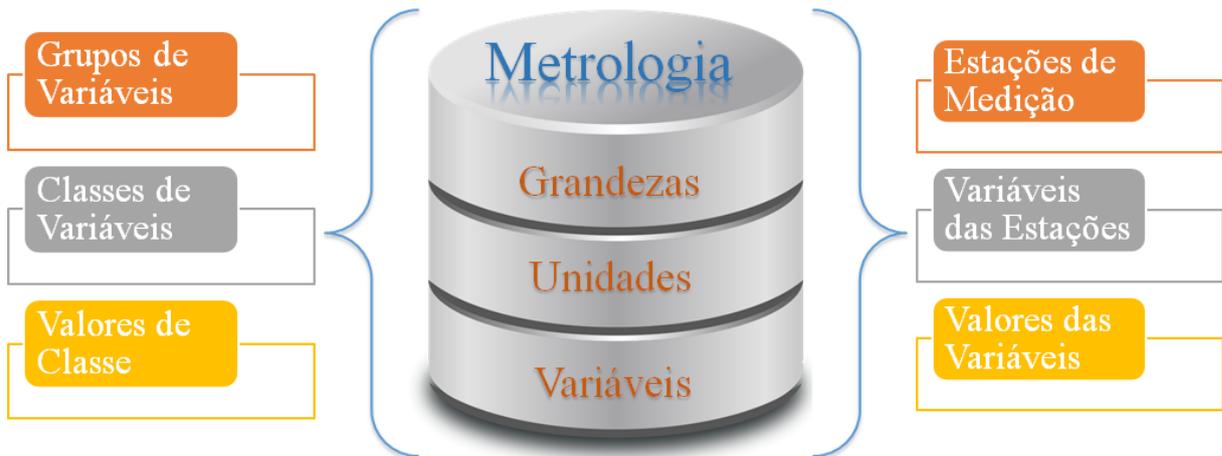
O número de grandezas derivadas importantes para a ciência e a tecnologia é seguramente ilimitado. Quando novas áreas científicas se desenvolvem, novas grandezas são introduzidas pelos pesquisadores, a fim de representarem as propriedades da área, e com essas novas grandezas vêm novas equações que se relacionam com grandezas familiares, e depois com as grandezas de base (INMETRO, 2012).

Por exemplo, planejadores se deparam com um problema de aumento de uma população animal (ex.: caramujo africano) em uma sub-bacia tornando-se uma peste. Eles utilizam então modelos de otimização para encontrar a menor quantidade de pesticida utilizado para diminuir esta população de animais, considerando a área da sub-bacia entre outras variáveis, a fim de minimizar custos e danos ambientais. A variável de decisão neste modelo pode ser modelada facilmente em um banco de dados num sistema que permita a inclusão de novas variáveis e que possua suporte a composição de grandezas e transformação de unidades, que no exemplo poderia ser o volume de pesticida por animal por quilometro quadrado.

Utilizando conceitos de metrologia aplicado ao banco de dados é possível criar variáveis com grandezas derivadas que podem ser cadastradas ao banco de dados, bem como suas unidades. O valor de uma grandeza é geralmente expresso sob a forma do produto de um número por uma unidade. A unidade é um exemplo específico da grandeza em questão, usada como referência. O número é a razão entre o valor da grandeza considerada e a unidade (INMETRO, 2012).

A Figura 3.11 apresenta a estrutura de armazenamento de variáveis do SSD seguindo conceitos de metrologia.

Figura 3.11 - Estrutura de armazenamento de variáveis SSD seguindo conceitos de metrologia.



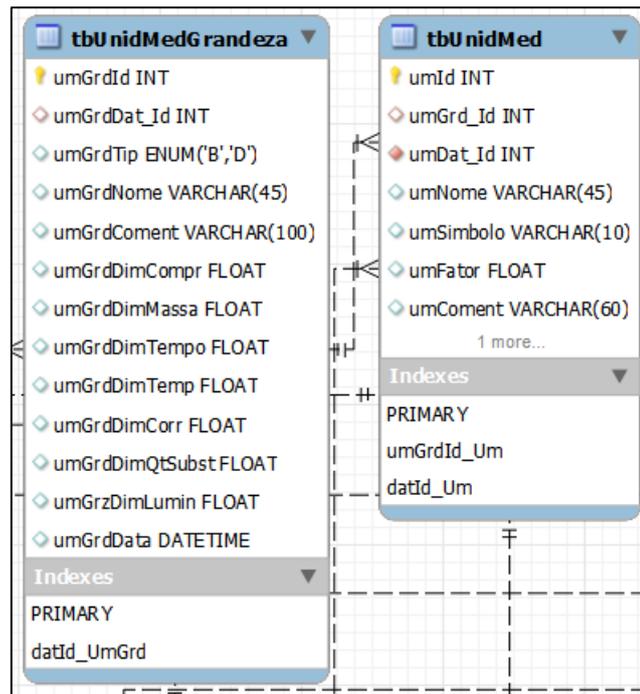
Fonte: Autor.

A estrutura do banco de dados do SSD para armazenamento de variáveis além de envolver conceito de metrologia, considera outros aspectos afim de melhor caracterizar os processos envolvidos nos sistemas hídricos. Como pode ser observado na Figura 3.11 as variáveis podem ser agrupadas, onde as mesmas possuam características ou propósitos semelhantes. Por exemplo, pode-se criar um grupo de variáveis Limnológicas para o monitoramento de seus parâmetros relacionados.

As variáveis também podem ser classificadas segundo alguma definição, por exemplo, pode-se criar uma classificação para a variável Precipitação denominada Semiárido onde são definidos as alturas máximas e mínimas médias de chuva segundo uma série histórica meteorológica.

A Figura 3.12 apresenta diagramas EER do grupo de tabelas responsáveis pelo armazenamento de dados referentes a unidade de medida e grandeza para cadastro no SSD.

Figura 3.12 - Diagramas EER do grupo de tabelas referentes a unidade de medida e grandeza.

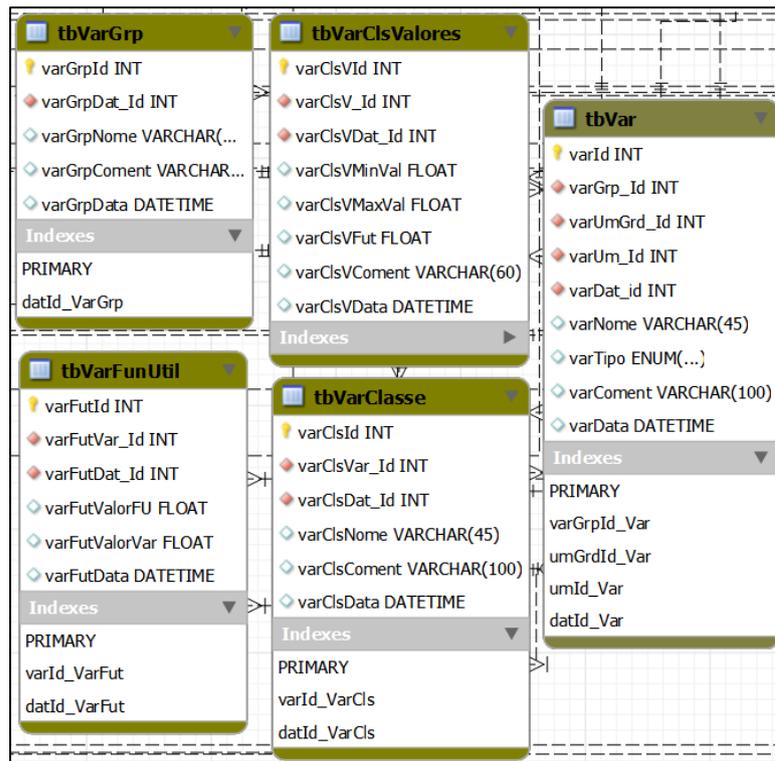


Fonte: Autor.

A tabela `tbUnidMedGrandeza` armazena os dados referentes as grandezas como seu tipo no campo `umGrdTip` (básica o derivada), seu nome e o expoente de sua dimensão estabelecido no Sistema Internacional (SI^2), porém também sendo possível cadastrar grandezas de outros sistemas. A tabela `tbUnidMed` armazena dados referentes as unidades de medidas de alguma grandeza. A tabela possui um campo com chave estrangeira que relaciona a unidade a grandeza. Entre seus campos estão o `umSimbolo`, que armazena o símbolo da unidade (m^2 , m^3/s , etc.), e o `umFator` que armazena o valor do fator de conversão da unidade.

A Figura 3.13 apresenta diagramas EER do grupo de tabelas responsáveis pelo armazenamento de dados referentes ao grupo e classe de variáveis, valores de classe, variáveis e função utilidade para cadastro no SSD.

Figura 3.13 - Diagramas EER do grupo de tabelas referentes as caracterização das variáveis.



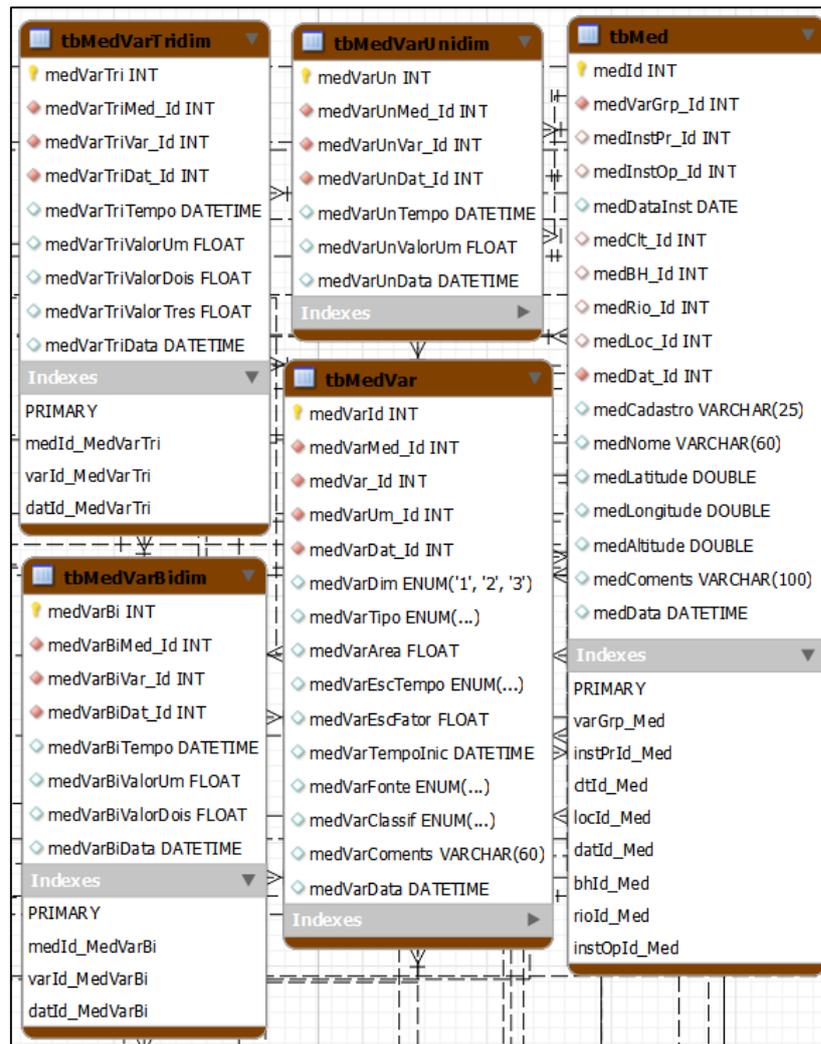
Fonte: Autor.

Como mencionado anteriormente as variáveis mantidas no banco de dados do SSD podem ser agrupadas e classificadas, sendo apresentadas as respectivas tabelas na Figura 3.13. A tabela tbVarGrp armazena os dados referentes aos grupos de variáveis (meteorológicas, qualidade de água, etc.).

Na tabela tbVar são armazenados dados referentes a variável em si (precipitação, vazão, temperatura, etc.) como: a qual grupo de variáveis ela pertence, seu tipo (Recurso, Custo, Benefício ou nenhuma delas) que está associada a alguma função utilidade - já considerando a interface com modelos de otimização - e qual unidade e grandeza está associada. A tabela tbVarClasse armazena dados referentes as classes que uma variável pode pertencer, assumindo que uma variável pode ser classificada de diferentes formas.

A Figura 3.14 apresenta diagramas EER do grupo de tabelas responsáveis pelo armazenamento de dados referentes ao sistema de medições para cadastro no SSD.

Figura 3.14 - Diagramas EER do grupo de tabelas referentes ao sistema de medição.



Fonte: Autor.

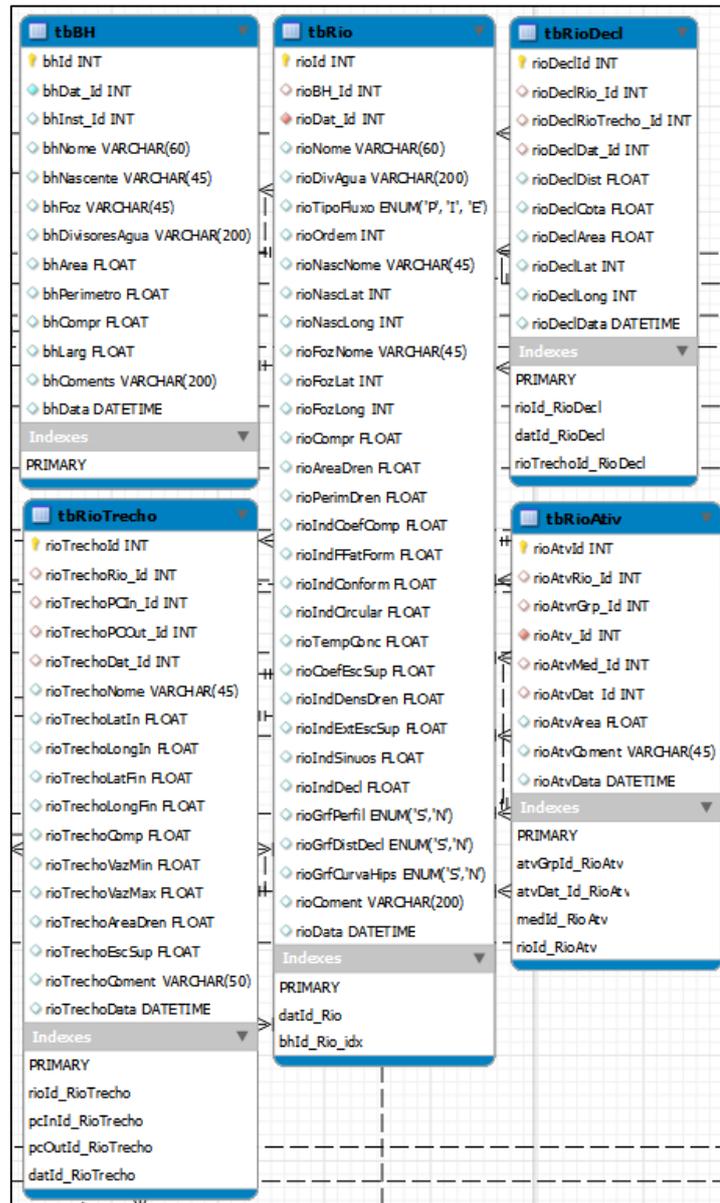
A tabela `tbMed` apresentada na Figura 3.14 armazena dados referentes a estação de medição como: código da estação, instituição e operador responsável e localização espacial. Campos com chaves estrangeiras também relaciona a estação com o rio, bacia hidrográfica e localidade, a qual a mesma pertence. As variáveis de uma estação de medição são definidas pelo relacionamento do campo `medVarGrp_Id` que relaciona a tabela grupo de variáveis. Já a tabela `tbMedVar` cadastra as variáveis em uma estação de medição, definindo sua dimensão (`medVarDim`), escala de tempo (`medVarEscTempo`), unidade de medida (`medVarUm_Id`), entre outras características.

Os valores das variáveis em si são armazenados em três tabelas que diferenciam na dimensão da variável armazenada, são as tabelas: `tbmedvarunidim`, `tbmedvarbidim` e

tbmedvartridim. Elas possuem relacionamentos com a tabela tbMedVar associando assim a variável cadastrada na estação de medição aos valores.

A Figura 3.15 apresenta diagramas EER do grupo de tabelas responsáveis pelo armazenamento de dados referentes a bacia hidrográfica e rio.

Figura 3.15 - Diagramas EER do grupo de tabelas referentes a bacias hidrográfica e rio.



Fonte: Autor.

O grupo de tabelas apresentadas na Figura 3.15 são responsáveis pelo armazenamento de dados referentes a bacias hidrográficas e rios, tais como: características físicas da bacia, característica hidráulicas, dados espaciais e gerais. Em suma grande parte destes dados só

podem ser obtidos a partir de análises espaciais em um SIG e a partir da análise das séries históricas que estarão armazenadas nas estações de medição no subsistema de análise do SSD.

Dessa forma estas tabelas podem ser atributos de um objeto espacial que poderão ser calculados no SIG e ao mesmo tempo podem ser modificadas em uma tela amigável que poderá incluir outros dados. Isso só pode ser feito devido ao fato do SGDB escolhido, MySQL, suportar tanto a geometria como os atributos dos dados espaciais.

É importante salientar que atualmente existem uma infinidade fontes de dados, o SSD proposto não pretende apenas se utilizar dados primários, aliás uma característica importante de um SSD é permitir o acesso a banco de dados e arquivos externos seja através de aplicativos, serviços ou quaisquer outras formas de aquisição.

Uma agência de águas pode fornecer, por exemplo, um arquivo vetorial com todos os reservatórios, que pode ser, por meio de mídia digital ou por serviços como o WMS (*Web Map Service*). Este arquivo é incluído no banco de dados do SSD, podendo então ser adicionado atributos customizados que não estão presentes no arquivo original.

A Agência Executiva de Gestão das Águas do Estado da Paraíba (AESAs) disponibiliza dados espaciais gerados no Plano Estadual de Recursos Hídricos, são arquivos vetoriais no formato *shapefile* que podem ser importados para o banco gerenciado pelo MySQL utilizando ferramentas disponibilizada pela OSGeo (*Open Source Geospatial Foundation*). A Figura 3.16 apresenta conjunto de tabelas responsáveis pelo armazenamento da geometria, referência espacial e atributos de alguns dados espaciais disponibilizados pela AESA.

Figura 3.16 - Conjunto de tabelas responsáveis por armazenar atributos e geometria de dados espaciais.

Table Name	Fields	Indexes
spatial_ref_sys	SRID INT(11) AUTH_NAME VARCHAR(256) AUTH_SRID INT(11) SRTEXT VARCHAR(2048)	
geometry_columns	F_TABLE_CATALOG VARCHAR(256) F_TABLE_SCHEMA VARCHAR(256) F_TABLE_NAME VARCHAR(256) F_GEOMETRY_COLUMN VARCHAR(256) COORD_DIMENSION INT(11) SRID INT(11) TYPE VARCHAR(256)	
tbbacias	OGR_FID INT(11) SHAPE GEOMETRY objectid DECIMAL(9,0) nome VARCHAR(30) comite VARCHAR(30) perimetro DOUBLE(19,11) area_km2 DOUBLE(19,11) shape_leng DOUBLE(19,11) shape_area DOUBLE(19,11) cod DECIMAL(4,0)	OGR_FID SHAPE
tbrios	OGR_FID INT(11) SHAPE GEOMETRY objectid DECIMAL(9,0) nome VARCHAR(30) ordem VARCHAR(20) zona DOUBLE(19,11) dominio VARCHAR(50) codigootto DOUBLE(19,11) shape_leng DOUBLE(19,11)	OGR_FID SHAPE
tbacudes	OGR_FID INT(11) SHAPE GEOMETRY objectid DECIMAL(9,0) nome VARCHAR(50) executor VARCHAR(50) propriet VARCHAR(50) municipio VARCHAR(30) microrreg VARCHAR(50) mesorreg VARCHAR(50) finalidade VARCHAR(200) a_espelho DOUBLE(19,11)	

Fonte: Autor.

A tabela `spatial_ref_sys` armazena dados referentes a referência espacial dos objetos como sistema de coordenadas e datum. A tabela `geometry_columns` armazena dados referentes a geometria dos objetos associando a referência espacial com base no nome da tabela do objeto espacial.

As tabelas `tbbacias`, `tbrios` e `tbacudes` são resultado do processo de importação dos arquivos *shapefiles* de bacias, rios e açudes do estado da Paraíba fornecidos pela AESA. Nestas tabelas a geometria é armazenada no campo `SHAPE GEOMETRY` num formato binário e os atributos em campos comuns. Elas também possuem indexação espacial para realizar o relacionamento com as tabelas `spatial_ref_sys` e `geometry_columns`.

3.3.3 Visualização de Dados Espaciais

Como discutido no capítulo anterior, há diversos tipos de SIGs, porém as principais características que os diferenciam são a forma como armazenam os dados espaciais, esta que está diretamente ligada a sua arquitetura e o público alvo do SIG.

Alguns SSDs para recursos hídricos não utilizam SIGs de forma integrada ao sistema. Geralmente, os modelos importam dados gerados por análises realizadas em um SIG disponível no mercado, como QGIS e ArcGIS, para então serem aplicados em modelos sobre recursos hídricos. Dados estes, que já teriam sido previamente exportados para um formato que o modelo

possa importar. Isso gera problemas com a integridade dos dados e quebras no fluxo no funcionamento do sistema.

Há também a possibilidade de criar um SIG customizado para a aplicação com uso de componentes GIS desktop, livres ou proprietários, disponíveis no mercado como o MapWindow e TatumGIS. Dessa forma com o SIG integrado ao núcleo do sistema é possível evitar os problemas mencionados anteriormente. Porém, o SSD aqui proposto também possuirá interface web, sendo necessário então desenvolver um SIG para uso desktop e outro para web.

Considerando as questões acima, optou-se pela utilização de um *framework* SIG baseado na web que também pode ser utilizado na interface desktop, o i3Geo (Interface Integrada para Internet de Ferramentas de Geoprocessamento). Um *framework* é um conjunto de classes cooperantes que constroem um projeto reutilizável para uma determinada categoria de software (Biggerstaff, 1989; Johnson e Foote, 1988).

Segundo Gamma *et al.*, (2008), o *framework* dita a arquitetura da aplicação. Em parte isso se aplica ao SSD proposto, pois o i3Geo também possui arquitetura em camadas cliente-servidor, porém como o SSD é uma aplicação que possui muitos componentes, isso apenas se aplica ao componente SIG. Tal afirmação se aplicaria por completo caso se utilizasse um *framework* de um SSD com arquitetura definida e classes abstratas para customização e criação de subclasses.

A principal vantagem do uso de se usar *frameworks* é a praticidade. Eles possuem funcionalidades nativas para resolver problemas e atender necessidades de aplicações e softwares. Outra vantagem é a extensibilidade, eles podem ser customizados por meio da implementação de novas funcionalidades estendendo seus usos.

Os desenvolvedores de *frameworks* geralmente disponibilizam material de suporte e documentação na web e repositórios oficiais. Além disso, a maioria dos frameworks têm uma comunidade de desenvolvedores ativa e dispostos a compartilhar conhecimento.

O i3Geo é um software livre, licenciado como GPL⁵ e criado pelo Ministério do Meio Ambiente (MMA) do Brasil em 2004. O i3Geo integra o Portal do Software Público Brasileiro (PSPB), que é um projeto do Ministério do Planejamento (MP) para apoiar o desenvolvimento

⁵ Software sobre licença GPL permite que qualquer usuário tenha acesso aos códigos do software para alterá-los, aprender e compartilhar o conhecimento adquirido. A licença também impede que os códigos sejam fechados, assim, as modificações feitas devem também ser compartilhadas sem nenhum custo.

de soluções de código aberto de interesse da sociedade e do governo, possuindo grande comunidade e vasta documentação.

O i3Geo é um software utilizado para a construção de mapas interativos na internet e disseminação de dados geoespaciais e foi desenvolvido para ser instalado em servidores acessíveis por meio web, mas pode, também, ser instalado em computadores pessoais. O software foi criado para permitir adaptações que atendam às necessidades específicas dos usuários de determinada instalação ou aplicação (i3Geo, 2012).

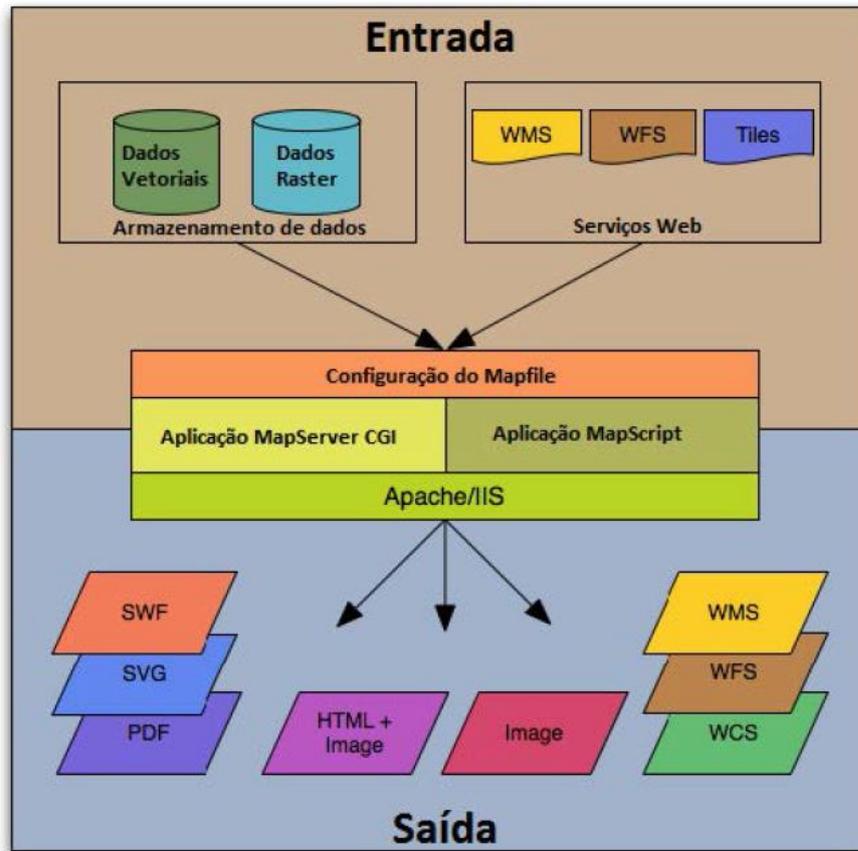
O i3Geo tem seu núcleo base desenvolvido no software MapServer. O MapServer é um projeto popular *Open Source* cujo objetivo é exibir mapas espaciais dinâmicos através da internet (MapServer, 2014). Algumas funcionalidades do MapServer são:

- Suporte para a exibição e consulta de centenas de formatos *raster*, vetorial e de bancos de dados;
- Capacidade de funcionar em vários sistemas operacionais (Windows, Linux, Mac OS X, etc.);
- Suporte para linguagens de script populares e ambientes de desenvolvimento (PHP, Python, Perl, Ruby, Java, NET);
- Projeções *on-the-fly*⁶;
- Renderização de alta qualidade;
- Saída do aplicativo inteiramente customizável;
- Muitos ambientes de aplicativos *Open Source* prontos para uso.

A Figura 3.17 apresenta a arquitetura básica de aplicativos que usam o MapServer.

⁶ Capacidade de reprojeter uma camada quando inserida para a projeção definida no mapa base caso a mesma tenha projeção diferente.

Figura 3.17 - Arquitetura básica de aplicativos MapServer.



Fonte: Adaptado de MapServer, 2014 (tradução).

A entrada dos dados geográficos para a produção de mapas podem vir por meio de duas fontes de dados, via banco de dados gerenciado por um SGDB, no caso o MySQL, e por meio de *Web Services* como o WMS. Em alguns casos pode-se adicionar dados diretamente de diretório do disco.

Um arquivo de configuração Mapfile define área do mapa e diz ao MapServer onde os dados estão armazenados. Ele também define projeções e simbologia. O *Driver ODBC* para realizar a interface com o MySQL é fornecido pela *OGR/GDAL (OpenGIS Simple Features Reference/ Geospatial Data Abstraction Library)*.

O MapServer CGI recebe as requisições e gera os mapas em diversos formatos. Como o MapServer é um servidor de mapas internet ele precisa de um servidor web como o Apache ou IIS. O servidor web utilizado foi o Apache. O Apache é atualmente o servidor web livre mais utilizado no mundo, tem como suas principais características a segurança e a eficiência.

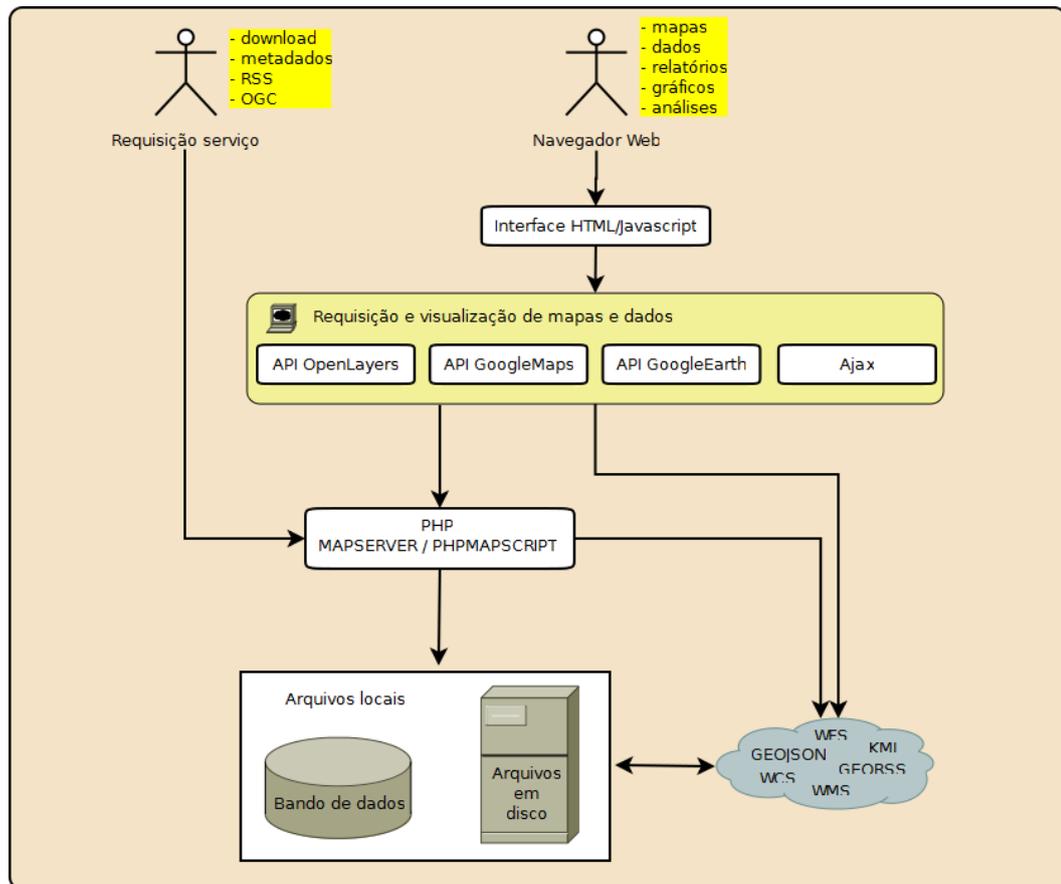
Assim as características do i3Geo são, em grande parte determinadas pelas características e limitações do Mapserver (i3Geo, 2012).

O lado servidor do i3Geo utiliza a linguagem de programação PHP mais PHP Mapscript fornecido pelo Mapserver, que é executado no servidor onde o i3Geo se encontra. O lado cliente utiliza a linguagem Javascript, sendo executado por um navegador web ou na aplicação desktop utilizado pelo usuário.

A comunicação entre o navegador e o servidor ocorre por meio de requisições AJAX e elementos HTML como os marcadores `<script>`, `<iframe>` e ``. Além dos códigos Javascript que foram desenvolvidos especificamente para o i3Geo (i3Geo, 2012).

Entre as ferramentas para análise geográfica estão: grades, distância entre pontos, distribuição de pontos, centroide, centro médio, *dissolve*, *buffer*, agrupar, gráficos e linha do tempo. Outro bom recurso do i3Geo é sua integração com o ambiente estatístico R que possibilita realizar operações espaciais como Diagrama de Voronoy (Polígonos de Thiessen) e Mapa de Pontos Quentes. A Figura 3.18 apresenta a arquitetura do i3Geo.

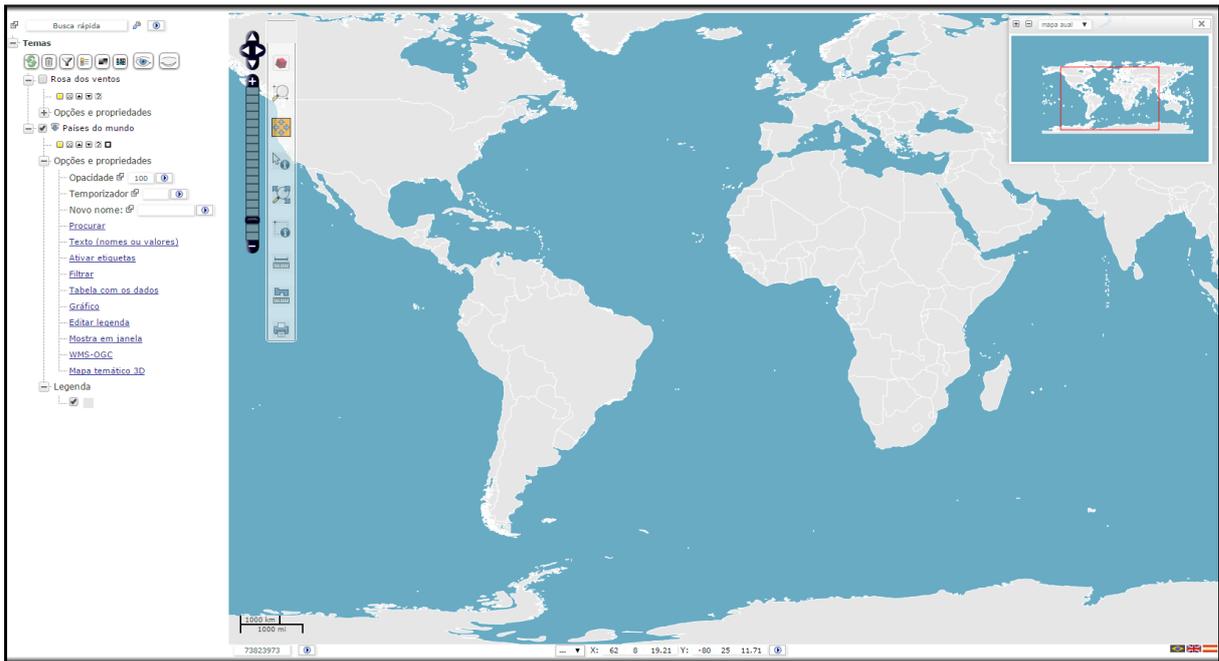
Figura 3.18 - Arquitetura do i3Geo.



Fonte: i3Geo, 2012.

O software vem com um banco de dados padrão SQLite com tabelas e dados referentes a administração do sistema. O i3Geo ainda se utiliza das APIs do OpenLayers, GoogleMaps e GoogleEarth, o que possibilita mais funcionalidades para o mesmo e maior interação como usuário. A Figura 3.19 apresenta a interface básica do i3Geo que pode ser totalmente customizada.

Figura 3.19 - Interface básica do i3Geo.



Fonte: Captura de tela em ambiente Windows.

3.3.4 Desenvolvimento Lado Servidor

O lado servidor do SSD é o núcleo do sistema, possuindo as camadas que disponibilizarão métodos e dados para serem consumidos pelas camadas de apresentação.

O Delphi dispõe de um *framework* nativo denominado DataSnap para o desenvolvimento em multicamadas. A grande vantagem da utilização do DataSnap, incluída desde a versão 2009, é que as camadas cliente podem ser desenvolvidas em qualquer linguagem de programação que possua suporte à JSON (*JavaScript Object Notation*).

JSON é um formato dados leve de intercâmbio para todas as linguagens de programação. O formato é fácil de ser lido e escrito pelos humanos e fácil para os computadores de analisar e gerar. Ele é baseado em um subconjunto da linguagem de programação JavaScript (ECMA, 2013).

Apesar de muito simples, o JSON tem sido bastante utilizado por aplicações web devido a sua capacidade de estruturar informações de uma forma bem mais compacta do que o modelo XML, tornando mais rápido o *parsing* (análise) dessas informações. Empresas como Google e Yahoo, que possuem aplicações que necessitam transmitir grandes volumes de dados adotaram JSON para o intercâmbio de dados.

Outras formas de intercâmbio de dados para web também muito utilizadas são os *Web Services*, que também foram criados para permitir a integração cliente e servidor sem expor detalhes como plataforma, linguagens, etc. De forma geral os *Web Services* vieram para permitir que aplicações desenvolvidas em diferentes linguagens, rodando sobre diferentes plataformas, pudessem trocar dados entre si de forma transparente.

O padrão de comunicação entre as aplicações que utilizam *Web Service* é o protocolo SOAP (*Simple Object Access Protocol*) de comunicação e os dados são estruturados via XML. A ANA usa *Web Services* sobre protocolo SOAP para disponibilizar acesso aos dados da HidroWeb.

Porém os *Web Services* utilizam SOAP sobre HTTP, ou seja, somente o HTTP não é suficiente para se trabalhar com *Web Services*. O SOAP encapsula as mensagens enviadas entre cliente e servidor, resultando em mais trabalho e causando certa lentidão na transmissão das informações.

Dessa forma optou-se por utilizar um novo paradigma que tem se tornado uma alternativa viável ao SOAP e bem utilizada em aplicações multicamadas, o REST (*Representational State Transfer*).

O REST surgiu da tese de doutorado sobre a web escrita por Roy Fielding, que foi um dos idealizadores do protocolo HTTP, e tem como princípio conceitos de *separation of concerns* com objetivo de simplificar interfaces em um sistema, aumentar a escalabilidade, performance, portabilidade e customização (até em tempo de execução) (Fielding, 2000).

Uma aplicação RESTful - termo utilizado para identificar um sistema que segue as ideias REST - combina o uso dos princípios estabelecidos pela técnica do REST como: um protocolo cliente-servidor sem estado (*stateless*), ou seja, cada mensagem HTTP contém toda a informação necessária para compreender uma requisição; e um conjunto bem definido de operações que se aplica a todos os recursos de tratamento de dados.

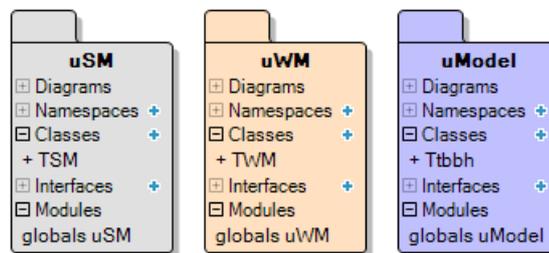
As operações de uma aplicação RESTful se assemelham as operações CRUD na persistência de dados tradicional, são elas: *POST*, *GET*, *PUT* e *DELETE*. O DataSnap possibilita criação de servidor REST com padrão de intercâmbio de dados JSON.

A *engine* de acesso ao banco de dados utilizada foi o dbExpress, que é uma tecnologia presente no Delphi que nos últimos anos se tornou muito estável e rápida na persistência de

dados, ela possui interfaces com os principais bancos de dados, sendo compatível com MySQL via driver ODBC.

O servidor possui três *units* principais para trabalhar com o banco de dados, as classes e métodos, são elas: a uSM que possui as classes e métodos para realizar a conexão com o SGDB e disponibilizar os métodos para serem utilizados pelo lado cliente desktop, a uModel que possui todas as classes modeladas a partir banco de dados e a uWM que disponibiliza os métodos para serem utilizados pelo lado cliente web. A Figura 3.20 apresenta as principais *units* presentes no servidor de aplicação.

Figura 3.20 - Principais *units* do servidor de aplicação.

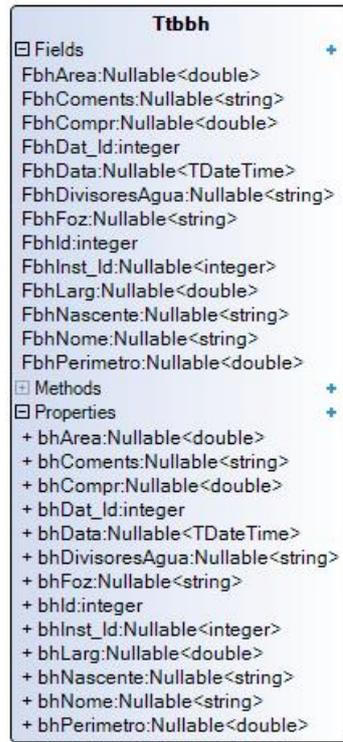


Fonte: Autor.

As *units* uSM (*Server Module*) e uWM (*Web Module*) apresentadas na Figura 3.20 apresentam suas principais classes TSM e TWWM, respectivamente. A classe TSM é uma classe que herdeira da classe TDSServerModule que expõe *datasets* e métodos do servidor para aplicações cliente. A classe TWWM herda seus atributos e métodos da classe TWebModule que permite a aplicação servidor web responder a mensagens de solicitação HTTP, passando os objetos da solicitação.

A *unit* uModel possui todas as classes modeladas a partir do banco de dados, na Figura 3.20 é apresentada a classe Tbbh. Por exemplo, a classe Tbbh possui em seus atributos nome, área e id que são campos da tabela tbbh que está presente no banco de dados. As classes modeladas, geralmente chamadas de classes entidades, ainda se utilizam de conceitos RTTI e *Custom Attributes* para serem customizadas de forma a identificar dentro da mesma a tabela do banco de dados e seus campos equivalentes, realizando assim o que se chama de Mapeamento Objeto-Relacional (ORM – *Object-Relational Mapping*). A Figura 3.21 apresenta o diagrama de classe da classe Tbbh.

Figura 3.21 - Diagrama de classe da classe Tbbbh.

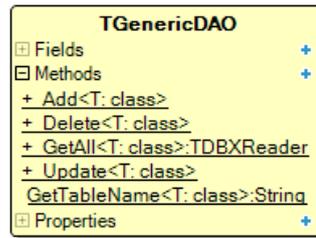


Fonte: Autor.

Esta abordagem de implementação para as classes modeladas do sistema de recursos hídricos, foi utilizada para contornar o problema da impedância objeto-relacional discutido anteriormente neste capítulo. É utilizado então um padrão de projeto denominado DAO (*Data Access Object*). Simplificadamente o DAO recupera dados no banco e converter em estados de um objeto para ser utilizado pela aplicação. Da mesma forma, ele salva, deleta e atualiza dados de uma tabela em um banco, neste caso relacional, a partir do estado do objeto correspondente, traduzindo em instruções SQL a partir dos atributos customizados da classe entidade.

O DAO encapsula e abstrai todo o acesso aos dados tornando a persistência de dados totalmente independente, fazendo com que seja muito utilizado em sistemas multicamadas. Existem alguns *frameworks* DAO consagrados como o Doctrine para PHP, e Hibernate para Java. Foram testados dois *frameworks* no processo de desenvolvimento do SSD, o InstantObjects (Gratuito) e uma versão trial do TMS Aurelius (Proprietário), porém ambos se mostraram complicados de se integrar ao sistema, em especial o InstantObjects com o MySQL, e também deixaram o sistema um pouco pesado, em especial o TMS Aurelius. Dessa forma decidiu-se implementar uma classe DAO própria com os métodos para persistir os dados. A Figura 3.22 apresenta o diagrama de classe da classe DAO.

Figura 3.22 - Diagrama de classe da classe TGenericDAO.



Fonte: Autor.

A classe TGenericDAO fornece métodos genéricos para realizar o CRUD para qualquer classe de entidade. O DAO também é denominado como camada de persistência (Elliott, O'Brien e Fowler, 2009).

Apesar de ter sido desenvolvido a classe DAO genérica ainda é necessário implementar os métodos para cada classe, onde é realizado a conversão do estado do objeto para o padrão JSON e então o mesmo ser persistido no bando de dados. O objetivo da criação da camada DAO neste trabalho é permitir que a camada de apresentação web possa consumir os métodos do servidor de aplicações quando a mesma for integrada ao sistema. A camada DAO também possibilitará recuperar os dados do banco para serem trabalhos pelos modelos em formatos mais interessantes como listas genéricas, *arrays*, etc.

Como o objetivo deste trabalho foi desenvolver o lado cliente desktop e o mesmo foi desenvolvido na mesma linguagem do servidor, decidiu-se aproveitar das ferramentas que o próprio Delphi disponibiliza para a persistência dos dados no lado cliente. O dbExpress junto ao DataSnap oferecem classes que constroem automaticamente comandos SQL para serem enviados para o servidor e então executados, isto é possível utilizando componentes específicos que a IDE oferece denominados *Data Controls*. Os mesmos possuem algumas limitações, que podem ser contornadas por meio da criação de comandos SQL adicionais, porém automatizam bem o sistema tornando ao desenvolvimento ágil.

Dessa forma, a classe TSM (Figura 3.20) também possui componentes derivados das classes TSQLConnection e TSQLDatasetProvider que criam os *datasets* para serem acessados pelo lado cliente, que será descrito no próximo tópico.

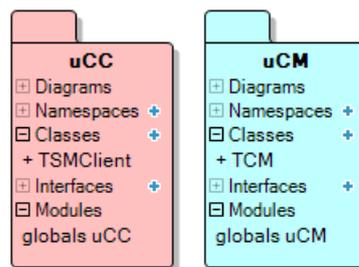
Outra *unit* importante no lado servidor é a uMetodos que contém diversas funções e procedimentos para serem utilizadas nos diversos módulos do sistema como: login, *logger*, criar e ler extensões do sistema, realizar pesquisas, validações, etc.

3.3.5 Desenvolvimento Lado Cliente Desktop

O lado cliente desktop do SSD será composto pela interface visual do usuário, classes de acesso aos métodos do servidor de aplicação e algumas regras de negócio que não comprometa a arquitetura do sistema em termos de independência das camadas, muitas regras de validação, por exemplo, são possíveis de serem definidas nos próprios componentes visuais que a IDE do Delphi disponibiliza.

Duas *units* principais do lado cliente são responsáveis por realizar a conexão com o servidor de aplicações e construir interface para acessar *datasets* e métodos remotamente, a *uCM* (*Client Module*) e *uCC* (*Client Classes*), respectivamente. A Figura 3.23 apresenta as principais *units* no lado cliente.

Figura 3.23 - Principais *units* do lado cliente.



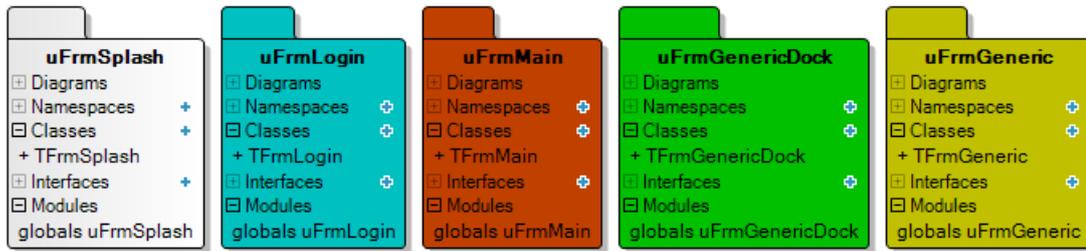
Fonte: Autor.

A classe TCM é responsável por criar a conexão com o servidor REST e abrigar os *datasets* no lado cliente que são disponibilizados por um *provider* (provedor) no lado servidor. Já a classe TSMClient realiza a interface para acessar todos os recursos que o servidor disponibiliza, ela utiliza um *proxy* (procurador) para acessar e encapsular objetos remotos, essa abordagem é conhecida como *proxy pattern* (padrão procurador).

A classe TSMClient herda da classe TDSAdminRestClient, uma classe interessante do Delphi que abstrai e encapsula muitos recursos para trabalhar com servidor REST e estrutura de dados JSON. Na classe TSMClient estão todos os métodos necessários pelo lado cliente, como por exemplo, o método para autenticação do sistema, acesso aos modelos, *logger*, entre outros.

A camada de apresentação desktop possui cinco *units* principais com suas respectivas classes de formulários ou *forms* que constroem a interface com o usuário, apresentadas na Figura 3.24.

Figura 3.24 - Principais *units* que constroem a interface com o usuário.



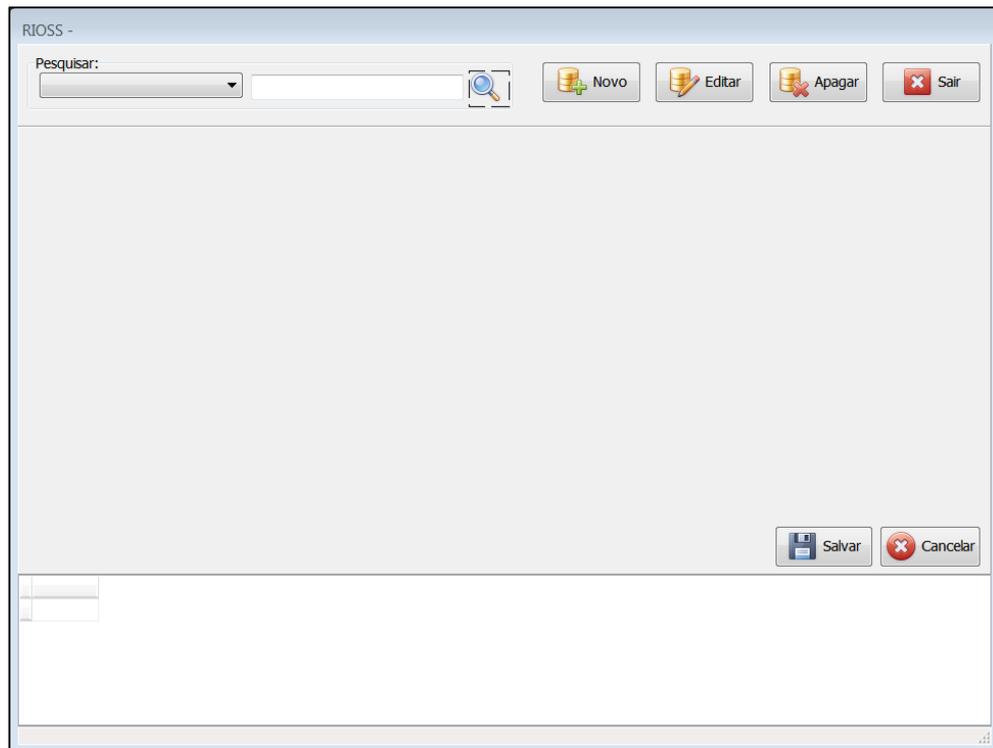
Fonte: Autor.

A *unit* `uFrmSplash` contém a tela ou *view* de abertura do sistema, também conhecida como *splash screen*. A *unit* `uFrmLogin` apresenta a tela de autenticação do usuário, ela é a primeira tela apresentada ao usuário quando o sistema é iniciado. A *unit* `uFrmMain` é a tela principal do SSD, nela estão diversos controles como a barra de ferramentas, que dá acesso as outras telas da aplicação.

As *units* `uFrmGeneric` e `uFrmGenericDock` contém as classes dos *forms* genéricos que possuem métodos e componentes visuais que são utilizados por mais de uma tela, outros *forms* são criados a partir deles utilizando um recurso chamado Herança Visual, que permite os *forms* herdeiros herdarem os métodos e os componentes visuais de um *form* genérico, já que o *form* na verdade é uma classe. Por exemplo, uma tela de cadastro de um usuário possui quatro botões visuais essenciais para incluir, editar, deletar e atualizar um registro no banco de dados, cada botão está associado a sua respectiva função, então outra tela para cadastro de instituições, por exemplo, pode ser criada herdando estas características.

A telas advindas da classe `TFrmGeneric` possuem comportamento normal como minimizar, maximizar, redimensionar, etc, já as telas herdadas da classe `TFrmGenericDock` podem ser flutuantes, fixadas em algum lugar da tela principal, se esconder, virar abas, etc. A Figura 3.25 apresenta a interface do *form* genérico de cadastro da classe `TFrmGeneric`.

Figura 3.25 - Interface do *form* genérico de cadastro.

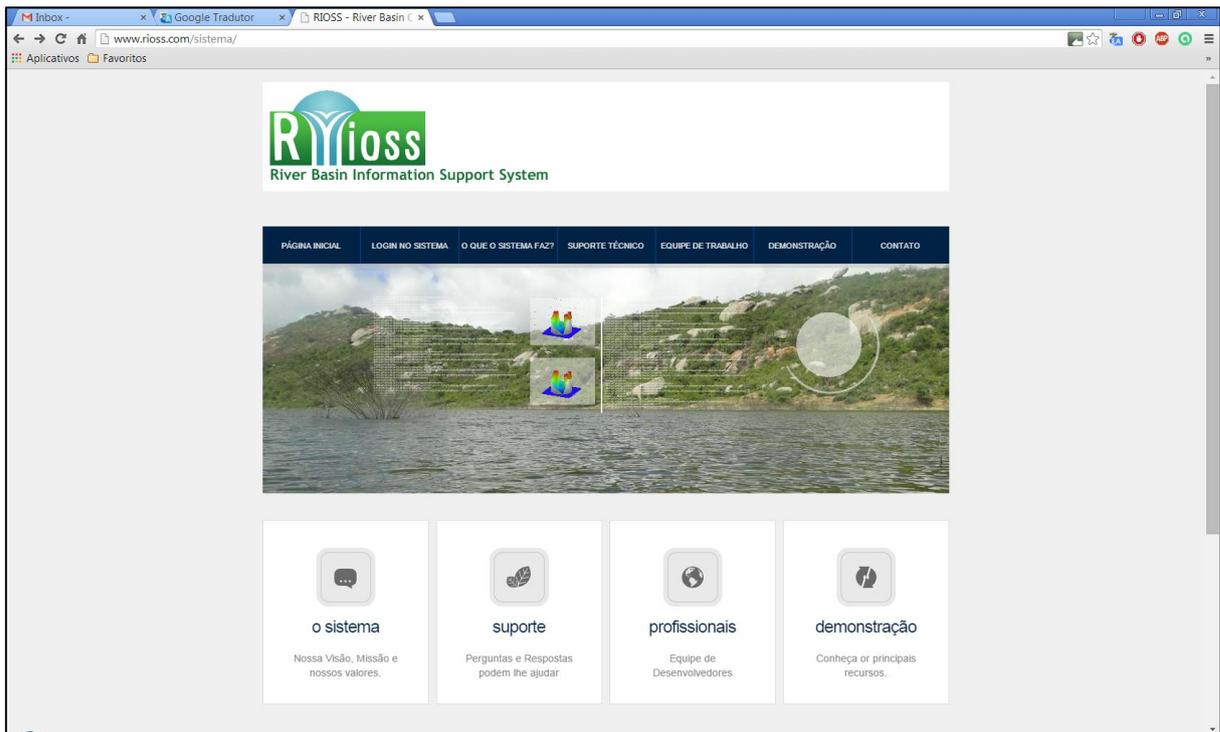


Fonte: Autor.

3.3.6 Lado Cliente Web

Como abordado anteriormente neste capítulo, este trabalho tem como foco o desenvolvimento lado cliente desktop, porém o servidor estará apto a ser integrado a outros clientes sendo eles capazes de consumir os serviços do SSD. A Figura 3.26 apresenta a interface web que está sendo desenvolvida para o RIOSS pelo grupo GOTA.

Figura 3.26 - Interface web para o RIOSS.



Fonte: Autor.

3.3.6 Sistema de Modelos

Na literatura consultada basicamente existem duas abordagens de desenvolvimento para um sistema de modelos de um SSD em recursos hídricos. A primeira é implementar os modelos na linguagem de programação base que o sistema foi desenvolvido. Uma segunda é criar um mecanismo para utilizar modelos já implementados em outras linguagens de programação. As duas maneiras também podem ser utilizadas de forma conjunta.

Algumas linguagens têm aptidão e disponibilizam ferramentas para implementação de certos tipos de modelos, como estatísticos de previsão, regressão, etc. Outras vezes a integração de modelos implementados em linguagens diferentes da do sistema nativo pode prejudicar o funcionamento do sistema em termos de performance ou simplesmente não podem ser integrados. Dessa forma deve-se considerar muitos fatores no desenvolvimento do sistema de gerenciamento de modelos do SSD.

Os modelos de recursos hídricos têm sido geralmente implementados nas linguagens procedurais ou estruturadas como C, Fortran, MatLab, Basic e Pascal. Os principais motivos são a velocidade que os algoritmos escritos nessas linguagens são processados e a facilidade de implementação devido ao grande número de funções e ferramentas disponíveis. O MatLab, por

exemplo, além de processar rapidamente uma rotina possui uma infinidade de funções pré-programadas que facilitam a modelagem em recursos hídricos.

O principal problema nesses modelos é a falta de interface gráfica para serem utilizados por usuário finais, o que acabou tornando os modelos com baixa ou nenhuma aplicabilidade. Tanto importante é essa interação que empresas desenvolvedoras de compiladores e IDEs tem incorporando suporte a orientação a objetos a linguagens procedurais, incorporando as vantagens da orientação a objetos e possibilitando a produção de interfaces amigáveis para os usuários finais.

Por exemplo, a MathWorks empresa que mantém o MatLab, além de ter incorporado a orientação a objetos a linguagem ainda disponibiliza ferramentas para construção de interfaces para o usuário e para compilar e compartilhar os aplicativos desenvolvidos na linguagem para serem utilizados por outros sistemas desenvolvidos em outras linguagens como C, C++, Java e .Net.

Um SSD para recursos hídricos é um sistema complexo e seu sistema de modelos necessita de diversas ferramentas e modelos de otimização, simulação, estatísticos, naturais do sistema de recursos hídricos (chuva-vazão, erosão, escoamento, etc.), multicriteriais, entre outros, assim na estrutura do sistema aqui proposto, em um primeiro momento, buscou-se a integração de modelos já disponíveis e implementados em linguagens - diferentes do SSD - geralmente utilizadas em recursos hídricos.

No desenvolvimento do SSD proposto foram utilizadas diversas bibliotecas, frameworks e *toolkits* já implementados por terceiros. Atualmente o desenvolvimento, principalmente *Open Source*, tem sido de forma colaborativa, aliás um dos conceitos do desenvolvimento aberto é analisar códigos, melhorá-los e disponibilizá-los para serem reutilizados por outros usuários. Porém, na área de desenvolvimento de ferramentas computacionais para serem utilizadas em recursos hídricos o compartilhamento e colaboração não é prática corriqueira, salvo raras exceções.

Foi selecionado um modelo para integração ao sistema de modelos frequentemente utilizado por hidrólogos no planejamento em recursos hídricos, um modelo do tipo chuva-vazão, o SMAP⁷ (*Soil Moisture Accounting Procedure*).

O SMAP possui versões implementadas em algumas linguagens de programação como Fortran, MatLab e R. Foi escolhida uma versão implementada na linguagem R⁸ de acesso aberto disponibilizada em forma de pacote/biblioteca no repositório GitHub pelos autores Lourenço *et al.* (2014).

Nos últimos anos vários pacotes na área de engenharia de recursos hídricos vêm sendo desenvolvidos e disponibilizados por hidrólogos de diversas instituições para trabalhar com dados hidrológicos no R (Lourenço *et al.*, 2014). O R também foi utilizado para realização de algumas operações espaciais no i3Geo, como apresentado anteriormente, mostrando a diversidade de dados que o mesmo pode trabalhar.

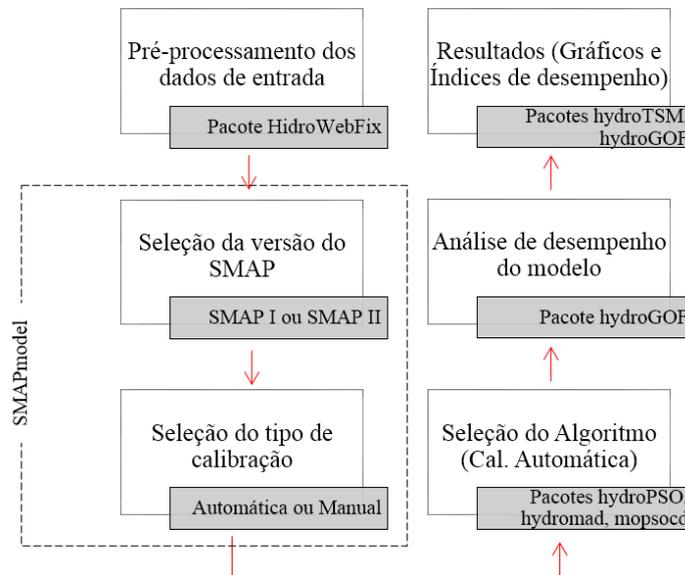
O pacote R denominado SMAPmodel implementa o modelo SMAP nas estruturas de Lopes *et al.* (1982) e Diniz (2008) em suas versões mensais, podendo ser aplicado tanto em bacias que são regidas principalmente pela precipitação e evaporação, como em bacias com solos profundos, onde o escoamento de base tem significativa influência no balanço de massa da bacia. O modelo pode ter a calibração do tipo manual, ou automática mono-objetivo, ou automática multiobjetivo.

A Figura 3.27 apresenta as etapas de processamento e modelagem do SMAPModel.

⁷ O SMAP é um modelo determinístico, conceitual e concentrado de simulação hidrológica do tipo transformação chuva-vazão desenvolvido por Lopes *et al.* (1982).

⁸ O R é uma linguagem e ambiente para computação estatística e gráfica, de licença livre, que vem ganhando adesão na análise e modelagem hidrológica. O ambiente possui uma linguagem relativamente simples com uma gama de ferramentas disponibilizadas pelos seus usuários, possibilitando a realização de diversas tarefas em diferentes áreas do conhecimento.

Figura 3.27 - Etapas do processamento e modelagem do SMAPmodel.



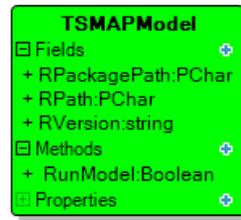
Fonte: Lourenço *et al.*, 2014.

Como apresentado na Figura 3.27 o modelo se utiliza de outras bibliotecas R disponíveis como a HidroWebFix, para pré-processamento das séries históricas de vazão e precipitação, hydroPSO que implementa o algoritmo de otimização *Particle Swarm Optimization* (PSO) para calibração do modelo, e funções da biblioteca hydroGOF, para realizar a análise de desempenho do modelo.

Para integrar o modelo ao SSD foi necessário realizar algumas modificações no código fonte original do SMAPModel. Como todos os dados do sistema estão em um banco de dados centralizado e são gerenciados pelo SGDB MySQL, os dados de entrada do modelo deverão partir do banco de dados, e os dados de saída do modelo deverão ser gravados no banco de dados. A etapa de pré-processamento de dados foi retirada do modelo já que os dados são providos pelo sistema de informações do SSD, e onde os mesmos deverão ser pré-processados pelo sistema de análise.

O R pode se conectar ao MySQL por meio da biblioteca RMySQL, dessa forma o SMAPModel foi configurado para se conectar ao MySQL possibilitando ler os dados de entrada definidos pelo usuário e posteriormente gravar as saídas. A Figura 3.28 apresenta o diagrama de classe da classe TSMAPModel que possui o método RunModel que invoca o modelo para ser rodado no R, ambos instalados no lado servidor.

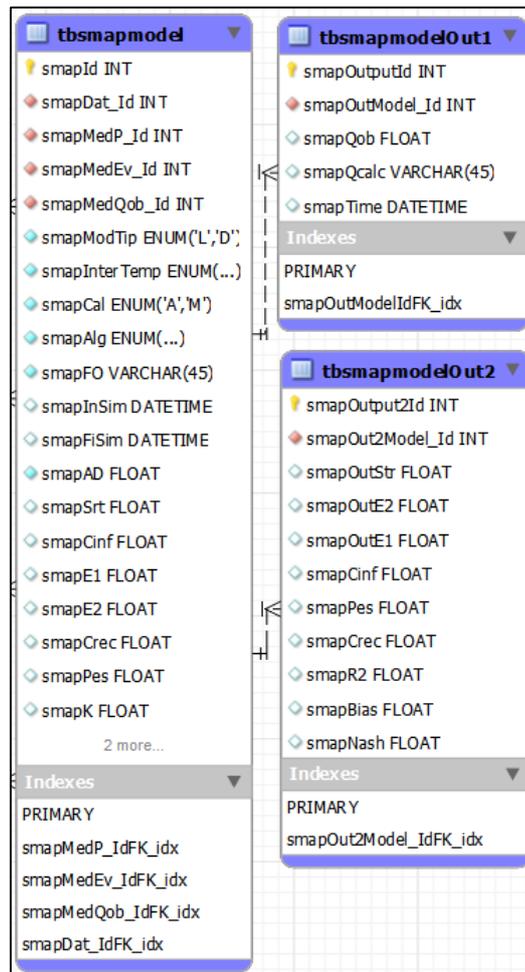
Figura 3.28 - Diagrama de classe da classe TSMAPModel.



Fonte: Autor.

Para o armazenamento dos dados de entrada e saída do modelo foi criado um grupo de tabelas. A Figura 3.29 apresenta diagramas EER do grupo de tabelas referente ao SMAPModel.

Figura 3.29 - Diagramas EER do grupo de tabelas referente ao SMAPModel.



Fonte: Autor.

A tabela tbsmapmodel apresentada na Figura 3.29 é responsável por armazenar dados referentes a entrada do modelo como parâmetros, caso calibração manual, id das estações de medição cadastradas no sistema de informação com os dados de precipitação, evaporação e

vazão, forma de calibração, intervalo de simulação entre outros. A tabela `tbsmapmodelOut1` armazena os dados de saída para a vazão calculada e vazão observada para cálculo da eficiência do modelo.

A tabela `tbsmapmodelOut2` também armazena dados de saída do modelo como os parâmetros resultantes da calibração e os índices de eficiência. As duas tabelas com dados de saída possuem campos estrangeiros que relacionam o id da simulação realizada.

3.3.7 Acesso a Dados Externos

Uma importante característica do sistema de informações de um SSD é possibilitar o acesso a dados externos por meio de mecanismos de importação ou outras interfaces de acesso a outros bancos de dados, dessa forma foram desenvolvidos dois aplicativos para acesso aos dados hidrológicos fornecidos pela Agência Nacional da Águas (ANA) e Agência Executiva de Gestão das Águas do Estado da Paraíba (AESA).

Para um usuário comum o sistema de informações da ANA, o HidroWeb, fornece os dados on-line mediante consulta de formulário, sendo gerado um arquivo compactado .ZIP com um único arquivo interno que pode ser no formato .TXT ou .MDB, dependendo da escolha do usuário, com a série temporal da variável hidrológica escolhida. O arquivo possui um cabeçalho com informações sobre a estação e legendas e os dados são separados em colunas e linhas por ponto-e-vírgula.

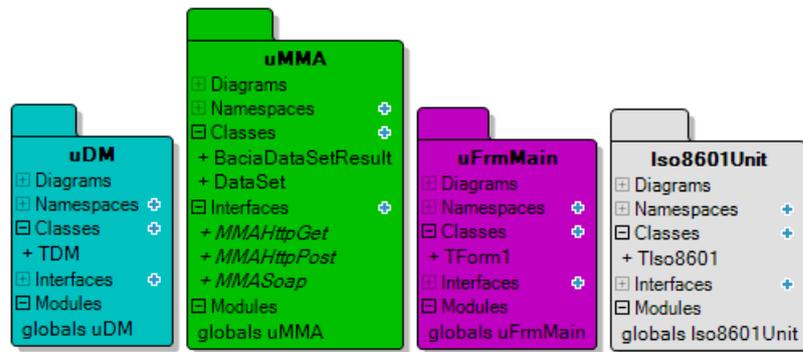
A AESA disponibiliza os dados em formato semelhante via arquivos no formato .XLS mediante requisição junto a instituição via e-mail ou pessoalmente, e também disponibiliza a visualização dos dados no seu website.

A ideia dos aplicativos propostos é realizar o acesso direto e automatizado a estes dados pelo SSD, sem a necessidade do usuário fazer download ou solicitar os dados a instituição para então por meio de algum mecanismo de importação, carregar os dados para o sistema.

Para usuários avançados, a ANA disponibiliza *Web Services*, que, como discutido anteriormente, possibilitam a integração de sistemas e a comunicação entre aplicações diferentes, que neste caso possibilitará que o SSD possa consumir os dados do sistema de informações da ANA, o HidroWeb. Os dados via *Web Services* são intercambiáveis via XML e a especificação WSDL (*Web Services Description Language*), que é um padrão também baseado em XML, ele descreve todo o serviço e traz os métodos do *Web Service* para serem

utilizados na aplicação que utiliza o serviço. A Figura 3.30 apresenta as *units* que compõem o ANA App.

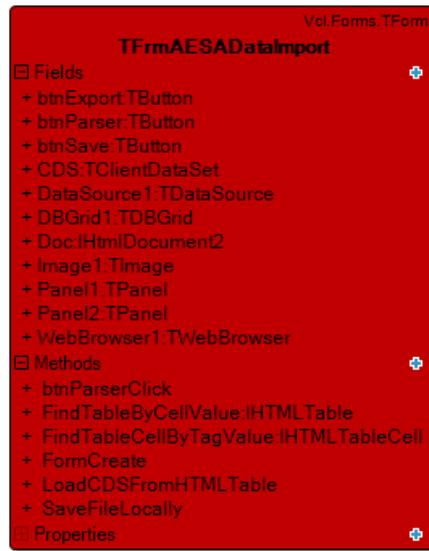
Figura 3.30 - *Units* que compõem o ANA App.



Fonte: Autor.

A principal *unit* da ANA App é uMMA (Ministério do Meio Ambiente), nela estão as classes e métodos que possibilitam a requisição dos dados junto ao servidor da ANA. Na *unit* uDM está a classe que realiza a conexão com o banco de dados do SSD e possibilitará, caso necessário, gravar os dados no servidor. A interface com o usuário está presente na uFrmMain e a *unit* Iso8601Unit contém a classe Tiso8601 que possui métodos para realizar conversões de padrões de data em arquivos XML definidos pela W3C (*World Wide Web Consortium*). A Figura 3.31 apresenta diagramas de classe das classes presentes na *unit* uMMA.

Figura 3.32 - Diagrama de classe da classe TFrmAESADaImport.



Fonte: Autor.

Capítulo 4

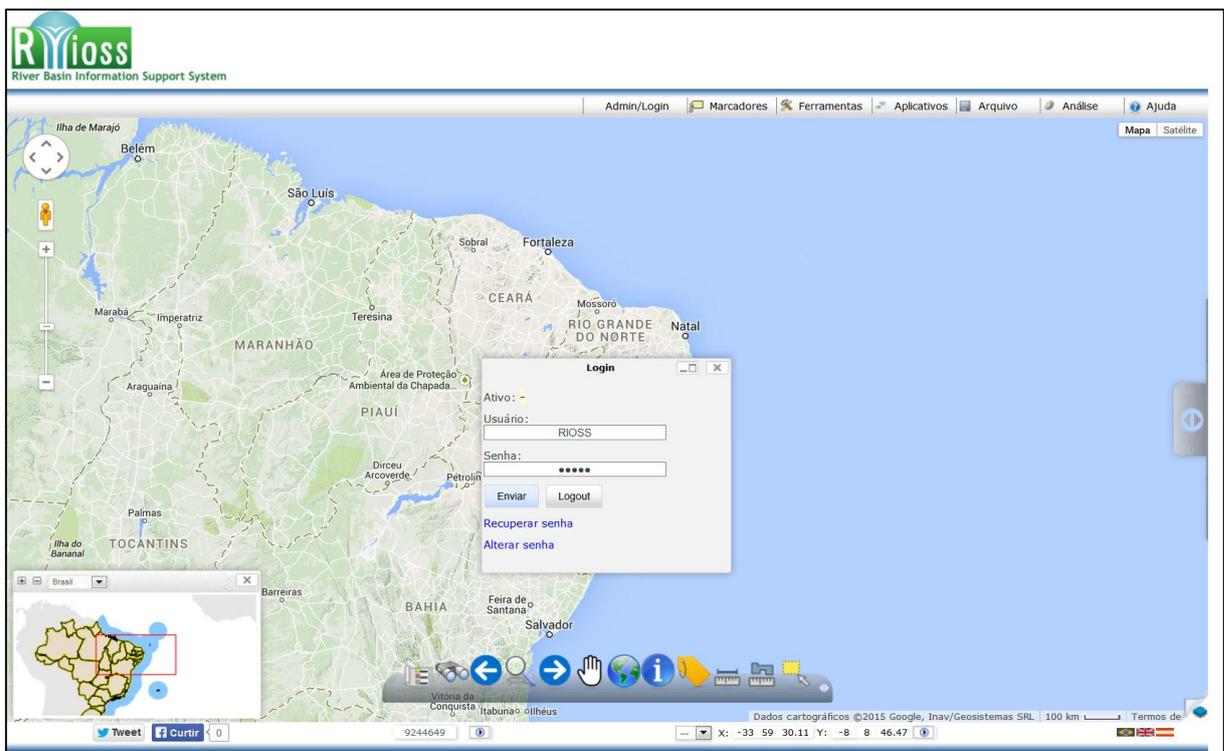
4 Resultados e Discussão

Neste capítulo serão apresentados os resultados do processo de modelagem e desenvolvimento das partes integrantes do SSD. São apresentadas as interfaces criadas e funcionalidades do sistema como: acesso ao SIG e operações básicas, interface geral do SSD e as telas de cadastro dos módulos do sistema, funcionamento dos aplicativos para acesso a dados externos e interface do modelo integrado ao SSD. São apresentadas algumas aplicações e exemplos de cadastros no sistema e análises simples. Também é verificado por meio de uma aplicação do desempenho do modelo integrado ao SSD comparado ao modelo original com intuito de validar a abordagem do sistema de gerenciamento de modelos.

4.1 Sistema de Informações Geográficas do SSD

Após as configurações e customizações do *framework* i3Geo a interface inicial do SIG é apresentada na Figura 4.1.

Figura 4.1 - Interface inicial do SIG.



Fonte: Autor.

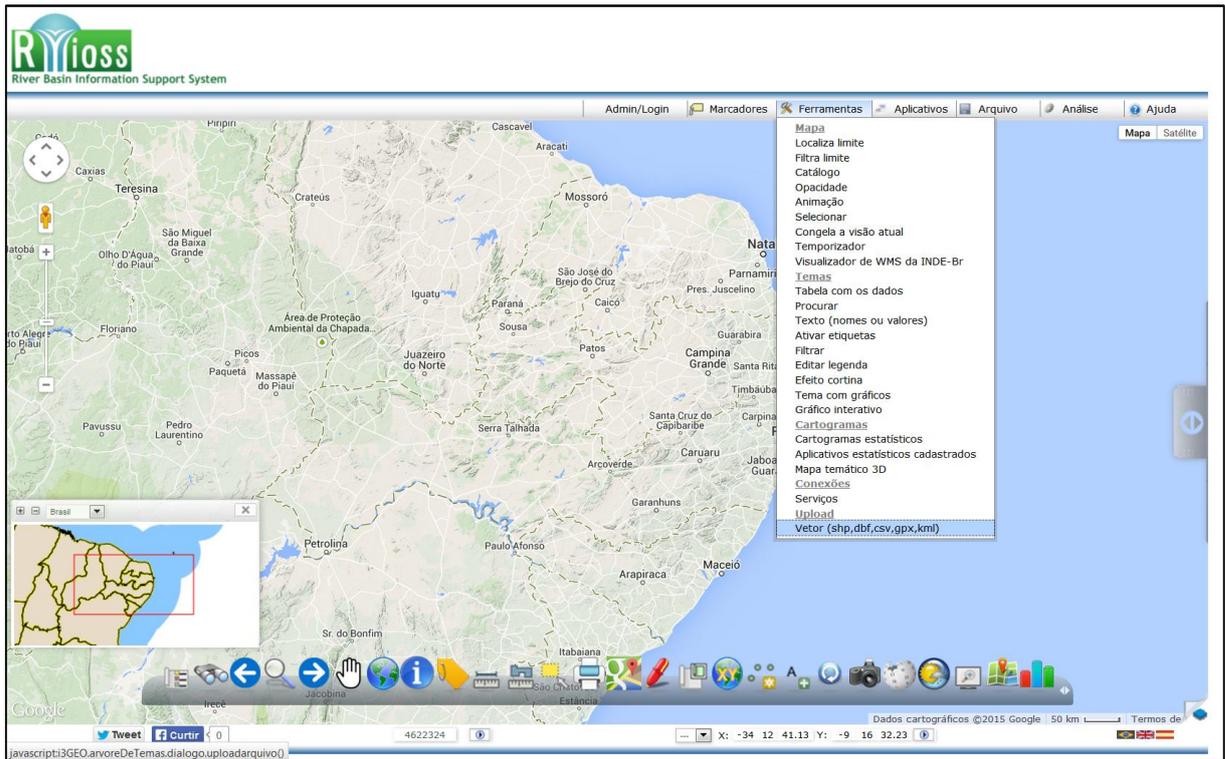
Na interface inicial do SIG apresentada na Figura 4.1 é possível identificar os principais controles para interação com o usuário como o menu, barra de botões e a guia móvel (escondida). Outros componentes presentes são o mapa de referência e a barra de status onde estão presentes a interação com redes sociais, escala numérica, seletor de idioma e coordenadas. As coordenadas podem ser visualizadas no Sistema de Coordenadas Geográficas e UTM (*Universal Transversa de Mercator*).

A tela de login apresentada refere-se ao sistema para controle de usuários e permissões do SIG. É possível definir ações chamadas de “operações” que recebem um código. Cada operação pode ser executada por usuários que possuem determinados papéis. Por exemplo, apenas os usuários com o papel de “editor” podem alterar algum MapFile. Este sistema de autenticação apenas é necessário quando o SIG é acessado fora do SSD, ou seja, quando não é realizada a autenticação prévia.

Por exemplo, o administrador pode acessar remotamente o SIG e pode também disponibilizar um link para outro usuário acessar sem necessariamente estar no SSD.

A barra de botões do tipo olho de peixe pode se expandir apresentando os outros botões programados. Foi definido um padrão com 12 botões. A Figura 4.2 apresenta outra visão do SIG apresentando a barra de botões completa e o menu ferramentas.

Figura 4.2 - Barra de botões e menu ferramentas do SIG.

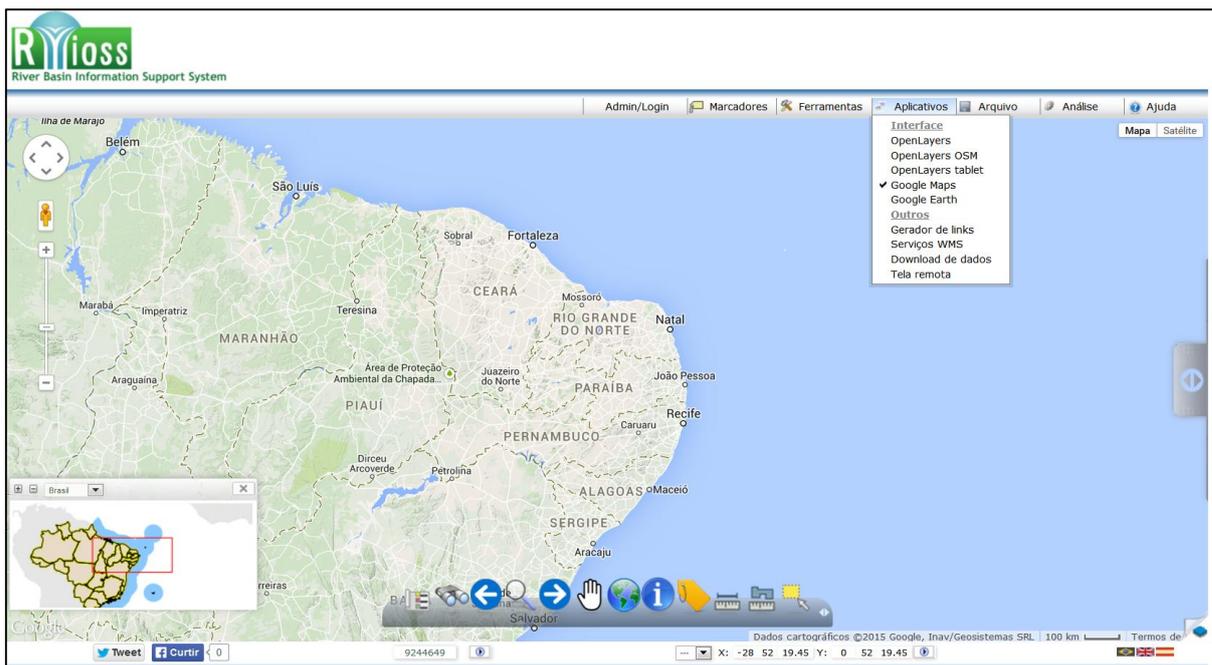


Fonte: Autor.

Os botões presentes na barra de botões possibilitam realizar ações básicas de interação com o mapa, como arrastar e soltar, visão geral, obter informações sobre as camadas e selecionar objetos. Também estão presentes botões para realizar cálculos de distância e área e edição vetorial. Outras funções interessantes nos botões possibilitam buscar informações com base nos atributos das camadas no Wikipédia e imagens relacionadas na web.

No menu de ferramentas estão funções para trabalhar e customizar o mapa e temas/camadas, como definir opacidade, legendas, etiquetas/*labels* e realizar pesquisas com base nos atributos das camadas. Também é possível realizar *uploads* de arquivos para serem visualizados e realizar conexões a serviços de dados geográficos em tempo de execução do SIG sem necessidade de configurações adicionais em MapFiles ou banco de dados. É também possível gerar gráficos com base nos atributos das camadas diretamente sobre elas. A Figura 4.3 apresenta o menu de aplicativos do SIG.

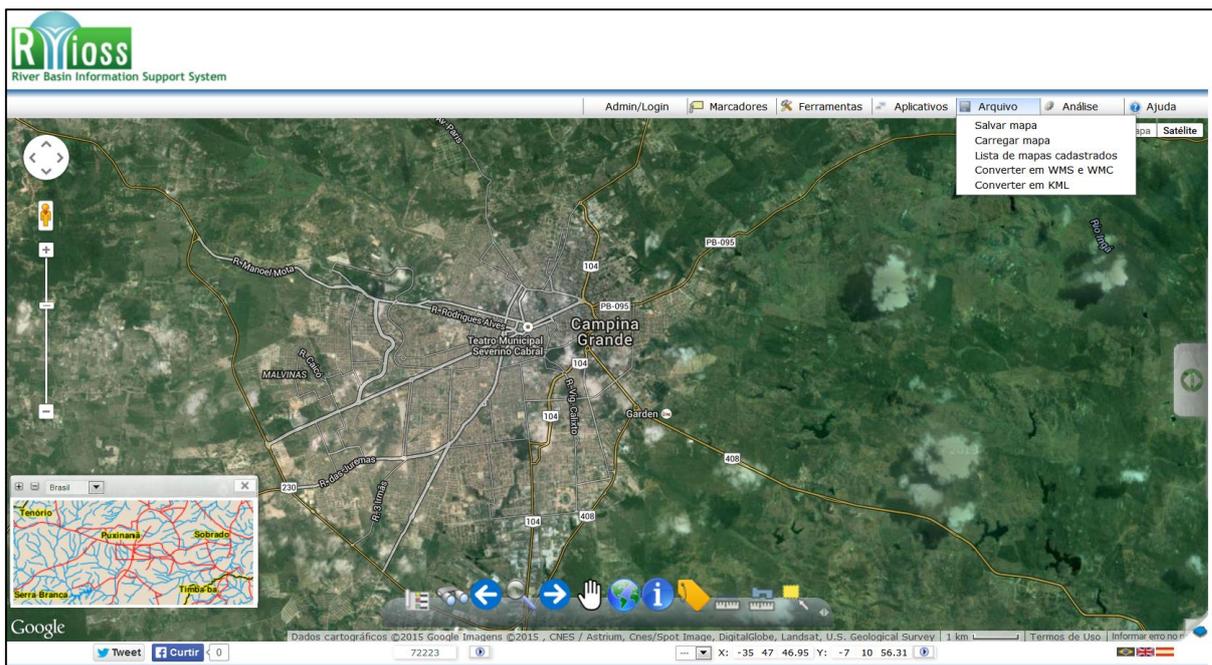
Figura 4.3 - Menu Aplicativos do SIG.



Fonte: Autor.

No menu de aplicativos é possível mudar a interface do mapa base escolhendo entre as APIs disponíveis. É possível gerar links para compartilhá-los e disponibilizar uma tela remota do mapa em apresentação para ser visualizado em outro monitor, muito útil para apresentações. A Figura 4.4 apresenta o menu arquivo e a interface do Google Maps com camada imagem de satélite.

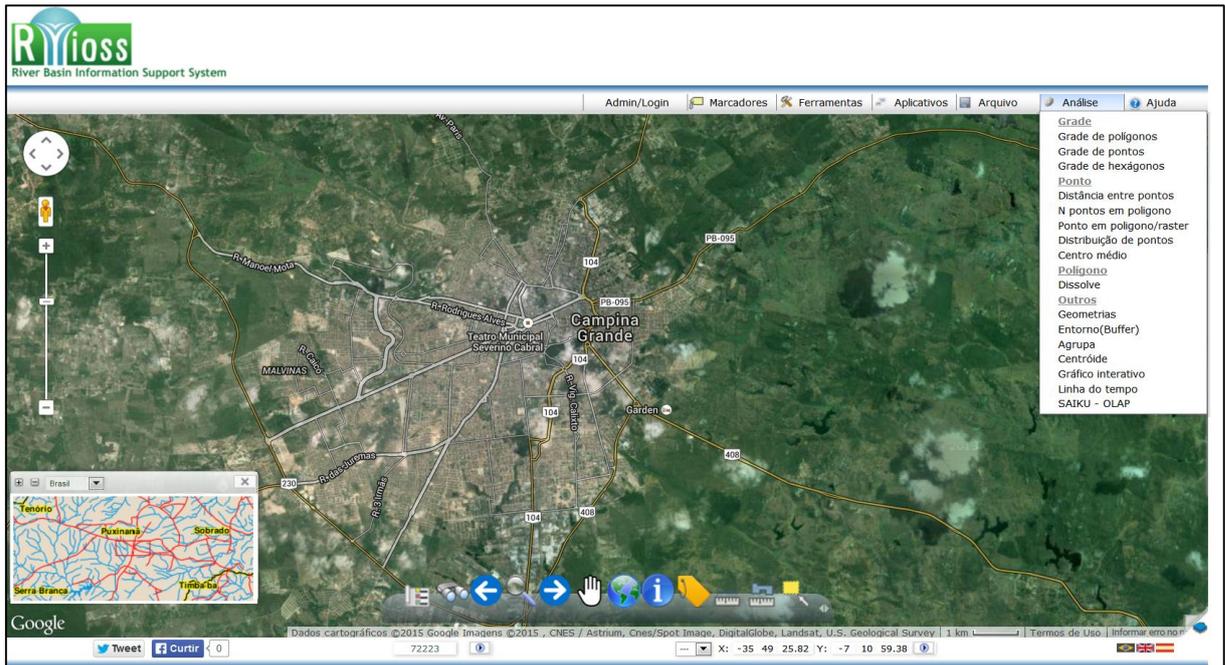
Figura 4.4 - Menu arquivo do SIG.



Fonte: Autor.

No menu arquivo é possível salvar o mapa que foi trabalhado, carregar os mapas salvos, listar todos os mapas cadastrados e converter o mapa atual para ser consumido, por exemplo, via serviço WMS ou arquivo KML (*Keyhole Markup Language*). A Figura 4.5 apresenta o menu análise do SIG.

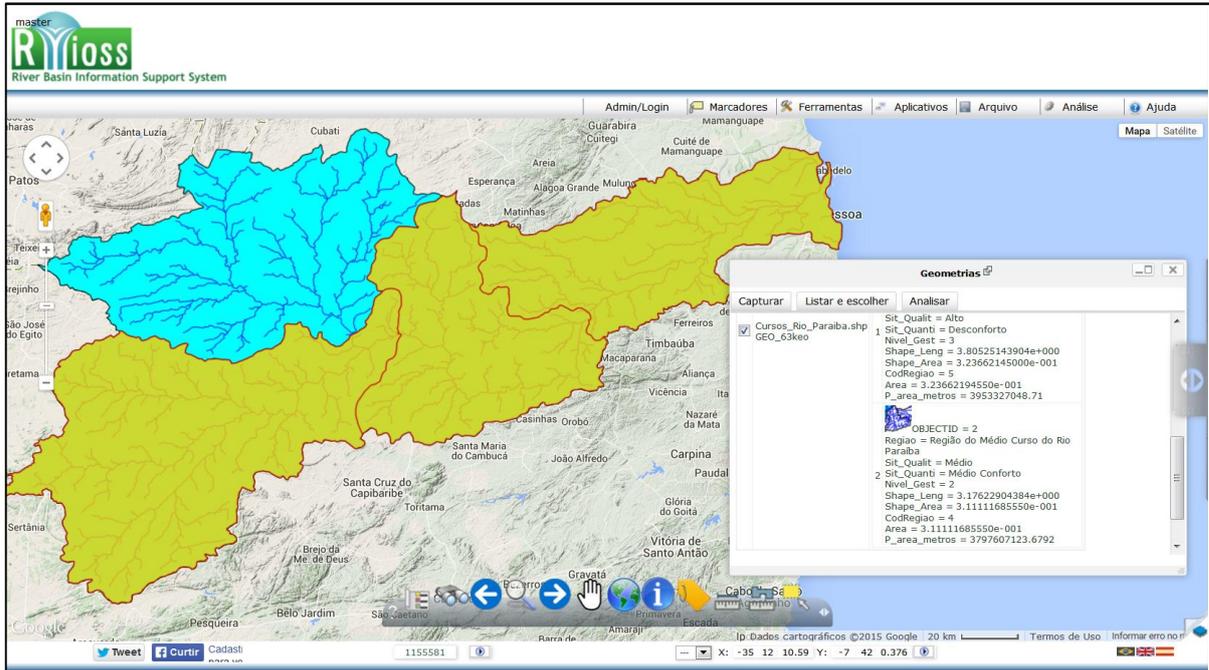
Figura 4.5 - Menu análise do SIG.



Fonte: Autor.

No menu análise estão as operações espaciais que podem ser realizadas no SIG, entre elas estão: grades, distância entre pontos, distribuição de pontos, centroide, centro médio, *dissolve*, *buffer* e agrupar. Há também a possibilidade de construir gráficos interativos de diversos tipos utilizando os atributos das camadas e exportá-los. Algumas dessas operações são realizadas utilizando bibliotecas do R, sendo o mesmo configurado previamente no servidor. A Figura 4.6 apresenta o cálculo de área utilizando a opção Geometrias.

Figura 4.6 - Cálculo de geometrias.

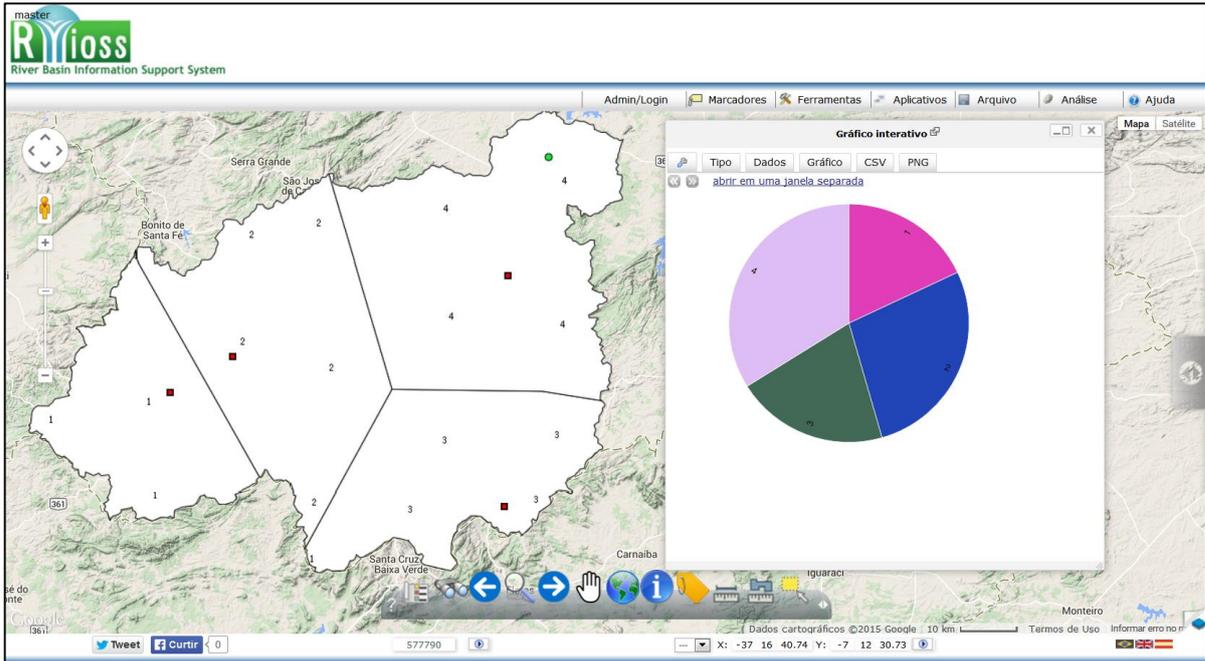


Fonte: Autor.

Na Figura 4.6 a opção Geometrias possibilita o cálculo das áreas correspondentes as regiões do curso do Rio Paraíba.

A Figura 4.7 apresenta gráfico pizza gerado para as áreas de influência de alguns postos pluviométricos localizados na bacia de drenagem do Rio Piancó antes de ser barrado pelo reservatório Coremas no estado da Paraíba, utilizando a opção Gráfico Interativo no menu análise.

Figura 4.7 - Geração de gráficos.



Fonte: Autor.

A Figura 4.8 apresenta guia do tipo sanfona, que a mostra onde estão presentes a visualização das camadas, o catálogo de dados, a legenda e os links para os mapas salvos cadastrados.

Figura 4.8 - Guia sanfona do SIG.



Fonte: Autor.

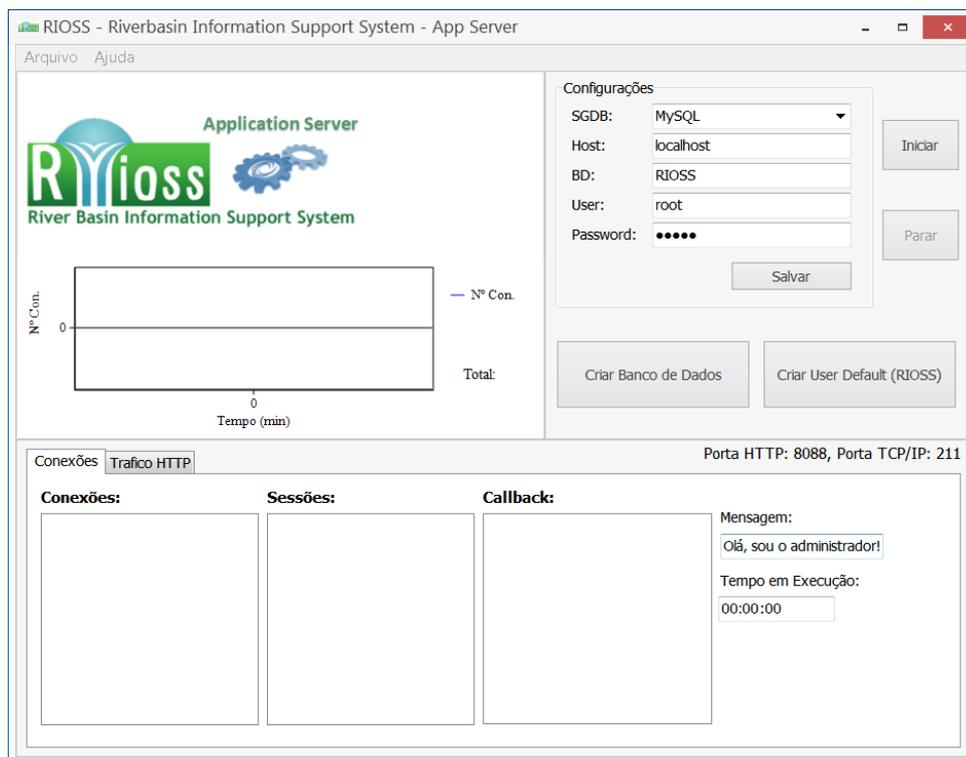
A guia sanfona é interessante pois possibilita uma área maior de visualização do mapa na tela. Nela é possível visualizar as camadas presentes no mapa, editar legendas e acessar os mapas salvos anteriormente. Nela também está o catálogo de acesso aos dados que incluem os dados no servidor gerenciados e não gerenciados pelo SGDB e os dados advindos de serviços de mapas.

A Figura 4.8 também apresenta o acesso aos dados disponibilizados pelos serviços da INDE (Infraestrutura Nacional de Dados Espaciais), que é um projeto do governo federal que tem como propósito catalogar, integrar e harmonizar dados espaciais existentes nas instituições do governo brasileiro, produtoras e mantenedoras desse tipo de dado, de maneira que possam ser facilmente localizados, explorados e acessados para os mais diversos usos, por qualquer cliente que tenha acesso à Internet. Assim em termos de acesso a dados espaciais de bancos externos pelo SIG a grande facilidade e inúmeras fontes possíveis.

4.2 Servidor de Aplicações

O servidor de aplicações é responsável por intermediar a interação do usuário e as ações de acesso e manipulação de dados disponibilizados pelo servidor de dados, além disso o mesmo permite gerenciar as conexões e sessões com os clientes, entre outras atividades. A Figura 4.9 apresenta a interface do servidor de aplicações.

Figura 4.9 - Interface do servidor de aplicações.

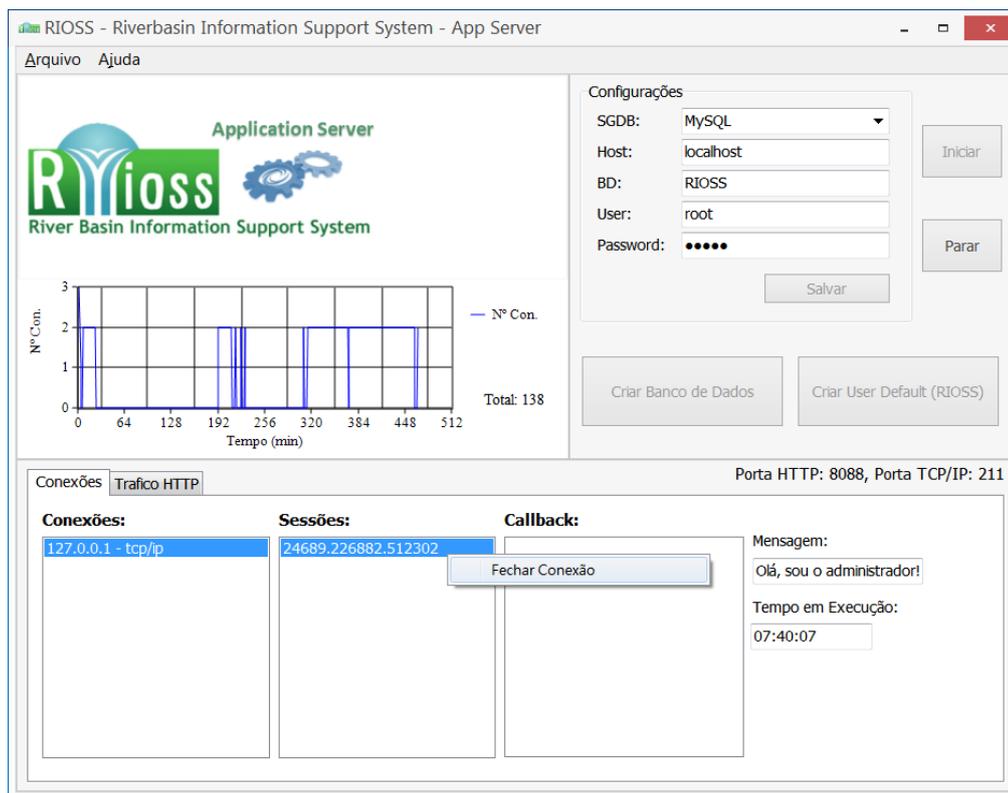


Fonte: Autor.

Geralmente o software do servidor de aplicações não possui interface com o usuário, o sistema funciona como um serviço na máquina servidora que é iniciado junto a inicialização do sistema operacional. Porém, para facilitar o gerenciamento do sistema, foi criada uma interface que permite o administrador realizar configurações de acesso ao servidor de dados, inicialização do servidor e monitoramento das conexões com os clientes (HTTP e TCP/IP).

Na interface apresentada na Figura 4.9 ainda é possível por meio de dois botões criar o banco de dados no SGDB e o usuário padrão para acesso do cliente. A Figura 4.10 apresenta o servidor de aplicações em execução.

Figura 4.10 - Servidor de aplicações em execução.

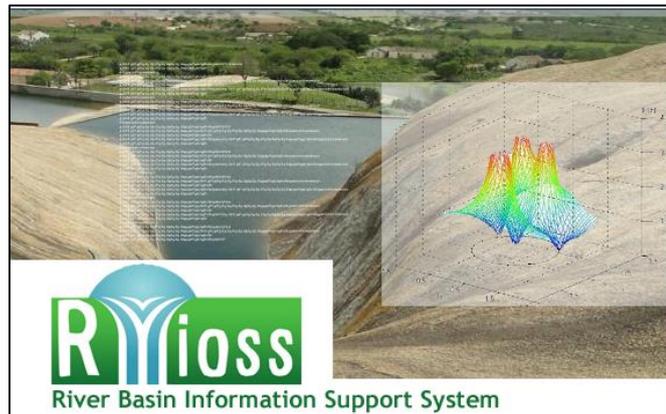


Fonte: Autor.

Na Figura 4.10, na aba Conexões é apresentado o endereço IP da máquina cliente que está conectada e o ID gerado da sessão criada para aquele usuário, onde também é possível encerrar a conexão com o cliente. Ainda é possível num gráfico observar o histórico de conexões durante o tempo de execução do servidor. O servidor também foi configurado para monitorar as futuras conexões dos clientes web via protocolo HTTP.

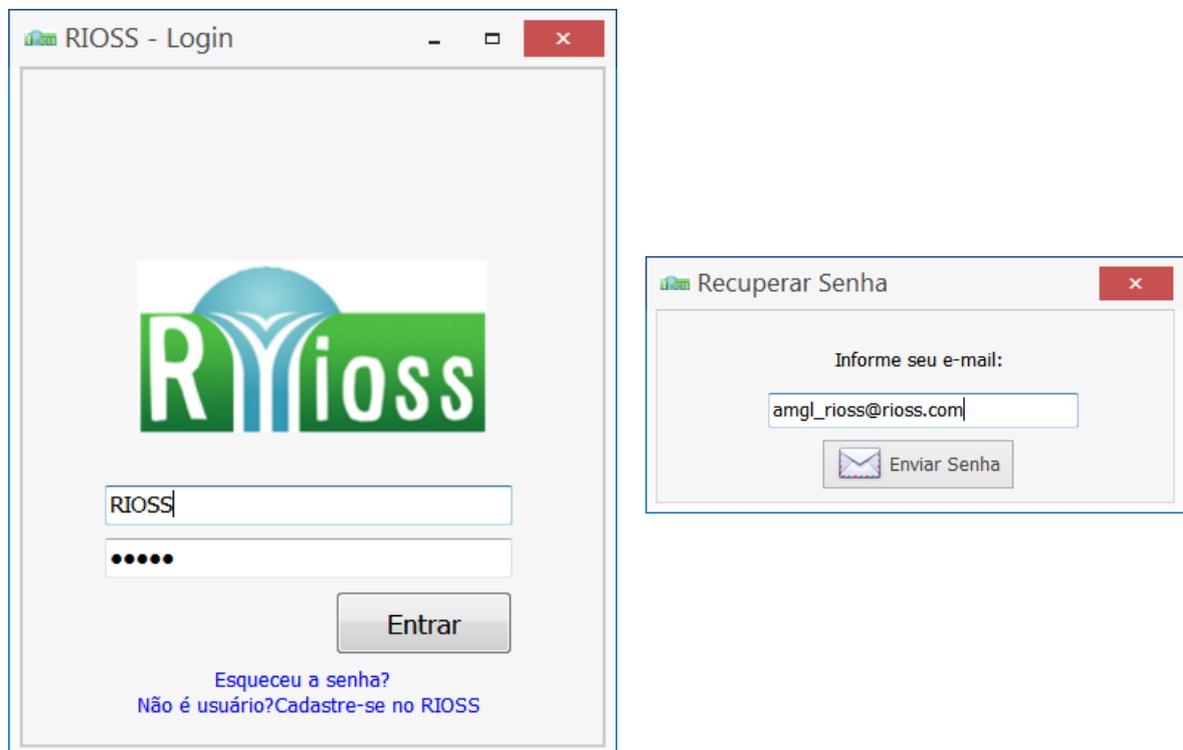
4.3 Cliente Desktop

A primeira tela apresentada ao SSD ser iniciado é a tela de abertura, ela apresenta informações do sistema ao mesmo tempo que a interface principal e outros componentes são carregados num processo chamado de *multithreading*. A Figura 4.11 apresenta a tela de abertura do SSD.

Figura 4.11 - Tela de abertura do SSD.

Fonte: Autor.

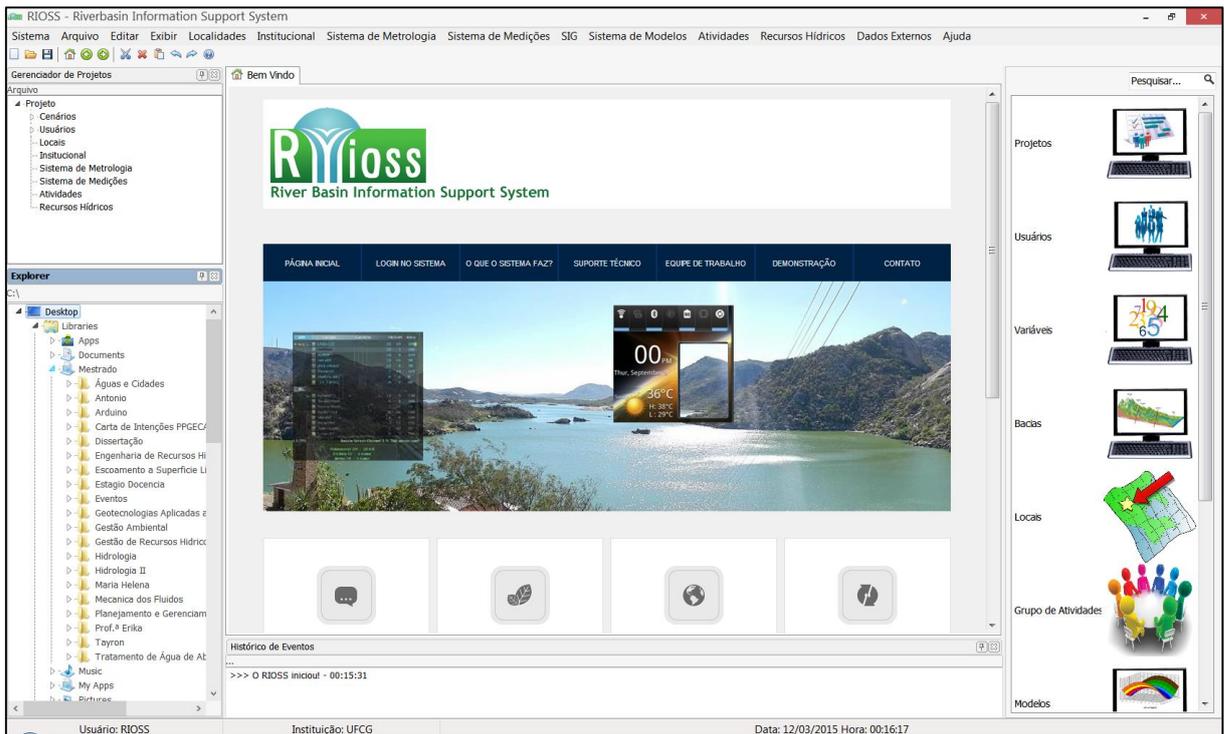
A primeira atividade do sistema é a realização do login pelo usuário cadastrado no banco de dados. O sistema possui uma conta padrão com todos os privilégios que pode iniciar um projeto e cadastrar novos usuários. Também é possível recuperar a senha do usuário, sendo a mesma enviada para o e-mail cadastrado no sistema. A Figura 4.12 apresenta a tela de login e recuperação de senha do SSD.

Figura 4.12 - Tela de login e recuperação de senha do SSD.

Fonte: Autor.

A Figura 4.13 apresenta a tela principal do SSD.

Figura 4.13 - Tela principal do SSD.



Fonte: Autor.

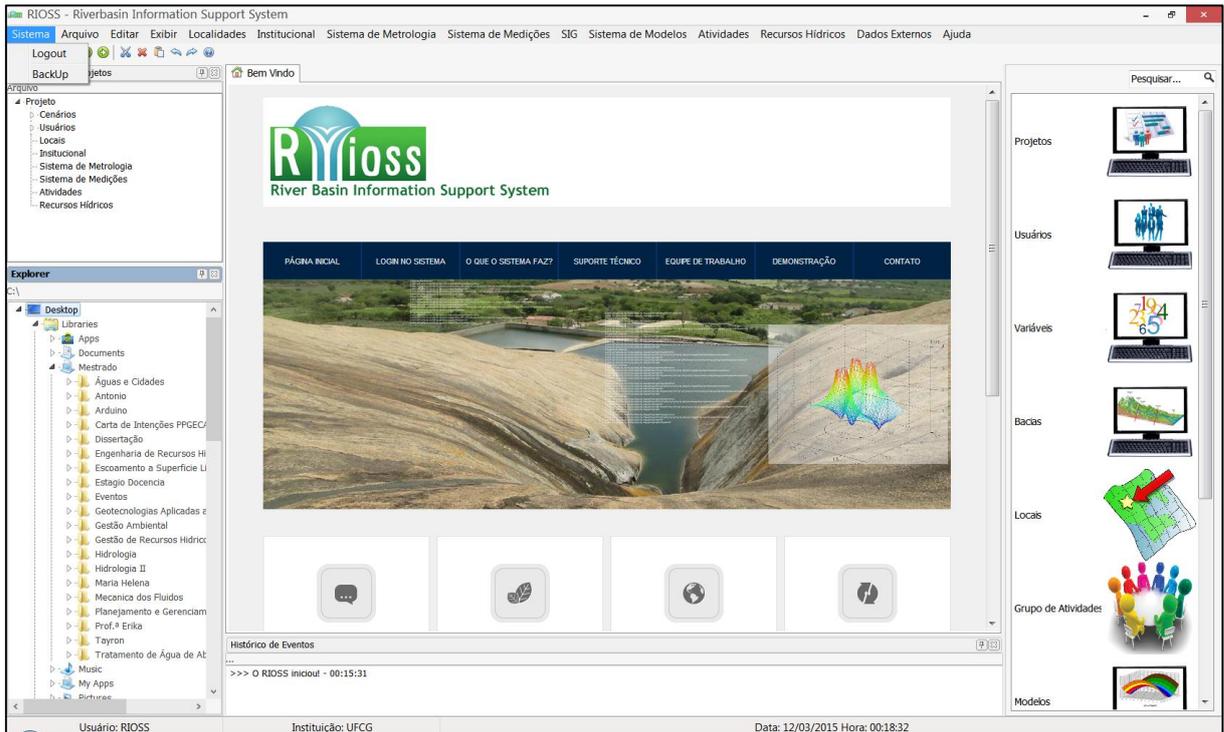
Na tela principal do sistema apresentada na Figura 4.13 há uma aba de boas-vindas a qual permite o acesso à interface web do RIOSS. Existem ainda mais quatro janelas, são elas: Gerenciador de Projetos, Explorer, Acesso rápido e Histórico de Eventos. O Gerenciador de Projetos apresenta a estruturação do projeto e o acesso as suas partes. No Explorer é possível ter acesso aos arquivos do disco onde o SSD está instalado. Na janela de Histórico de Eventos são disparadas mensagens relatando eventos que ocorrem no sistema. Finalmente na janela de atalhos é possível acessar as demais telas do sistema e inclusive filtrar as funcionalidades.

Todas as janelas podem ser movidas e alinhadas, além de escondidas e transformando-se em abas, conforme desejo do usuário, possibilitando a customização da interface. A barra de status apresenta informações como: usuário que fez o login, instituição que pertence, data e hora.

É importante salientar que o objetivo principal deste trabalho foi definir uma estrutura e arquitetura de um SSD para recursos hídricos, a interface desktop não teve todos os seus módulos implementados, mas apenas os necessários para realizar a validação do sistema.

Ainda na tela principal está o menu para acesso aos diversos módulos do sistema. Na Figura 4.14 é apresentado o menu sistema.

Figura 4.14 - Menu Sistema.

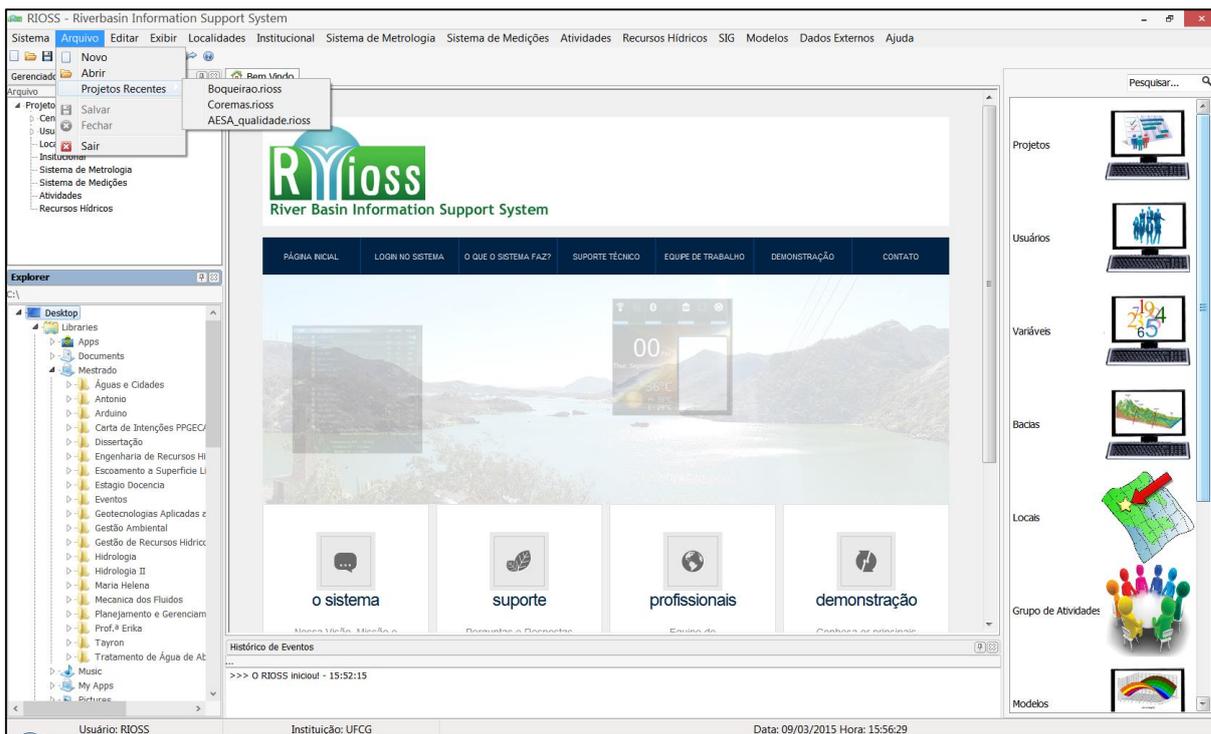


Fonte: Autor.

No menu Sistema é possível realizar duas operações: o Logout e o Backup. Quando acionado o Logout são realizadas verificações do status do projeto em andamento, salvamento de dados no banco pelo sistema *logger*, finalização da sessão atual do usuário e finalmente retorno a tela de Login. A opção Backup só pode ser realizada pelo administrador geral do sistema e só é utilizado em situações extremas ou quando o DBA por ventura necessitar configurar um servidor momentâneo, a opção gera um arquivo SQL com uma cópia banco do dados para ser acessada remotamente.

A Figura 4.15 apresenta o menu Arquivo onde é possível criar e abrir projetos salvos.

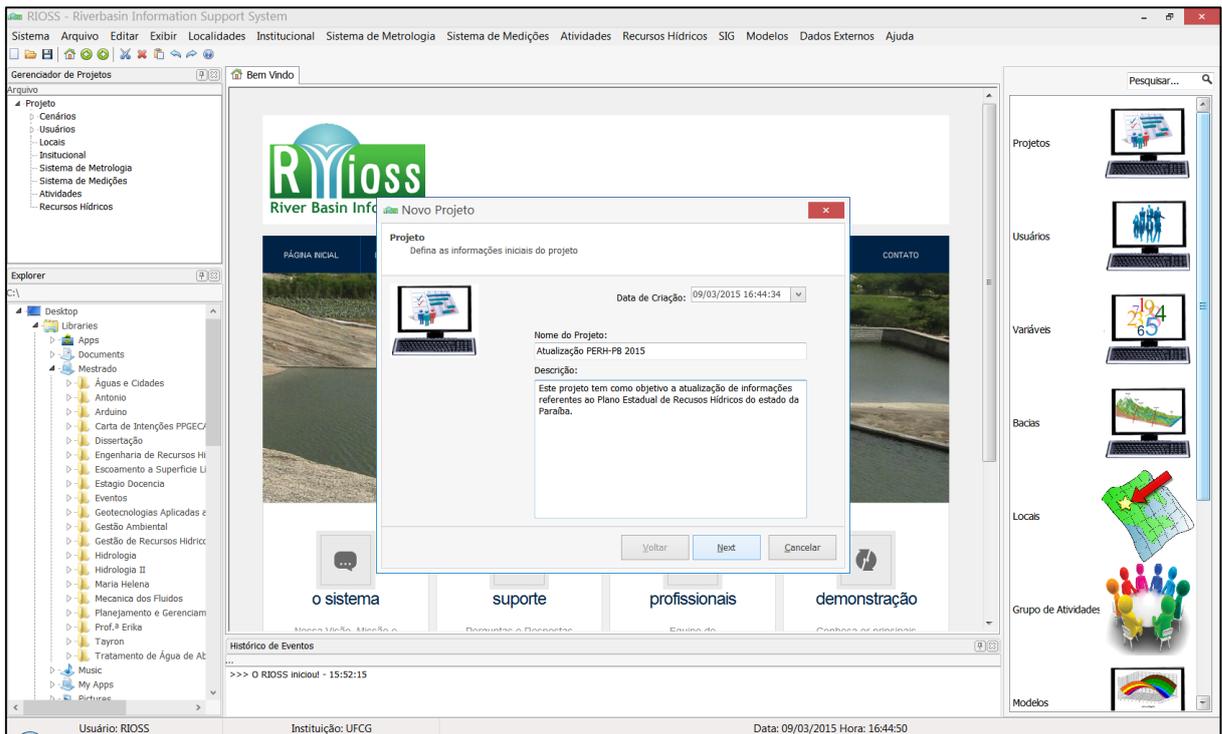
Figura 4.15 - Menu Arquivo.



Fonte: Autor.

A opção Novo é o ponto de partida para criação de algum projeto no sistema para então ter acesso aos demais módulos. Ao clicar em Novo é aberta uma janela com um assistente para criação de um projeto para pode ser trabalhado. A Figura 4.16 apresenta o assistente para criação de projeto.

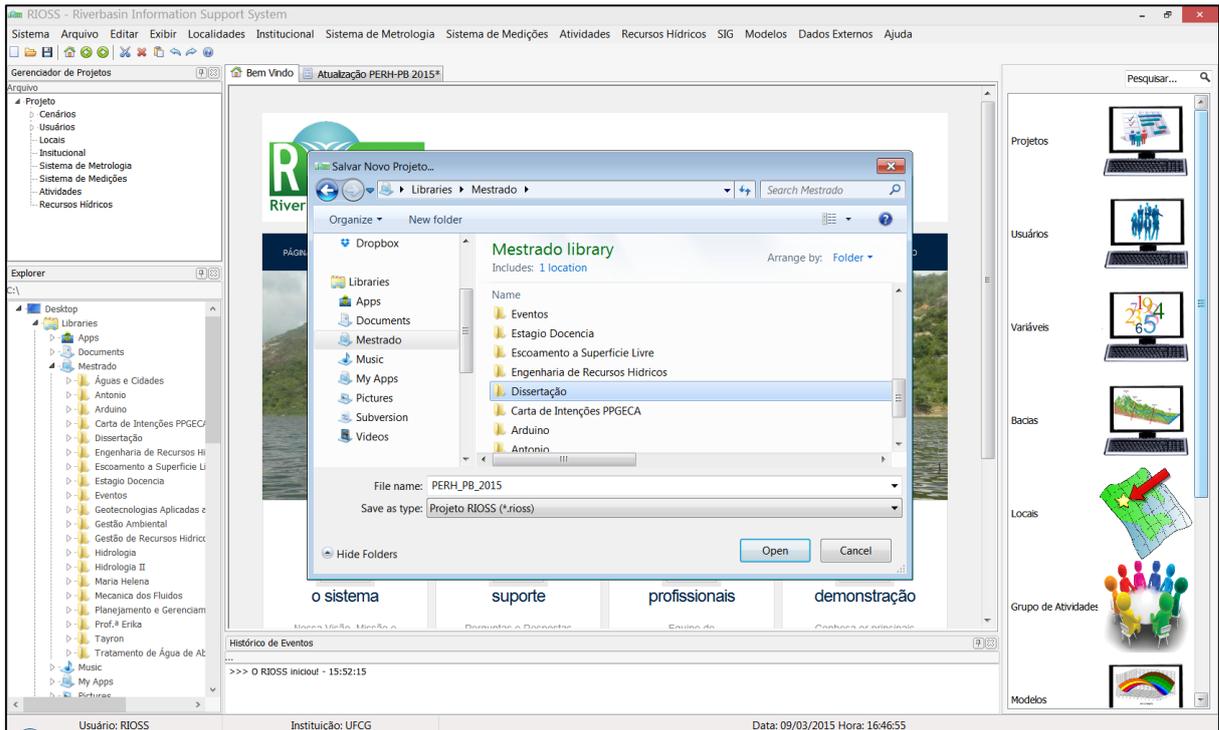
Figura 4.16 - Criando novo projeto.



Fonte: Autor.

O novo projeto pode então ser salvo, onde é criado um arquivo com extensão .RIOSS com informações para serem carregadas quando o projeto for iniciado. A Figura 4.17 apresenta a tela de diálogo para salvar o projeto.

Figura 4.17 – Salvando projeto.



Fonte: Autor.

O SSD ainda não possui um sistema de instalação para configurar o mesmo no computador o qual está sendo utilizado. Dessa forma, na primeira vez que o SSD é iniciado é realizado um procedimento que registra a extensão .RIOSS no registro da máquina, associando ao arquivo binário do sistema e ao ícone padrão, assim possibilitando os arquivos serem abertos diretamente pelo SSD.

4.3.1 Localidades

No menu localidades é possível cadastrar registros associados a localização e delimitação espacial no qual o projeto está inserido. A Figura 4.18 apresenta a tela de cadastro do item Estados no menu localidades.

Figura 4.18 - Item Estados no menu localidades.

RIOSS - Estados

Pesquisar: Nome 



Nome:

Sigla:

País:

Comentário:

Nome	Sigla	País
▶ Acre	AC	Brasil
Alagoas	AL	Brasil
Amapá	AP	Brasil
Amazonas	AM	Brasil

27 Registros cadastrados.

Fonte: Autor.

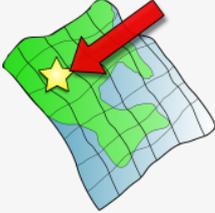
No item Estados é possível cadastrar os estados que integram o projeto ou delimitação administrativa semelhante a qual dependerá do país em questão. A Figura 4.19 apresenta a tela de cadastro de Tipos de Locais no menu localidades.

Figura 4.19 - Item Tipos de Locais no menu localidades.

RIOSS - Tipos de Locais

Pesquisar: Nome 

 Novo  Editar  Apagar  Sair



Nome:

Sigla:

Tipo: Urbano Rural

Comentário: Uma cidade ou urbe é uma área urbanizada, que se diferencia de vilas e outras entidades urbanas.

 Salvar  Cancelar

Nome	Sigla	Tipo	Comentário
▶ Cidade	C	U	Uma cidade ou urbe é uma área urbanizada, que se diferencia de vilas e c
Fazenda	FA	R	Área rural geralmente possuidora de atividades produtivas no ramo agr

2 Registros cadastrados.

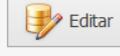
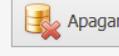
Fonte: Autor.

Na tela Tipos de Locais é possível cadastrar um tipo de local que especifica o local ou locais aonde o projeto está sendo executado. A Figura 4.20 apresenta a tela de cadastro do item Locais no menu localidades.

Figura 4.20 - Item Locais no menu localidades.

RIOSS - Locais

Pesquisar: Nome 

 Novo  Editar  Apagar  Sair



Nome:

Latitude:

Sigla:

Longitude:

Comentário:

Altitude:

Bacia Hidrográfica: 

Estado: 

Tipo: 

Rio: 



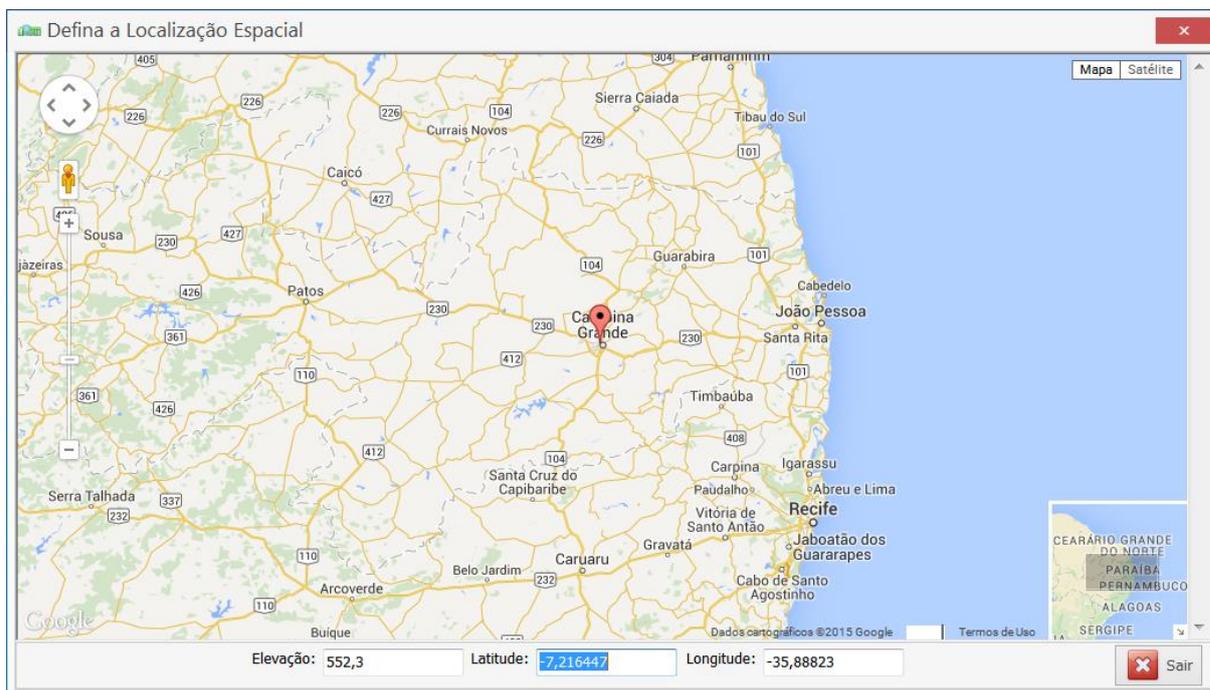
 Salvar  Cancelar

Nome	Sigla	Latitude	Longitude	Altitude	Comentário
Camalaú		-7,893542	-36,788875	0	Cidade localizada no Borborema na r
▶ Campina Grande	CG	-7,216447	-35,88823	552,3	Cidade localizada no Agreste Paraíba
Capim		-6,538543	-35,51827	0	Cidade localizada no Mata Paraibana
Caraúbas		-6,900824	-35,169876	0	Cidade localizada no Borborema na r

223 Registros cadastrados.

Fonte: Autor.

Na tela de cadastro apresentada na Figura 4.20 é possível cadastrar de fato o local ou os locais de abrangência do projeto. Na tela é associado ao local o estado, tipo de local, bacia hidrográfica e rio. Também é possível utilizar uma ferramenta para obter dados espaciais como latitude, longitude e elevação. Ao clicar no botão de localização espacial uma nova janela é aberta com uma interface do SIG customizada (*mashup*) onde o usuário pode escolher a localização atualizando automaticamente as informações no formulário de cadastro. A Figura 4.21 apresenta a tela de acesso rápido ao SIG para definição da localização espacial.

Figura 4.21 - Tela de definição da localização espacial.

Fonte: Autor.

4.3.2 Institucional

No menu Institucional é possível cadastrar instituições, usuários e clientes que farão parte dos projetos. Também é possível cadastrar documentos relativos as instituições e clientes. A Figura 4.22 apresenta a tela de cadastro do item Instituições no menu institucional.

Figura 4.22 - Item Instituições no menu institucional.

RIOSS - Instituições

Pesquisar: 


Nome: AESA/CG - Agência **Estado:** Paraíba

Endereço: Av. Aprígio Veloso, **Local:** Campina Grande

CEP: 58109-970 **Comentário:** A AESA foi criada pela Lei nº 7.779, de 07/07/2005, sob a forma jurídica de uma Autarquia,

ID	Nome
7	AESA/CG - Agência Executiva de Gestão das Águas do Estado da Paraíba
8	ANA - Agência Nacional de Águas
5	DNPM - DEPARTAMENTO NACIONAL DE PRODUÇÃO MINERAL
1	EMBRAPA - EMPRESA BRASILEIRA DE PESQUISA AGROPECUÁRIA

8 Registros cadastrados.

Fonte: Autor.

Na tela Instituições é possível cadastrar diversas instituições participantes de um projeto e associa-las a um local num determinado estado. A Figura 4.23 apresenta a tela de cadastro do item Clientes no menu institucional.

Figura 4.23 - Item Clientes no menu institucional.

RIOSS - Clientes

Pesquisar: Nome 

 Novo  Editar  Apagar  Sair



Nome: Estado:

Endereço: Local:

CEP: Instituição:

Caixa Postal:

Sexo: Feminino Masculino

Comentário:

 Cadastrar Documento  Salvar  Cancelar

Nome	Sexo	Endereço	CEP	Caixa F
▶ Artur Moises Gonçalves Lourenço	M	Rua Maria Minervina	58802-280	
Wilson Fadlo Curi	M	Rua Aprigio Veloso	58802-480	

2 Registros cadastrados.

Fonte: Autor.

Na tela de cadastro Clientes é possível cadastrar todo o pessoal físico e jurídico, participantes em um projeto, bem como os seus respectivos documentos. A Figura 4.24 apresenta a tela de cadastro do item Usuários no menu institucional.

Figura 4.24 - Item Usuários no menu institucional.

RIOSS - Usuários

Pesquisar: Login

Novo Editar Apagar Sair

Nome: Artur_RH

Senha: ●●●●●●

Comentário: Usuário do projeto Boqueirão.

Cliente: Artur Moises Gonçalves Lou

Tipo:
 Administrador
 Manutenção
 Usuário

Salvar Cancelar

Login	Tipo	Cliente	Data de Criação	Comentário
▶ Artur_RH	M	Artur Moises Gonçalv	10/03/2015 14:47:32	Usuário do projeto Boqueirão.
RIOSS	M		10/02/2015 22:22:50	Usuário padrão.
Rosires	U		10/03/2015 14:48:58	Usuário do projeto Boqueirão.
WfCuri	M	Wilson Fadlo Curi	10/03/2015 14:48:53	Usuário do projeto Boqueirão.

4 Registros cadastrados.

Fonte: Autor.

Na tela de cadastro de usuários apresentada na Figura 4.24 é possível cadastrar os usuários que irão ter acesso ao login no sistema, definindo seu tipo, que irá acarretar seus privilégios, e associando a um cliente quando necessário.

4.3.3 Sistema de Metrologia

O sistema de metrologia do SSD permite cadastrar, caracterizar e compor variáveis de forma dinâmica segundo suas grandezas básicas ou derivadas. Também permite compor suas unidades de medida relativas, agrupa-las de forma conveniente, classifica-las e atribuir valores as suas classes para posterior análise, entre outras necessidades.

Outra atividade no sistema de metrologia é atribuir valores relativos a função utilidade associado a uma variável, dessa forma permitindo a análise do comportamento de uma variável em uma serie. A Figura 4.25 apresenta a tela de cadastro do item Grandezas no menu sistema de metrologia.

Figura 4.25 - Item Grandezas no menu sistema de metrologia.

RIOSS - Grandezas

Pesquisar: 



Nome:

Tipo:

Básica

Derivada

Dimensão Comprimento:

Dimensão Massa:

Dimensão Tempo:

Dimensão Temperatura:

Corrente Elétrica:

Dimensão Quantidade de Substancia:

Dimensão Luminosidade:

Comentário:

Tipo	Nome	Comentário
D	Trabalho, Energia, Calor, Quantidade De Calor	
D	Vazão Mássica	
▶ D	Vazão Volumétrica	Volume de um fluido por unidade de tempo.
D	Velocidade	

90 Registros cadastrados.

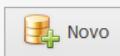
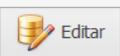
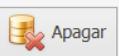
Fonte: Autor.

Na tela Grandezas é possível cadastrar qualquer grandeza para compor uma variável, definindo as dimensões segundo sistema internacional ou não, caso a grandeza for de outro sistema ou não estar em um sistema (DBO, DQO, etc.). A Figura 4.26 apresenta a tela de cadastro do item Unidades de Medida no menu sistema de metrologia.

Figura 4.26 - Item Unidades de Medida no menu sistema de metrologia.

RIOSS - Unidades de Medida

Pesquisar: Nome 

 Novo  Editar  Apagar  Sair



Grandeza: Vazão Volumétrica

Nome: Metro Cúbico por S

Símbolo: m³/s

Fator: 1

Comentário: Mensurar vazão volumétrica.

 Salvar  Cancelar

Grandeza	Nome	Símbolo	Fator	Comentário
Pressão,Tensão	Megapascal	Mpa	1E-6	Mensurar Pressão.
Comprimento	Metro	m	1	Mensurar comprimento.
Volume	Metro Cúbico	m ³	1	Mensurar Volume.
▶ Vazão Volumétrica	Metro Cúbico por Segundo	m ³ /s	1	Mensurar vazão volumétrica.

44 Registros cadastrados.

Fonte: Autor.

Na tela de cadastro apresentada na Figura 4.26 é possível cadastrar as unidades de medida de uma grandeza, cada unidade possui seu fator que será utilizado na conversão dos valores das variáveis quando for solicitada pelo usuário. A unidade do sistema internacional de medida é tomada com base (valor 1) e as outras unidades tem seus valores relativos a esta. A Figura 4.27 apresenta a tela de cadastro do item Grupo de Variáveis no menu sistema de metrologia.

Figura 4.27 - Item Grupo de Variáveis no menu sistema de metrologia.

RIOSS - Grupo de Variáveis

Pesquisar:

Nome 

Nome:

Comentário:

Nome	Comentário
Reservatórios	Conjunto de v
Hidrogeológicas	Conjunto de v
Limnológicas	Conjunto de v
▶ Qualidade de Água	Conjunto de v

6 Registros cadastrados.

Fonte: Autor.

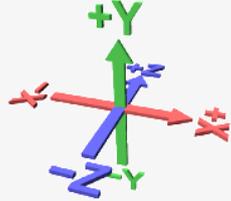
No item Grupo de Variáveis é possível definir grupos de variáveis que agrupam variáveis que tem relações em comum ou determinado propósito. A Figura 4.28 apresenta a tela de cadastro do item Variáveis no menu sistema de metrologia.

Figura 4.28 - Item Variáveis no menu sistema de metrologia.

RIOSS - Variáveis

Pesquisar: Nome 

 Novo  Editar  Apagar  Sair



Nome:

Grandeza:

Unidade de Medida:

Grupo de Variáveis:

Tipo:

Recurso

Custo

Benefício

Nenhum

Comentário:

Superfície contínua de água exposta à atmosfera e visível de uma determinada altitude

 Função Utilidade  Criar Classes  Salvar  Cancelar

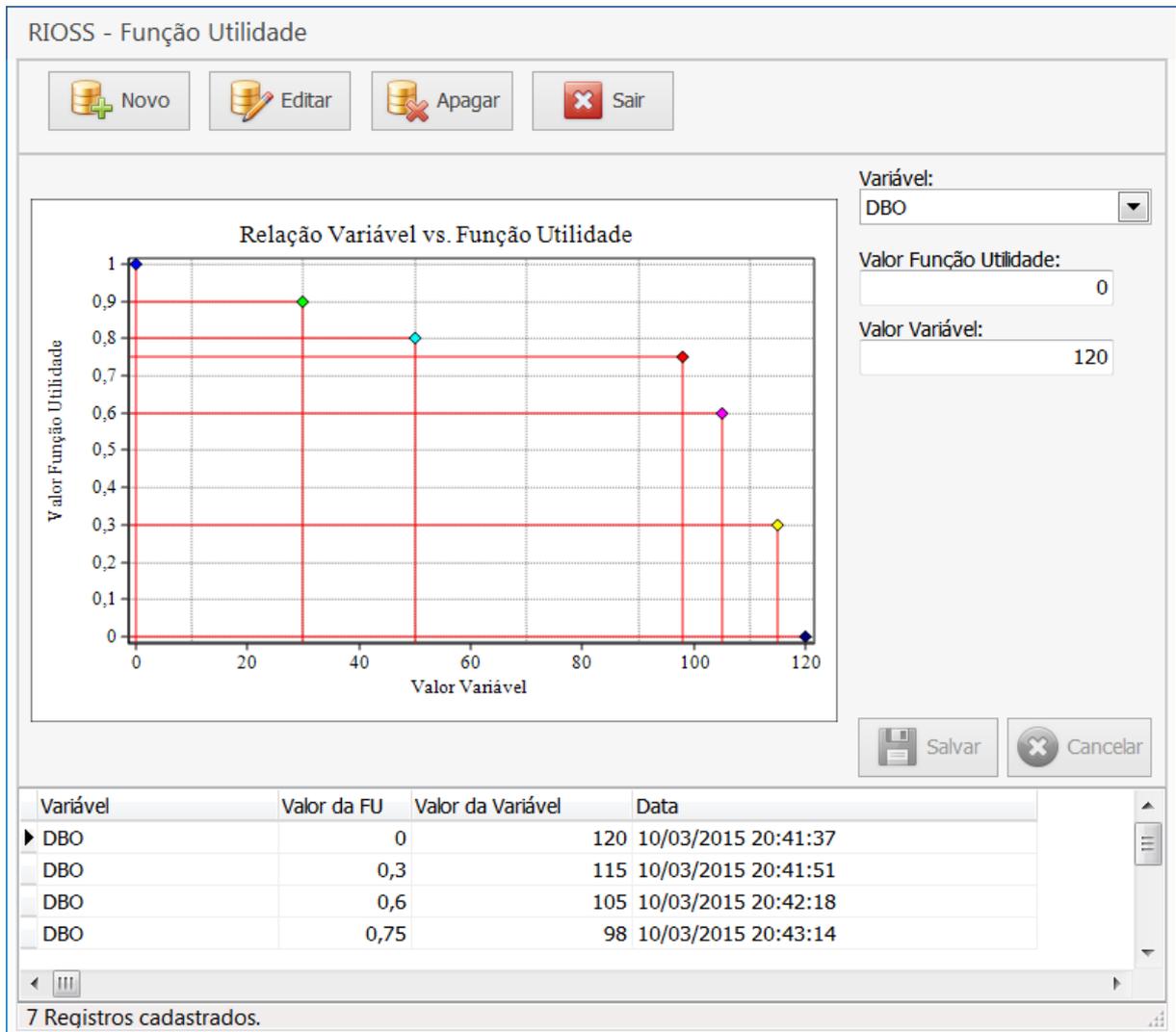
Nome	Tipo	Unidade de Medida	Grandeza	Grupo de Variav
Cota	N	Metro	Comprimento	Reservatórios
▶ Espelho d'Água	N	Metro Quadrado	Área	Reservatórios
Precipitação	N	Milímetro	Comprimento	Meteorológicas
Vazão	N	Metro Cúbico por Segundo	Vazão Volumétrica	Fluviométricas

5 Registros cadastrados.

Fonte: Autor.

Na tela apresentada na Figura 4.28 finalmente é possível cadastrar uma variável definindo sua grandeza, unidade e a qual grupo ela pertence. A unidade de medida definida geralmente é a unidade base cadastrada para aquela grandeza. Nesta tela além dos botões básicos existem mais dois que permitem cadastrar valores da função utilidade e criar classes para a variável escolhida. A Figura 4.29 apresenta a tela de cadastro Função Utilidade no item variáveis.

Figura 4.29 - Cadastro Função Utilidade no item variáveis.



Fonte: Autor.

Na tela Função Utilidade é possível cadastrar os valores da função utilidade, geralmente valores de 0 a 1, e os valores relativos da variável escolhida que são plotados para visualização num gráfico. A partir destes valores serão extraídas as relações quando realizada uma análise com uma serie temporal de alguma variável. No exemplo apresentado na Figura 4.29 é realizado um de cadastro para a variável DBO (Demanda Bioquímica de Oxigênio), onde, o valor mínimo da função utilidade é o valor máximo permitido da variável DBO segundo a resolução brasileira para lançamento de efluentes e o valor máximo da função o menor valor daquela variável (zero). A Figura 4.30 apresenta a tela de cadastro Classe de Variáveis no item variáveis.

Figura 4.30 - Cadastro Classe de Variáveis no item variáveis.

RIOSS - Classe de Variáveis

Variável:

Nome:

Comentário

Nome	Variável
▶ Conama Nº 430	DBO

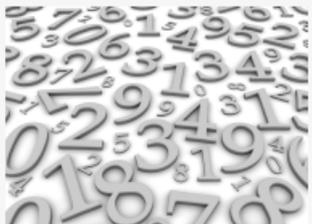
1 Registro cadastrado.

Fonte: Autor

Na tela de cadastro Classe de Variáveis pode-se criar uma classe para a variável escolhida e definir valores máximo e/ou mínimos. No exemplo apresentado na Figura 4.30 é realizado um cadastro de classe para a variável DBO, onde é criada uma classe denominada Conama Nº 430 a qual se refere a resolução brasileira que dispõe sobre padrões de lançamento de efluentes. Para definir valores de classe é necessário então clicar no botão Incluir Valores. A Figura 4.31 apresenta a tela de cadastro Valores de Classe.

Figura 4.31 - Cadastro Valores de Classe.

RIOSS - Valores de Classe



Classe de Variáveis:

Nome:

Valor Mínimo:

Valor Máximo:

Valor de Classe Associado a Função Utilidade:

Comentário

Nome	Classes de Variáveis	Valor Mínimo	Valor Máximo	Va
▶ DBO 5,20	Conama Nº 430		120	

1 Registro cadastrado.

Fonte: Autor.

Na tela de cadastro Valores de Classe é possível cadastrar os valores máximo e/ou mínimos para os intervalos de classe bem como denomina-los. Também é possível definir um valor para função utilidade associada. A Figura 4.32 apresenta um exemplo de cadastro de uma classe para a variável precipitação e seus valores correspondentes.

Figura 4.32 - Cadastro Valores de Classe com dois intervalos.

RIOSS - Classe de Variáveis

Novo Editar Apagar Sair

Variável: Precipitação

Nome: Climas

Comentário: Classificação climática segunda as alturas pluviométricas referentes a uma série climatológica.

Valores de Classe Incluir Valores Salvar Cancelar

Nome	Variável
▶ Climas	Precipitação

1 Registro cadastrado.

RIOSS - Valores de Classe

Novo Editar Apagar Sair

Classe de Variáveis: Climas

Nome: Semiárido Quente

Valor Mínimo: 450

Valor Máximo: 900

Valor de Classe Associado a Função Utilidade:

Comentário: Clima encontrado principalmente na região nordeste brasileira.

Salvar Cancelar

Nome	Classes de Variáveis	Valor Mínimo	Valor Máximo
▶ Semiárido Quente	Climas	450	900
Tropical Chuvoso	Climas	750	1500

2 Registros cadastrados.

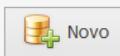
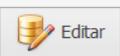
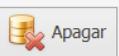
Fonte: Autor.

4.3.4 Sistema de Medições

O sistema de medições do SSD é composto pelas estações de medição, as variáveis medidas e os valores das variáveis. A Figura 4.33 apresenta a tela de cadastro do item Estações de Medição no menu Sistema de Medições.

Figura 4.33 - Item Estações de Medição no menu sistema de medições.

RIOSS - Estações de Medição

Pesquisar: Nome     

Informações Gerais:

Nome: Grupo de Variáveis: Operador:

Código: Instituição Responsável: Comentário:

Data de Instalação: Cliente:

Dados Espaciais:

Altitude: Baía Hidrográfica:

Latitude:  Rio:

Longitude: Local:

Código	Nome	Latitude	Longitude	Altitude
0001	Boqueirão	-7,484081	-36,133362	3
▶ 735026	Campina Grande	-7,208272	-35,874497	5

2 Registros cadastrados.

Fonte: Autor.

Na tela Estações de Medição é possível cadastrar uma estação de medição relacionando aos seus dados gerais e espaciais. As variáveis medidas naquela estação são definidas pelo grupo de variáveis que será associado, igualmente a sua localização que deverá estar nos limites da bacia hidrográfica escolhida. Também é possível definir a localização espacial da estação utilizando botão localização espacial. A Figura 4.34 apresenta a tela de cadastro do item Variáveis Medidas no menu Sistema de Medições.

Figura 4.34 - Item Variáveis Medidas no menu Estações de Medição.

RIOSS - Variáveis Medidas

Pesquisar: Estação de Medição

Novo Editar Apagar Sair

Estação de Medição: Campina Grande

Variável: Precipitação

Unidade de Medida: Milímetro

Área de Influência:

Fator de Tempo:

Início das Medições: 02/03/2015 16:34:53

Fonte:

- Gerados
- Medidos
- Telemétricos
- Outros

Classificação:

- Importados
- Calculados
- Repetidos
- Temporários

Dimensão:

- Unidimensional
- Bidimensional
- Tridimensional

Tipo:

- Topo
- Médio
- Lower
- Instantaneo

Escala de Tempo:

- Horário
- Mensal
- Anual
- Nenhum

Comentário:

Cadastrar Valores Salvar Cancelar

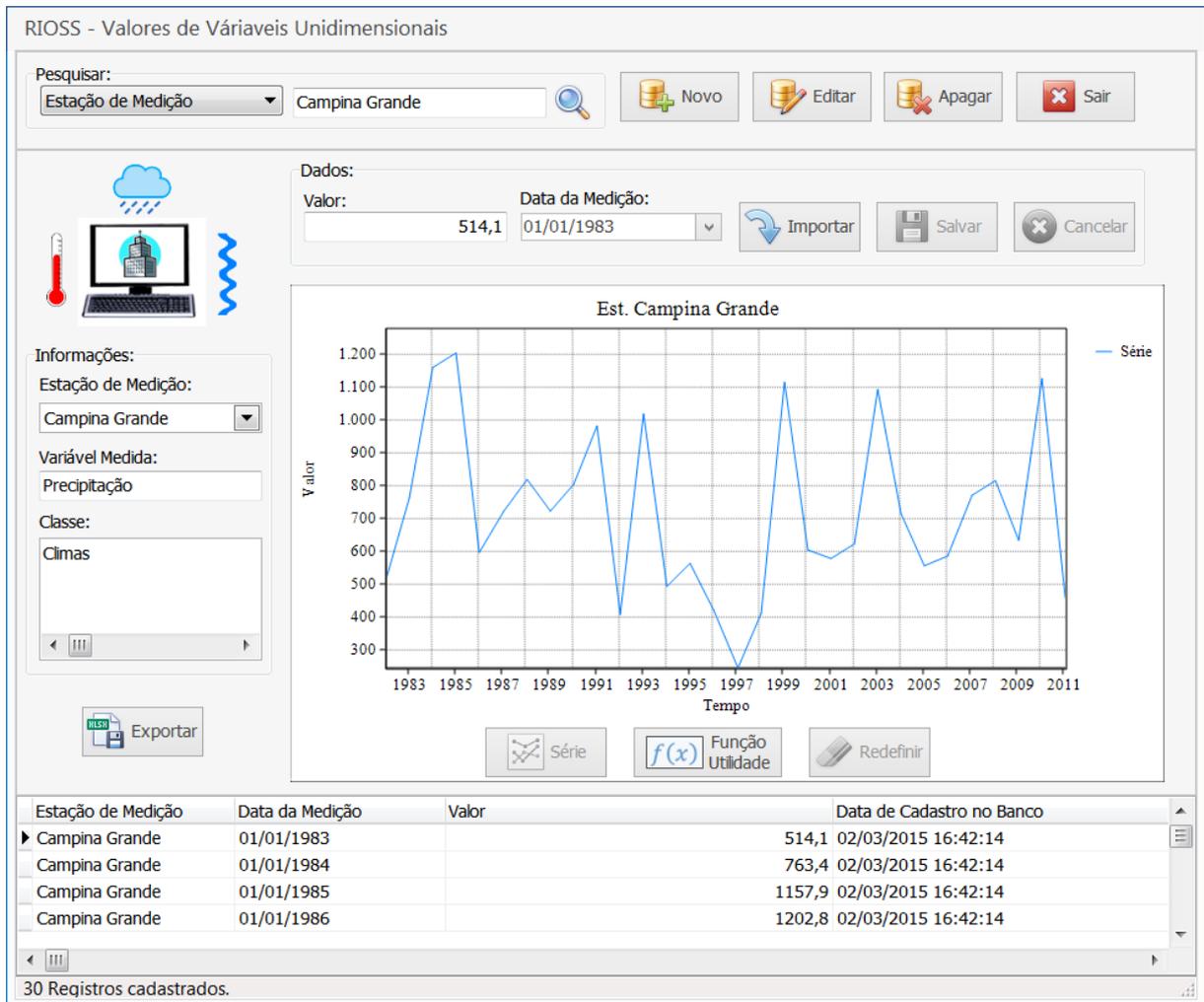
Estação de Medição	Variavel	Unidade de Medida	Dimensão	Tipo	Área de Impacto	Escala de Tempo	Fator de Escala	Inick
▶ Campina Grande	Precipitação	Milímetro	1	I		A		02/C

1 Registro cadastrado.

Fonte: Autor.

Na tela de cadastro apresentada na Figura 4.34 é possível cadastrar a variável para medição a uma estação de medição. Após a escolha da estação de medição é associado a variável, dentro do grupo de variáveis a qual a estação foi relacionada, e a unidade de medida das medições. Também são definidas algumas informações como a fonte dos dados, classificação, escala de tempo, dimensão, entre outros. Após as definições pelo usuário é possível então cadastrar valores da série utilizando o botão Cadastrar Valores. A Figura 4.35 apresenta a tela de cadastro Valores de Variáveis.

Figura 4.35 - Cadastro Valores de Variáveis.

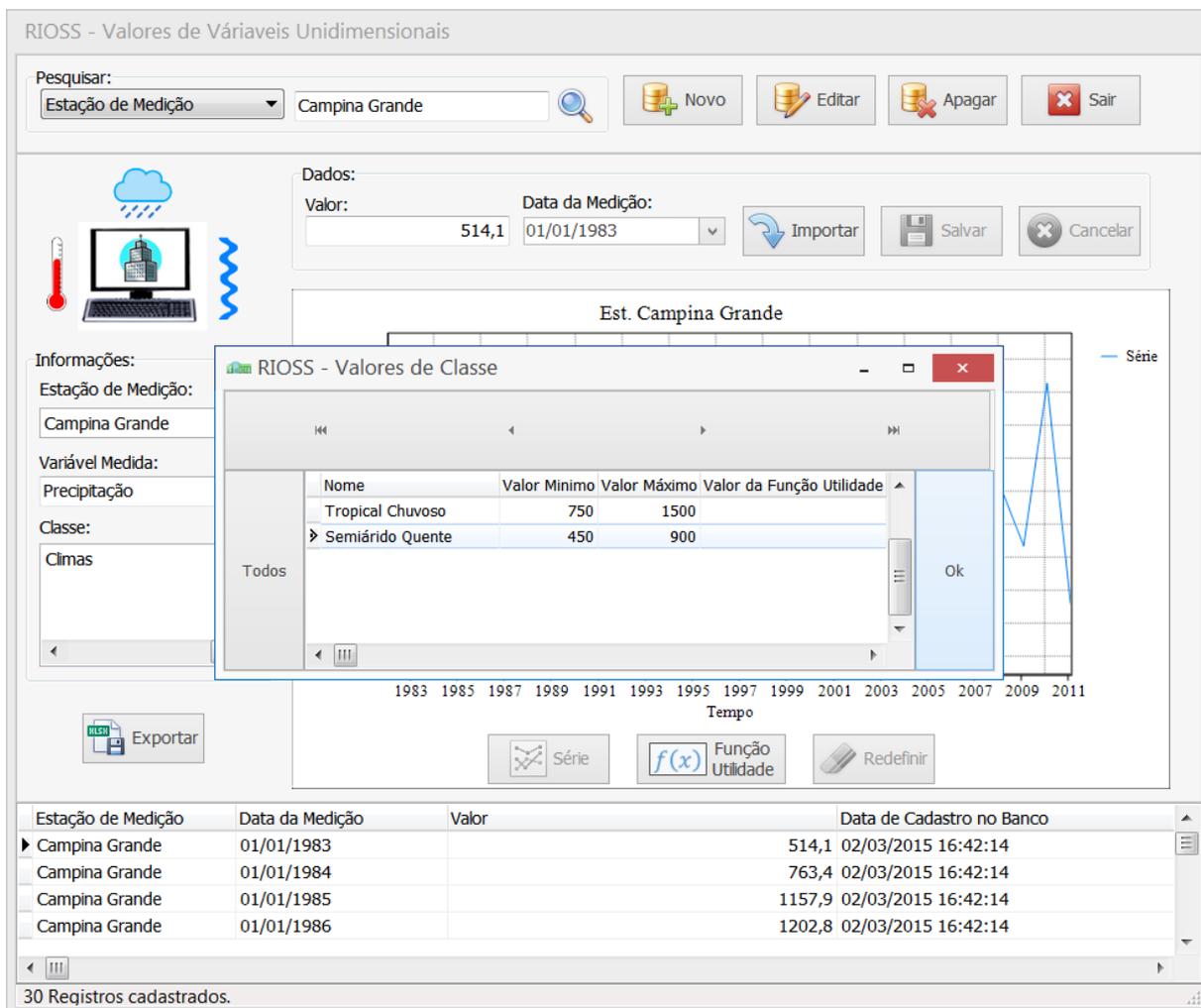


Fonte: Autor.

Na tela de cadastro Valores de Variáveis é possível cadastrar a série da variável medida em si. A série pode ser inserida de forma manual, quando viável, ou por meio dos mecanismos de importação (ANA Data Import, AESA Data Import e Importador de Arquivos). Na tela, a série pode ser visualizada num gráfico e em formato de tabela. Também é possível exportar os dados no formato XLSX utilizando o botão Exportar.

Apesar deste trabalho não ter sido implementado o sistema de análises do SSD, é possível realizar algumas análises simples com algumas funções implementadas. Na tela apresentada na Figura 4.35 é possível através de um clique duplo visualizar para as classes disponíveis para aquela variável, os valores dos intervalos para plotagem no gráfico. A Figura 4.36 apresenta a tela para escolha dos valores de intervalo de classe para plotagem.

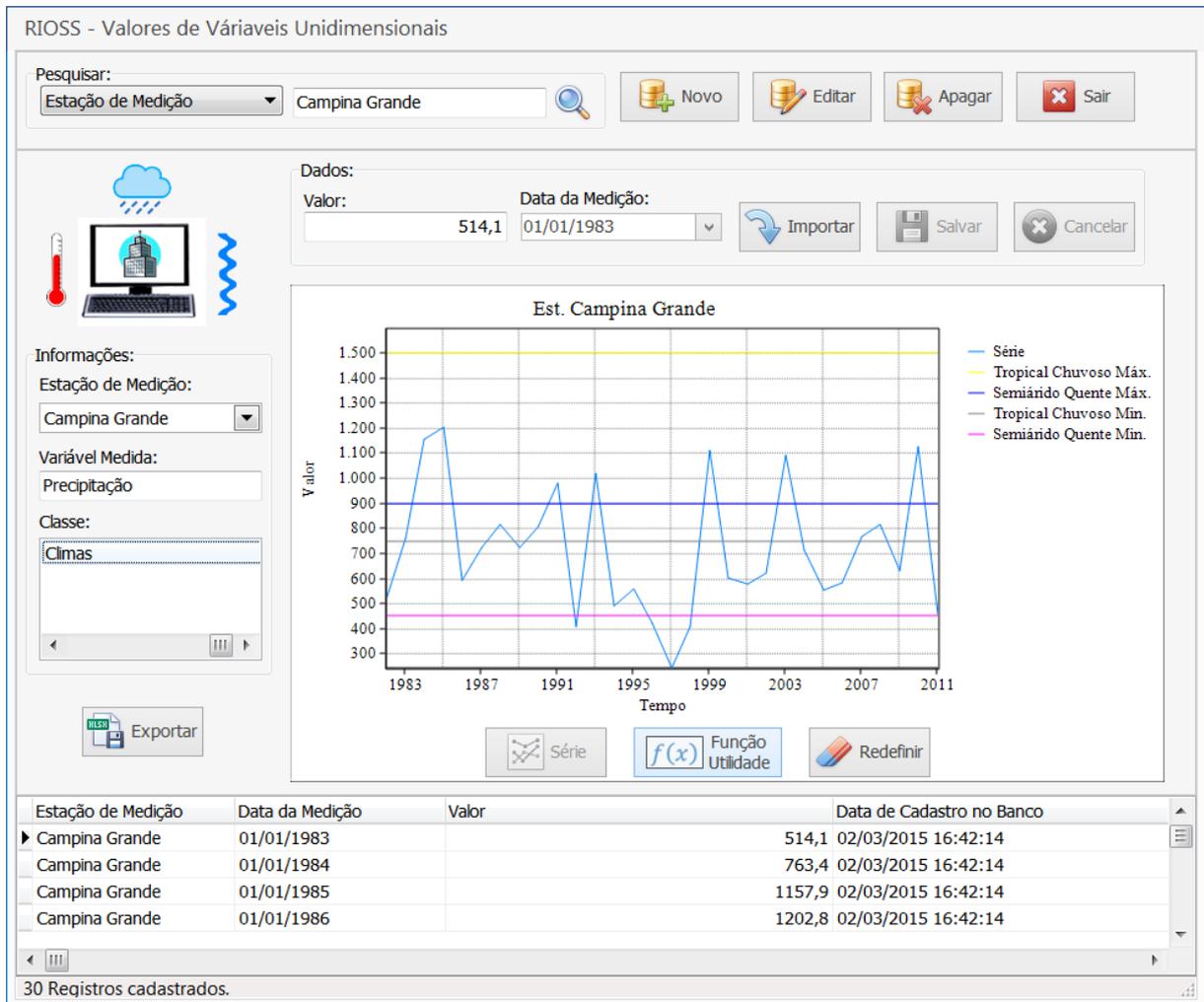
Figura 4.36 - Tela para escolha dos valores de classe.



Fonte: Autor.

Na tela apresentada na Figura 4.36 é possível escolher entre os intervalos de classe para serem plotados no gráfico, também é possível escolher todos os intervalos. A Figura 4.37 apresenta os valores de classe visualizados graficamente para série escolhida.

Figura 4.37 - Valores de classe para série escolhida.

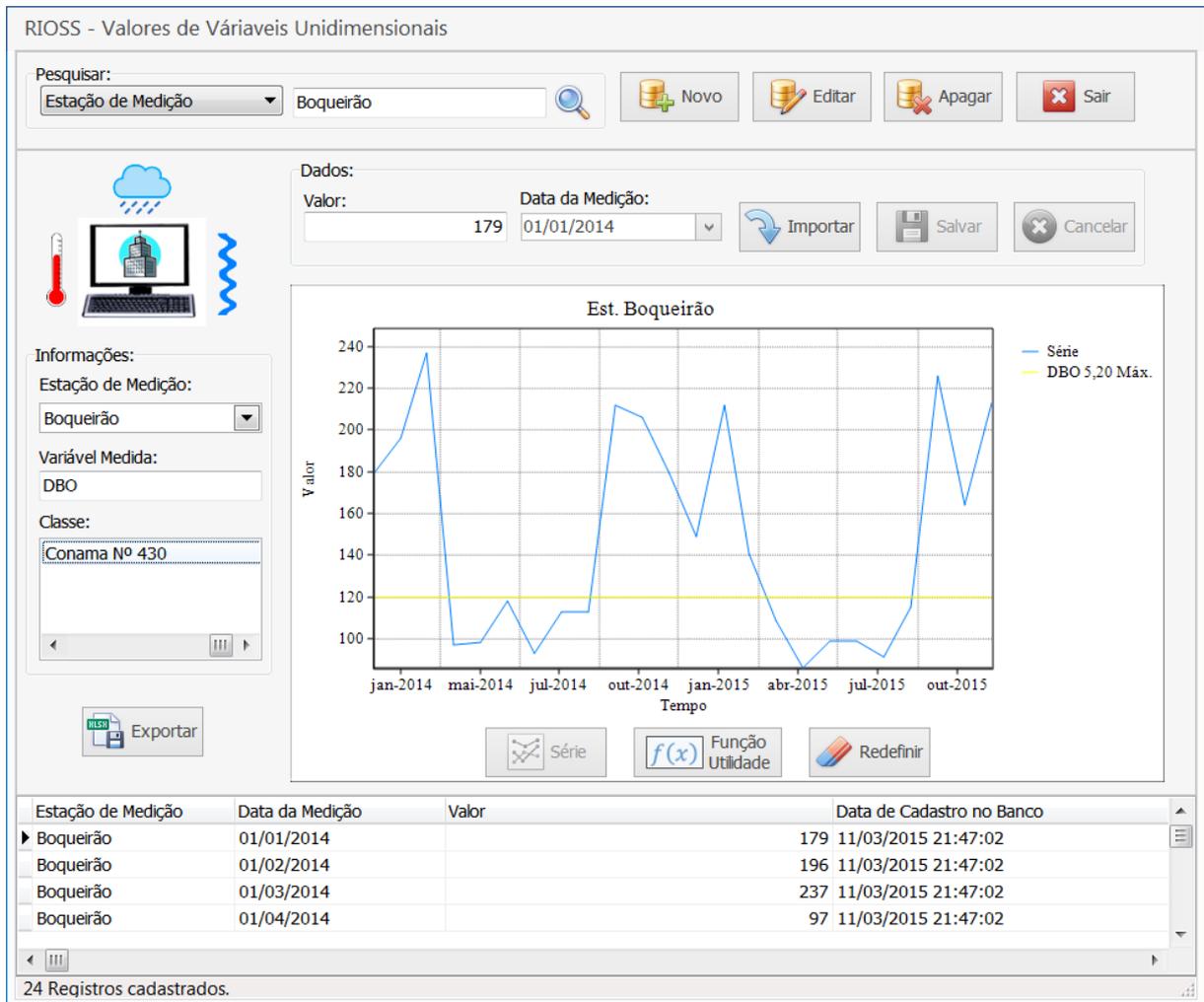


Fonte: Autor.

Na Figura 4.37 é exemplificado o uso de classes para realizar análises rápidas, onde no gráfico da série de 30 anos de precipitação da cidade de Campina Grande - PB, é sobreposto todos os valores da classe Climatas. A região de Campina Grande é classificada climaticamente como Semiárida Quente, como é possível visualizar no gráfico dentro do intervalo relativo mesmo com as alturas pluviométricas sendo irregulares.

Para redefinir o gráfico para plotagem apenas um intervalo de classe, o Semiárido Quente por exemplo, clica-se no botão Redefinir e novamente em Climatas escolhendo apenas um intervalo. Outra funcionalidade presente na tela Valores de Variáveis é a transformação do gráfico para os valores relativos a função utilidade cadastrada para aquela variável. A Figura 4.38 apresenta um exemplo de uma série de dados de DBO com sua classe respectiva.

Figura 4.38 - Exemplo de série de dados de DBO cadastrada.

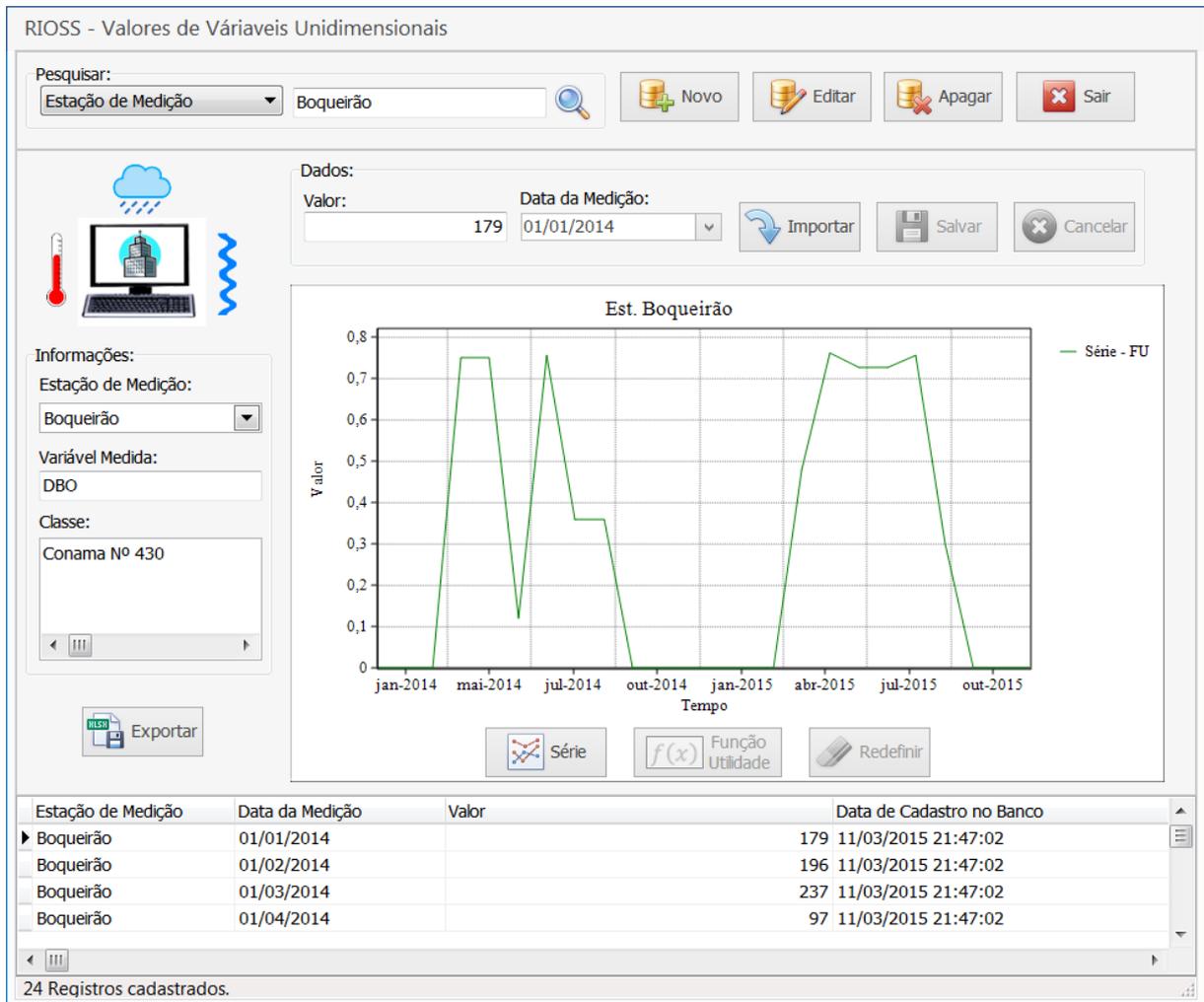


Fonte: Autor.

No exemplo apresentado na Figura 4.38 uma série mensal de dados de efluente tratado em uma estação experimental de tratamento de esgoto é plotada junto a classe Conama N° 430, permitindo a visualização dos meses os quais a eficiência de tratamento não proporcionou valores máximos permitido segundo a resolução. Clicando no botão Função Utilidade a série é interpolada realizando a transformação segundo os valores cadastrados para aquela variável.

A Figura 4.39 apresenta os valores cadastrados respectivos a função utilidade para a variável DBO e a série da variável plotada utilizando a ferramenta Função Utilidade na tela Valores de Variáveis.

Figura 4.39 - Utilização da ferramenta Função Utilidade.



Fonte: Autor.

Na Figura 4.39 é apresentado, a partir dos dados cadastrado na tela Função Utilidade para a variável DBO (ver Figura 4.29), a série de DBO transformada, onde, os valores próximos de 1 seriam aqueles mais distantes do máximo permitido segundo o intervalo de classe Conama Nº 430 (Figura 4.38) e valores próximos de 0 mais próximos do valor máximo permitido.

Pode-se aferir, por exemplo, visualizando o gráfico, que nos meses de precipitação alta a eficiência de tratamento foi boa pois possivelmente a carga orgânica do efluente era baixa devido a diluição do esgoto por infiltrações do sistema de drenagem daquela cidade - comum em cidades do interior -, ocorrendo o oposto nos meses mais secos. É claro que devem ser considerados muitos outros fatores par a análise e ressaltando que este é apenas um exemplo para ilustração no software.

4.3.5 Atividades

As atividades em um projeto se referem a um determinado conjunto de ações que estão ocorrendo simultaneamente, ou não, para alcançar objetivos diferentes. O menu Atividades apresenta os itens Grupo de Atividades, Atividade e Local da Atividade. A Figura 4.40 apresenta a tela de cadastro do item Grupo de Atividades no menu Atividades.

Figura 4.40 - Item Grupo de Atividades no menu atividades.

Pesquisar:

Nome

Novo Editar Apagar Sair

Nome:
Revitalização Boqueirão

Comentário:
Ações e atividades relacionadas a revitalização do açude Boqueirão.

Salvar Cancelar

Nome	Comentário
▶ Revitalização Boqueirão	Ações e atividades relacionadas a revitalização do

1 Registro cadastrado.

Fonte: Autor.

Na tela Grupo de Atividades é possível cadastrar um grupo de atividades dentro de um projeto, por exemplo, um projeto denominado “Revitalização dos Reservatórios para Abastecimento Humano na Paraíba” o grupo de atividades denominado Revitalização Boqueirão pode estar incluído neste projeto. A Figura 4.41 apresenta a tela de cadastro do item Atividade no menu Atividades.

Figura 4.41 - Item Atividade no menu atividades.

RIOSS - Atividade

Pesquisar: Nome

Novo Editar Apagar Sair

Nome: Batimetria Boqi

Tipo de Atividade:

 Urbana

 Rural

Grupo de Atividades: Revitalização Boqueirão

Grupo de Variáveis: Reservatórios

Variável: Cota

Comentário: Levantamento batimétrico para atualização da curva cota-área-volume do açude Boqueirão.

Salvar Cancelar

Nome	Local	Comentário
▶ Batimetria Boqueirão	R	Levantamento batimétrico para atualização da curva cota-área-volume do açude Boqueirão.

1 Registro cadastrado.

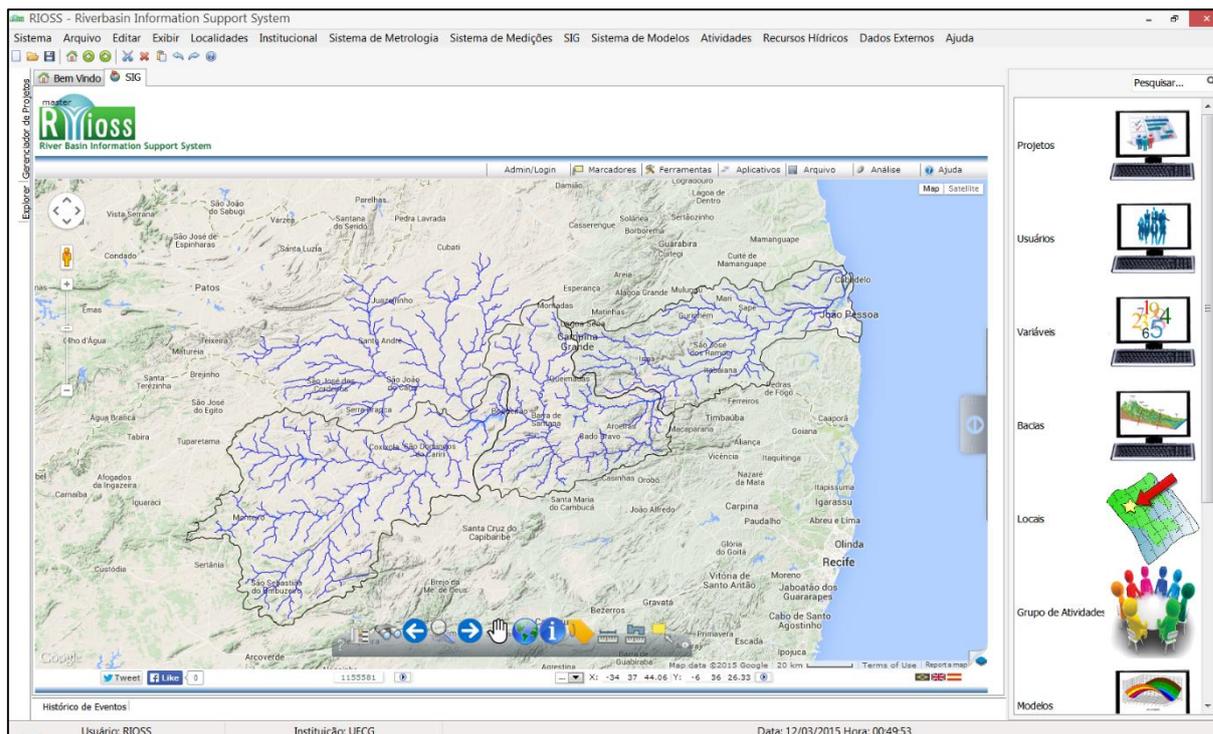
Fonte: Autor.

Na tela Atividade é possível cadastrar uma atividade e relacionar ao seu grupo, por exemplo, a atividade Batimetria Boqueirão, como descrita na Figura 4.41, que pode ser uma entre outras atividades no seu grupo.

4.3.6 SIG

Ao acionar o menu SIG o mesmo é aberto em uma nova aba na área de trabalho na tela principal. A interface e algumas funcionalidades do SIG foram apresentadas anteriormente (ver subtópico 4.1 Sistema de Informações Geográficas do SSD), dessa forma é apresentado aqui apenas o SIG funcionando de forma integrada ao sistema. A Figura 4.42 apresenta a tela principal do SSD com o SIG em funcionamento.

Figura 4.42 - SIG em funcionamento no SSD.



Fonte: Autor.

Na Figura 4.42 a tela principal do SSD apresenta a aba SIG onde pode ser trabalhado os dados espaciais. Também é possível notar a customização da aparência da tela principal, aonde foi possível esconder as janelas permitindo uma maior área para visualização no SIG.

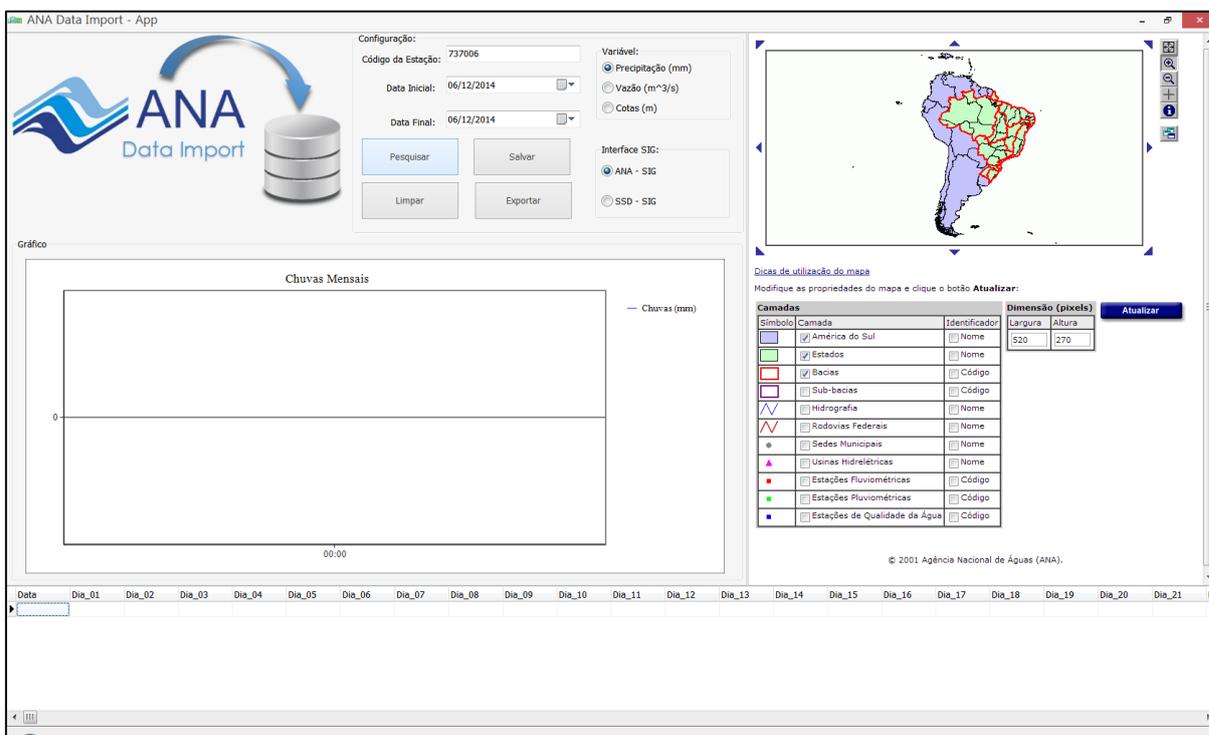
4.4 Aplicativos de Acesso a Dados Externos

O acesso a bancos de dados externos pelo SSD é muito importante, grande parte dos dados utilizados no planejamento de recursos hídricos são longas series históricas de variáveis armazenadas por diversas instituições. Enquanto o SNIRH ainda se estrutura para de fato centralizar as informações e disponibilizá-las por serviços e formatos viáveis de consumo, é necessário utilizar mecanismos para incorporar estes dados ao SSD.

Na era da computação em nuvem e grande conectividade, está cada vez mais viável utilizar dados e serviços sem a necessidade de armazená-los. Foi nessa perspectiva que estes aplicativos foram criados, nem sempre será necessário armazenar todas as series históricas no banco de dados do servidor, possibilitando a utilização imediata na operação corrente para obter os resultados desejados.

O fato da ANA disponibilizar os *Web Services* facilitou muito a criação do mecanismo para consumo dos dados. A Figura 4.43 apresenta a interface do aplicativo para consumo de dados da ANA.

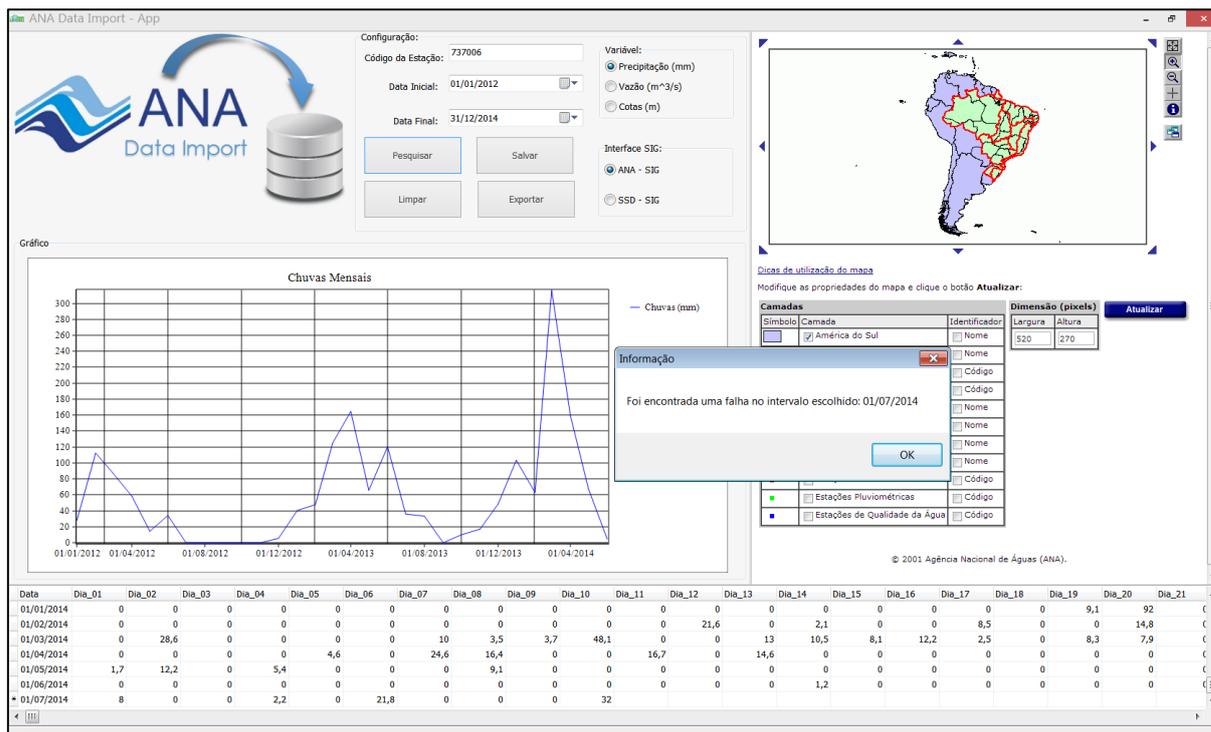
Figura 4.43 - Interface do aplicativo de acesso ao banco dados da ANA.



Fonte: Autor.

No aplicativo é possível escolher a variável para requerer os dados e o intervalo da série histórica. O código da estação desejada é obtido na janela que apresenta SIG com os postos e os códigos. Pode-se ainda escolher entre a interface do SIG da própria ANA ou do SSD. A Figura 4.44 apresenta o aplicativo após a realização de uma consulta.

Figura 4.44 - Consulta no aplicativo.



Fonte: Autor.

Após a realização da consulta os dados são visualizados na tabela e no gráfico e, caso ocorra alguma falha no intervalo escolhido, o aplicativo apresenta uma mensagem com a data que ocorreu a falha. Após a definição da série pelo usuário é possível então salvar os dados no banco do SSD, quando realmente necessário, e exportar para alguns formatos, CSV, XLS e TXT.

Diferentemente da construção de um mecanismo automatizado de acesso aos dados da AESA, foi necessário estudar a estrutura dos documentos HTML onde estão presentes os dados, para então desenvolver um algoritmo de análise e conversão de dados, ou seja, não ocorre acesso a serviços de banco de dados como no caso ANA.

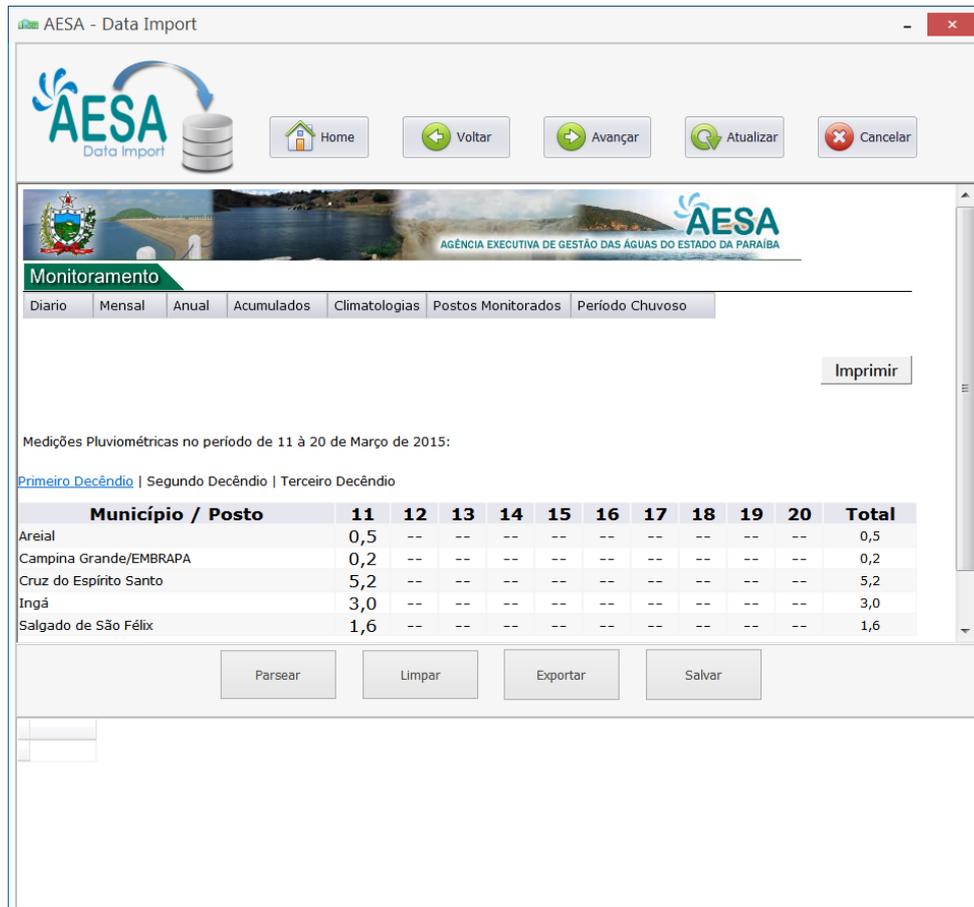
A Lei Nº 12.527, de 18 de Novembro de 2011, conhecida como Lei de Acesso a Informação no Brasil, definiu procedimentos para a viabilização de acesso e transparência de dados armazenados por qualquer entidade controlada por União, Estados, Distrito Federal e Municípios. No item três do terceiro parágrafo no artigo oitavo, lê-se: “possibilitar o acesso automatizado por sistemas externos em formatos abertos, estruturados e legíveis por máquina”.

A Paraíba é um estado que ainda não possui a implementação do instrumento de gestão Sistemas de Informações, mesmo com mais de dez anos da promulgação de sua lei marco da

gestão de recursos hídricos. Mais uma vez, destaca-se a importância e necessidade de sistemas de informação e suporte a decisão na área de recursos hídricos.

A Figura 4.45 apresenta a interface do aplicativo de acesso aos dados da AESA.

Figura 4.45 - Interface do aplicativo de acesso aos dados da AESA.



Fonte: Autor.

Na Figura 4.45 pode-se notar que a tabela que será populada com os dados ainda está sem estrutura. É diferente do caso ANA onde o serviço dispõe de uma estrutura via XML do *dataset*, que é incorporado pelo aplicativo, para carregar os dados da consulta. Neste caso, o *dataset* ainda não foi construído e só será construído após a varredura realizada pelo algoritmo. Como o algoritmo gera a estrutura XML para carregar os dados do arquivo HTML, seria uma opção deixar a tabela no aplicativo pronta, porém o algoritmo é capaz de analisar qualquer tabela de dados do site Monitoramento da AESA possibilitando acesso a outros dados. A Figura 4.46 apresenta o aplicativo após a análise e importação os dados.

Figura 4.46 - Análise e importação dos dados.

Medições Pluviométricas no período de 1 à 10 de Março de 2015:

Primeiro Decêndio | [Segundo Decêndio](#) | Terceiro Decêndio

Município / Posto	1	2	3	4	5	6	7	8	9	10	Total
Agua Branca	0,0	11,0	1,1	0,0	0,0	19,4	0,0	0,0	25,0	20,2	76,7
Aguiar	--	29,2	--	--	--	--	--	--	33,2	23,9	86,3
Alagoa Grande	0,0	10,0	5,0	0,0	0,0	23,6	0,0	6,8	--	--	45,4
Alagoa Nova	0,0	0,0	4,8	0,0	0,0	0,0	31,0	0,0	1,7	40,3	77,8
Alagoinha	0,0	1,1	4,9	0,0	0,0	21,6	2,5	3,4	5,8	4,3	43,6

Parsear Limpar Exportar Salvar

ID	Município / Posto	1	2	3	4	5	6	7	8	9	10
1	Agua Branc	0,0	11,0	1,1	0,0	0,0	19,4	0,0	0,0	25,0	20,2
2	Aguiar	--	29,2	--	--	--	--	--	--	33,2	23,9
3	Alagoa Gra	0,0	10,0	5,0	0,0	0,0	23,6	0,0	6,8	--	--
4	Alagoa Nov	0,0	0,0	4,8	0,0	0,0	0,0	31,0	0,0	1,7	40,3
5	Alagoinha	0,0	1,1	4,9	0,0	0,0	21,6	2,5	3,4	5,8	4,3
6	Alhandra	0,0	0,0	1,9	4,7	11,8	94,7	0,0	35,5	9,5	6,9

Fonte: Autor.

Após importar os dados da página que está sendo navegada pelo usuário é possível então salvar os dados no banco do SSD e exportar para formatos CSV, XLS e TXT. Como mencionado anteriormente, estes aplicativos serão utilizados durante operações realizadas no SSD, assim deverão ser configurados conforme a necessidade.

É importante destacar que a abordagem utilizada para ter o acesso automatizado dos dados da AESA fica à mercê de alguma alteração que seja feita pelo administrador do site, mesmo o algoritmo desenvolvido ter sido implementado de forma genérica e pensado em alterações futuras. Na verdade o que se espera é uma nova estrutura de disponibilização de dados da parte da AESA tanto para o público em geral, como para serem consumidos por sistemas especializados.

4.5 Sistema de Modelos

A proposta do sistema de modelos para o SSD, num primeiro momento, foi integrar um modelo já disponível pela comunidade de recursos hídricos e implementado em linguagem de

programação frequentemente utilizada na mesma comunidade, a qual é diferente da linguagem nativa do sistema aqui proposto.

O uso dessa abordagem foi motivada para verificar a potencialidade do sistema na integração à outras tecnologias, já que a proposta do sistema é ser genérico, e validar o sistema de informações como provedor de dados de entrada para os modelos. Para isso foram criadas classes e métodos para viabilizar este acesso, bem como uma interface amigável para utilização do modelo. A Figura 4.47 apresenta a interface criada para o modelo SMAPModel.

Figura 4.47 - Interface criada para o SMAPModel.

Fonte: Autor.

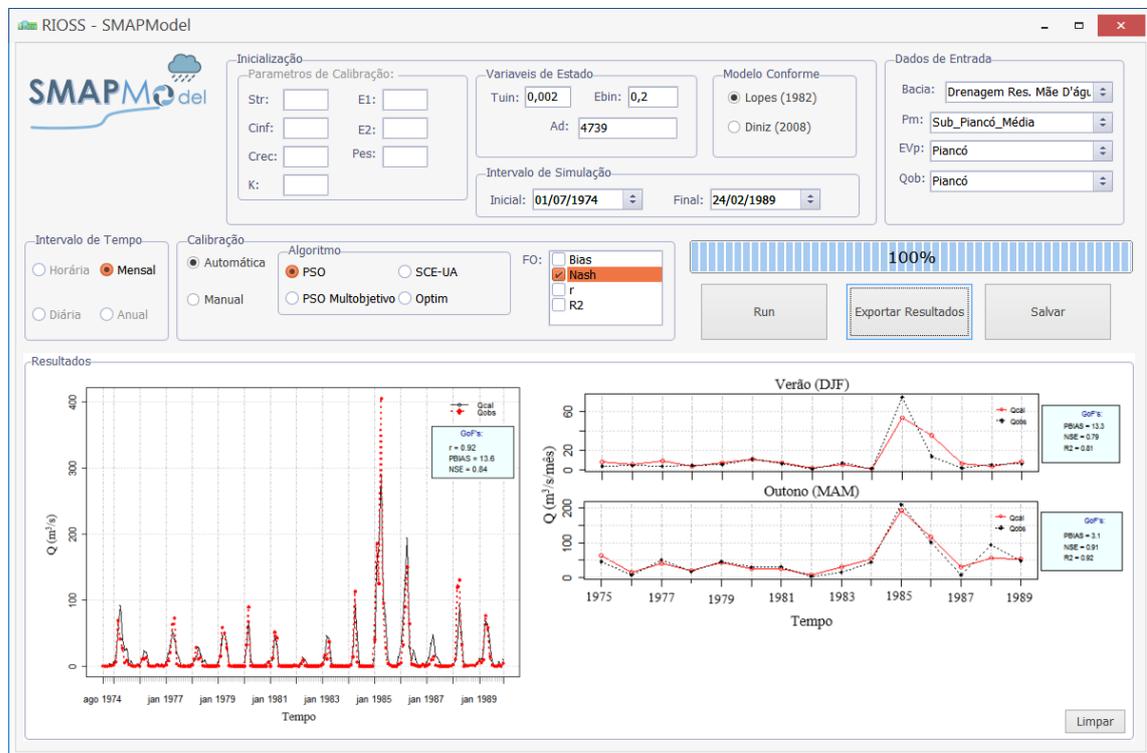
A interface criada para o modelo SMAPModel, uma implementação do modelo SMAP em ambiente R (pacote SMAPmodel), permite o usuário realizar todas as configurações do modelo via interação com os diversos controles. O SMAPModel permite a simulação em nível mensal e calibração automática e manual. Como abordado no capítulo anterior, o modelo utiliza outras bibliotecas R para otimização e análise da eficiência, o que também permite a utilização em outros módulos do SSD.

Na interface apresentada na Figura 4.47 o usuário; após definir configurações como intervalo de tempo, inicialização, calibração e dados de entrada; pode iniciar o processo de modelagem. Os dados de entrada do modelo, definidos na área Dados de Entrada, são

fornechos pelo sistema de medições do SSD, dessa forma após o usuário selecionar a bacia é possível escolher as estações de medição respectivas para cada variável. De posse do nome da variável e ID das estações o SMAPModel realiza a consulta no SGDB requisitando os dados via SQL, estruturando no R e realizando o processamento e gravação dos resultados de saída.

Na sessão Resultados é possível visualizar os resultados gráficos junto aos índices de eficiência da modelagem. O usuário ainda pode exportar os gráficos para arquivos do tipo PNG e salvar os índices no banco de dados. A Figura 4.48 apresenta o SMAPModel após a realização de uma simulação.

Figura 4.48 - Simulação no SMAPModel.



Fonte: Autor.

Para nível de comparação, na aplicação realizada apresentada na Figura 4.48 foi realizada a metodologia conforme Lourenço *et al.* (2014) de posse dos dados e arquivos disponibilizados, dessa forma foi possível comparar o desempenho do fluxo de operações do modelo original rodado no console R e o modificado integrado ao SSD.

O SMAPModel foi utilizado no modelo conforme Diniz (2008), no período mensal, sendo aplicado na bacia de drenagem Rio Piencó até ser barrado pelo reservatório Coremas, o qual foi apresentada anteriormente na aplicação do SIG. A calibração do modelo foi realizada

pelo algoritmo de otimização *Particle Swarm Optimization (PSO)* utilizando a função objetivo o coeficiente de Nash-Sutcliffe.

É importante salientar que o processamento do modelo ocorre todo pelo compilador do R, o mecanismo desenvolvido para integrar o modelo ao SSD apenas faz com que seja possível passar parâmetros e os dados por meio da interface criada ao R via SSD, então caso o modelo original em R não tivesse sofrido nenhuma alteração a velocidade de processamento do modelo seria praticamente a mesma, apenas adicionado o tempo que o SSD levaria para invocar o R.

Como descrito no capítulo anterior foi necessário realizar modificações no modelo original para permitir sua conexão ao banco de dados, que então iria prover os dados de entrada do modelo, no modelo original os dados de entrada são providos de arquivos TXT que são mais rápidos de serem lidos comparados a uma conexão a um banco de dados.

Comparando as duas aplicações em tempo de processamento o modelo integrado ao SSD levou aproximadamente um minuto a mais que o rodado diretamente no console R para terminar todo o processamento até chegar a tela apresentada na Figura 4.48. Essa diferença pode ser considerada razoável já que neste ocorrem operações entre o modelo e o banco de dados.

Assim, pode-se considerar neste caso a viabilidade da integração das duas ferramentas resultando no bom funcionamento do sistema. Porém é importante ressaltar que o SMAP é um modelo relativamente simples de ser implementado em Delphi onde certamente a velocidade de processamento seria superior, utilizando as técnicas corretas. Neste caso a disponibilidade de bibliotecas de otimização gratuitas do R, estas um processo mais complexo de implementação em Delphi, foi um condicional, já que o Delphi não disponibiliza estas bibliotecas nativamente e gratuitamente.

O grupo GOTA tem iniciado o desenvolvimento de uma interface do usuário para o modelo ORNAP, originalmente em C++, mas que deverá ser reprogramado para Delphi. O modelo traz uma proposta de utilização de matrizes esparsas que tem apresentado em softwares científicos, que utilizam métodos numéricos, ótimos resultados em termos de performance superiores a linguagens como o Matlab, dessa forma esta metodologia pode ser testada e implementada ao modelo SMAPModel comparando a sua performance.

Capítulo 5

5 Conclusões

O trabalho apresentado teve como proposta desenvolver e implementar uma arquitetura para um Sistema de Suporte a Decisão em recursos hídricos genérico, que pudesse ser aberto, expansível e adaptativo as características da unidade de planejamento bacia hidrográfica e que atendesse às necessidades atuais de seus planejadores. As principais conclusões foram:

- Com relação ao gerenciamento de dados:

O Sistema de Gerenciamento de Banco de Dados utilizado, o MySQL, se mostrou rápido e versátil, possibilitando o armazenamento de diversos modelos de dados, dando suporte a objetos espaciais e seus atributos. Entre os fatores positivos da utilização do mesmo está o fato de ser livre e possuir diversas ferramentas para modelagem de banco de dados por meio de diagramas, importação e exportação de bancos, integração com praticamente todas as linguagens de programação e ser multiplataforma.

- Com relação ao tratamento e visualização de dados espaciais:

O *framework* SIG utilizado apesar de ser desenvolvido para web, possui uma grande diversidade de ferramentas e atendeu, inicialmente, bem as necessidades para abordagem que foi utilizada neste trabalho. Além do mesmo ter sido desenvolvido com o que há de mais moderno, em termos de softwares (MapServer) e tecnologias programação (PHP e AJAX), sua integração com o R e uma boa comunidade para dar suporte, são características que tornam muito viável seu uso em um SSD.

- Com relação ao desenvolvimento do sistema:

O desenvolvimento do servidor de aplicações foi relativamente rápido na linguagem de programação utilizada, afinal o Delphi é considerado uma ferramenta RAD (*Rapid Application Development*). O servidor de aplicações DataSnap se mostrou robusto e versátil, possibilitando o bom atendimento para diversos clientes. Os diversos *frameworks* nativos possibilitaram boa automação no desenvolvimento em camadas, porém deve-se destacar uma certa dificuldade na implementação dos mecanismos de persistência de dados via mapeamento objeto relacional.

Outro fato é a carência de suporte e material gratuito na comunidade no que diz respeito a técnicas avançadas em orientação a objetos, que pode se justificar pelo fato do software ser proprietário e não haver versões gratuitas ou *express*.

Em termos de desenvolvimento de interface gráfica para desenvolvimento da camada cliente o software utilizado foi muito eficiente, possuindo diversos componentes nativos e outros mais de terceiros disponíveis gratuitamente na web.

- Com relação a inserção de variáveis e seus valores:

O sistema de metrologia permitiu cadastrar, caracterizar e compor variáveis de forma dinâmica segundo suas grandezas básicas ou derivadas. Também permitiu compor as unidades de medida relativas, agrupa-las de forma conveniente, classifica-las e atribuir valores as suas classes para posterior análise, entre outras necessidades. Outra função no sistema de metrologia é permitir atribuir valores relativos a função utilidade associado a uma variável, dessa forma permitindo a análise do comportamento de uma variável em uma serie temporal.

O sistema de medições utilizou-se das caracterizações das variáveis no sistema de metrologia incorporando muitas informações as estações de medição além das normalmente encontradas. Ele também permitiu a exportação e importação de dados, utilizando diversos mecanismos, entre eles os aplicativos de importação desenvolvidos.

- Com relação ao sistema de modelos:

A integração do modelo chuva-vazão desenvolvido no ambiente R também se mostrou funcional possibilitando a expansão e testes com outros modelos utilizados em recursos hídricos. Porém quando não inviável, o qual não é o caso do modelo SMAP, é preferível a reprogramação para linguagem nativa do SSD, o que pode aumentar em termos de velocidade e eficiência o uso dos modelos.

- Com relação ao acesso de dados externos:

Os aplicativos de acesso a dados e bancos de dados externos (ANA e AESA) se mostraram úteis e de grande necessidade para integração do SSD a sistema externos, eles possibilitarão em alguns casos o consumo dos dados das instituições sem a necessidade de armazenamento no servidor, o que é uma tendência nas tecnologias da informação.

Nos dois casos foi possível perceber a necessidade da consolidação e centralização de fato das informações sobre recursos hídricos em nível nacional, e estruturação e implantação efetiva dos instrumentos, como o Sistemas de Informações, em nível estadual.

Assim o sistema proposto pode ter potencial de contribuir nos processos de decisões, principalmente quando implementado o sistema de análises, podendo ser reutilizado conforme a necessidade dos usuários e instituições, como em agencias de recursos hídricos.

6 Recomendações para Trabalhos Futuros

Apesar do SSD ter se mostrado funcional há diversas etapas e módulos que necessitam ser implementados, novas inclusões no processo de desenvolvimento e testes com outras tecnologias. São elas:

- Implementar os demais módulos não implementados;
- Realizar toda a documentação do software na linguagem UML como confecção de diagramas caso de uso, atividades, sequencia, componentes, entre outros, que possibilitam a reutilização do projeto por qualquer usuário capacitado com conhecimento outras linguagens de programação diferentes, entre outras vantagens;
- Realizar a documentação do software em termos de suporte para os usuários, como o *Help*;
- Desenvolver o sistema de instalação e configuração automatizada do software;
- Verificar e testar tecnologias de bancos de dados orientados a objetos;
- Testar a integração de outros modelos;
- Testar o consumo dos métodos por clientes web;
- Verificar e testar tecnologias livres e atendimento de requisitos e demandas.

Referências Bibliográficas

ABRAMSON, A.; LAZAROVITCH, N.; MASSOTH, S.; ADAR, E. Decision support system for economic assessment of water improvements in remote, low-resource settings. **Environmental Modelling & Software**, v. 62, p. 197–209, 2014.

ADELMAN, L. **Evaluating decision support and expert systems**. New York: John Wiley and Sons, 1992.

AGÊNCIA EXECUTIVA DE GESTÃO DAS ÁGUAS DO ESTADO DA PARAÍBA. **Meteorologia: Dados de Chuva Diária**, Paraíba, 2012. Disponível em: <<http://site2.aesa.pb.gov.br/aesa/medicaoPluviometrica.do?metodo=chuvasDiariasMapa>>. Acesso em: 30 nov. 2014.

AGÊNCIA NACIONAL DE ÁGUAS. Disponível em: <<http://hidroweb.ana.gov.br>>. Acesso em: 07 dez. 2014.

AHMAD, S.; SIMONOVIC, S. P. A Decision Support Tool for evaluation of impacts of flood management policies. **Journal of Hydrological Science and Technology**, v. 17, p. 11-22. 2001.

ALENCAR, V. C. **Análises multiobjetivo, baseada em programação linear, e comparativa para agriculturas de manejo convencional e orgânico**. 2009. 374 f. Tese (Doutorado em Recursos Naturais) – Universidade Federal de Campina Grande, Campina Grande, PB, 2009.

ARSENE, C. T. C.; GABRYS, B.; AL-DABASS, D. Decision support system for water distribution systems based on neural networks and graphs theory for leakage detection. **Expert Systems with Applications**, v. 39, n. 18, p. 13214–13224, 2012.

BIGGERSTAFF, T. J. **Software Reusability Volume II: Applications and Experience**. Acm Press Frontier Series. 1989.

BERTALANFFY, L. VON. **General System Theory. Foundations, development and applications**. New York: George Braziler, 1968.

BEWOOR, A. K.; KULKARNI, V. A. **Metrology & Measurement**. McGraw-Hill Education, 2009.

BRAGA, B.; BARBOSA, P. S. F.; NAKAYAMA, P. T. Sistema de Suporte à Decisão em Recursos Hídricos. **Revista Brasileira de Recursos Humanos**, v. 3, n. 3. 1998.

CÂMARA, G.; DAVIS, C.; MONTEIRO, A. M.V. **Introdução à Ciência da Geoinformação**. São Jose dos Campos, SP: INPE. 2001.

CURI, W. F.; CURI, R. C. CISDERGO - Cropping and Irrigation System Design with Reservoir and Groundwater Optimal Operation. In: SIMPÓSIO BRASILEIRO DE RECURSOS HÍDRICOS, 14., 2001, Aracaju, SE. **Anais...** Aracaju, SE: ABRH, 2001a. CD-ROM.

CURI, W. F.; CURI, R. C. ORNAP - Optimal Reservoir Network Analysis Program. In: SIMPÓSIO BRASILEIRO DE RECURSOS HÍDRICOS, 14., 2001, Aracaju, SE. **Anais...** Aracaju, SE: ABRH, 2001b. CD-ROM.

CURI, W. F.; LEAL, E. O. M. Um Sistema de Informações para a plataforma RIOSS. In: SIMPÓSIO DE RECURSOS HÍDRICOS DO NORDESTE, 12., 2014, Natal, RN. **Anais eletrônicos...** Natal, RN: ABRH, 2014. Disponível em: <<http://www.abrh.org.br/xiisrhn/anais/>> Acesso em: 15 dez. 2014

DATE, C. J. **An Introduction to Database Systems**, 8th ed. Boston, MA: Addison-Wesley, 2004.

DAWOUD, M. A. The Development of Integrated Water Resource Information Management System in Arid Regions. **Arabian Journal of Geosciences**. v. 6, n. 5, p. 1601–1612, 2011.

DAVENPORT, T.H.; PRUSAK L. **Working Knowledge**. Boston, MA: Harvard Business School Press, 1998.

DIAS, D. S. **O sistema de informação e a empresa**. Rio de Janeiro: Livros Técnicos e Científicos, 1985.

DINIZ, L. S. **Regionalização de Parâmetros de Modelo Chuva-Vazão Usando Redes Neurais**. 2008. 213 f. Tese (Doutorado em Recursos Hídricos) – Universidade Federal do Rio Grande do Sul, Porto Alegre, RS. 2008.

DOOGE, J. C. I. **Linear Theory of Hydrologic Systems**. Washington: U.S. Dept. Agr. ARS Tech. Bull. 1468 Superintendent of Documents, 1973.

ECMA - European Computer Manufacturers Association. **The JSON Data Interchange Format**. ECMA International. Standard ECMA-404, 2013.

ELLIOTT J.; O'BRIEN T.M.; FOWLER R. **Dominando Hibernate**. Tradução do original: *Harnessing Hibernate*. Rio de Janeiro: AltaBooks, 2009.

ELMASRI, R.; NAVATHE, S. B. **Fundamentals of Database Systems**. 6th ed. Pearson Addison Wesley, 2011.

FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. Doctoral dissertation, University of California, Irvine, 2000.

FINLAY, P. N. **Introducing Decision Support Systems**. Oxford: Blackwell Publishers, 1994.

GAMMA, E.; HELM, R.; JHONSON, R.; VLISSIDES, J. **Padrões de Projeto**. Porto Alegre, RS: Bookman, 2008.

GIUPPONI, C.; SGOBBI, A. Decision Support Systems for Water Resources Management in Developing Countries: Learning from Experiences in Africa. **Water. LOCAL**, v. 5, n. 2, p. 798–818, 2013.

GRIGG, N. S. **Water Resources Management: Principles, Regulations, and Cases**. New York: McGraw Hill Book Co., Inc. 1996.

I3GEO - Interface Integrada para Internet de Ferramentas de Geoprocessamento. **i3Geo 5.0 - Manual de Administração**. LOCAL: Associação gvSIG. 2012.

INMETRO - Instituto Nacional de Metrologia, Qualidade e Tecnologia. **Sistema Internacional de Unidades: SI**. Duque de Caxias, RJ: INMETRO/CICMA/SEPIN, 2012.

JASHAPARA, A. **Knowledge Management: an Integrated Approach**. FT Prentice Hall, Harlow, 2005.

JOHNSON L. E. **Geographic information systems in water resources engineering**. CRC Press, 2009.

JOHNSON, R. E.; FOOTE, B. Designing reusable classes. **Journal of Object-Oriented Programming**. v. 1, p. 22-35, 1988.

KERSTEN, G. E.; MIKOLAJUK, Z.; YEH, A. G. O. **Decision support systems for sustainable development: a resource book of methods and applications**. Ottawa: IDRC, 2000.

KOUTSOYIANNIS, D.; KARAVOKIROS, G.; EFSTRATIADIS, A; et al. A decision support system for the management of the water resource system of Athens. **Physics and Chemistry of the Earth, Parts A/B/C**, v. 28, n. 14-15, p. 599–609, 2003.

LABADIE, J. W. Optimal operation of multireservoir systems: state-of-the-art review. **Journal of Water Resources Planning and Management**. LOCAL, v. 130, n. 2, p. 93-111. 2004.

LAUDON K. C.; LAUDON J. P. **Management Information Systems. Managing the Digital Firm**, 12th ed. Pearson Publication Limited, 2012.

LONGLEY, P. A.; MICHAEL F. G.; DAVID J. M.; DAVID W. R. **Geographic Information Systems and Science**. 3rd ed. Hoboken, NJ: Wiley, 2010.

LOBO, E. J. R. **Curso de Engenharia de Software**. São Paulo: Digerati Books.2008.

LOPES J. E. G.; BRAGA B. P. F.; CONEJO J. G. L. SMAP - A Simplified Hydrological Model, Applied Modelling in Catchment Hydrology, **Water Resources Publications**. 1982.

LOUCKS, D. P.; DA COSTA, J. R. **Computer aided decision support in water resources planning and management**. In: LOUCKS, D. P.; DA COSTA, J. R. Decision Support Systems, Water Resources Planning. Berlin v. 26, Springer-Verlag, 1991.

LOUCKS, D. P.; FRENCH, P. N.; TAYLOR, M. N. Developing and implementing Decision Support Systems. **Water Resources Bulletin**, LOCAL, v. 31, p. 571–582. 1985.

LOUCKS, D. P.; VAN BEEK, E. **Water resources systems planning and management**. Paris: United Nations Educational, Scientific and Cultural Organization (UNESCO), 2005.

LOUCKS, D.P.; GLADWELL, J.S. **Sustainability Criteria for Water Resource Systems**, International Hydrology Series, Cambridge: Cambridge University Press. 1999.

LOURENÇO, A. M. G.; FARIAS, C. A. S.; GALVÃO, C. O. Desenvolvimento de um modelo para processamento e análise de dados hidrológicos no ambiente R. In: SIMPÓSIO DE RECURSOS HÍDRICOS DO NORDESTE, 12., 2014, Natal, RN. **Anais eletrônicos...** Natal, RN: ABRH, 2014. Disponível em: <<http://www.abrh.org.br/xiisrh/anais/>> Acesso em: 15 dez. 2014

MAGIERA, E.; FROELICH, W. Integrated Support System for Efficient Water Usage and Resources Management (ISS-EWATUS). **Procedia Engineering**. v. 89, p. 1066–1072, 2014.

MAPSERVER - Open Source Web Mapping. **MapServer Documentation**. MapServer Team. 2014.

MCGOVERN, J. **A Practical Guide to Enterprise Architecture**. The Coad series. Prentice Hall Professional. 2004.

MCGEE, J.; PRUSAK, L. **Gerenciamento estratégico da informação**. Rio de Janeiro: Campus, 1997.

MOTA, J.C., CURI, W.F., ALMEIDA, M.M., ALENCAR, V.C. Desenvolvimento de um sistema de suporte a decisão para o planejamento e gerenciamento de recursos hídricos em nível de bacia hidrográfica. **Revista Verde de Agroecologia e Desenvolvimento Sustentável**. Campina Grande - PB, v. 6, p. 201 – 215, 2011.

MÜLBER, A. L.; AYRES, N. M. **Fundamentos para sistemas de informação**. 2nd ed. Palhoça: UnisulVirtual, 2005.

O'BRIEN, J. A. **Sistemas de informação e as decisões gerenciais na era da internet**. 2. ed. São Paulo: Saraiva, 2004.

OTERO, C. E. **Software Engineering Design: Theory and Practice**. CRC Press., 2012.

PAREDES, J.; ANDREU, J.; SOLERA, A. A Decision Support System for Water Quality Issues In The Manzanares River (Madrid, Spain). **Science of the Total Environment**. LOCAL, v. 408, p. 2576–2589. 2010.

PIERLEONI, A.; CAMICI, S.; BROCCA, L.; MORAMARCO, T.; CASADEI, S. Climate Change and Decision Support Systems for Water Resource Management. **Procedia Engineering**, v. 70, p. 1324–1333, 2014.

PORTO, R. L. L.; AZEVEDO, L. G. T. **Sistemas de Suporte a decisões aplicados a problemas de recursos hídricos**. In: PORTO, R. L. L (Org.). et al. Técnicas Quantitativas para o gerenciamento de recursos hídricos. 2. ed. Porto Alegre: Editora da UFRGS/Associação Brasileira de Recursos Hídricos. p. 43-95. 2002.

- REISSWITZ, F. **Análise De Sistemas Vol X: Linguagem de Programação Java**. Clube do Autores. 189p, 2009.
- RODRIGUES, A. C. L. **Nova proposta de operação de outorga aplicada a reservatórios - estudo de caso: bacia hidrográfica do rio Piancó**. 2007. 294f. Tese (Doutorado em Recurso Naturais) -Universidade Federal de Campina Grande, Campina Grande, PB.2007.
- ROWLEY, J. The wisdom hierarchy: representations of the DIKW hierarchy. **Journal of Information Science**, LOCAL. v. 33, n. 2, p. 163–180, 2007.
- SANTOS, V. S.; CURI, W. F.; CURI, R. C.; VIEIRA, A. S. Um modelo de otimização multiobjetivo para análise de sistema de recursos hídricos I: Metodologia. **RBRH – Revista Brasileira de Recursos Hídricos**. v.16 n.4 - Out/Dez, p. 49-60, 2011.
- SIMONOVIC, S.P. Tools For Water Management: One View of the Future, **Water International**. v. 25, p. 76–88. 2000.
- SOMMERVILLE, I. **Software Engineering**. Addison Wesley. 9th ed. 2010.
- STAIR, R. M. **Princípios de sistemas de informação: Uma abordagem gerencial**. Rio de Janeiro: Editora LTC. 1998.
- STAIR, R. M. **Princípios de sistemas de informação: Uma Abordagem Gerencial**. Rio de Janeiro: LTC, 2010.
- STAIR, R. M.; REYNOLDS, G. **Principles of Information Systems**. United States, Cengage Learning. 2014.
- TORRES, N. A. **Competitividade empresarial com a tecnologia de informação**. São Paulo: Makron Books, 1995.
- TUCCI, C. E. M. **Modelos Hidrológicos**. 2. ed. Porto Alegre: Editora da Universidade Federal do Rio Grande do Sul – ABRH., 2005.
- TURBAN, E. **Decision Support And Expert Systems: Management Support Systems**. Englewood Cliffs: Prentice Hall. 960p. 1995.
- TURBAN, E.; ARONSON, J. E.; LIANG, T. P.; SHARDA, R. **Decision Support and Business Intelligence Systems**. 8th ed. New Jersey: Pearson Education International, 2007.
- TURBAN, E; ARONSON, J. E. Decision Support Systems and. Intelligent Systems*. 6th ed. Prentice Hall, Upper Saddle River, NJ, 2001.
- US DOT – Department of Transportation. **Data Integration Glossary**. Office of Asset Management. Publication No. FHWA IF-01-017. 18 p. August, 2001.
- VIEGAS, J. S. F.; LANNA, A. E. L. O Paradigma da Modelagem Orientada a Objetos Aplicada a Sistemas de Recursos Hídricos (I) Modelo Básico de Objetos para uma Rede Hidrográfica. **RBRH – Revista Brasileira de Recursos Hídricos**. LOCAL. v. 8, n. I, p. 79–89, 2003.

VIEIRA, A. **Um Modelo de Simulação Via Programação Linear Sequencial, para Sistema de Recursos Hídricos**. 2007. 173f. Dissertação (Mestrado em Engenharia Civil e Ambiental) - Universidade Federal de Campina Grande-UFCG, Campina Grande, PB, 2007.

WATKINS, D. W.; MCKINNEY, D. C. **Recent Developments Associated with Decision Support Systems in Water Resources**, Reviews of Geophysics Supplement 33, American Geophysical Union. 1995.

WURBS, R. A. **Comparative Evaluation of Generalized River/Reservoir System Models**. Technical. Report, 282, Texas Water Resources Institute, College station, Tex. 2005.

YEH, W. W-G. Reservoir Management and Operation Models: A State-of-the-Art Review. **Water resources Research**. v. 21, n. 12, p.1797-1818. 1985.

ZENG, Y.; CAI, Y.; JIA, P.; JEE, H. Development of a web-based decision support system for supporting integrated water resources management in Daegu city, South Korea. **Expert Systems with Applications**, LOCAL. v. 39, n. 11, p. 10091–10102, 2012.