



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA  
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO  
TRABALHO DE GRADUAÇÃO



## Desenvolvimento do Componente Tema do SADE-GBHidro

BRUNO EDSON MARTINS DE ALBUQUERQUE FILHO  
([bemaf@cin.ufpe.br](mailto:bemaf@cin.ufpe.br))

Orientador:  
FERNANDO DA FONSECA DE SOUZA  
([dfd@cin.ufpe.br](mailto:dfd@cin.ufpe.br))

Recife, 11 de outubro de 2006



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA  
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO  
TRABALHO DE GRADUAÇÃO



## Desenvolvimento do Componente Tema do SADE-GBHidro

BRUNO EDSON MARTINS DE ALBUQUERQUE FILHO  
([bemaf@cin.ufpe.br](mailto:bemaf@cin.ufpe.br))

Orientador:  
FERNANDO DA FONSECA DE SOUZA  
([fdfd@cin.ufpe.br](mailto:fdfd@cin.ufpe.br))

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Pernambuco, como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Recife, 11 de outubro de 2006

*“É preciso sonhar, mas com a condição de crer em nosso sonho, de observar com atenção a vida real, de confrontar a observação com nosso sonho, de realizar escrupulosamente nossas fantasias. Sonhos, acredite neles”*

(Vladimir Lênin)

*“Viva como se fosse morrer amanhã. Aprenda como se fosse viver para sempre”*

(Mahatma Gandhi)

*A minha família, minha mãe, meu pai,  
minhas irmãs, meus avós e minha noiva  
pelo incentivo e apoio que todos me  
dedicaram em todos os momentos...  
...e principalmente para Maria Neide  
Ferreira de Farias (in memorian)...*

# Agradecimentos

Fica complicado citar nomes em agradecimento, pois alguém poderá ser esquecido. Assim, acredito que não excluiria ninguém ao agradecer a Deus, pais, familiares, professores, supervisores, orientadores, chefes, colegas de trabalho e faculdade, amigos e todos que influenciaram direta ou indiretamente na minha vida acadêmica, profissional e social, para tornar possível o desenvolvimento e a conclusão desta etapa de minha vida, no curso de bacharelado em ciência da computação.

Agora, em relação ao agradecimento às pessoas que influenciaram diretamente no desenvolvimento deste trabalho, agradeço primeiramente a Márcia, que me proporcionou a oportunidade de fazer parte desta equipe, coordenando o projeto de software do SADE-GBHidro, bem como por todas as reuniões importantíssimas para o seu desenvolvimento. Sem esquecer de Ioná, Sérgio e Paulo, que sempre participaram de tais reuniões, também fornecendo novos requisitos, nas quais Sérgio, o “Engenheiro de Testes”, sempre apontou melhorias para o mesmo.

Também agradeço aos bolsistas: Renata, Denise, Thiago, Silas, Rodrigo que concretizaram o sistema e em especial para Daniel, que passou o maior período conosco e desenvolveu boa parte do sistema, incluindo o Componente Tema, mostrando-se um excelente profissional.

Agradeço ao corpo docente do CIn-UFPE, funcionários e sua estrutura, que proporcionaram o alicerce básico para minha formação e todos os conceitos aplicados no desenvolvimento do sistema, em especial para o meu orientador, Professor Fernando.

# Resumo

## **Desenvolvimento do Componente Tema do SADE-GBHidro**

Por a água ser um recurso escasso em algumas regiões e pela poluição existente em várias partes do planeta, torna-se necessário gerir e planejar sua preservação. Para a gestão de bacias hidrográficas, um SADE torna-se ideal por possuir características de SAD e SIG. Um modelo do SADE-GBHidro, o de otimização GAMS, hidrológico e econômico, juntamente com uma representação específica de dados, o banco de dados de temas, a partir do mapeamento de elementos, atributos e relações, condições e restrições podem ser utilizados para se conseguir uma melhor alocação de água. Além disso, o banco de dados de temas é utilizado como um elemento intermediário de transporte entre a base de dados geográfica e o modelo, auxiliando o usuário a relacioná-los.

O objetivo deste trabalho de graduação é a documentação do desenvolvimento de um componente importante, responsável pela representação gráfica e gerenciamento deste banco de dados de temas, fornecendo uma representação abstrata do mundo real a ser utilizada pelo modelo, o Componente Tema.

**Palavras-chave:** SADE, Integração de dados, Desenvolvimento de Software baseado em Componentes, XML.

# Abstract

## **Development of the Theme Component of SADE-GBHidro**

For being the water a scarce resource in some regions and because the existing pollution in some parts of the planet, it is necessary to manage and to plan its preservation. For the hydrographic basins management, a SDSS becomes ideal for having characteristics of DSS and GIS. A model of SADE-GBHidro, a GAMS optimization, economics and hydrologic model, together with a specific data representation, the thematic data base, from the mapping of elements, attributes and relations, conditions and restrictions can be used to obtain better water allocation. Moreover, the thematic data base is used as an intermediate transport element between the geographic database and the model, assisting the user to relate them.

The objective of this undergraduation work is the documentation of the development of an important component, responsible for the graphical representation and management of such a thematic data base, supplying a real world abstract representation to be used by the model, the Theme Component.

**Keywords:** SDSS, Data Integration, Component-Based Software Development, XML.

# Índice

1.	Introdução.....	9
1.1.	Objetivos.....	10
1.2.	Relevância .....	10
1.3.	Organização do Trabalho.....	11
2.	Fundamentos Conceituais.....	12
2.1.	O SADE na Gestão de Recursos Hídricos.....	12
2.1.1.	Sistemas de Informação Geográfica.....	13
2.1.2.	Sistemas de Apoio a Decisão.....	14
2.1.3.	Sistemas de Apoio a Decisão Espacial.....	16
2.2.	Modelagem e Representação Utilizadas.....	17
2.2.1.	Modelo de Otimização GAMS (General Algebraic Modelling System) .....	17
2.2.2.	Representação em Rede de Nós e Links.....	19
2.2.3.	Base de Dados de Temas .....	21
2.3.	Elementos da Linguagem Utilizados.....	22
2.3.1.	Componentes de Software.....	23
2.3.2.	Dados XML (eXtensible Markup Language).....	24
2.4.	Conclusões.....	26
3.	O SADE-GBHidro.....	27
3.1.	Estratégias de Desenvolvimento.....	27
3.2.	Arquitetura do Sistema .....	29
3.3.	Utilização de Componentes no Sistema .....	30
3.4.	Fluxo de Operações .....	34
3.5.	Requisitos do Sistema.....	37
3.5.1.	Funcionalidades dos Componentes .....	40
3.5.2.	Funcionalidades do Componente Tabela.....	42
3.5.3.	Funcionalidades do Componente Tema .....	43
3.6.	Conclusões.....	44
4.	O Componente Tema.....	45
4.1.	Motivação .....	45
4.2.	Implementação do Componente Tema.....	46
4.2.1.	Criação da Interface de Representação.....	46
4.2.2.	Criação da Visualização e Edição Genérica.....	48
4.3.	Conclusões.....	50
5.	Aplicações do Componente Tema.....	51
5.1.	Representação de Dados.....	51
5.1.1.	Representação da Bacia do Pirapama.....	52
5.1.2.	Exemplos de Representações de Dados .....	56
5.2.	Simulações.....	59
5.3.	Conclusões.....	60
6.	Conclusões.....	61
6.1.	Dificuldades Encontradas .....	61

6.2.	Trabalhos Futuros .....	62
6.3.	Considerações Finais .....	63
	Referências Bibliográficas.....	65
	Anexos.....	69
Anexo A	- Rede do Pirapama antes do sistema (criação manual) [18].....	70
Anexo B	- Rede do Pirapama depois do sistema (classes fixas) .....	73
Anexo C	- Rede do Pirapama depois do sistema (rede genérica) .....	75
Anexo D	- Exemplo de Código Comentado GAMS.....	77
Anexo E	- Breve Manual do Componente Tema do SADE-GBHidro.....	79
Anexo F	- Interfaces utilizadas relacionadas ao Componente Tema .....	90
	Interface IModulo .....	90
	Interface ITables .....	90
	Interface IForm .....	91
	Interface ISimbolo .....	91

# Lista de Figuras

Figura 2.1 - Integração no SIG [28]. .....	13
Figura 2.2 - Exemplo de Camadas do Mapa [30].....	14
Figura 2.3 - Componentes de um SAD [26] apud [5]. .....	15
Figura 2.4 - Acoplamento Fraco no SADE [20].....	16
Figura 2.5 - Acoplamento Pleno no SADE [20].....	16
Figura 2.6 – Dados espaciais: Bacia do rio Karshikardaya na Ásia Central [13].....	20
Figura 2.7 – Dados do Tema: Um exemplo de rede para a bacia do rio Karshikardaya [13] .....	20
Figura 2.8 - Integração XML na plataforma .NET.....	25
Figura 3.1 - Arquitetura do SADE-GBHidro .....	29
Figura 3.2 - Componentes do SADE-GBHidro.....	31
Figura 3.3 – Exemplo da hierarquia utilizada pelo sistema.....	32
Figura 3.4 - Diagrama Atual dos Componentes do SADE-GBHidro .....	33
Figura 3.5 - Fluxo de Operações do Sistema.....	35
Figura 3.6 – Relação entre os Componentes e o Fluxo de Operações.....	37
Figura 3.7 - Exemplo de utilização de Caso de Uso.....	38
Figura 3.8 - Funções do Sistema divididas.....	38
Figura 3.9 - Funções do Sistema - Sessão e Projeto.....	39
Figura 3.10 - Legenda para os Componentes .....	40
Figura 3.11 - Funções Gerais dos Componentes do Sistema .....	41
Figura 3.12 - Funcionalidades do Componente Tabela.....	42
Figura 3.13 - Funcionalidades do Componente Tema.....	43
Figura 4.1 - Exemplo de uso do componente tabela dentro do componente tema.....	46
Figura 4.2 - Exemplo de descrição utilizada na representação.....	47
Figura 4.3 - Exemplo de edição dos atributos do reservatório Pirapama .....	47
Figura 4.4 - Estrutura básica das tabelas para configuração de visualização (VRX).....	49
Figura 4.5 - Exemplo de vários símbolo para um mesmo elemento .....	50
Figura 5.1 - Diagrama de Representação para o Pirapama.....	56
Figura 5.2 - Exemplo de árvore genealógica.....	57
Figura 5.3 - Exemplo de diagrama de fluxo .....	58
Figura 5.4 - Exemplo de grafo.....	59

# 1. Introdução

Por a água ser um recurso escasso em algumas regiões, assim como pela poluição que a tem atingido, a gestão e o planejamento tornam-se necessários para preservação deste recurso natural. Dessa forma, a política nacional de recursos hídricos define instrumentos de gestão, como outorga, cobrança e enquadramento. Estes são utilizados para controlar o uso com o objetivo de que o mesmo seja realizado da melhor maneira possível, segundo a LEI Nº. 9.433 [2].

A melhor maneira de gerir bacias seria dando o suporte tanto à tomada de decisões quanto a informações geográficas, fornecendo modelos que representem o mundo real. Neste caso, uma das estratégias que podem ser utilizadas é a criação de um SADE (Sistema de Apoio a Decisão Espacial) que une as características de um SAD (Sistema de Apoio a Decisão) e um SIG (Sistema de Informações Geográfico), sendo constituído de uma base de modelos para apoio a decisão e base de dados geográficos [20, 19].

Para promover a integração entre a base geográfica e a base de modelos para apoio à decisão, é criado um elemento intermediário de transporte que possui informações não espaciais obtidas a partir da base geográfica. Além disso, devem ser incluídas informações adicionais mais detalhadas que são focadas no modelo criado, como informações econômicas, possibilitando seu acesso pelo usuário através de uma interface mais amigável criada para fornecer uma representação gráfica dos dados e auxiliar o usuário a relacionar a base geográfica e o modelo.

Este elemento intermediário pode ser chamado de banco de dados de temas e sua representação gráfica, realizada pelo Componente Tema, consiste em visualizar os elementos e as relações existentes entre eles, possibilitando a edição de atributos pelo usuário. Assim este trabalho apresenta o Componente Tema do SADE-GBHidro [19], detalhando o seu desenvolvimento, englobando também outras partes do sistema.

## 1.1. Objetivos

Dessa forma, um dos objetivos do desenvolvimento do SADE-GBHidro é o apoio ao decisor, gestor de recursos hídricos, através de uma interface amigável, para criação de modelos que forneçam subsídios científicos para tomada de decisão ao utilizar os instrumentos de gestão.

Um dos principais componentes do sistema é o Componente Tema, objeto deste trabalho de graduação. Este componente é responsável pela representação e gerenciamento de dados do tema, pois fornece aos usuários, uma representação abstrata do mundo real a ser utilizada pelo modelo. Além das estruturas de apoio, a representação gráfica do banco de dados de temas deve ser realizada de forma genérica, permitindo representar os dados de forma mais heterogênea, flexibilizando sua utilização em outros modelos.

Partindo disso, o objetivo deste trabalho se encontra na documentação deste componente, e suas relações com os demais componentes do sistema, principalmente relacionando-o à área de banco de dados. Dessa forma, inicialmente devem ser analisadas estratégias a serem utilizadas para implementação e documentação do Componente Tema.

## 1.2. Relevância

Inicialmente, pensou-se em realizar um trabalho sobre todo o sistema, no qual seus componentes seriam abordados brevemente. Em seguida, foi proposto focar apenas em um determinado componente, e na escolha deste componente, com três principais opções: os componentes geográfico, tema e de otimização.

O componente geográfico, basicamente, não possui nenhuma inovação, pois sua funcionalidade já é bem difundida. Já o componente de otimização é constituído de uma gerência de código fonte do modelo, execução local e remota do modelo internamente ao sistema e saídas gráficas como resultado. Enfim, da mesma forma, tudo o que ele representa não possui caráter inovador. Entretanto, a representação gráfica e genérica de dados e sua gerência, além de estar vinculada a área de banco de dados, possuem este caráter e pode ser utilizado em outras aplicações.

Então, a relevância deste trabalho está na especificação de um componente responsável pela representação de dados do tema do sistema de forma a representar os

objetos e seus relacionamentos de forma diferenciada e genérica, ou seja, diferenciando objetos de tipos e relações diferentes, assim como genericamente ser adaptável a determinados tipos de representação e estrutura de dados. Além de fornecer a representação dos dados, deve permitir posicionamento, edição dos seus atributos, inclusão e remoção, garantindo ainda a consistência dos dados e suas relações.

Assim, esta representação consiste numa abstração de dados, na qual são representados apenas os dados em que são focados. Dessa forma, este componente deverá ser utilizado pelo usuário para focar os dados em função de um determinado modelo, proporcionando uma interface mais amigável e interativa, permitindo-o a manipular os dados no sistema.

O desenvolvimento de uma base de dados consistente de uma bacia hidrográfica é importante para fazer com que a sua gestão aconteça de forma coerente, pois a partir desta base, podem ser obtidas informações que são úteis na tomada de decisão e utilização de modelos.

### **1.3. Organização do Trabalho**

Inicialmente, no Capítulo 2, será realizada uma fundamentação conceitual para iniciar o desenvolvimento do sistema, como a utilização do SADE na gestão de bacias, a modelagem e representação utilizadas e conceitos sobre componente de software e a linguagem utilizada.

A partir daí, no Capítulo 3, o sistema completo, SADE-GBHidro, é descrito, contendo as estratégia de desenvolvimento, arquitetura, componentes, fluxo de operações e requisitos desta aplicação.

Em seguida, é detalhado o Componente Tema em relação a sua implementação, foco do trabalho, no Capítulo 4

Após este detalhamento, são mostradas, no Capítulo 5, as suas aplicações com exemplos de sua utilização.

Finalmente, no Capítulo 6, são realizadas conclusões, críticas em relação ao trabalho desenvolvido e propostas de desenvolvimentos e atualizações futuras.

## 2. Fundamentos Conceituais

Para melhor entender o objetivo da criação do sistema e desenvolvimento de um componente responsável pela gerência de uma base de dados, é necessário compreender no que ele se baseia, os problemas existentes, assim como o tipo de sistema e metodologia que deva ser utilizada. Para isso, são definidos inicialmente: a base para este tipo de sistema, SADE, utilizado na gestão de recursos hídricos; a estrutura da representação utilizada para tal fim; e elementos da linguagem utilizados no seu desenvolvimento.

Além disso, por ser um sistema relativamente complexo, considerando ser multidisciplinar, envolvendo também bases de dados distintas e modelos, necessita-se da introdução de conceitos básicos para fundamentação e compreensão do contexto em que o sistema está inserido, assim como sobre elementos da linguagem de programação utilizada para seu desenvolvimento.

### 2.1. O SADE na Gestão de Recursos Hídricos

Ao utilizar os instrumentos de gestão, é necessário se basear nas relações existentes no mundo real e avaliar como seria o melhor uso. Neste caso, para dar suporte ao gestor da bacia, as relações espaciais do mundo real podem ser representadas por um SIG (Sistema de Informações Geográficas), incluindo uma base de dados geográfica da bacia em questão e sendo utilizado para planejamento e gerenciamento. Por outro lado, a avaliação do melhor uso da água pode ser obtida através de uma modelagem científica do mundo real, incluindo também relações não-espaciais, realizada por um SAD (Sistema de Apoio a Decisão), por se tratar de um problema não ou parcialmente estruturado. Esta modelagem fornece como resultado, valores de referência para utilização dos instrumentos de gestão. Assim, um sistema que reúna as características de um SIG e um SAD, que é definido com um SADE (Sistema de Apoio a Decisão Espacial), seria ideal para apoiar esta gestão.

Assim, ambos, SIG e SAD, têm sido extensivamente usados na gestão e planejamento dos recursos hídricos. O uso conjunto dessas ferramentas, os chamados SADE, começam mais recentemente, a trazer contribuições no apoio desta que é um tipo de tomada de

decisão pouco ou completamente não-estruturada. Comparados aos tradicionais SAD, os SADE na gestão de águas ao incorporar os SIG, melhoram a sua performance nos aspectos: base de dados, interface e conexão de modelos [19].

### 2.1.1. Sistemas de Informação Geográfica

Um SIG constitui um sistema computacional que captura, armazena, analisa e exhibe informações georreferenciadas, identificadas de acordo com a localização. Outras definições incluem procedimentos e dados espaciais inseridos no sistema [28].

Através do SIG é possível relacionar informações de diferentes fontes topologicamente, ou seja, através da sua localização espacial, ou também a partir de seus atributos. Dessa forma, é possível realizar uma integração destas informações, utilizando tipos de informações e estruturas de dados diferentes.

Além disso, são utilizados, na representação espacial, sistemas de projeção e sistemas de coordenadas específicos, considerando que dados em diferentes sistemas podem ser convertidos com o objetivo de se utilizar um único sistema de projeção e coordenada.

A entrada de informações no SIG pode ser feita por digitalização de imagens e coordenadas GPS (*Global Positioning System*), dentre outras. Um exemplo desta integração é mostrado, na Figura 2.1.

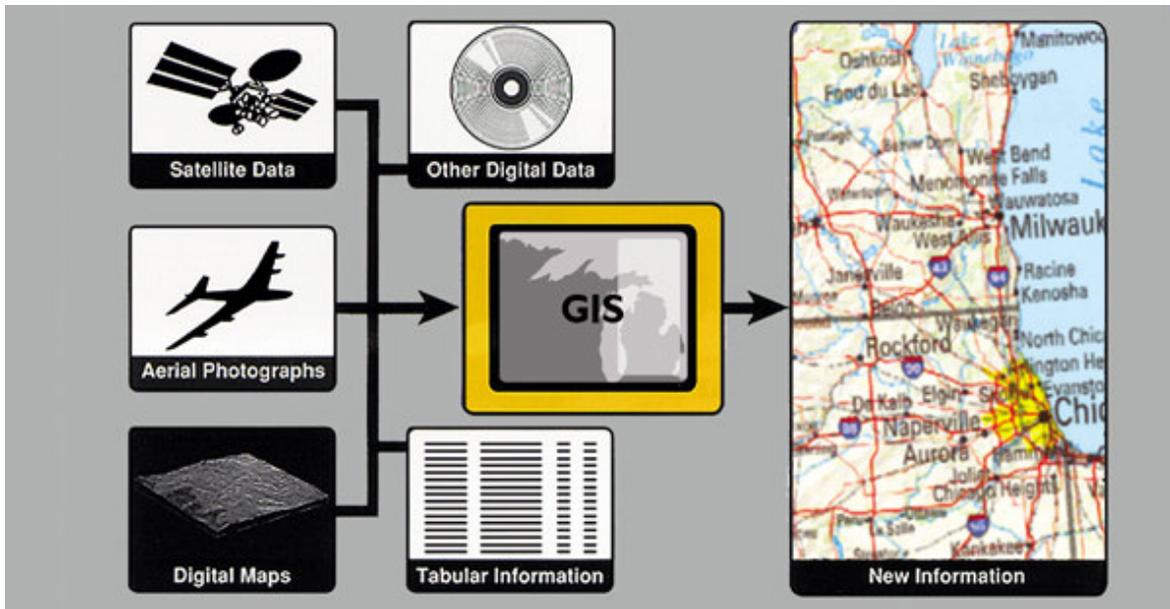
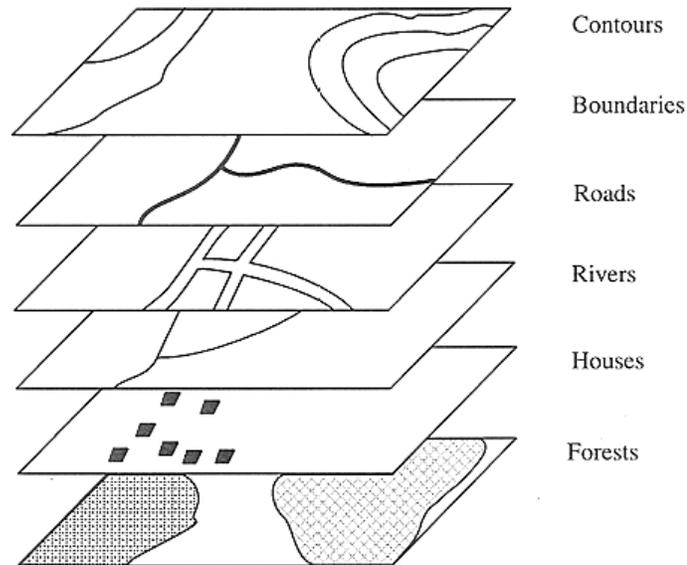


Figura 2.1 - Integração no SIG [28].

Além disso, como resultado, os dados podem ser geoprocessados, recuperando-se também informações e realizando estatísticas.

Os objetos espaciais na representação digital podem ser agrupados em camadas como mostrado na Figura 2.2. Por exemplo, um mapa pode ser dividido em um conjunto de camadas do mapa que consistem nos contornos, limites, estradas, rios, casas, florestas, entre outros [34].



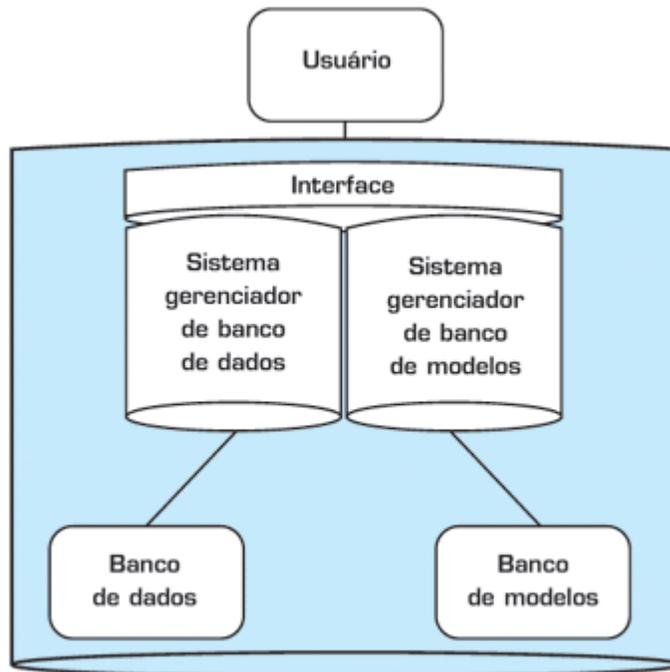
**Figura 2.2 - Exemplo de Camadas do Mapa [30]**

Considerando que o dado espacial corresponde à bacia gerida, as possibilidades que o SIG fornece permitem a aquisição de dados, seu tratamento e visualização de resultados, auxiliando na gestão, possibilitando para o usuário, relacionar os dados espacialmente. Como mostrado anteriormente na integração do SIG, este sistema pode obter informações de diversas fontes, no caso da bacia hidrográfica, dados como cadastro de usuários de água (CNARH [6]), dados pluviométricos e fluviométricos (HidroWeb [12]), que podem ser obtidos através de Internet.

### 2.1.2. Sistemas de Apoio a Decisão

Em [5], são apresentadas definições para SAD como sistemas computacionais – um conjunto de hardware, software e elemento humano – que auxiliam a tomada de decisão sobre problemas não ou parcialmente estruturados, de qualquer nível ou natureza, considerando normalmente aspectos e critérios variados.

O modelo conceitual proposto por SPRAGUE e WATSON [26] de um SAD agrega dois bancos – bancos de dados e bancos de modelos – seus gerenciadores (SGBD – Sistema Gerenciador de Banco de Dados e SGBM – Sistema Gerenciador de Banco de Modelos, respectivamente) e uma interface de diálogo com o usuário. Estes componentes formam a arquitetura básica de um SAD mostrada na Figura 2.3, a seguir.



**Figura 2.3 - Componentes de um SAD [26] apud [5].**

Os modelos que compõem o SAD são utilizados para inferir sobre os dados e apontar melhores soluções, considerando que a solução auxilia na tomada de decisões. Na base de dados, podem ser extraídas as informações necessárias ao apoio à decisão, sendo a interface gráfica responsável pela interação com o usuário, procurando conhecer as suas necessidades e apontar melhores caminhos.

Para o desenvolvimento do SAD deve-se contar com a efetiva participação do decisor e profissional de TI (Tecnologia da Informação). A interação entre estas duas categorias deve ser capaz de permitir mudanças rápidas e eficazes ao sistema [4]. Dessa forma uma estratégia indicada para desenvolvimento deste tipo de sistema é adaptativa, interativa e incremental, moldando-se às necessidades do decisor, segundo SPRAGUE e WATSON [26] *apud* [5].

Segundo SHIM *et al.* [25] *apud* [5], o sucesso de um SAD está numa interface bastante amigável e simples, capaz de acessos interativos, incorporando tipos como menus, ícones, telas sensíveis ao toque e adaptável à necessidade do usuário, além de componentes de gerenciamento de dados.

De acordo com esta explicação, a criação de um componente que represente os dados e as relações graficamente, como o Componente Tema, busca de certa forma atender estas necessidades.

### 2.1.3. Sistemas de Apoio a Decisão Espacial

Por possuir características de um SIG e SAD, o SADE pode relacionar dados e integrá-los, a ponto de facilitar o entendimento pelo usuário, auxiliando no caráter decisório. Além disso, existe a integração entre dados e modelos, possibilitando a criação de modelos dotados de características espaciais que darão suporte à decisão.

Neste caso, o banco de dados também contém dados geograficamente referenciados, realizando uma integração do sistema com o mundo real.

Considerando o acoplamento existente no SADE, que permite a integração entre a base de dados geográfica e modelagem científica através de um controle, que pode ser classificado desde fraco até pleno [20], podendo considerar apenas o fraco e o pleno para classificar o SADE-GBHidro, do ponto de vista de componentes e sistema, respectivamente.

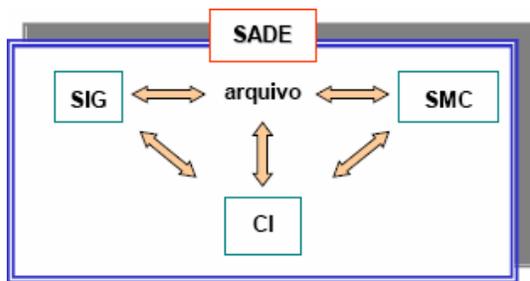


Figura 2.4 - Acoplamento Fraco no SADE [20]

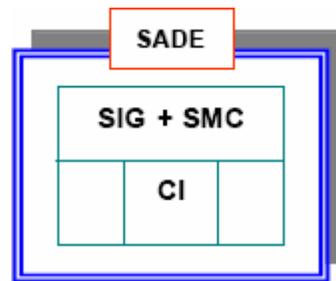


Figura 2.5 - Acoplamento Pleno no SADE [20]

A Figura 2.4 e a Figura 2.5, representam estes dois níveis de acoplamento, nas quais SMC é o Sistema de Modelagem Científica, que corresponde à base de modelos e o CI é o Controle de Integração, que realiza a integração entre as bases.

Em outras palavras, no ponto de vista dos componentes do sistema, o acoplamento seria considerado fraco, pois cada elemento, existente na Figura 2.4, possui um componente responsável por sua gerência. Já por partilharem a mesma plataforma do sistema e fornecer tal integração, no ponto de vista do sistema, o acoplamento deverá ser classificado como pleno, conforme a Figura 2.5.

A integração entre SAD e SIG fornece vantagens únicas, para gestão de água nos seguintes aspectos [14] *apud* [18]:

- Representação das relações espaciais do mundo real numa forma visual e analítica;
- Capacidade de integração de técnicas dos componentes sócio-econômicos, ambientais e físicos numa base de dados ampla; e
- Capacidade de integração de técnicas de simulação e otimização na modelagem para apoiar a resolução de problemas.

Além disso, agregam uma dimensão espacial aos dados existentes, requerida geralmente nos problemas parcialmente ou não-estruturados, que, segundo NETO [20], decorrem de processo do meio ambiente natural.

## **2.2. Modelagem e Representação Utilizadas**

Para o desenvolvimento deste sistema, de acordo com o que foi visto anteriormente, considerando que um SAD possui uma base de modelos além da base de dados, é necessário definir os modelos que serão utilizados.

Além disso, considerando que este SADE possui um elemento intermediário de integração entre a base de dados geográfica e os modelos, definido como uma abstração de dados focada para a aplicação de modelos. Este elemento possui uma representação gráfica específica, onde suas origens devem ser definidas também.

### **2.2.1. Modelo de Otimização GAMS (General Algebraic Modelling System)**

A modelagem científica utilizada para apoio a decisão, neste caso, consiste num modelo matemático de otimização que implementa as relações do mundo real, no qual seu resultado pode servir como referência de um ponto de equilíbrio na tentativa de se atingir o maior benefício econômico social.

Segundo MORAES [18], uma avaliação da melhor utilização da água pode ser feita hidrológica [32] e economicamente, avaliando-se os benefícios obtidos pelos usuários, através de funções de produção e custo benefício, bem como através de restrições físicas e legais, como quantidade e qualidade, relacionando a oferta e a demanda por água.

Para resolução de problemas que possam ser solucionados através de programação matemática, buscando valores ótimos através de algoritmos numéricos, foi usado um sistema chamado GAMS (*General Algebraic Modelling System*) [9], que identifica a melhor alocação de água, levando em conta suas restrições. Para utilização deste sistema é necessária a criação de um modelo de otimização. Resumidamente, o modelo de otimização criado no GAMS consiste numa codificação textual na linguagem específica utilizada pelo sistema, compreendendo um arranjo entre parâmetros e variáveis indexados, ou seja, relacionados a determinados índices. Este arranjo é submetido a comandos, restrições e equações, no qual o objetivo é a maximização ou minimização de uma determinada variável, estando esta associada à função objetivo do modelo, de acordo com ROSENTHAL [23] e BROOKE *et al.* [3]. A partir desta codificação, o modelo pode ser compilado e executado, fornecendo, ou não, valores ótimos como resultado. Um exemplo simples de um modelo GAMS é apresentado no Anexo D.

Anteriormente, foi criado um modelo de otimização econômico-hidrológico para a Bacia do Pirapama, realizando o mapeamento dos dados e desenvolvendo funções de produção e benefício, desenvolvido na UFPE por MORAES [18]. Foram utilizados dados geográficos que compuseram posteriormente o projeto do SIGMA Pirapama [21], enquanto que a similaridade e médias das observações auxiliaram o cálculo de vazões específicas.

Atualmente, em um projeto iniciado em 2005 (Projeto SADE-SF) com o apoio do FINEP [8] através do fundo setorial CT-HIDRO [7], é possível contar com uma equipe maior para aplicação de um modelo hidrológico, criação de um novo modelo de otimização e uma base de dados geográfica para a Bacia do São Francisco, e desenvolvimento do sistema. Diante disso, o SADE-GBHidro [19] é um sistema que permite, através de uma interface gráfica amigável, a integração destes dados, geração de modelos e visualização de resultados gráficos, dando um maior suporte à tomada de decisões.

O modelo possibilita a inclusão de um grande número de relações físicas e econômicas, além de algumas institucionais e outras agronômicas. Os seguintes componentes podem ser incluídos dependendo da aplicação [19]:

1. Componentes de Hidrologia Qualitativa – estes componentes sempre estarão presentes em qualquer aplicação; levam em conta o balanço hídrico com respeito à quantidade, nos nós que representam os diversos trechos do rio, bem como nos seus afluentes e reservatórios;
2. Componentes de Qualidade das Águas – pode incluir o balanço, bem como o processo de decaimento e transporte ao longo do rio, de constituintes básicos utilizados para avaliar qualidade de água;
3. Componentes Econômicos – estes também sempre estarão presentes, através de funções de benefício líquido para os usos consuntivo e não-consuntivo, podendo ser simuladas funções de produção para os principais produtos da bacia levando em conta a água alocada como insumo;
4. Componentes Institucionais – regras e políticas institucionais tais como outorga, níveis mínimos de constituintes de qualidade permitidos, fluxos mínimos exigidos por demandas ecológicas, entre outros; e
5. Componentes Agronômicos – são consideradas diferentes produtividades de culturas em função da água aplicada, e do mês que a mesma foi aplicada, bem como dos estágios fenológicos da planta.

### 2.2.2. Representação em Rede de Nós e Links

Para iniciar a criação deste modelo de otimização GAMS é necessário mapear os elementos do mundo real, atributos, relações, condições e restrições, utilizando uma representação específica. Neste caso, a representação utilizada na criação deste modelo baseou-se na rede de nós e *links* desenvolvida por MCKINNEY *et al.* [13], representando os objetos que fazem parte da bacia. A partir desta representação, foram realizadas adaptações devido ao cenário atual e às necessidades do modelo, bem como adicionado o modelo qualitativo [18].

Para representar os elementos mais importantes que serão utilizados no modelo e suas relações, torna-se necessário determinar sua simbologia e estrutura. Além disso, o mundo real possui valores contínuos, enquanto que os elementos representados são discretos. Assim, deverá haver uma discretização da bacia, onde deverão ser determinados trechos do rio a serem modelados.

Dessa forma é realizado um mapeamento entre os elementos espaciais e suas relações, como observado na Figura 2.6.

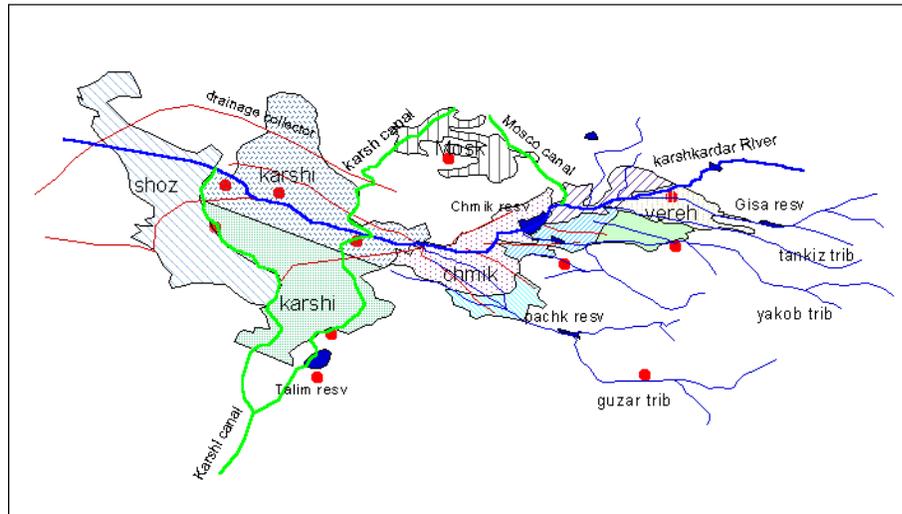


Figura 2.6 – Dados espaciais: Bacia do rio Karshikardaya na Ásia Central [13]

Utilizando estes elementos espaciais cria-se a representação que pode ser observada na Figura 2.7, a seguir, originada a partir destes dados espaciais, na qual podem ser observadas a simbologia e estrutura utilizada para a representação da rede de nós e links.

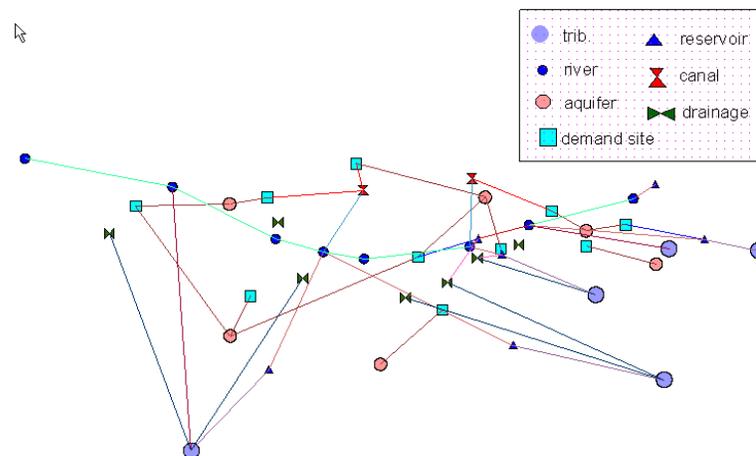


Figura 2.7 – Dados do Tema: Um exemplo de rede para a bacia do rio Karshikardaya [13]

Baseando-se neste tipo de representação e modelagem foi criado um modelo de otimização para a bacia do rio Pirapama, além de uma representação de forma manual, que pode ser observado no Anexo A.

O modelo representa a bacia hidrográfica como um todo, através de uma rede de nós e *links* [22], no qual os nós representam entidades físicas e os *links*, as conexões entre estas entidades. Os nós podem ser de dois tipos: os nós-fonte, que representam entidades físicas tais como: rios, reservatórios, águas subterrâneas, entre outros; e os nós-demanda, que podem ser: indústrias, cidades, centrais hidroelétricas, entre outros [19].

### 2.2.3. Base de Dados de Temas

Segundo YURAS [34], as entidades do mundo real são tão complexas que se levando em consideração a utilização de uma base de dados geográfica, os dados encontrados dentro desta base podem ser classificados em espaciais e de temas(ou não espaciais), que correspondem às informações de posicionamento geográfico e aos atributos relacionados aos mesmos objetos espaciais, respectivamente. Os objetos em uma base de dados espacial são definidos como representações destas entidades com atributos associados, nas quais cada atributo define suas características próprias.

Os valores dos atributos encontram-se freqüentemente em tabelas de atributos que estabelecem relacionamentos entre estes atributos e os dados espaciais tais como objetos ponto, linha e área, e também entre os atributos [34].

Da mesma forma, os objetos do tema constituem dados não espaciais, os quais são utilizados na criação de modelos. Nesta modelagem, a rede se torna uma ponte entre o mundo real e os modelos matemáticos, nos quais os objetos espaciais são nós e as relações espaciais são *links* [13].

No SADE-GBHidro, esta base de dados de temas é estendida, compondo uma base distinta da base geográfica, embora boa parte de seus dados sejam originados a partir dela, através de uma integração de dados específica. No acoplamento do SADE, no nível dos componentes do sistema, ela corresponde ao arquivo intermediário de integração entre as bases de dados geográficos e modelos do sistema, conforme a Figura 2.4.

Esta divisão da informação nesta base de dados de temas possibilita ao usuário, realizar uma melhor análise e compô-la de forma mais detalhada e focada no modelo a ser

desenvolvido. Nesta integração é feita uma filtragem dos elementos, passando apenas os mais relevantes, reduzindo o nível de complexidade do sistema, considerando que agora todos os elementos da base de temas serão agregados ao modelo.

Essa redução da quantidade de elementos que compõe esta base permite ao usuário, uma maior compreensão das informações e reduz também o número de variáveis do modelo, melhorando seu desempenho. Além disso, como o modelo demanda determinadas informações dos elementos que o compõem, diminui também a necessidade de buscá-las.

Além disso, a expressão “de temas” pode ser referir a algo que está relacionado a um determinado tema, e, para o sistema, este tema consiste numa abstração discretizada da bacia, focada no modelo de otimização.

### **2.3. Elementos da Linguagem Utilizados**

Para decidir que a linguagem de programação utilizada seria VB.NET (Visual Basic .NET) [17], foram avaliados vários fatores. Inicialmente, seria trabalhado com o software ArcGIS [1] para desenvolvimento da base de dados geográfica e criação do componente responsável pelo seu gerenciamento, que possui integração com VB. Entretanto, devido à necessidade do usuário final do sistema possuir uma licença do produto, optou-se posteriormente pelo TatumGIS [27], atualmente em avaliação e já utilizado em outro projeto do mesmo órgão financiador, CT-HIDRO, que também fornece suporte à linguagem VB.NET.

Além desse fato, em outro projeto parceiro, já foi desenvolvida uma aplicação para gerência da bacia do Pirapama utilizando VB 6.0. Deste modo, pensou-se em reutilizar algo já desenvolvido. Entretanto, esta reutilização não foi posta em prática.

Por ser um SADE, no qual a interface gráfica constitui num elemento importante do sistema, a experiência adquirida anteriormente com a referida linguagem, por constituir uma plataforma relativamente nova e a facilidade para criação de interfaces gráficas, pesou na decisão de escolha desta linguagem. Contudo, a utilização desta linguagem é baixa, considerando a principal origem dos bolsistas desenvolvedores, o Centro de Informática da Universidade Federal de Pernambuco (CIn-UFPE), além de outras universidades do estado de Pernambuco, mas por ser relativamente simples, possui uma boa e rápida adaptabilidade.

Dessa forma, são descritos, a seguir, dois elementos da linguagem utilizados no seu desenvolvimento: componentes de software e dados XML.

### 2.3.1. Componentes de Software

Recentemente, a tecnologia de componentes de software, que é baseada na construção de sistemas de software a partir de componentes reusáveis, tem atraído a atenção por conta da capacidade de redução dos custos com desenvolvimento de software [4].

Esta abordagem de desenvolvimento baseado em componentes, segundo HAINES *et al.* [10], aumenta a flexibilidade e a manutenibilidade dos sistemas, podendo ser usada para montar sistemas rapidamente, reduzir custos e a manutenção associado com o suporte e o *upgrade* (melhoramento e atualização) de grandes sistemas.

As características desta abordagem são [4]:

- Reuso como caixa-preta;
- Granularidade do componente e controle reativo;
- Interfaces especificadas contratualmente; e
- Mercado de componentes de software.

Componentes de software compreendem uma área de estudo da engenharia de software, baseada nas teorias de objetos, arquitetura, plataforma e padrões de projeto de software, e da vasta teoria de programação e projeto orientado a objetos [33].

Segundo Wikipedia [33], um componente de software é um elemento do sistema que oferece serviços predefinidos e comunica-se com outros componentes. Para que um componente de software seja considerado como tal, ele deve obedecer aos seguintes critérios: possuir múltiplo uso; não ter um contexto específico; ser composto por outros componentes; ser encapsulado (i.e. não investigável através de suas interfaces); e constituir uma unidade independente de desenvolvimento e versão.

Uma outra definição é a de software componente, que segundo Webopedia [30] também chamado de “*componentware*”, consiste em um software projetado para trabalhar como um componente de uma extensa aplicação. Uma boa analogia é a forma como os computadores são montados, a partir de uma coleção de componentes padronizados. Pelo

fato das interfaces entre os componentes serem padrão, é possível misturar componentes de diversos fabricantes num único sistema.

Da mesma maneira, o objetivo do software componente é padronizar interfaces entre os componentes de software, para que trabalhem juntos significativamente (...). Muitos analistas acreditam que o software componente é uma extensão natural da programação orientada a objeto, e que se tornará um paradigma de programação padrão dos próximos anos [30].

Assim, conforme HEINEMAN e COUNCILL [11], um componente implementa ou realiza uma interface se o componente tiver uma implementação de todas as operações definidas por ela, que também oculta a implementação pelo componente. Um componente necessita de uma interface se o mesmo requer uma interação definida, na qual outros elementos de software possam implementar esta mesma interface. Um componente pode ser incapaz de realizar uma interface se uma de suas interfaces requeridas não for cumprida.

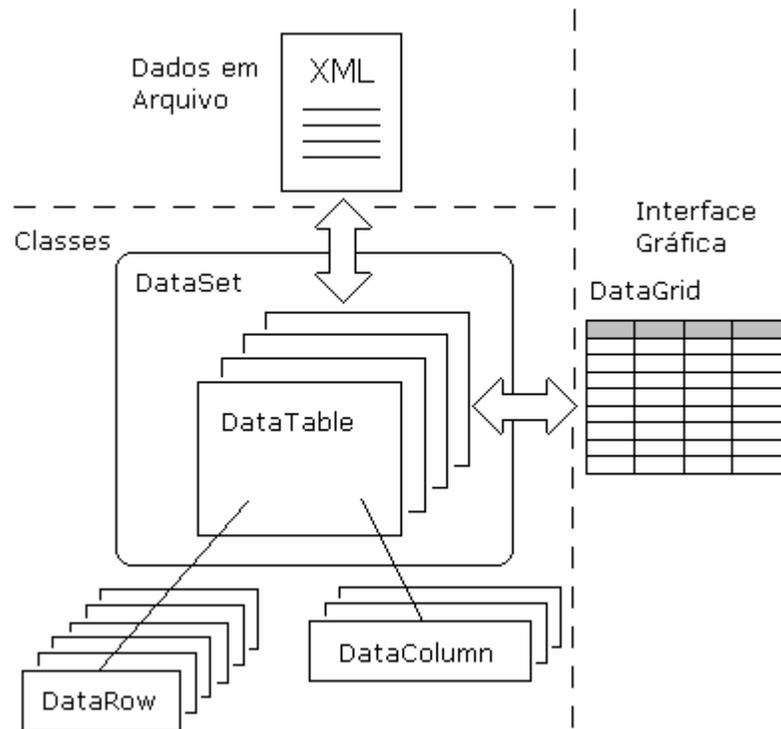
Em VB.NET, os componentes .NET podem ser adicionados aos projetos de software como uma biblioteca de ligações dinâmicas, comumente chamadas de DLL (*Dynamic Link Library*).

Na plataforma .NET, a DLL possui um conjunto de funções executáveis ou dados que são usadas pela plataforma e podendo ser usada por várias aplicações ao mesmo tempo. Algumas DLL são encontradas no sistema operacional Windows [15] e disponível a qualquer aplicação Windows. Outras DLL são desenvolvidas para uma aplicação em particular e são carregadas com a mesma [31].

A abordagem utilizada para desenvolver os componentes, foi a utilização de *UserControl* [16], ou seja, um elemento gráfico customizado e com controle autônomo.

### 2.3.2. Dados XML (eXtensible Markup Language)

A tecnologia XML [29] tem se firmado como um padrão para integração de dados entre diversos tipos de sistemas, sendo um arquivo de texto, facilitando sua mobilidade. Visando essa interoperabilidade e com esta tecnologia, a plataforma .NET possui formas de integração, uma delas é através da classe *DataSet*, como é mostrada na Figura 2.8.



**Figura 2.8 - Integração XML na plataforma .NET**

Nesta forma de integração, o *DataSet* é um objeto gerenciador dos dados, recuperando e armazenando-os em arquivos XML, assim como os fornecendo ao *DataGrid*, que, por sua vez, compõe a interface gráfica.

*DataSet* é uma classe que possui uma coleção de tabelas, na qual cada tabela corresponde a um objeto da classe *DataTable*. Analogamente como uma tabela possui colunas e linhas, este objeto *DataTable* possui também uma coleção de colunas, que consiste num objeto *DataColumn*, e uma coleção de linhas, que da mesma forma corresponde ao *DataRow*.

Com relação à integração XML, as tabelas correspondem à definição dos elementos dentro do arquivo XML, enquanto que as colunas, a definição dos seus atributos, ou elementos internos, sendo cada elemento equivale a um registro ou linha desta tabela.

Os métodos de integração do *DataSet* com XML são *ReadXML* e *WriteXML*, para ler e escrever em arquivo XML, respectivamente. Os arquivos XML armazenam os dados de forma textual, e devido à necessidade de armazenar diversos tipos de dados, assim como manter a estrutura e relação entre eles, o *XMLSchema* torna-se responsável por isso. Dessa

forma, o *DataSet* fornece os métodos *ReadXMLSchema* e *WriteXMLSchema*, para ler e escrever a estrutura de dados XML num arquivo XSD, respectivamente.

Um exemplo desta integração XML na plataforma .NET pode ser obtido em SANT'ANNA [24].

## 2.4. Conclusões

Assim como a utilização de SADE na gestão de bacias, na qual foram apresentados o SIG e o SAD, foi também apresentado o modelo, a representação utilizados e sua relação com a base de dados de temas.

Além disso, elementos da linguagem utilizados no desenvolvimento, fortalecem o conhecimento sobre a implementação da aplicação.

A partir da definição destes conceitos relacionados diretamente ao desenvolvimento do SADE-GBHidro e do componente tema, é possível abordá-los posteriormente com uma maior clareza. Dessa forma, no Capítulo 3, será detalhado o SADE-GBHidro, e no Capítulo 4, o componente tema.

## 3. O SADE-GBHidro

O SADE-GBHidro consiste numa ferramenta de apoio a decisão utilizada na gestão de recursos hídricos. Este sistema foi apresentado anteriormente por MORAES *et al.* [19], tendo a colaboração do autor do presente trabalho de graduação. Sendo assim, este capítulo compreende uma descrição deste sistema para poder entender o contexto em que o componente tema está inserido.

Com isso, são apresentadas estratégias utilizadas no seu desenvolvimento, sua arquitetura, seus componentes, bem como seu fluxo de operações. Além disso, são mostrados diagramas de casos de uso do sistema representando as suas funcionalidades principais, incluindo o componente tema e tabela, do qual o tema é dependente.

### 3.1. Estratégias de Desenvolvimento

O sistema, dentro do escopo do projeto, deverá disponibilizar uma base de dados geográfica para a bacia do rio São Francisco (restringindo-se à região compreendida entre as Barragens de Sobradinho e Xingó) e para a bacia do Pirapama, bem como modelos de otimização econômico-hidrológicos integrados e construídos para cada uma delas.

Como já existia um modelo de otimização voltado para a bacia do rio Pirapama, iniciou-se, com este modelo, a base de modelos. Já as bases de dados geográficas, necessárias para ambas as bacias, foram iniciadas simultaneamente.

Como o modelo para o Pirapama já foi criado, foi desenvolvida inicialmente uma versão do sistema, incorporando suas necessidades. Além disso, pela base de dados geográficos do Pirapama ainda não estar pronta, optou-se por desenvolver o sistema no sentido contrário ao fluxo normal das operações, desenvolvendo na ordem: o gerenciador de modelos de otimização, o gerenciador de dados do tema, o gerenciador de dados geográficos e, internamente, os seus integradores. Assim, utilizou-se, em alguns casos, uma engenharia reversa dos dados já obtidos para compor seus dados originais.

Posteriormente, serão realizados ajustes para adequar o sistema às necessidades encontradas no desenvolvimento para a bacia do São Francisco, considerando que estes ajustes serão, em sua maioria, nos arquivos que compreendem as estruturas de apoio.

Portanto, existirão poucas alterações na aplicação, devido à flexibilidade fornecida pelas estruturas de apoio, que são arquivos XML com tabelas que refletem a forma como deve ser realizada determinada conversão.

Dessa forma, o sistema pode se adaptar à entrada de novos modelos, ser aplicado a outras bacias, reduzindo as dificuldades na conversão dos dados. Assim, para qualquer alteração na conversão é realizada, apenas são alteradas tais estruturas, sem alteração no código fonte do software.

Dessa forma, o sistema tenta ser o mais genérico possível, pois além da mudança contínua nos requisitos, devido a possibilidades de alterações no modelo e na representação, o sistema tenta ser adaptável, e a melhor forma de conseguir é criando componentes genéricos. Dessa forma, como os formatos dos dados geográficos e tabulares não mudam, assim como a estrutura do código GAMS continua a mesma, o único componente que necessita de adaptações seria o componente tema, que deverá ser genérico.

Devido aos diferentes tipos de dados envolvidos e visando reduzir custos de desenvolvimento e flexibilização do sistema, uma estratégia foi seu desenvolvimento baseando-se na criação de componentes [10], realizando um gerenciamento centralizado e distribuído, na qual cada componente gerencia um determinado tipo de dado, que constituem os Arquivos de Projeto (AP), enquanto o sistema gerencia todos os componentes, sessões e projetos.

Esta abordagem na utilização de componentes internos - diferentemente da utilização de componentes prontos de terceiros, e que também é feito neste projeto - é bastante interessante, pois aumenta o paralelismo no desenvolvimento (dividir pra conquistar) e sua reutilização. Além disso, o fornecimento interfaces bem definidas permite criação de novos componentes e auxiliam na integração do sistema.

Em resumo, os pontos fortes do projeto são:

- A utilização de componentes, sendo eles genéricos ou de utilização geral;
- A realização de um protótipo inicial com a bacia do Pirapama, com engenharia reversa dos dados para criação completa do sistema; e

- A utilização de arquivos de estrutura para conversão de dados, ao invés de modificar o comportamento do software.

### 3.2. Arquitetura do Sistema

De acordo com a arquitetura de um SADE, abordada anteriormente, surge a idéia de criação de um sistema para apoio a gestão de bacias, chamado SADE-GBHidro. Neste sistema, a base de dados é dividida em uma base de dados geográfica e outra de temas, esta última utilizada para representar os dados de uma forma mais voltada para aplicação de modelos, na qual ambas são integradas pelo sistema com o objetivo de criar modelos de apoio a decisão, que forneçam subsídios científicos.

Dessa forma, a arquitetura do SADE-GBHidro, considerando interface, gerenciadores, integradores e base de dados e modelos, pode ser mostrada na Figura 3.1.

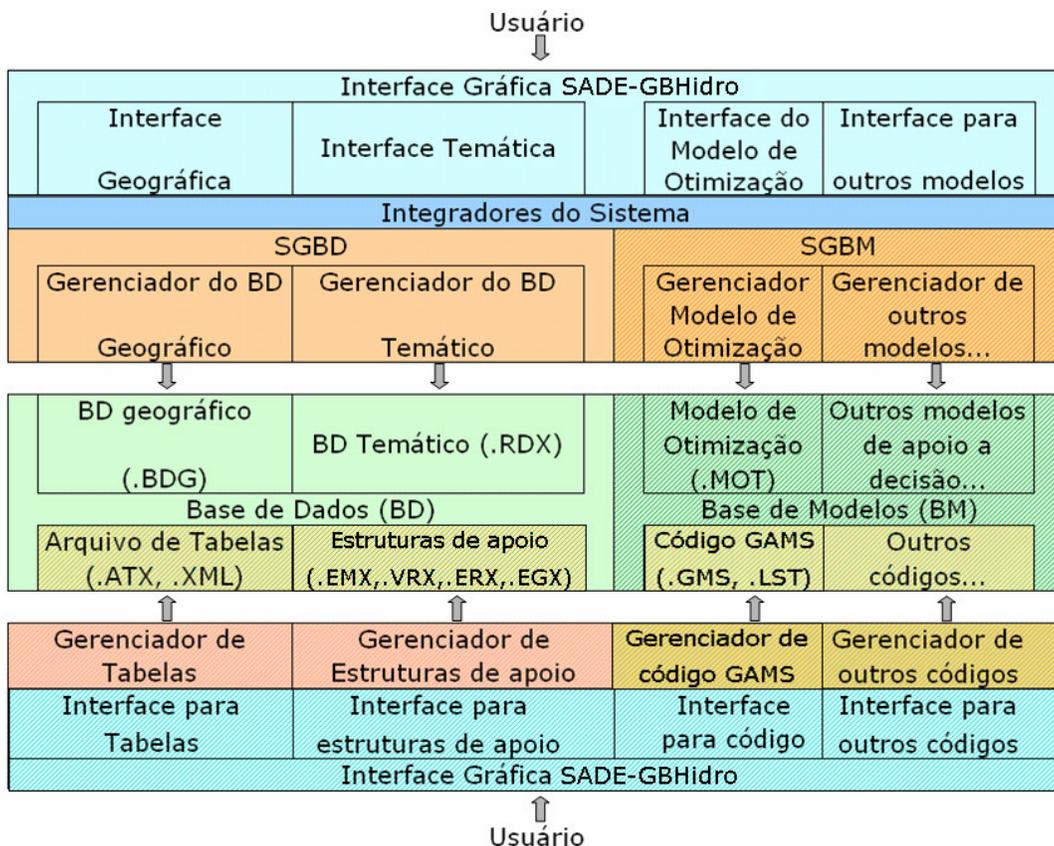


Figura 3.1 - Arquitetura do SADE-GBHidro

A base de modelos atualmente é composta apenas por modelos de otimização GAMS, prevendo ser futuramente adicionados outros tipos de modelos. Considerando que

o objetivo deste trabalho engloba o desenvolvimento da base de dados para gestão de recursos hídricos, serão abordados os gerenciamentos destes dados, bem como toda integração e interação com o usuário.

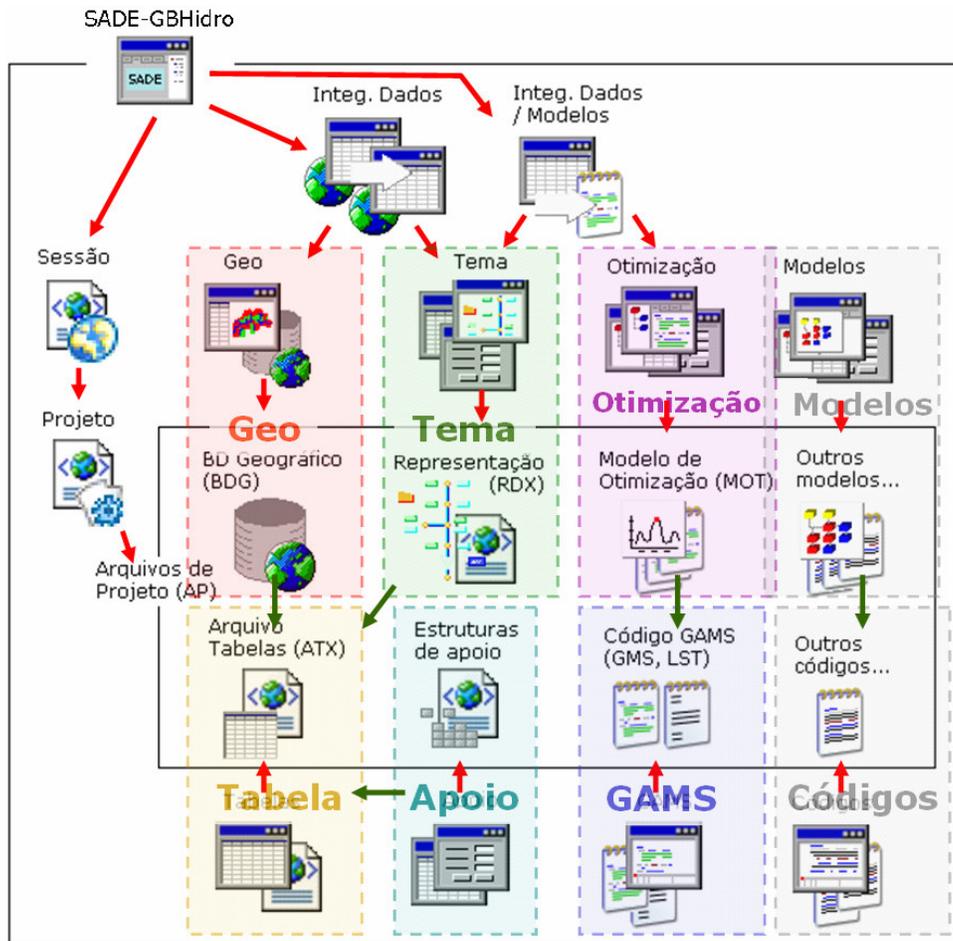
Nesta arquitetura da Figura 3.1, existem dois acessos pelo usuário: superior e inferior. O superior está relacionado à utilização das bases de dados, modelos e integrações, comumente realizadas por decisores. Já o inferior está relacionado aos arquivos de tabelas, estruturas de apoio a integrações e codificações do modelo, que corresponde a um acesso avançado, destinado a um usuário modelador.

De acordo com o que é visto, pode-se considerá-la como uma aplicação em três camadas: interface gráfica; controle e dados (lembrando de incluir nesta camada os modelos). Verticalmente pode-se afirmar que a união entre interface gráfica e controle corresponde a cada componente, enquanto que os dados são tratados como AP.

Nessa arquitetura, a interface e gerência das estruturas de apoio não foram implementadas, sendo estas gerenciadas pelo componente tabela. Da mesma forma, a interface e gerência dos modelos e códigos constituem exemplos para futuras agregações de novos tipos de modelos no sistema.

### **3.3. Utilização de Componentes no Sistema**

Como o SADE-GBHidro foi desenvolvido baseando-se na criação de componentes, os componentes do sistema, bem como os elementos gerenciados – sessão, projeto e AP – e suas relações também podem ser mostrados na Figura 3.2, a seguir.



**Figura 3.2 - Componentes do SADE-GBHidro**

Na Figura 3.2, o sistema é composto de alguns componentes que são responsáveis por um gerenciamento distribuído dos arquivos. Dessa forma, assumindo o papel de um SIG, o componente geográfico (Geo) gerencia a base de dados geográfica. Já o componente tema gerencia a base de dados de temas, enquanto que o componente tabela além de gerenciar as tabelas do sistema, conforme a integração XML, detalhada na secção 2.3.2, pode ser utilizado por outros componentes para gerência das tabelas que possuem, sendo essa dependência observada através das setas verdes entre os componentes. Estes três componentes: geográfico, tema e tabela, são os componentes principais nos quais são realizadas a entrada e representação dos dados no sistema. Além disso, é mostrado o componente de apoio, que ainda não possui implementação, assim como é mostrada a dependência entre os modelos e a gerência do código fonte, tanto na otimização GAMS, que já está implementada, como no exemplo associado à aplicação de novos modelos.

Em relação a como os arquivos de projeto estarão dispostos dentro do sistema, cada arquivo de projeto poderá estar alocado a um projeto, e o sistema poderá abrir vários projetos e arquivos de projeto, que compõem uma sessão. Assim, a aplicação só poderá estar com uma sessão aberta por vez, que pode conter vários projetos. Esta hierarquia é utilizada para facilitar a reutilização de projetos por diferentes sessões, compartilhando os AP, pois o sistema realiza várias atividades e trabalha com diferentes tipos de dados e arquivos. A hierarquia proposta é mostrada na Figura 3.3, a seguir:

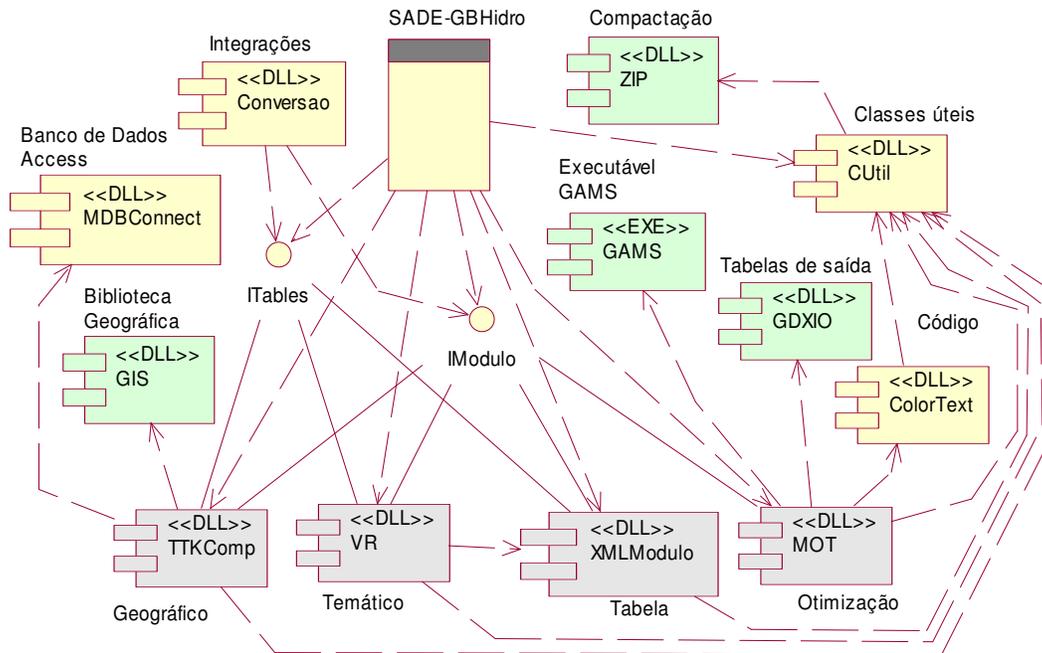


**Figura 3.3 – Exemplo da hierarquia utilizada pelo sistema**

Como pode ser observado na Figura 3.3, o “projeto 3” possui um ícone diferenciado (cor cinza), e os seus AP estão desabilitados. Isso acontece pelo fato do “projeto 3” ser um projeto protegido, ou seja, é um tipo de projeto no qual seus AP são utilizados apenas para integrações e não para serem abertos. É necessário utilizar este tipo de projeto para proteção dos arquivos e compartilhamento.

A idéia de criação de componentes, além de uma melhor organização, tem-se o aumento de produtividade, pelo reuso e paralelismo das atividades, e a integração de todas as partes do sistema. Inicialmente, foi proposta a utilização de vários módulos, buscando facilitar o entendimento das atividades de conversão e a distinção entre os elementos convertidos e suas funcionalidades específicas. Entretanto, a criação destes componentes e a utilização de um sistema unificado, também viabilizam este entendimento.

Considerando agora os componentes atualmente desenvolvidos, suas interfaces e relações com componentes do mundo real, pode ser mostrado o seguinte diagrama de componentes, exibido na Figura 3.4.



**Figura 3.4 - Diagrama Atual dos Componentes do SADE-GBHidro**

No diagrama da Figura 3.4, percebe-se que os componentes estão representados por três cores, no qual a cor verde representa componentes de terceiros adicionados ao sistema como a biblioteca geográfica (GIS) e a de compactação de arquivos (ZIP).

Além delas o GAMS é internamente executado local ou remotamente, sendo também referenciada uma biblioteca, GDXIO, utilizada para obtenção das tabelas de saída, como resultados das otimizações aplicados também em gráficos.

Pelo diagrama da Figura 3.4 corresponder a situação atual do sistema, ainda não foram criados os componentes de Apoio e de gerenciamento de novos modelos, assim como o componente geográfico, representado pelo TTKComp, ainda não possui referência ao componente tabela.

Já a cor cinza representa os componentes que gerenciam AP, implementando também a interface IModulo. Eles correspondem aos principais componentes do sistema: geográfico, tema, tabela e otimização. Os demais componentes em amarelo são componentes auxiliares e de integrações, como a conexão com banco de dados Access, todas as classes úteis do sistema, o componente responsável pela gerencia de código fonte GAMS e o componente de conversão, utilizado nas integrações.

No diagrama da Figura 3.4, também é possível ver a forma de integração entre eles, na qual são principalmente utilizadas as referências DLL como biblioteca de classes (*Class Libraries*) e cada componente gráfico consiste em um *UserControl*. Além desta forma de integração, são apresentadas duas interfaces utilizadas no acesso aos componentes – *IModulo* e *ITables*.

Estas interfaces padrão são utilizadas para definir as formas de interação do componente com o sistema. Neste caso, a interface *IModulo* é responsável pela interação em relação à gerência dos AP pelo sistema, como: abrir, salvar, fechar e notificar alterações. Já a *ITables* é utilizada principalmente nas integrações, pois ela fornece o acesso às tabelas dentro do AP, cujos dados são usados pelas integrações.

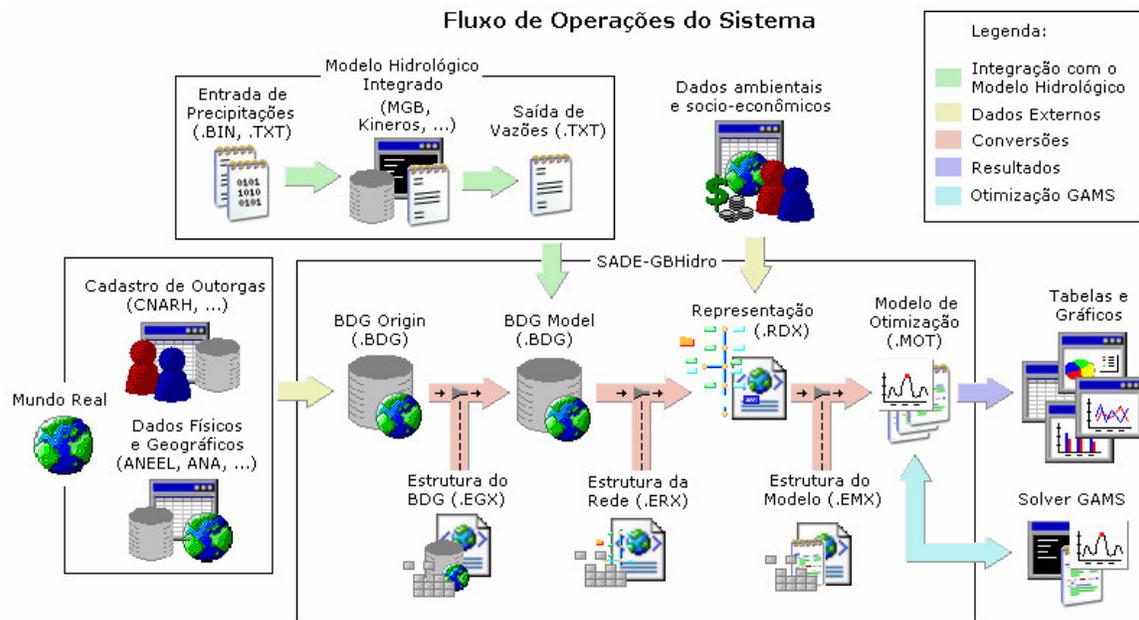
A interação (comunicação) entre o sistema e os componentes através destas interfaces é bilateral, de modo que o sistema atua no componente através de métodos (*sub e function*) e atributos (*property*) implementados pelo componente. Da mesma forma o componente atua no sistema através de eventos disparados pelo componente, tendo o sistema que capturar os eventos, como uma espécie de notificação.

A especificação destas interfaces, representando estas interações é apresentada no Anexo F, assim também como outras interfaces utilizadas internamente ao componente tema.

Os componentes que gerenciam AP são considerados como soluções agregadas ao sistema e, dessa forma, constituem numa unidade independente, passando por todas as etapas de desenvolvimento de software, inclusive testes. Para realização destes testes nestes componentes são criadas pequenas aplicações de testes dos componentes antes de serem testadas a sua integração com o sistema principal, o que também é feito com suas integrações.

### **3.4. Fluxo de Operações**

Para melhor descrever todas as etapas envolvidas no sistema, tais como aquisição de dados, avaliação dos resultados obtidos e integração entre todas as bases: geográfica, de temas e modelos, a Figura 3.5 mostra estas etapas como o fluxo de operações do sistema.



**Figura 3.5 - Fluxo de Operações do Sistema**

Antes de explicar as operações do sistema, é importante definir as entidades representadas. Neste caso, o “Mundo Real” consiste nas fontes de informações geográficas, dentre outras, utilizadas na criação do BDG (Banco de Dados Geográficos). O Modelo Hidrológico Integrado corresponde a uma aplicação distinta que fornece ao SADE-GBHidro, dados de vazão. Outras entradas podem ser encontradas como dados ambientais e sócio-econômicos, assim como as saídas de tabelas e gráficos a partir do modelo de otimização, que utiliza um solver GAMS para obter seus resultados. O solver corresponde a uma ferramenta utilizada para resolução de um determinado problema. No SADE-GBHidro é utilizado o CONOPT3, como solver GAMS.

Dentro do SADE-GBHidro, existem dois BDG; incluindo o BDG Origin que corresponde a um banco que contém uma quantidade grande de informações, que estão parcialmente relacionadas e não processadas. Já o BDG Model, voltado para a geração de modelos, possui informações fortemente relacionadas e geoprocessadas de acordo com as necessidades do modelo.

Também dentro do sistema, a representação de dados (RDX – Representação de Dados em XML), que corresponde à base de dados de temas, pode ser observada junto ao modelo de otimização (MOT). Além dos dados e modelos, existem as informações utilizadas nas integrações, as estruturas de apoio. Elas são EGX (Estrutura Geográfica em

XML), ERX (Estrutura de Representação em XML) e EMX (Estrutura do Modelo em XML).

Tendo sido definidos todos os elementos, podem ser descritas as etapas que compõem o sistema. As etapas partem da aquisição de dados, onde são adicionados diretamente a uma base de dados inicial, definida como BDG Origin, os elementos do mundo real, como dados físicos (hidrografia, uso do solo, entre outros) obtidos a partir de diversas fontes, assim como dados mais dinâmicos como outorgas, que constitui no cadastro de usuários de água.

Outra forma de inserção direta de dados no sistema é através da inserção de dados voltados para o modelo na base de dados de temas (representação), como dados ambientais e sócio-econômicos.

Além disso, uma forma indireta de inserir dados no sistema é através da integração com o modelo hidrológico, que basicamente consiste num sistema calibrada para que a partir de uma entrada de precipitações possam ser obtidas séries de vazão de determinados pontos da bacia. Essas informações são adicionadas ao BDG Model.

As operações mais importantes do sistema são as integrações realizadas entre dados e modelos. Elas correspondem a três conversões:

1. Geração do BDG Model;
2. Geração da Representação; e
3. Geração do Model.

A geração do BDG Model consiste em inicialmente realizar uma escolha e validação nas informações do BDG Origin, em relação ao tipo do dado, sistemas de coordenadas e projeção, além da presença de documentação, como fonte, escala e data de levantamento. A partir desta validação podem ser realizadas funções de geoprocessamento, como posicionar elementos, relacioná-los, realização de diversos cortes, entre outras. A definição das informações necessárias para a conversão, assim como das funções realizadas estão contidas no EGX.

Na geração da representação, devido à complexidade do BDG e com o objetivo de melhorar o desempenho dos modelos, inicialmente são filtrados os elementos que a constituirão a partir de critérios pré-definidos, também permitindo a escolha pelo usuário. Esta conversão consiste num mapeamento dos atributos (dados do tema) encontrados no

BDG para a representação, também com a possibilidade de realizar a conversão dos dados e relacioná-los de uma forma diferente. A estrutura que define esta conversão é o ERX.

Já a geração do modelo, compreende o mapeamento dos dados encontrados na representação que devem compor o modelo. Os elementos criados por esse mapeamento são elementos próprios do modelo e no caso do modelo de otimização correspondem aos sets – conjuntos, subconjuntos, *links* que são índices utilizados pelo modelo para indexar parâmetros, variáveis, restrições e equações, relacionando-os – e parâmetros. Outras informações estáticas, como variáveis, restrições, simulações, equações e outras configurações de execução que estão contidas no modelo são encontradas na estrutura que define como é feita a conversão (EMX). O resultado desta integração é a geração de um modelo de otimização com um conjunto de códigos fonte e suas interdependências.

Para um melhor entendimento, pode-se relacionar o fluxo de operações do sistema com os componentes do sistema, mostrado na Figura 3.6,.

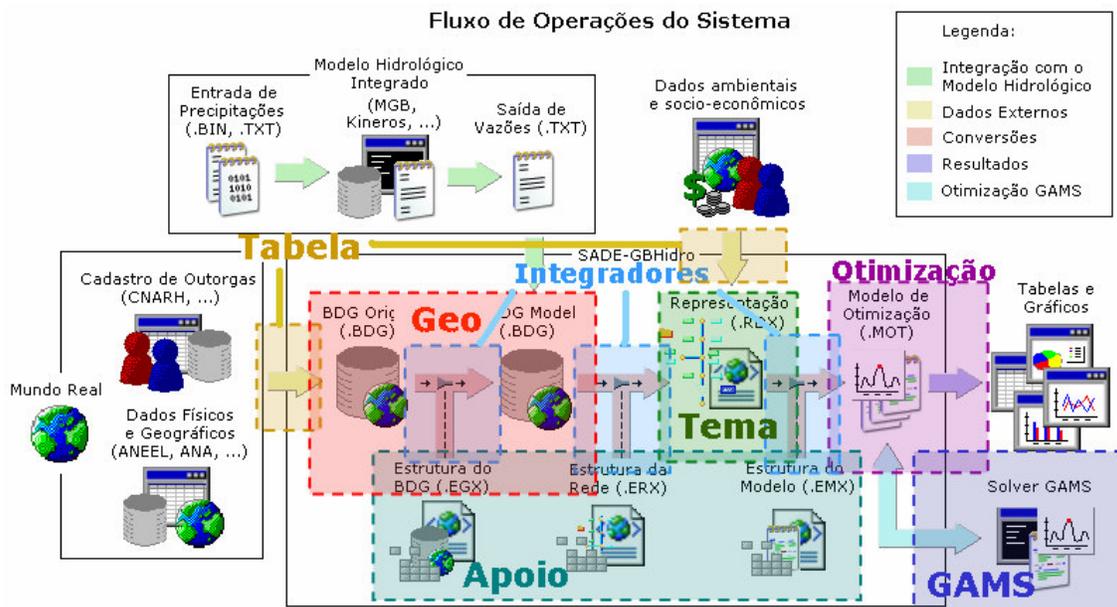
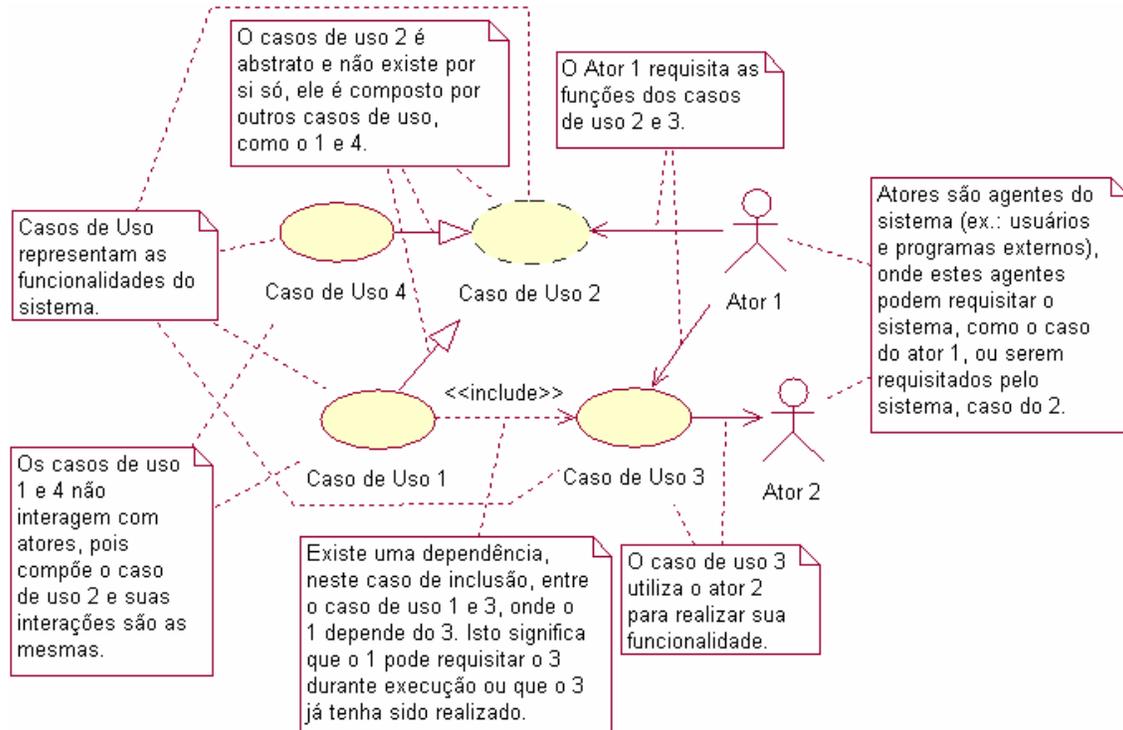


Figura 3.6 – Relação entre os Componentes e o Fluxo de Operações

### 3.5. Requisitos do Sistema

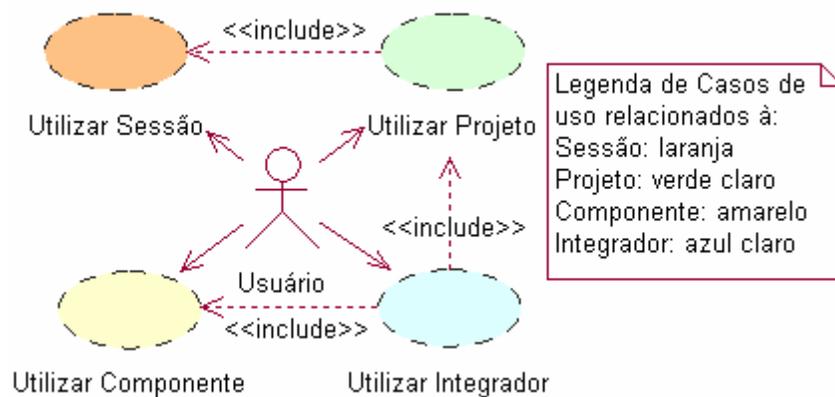
As funcionalidades do sistema correspondem aos requisitos funcionais do sistema que podem ser representados através de um diagrama de casos de uso. Este diagrama é composto por casos de uso, atores e pela relação entre eles. Para um melhor entendimento

deste tipo de diagrama, pode-se descrever brevemente um exemplo contendo representações básicas que são utilizadas, conforme a Figura 3.7.



**Figura 3.7 - Exemplo de utilização de Caso de Uso**

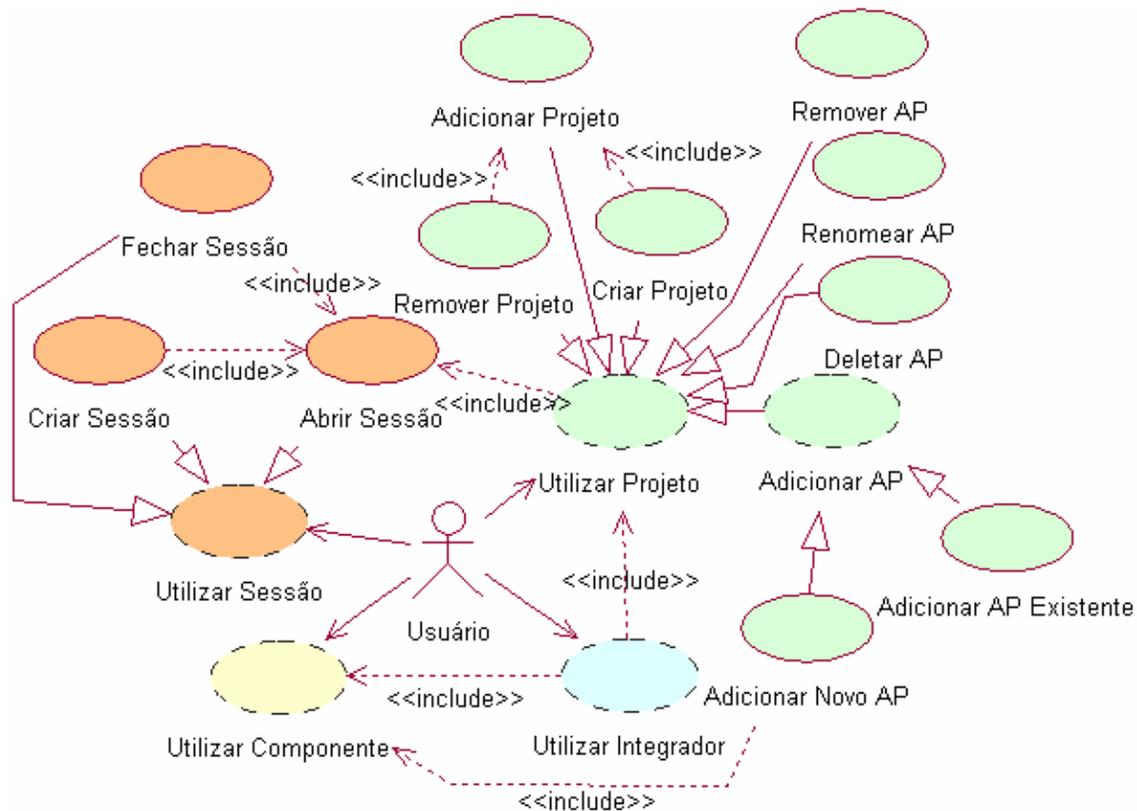
Inicialmente, as funções do sistema foram divididas em quatro partes em relação a sua atuação. Assim, as funções do sistema disponibilizam ao usuário, a utilização de: sessão, projeto, componente e integradores. Dessa forma, obtemos o diagrama da Figura 3.8.



**Figura 3.8 - Funções do Sistema divididas**

No diagrama da Figura 3.8 é mostrado que só é possível utilizar projeto dentro de uma sessão, assim como para a integração é necessário existir um projeto na sessão aberta e também é preciso utilizar componentes para ter acesso aos AP desejados.

Expandindo apenas as partes de sessão e projeto, para mostrar a inter-relação entre elas, podemos obter o diagrama exibido na Figura 3.9.



**Figura 3.9 - Funções do Sistema - Sessão e Projeto**

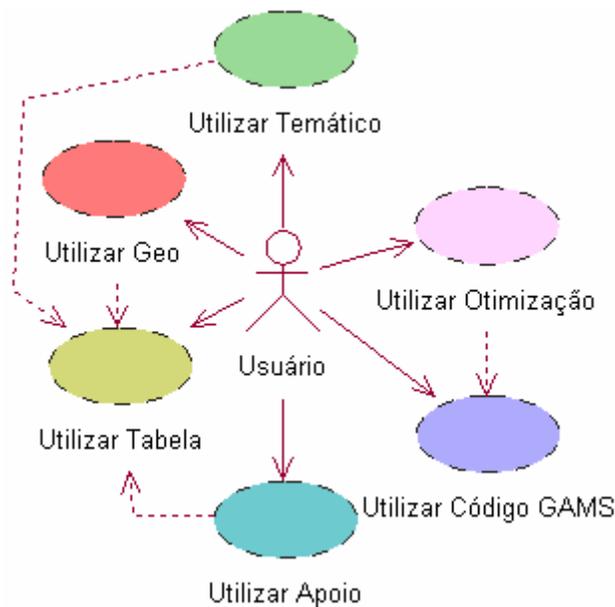
No diagrama da Figura 3.9 é mostrada a gerência dos AP considerando a hierarquia, sessão, projeto e AP. Em outras palavras, de acordo com este diagrama, o usuário pode criar uma nova sessão e ao criá-la, a mesma será aberta (relação de inclusão). Da mesma forma, ao fechar uma sessão, a mesma deve estar aberta, ou simplesmente o usuário pode abrir uma sessão, anteriormente criada, considerando que apenas uma sessão estará aberta.

Para utilizar projetos, uma sessão deve estar aberta. Neste caso ao criar um projeto, o mesmo é adicionado à sessão atual, e para removê-lo ele deve pertencer à sessão, além de poder adicionar um projeto já existente de outra sessão, possibilitando o compartilhamento de projetos entre sessões.

Além disso, numa sessão que possui um projeto, ao projeto pode ser adicionado um AP novo ou existente, renomear, remover ou apagar definitivamente um AP que pertença a algum projeto.

### 3.5.1. Funcionalidades dos Componentes

Para discriminar a utilização de cada componente do sistema, pode-se dividi-la em relação a um conjunto de cores, distinguindo cada um como na Figura 3.10, sendo as cores utilizadas nestes casos de uso são semelhantes às cores utilizadas na identificação dos componentes, mostrados anteriormente na Figura 3.2.



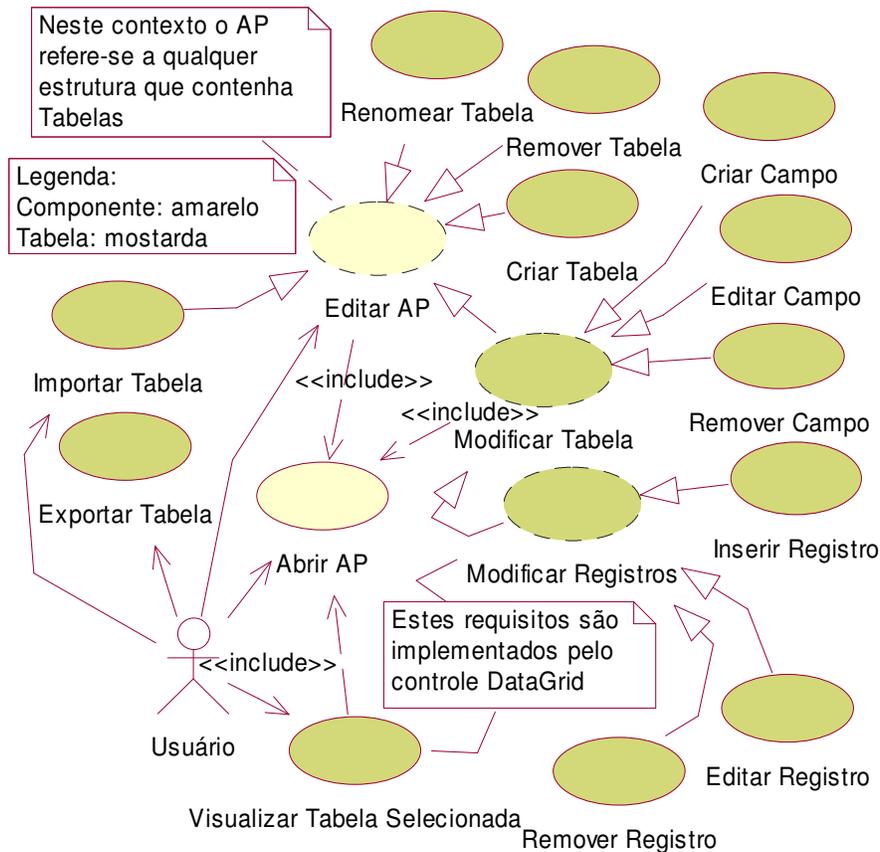
**Figura 3.10 - Legenda para os Componentes**

Inicialmente, para descrever as funcionalidades dos componentes, é necessário descrever as funcionalidades comuns a todos, que são gerenciadas de forma centralizada pelo sistema e realizadas pelos componentes. Assim, finalmente expandindo o diagrama em relação aos componentes, gerenciando o AP, obtem-se o diagrama da Figura 3.11.



### 3.5.2. Funcionalidades do Componente Tabela

Detalhando as funcionalidades relacionadas ao Componente Tabela, pode-se esboçar o seu diagrama de casos de uso, na Figura 3.12.



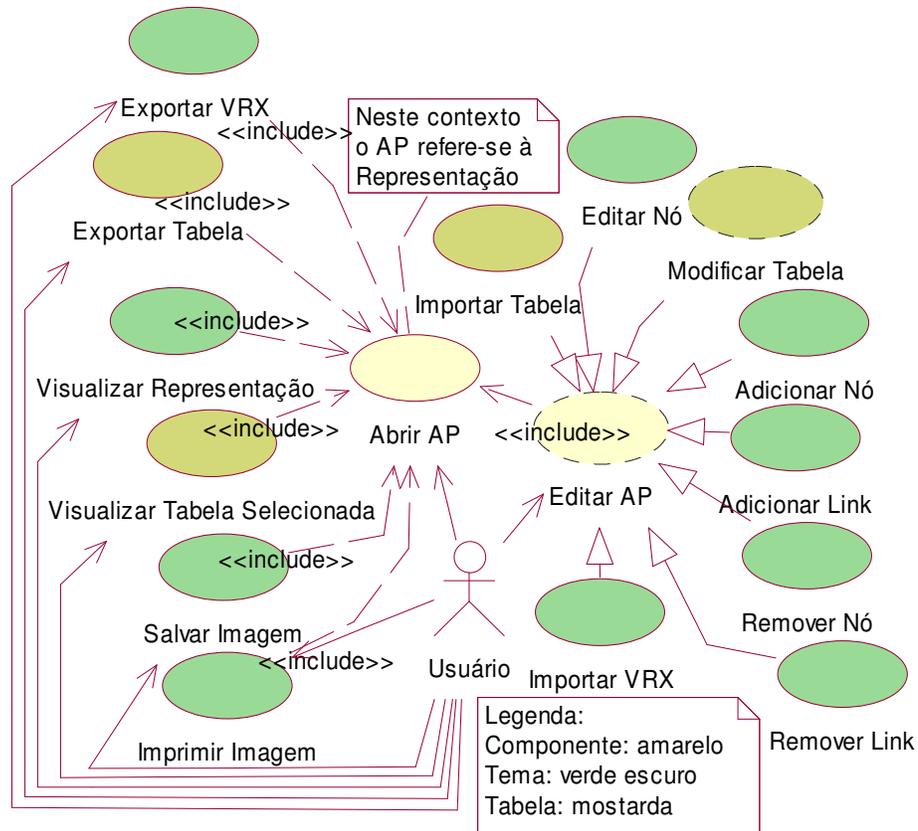
**Figura 3.12 - Funcionalidades do Componente Tabela**

Este componente é responsável pelo gerenciamento dos dados tabulares do sistema, que correspondem a arquivos de tabelas (i.e. possui um conjunto de tabelas) encontrados no formato XML. Dessa forma tabelas escolhidas podem ser importadas e exportadas a partir ou para arquivos neste mesmo formato. As tabelas podem ser criadas, renomeadas, removidas deste arquivo e modificadas, onde estas modificações podem ser em suas estrutura, modificando os campos que lhe pertencem, criando, editando seu nome e tipo do dado ou removendo-o, ou ser no seu conteúdo, modificando os registros, inserindo, editando ou removendo-os. Além disso, a tabela selecionada deve ser visualizada para que

haja a modificação no seu conteúdo, onde esta visualização e modificação são gerenciadas pelo *DataGrid*.

### 3.5.3. Funcionalidades do Componente Tema

Para mostrar os requisitos do Componente Tema, é utilizado o seu diagrama de casos de uso, na Figura 3.13.



**Figura 3.13 - Funcionalidades do Componente Tema**

Pode ser observado, na Figura 3.13, que algumas funcionalidades do componente tabela estão presentes neste diagrama devido à dependência existente entre o componente, omitindo a composição em “Modificar Tabela”, presente no diagrama anterior. Neste caso, as funcionalidades adicionais ao componente tema são relacionadas aos nós e *links*, onde existem adição e remoção, além de editar os atributos de um nó. As configurações da visualização genérica estão presentes num conjunto de tabelas definido como VRX (Visualização de Representação em XML), que podem ser exportadas e importadas a uma

representação. Obviamente a representação deve ser visualizada através da interface gráfica, existindo a possibilidade de salvá-la como arquivo em diversos formatos de imagem, bem como imprimi-la.

### **3.6. Conclusões**

Tendo definido todo o sistema, ao qual o componente tema pertence, considerando as estratégias utilizadas, sua arquitetura, seus componentes, suas operações e suas funcionalidades, torna-se mais clara a função do componente tema na aplicação.

Conhecendo a função deste componente no SADE-GBHidro, é possível detalhar a sua implementação, o que é realizado no Capítulo 4.

## 4. O Componente Tema

Como já foi dito anteriormente, o componente tema é um elemento do sistema responsável pela gerência da base de dados de temas, ou também chamada de representação, permitindo uma visualização da informação considerando os elementos e a relação entre eles, e garantindo a sua consistência. Este capítulo apresenta a motivação para criação e implementação deste componente.

### 4.1. Motivação

A motivação para criação deste componente está na criação de um elemento intermediário de integração entre a base de dados geográfica e o modelo, inserindo novas informações como ambientais e socioeconômicas. Este possui uma representação gráfica para o usuário, de forma a aumentar sua compreensão, podendo entender a discretização da bacia, as relações entre todos os elementos envolvidos e também como o problema está sendo resolvido, fornecendo mais uma ferramenta para apoio a decisão.

Devido a mudanças constantes nos requisitos em relação aos modelos, os quais possuem representações distintas, existe a necessidade de que sua visualização seja genérica. Em outras palavras, é necessário que a representação seja realizada pelo componente sem mudança no código fonte, mudando apenas algumas tabelas que configurem tal visualização.

Diante disso surgem alguns desafios:

- Como criar uma visualização de uma rede de nós e *links*?
- Como armazenar os dados e seus tipos?
- Como editar os atributos dos nós?
- Como relacionar estes nós com outras tabelas?
- Como criar uma visualização genérica?

## 4.2. Implementação do Componente Tema

O componente tema possui duas formas de edição dos dados: através da representação, que garante a consistência das informações, ou através diretamente das tabelas, gerenciada pelo componente tabela, até porque nem todos os elementos são representados, existindo também a necessidade de editá-los (e.g. tabelas econômicas de produção).

Dessa forma, o componente tabela é inserido dentro do componente tema, podendo ser utilizado, como mostrado na Figura 4.1 a seguir.



Figura 4.1 - Exemplo de uso do componente tabela dentro do componente tema

A solução para estes desafios está na criação de uma interface para representação e fazer com que a sua visualização e edição sejam realizadas de forma genérica.

### 4.2.1. Criação da Interface de Representação

Para conseguir realizar uma representação que utilize nós e *links*, foram criados três controles (*UserControl*): elemento (nós), descricao (descrição do nó) e ligacao (*links*).

O controle elemento possui uma imagem (*PictureBox*), que corresponde ao seu símbolo e um rótulo (*Label*) com sua descrição, caso o controle tenha sua descrição interna. Caso tenha uma descrição externa, o controle elemento possui um controle descricao como referência. Já o controle descricao possui, ao invés da imagem, uma caixa de texto

(*TextBox*), para poder realizar a edição de sua descrição. Na Figura 4.2, a seguir, exemplos de descrição interna e externa ao nó podem ser observados.

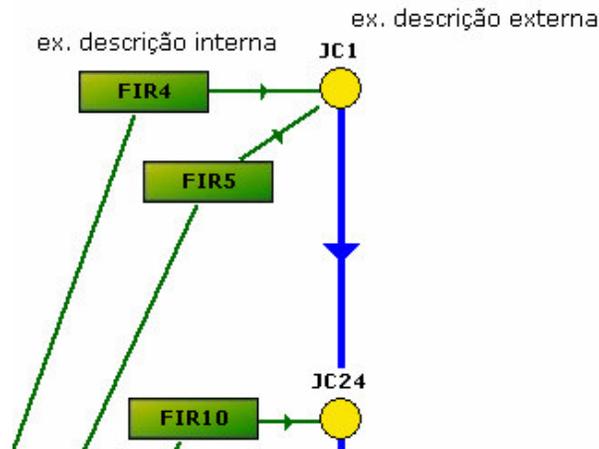


Figura 4.2 - Exemplo de descrição utilizada na representação

Estes dois controles (elemento e descricao), através do mapeamento de eventos do mouse, podem ser editados seus atributos, como mostrado na Figura 4.3, e posicionados na tela conforme a necessidade do usuário, de modo que ele defina a melhor posição, tanto do controle elemento, quanto do controle descricao, arrastando-os pela tela.

A imagem mostra uma janela de software intitulada 'Detalhes de Pirapama'. Ela possui duas abas: 'Dados' (selecionada) e 'Tabelas relacionadas'. Abaixo das abas, há uma seção com o título 'Tabela reservatorio' em fundo vermelho. Esta seção contém vários campos de entrada:

- noDespejo**: campo de seleção com o valor 'JC26'.
- ehAfluente**: caixa de seleção desativada.
- idNC**: campo de texto com o valor '15'.
- evaporacoes**: campo de seleção com o valor 'PirapamaEVAP'.
- vazoesMedias**: campo de seleção com o valor 'PirapamaVM'.
- capacidadeMaxAgua**: campo de texto com o valor '60.937'.
- capacidadeMinAgua**: campo de texto com o valor '8.4'.
- saidaMinima**: campo de texto com o valor '3.1104'.
- concen\_fosforo**: campo de texto com o valor '0.1'.

Na base da janela, há dois botões: 'OK' e 'Cancelar'.

Figura 4.3 - Exemplo de edição dos atributos do reservatório Pirapama

Para realizar a edição dos atributos dos nós, é realizado um mapeamento dos relacionamentos existentes entre todas as tabelas, bem como avaliado o tipo do dado, mesmo sendo texto e representando na verdade, um número ou booleano. Assim, é mantida a consistência em relação a isso, fazendo verificações ou fornecendo controles específicos, como no formulário da Figura 4.3, disponibilizando uma lista de opções (*ComboBox*) quando há uma relação, ou uma checagem (*CheckBox*) para booleanos. Boa parte deste esforço não é necessária, pois no sistema atualmente é possível armazenar dados de tipos diferentes do tipo de texto (*String*), através do armazenamento do seu esquema (*XMLSchema*) dentro do próprio arquivo.

Além disso, o controle elemento possui referências para os controles ligação que estão relacionados a ele. Assim, ao posicionar um controle elemento, este informa a seus *links* que ele mudou de posição.

O controle ligação possui duas referências do nó início e nó fim do link. Dessa forma este controle se ajusta à posição destes nós para ser representado.

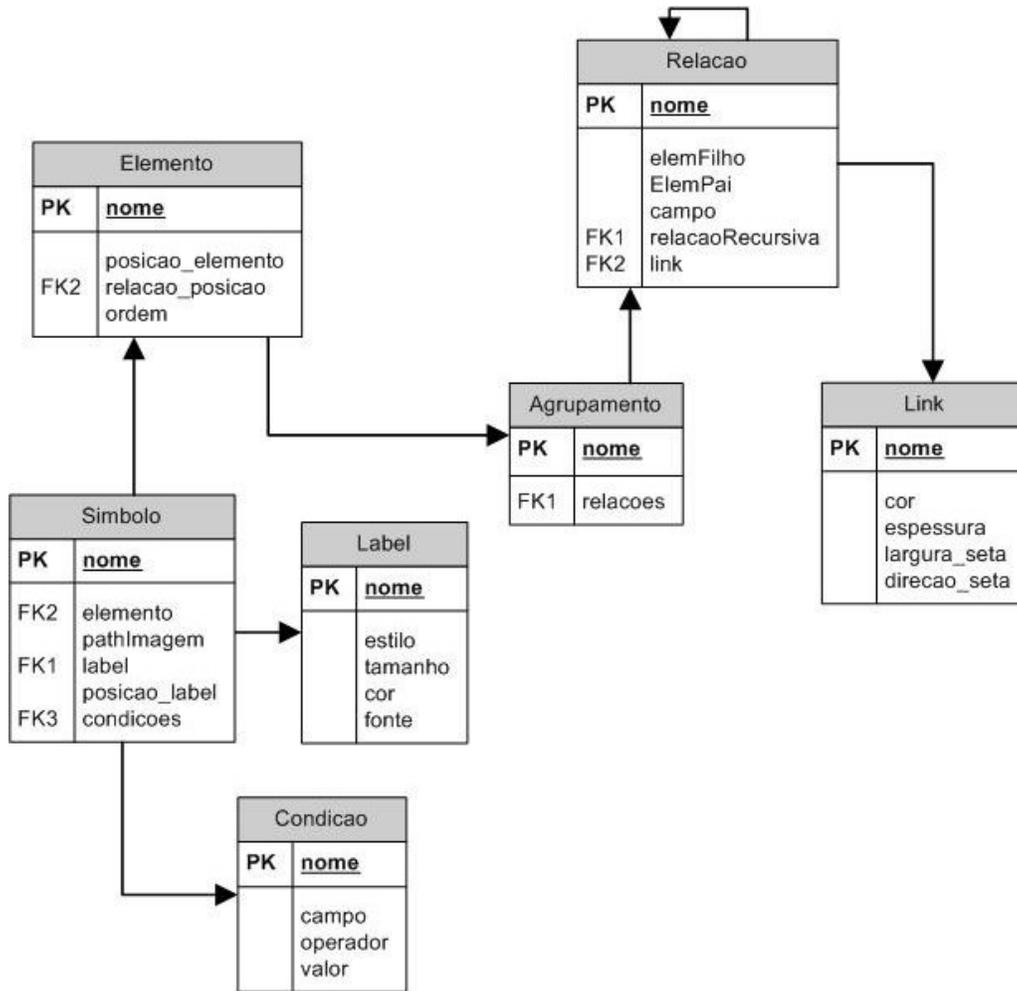
Existem duas interfaces, a *ISimbolo*, implementada pelo controle elemento e *IForm*, implementada por quem gerencia alguns comportamentos destes controles.

Para criar a representação, basta inserir estes controles relacionados entre si dentro de um painel (*Panel*), e o objeto responsável pela gerência implementar *IForm* e passar uma referência deste objeto gerenciador para os controles.

Estes controles possuem propriedades próprias como: imagem, tamanho, largura, dentre outras propriedades que são utilizadas para se adaptar melhor a outras representações.

#### 4.2.2. Criação da Visualização e Edição Genérica

Para realizar a visualização genérica, antes de tudo é necessário determinar a estrutura na qual as configurações da visualização vão estar. Inicialmente, o conjunto de tabelas utilizadas para configuração da representação pode ser mostrado a seguir na Figura 4.4.

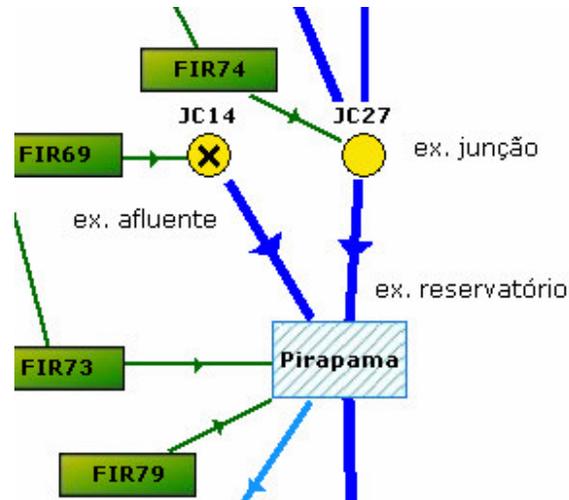


**Figura 4.4 - Estrutura básica das tabelas para configuração de visualização (VRX)**

Esta configuração consiste em realizar um mapeamento entre os nós e *links* que serão representados, correspondendo respectivamente, aos elementos e relações. Além de selecionar os nós e *links* que deverão aparecer, os nós podem ser classificados e possui símbolos diferentes, ou não serem representados a partir de determinadas condições.

Quando existem mais de uma relação, estas podem ser agrupadas gerando um nó de agrupamento que não possui representação, apenas as mesmas configurações do *link* que o determinou.

Após o mapeamento, são configurados os símbolos e as condições associadas, o aparecimento das descrições e os *links*. Na Figura 4.5, pode ser observado que a tabela segRede possui três símbolos, junção, afluente e reservatório, classificados de acordo com suas propriedades.



**Figura 4.5 - Exemplo de vários símbolos para um mesmo elemento**

Ao salvar a representação, são guardadas informações sobre o posicionamento dos elementos. Logo no início, caso não haja informações de posição dos nós, estes são reorganizados seguindo algumas propriedades de visualização, que para facilitar utiliza um *grid* virtual, no qual cada posição do *grid* só é ocupada por um único nó, para não haver sobreposição entre os mesmos. Esta reorganização pode ser realizada a qualquer momento, através da opção Visualizar >> Reorganizar. Da mesma forma que pode reorganizar, o sistema permite restaurar as últimas posições salvas pela opção Visualizar >> Restaurar.

Além disso, estas configurações constituem tabelas específicas, onde seu conjunto é denominado VRX (Visualização de Representação em XML), que pode ser importado e exportado.

### 4.3. Conclusões

No presente capítulo foi possível apresentar o componente tema, considerando a motivação de sua criação e sua implementação, no qual a criação de uma interface de representação e a visualização e edição genérica tiveram destaque.

Sendo assim, pode ser questionado em que este componente pode ser aplicado, dado também que ele está sendo desenvolvido de forma genérica. Dessa forma, no Capítulo 5, são mostradas as aplicações deste componente.

## 5. Aplicações do Componente Tema

Este capítulo apresenta aplicações do componente tema tanto na criação de diversas representações, destacando a principal representação, focada no modelo para a bacia do rio Pirapama, quanto na sua utilização em simulações, que auxiliam do planejamento e gerenciamento de bacias hidrográficas.

### 5.1. Representação de Dados

Diante da grande quantidade de informações contidas nas bases de dados e buscando um maior entendimento e envolvimento do usuário, torna-se necessário representar quais são os elementos mais importantes, assim como as suas inter-relações, focando no contexto em que está inserido. O banco de dados geográfico, utilizando apenas estes elementos principais, consegue fornecer geograficamente uma boa representação para o usuário, pois é seu foco principal. Entretanto, algumas relações de naturezas diferentes, como econômicas, hidrológicas, entre outras, não são representadas geograficamente. Dessa forma, o gerenciador da base de dados geográfica torna-se responsável por inferir as relações espaciais e realizar cálculos de acordo com estas relações, enquanto que o componente tema pode representar para o usuário qualquer tipo de relação entre qualquer entidade, através dos nós e *links*.

Para obter uma representação de dados adequada é preciso modelar a interação com os elementos e descrever como e quais elementos e relações serão representados, através das configurações de visualização (VRX).

Considerando este benefício em representar elementos e relações através de uma interface gráfica, primeiramente é mostrada a aplicação do componente na representação utilizada na bacia do Pirapama que deu origem ao componente. Em seguida, são mostrados exemplos de outras representações de dados que podem ser implementadas.

### 5.1.1. Representação da Bacia do Pirapama

A representação de dados corresponde a um conjunto de tabelas ou entidades, e relações entre elas, armazenadas em arquivos XML, denominado RDX, do qual se pode obter uma representação gráfica simplificada dos dados, ou seja, utilizando uma simbologia para representar as entidades e suas inter-relações. Com essa representação simplificada, abstraindo elementos menos importantes do mundo real, torna mais fácil para o usuário final observar as relações entre as entidades e editar seus atributos através da interface gráfica.

Para obter essa representação são utilizadas tabelas auxiliares, que podem estar armazenadas num arquivo, para determinar quais elementos e relações serão representados, pois nem tudo é visualizado, assim como sua simbologia associada, tais como imagens, cores, tamanhos, formas de representar e posicionamento. O arquivo que armazena estas configurações é denominado VRX.

Uma representação utilizada atualmente é a Rede de nós e *links*, que representa o fluxo da água e elementos do rio (rios, afluentes, reservatórios e hidroelétricas), assim como os demandantes (usuários de variados tipos de uso), sendo tudo de forma discretizada. Assim, o rio é segmentado e são determinadas entidades, nas quais é calculado o balanço hídrico, denominadas junções, que foram determinadas inicialmente por pontos encontrados na base de dados geográfica, nos quais é possível obter o aporte hídrico.

A bacia, de uma forma geral, será modelada através desta rede, na qual os nós representam entidades físicas, enquanto que os links representam conexões entre estas entidades. Utilizando a nomenclatura adotada por ROSENGRANT *et al.* [22], os nós incluídos na rede podem ser:

- Nós-Fonte (NF) - representam a oferta / disponibilidade hídrica em alguns pontos da bacia, ou seja, de onde a água pode ser retirada e o balanço hídrico pode ser calculado. Ex.: Rios, afluentes (junções), reservatórios.
- Nós-Demanda (ND) – representam os demandantes, usuários consuntivos e não consuntivos, a se modelarem através da rede. Serão sobre estes usuários que serão decididos os valores de alocação de recursos por mês. No caso, a água é o principal recurso, entretanto, existem outros utilizados no modelo, como o vinhoto oriundo das agroindústrias utilizado na fertirrigação. Ex.:

Municípios, indústrias, hidroelétricas, áreas de irrigação, área fertirrigada, entre outros.

Seguindo esta definição, e de acordo com os dados necessários para criação do modelo de otimização, foram definidas novas formas representativas, que são as entidades físicas ou econômicas definidas a seguir.

### **Entidades Físicas**

As entidades físicas são elementos da rede que possuem características físicas, como elementos reais, assim também como características hidrológicas do modelo. São elas:

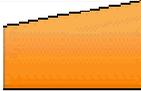
- a. ADA (Área Dinamicamente Afetada) – Corresponde às áreas nas quais o modelo utilizado, no nosso caso o modelo de otimização, realiza algum tipo de alteração em suas características com o objetivo de obter um maior benefício. No caso do Pirapama, as ADA correspondem às áreas de fertirrigação (FIRRIG) nas quais o modelo determina a melhor quantidade de vinhoto a ser utilizada.
- b. Hidroelétrica – Representam pequenas centrais hidroelétricas, assim como hidroelétricas de grande porte. O modelo determina o seu benefício, se baseando no fluxo turbinado que determina a quantidade de energia gerada.
- c. Reservatório – Representa a operação de uma barragem sobre um reservatório, considerando perdas e balanço de massa, dentre outros fatores.
- d. Demanda Consuntiva (Municípios e Agroindústria) – Representam os usuários outorgados demandantes de água. Estes usuários são classificados pelo uso da água, e assim, determina-se como deve ser calculado o seu benefício.
- e. Elemento Área – Correspondem às AC (áreas de contribuição) encontradas no BDG, nas quais o valor de área é utilizado para cálculo da vazão a partir da vazão específica, bem como para cálculo de outro tipo de contribuição relacionada, por exemplo, a carga orgânica de vinhoto no Pirapama, onde essa contribuição relacionada pode ser expandida para outros tipos.
- f. Segmento de Rede (segRede) – Corresponde ao trecho que é determinado por um Nó Fonte. Como, teoricamente, um Nó Fonte determina um único

trecho, as informações do trecho, tanto de qualquer ponto no fluxo do rio, quanto de um reservatório, devem estar contidas na tabela segRede.

- g. Elemento Vazão – Corresponde ao PIH (Ponto de Integração Hidrológica), pois é a maior resolução sobre vazão, e como uma mesma vazão específica estará relacionada a mais de um elemento área, se torna conveniente utilizar um elemento para representar esta vazão específica.
- h. Tabela CAV (cota  $\times$  área  $\times$  volume) – Possui dados de cota  $\times$  área  $\times$  volume dos reservatórios, na qual o modelo realiza a relação entre estes dados para determinação de um polinômio que a represente. A partir daí é possível determinar a área do reservatório a partir do seu volume, determinando, assim, a evaporação. A tabela CAV é inserida no BDG como uma informação adicional ao reservatório. Em alguns casos, os coeficientes do polinômio de relação podem ser fornecidos.

O Quadro 5.1 serve para mostrar as correspondências entre as entidades físicas que possuem representação gráfica no sistema.

**Quadro 5.1 - Correspondência entre entidade física e representação**

<b>Entidade Física</b>	<b>Representação</b>
ADA	
Hidroelétrica	
Reservatório	
Demanda Consuntiva (Município)	
Demanda Consuntiva (Agroindústria)	
Segmento de Rede (Junção)	
Segmento de Rede (Afluente)	

### **Entidades Econômicas**

Estas entidades correspondem às tabelas que dão suporte econômico ao modelo em questão ou que fornecem alguma especificidade em relação a quaisquer entidades encontradas. São elas:

- a. Produção – Possui dados econômicos de produção como: produtor, produto, valor produzido, quantidade de água e insumos adicionais.
- b. Produto – Possui o valor do produto para o modelo econômico.
- c. Tempo – Possui informações temporais sobre períodos específicos, como períodos secos, chuvosos, colheita, dentre outros períodos que podem ser determinados.
- d. Custo CB (consumer block – bloco de consumo) – Refere-se aos custos relacionados a determinadas parcelas de usuários, em que é realizada uma proporção para cálculo de um custo médio.

Considerando que antes de ser criada a representação genérica, a representação era fixa, na qual foi utilizada uma coleção de objetos das seguintes classes, mostrada na Figura 5.1, a seguir, para mapear os elementos da base de dados de temas em nós e *links*.

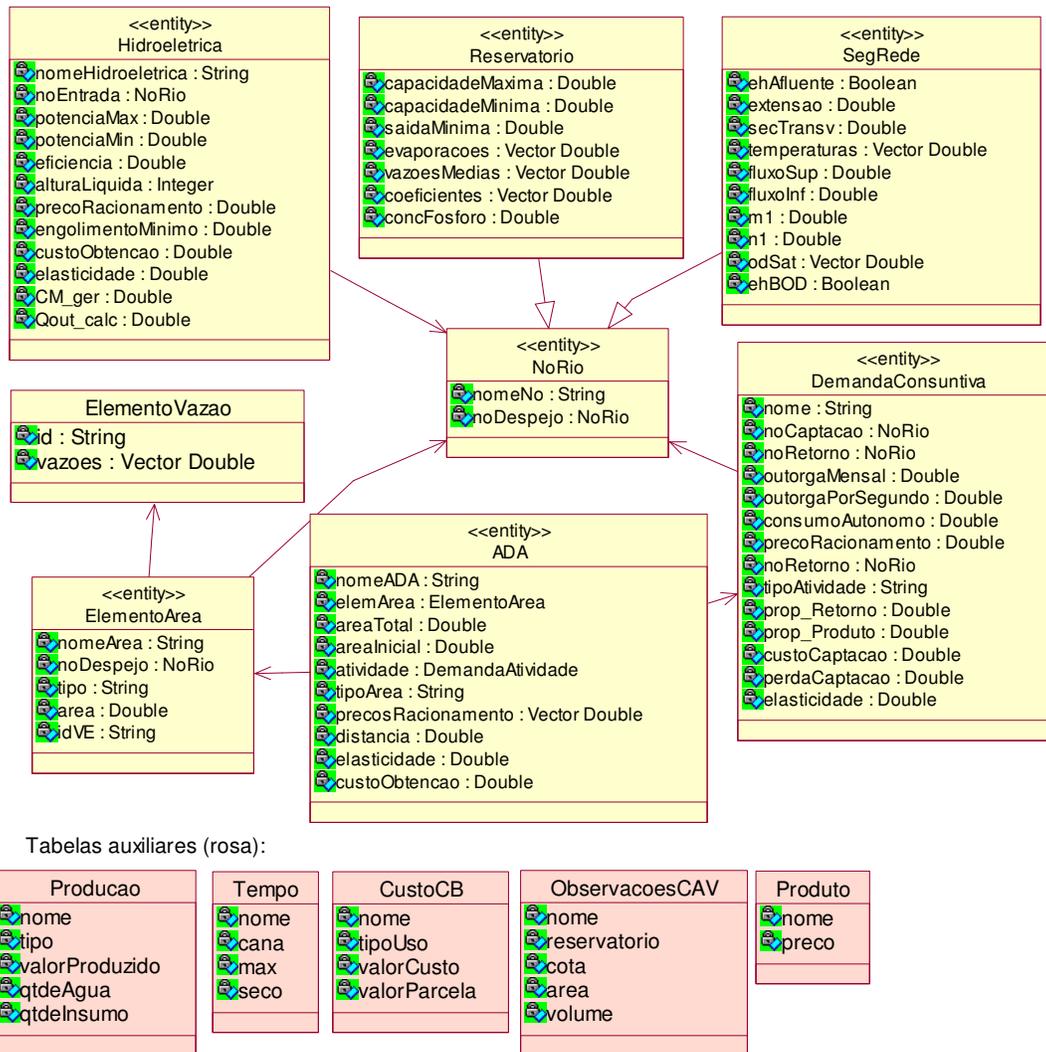


Figura 5.1 - Diagrama de Representação para o Pirapama

A representação da bacia do pirapama é apresentada nos anexos A, B e C. O anexo A consiste na representação manual realizada anteriormente, o anexo B, na representação automática fixa, utilizando as classes do diagrama da Figura 5.1, e o anexo C contém a representação genérica, utilizando as tabelas de configuração da representação.

### 5.1.2. Exemplos de Representações de Dados

Considerando que o componente tema pode ser utilizado para alguns tipos de representação de dados, consideram-se alguns exemplos como: árvore genealógica, diagrama de atividades e grafo, para demonstrar a sua aplicabilidade.

### Árvore Genealógica

A árvore genealógica da Figura 5.2 , possui *links* distintos, pai e mãe de cores diferentes, assim como possuem símbolos diferentes de acordo com o sexo, como o círculo rosa para o sexo feminino e o azul para o masculino. Outra característica desta representação é o posicionamento, onde os filhos estão sempre abaixo dos pais, pode ser observado na Figura 5.2.

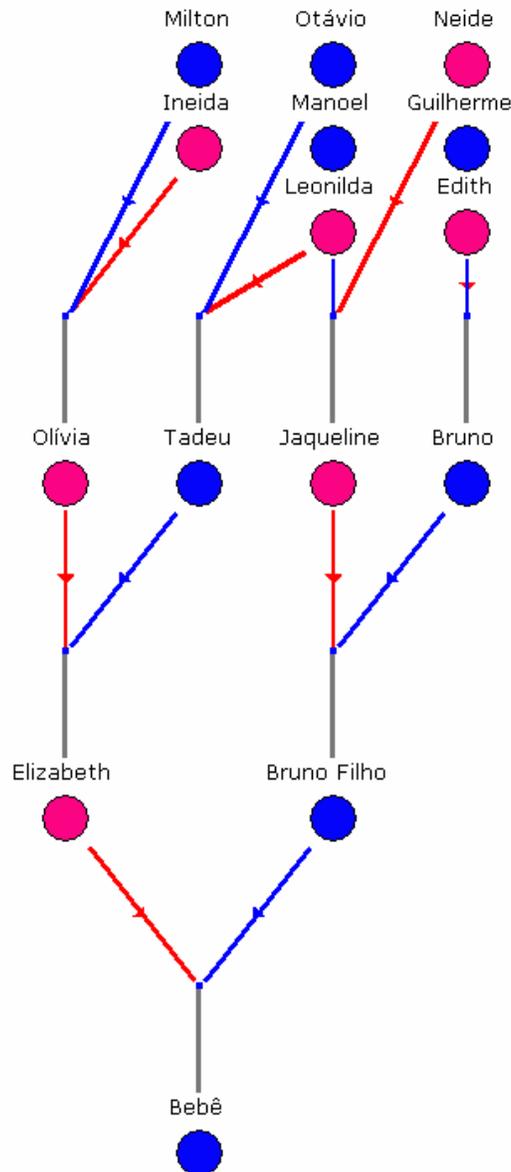


Figura 5.2 - Exemplo de árvore genealógica

### Diagrama de Fluxo

No diagrama de fluxo da Figura 5.3, são representados dois elementos: as atividades e os responsáveis. Dessa forma, as atividades são posicionadas à direita das atividades as quais ela depende, e os responsáveis estão acima das atividades que gerenciam.

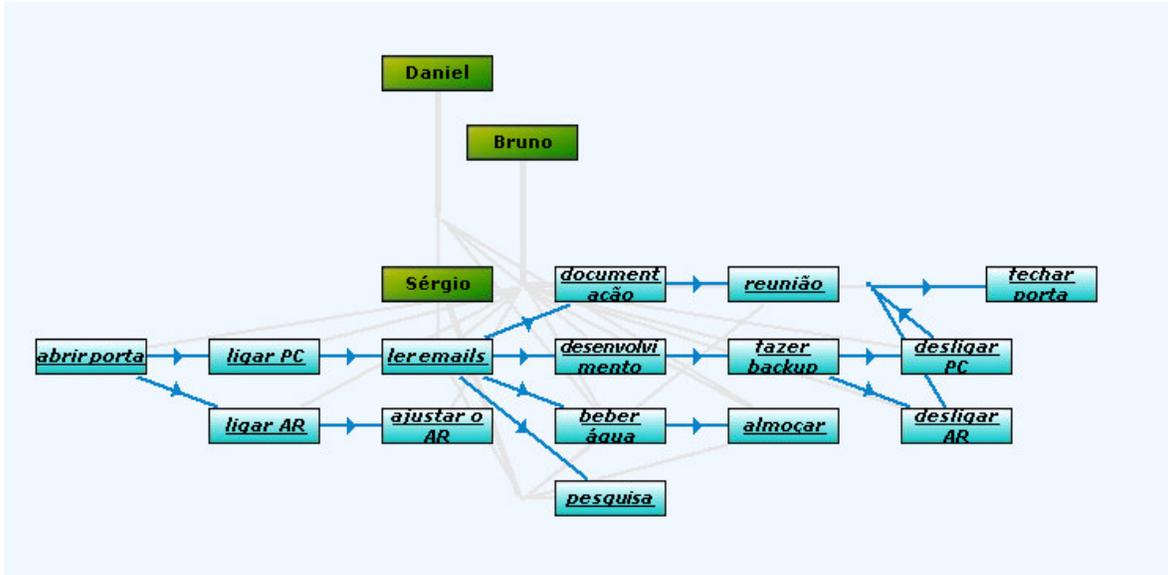


Figura 5.3 - Exemplo de diagrama de fluxo

### Grafo

No exemplo de grafo da Figura 5.4, é mostrado o comportamento aleatório dos estados, enquanto que as pessoas estão acima de quem estão relacionadas.

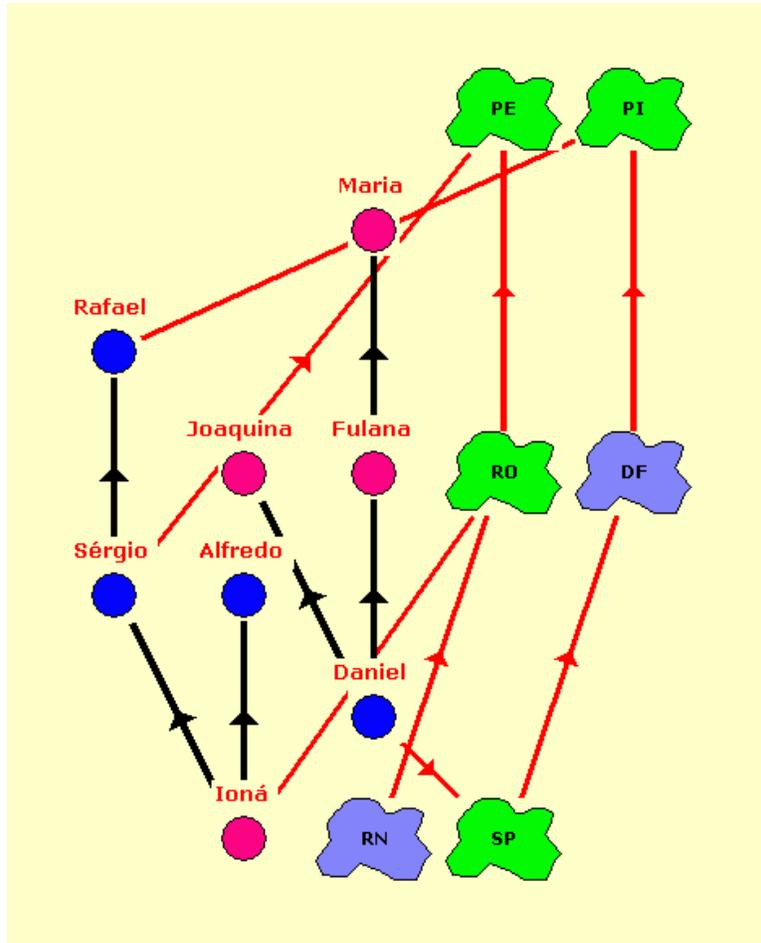


Figura 5.4 - Exemplo de grafo

## 5.2. Simulações

Auxiliando o planejamento e gerenciamento de recursos hídricos, existe a possibilidade de realizar simulações de valores e elementos, como alteração de valores de outorgas e inserção de novas usinas hidroelétricas, respectivamente.

Assim, as simulações consistem na alteração de características físicas ou não da representação (base de dados de temas), de modo que se possam avaliar alternativas, como a localização para construção de novas centrais hidroelétricas em determinadas regiões, estudar a inclusão de grandes usuários como na chegada de uma nova indústria e avaliação do potencial hídrico regional.

A utilização das simulações possui um caráter gerencial muito importante no planejamento, do qual podem ser traçadas novas metas e soluções para os problemas que possam ser encontrados.

Realizando tais alterações na representação, ainda é necessária a geração de modelos que auxiliam a tomada de decisão a partir da mesma. Sendo assim, os resultados destas alterações (simulações) podem ser avaliados através de tabelas e resultados gráficos.

Dessa forma, o componente possui a capacidade de inserir novos elementos, fornecendo para o usuário a capacidade de realizar simulações.

### **5.3. Conclusões**

Assim, apresentando a representação de uma bacia, bem como outros exemplos de representações, é possível mostrar a sua aplicabilidade em relação ao seu desenvolvimento de forma genérica, no qual o componente é configurado para cada tipo de representação.

Também, através das simulações, torna-se viável realizar um planejamento, realizando alterações na bacia, com a finalidade de gerenciá-la.

## 6. Conclusões

De acordo com o que foi apresentado, pode-se verificar que a utilização do componente tema na gestão de bacias facilita o planejamento e gerenciamento das bacias, considerando a forma de representar os dados e seu controle, havendo possivelmente um entendimento melhor pelos usuários. A sua integração com a base geográfica e o modelo, servindo de ponte entre eles, promove um entendimento maior sobre o problema e sobre o modelo a ser aplicado. Outro fato que auxilia na gestão é a facilidade com que são realizadas alterações tanto em atributos quanto estruturais.

Além disso, a utilização de uma representação genérica promove a criação de novas representações, assim como inclusão de novos modelos que possuam necessidades de representações distintas. Isto propicia a utilização deste componente para outros tipos de representações e aumenta o seu reuso, caracterizando-o como um componente.

### 6.1. Dificuldades Encontradas

Devido à existência de determinados riscos em toda atividade a ser realizada, o desenvolvimento deste sistema também foi marcado por alguns acontecimentos que influenciaram a sua execução.

Um dos riscos de mais impacto é a rotatividade dos bolsistas que também são responsáveis pela implementação do sistema, pois existe um grande esforço inicial de adaptação ao desenvolvimento do sistema, além de sua seleção. Coincidentemente, no mês de agosto deste ano, os dois únicos bolsistas deixaram o projeto, e o autor deste trabalho foi submetido a uma cirurgia de correção de um desvio de septo nasal, devido a dificuldades respiratórias.

Durante um período superior a um ano, a equipe foi reduzida, contando apenas com um bolsista. Isto promoveu uma sobrecarga de funções como: gerente, analista, suporte e desenvolvedor. Essa sobrecarga dificulta o desempenho de todos papéis da melhor maneira possível.

Conforme a grande preocupação e necessidade do cliente em visualizar os resultados da implementação do sistema, sempre existiu uma constante e rápida prototipagem e geração de versões do sistema. Considerando o tamanho da equipe, isto dificultou o desenvolvimento de uma documentação inicial apropriada, que atualmente vem sendo realizada paralelamente. Pelos mesmos motivos, parte dos testes do sistema era realizada pelos seus utilizadores, elicitando também novos requisitos.

Como o sistema está baseado em componentes, existe uma grande quantidade de referências entre eles (DLL), possuindo uma relação de dependência, o que em alguns casos, provocou problemas de versão e ocasionou comportamentos inesperados. Hoje isto está bem resolvido, pois tais referências foram reorganizadas.

## 6.2. Trabalhos Futuros

Porém, ainda existe muita coisa a ser acrescentada neste componente gerando novos trabalhos ao sistema, como adição de novos usos, a criação de novos modelos com objetivos diversificados, implementação de novos componentes que gerenciem modelos não GAMS, forçando a criação de diferentes representações.

Dentre as necessidades exclusivas do componente tema estão:

- A utilização de um algoritmo de *Path Finding* (como A\* - A estrela), para posicionamento de nós e *links*, incluindo dobras nos *links* e realizando um posicionamento próximo dos elementos que se relacionem, bem como possibilitar a utilização de algum atributo para auxiliar o posicionamento, realizando uma representação eficaz;
- Suporte a criação de representações, pois atualmente as tabelas de configuração da visualização são criadas manualmente, também utilizando o auto-relacionamento das tabelas já existente na edição dos dados para auxiliar na criação das representações de forma automática;
- Ampliação da consistência das alterações da base de temas, como a permissão de edição das tabelas relacionadas e remoção de *links*;
- Inclusão de impressão para tabelas, bem como implementar os métodos *PrintCurrent* e *PrintAll*, que futuramente serão adicionados à interface IModulo;

- Possibilitar a visualização ao mesmo tempo das tabelas junto à representação gráfica;
- Melhorar a captura da imagem da representação, que atualmente é realizada com captura das imagens da tela;
- Incluir novos tipos de retas e setas, como pontilhadas, tracejadas, seu posicionamento, assim como utilizar símbolos para os agrupamentos, pois atualmente a reta é sólida, a seta está no meio do segmento e os agrupamentos não possuem símbolos; e
- Inclusão das imagens utilizadas na representação dentro dela mesma, arquivo XML, evitando problemas sobre imagens não encontradas, fornecendo uma representação independente delas.

Agora considerando o sistema como um todo existe a necessidade de:

- Realizar integrações entre modelos e dados, possibilitando agregar resultados dos modelos às bases de dados, constituindo num fluxo inverso do qual é realizado atualmente;
- Atualização da plataforma .NET utilizada, onde se deve migrar da versão 1.1 para a nova versão 2.0, ou esperar o lançamento de uma versão ainda mais nova, evitando um trabalho a mais; e
- Suporte pelo componente tabela gerenciar outros tipos de bases de dados. Já existe um componente para gerência para bancos de dados Access, que comporá um tipo suportado.

Enfim, acredita-se também que melhor forma de conhecer novas necessidades seja através da utilização do sistema por usuários que tenham, mas não necessariamente, domínio do problema a ser tratado, fornecendo novos requisitos ao sistema.

### **6.3. Considerações Finais**

A sobrecarga de funções estimula o aprendizado em diversas áreas de atuação e o ganho de experiência em gerência de projetos, análise de sistemas, implementação, linguagem (VB.NET), o *NameSpace* .NET, dentre outros conceitos como: utilização de componentes de software em projetos; agregar conhecimentos nas áreas de recursos

hídricos, economia e otimização, por ser um projeto multidisciplinar; integração de dados; e inovação em relação à representação gráfica de dados.

No desenvolvimento deste trabalho, por já existir um sistema pronto, muitos conceitos aprendidos não foram abordados, muitos detalhes não foram mostrados, devido também ao tempo e pela quantidade informação, resolvendo-se focar apenas neste componente.

# Referências Bibliográficas

- [1] ArcGIS. Disponível em: <http://www.esri.com/software/arcgis/index.html>. Acesso em: 01 de outubro de 2006.
- [2] BRASIL, *LEI Nº 9.433, DE 8 DE JANEIRO DE 1997*. Disponível em: <http://www.ana.gov.br/Institucional/Legislacao/leis/lei9433.pdf> Acesso em 02 de julho de 2006.
- [3] BROOKE, A., KENDRICK, D., MEERAUS, A., RAMAN, R. (1998) *GAMS A USER'S GUIDE*. GAMS Development Corporation, Washington USA.
- [4] CBSE Group – Component-Based Software Engineering Group, Fukazawa Laboratory, Waseda University. Disponível em: <http://www.fuka.info.waseda.ac.jp/Project/CBSE/> Acesso em: 28 de setembro de 2006.
- [5] CLERICUZI, A. Z., ALMEIDA, A. T., COSTA, A. P. C. S. *Aspectos relevantes dos SAD nas organizações: um estudo exploratório*. Prod., São Paulo, v. 16, n. 1, 2006. Disponível em: <http://www.scielo.br/pdf/prod/v16n1/a02v16n1.pdf> Acesso em: 02 de julho de 2006.
- [6] CNARH – Cadastro Nacional de Recursos Hídricos da ANA – Agência Nacional de Águas. Disponível em: <http://cnarh.ana.gov.br/> Acesso em: 02 de julho de 2006.
- [7] CT-HIDRO – Fundo Setorial de Recursos Hídrico – *O que é o CT-HIDRO?* Disponível em: [http://www.finep.gov.br/fundos\\_setoriais/ct\\_hidro/ct\\_hidro\\_ini.asp?codFundo=3](http://www.finep.gov.br/fundos_setoriais/ct_hidro/ct_hidro_ini.asp?codFundo=3) Acesso em: 02 de julho de 2006.
- [8] FINEP – Financiadora de Estudos e Projetos. Disponível em: <http://www.finep.gov.br/> Acesso em: 02 de julho de 2006.
- [9] GAMS Home Page. Disponível em: <http://www.gams.com/> Acesso em 02 de julho de 2006.
- [10] HAINES, C. G., CARNEY, D., FOREMAN, J. *Component-Based Software Development / COTS Integration*. SEI – Software Engineering Institute.

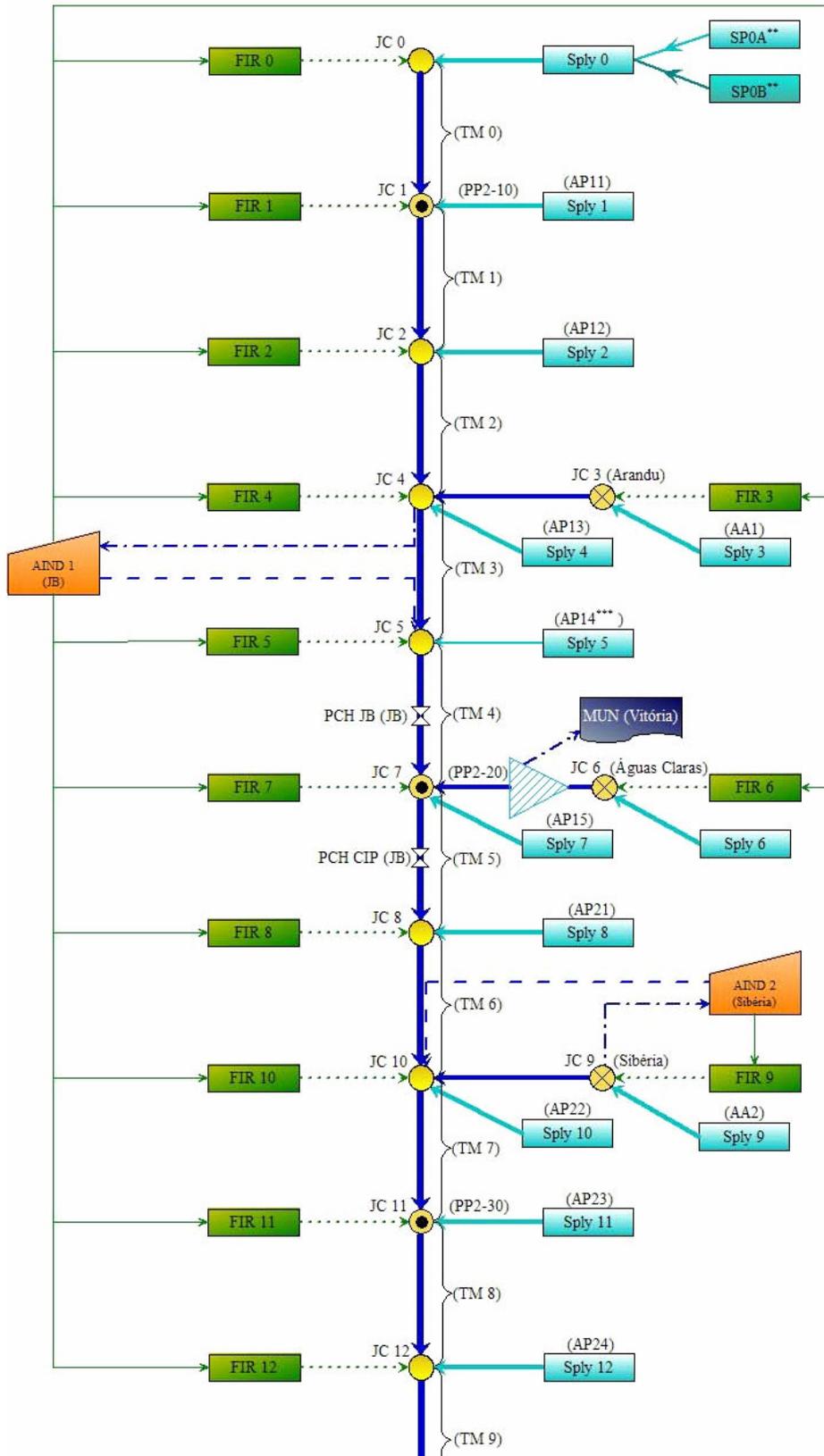
- Disponível em: <http://www.sei.cmu.edu/str/descriptions/cbsd.html> Acesso em 02 de julho de 2006.
- [11] HEINEMAN, G. T.; COUNCILL, W.T. *Component-Based Software Engineering: Putting the Pieces Together*, Addison-Wesley, 2001.
- [12] HidroWeb – Sistema de Informações Hidrológicas da ANA – Agência Nacional de Águas. Disponível em: <http://hidroweb.ana.gov.br/> Acesso em: 02 de julho de 2006.
- [13] MCKINNEY, D. C., CAI, X., MAIDMENT, D. R. *A Prototype GIS-Based Decision Support System for River Basin Management*. ESRI International User Conference, San Diego, Califórnia, 1997. Disponível em: <http://gis.esri.com/library/userconf/proc97/proc97/to200/pap164/p164.htm> Acesso em: 02 de julho de 2006.
- [14] MCKINNEY, D. C.; CAI, X.; ROSEGRANT, M.; RINGLER, C.; SCOTT, A., *Modeling water resources management at the basin level: Review and future directions*, SWIM Paper, 1999.
- [15] Microsoft Corporation. *Microsoft Windows Family Home Page*. Disponível em: <http://www.microsoft.com/windows/default.mspx>. Acesso em: 02 de outubro de 2006.
- [16] Microsoft Corporation. *UserControl Class (System.Windows.Forms)*. Disponível em: <http://msdn2.microsoft.com/en-us/library/system.windows.forms.usercontrol.aspx>. Acesso em: 02 de outubro de 2006.
- [17] Microsoft Corporation. *Visual Basic Reference*. Disponível em: <http://msdn2.microsoft.com/en-us/vbasic/ms789181.aspx>. Acesso em: 02 de outubro de 2006.
- [18] MORAES, M. M. G. A. *Modelo Econômico-Hidrológico Integrado para Alocação Ótima de Água em Diferentes Usos e Vinhoto em Áreas Plantadas de Cana na Bacia do Rio Pirapama*. Doutorado em Economia, Universidade Federal de Pernambuco, UFPE, Brasil, 2003.
- [19] MORAES, M. M. G. A.; ALBUQUERQUE FILHO, B. E. M; ROCHA, S. P. V.; SILVA, D. F. *Sistema de Apoio a Decisão Espacial para Gestão de Bacias*

- Hidrográficas Incorporando Modelagem Econômico-Hidrológica Integrada para Alocação de Água: SADE-GBHidro*. Artigo submetido e aprovado para apresentação no VIII Simpósio de Recursos Hídricos do Nordeste. Gravatá. 2006.
- [20] NETO, S. L. R. *Sistemas de Apoio à Decisão Espacial: uma contribuição à teoria em Geoprocessamento*. UDESC - Universidade do Estado de Santa Catarina. COBRAC 2004 - Florianópolis SC. Disponível em: [http://geodesia.ufsc.br/Geodesia-online/arquivo/cobrac\\_2004/096.pdf](http://geodesia.ufsc.br/Geodesia-online/arquivo/cobrac_2004/096.pdf) Acesso em: 02 de julho de 2006.
- [21] Projetos de Pesquisa CAPES, Ano Base 2003, Programa: 25001019040p-2 Engenharia Civil – UFPE. Disponível em: [http://www1.capes.gov.br/estudos/dados/2003/25001019/010/2003\\_010\\_25001019040P2\\_ProjPesq.pdf](http://www1.capes.gov.br/estudos/dados/2003/25001019/010/2003_010_25001019040P2_ProjPesq.pdf) Acesso em: 02 de julho de 2006.
- [22] ROSEGRANT, M. W.; RINGLER, C.; MCKINNEY, D.C.; CAI, X.; KELLER, A.; DONOSO, G. *Integrated Economic-Hydrologic Water Modeling at the basin scale: The Maipo River Basin*, Agricultural Economics, 2000. pp.33-46
- [23] ROSENTHAL, R. E. *A GAMS Tutorial*. Naval Postgraduate School, Monterey, California USA.
- [24] SANT'ANNA, M. *Linha de Código – A Classe DataSet e XML no ADO.NET*. Disponível em: [http://www.linhadecodigo.com.br/artigos.asp?id\\_ac=18&pag=1](http://www.linhadecodigo.com.br/artigos.asp?id_ac=18&pag=1). Acesso em: 02 de outubro de 2006.
- [25] SHIM, J.; et al. *Past, present, and future of decision support technology*. Decision Support Systems, 2002.
- [26] SPRAGUE, Jr., WATSON, H. *Decision support systems: putting theory into practice*. USA: Prentice-Hall, 1989.
- [27] TatukGIS. Disponível em: <http://www.tatukgis.com>. Acesso em: 01 de outubro de 2006.
- [28] USGS (United States Geological Survey). *Geographic Information Systems (GIS) Poster*. 2005. Disponível em: [http://erg.usgs.gov/isb/pubs/gis\\_poster/](http://erg.usgs.gov/isb/pubs/gis_poster/) Acesso em 19 de julho de 2006.

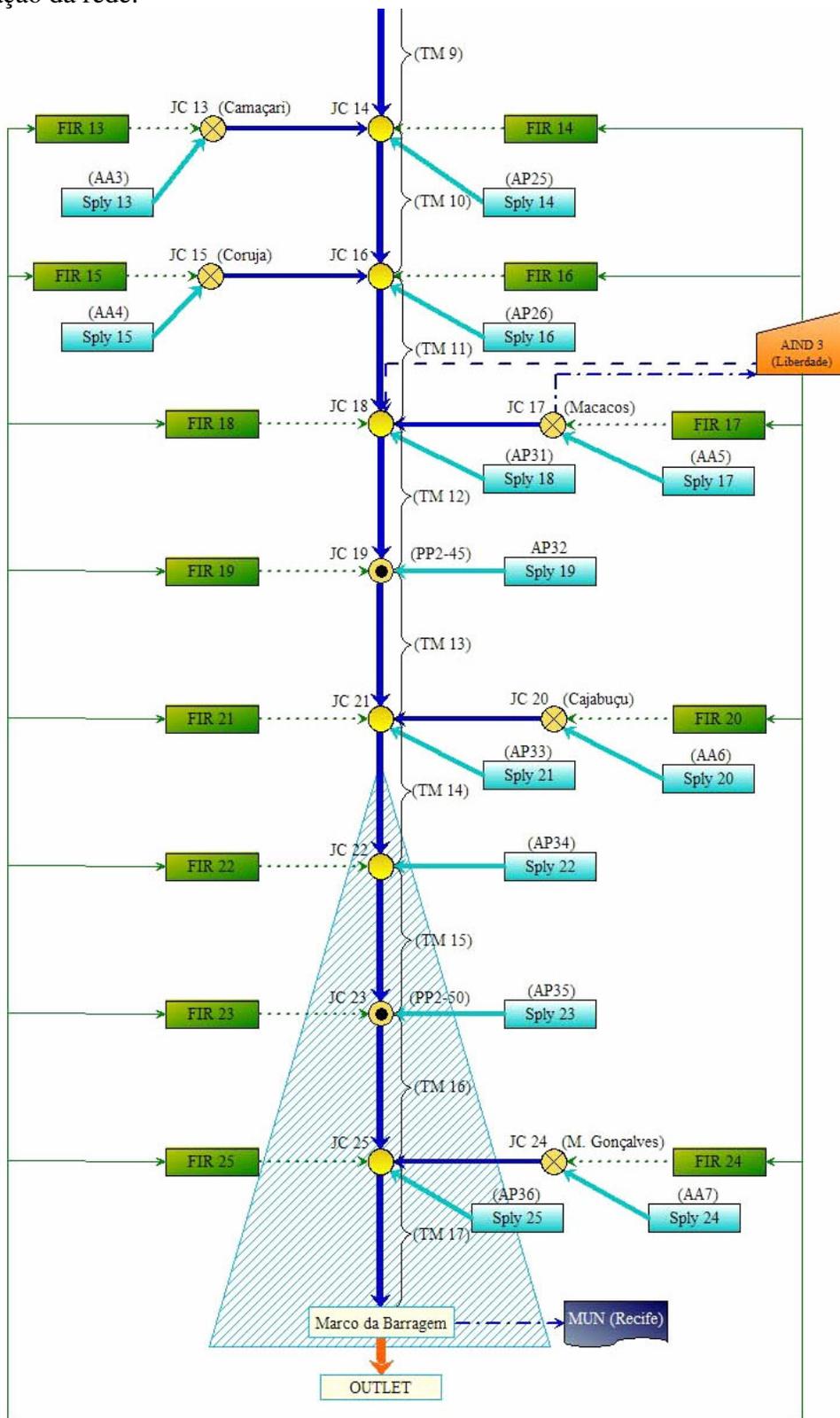
- [29] W3C. *Extensible Markup Language (XML)*. Disponível em: <http://www.w3.org/XML/>. Acesso em: 02 de outubro de 2006.
- [30] Webopedia. *What is a component software? – A World Definition From the Webopedia Computer Dictionary*. Disponível em: [http://www.webopedia.com/TERM/C/component\\_software.html](http://www.webopedia.com/TERM/C/component_software.html). Acesso em 01 de outubro de 2006.
- [31] Webopedia. *What is DLL? World Definition From the Webopedia Computer Dictionary*. Disponível em: <http://www.webopedia.com/TERM/D/DLL.html>. Acesso em: 02 de outubro de 2006.
- [32] Wikipedia. *Hidrologia*. Disponível em: <http://pt.wikipedia.org/wiki/Hidrologia>. Acesso em: 02 de julho de 2006.
- [33] Wikipedia. *Software componentry*. Disponível em: [http://en.wikipedia.org/wiki/Software\\_componentry](http://en.wikipedia.org/wiki/Software_componentry). Acesso em 29 de setembro de 2006.
- [34] YURAS, G. *2-7 Thematic Data Modeling*. Disponível em: <http://www.profc.udec.cl/~gabriel/tutoriales/giswb/vol1/cp2/cp2-7.htm>. Acesso em: 20 de setembro de 2006.

# Anexos

## Anexo A - Rede do Pirapama antes do sistema (criação manual) [18]



Continuação da rede:



Continuação da rede (legenda):

Simbolo	Descrição
	Limite dos Segmentos Modelados
	Nó Limite do segmento com PP
	Nó limite que representa afluente
	Aporte de Vazões Naturais Médias
	Área Fertirrigada
	Usina de Beneficiamento de Cana-de-açúcar
	Hidroelétrica de Pequeno Porte (PCH)
	Marco da barragem do Rio Pirapama
	Demanda Municipal
	Reservatório

Códigos	Descrição
JC	Trecho a Montante
Sply	Aporte Pluviométrico
SP A	Aporte Pluviométrico de Área sem Cana
SP B	Aporte Pluviométrico de Área com Cana
FIR	Demanda por Fertirrigação
Outlet	Trecho a Jusante
AIND	Demanda de Agroindústria
PCH	Demanda por Geração de Energia
MUN	Demanda Municipal

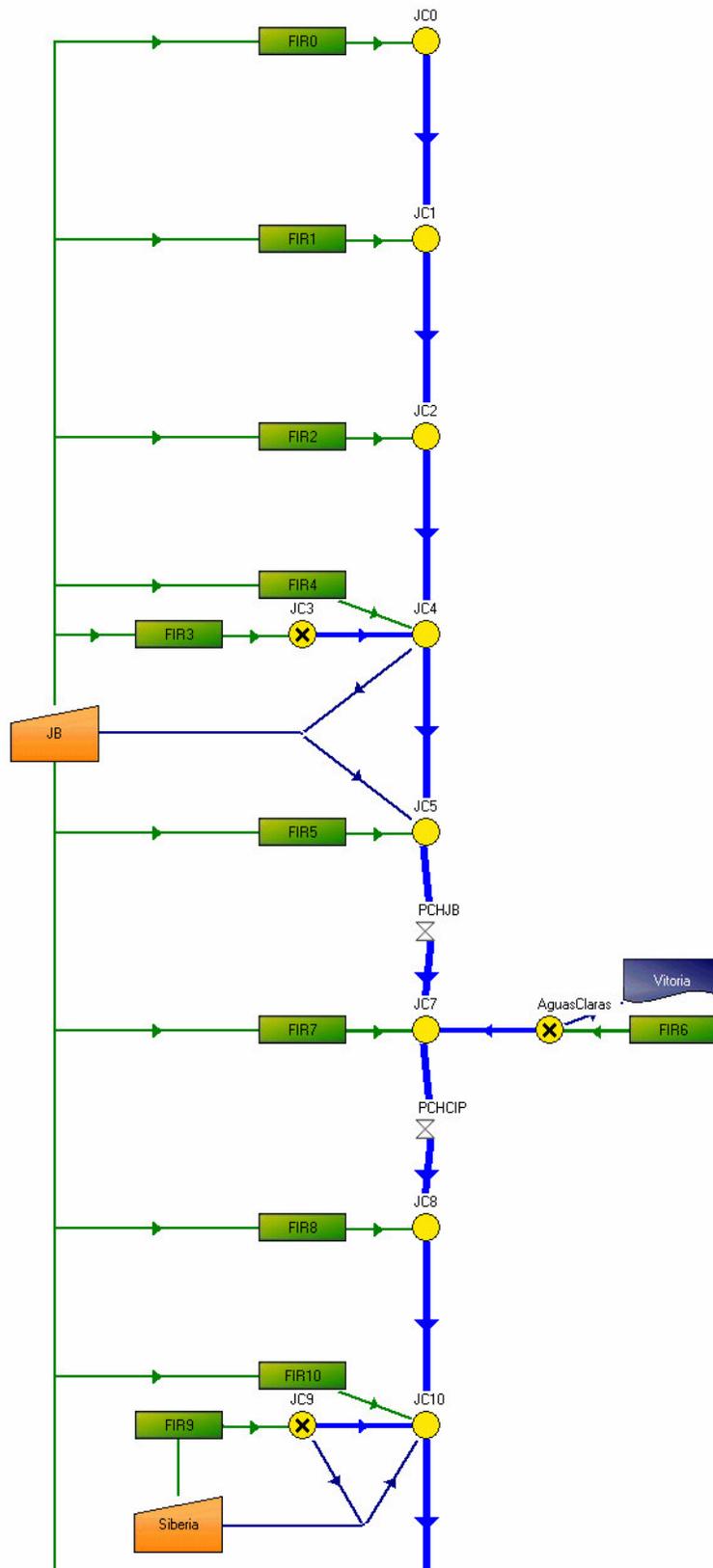
	Totais Por Trecho do Rio Principal
	Lançamentos Naturais Médios
	Lançamento de Afluentes
	Retorno de Efluente
	Captação de Água
	Retorno de Água

\* Os Símbolos em Parenteses Referem-se ao Modelo de Calibração

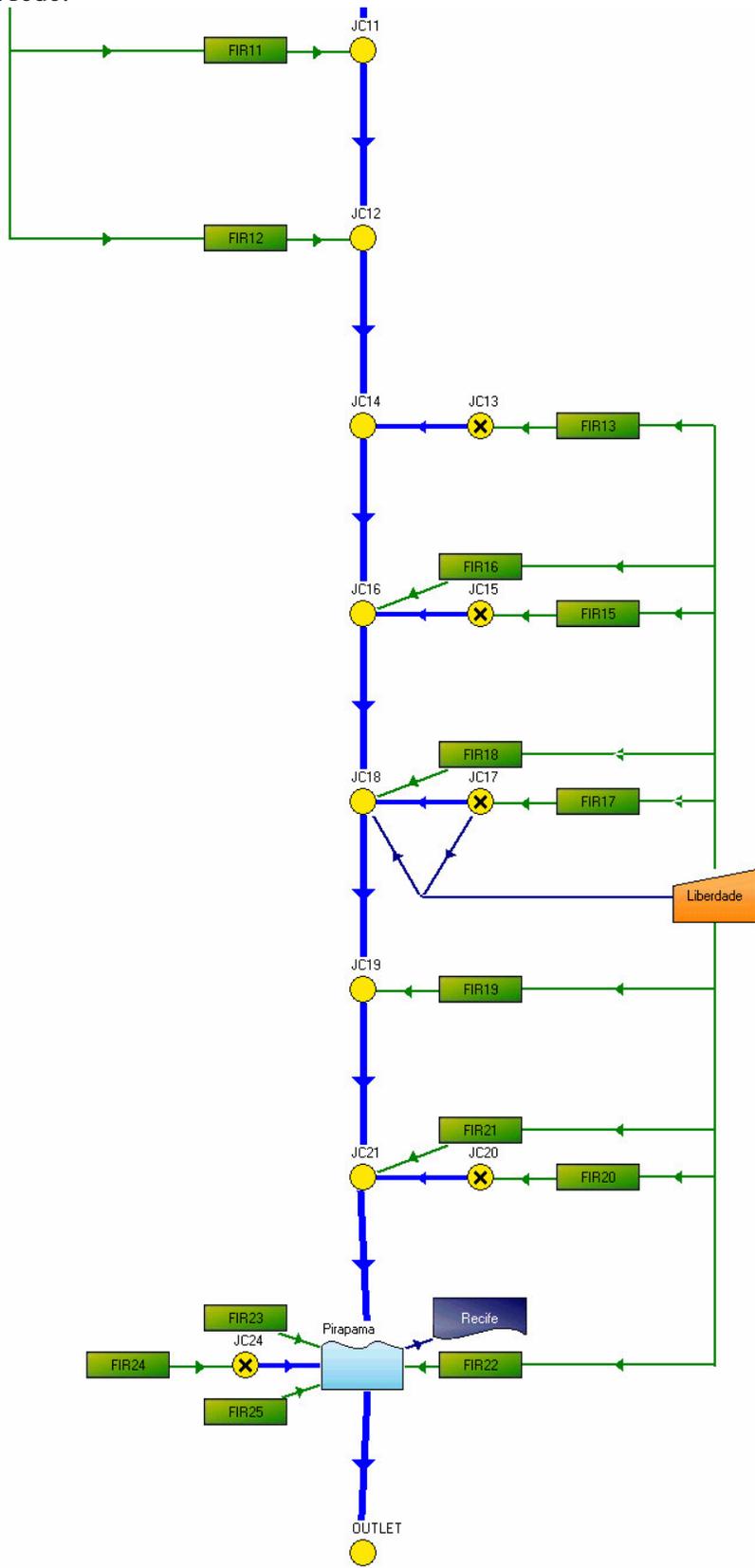
\*\* Todos os Sply's subsequentes tem como seus componentes SP\_A e SP\_B

\*\*\*O Sply 5 é associado a AP14 mas não possui o mesmo valor que o AP14 assume no modelo de calibração pois este Sply 5 não inclui Águas Claras ao contrário de AP14.

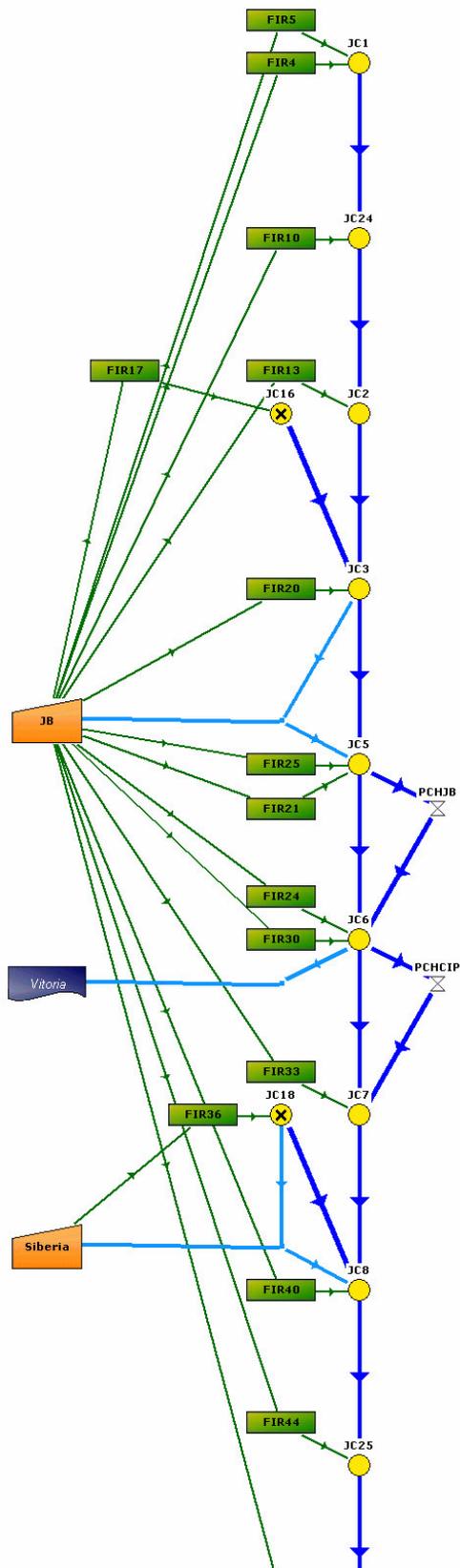
## Anexo B - Rede do Pirapama depois do sistema (classes fixas)



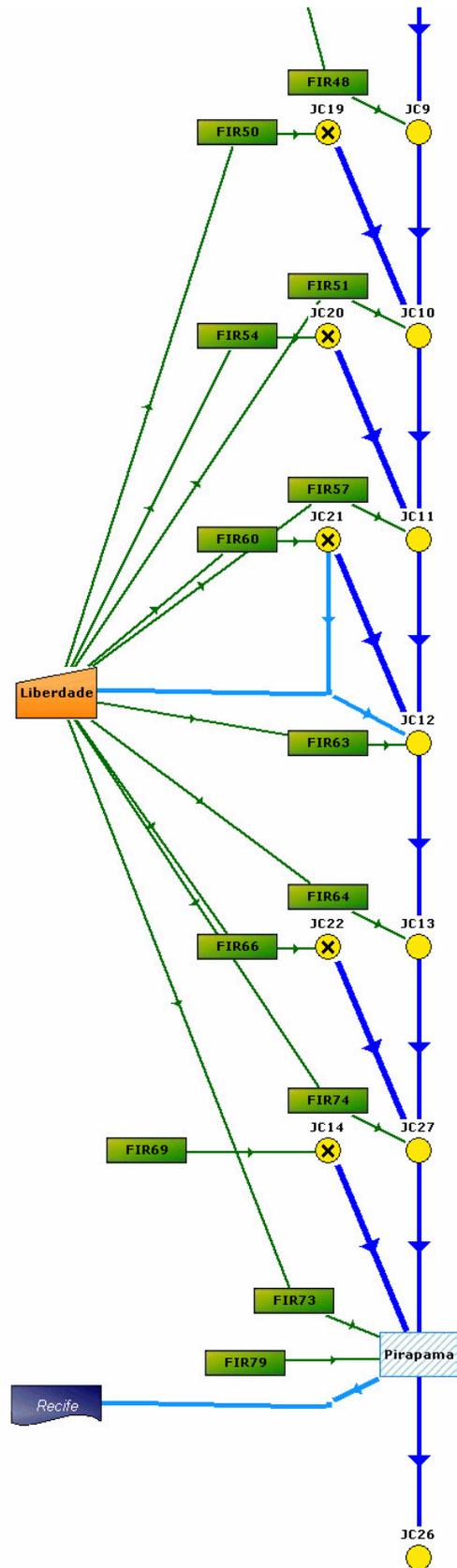
Continuação da rede:



### Anexo C - Rede do Pirapama depois do sistema (rede genérica)



Continuação da rede:



## Anexo D - Exemplo de Código Comentado GAMS

```
* Sets correspondem ao conjunto de indices utilizados, podendo haver
* links relacionando mais de um set.
Set pontos /p1, p2, p3, p4/
eixo / X, Y, Z/
variaveisX /i-0*i-39/
;
* exemplo de link relacionando sets.
set link1(pontos, eixo) /p1.X, p2.Y, p3.Y, p4.Z/

* Parameters sao valores associados aos sets por expressoes ou a
* cada indice.
parameter
* associacoes por indice:
xValor(pontos) !pontos!/p1 = -3, p2 = -1, p3 = 1, p4 = 3/
yValor(pontos) !pontos!/p1 = 3, p2 = 1, p3 = -1, p4 = -3/
zValor(pontos) !pontos!/p1 = 8, p2 = 10, p3 = 10, p4 = 8/
* associacoes por expressao:
grafLinha(pontos, eixo) !pontos eixo!
abscissa(variaveisX) !variaveisX!
cte02(variaveisX) !variaveisX!;
cte02(variaveisX) = 0.5;

grafLinha(pontos, 'X') = xValor(pontos);
grafLinha(pontos, 'Y') = yValor(pontos);
grafLinha(pontos, 'Z') = zValor(pontos);

abscissa(variaveisX) = -15;
loop(variaveisX,
  abscissa(variaveisX+1) = abscissa(variaveisX) + cte02(variaveisX);
);

* Variables correspondem aos valores associados pelo solver na resolucao
* de um problema
variable
delta
valorX
a
b
za
zb
zc
grafico(variaveisX, eixo) !variaveisX eixo!
;

* Equations consistem nas condicoes do problema que devem ser
* satisfeitas.
* declaracao:
equation
reta
eqx(variaveisX, eixo)
```

```

eqy(variaveisX, eixo)
eqz(variaveisX, eixo)
;

* corpo:
reta.. delta =e= sum(pontos,power((a*xValor(pontos) + b - yValor(pontos))
, 2))+ sum(pontos,power((za *power(xValor(pontos),2) + zb*xValor(pontos)
+ zc - zValor(pontos)) , 2));
eqx(variaveisX, eixo).. grafico(variaveisX, 'X') =e=
abscissa(variaveisX);
eqy(variaveisX, eixo).. grafico(variaveisX, 'Y') =e=
a*abscissa(variaveisX)+b;
eqz(variaveisX, eixo).. grafico(variaveisX, 'Z') =e= za
*power(abscissa(variaveisX),2)+zb*abscissa(variaveisX)+zc;

* Modelo a ser rodado
model RetaXY /all/
* opcoes do modelo:
option nlp = CONOPT3;
option sysout = on;
* comando solve:
solve RetaXY using NLP minimizing delta;

* Declaracao das variaveis utilizadas na saida GDX (tabelas)
$GDXOUT RetaXY.gdx
$UNLOAD pontos xValor yValor a b delta reta grafLinha za zb zc grafico
variaveisX
execute_unload 'RetaXY',
pontos xValor yValor a b delta reta grafLinha za zb zc grafico variaveisX

```

## **Anexo E - Breve Manual do Componente Tema do SADE-GBHidro**

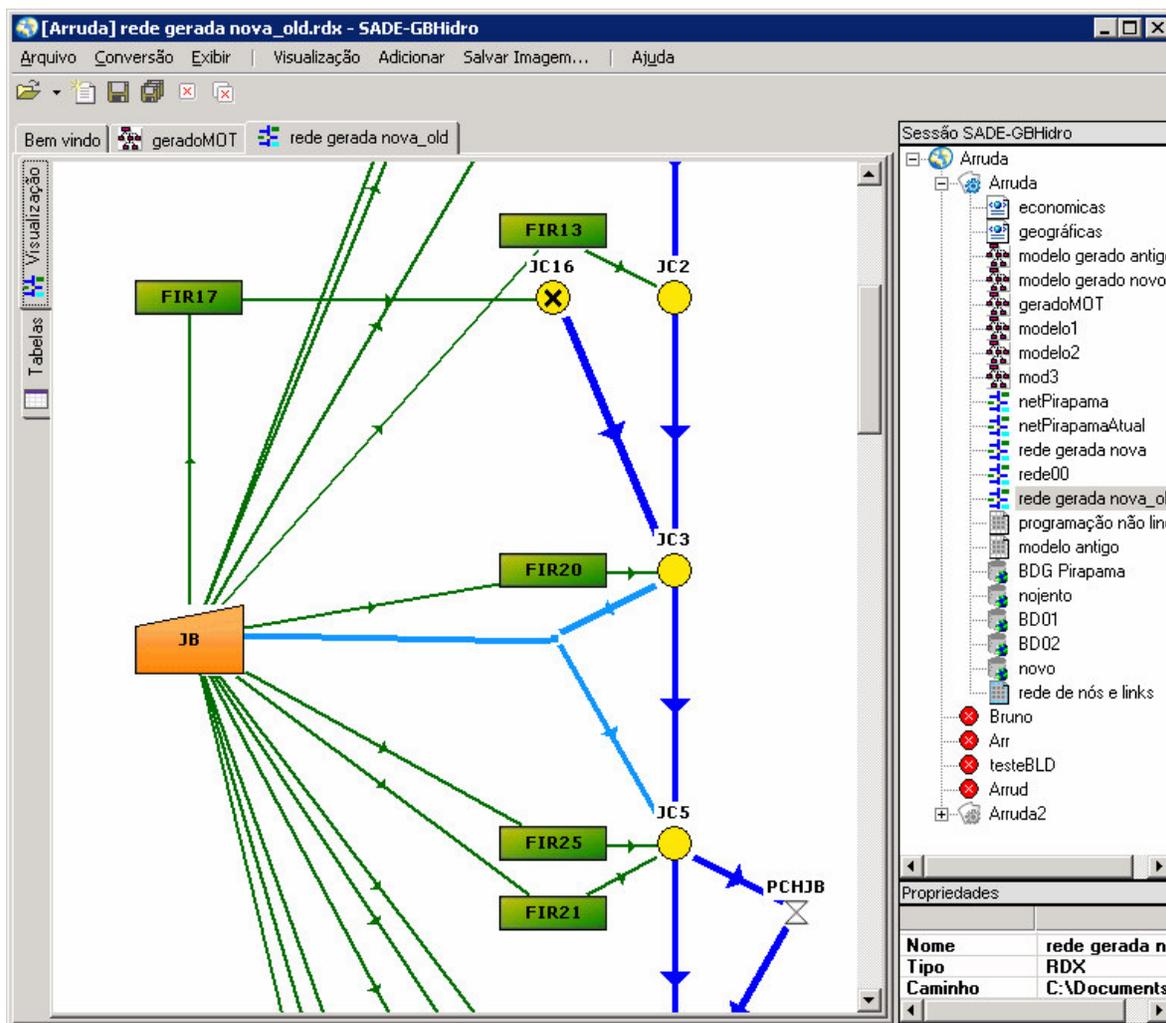
*Breve Manual do Componente  
Tema do SADE-GBHidro*

*Projeto SADE-SF*

*Versão 1.0*

## 1. Componente Tema

O componente tema é de extrema importância para o usuário, pois dá suporte a uma forma eficaz e intuitiva de visualização de dados do tema. Um exemplo disso é a visualização da bacia através de uma rede de nós e links, bem ilustrada e facilmente manipulável pelo usuário. A interface do componente tema permite dois tipos de visualização: a parte gráfica da rede representativa e as tabelas de dados que montam a rede.



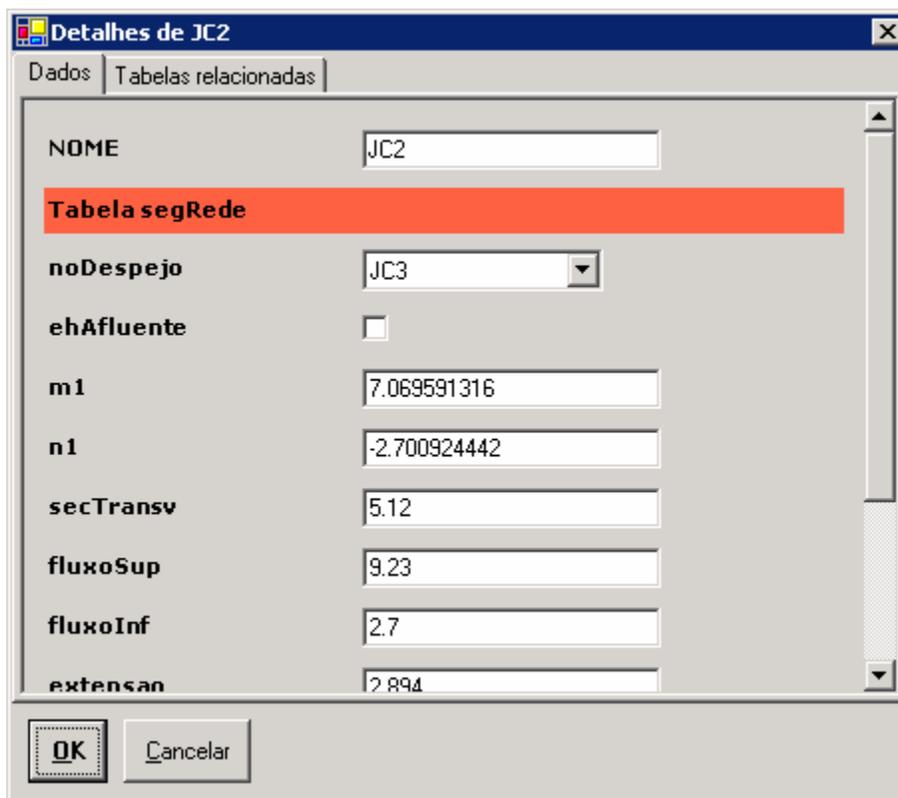
Aba Visualização do componente tema.

Como pode ser observado, o componente tema não exibe nenhum item adicional na barra de ferramentas, porém, no menu, existem funções específicas para manipulação de

seus arquivos de projeto. Analisando a barra de menu do componente temos que as opções são Visualização, Adicionar e Salvar Imagem.

### Propriedades da Interface

Conforme visualizado acima, a interface do componente tema tem como objetivo ser intuitiva e fácil de manipular. Os objetos contidos na tela, que representam os elementos geográficos, foram desenvolvidos para propiciar uma maior interação com o usuário. Ao clicar em cada nó ou símbolo, uma janela de edição é aberta, contendo informações sobre os atributos do elemento e a possibilidade de atualização de algum dado. Na figura abaixo, segue um exemplo. Vamos supor que, tendo por base a figura anterior, o usuário clicou duas vezes no nó JC2, aparecendo como resultado esta janela de edição referente a ele. Pode-se observar a existência de 2 abas: Dados e Tabelas Relacionadas.



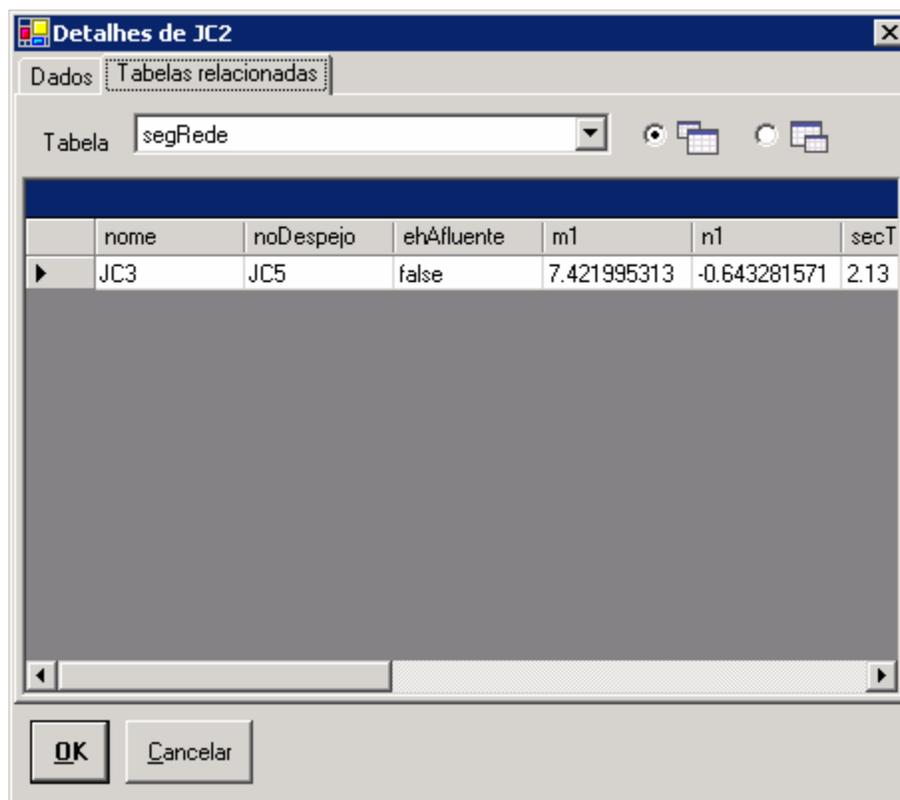
The image shows a dialog box titled "Detalhes de JC2" with two tabs: "Dados" and "Tabelas relacionadas". The "Dados" tab is active, displaying a list of attributes and their values. The attributes are: NOME (JC2), Tabela segRede (highlighted in red), noDespejo (JC3), ehAfluente (checkbox), m1 (7.069591316), n1 (-2.700924442), secTransv (5.12), fluxoSup (9.23), fluxoInf (2.7), and extensao (2.894). At the bottom, there are "OK" and "Cancelar" buttons.

Atributo	Valor
NOME	JC2
Tabela segRede	
noDespejo	JC3
ehAfluente	<input type="checkbox"/>
m1	7.069591316
n1	-2.700924442
secTransv	5.12
fluxoSup	9.23
fluxoInf	2.7
extensao	2.894

Janela de edição do nó JC2.

Na aba de Dados, mostrada na figura anterior, é possível observar todos os campos referentes ao nó, juntamente com seus atributos. Já na aba Tabelas Relacionadas, são visualizadas as tabelas que os nós fazem referência e as que fazem referência ao nó. Elas são separadas por uma opção que existe ao lado do nome da tabela, na parte superior da janela. A opção mais a direita só mostra as tabelas que a atual tabela faz referencia. No caso exemplo, na tabela atual, JC2 faz referência a JC3.

Se mudarmos a opção, a tabela que apareceria seria a do JC24, pois ela faz referência a JC2.



Aba Tabelas Relacionadas, do nó JC2.

### Visualização >> Restaurar / Reorganizar



Opções de visualização, na barra de menus.

O comando Visualização >> Restaurar tem a função de voltar ao estado inicial de disposição da rede de nós e links. Isso é importante caso o usuário faça alguma modificação indesejada e decida voltar à configuração inicial.

Já o Visualização >> Reorganizar posiciona novamente os elementos na tela, para que fiquem dispostos da melhor maneira possível.

### **Visualização >> Importar (VRX)**

Essa opção permite ao usuário inserir em seu projeto arquivos VRX já criados. Conforme explicado anteriormente, o arquivo VRX serve para estruturar o modelo de temas a ser criado, criando a configuração da representação.

### **Visualização >> Exportar (VRX)**

Já a opção Exportar, é responsável por salvar um arquivo no formato VRX, que o usuário utilizou ou criou em sua representação.

### **Adicionar >> Elementos**

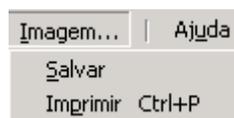
A opção Adicionar insere na rede algum elemento da escolha do usuário.



**Opção Adicionar, com os elementos disponíveis no arquivo.**

No caso da figura acima, os elementos passíveis de adição são os listados. Porém, cada projeto possui seus próprios elementos. Eles são criados na tabela relativa, que dá suporte à sua representação.

### **Imagem >> Salvar / Imprimir**



**Opções do imagem.**

Na opção de Salvar Imagem, o sistema permite que o usuário salve seu gráfico da rede num formato de figura. Dentre as opções, existem os formatos: .BMP (Bitmap), .PNG (Portable Document Graphics), .JPEG ( Joint Photographic Expert Group), .EMF (Enhanced Metafiles), .GIF (Graphics Interchange Format), .TIFF(Tagged Image File Format) e .WMF(Windows Metafile Format).

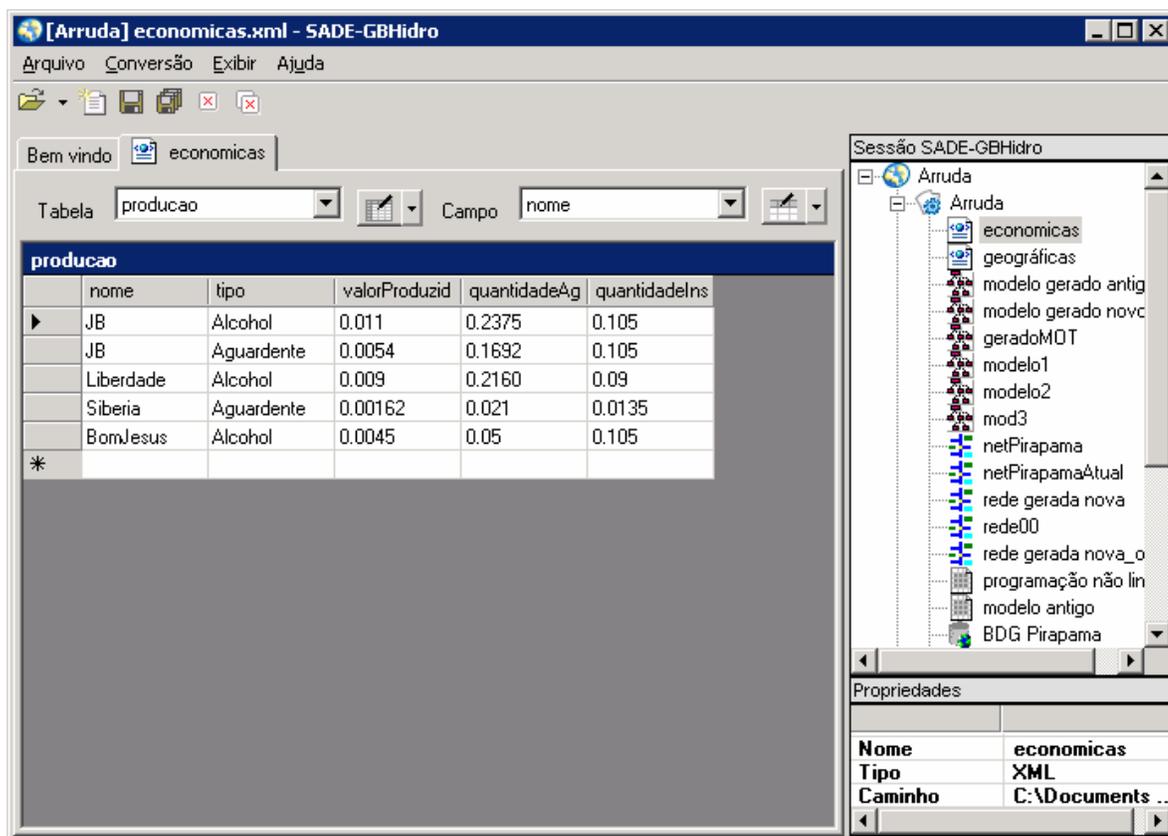
Esses formatos são todos amplamente difundidos na Internet e, certamente, o usuário deve conhecer pelo menos um. As variações entre eles são no tamanho da imagem, resoluções, dentre outras propriedades. Fica a cargo do usuário escolher o seu favorito.

Já na opção Imagem >> Impressão, o gráfico da rede será enviado para impressão.

Como pode ser observada na figura inicial, a interface do componente possui 2 abas: Visualização e Tabela. Acima foram explicadas as funcionalidades referentes à aba Visualização. A aba Tabela será explicada no componente Tabela, na próxima seção deste documento.

## 2. Componente Tabela

O componente Tabela tem a função de manipular os arquivos XML do sistema. Ele provê meios para facilitar a manipulação dos dados por parte do usuário. A idéia de tabela é extremamente útil no sistema, pois é uma forma simples e eficiente de organizar os dados necessários para representar diversas partes do programa.



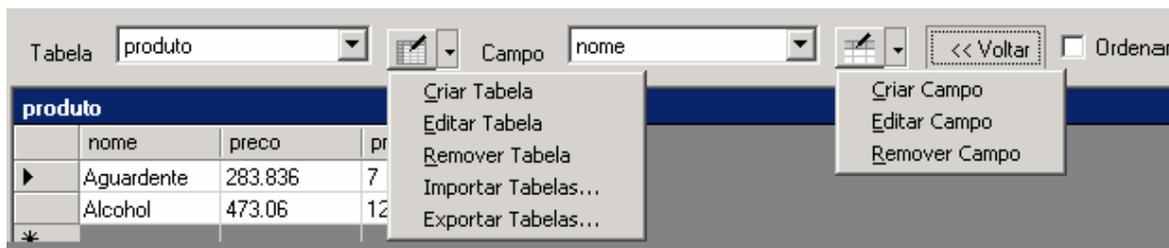
**Interface do componente Tabela.**

Na figura acima, pode-se perceber alguns elementos que tornam as tabelas bem manipuláveis. Percebe-se ainda que o componente Tabela não implica em novos botões na barra de ferramentas e nem em novos itens na barra de menu.

No canto superior esquerdo existe uma caixa para escolha da tabela a ser trabalhada. No exemplo, a tabela é Produção. Do lado do nome da tabela existe um botão que, quando clicado, exhibe as funções existentes para a tabela.

Mais a direita tem a caixa com o campo da tabela desejado e, do lado, um botão que exhibe as funções que podem ser efetuadas para o determinado campo. A figura abaixo ilustra essas funções. É importante informar que algumas dessas funções possuem atalhos

no teclado quando o foco está da *ComboBox* específica, ou de tabelas ou campos. Nas que possuem, o atalho será encontrado em sua definição entre parênteses.



**Edição da tabela e seus campos.**

### **Criar Tabela (Insert)**

Começando pelas funcionalidades referentes à tabela, temos o Criar Tabela. Ao clicar nessa opção, o usuário informa o nome da nova tabela e ela será criada no arquivo de projeto aberto atualmente. É importante lembrar que a tabela criada não possui nenhum campo e, conseqüentemente, nenhum atributo.



**Janela do Criar Tabela.**

### **Editar Tabela (F2)**

No Editar Tabela, o usuário tem a opção de renomear a tabela que ele criou.



**Janela do Editar Tabela.**

### **Remover Tabela (Delete)**

O Remover Tabela exclui do sistema a tabela desejada. O usuário escolhe a tabela e clica no Remover.

### **Importar Tabelas**

Ao clicar no Importar Tabela, o usuário pode adicionar tabelas já existentes a seu arquivo. O sistema pode informar tabelas do tipo XML e de Apoio. O usuário apenas

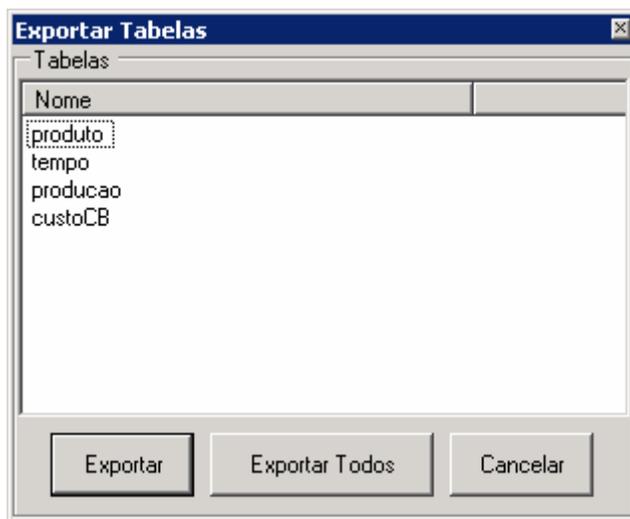
informa o caminho que elas estão e o programa as insere no componente. O usuário tem a opção de importar todas as tabelas ou algumas.



**Interface do Importar Tabelas.**

### **Exportar Tabelas**

Nesta opção, o usuário pode salvar as tabelas criadas em formato XML ou tabelas de Apoio. Ao clicar na opção, uma janela é aberta pedindo o caminho onde a tabela será salva e seu respectivo formato. A janela segue abaixo:

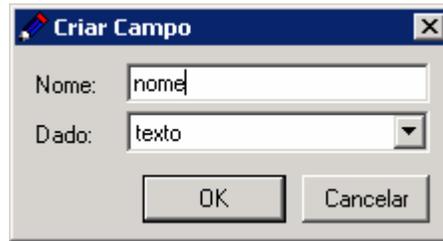


**Exportar Tabelas.**

### **Criar Campo (Insert)**

O Criar Campo adiciona campos às tabelas. O usuário deve informar o nome do campo e seu tipo dentre os oferecidos, que são: texto, inteiro, decimal, booleano e objeto.

Na explicação do Criar Tabela, adicionamos a tabela Teste. Vamos, aqui, adicionar um campo nome, do tipo texto.



**Criando um Campo do tipo texto.**

### **Editar Campo (F2)**

O Editar Campo serve para o usuário renomear o campo desejado e modificar seu tipo.



**Mudando o campo nome.**

### **Remover Campo (Delete)**

Ao clicar no Remover Campo, o sistema pergunta ao usuário se ele deseja remover o campo em questão e, caso afirmativo, exclui o campo da respectiva tabela.

## Anexo F - Interfaces utilizadas relacionadas ao Componente Tema

### Interface IModulo

Interface responsável pelo gerenciamento central dos componentes do sistema que possuem interface gráfica, fornecendo uma padronização para o gerenciamento dos AP.

```
Public Interface IModulo

    Property NivelAcesso() As IMNivelAcesso

    ReadOnly Property ToolBar() As ToolBar
    ReadOnly Property MenuItens() As MenuItem.MenuItemCollection
    ReadOnly Property ImageList() As ImageList
    ReadOnly Property Label() As String
    Property NomeArquivo() As String
    ReadOnly Property isChanged() As Boolean

    Event Mudou() 'Chamado quando algum MENU é alterado.
    Event WaitForMe() 'Chamado quando algum batch irá acontecer
    Event IamBack() 'Chamado quando o batch foi concluído
    Event Exception(ByVal ex As Exception) 'Chamado quando ocorreu alguma
    exceção
    Event Alterado() 'Chamado quando alguma alteração no arquivo foi
    realizada. Associado com isChanged

    Sub abrir()
    Sub abrir(ByVal strPath As String)
    Sub novo()
    Sub salvar()
    Sub salvarComo(Optional ByVal path As String = "")
    Sub descarregar()

    Function fechar() As Boolean

    Enum IMNivelAcesso
        Nenhum = 0
        ACQUA = 1
        Intermediario = 2
        Desenvolvedor = 3
    End Enum

End Interface
```

### Interface ITables

Interface responsável pelo gerenciamento em integrações do sistema, fornecendo o acesso às tabelas do componente que a implementa.

```
Public Interface ITables

    Property TabelaRestricao() As DataTable
```

```

Function getTable(ByVal nome As String) As DataTable
Function getGeoTable(ByVal nome As String) As DataTable
Function getTableNames() As String()

Sub setTable(ByVal table As DataTable, Optional ByVal geoTable As
DataTable = Nothing)

Sub setSelected(ByVal tabela As String, ByVal identificador As
String, ByVal arrIds As ArrayList)

Function getSelected(ByVal tabela As String, ByVal identificador As
String) As ArrayList

End Interface

```

### Interface IForm

Interface implementada pelo componente tema, fornecendo aos objetos de representação o acesso ao componente.

```

Public Interface IForm

Sub mostrarDetalhes(ByVal obj As ISimbolo)
Sub remover(ByVal obj As ISimbolo)
Function atualizarNome(ByVal nomeTabela As String, ByVal nomeAntigo
As String, ByVal nomeNovo As String) As Boolean
ReadOnly Property path() As String

End Interface

```

### Interface ISimbolo

Interface implementada pelos objetos de representação, fornecendo ao componente tema, o acesso a eles.

```

Public Interface ISimbolo

Property Descricao() As RedeControl.descricao
Property PathImagem() As String
Property NomeTabela() As String

Sub setLigacao(ByRef lig As RedeControl.ligacao)

Function estahEmCima(ByVal outroPonto As ISimbolo) As Boolean
Function estahDireita(ByVal outroPonto As ISimbolo) As Boolean
Function getPontoBorda(ByVal outroPonto As ISimbolo) As Point
Function getCentro() As Point

End Interface

```