

**IVANA FONTANIVE CAPANEMA**

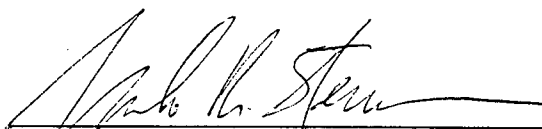
**AMBIENTE PARA CONSTRUÇÃO DE  
SISTEMAS ESPECIALISTAS EM TEMPO REAL  
PARA A ITAIPU BINACIONAL**


**FLORIANÓPOLIS**  
**1999**

# AMBIENTE PARA CONSTRUÇÃO DE SISTEMAS ESPECIALISTAS EM TEMPO REAL PARA A ITAIPU BINACIONAL

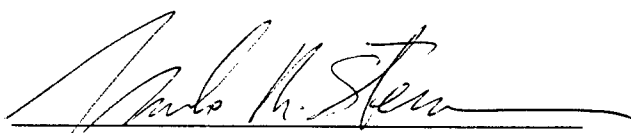
Ivana Fontanive Capanema

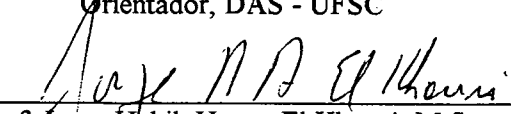
Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em *Controle, Automação e Informática Industrial*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

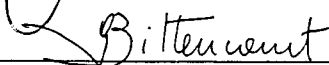
  
Prof. Marcelo Ricardo Stemmer, Dr.  
Orientador

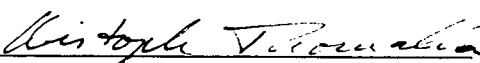
  
Prof. Ildemar Cassana Decker, D.Sc.  
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

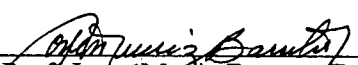
Banca Examinadora:

  
Prof. Marcelo Ricardo Stemmer, Dr.  
Orientador, DAS - UFSC

  
Prof. Jorge Habib Hanna El Khouri, M.Sc.  
Co-Orientador, ITAIPU Binacional

  
Prof. Guilherme Bittencourt, Dr.  
DAS - UFSC

  
Prof. Christoph Thomalla, Dr.  
DAS - UFSC

  
Prof. Jorge Muñoz Barreto, Dr.  
INE - UFSC

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**AMBIENTE PARA CONSTRUÇÃO DE  
SISTEMAS ESPECIALISTAS EM TEMPO REAL  
PARA A ITAIPU BINACIONAL**

Dissertação submetida à  
Universidade Federal de Santa Catarina  
como parte dos requisitos para a  
obtenção do grau de Mestre em Engenharia Elétrica.

**IVANA FONTANIVE CAPANEMA**

Florianópolis, Junho de 1999.

*A meus pais, Lourdes e Sid.*

*"Se você pode sonhar, pode fazer."  
Walt Disney*

## Agradecimentos

- *Ao Eng.º Jorge Habib Hanna El Khouri, orientador deste trabalho, pela dedicação e pelos ensinamentos transmitidos durante o período de desenvolvimento do projeto.*
- *Aos Professores Marcelo Ricardo Stemmer e Guilherme Bittencourt, pela orientação e atenção dada ao trabalho.*
- *Ao Eng.º João Ricardo Camargo, pelo apoio e incentivo passados durante este ano, fundamentais para a realização desta dissertação.*
- *Ao Professor Augusto Humberto Bruciapaglia, meu grande Mestre, pela amizade.*
- *Aos amigos de coração, que estiveram comigo nos momentos alegres e difíceis, em especial a Carlos Alberto Franzoi, Carlos Fernando Martins, Nicole Sharon Orfali, Sandra Regina Franzoi, Temiam Almeida de Moraes, Teresinha Arnauts e Veranildes Fontanive.*
- *Aos colegas do LCMi e da ITAIPU Binacional, que contribuíram com seus conhecimentos e amizade.*
- *À ITAIPU Binacional, que permitiu o desenvolvimento deste projeto.*
- *À Universidade Federal de Santa Catarina e ao CNPq, pelo apoio à formação discente.*

- *Ao Tio João Alberto Silva, por acreditar no meu potencial como pessoa, e cujo incentivo e carinho permitiram a realização deste trabalho.*
- *A Anderson Cattoni, pelo amor e companheirismo nesta fase importante de minha vida.*
- *A meus pais, Sid e Lourdes, que são meus grandes mestres, pois seus ensinamentos são a base de todo este trabalho.*
- *A Deus, que guiou meus passos até aqui.*

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

## **AMBIENTE PARA CONSTRUÇÃO DE SISTEMAS ESPECIALISTAS EM TEMPO REAL PARA A ITAIPU BINACIONAL**

**Ivana Fontanive Capanema**

Junho/1999

Orientador: Marcelo Ricardo Stemmer

Área de Concentração: Controle, Automação e Informática Industrial

Palavras-chave: Mecanismo de Inferência, Sistema Especialista, Tempo Real

Número de Páginas: 166

**RESUMO:** Esta dissertação descreve o desenvolvimento de um ambiente computacional, orientado à construção de sistemas especialistas em tempo real (*R-TESE - Real Time Expert System Environment*), cujo objetivo é possibilitar a implementação de Sistemas Especialistas que auxiliam o usuário do Sistema Elétrico a desempenhar suas tarefas de forma rápida e segura.

O R-TESE considera aspectos essenciais para aplicativos tempo real, como prioridade, multi-threads, regras e variáveis temporizadas, integração com banco de dados em tempo real.

O ambiente especialista recebe informações de outros sistemas em operação na Usina Hidrelétrica de ITAIPU, e analisa conjuntos de dados com informações sobre os pontos digitais e analógicos supervisionados, aplicando o raciocínio baseado em regras e apresentando ao operador instruções para solucionar eventuais problemas ocorridos.

Na seqüência deste trabalho, este ambiente será detalhado funcionalmente, utilizando o paradigma da orientação a objetos para representar a estrutura do sistema.

A implantação de sistemas desenvolvidos com o R-TESE disponibilizará, à supervisão do processo industrial da ITAIPU, ferramentas com concepção tecnologicamente atualizada, que agregam atributos de qualidade ao Sistema de Supervisão de Geração de Energia.



Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

## **ENVIRONMENT FOR CONSTRUCTION OF REAL-TIME EXPERT SYSTEMS TO ITAIPU BINACIONAL**

**Ivana Fontanive Capanema**

June/1999

Advisor: Marcelo Ricardo Stemmer

Area of Concentration: Control, Automation and Industrial Computing

Keywords: Inference Machine, Expert System, Real-Time

Number of Pages: 166

**ABSTRACT:** This dissertation describes an environment for the development of expert systems in real-time (*R-TESE - Real Time Expert System Environment*). Its goal is to facilitate the development of Expert Systems to aid the user of the ITAIPU System to carry out its tasks in a fast and safe way, decreasing the complexity in operational procedures.

R-TESE considers essential aspects of real-time applications, as priority, multi-threads, temporized rules and variable, integration with database in real-time.

The environment receives information, in real-time, from other systems in operation at the ITAIPU Power Plant. The program is able to analyze groups of data, containing information about the supervised digital and analogical points, using a rule-based reasoning, and to present to the operator the instructions to solve eventual problems.

In the course of this work, this project will be functionally detailed, and the object-oriented paradigm will be employed to represent the system structure.

The systems developed with R-TESE will provide ITAIPU operation with state-of-the-art tools, which aggregate quality attributes to the Energy Generation Supervisory System.

# Sumário

<b>Capítulo I - Introdução</b> .....	<b>1</b>
<b>Capítulo II - Delimitação do Problema</b> .....	<b>7</b>
II.1. Introdução.....	7
II.2. O Papel do SEs em Sistemas de Geração de Energia.....	8
II.3. A Hidrelétrica de ITAIPU no Contexto de Sistemas Interligados.....	11
II.3.1. Arquitetura Geral do Sistema Elétrico.....	12
II.4. Ambiente de Supervisão e Controle.....	15
II.4.1. Sistema de Supervisão da Operação - SSO.....	15
II.4.1.1. Componentes do Sistema de Supervisão.....	16
II.4.1.2. Arquitetura do SSO.....	21
II.4.2. Sistema SCADA.....	22
II.4.2.1. Funções Gerais.....	23
II.4.2.2. Sistema Especialista Previsto.....	24
II.4.3. Sistema SCC.....	24
II.5. Ambiente de Apoio à Operação.....	26
II.5.1. Sistemas Especialistas.....	26
II.5.2. Sistema Especialista de 50 Hz - IOI50.....	28
II.5.2.1. Objetivo e Funcionalidade.....	29
II.5.2.2. Arquitetura do Sistema.....	30
II.5.3. Sistema Especialista de 60 Hz - IOI60.....	31
II.5.3.1. Objetivo e Funcionalidade.....	32
II.5.3.2. Arquitetura do Sistema.....	33
II.6. Conclusão.....	35
<b>Capítulo III - Ambiente para Construção de Sistemas Especialistas</b> .....	<b>36</b>
III.1. Introdução.....	36
III.2. Especificação dos Requisitos.....	38
III.2.1. Requisitos Funcionais.....	38
III.2.2. Requisitos de Desempenho.....	38
III.2.3. Requisitos de Banco de Dados.....	39

III.3. Projeto da Arquitetura.....	40
III.3.1. Interfaces Internas.....	44
III.3.1.1. Objeto Gerenciador do R-TESE .....	45
III.3.1.2. Objeto Gerenciador de Variáveis .....	47
III.3.1.3. Objeto Gerenciador de Regras .....	47
III.3.1.4. Objeto Gerenciador de Pacotes.....	48
III.3.2. Interface de Programação.....	49
III.3.3. Interfaces Externas.....	50
III.3.3.1. Objeto responsável pela Interface SSO/R-TESE.....	51
III.3.3.1.1. Thread Associada.....	53
III.3.3.2. Objeto responsável pela Interface R-TESE/SSO .....	54
III.3.3.3. Bancos de Dados Associados.....	58
III.4. Estruturação do Ambiente.....	60
III.5. Considerações.....	64
III.6. Conclusão.....	65
<b>Capítulo IV - Gerenciador de Variáveis .....</b>	<b>66</b>
IV.1. Introdução .....	66
IV.2. Estrutura das Variáveis .....	67
IV.2.1. Bancos de Dados Associados.....	67
IV.2.2. Arquitetura Interna .....	68
IV.3. Gerenciador de Variáveis.....	75
IV.3.1. Dados.....	77
IV.3.2. Métodos .....	79
IV.3.3. Thread Associada .....	81
IV.4. Considerações.....	83
IV.5. Conclusão.....	83
<b>Capítulo V - Gerenciador de Regras .....</b>	<b>84</b>
V.1. Introdução .....	84
V.2. Estrutura das Regras .....	85
V.2.1. Bancos de Dados Associados.....	86
V.2.2. Arquitetura Interna.....	87
V.2.3. Funções Externas.....	93
V.3. Gerenciador de Regras.....	97
V.3.1. Modo de Raciocínio.....	99
V.3.2. Estratégia de Busca.....	99
V.3.3. Resolução de Conflito.....	100
V.3.4. Dados .....	101
V.3.5. Métodos.....	103
V.3.6. Thread Associada.....	106
V.4. Considerações.....	107
V.5. Conclusão.....	108

<b>Capítulo VI - Sistema Especialista de Apoio à Operação.....</b>	<b>109</b>
VI.1. Introdução.....	109
VI.2. Projeto de Sistemas Especialistas.....	110
VI.2.1. Interface com o Usuário.....	112
VI.2.2. Base de Conhecimento.....	113
VI.2.3. Base de Fatos.....	116
VI.3. Aspectos Gerais.....	118
VI.4. Interface com Ambientes Externos.....	123
VI.5. Testes e Validação dos Resultados.....	124
VI.6. Considerações.....	130
VI.7. Conclusão.....	132
<b>Capítulo VII - Conclusão.....</b>	<b>133</b>
<b>Apêndice 1 - Bancos de Dados Utilizados na Estruturação do R-TESE.....</b>	<b>137</b>
<b>Apêndice 2 - Base de Conhecimento Real     (notação dos operadores do Sistema ITAIPU).....</b>	<b>140</b>
<b>Apêndice 3 - Base de Conhecimento Real (notação no R-TESE).....</b>	<b>146</b>
<b>Apêndice 4 - Estruturação de Regras em Árvores de Objetos.....</b>	<b>157</b>
<b>Referências Bibliográficas.....</b>	<b>162</b>

# Lista de Figuras

## Capítulo II

Figura II.1 - Localização Geográfica da Usina Hidrelétrica ITAIPU Binacional .....	11
Figura II.2 - Diagrama Unifilar do Sistema de 60 Hz.....	13
Figura II.3 - Estrutura do Sistema de Supervisão da Operação.....	16
Figura II.4 - Diagrama Unifilar Principal do SSO.....	20
Figura II.5 - Subsistema de 60 Hz do SSO .....	20
Figura II.6 - Arquitetura do SSO .....	21
Figura II.7 - Arquitetura do SCC.....	25
Figura II.8 - Sistema Unifilar de 50 Hz.....	29

## Capítulo III

Figura III.1 - Arquitetura Típica do Ambiente Especialista.....	41
Figura III.2 - Fluxo de Dados do R-TESE .....	42
Figura III.3 - Intercâmbio de Informações entre SSO e R-TESE.....	51
Figura III.4 - Fluxo de Dados do Mecanismo de <i>Thread</i> .....	53
Figura III.5 - Fluxo de Dados do Protocolo de Comunicação.....	56
Figura III.6 - Inter-relação entre BDs Externos do R-TESE.....	59
Figura III.7 - Diagrama Representativo do R-TESE .....	61

## Capítulo IV

Figura IV.1 - Variáveis do R-TESE.....	69
Figura IV.2 - Arquitetura Interna das Variáveis.....	71
Figura IV.3 - Diagrama Representativo do Gerenciador de Variáveis.....	76
Figura IV.4 - Funcionamento do Mecanismo de <i>Thread</i> .....	81

## Capítulo V

Figura V.1 - Regras do R-TESE .....	87
Figura V.2 - Hierarquia do Objeto <i>Regra</i> .....	90
Figura V.3 - Arquitetura Interna das Regras .....	90
Figura V.4 - Funções Externas Associadas .....	94
Figura V.5 - Diagrama Representativo do Gerenciador de Regras .....	98
Figura V.6 - Fluxo de Dados do Mecanismo de <i>Thread</i> .....	106

## Capítulo VI

Figura VI.1 - Projeto de Sistemas Especialistas .....	111
Figura VI.2 - Itens de Menu para Execução de Tarefas.....	112
Figura VI.3 - Tela Especifica do R-TESE - Atributos.....	114
Figura VI.4 - Tela Especifica do R-TESE - Testes.....	114
Figura VI.5 - Tela Especifica do R-TESE - Variáveis.....	117
Figura VI.6 - Bancos de Dados Externos.....	119
Figura VI.7 - Calendário .....	120
Figura VI.8 - Visualizador de Regras em Estrutura de Árvore.....	121
Figura VI.9 - Janela Indicativa de Problemas Ocorridos .....	122
Figura VI.10 - Janela de Depuração do Mecanismo de Inferência.....	123
Figura VI.11 - IHM Externa.....	124
Figura VI.12 - Alteração da Tensão da Barra de 500 kV da GIS.....	128
Figura VI.13 - Disparo da Regra Relativa ao Período de Carga .....	129
Figura VI.14 - Alteração do Período de Carga.....	130
Figura VI.15 - Ajuda do R-TESE .....	131

# Lista de Abreviaturas

<b>ANDE</b>	–	Administración Nacional de Eletricidad
<b>BC</b>	–	Base de Conhecimento
<b>BD</b>	–	Banco de Dados
<b>BF</b>	–	Base de Fatos
<b>CAG</b>	–	Controle Automático de Geração
<b>CCR</b>	–	Central Control Room
<b>CF</b>	–	Casa de Força
<b>CLP</b>	–	Controlador Lógico Programável
<b>CMO</b>	–	Comissão Mista de Operação
<b>CNOS</b>	–	Centro Nacional de Operação do Sistema
<b>CPM</b>	–	Controlador do Painel Mímico
<b>DAS</b>	–	Data Acquisition System
<b>DBA</b>	–	Data Base Administrator
<b>DCM</b>	–	Dispositivo de Comunicação Microprocessado
<b>DESPRO</b>	–	Programa de Análise de Desligamentos Programados
<b>DS</b>	–	Data Server
<b>EC</b>	–	Estação Central
<b>EMS</b>	–	Energy Management System
<b>FE</b>	–	Front-End
<b>GIS</b>	–	Gas Insulated Substation
<b>HFD</b>	–	Historical and Future Date Management
<b>IHM</b>	–	Interface Homem-Máquina
<b>IOI</b>	–	Instrução de Operação da Interligação
<b>MD</b>	–	Margem Direita
<b>MI</b>	–	Mecanismo de Inferência
<b>MINOS</b>	–	Módulo Inteligente para Operação de Subestações
<b>MMI</b>	–	Man-Machine Interface
<b>LAN</b>	–	Local Area Network

<b>PM</b>	–	Painel Mímico
<b>PC</b>	–	Personal Computer
<b>R-TESE</b>	–	Real Time Expert System Environment
<b>RTES</b>	–	Real Time Expert System
<b>SCADA</b>	–	Supervisory Control And Data Acquisition
<b>SCC</b>	–	Sistema de Controle Computadorizado
<b>SEAO</b>	–	Sistema Especialista de Apoio à Operação
<b>SEMD</b>	–	Sistema Especialista da Margem Direita
<b>SF6</b>	–	Hexa Fluoreto de Enxofre
<b>SE</b>	–	Sistema Especialista
<b>SE-FI</b>	–	Subestação de Foz do Iguaçu
<b>SE-IPU</b>	–	Subestação de Itaipu
<b>SE-IT</b>	–	Subestação de Itaberá
<b>SE-IV</b>	–	Subestação de Ivaiporã
<b>SE-TP</b>	–	Subestação de Tijuco Preto
<b>SGBD</b>	–	Sistema de Gerenciamento de Banco de Dados
<b>SSO</b>	–	Sistema de Supervisão da Operação
<b>TIF</b>	–	Telecontrol Interface
<b>UTR</b>	–	Unidade Terminal Remota



# Capítulo I

## Introdução

O setor elétrico tem recebido grande enfoque nos estudos que envolvem novas tecnologias, devido ao papel estratégico que as fontes de energia e todo o seu aparato desempenham no suporte ao desenvolvimento de um país.

De modo mais específico, pode ser citada a automatização crescente nas atividades de geração, transmissão e distribuição de energia. Algumas companhias brasileiras já passaram pela barreira da consolidação de seus serviços e se defrontam, atualmente, com o aprimoramento dos seus procedimentos e processos operacionais, em busca da qualidade total. Esta busca pela qualidade representa pesquisar e implantar técnicas modernas de apoio e execução de atividades essenciais ao funcionamento do sistema elétrico, em todas as suas etapas [El Khouri 93].

Vários projetos estão sendo conduzidos com o objetivo de aumentar o nível de automação do processo elétrico, envolvendo o emprego de vários tipos de equipamentos digitais, cada qual com um conjunto de funções distintas, principalmente voltados à área de aquisição de informações, supervisão e controle de processos.

A ITAIPU Binacional, sensível à necessidade de usufruir destas novas tecnologias, e devido à sua importância estratégica como fornecedora de energia elétrica ao Paraguai e ao Brasil, manifestou o interesse em desenvolver sistemas capazes de auxiliar os operadores e despachantes de carga a tomar decisões de forma rápida e efetiva.

Esta situação impõe maiores investimentos em treinamento periódico e na pesquisa de mecanismos mais avançados e automatizados de auxílio à tomada de decisões. Constatase que a maioria dos trabalhos busca otimizar o tempo de resolução dos diversos problemas rotineiros e emergenciais associados à operação de sistemas elétricos, propondo a interação entre os programas convencionais e aqueles baseados em heurísticas, que levam em consideração a experiência dos operadores, obtida ao longo de muitos anos de trabalho [Fernandes 95].

Hoje, há uma tendência de aplicação dos métodos de Inteligência Artificial [Shirai 82][Levine 88] para a solução destes problemas. A ênfase atual recai no uso de *Sistemas Especialistas* pelo fato destes modelarem adequadamente a forma de pensar humana, ao resolver um problema complexo em um domínio restrito.

Um Sistema Especialista pode ser conceitualmente definido como um programa cujo comportamento assemelha-se ao de um perito em algum domínio de aplicação [Charniak 85][Barr 86][El Khouri 93][Bittencourt 98]. São programas de computador que procuram atingir soluções de determinados problemas do mesmo modo que especialistas humanos, se estiverem sob as mesmas condições.

Tradicionalmente, técnicas de resolução de problemas baseadas em conhecimento têm sido aplicadas em domínios onde os dados são estáticos, e não são exigidas respostas críticas no tempo. Muitos dos novos sistemas de diagnóstico, projeto e configuração apresentam estas características. Entretanto, pesquisadores em Inteligência Artificial encontram no domínio dos *Sistemas Especialistas em Tempo Real* um novo conjunto de problemas complexos, onde alta performance, velocidade de resposta e integração com outros componentes são fundamentais [Laffey 88][Dodhiawala 89].

Estes sistemas são adequados para garantir a segurança de plantas industriais complexas, pois auxiliam os operadores na tomada de decisões corretas de forma rápida, melhorando a qualidade dos serviços.

Neste sentido, a tendência à utilização de Sistemas Especialistas que auxiliem a supervisão da operação do processo elétrico tem se mostrado irreversível, devido à complexidade da geração e transmissão de energia, e à grande confiabilidade requerida do

sistema pelas conseqüências sociais representadas pela falta de energia nos centros urbanos e em complexos industriais [Capanema 97].

Atualmente, a Usina Hidrelétrica de ITAIPU Binacional dispõe de um Sistema de Supervisão da Operação [Capanema 97], que acompanha as condições operativas do sistema elétrico, com o objetivo de assegurar a qualidade do suprimento, monitorando o funcionamento do processo como um todo. Esta ferramenta proporciona uma visão instantânea do modo de operação dos principais elementos que compõem o Sistema ITAIPU.

Destaca-se que seus operadores (despachantes) são encarregados de observar o comportamento do sistema de potência, coordenando todos os procedimentos estabelecidos pelas instruções de operação da planta.

Assim, analisando-se as funcionalidades do sistema disponível, vislumbra-se a possibilidade de incrementar os serviços oferecidos. Busca-se, então, uma forma de aprimorá-los, através do desenvolvimento de um mecanismo inteligente que facilite as ações/decisões do usuário, proporcionando maior confiabilidade, facilidades de interface e capacidade de expansão.

Desta forma, o desenvolvimento de um ambiente, orientado à construção de Sistemas Especialistas em tempo real torna-se a base fundamental deste projeto. Este ambiente denomina-se R-TESE (*Real Time Expert System Environment*), e caracteriza-se como uma ferramenta computacional que disponibiliza técnicas de Inteligência Artificial para construção de Sistemas Especialistas.

O R-TESE utiliza um modelo de representação do conhecimento baseado em regras de produção, tendo como objetivo principal simplificar o trabalho de implementação de *Sistemas Inteligentes* através do uso de um mecanismo de inferência, e da utilização de explicações sensíveis ao contexto da Base de Conhecimento modelada, além de considerar restrições devido ao tempo de resposta, como a utilização de prioridades, multi-threads, regras e variáveis temporizadas e integração com banco de dados em tempo real.

Com o auxílio do R-TESE, busca-se pesquisar, desenvolver e apresentar um protótipo de um Sistema Especialista em tempo real que supra as necessidades da Usina de

ITAIPU. No caso do SE em discussão, o domínio da aplicação é o contexto de abrangência do Sistema Elétrico interligado de ITAIPU - setor de 60 Hz, que é regido pelas Instruções de Operação de 60 Hz.

O SE recebe informações em tempo real sobre a configuração do sistema elétrico, vindas do Sistema de Supervisão da Operação - SSO. A partir desta informação, o sistema seleciona as regras envolvidas e apresenta ao despachante as causas de uma eventual anormalidade ocorrida, bem como as ações pertinentes previstas nas Instruções de Operação.

O ambiente proposto possui uma concepção tecnologicamente atualizada, capaz de analisar o estado de linhas, barras, disjuntores, seccionadoras, e informar ao operador do Sistema ITAIPU a melhor maneira de agir sobre estes elementos. Permite a melhoria das funções operacionais em condições normais, e uma maior agilidade em situações de emergência, considerando o volume de dados sempre crescente.

Amplia-se, assim, o Sistema de Supervisão da Operação, que atua paralelamente ao ambiente inteligente, confirmando suas características de escalabilidade, usabilidade e outros atributos de qualidade.

Objetivando a melhoria nos procedimentos envolvidos na operação da Usina Hidrelétrica de ITAIPU, este estudo pretende atingir as seguintes metas:

- Desenvolver um ambiente para construção de Sistemas Especialistas, voltado ao setor de geração e transmissão de energia, que apresente alta velocidade de resposta. Esta ferramenta deve permitir a estruturação de Sistemas Inteligentes de forma fácil, com o intuito de suprir as necessidades da área de operação de usinas e subestações;
- Utilizar como base as funcionalidades do Sistema de Supervisão da Operação, existente na Usina e, a partir deste, construir o protótipo de um Sistema Especialista de apoio à operação, cuja Base de Conhecimento seja fundamentada nas Instruções de Operação da Interligação do setor de 60 Hz;

- Definir uma Interface Homem-Máquina que permita a análise dos dados de forma simples, fazendo com que o usuário possa intervir sobre o processo de maneira rápida e segura;
- Implementar um Editor de Regras que auxilie o desenvolvedor dos Sistemas Especialistas na definição de Bases de Fatos e Conhecimento;
- Atender os requisitos funcionais e de desempenho (tempo de resposta) desejados, apresentando resultados confiáveis para auxílio ao operador/despachante de carga do Sistema ITAIPU.

Com o intuito de atingir os objetivos propostos, divide-se este trabalho em sete capítulos, organizados da seguinte forma:

O capítulo I destaca a importância dos Sistemas Especialistas no apoio à operação dos sistemas elétricos, devido às exigências de qualidade e confiabilidade no atendimento aos consumidores de energia. Mostra-se a necessidade de desenvolvimento de uma ferramenta capaz de modelar Sistemas Especialistas direcionados à área de operação do sistema elétrico, que auxiliem os operadores na análise das condições do sistema e na tomada de decisões de forma rápida e eficiente, aumentando o grau de automatização do processo.

O capítulo II descreve a Hidrelétrica de ITAIPU, definindo sua arquitetura, e os sistemas que a compõem, detalhando o ambiente de supervisão e controle da ITAIPU, responsável pela execução de funções de acompanhamento e monitoração, em tempo real, das grandezas analógicas e digitais do Sistema Elétrico. O capítulo destaca, também, a evolução histórica dos Sistemas Especialistas no setor elétrico, descrevendo as ferramentas disponíveis na empresa, culminando com o trabalho proposto, que contribui para a qualidade dos serviços de supervisão da área operacional.

O capítulo III descreve a ferramenta para construção de Sistemas Especialistas em tempo real, intitulado R-TESE. Os requisitos funcionais deste ambiente computacional, bem como as características básicas de seu modo de operação, são ilustrados. É feito o

detalhamento de sua estrutura de dados e das funções que compõem o processo, destacando sua forma de intercâmbio de informações com entidades externas.

O capítulo IV define o Gerenciador de Variáveis do R-TESE, estrutura representativa da Base de Fatos do sistema. Detalha-se sua estrutura de dados e as funções que o compõem, destacando o funcionamento de seu mecanismo de *thread*, na atualização das variáveis do sistema em estudo.

O capítulo V detalha o funcionamento do Gerenciador de Regras do R-TESE, que administra as ações sobre a Base de Conhecimento do sistema. Como no capítulo anterior, ilustra-se sua estrutura de dados, definindo o funcionamento do mecanismo de inferência e como este interage com o processo, caracterizando o escopo do ambiente computacional relatado neste trabalho.

No capítulo VI são mostradas as características da implementação do Sistema Especialista de Apoio à Operação, a integração com o Sistema de Supervisão local, os testes realizados para a validação dos dados, e a Interface Homem-Máquina adotada para interagir com o usuário, culminando com o protótipo de um SE que oferece ao operador do ambiente ITAIPU facilidades na área de operação.

O capítulo VII conclui o trabalho, evidenciando a aplicabilidade do sistema desenvolvido, suas restrições e sugestões para projetos futuros.

O apêndice engloba a Base de Conhecimento do setor de 60 Hz (utilizada na validação do sistema), a arquitetura dos bancos de dados que armazenam as informações relevantes ao funcionamento do ambiente inteligente, e os detalhes de estruturação das regras em árvores de objetos.

# Capítulo II

## Delimitação do Problema

Este capítulo descreve os principais aspectos da Usina Hidrelétrica de ITAIPU Binacional, caracterizando a abrangência necessária para englobar necessidades de supervisão e controle, devido à sua importância estratégica como fornecedora de energia ao Brasil e ao Paraguai.

Define-se o *Sistema de Supervisão da Operação* - SSO, responsável pela execução de funções de acompanhamento e monitoração, em tempo real, das grandezas analógicas e digitais do Sistema ITAIPU, e realiza-se, também, uma abordagem resumida do SCC e do projeto previsto para o Sistema SCADA, destacando sua importância no contexto da automação industrial.

Detalha-se, ainda, os Sistemas Especialistas que compõem a área de operação da Usina, com o objetivo de definir o escopo do ambiente proposto, cuja diretriz principal é suprir a empresa em questão com um ambiente para desenvolvimento de Sistemas Especialistas atualizado, que venha contribuir para a melhoria da qualidade de operação.

### II.1. Introdução

A construção, operação e manutenção de uma usina como a ITAIPU determina que se utilize o conhecimento oriundo de quase todas as ciências. Hoje, a ITAIPU tem à

disposição sistemas especialistas, sistemas em tempo real, sistemas em rede, sistemas em ambiente de grande porte, sistemas em PC, sistemas dedicados à aquisição de dados, entre outros.

Com a crescente necessidade de melhoria da qualidade dos serviços prestados, surge a importância de ambientes cada vez mais automatizados.

Neste contexto, destaca-se a arquitetura do Sistema ITAIPU e a experiência adquirida com o desenvolvimento do Sistema de Supervisão da Operação, que incluiu desde o projeto e fabricação dos circuitos eletrônicos de aquisição de dados, até o software em tempo real, com interface completamente gráfica.

## **II.2. O Papel dos SEs em Sistemas de Geração de Energia**

O progresso industrial de uma nação pode ser medido pelo grau de aproveitamento de suas fontes de energia. A descoberta destas fontes na natureza, o transporte da energia em suas várias formas, e a conversão dessa energia para formas mais úteis são partes essenciais de uma economia industrial.

A geração e transmissão de energia elétrica envolve uma estrutura extremamente complexa de componentes interligados: usinas, subestações, linhas, operadores, despachantes, sendo que os esforços despendidos para assegurar a continuidade no fornecimento desta energia constituem tarefas críticas para a qualidade dos serviços prestados [El Khouri 91].

A operação de um sistema de potência, pelo elevado grau de incerteza e pelas inúmeras variáveis que manipula, é intrinsecamente complexa. As várias ações de operação e controle requerem a presença de um operador capaz de, manipulando vários tipos de dados e informações, responder às mais diversas solicitações de forma eficiente e efetiva [Ribeiro 95]. Assim, com o intuito de garantir a confiabilidade do sistema, são necessárias abordagens avançadas nas áreas de controle, operação e supervisão dos sistemas de potência.



Com a crescente digitalização de subestações, as técnicas de inteligência artificial têm sido apontadas como a alternativa mais adequada para a representação e solução dos problemas de operação e controle de sistemas elétricos, possibilitando a substituição da decisão/ação humana por uma ação artificial de mesmo nível de eficácia e otimizando os processos operativos do ponto de vista funcional e econômico [Ribeiro 95].

Assim, os Sistemas Especialistas ganham espaço, e estão sendo utilizados em usinas e subestações elétricas, garantindo a evolução da supervisão e do controle local.

Em [Lacerda 87] é apresentado o protótipo de um Sistema Especialista denominado MINOS - Módulo Inteligente para Operação de Subestações. Este sistema foi implementado e testado em um microcomputador IBM-PC compatível, em linguagem PROLOG, onde são destacadas as funções de diagnóstico de defeitos e tratamento de alarmes. A análise do desempenho do protótipo proposto permitiu mostrar a importância do uso de um Sistema Especialista no diagnóstico e na tomada de decisões em subestações e validar as técnicas de representação de conhecimentos e de procura adotadas. Os testes efetuados nesta primeira versão mostraram que o protótipo permite o tratamento de um número de alarmes compatível com as ocorrências das subestações estudadas, bem como comprovaram as melhorias em termos de rapidez na execução de uma tarefa e a diminuição do grau de incerteza na tomada de decisões.

Em [Pinto 98] é apresentado o Programa de Análise de Desligamentos Programados - DESPRO, que objetiva a aplicação de métodos de análise baseados em regras heurísticas, complementares aos programas de análise de rede utilizados nos estudos de planejamento da operação. O programa consiste em um Sistema Especialista que, quando em operação, é capaz de analisar um conjunto de requisições de desligamento, efetuar o raciocínio baseado nas regras e, por fim, compor bases de dados do sistema de potência degradado. Estas bases de dados são enviadas aos programas de análise de rede e um relatório final é emitido. Como consequência, o especialista tem como avaliar a severidade das violações pós-contingência, além de fazer análises otimizadas para redespacho de geração e corte de carga, e realizar estudos estatísticos das requisições de desligamento sob o ponto de vista econômico. O DESPRO foi escrito em linguagem C padrão, com bibliotecas comerciais de uso específico para o Sistema Especialista, portátil para diversas máquinas.

Em [Silva 97] objetiva-se a aplicação de um sistema híbrido, integrando lógica nebulosa, redes neurais e Sistemas Especialistas, para diagnóstico de falhas a partir do processamento de alarmes, originados de um sistema elétrico real, composto por duas subestações e seus componentes associados. O sistema elétrico escolhido é formado pelas linhas de transmissão de 230 kV existentes entre as subestações de Palhoça e Jorge Lacerda (ambas da ELETROSUL) e seus respectivos transformadores. A partir dos pontos de monitoração, é gerada uma lista de falhas do sistema e dos alarmes que as caracterizam. O objetivo do sistema inteligente é gerar um diagnóstico da falha, isto é, identificar o distúrbio que originou um determinado conjunto de alarmes e fornecer uma possível localização do mesmo.

Para cada transformador 230/138 kV, utilizou-se um Sistema Especialista simples, usando regras de produção. Para as linhas de transmissão, optou-se por um modelo de sistema híbrido, englobando redes neurais, lógica nebulosa e Sistemas Especialistas, pois a relação entre falhas e alarmes é bem mais complexa, necessitando do emprego de técnicas mais aprimoradas. A taxa de acerto de 99% e o tempo de resposta de 55 milisegundos, obtidos com este sistema, são resultados bastante promissores, além de suas características de portabilidade e modularidade garantirem a viabilidade da aplicação.

Em [Crespo 96] descreve-se a implementação do RTES (Real-Time Expert System) que combina propriedades de regulação com técnicas de Sistemas Especialistas em tempo real, como predição de estados futuros. O algoritmo apresentado tem como objetivo resolver o problema desejado no menor tempo possível, respeitando os *deadlines* requeridos. Este sistema é capaz de executar cerca de 6000 predições em 2,85 segundos, e está sendo utilizado, no momento, para controlar o funcionamento de fornos de cimento.

Em [El Khouri 93] e [Herreros 96], destaca-se a experiência que a ITAIPU possui na utilização de Sistemas Especialistas para auxílio ao despacho de carga da Usina. As características destas ferramentas, suas funcionalidades e operacionalização serão detalhadas nas próximas seções.

Outros trabalhos, com ênfase na utilização de Sistemas Especialistas, estão sendo desenvolvidos, com o intuito de aumentar o grau de automação das subestações, prover maior qualidade e confiabilidade à operação e auxiliar o trabalho dos despachantes na tomada de decisões corretas mediante problemas operacionais complexos [IEEE

93][Martino 98][Électra 97].

O ambiente especialista proposto neste trabalho está orientado a minimizar os problemas na área de operação da Usina Hidrelétrica de ITAIPU, apresentando uma abordagem em tempo real, que auxilie o usuário do sistema a desempenhar suas tarefas.

### II.3. A Hidrelétrica de ITAIPU no Contexto de Sistemas Interligados

A Usina Hidrelétrica de ITAIPU é uma empresa binacional, localizada às margens do Rio Paraná, na fronteira entre Brasil e Paraguai, como ilustra a figura II.1.

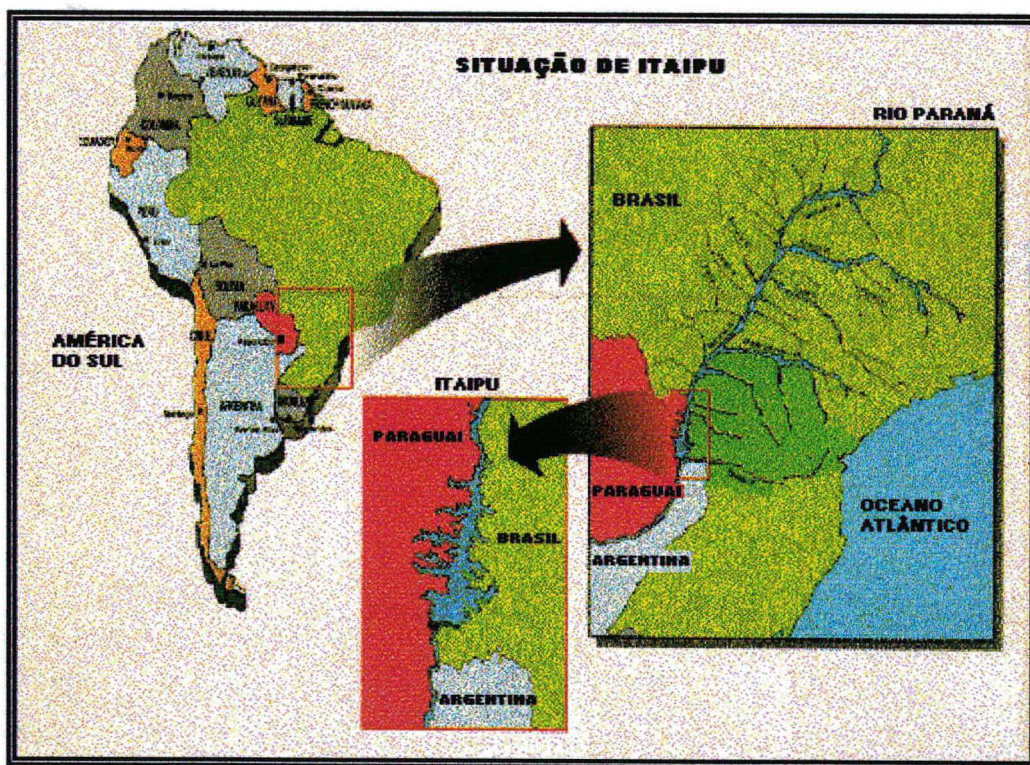


Figura II.1 - Localização Geográfica da Usina Hidrelétrica ITAIPU Binacional

A ITAIPU é responsável pelo suprimento aproximado de 32% da energia consumida no Brasil e 89% da energia consumida no Paraguai, possuindo elevada capacidade de transmissão (12600 MW).

A transmissão de energia, no Brasil, é feita através de um sistema misto, onde parte desta energia é transportada em corrente alternada (tensão de 750 kV), e parte em corrente contínua (tensão de 600 kV). Os dois sistemas partem da Subestação de Furnas, em Foz do Iguaçu, e conduzem energia à região de São Paulo, onde esta é injetada no sistema interligado brasileiro, atendendo às regiões Sul, Sudeste e Centro-Oeste.

A energia destinada à Região Sul utiliza apenas corrente alternada, a partir de Ivaiporã, onde uma subestação da Eletrosul recebe, em 500 kV, a parcela destinada aos Estados do Paraná, Santa Catarina e Rio Grande do Sul.

O Sistema de Transmissão em Corrente Contínua é constituído pelas Subestações Conversoras de Foz do Iguaçu e Ibiúna, e duas linhas de transmissão interligando estas duas subestações.

O Sistema de Transmissão em Corrente Alternada é, por sua vez, composto pelas seguintes subestações: Foz do Iguaçu, Ivaiporã, Itaberá, Tijuco Preto, e duas linhas de transmissão em circuito simples, interligando as quatro subestações, estando a terceira linha em fase de construção. A figura II.2 ilustra a configuração final deste sistema.

### **II.3.1. Arquitetura Geral do Sistema Elétrico**

A ITAIPU é composta basicamente pelas seguintes instalações e equipamentos [ITAIPU 89]:

- *Casa de Força e Unidades Geradoras*

A Casa de Força possui capacidade para 20 unidades geradoras, com capacidade individual de geração de 700 MW.

Atualmente, existem 18 unidades instaladas, sendo 2 para futuras instalações. As unidades numeradas de 1 a 9 possuem frequência de geração de 50 Hz, enquanto que as numeradas de 10 a 18 apresentam frequência de 60 Hz. As 18 máquinas são interligadas à Subestação Isolada a Gás (GIS), também localizada na Casa de Força.

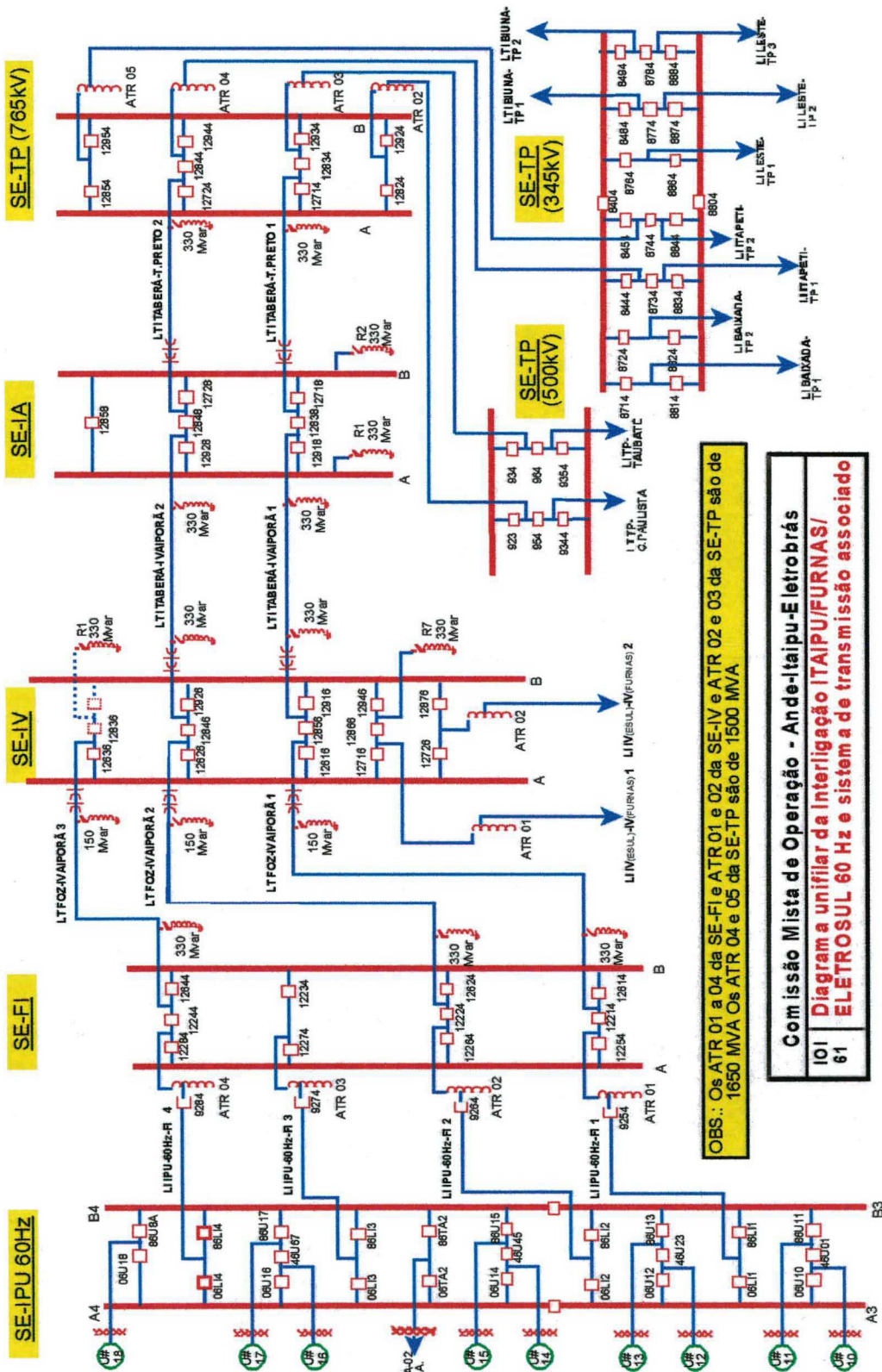


Figura II.2 - Diagrama Unifilar do Sistema de 60 Hz

- *Subestação GIS*

Na Subestação Isolada a Gás, a tensão das unidades geradoras é elevada de 18 kV para 500 kV, através de transformadores conectados à uma subestação blindada, isolada a gás SF<sub>6</sub> (Hexa Fluoreto de Enxofre). Este gás elimina a necessidade de longas distâncias para isolamento. A GIS é dividida em dois setores: setor de 60 Hz (interconectado à Subestação de FURNAS) e setor de 50 Hz (ligado à Subestação da Margem Direita).

- *Subestação da Margem Direita*

Localiza-se na margem direita do Rio Paraná, a dois quilômetros da Casa de Força. A comunicação entre a Subestação Isolada à Gás e a Subestação da Margem Direita é feita através de duas linhas de transmissão independentes.

A SEMD faz a ligação entre o Sistema ANDE (Administración Nacional de Eletricidad - Paraguai) e FURNAS (Brasil). A energia do setor de 50 Hz de FURNAS é convertida para corrente contínua, e transmitida aos centros de consumo do Brasil, onde é, novamente, convertida para corrente alternada, agora em 60 Hz.

- *Barragem Principal*

Localizada acima da Casa de Força, a Barragem Principal tem a função de direcionar a água do reservatório para as turbinas.

- *Vertedouro*

O Vertedouro está localizado na margem direita do Rio Paraná, e é responsável pela descarga da água que não vai para as turbinas.

A operação da ITAIPU é realizada em dois níveis hierárquicos: despacho de carga e operação da usina. O *despacho de carga* é responsável pela coordenação e supervisão da operação com as companhias ligadas às áreas de energia. Neste nível, são coordenados todos os equipamentos ligados ao Sistema ITAIPU, com tensão maior ou igual a 18 kV.

A área de *operação da usina* é responsável pela supervisão e execução de

chaveamento e ações de controle em todos os equipamentos da ITAIPU, ou seja, em equipamentos que funcionam abaixo de 18 kV.

Esta filosofia de operação adotada é suportada por *Sistemas de Supervisão e Controle*. Estes sistemas têm como função básica automatizar pontos de aquisição de dados, efetuando a supervisão de seus valores, e controlando-os conforme desejado.

A descrição dos sistemas disponíveis para monitoração da usina é feita nos próximos itens, onde é assinalada a importância da qualidade de seus serviços.

## **II.4. Ambiente de Supervisão e Controle**

A supervisão da operação pode ser definida como um conjunto de tarefas, associadas ao acompanhamento e à correção das condições operativas de um sistema elétrico, com o objetivo de assegurar a continuidade e a qualidade do suprimento, monitorando o funcionamento de todo o processo [Capanema 97].

Com a necessidade de dispor de um mecanismo com estas características, a fim de proporcionar uma visão instantânea do modo de operação dos principais elementos que compõem o Sistema Elétrico, foi concebido e implementado o Sistema de Supervisão da Operação. A estruturação de sua arquitetura e a especificação de seus requisitos são definidos a seguir, de forma a situá-lo no contexto deste estudo.

### **II.4.1. Sistema de Supervisão da Operação - SSO**

Após indefinições dos contratos que previam o fornecimento do sistema de supervisão e controle do tipo SCADA/EMS, a ITAIPU viu a necessidade de suprir o despacho e a operação da usina com uma ferramenta que proporcionasse o acompanhamento das principais grandezas de seu sistema elétrico. Assim, deu-se início a um projeto próprio, que incorporasse tais funcionalidades.

A primeira versão do SSO foi operacionalizada em maio de 1991, sendo desenvolvida por técnicos da ITAIPU, e implementada em ambiente DOS.

Com o crescente número de informações no sistema, fez-se necessidade a atualização do SSO, de forma a adequá-lo, cada vez mais, ao contexto dos sistemas de supervisão atuais.

O Sistema de Supervisão da Operação, atualmente, constitui a principal ferramenta de monitoração da Hidrelétrica de ITAIPU Binacional, sendo capaz de monitorar cerca de 1000 grandezas.

A essência de sua arquitetura está definida com base na interconexão dos subsistemas que o compõem, sendo eles: Unidades Terminais Remotas, Front-End, Estação Central e Painel Mímico, destacados na figura II.3.

Os elementos apresentados nesta figura serão descritos em mais detalhes nas seções que seguem.

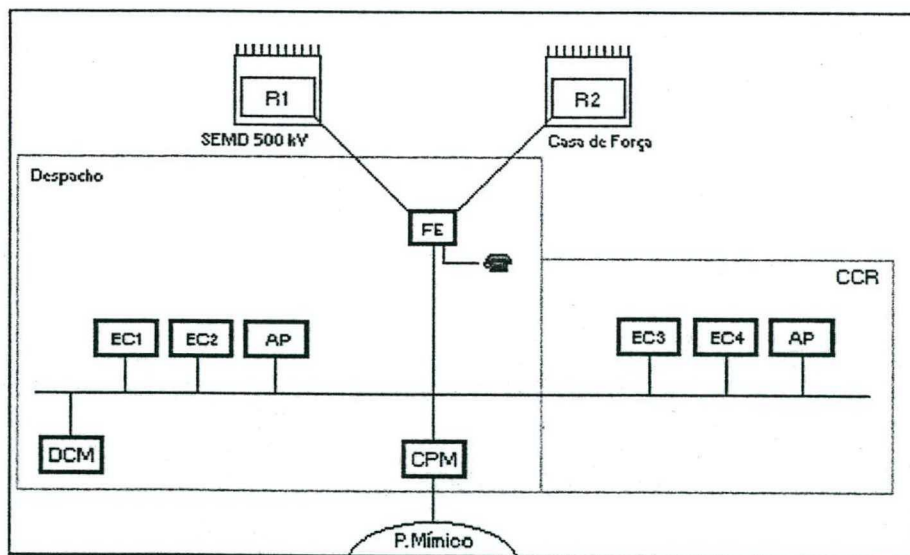


Figura II.3 - Estrutura do Sistema de Supervisão da Operação

#### II.4.1.1. Componentes do Sistema de Supervisão

Aqui, é feita uma descrição detalhada dos componentes do ambiente de supervisão, ilustrando a importância do SSO na garantia dos serviços prestados.



### **a. Unidades Terminais Remotas**

As Unidades Terminais Remotas R1 e R2 estão situadas, respectivamente, na Subestação da Margem Direita e na Casa de Força, e têm por objetivo o interfaceamento do Sistema de Supervisão com as grandezas analógicas e digitais presentes em campo.

As unidades de processamento possuem as seguintes características:

- ♦ Aquisição e tratamento de grandezas analógicas e digitais;
- ♦ Processamento em tempo real;
- ♦ Uso de memória não-volátil para parâmetros, programas e dados históricos;
- ♦ Reinicialização automática, no caso de falha na alimentação do sistema;
- ♦ Detecção de falhas, para assegurar status inoperante.

As remotas são responsáveis pela aquisição dos dados disponibilizados pelos transdutores, sendo constituídas por placas montadas em plataforma de PC. Estas placas são a essência da unidade, uma vez que realizam a aquisição e a conversão dos sinais analógicos em digitais, para serem agrupados no PC, que os envia a um processador de comunicação denominado Front-End, via comunicação serial RS-232.

### **b. Front-End**

O Sistema de Supervisão da Operação possui um gerenciador de comunicação, localizado entre a Estação Central e as duas Unidades Remotas.

Este gerenciador, denominado Front-End, é responsável pelo controle e sincronização da transmissão de dados dentro do processo, administrando a troca de informações entre remotas, Estação Central e linhas telefônicas. O Software do Front-End administra o protocolo de comunicação essencial ao SSO, sendo capaz de enviar informações seguras ao Sistema de Supervisão, através da análise das informações recebidas das unidades remotas, detectando e identificando erros ou falhas na aquisição de dados.

O Software do Front-End está implementado em DOS. É executado independente dos outros equipamentos, e está localizado nas sala de despacho da usina. Falhas no seu funcionamento causam o não recebimento das informações referentes às remotas, culminando com a não monitoração do processo elétrico.

### c. Estação Central

A Estação Central é responsável pela monitoração do dados vindos do Front-End, informando aos operadores do sistema o estado das grandezas da Usina.

Incorporando funcionalidades como detecção e monitoração de falhas, geração de alarmes, indicação de eventos, reinicialização e reconfiguração automática de computadores, front-ends e servidores, a Estação Central caracteriza-se como componente essencial do Sistema de Supervisão da Hidrelétrica de ITAIPU.

A tabela II.1 sintetiza a quantidade de informações supervisionadas em tempo real.

	Analógicos	Digitais	Total
Margem Direita	31	214	245
Casa de Força	62	70	132
Calculados	50	2	52
Manuais	90	461	551
Total	233	747	980

Tabela II.1 - Pontos Supervisionados na Hidrelétrica de ITAIPU

O Software da Estação Central recebe, em tempo real, todas as informações processadas pelo Front-End. Estas informações são analisadas e exteriorizadas, de forma a instruir o usuário sobre o estado de disjuntores, geradores, seccionadoras, pólos, filtros, e grandezas analógicas do sistema, cuja representação gráfica é exteriorizada em diagramas unifilares.

O operador tem acesso a informações sobre o estado do sistema elétrico, interagindo sobre ele através das facilidades e funções incorporadas.

O Software da Estação Central apresenta funcionalidades de sistemas de supervisão de grande porte, como disponibilização de log de dados, log de eventos, gráficos de tendências, tratamento de alarmes, indicação de falhas na recepção dos dados do processo, e envio de pacotes de informações a sistemas externos.

As figuras II.4 e II.5 ilustram telas que compõem o SSO - seu diagrama principal e o Subsistema de 60 Hz, estudado mais profundamente na estruturação e implementação deste trabalho.

#### **d. Painel Mímico**

O Painel Mímico está localizado na sala de despacho da usina, e é utilizado para prover, ao operador, informações sobre o status do sistema de transmissão de energia.

O Software do Painel Mímico lê pacotes de dados, contendo o valor e o estado de todos os pontos analógicos e digitais disponibilizados pela Estação Central, e exterioriza estas informações em *displays*, presentes no Painel, para acompanhamento em tempo real.

#### **e. Subsistemas Externos**

Existe um módulo, identificado como DCM (Dispositivo de Comunicação Microprocessado), que é responsável pelo gerenciamento da comunicação com o Centro Nacional de Operação do Sistema (CNOS), onde se realiza a supervisão a nível de Sistema Eletrobrás. A cada vinte segundos, os dados da usina são enviados para Brasília, e os dados do sistema elétrico brasileiro são enviados para a ITAIPU.

Há, ainda, outros softwares que fazem uso das informações concentradas no banco de dados da Estação Central, a fim de processar aplicativos que auxiliam nas atividades do despacho e da operação.

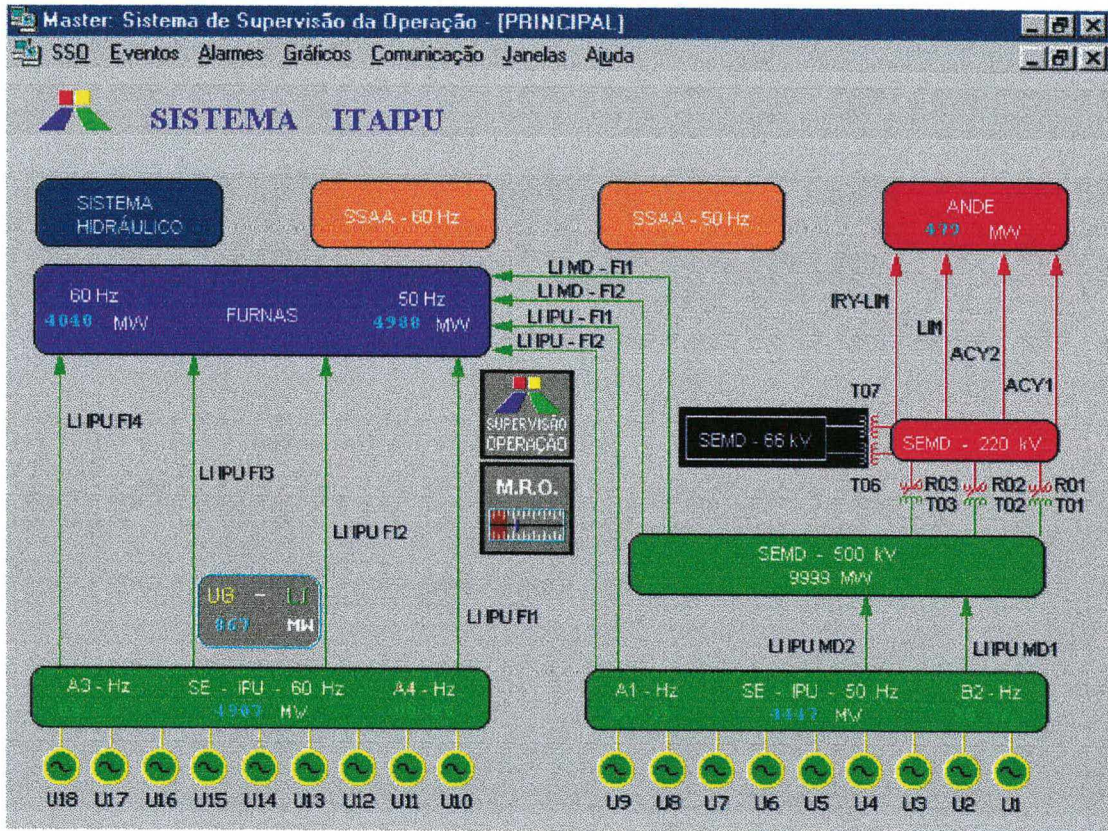


Figura II.4 - Diagrama Unifilar Principal do SSO

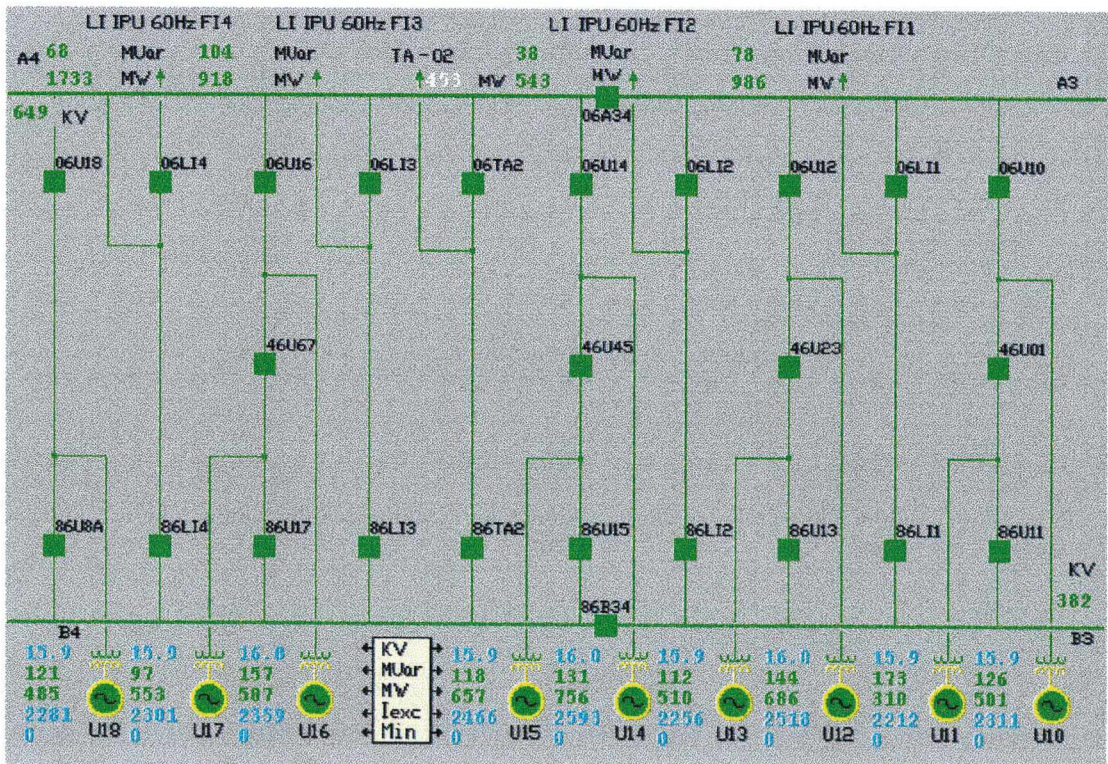


Figura II.5 - Subsistema de 60 Hz do SSO

### II.4.1.2. Arquitetura do SSO

Atualmente, o SSO opera em tempo integral na usina, sendo o responsável pela monitoração de todas as variáveis do sistema. O software aplicativo está em execução em dois locais distintos, na sala de Despacho de Carga e na CCR (Central Control Room). Sua arquitetura é sintetizada na figura II.6.

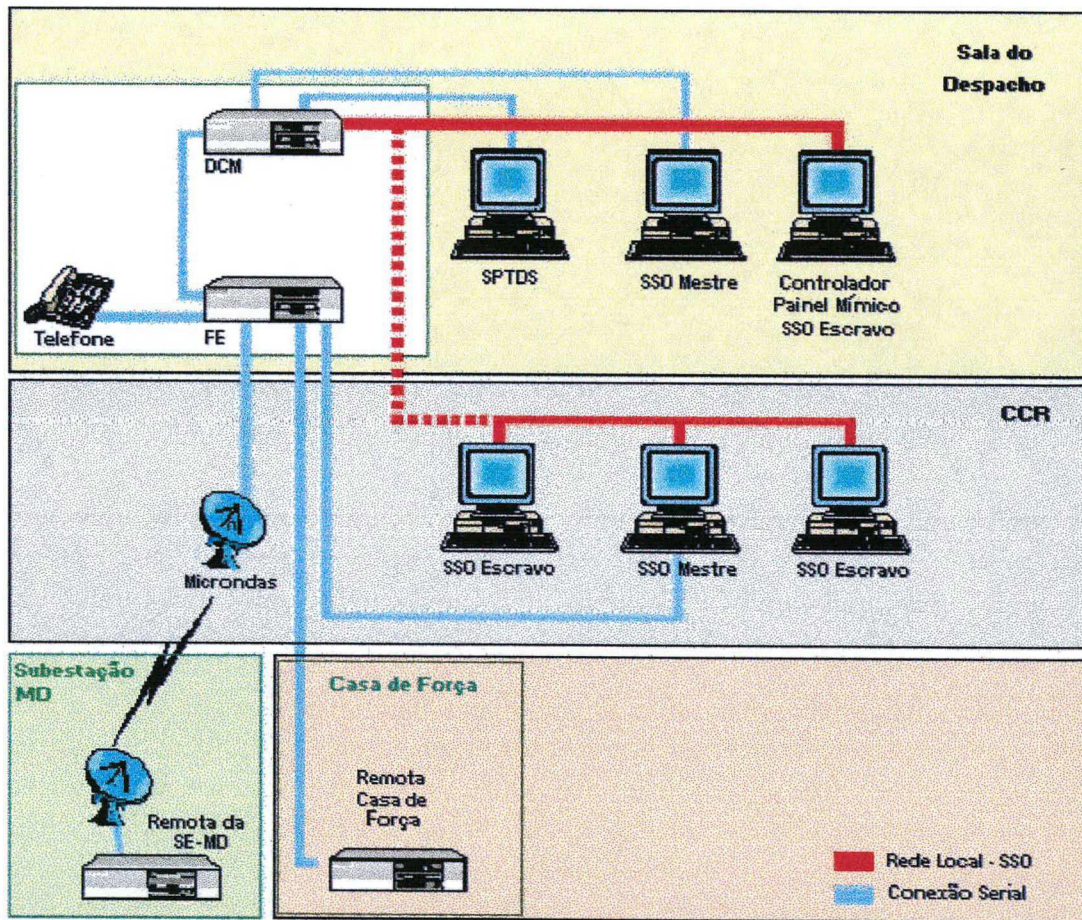


Figura II.6 - Arquitetura do SSO

As principais funcionalidades foram distribuídas pelos três equipamentos da rede, ou seja, a Estação Mestre e seus dois escravos.

A Estação Mestre é responsável pela leitura dos dados vindos do Front-End, tratamento e envio dos mesmos às estações escravas. Este é o componente essencial do SSO, cuja ausência indisponibiliza todas as informações do Sistema de Supervisão.

As estações escravas tratam os dados vindos da estação Mestre e disponibilizam-nos ao operador.

A distribuição descrita mantém uma independência entre as estações, o que justifica a sua escolha na implantação do sistema.

Softwares adicionais, com fins técnicos, cujo processamento seja imprescindível na rede, são instalados nos escravos. Os demais programas, de uso geral, são mantidos em um computador independente, cuja função é justamente atender estas demandas.

A sala do Despacho de Carga apresenta uma configuração um pouco diferente da CCR, já que suas redes não estão completamente interligadas. No Despacho há apenas dois equipamentos em rede: a Estação Mestre e um PC escravo. A Estação Mestre apresenta as mesmas funcionalidades descritas anteriormente. Na Estação Escrava, entretanto, está instalado também o Software do Painel Mímico, que exterioriza os dados do processo no Painel localizado nesta sala, como descrito na seção anterior.

Brevemente, a rede da CCR estará ligada à rede do Despacho, tornando integrada a operação do Despachador de Unidades Geradoras e do SSO, o que exigirá um cuidado maior na operação deste ambiente.

## **II.4.2. Sistema SCADA**

O SCADA (Supervisory Control And Data Acquisition) é um sistema de supervisão e controle capaz de automatizar a aquisição de, aproximadamente, 17500 pontos da Hidrelétrica de ITAIPU Binacional. Monitora o estado de chaves, disjuntores, compressores, bombas, válvulas, e supervisiona variáveis analógicas (tensão, potência, frequência, pressão, nível de abertura do vertedouro, entre outros), sendo capaz de efetuar até 2000 controles [SCADA 95].

O software aplicativo especificado para a ITAIPU consiste, basicamente, de um conjunto de programas que devem implementar os requisitos funcionais das atividades de Programação da Operação, Supervisão, Controle e Análise Pós-Operação, conferindo-lhe

características típicas de um Sistema de Gestão de Energia (Energy Management System).

O Sistema é dividido, logicamente, em quatro módulos básicos:

- *IHM*: a Interface Homem-Máquina engloba todas as funções necessárias para fornecer uma interação amigável entre usuários e sistema computacional;
- *SCADA*: realiza tarefas de aquisição de dados e armazenamento em tempo real, tratamento de alarmes e controle da atuação do processo;
- *EMS*: realiza funções de gerenciamento elétrico, ou seja, estimação de estados e análise de rede;
- *SGBD*: faz o gerenciamento das bases de dados do sistema, sendo responsável por manter a integridade, consistência e segurança das informações.

#### **II.4.2.1. Funções Gerais**

As funções básicas do SCADA, necessárias para a supervisão e o controle em tempo real das grandezas da Hidrelétrica ITAIPU, são as seguintes:

- *Aquisição e Processamento de Dados*: tem como objetivo a coleta de dados em tempo real, periodicamente, via UTRs, e o processamento dos dados adquiridos através da execução de funções básicas, como alarmes e eventos, identificação de mudanças no estado do sistema, conversão para unidades de engenharia, obtenção de valores analógicos calculados, processamento de medidas digitais.
- *Armazenamento de Dados*: armazenamento das informações em bancos de dados específicos do sistema.
- *Comandos de Controle*: transmissão e verificação de comandos às UTRs.
- *Intercâmbio de Informações*: troca de dados e mensagens, em tempo real, necessários para a comunicação com sistemas externos (Eletrobrás, ANDE, FURNAS) e internos.

### II.4.2.2. Sistema Especialista Previsto

O Sistema Especialista previsto para o SCADA consiste em um software aplicativo responsável pela análise de distúrbios para recomposição do sistema.

Esta ferramenta tem como funções analisar, em tempo real, desligamentos ocorridos no sistema por atuação de proteções, avaliar a possibilidade de efetuar restabelecimento imediato e estabelecer a seqüência de manobras para a recomposição (que poderá ser realizada em forma manual pelo operador).

O Sistema SCADA é um software de alto nível, com arquitetura aberta. Através de uma interface homem-máquina, as telas dos diagramas unifilares que compõem o sistema são apresentadas aos operadores, bem como suas funções e facilidades.

Este sistema só será disponibilizado a longo prazo, pois sua versão final ainda não foi adquirida. Desta forma, o SSO continua suprindo algumas das suas funcionalidades.

### II.4.3. Sistema SCC

O SCC é o Sistema de Controle Computadorizado da Subestação da Margem Direita. Este sistema constitui-se em uma ferramenta da área de operação, permitindo a supervisão e o controle dos setores de 500kV, 220kV, 66kV, Serviços Auxiliares e demais expansões do Sistema Elétrico da Usina de ITAIPU Binacional.

O SCC incorpora as funcionalidades características do Sistema SCADA, sendo um sistema de supervisão e controle composto por interfaces de hardware e software.

Seus componentes, que constituem a parte de hardware, são as unidades remotas propriamente ditas. Inclui-se, aqui, placas de aquisição de dados, transdutores, estações de trabalho, impressoras, gerenciador de comunicação, conexões de rede (hubs e cabos), temporizadores, entre outros.



A parte de software possui os seguintes componentes: sistema de supervisão e controle, gráfico de tendências, software de operação local da remota, administrador do banco de dados, e outros utilitários.

Sua arquitetura pode ser vista na figura II.7.

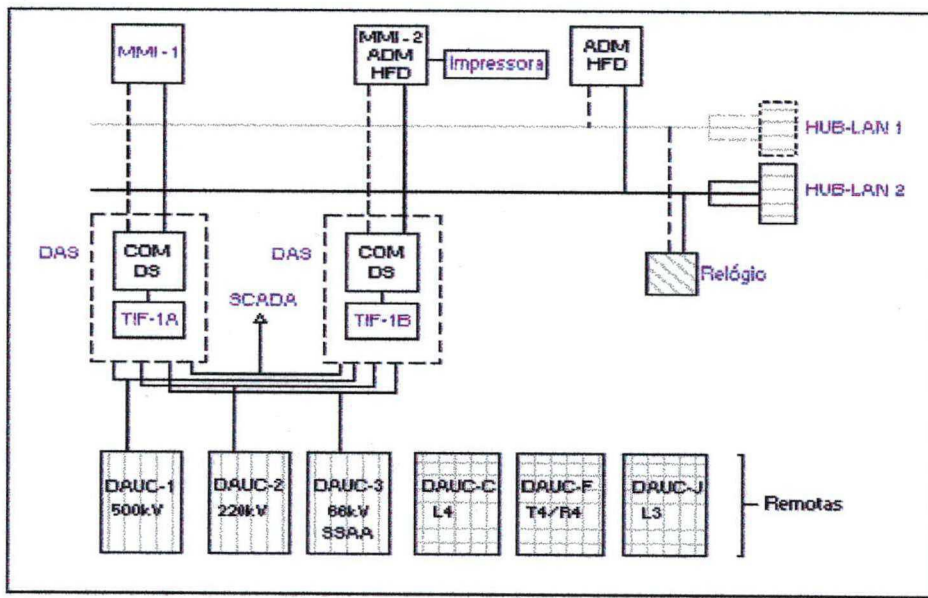


Figura II.7 - Arquitetura do SCC

Todas as suas remotas permitem, por intermédio de um *laptop*, a supervisão e o comando local, nos casos de perda total das estações de trabalho.

Os módulos denominados MMI (Man-Machine Interface) são estações de trabalho que permitem que o operador da Subestação da Margem Direita realize as atividades de supervisão e controle. O sistema de aquisição de dados (DAS - Data Acquisition System) é responsável pela comunicação com as remotas e com o centro de controle externo, sendo composto por um dispositivo de comunicação com as remotas (TIF - Telecontrol Interface) e um servidor de dados (DS - Data Server).

A construção de telas e a parametrização do banco de dados é feita por um software denominado DBA (Data Base Administrator), instalado em uma estação de trabalho, juntamente ao software HFD (Historical and Future Data Management) responsável pelo armazenamento e gerenciamento dos dados históricos.

Na operacionalização do Sistema SCADA, o SCC poderá ser incorporado ao mesmo, constituindo-se como uma unidade remota do sistema.

## **II.5. Ambiente de Apoio à Operação**

Esta seção define *Sistemas Especialistas* e seus requisitos de tempo real, além de detalhar as ferramentas existentes neste contexto, no apoio à operação da Hidrelétrica de ITAIPU.

Objetiva-se, com isto, definir o escopo do ambiente proposto, responsável pela otimização dos processos operativos sob o ponto de vista funcional, cuja diretriz principal é suprir a empresa com uma ferramenta que contribua com a melhoria da qualidade de operação, assegurando a continuidade do suprimento e o funcionamento do processo.

### **II.5.1. Sistemas Especialistas**

Sistemas Especialistas são ambientes computacionais que resolvem problemas de uma maneira bastante parecida com o perito humano. São sistemas com um conhecimento específico sobre campos restritos do saber, devendo caracterizar-se por um conhecimento poderoso e amplo, no limite da perícia, organizado com o objetivo de simplificar a busca da resposta requerida [Waterman 93][Rabuske 95].

O desenvolvimento de um Sistema Especialista consiste em adquirir, de peritos em alguma área, procedimentos, estratégias e regras, com o propósito de resolver um grupo de problemas e adicionar este conhecimento a um sistema computacional.

A arquitetura mais comum destes ambientes é a que envolve regras de produção. Essas regras são simplesmente um conjunto de condições no estilo *se/então*, com a possibilidade de inclusão de conectivos lógicos, relacionando os atributos no escopo do conhecimento [Ignizio 91].

Estruturalmente, um SE é constituído por uma Base de Conhecimento e uma Máquina de Inferência.

A Base de Conhecimento armazena os dados sobre o processo em questão, e é composta por uma Base de Fatos e uma Base de Regras. Na Base de Fatos representam-se as informações sobre o estado atual das variáveis envolvidas no problema a ser resolvido, e na Base de Regras representa-se o conhecimento que o especialista possui sobre o assunto.

A Máquina de Inferência, por sua vez, é responsável por avaliar, encadear e determinar quando e como este conhecimento deve ser utilizado. Ela faz com que a Base de Regras atue sobre a Base de Fatos, inferindo novos dados, para tentar achar uma solução adequada.

Apesar das limitações das máquinas, é possível, hoje, a construção de Sistemas Especialistas com alto grau de desempenho, dependendo da complexidade de sua estrutura e do grau de abrangência desejado.

Para desenvolvê-los, devem ser analisadas muitas ferramentas, sendo ressaltadas as seguintes características para a sua escolha:

- Desenvolvimento orientado a objetos;
- Ambiente de desenvolvimento interativo;
- Linguagem de desenvolvimento de alto nível;
- Habilidades de comunicação.

Além disso, deve levar-se em conta as restrições impostas no projeto, principalmente em sistemas dirigidos a plantas complexas, que necessitem de ações em tempo real.

O tempo é o atributo mais importante nestes tipos de aplicações, pois as distingue das demais, sendo um fator que deve integrar-se à metodologia do projeto, à linguagem de programação utilizada, e às técnicas empregadas para especificação e verificação do sistema.

A possibilidade de representação e de raciocínio sobre eventos e fatos atuais, passados e futuros, bem como o seqüenciamento de suas ocorrências são funcionalidades fundamentais e indispensáveis a exigir de um *Sistema Especialista em Tempo Real* [Beck 90][Sassen 90].

Entretanto, os *Sistemas Especialistas em Tempo Real* não funcionam isoladamente. As restrições temporais impostas sobre os eventos externos ao processo impõem limitações sobre suas ocorrências, fixando o prazo limite de uma resposta (deadline). Em sistemas complexos, pode ser conveniente a utilização de algoritmos para a realização de tarefas especiais, como leitura de informações advindas de outros sistemas e o tratamento de dados entrada/saída.

Este trabalho está centralizado neste contexto. A seguir, apresenta-se os sistemas já existentes, que serviram, entre outros [Wright 86][Tano 88][Liempd 90][Garnousset 91], como linha evolucionária para o desenvolvimento do projeto descrito nos próximos capítulos.

## **II.5.2. Sistema Especialista de 50 Hz - IOI50**

Na Central Hidrelétrica de ITAIPU, o setor de 50 Hz encontra-se interconectado aos sistemas ANDE (Paraguai) e FURNAS (Brasil), apresentando certas peculiaridades que impõem uma série de cuidados do ponto de vista da operação em tempo real.

A Comissão Mista de Operação (CMO) elabora e revisa freqüentemente as Instruções de Operação Interconectada para o setor de 50 Hz, responsáveis pelo estabelecimento de ações e recomendações operativas que devem ser observadas segundo as diversas configurações do sistema.

A figura II.8 mostra um diagrama unifilar simplificado do setor de 50 Hz.

Cada componente introduz sua cota de complexidade à planta, e as instruções de operação formam um corpo volumoso e de difícil tratamento pelos despachantes de carga, especialmente perante casos incomuns [Herreros 96].

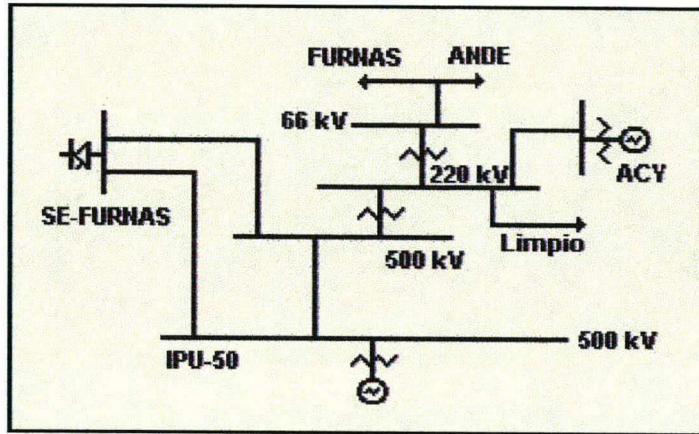


Figura II.8 - Sistema Unifilar de 50 Hz

Assim, levando-se em conta esta complexidade e percebendo-se que as normas das IOIs podem ser expressas formalmente como enunciados lógicos, surgiu a opção do desenvolvimento de um Sistema Especialista como uma possível solução para apoio aos despachantes. A partir desta decisão, foi desenvolvido um ambiente para interpretação das instruções operacionais ANDE/ITAIPU/FURNAS, com o objetivo de melhorar a qualidade e a confiabilidade da operação [Herreros 96].

O IOI50 recebe informações, em tempo real, sobre a configuração do sistema elétrico, disponibilizadas pelo sistema de supervisão da usina; a partir desta informação, são selecionadas as regras aplicáveis e apresentadas as causas de uma eventual anormalidade ocorrida, as ações pertinentes e a instrução de operação recomendada.

A seguir, descreve-se sucintamente as funcionalidades oferecidas por esta ferramenta.

### II.5.2.1. Objetivo e Funcionalidade

Definiu-se como objetivo do Sistema Especialista de 50 Hz prover assistência ao despachante de carga na tomada de decisões rápidas e precisas.

O SSO provê dados ao IOI50, mediante um dispositivo de comunicação denominado DCM (Dispositivo de Comunicação Microprocessado), encarregado de

distribuir dados entre ITAIPU, Eletrobrás e ANDE.

O Sistema Inteligente realiza, via comunicação serial, inferências sobre as informações enviadas pelo SSO, onde, a cada 9 segundos, novos dados são adicionados ao processo.

A Base de Conhecimento contém a totalidade das instruções e recomendações de operação referentes ao setor de 50 Hz, contidas nas IOIs, que implicam ações operativas, como também a experiência dos despachantes.

O IOI50 funciona coexistindo com outras aplicações, ou seja, a Máquina de Inferência recebe dados automáticos com a frequência estabelecida e realiza as inferências pertinentes, mesmo quando a interface gráfica não se encontra ativa. Permite-se, assim, que o usuário realize simulações sem deixar de realizar suas tarefas em tempo real.

### II.5.2.2. Arquitetura do Sistema

O IOI50 é constituído por três componentes principais:

- **Interface Homem-Máquina**, intermediária entre o usuário e os demais componentes. A IHM apresenta dados e resultados ao usuário de forma adequada, e é capaz de deduzir o estado de linhas e barras, a partir do estado de outros componentes do sistema elétrico, realizando uma análise de rede do processo;
- **Módulo de Comunicação**, encarregado de receber os dados do SSO e do usuário, comunicá-los à IHM e ativar a Máquina de Inferência quando ocorrem modificações no estado do sistema físico;
- **Máquina de Inferência**, responsável por percorrer as regras contidas na Base de Conhecimento, utilizando a imagem do sistema físico mantido pelo Módulo de Comunicação.

Para atender as exigências de tempo real, os três componentes do IOI50 devem funcionar de maneira independente e paralela. Requer-se um ambiente multitarefa preemptivo, onde o processador não pode ser monopolizado por uma tarefa, em detrimento de outras.

A fim de garantir o desempenho desejado, optou-se pelas seguintes soluções:

- As diversas tarefas do Módulo de Comunicação foram programadas como *threads* paralelas, algumas independentes e outras coordenadas mediante semáforos;
- Foi desenvolvida uma Máquina de Inferência própria, também sendo uma *thread* paralela ao Módulo de Comunicação, porém ativada por este.

É necessário salientar que a busca de soluções, no IOI50, não caracteriza-se como um mecanismo de inferência propriamente dito (apesar de ser utilizada esta terminologia), pois a varredura das regras que o compõem é realizada apenas como um processamento de condições, sem utilizar nenhum tipo de estratégia de busca ou resolução de conflito.

O IOI50 está em funcionamento desde maio de 1995. Neste período, puderam-se observar alguns aspectos vinculados à sua utilização, como a grande dificuldade em se implantar um Sistema Especialista em tempo real, devido à necessidade de uma Base de Conhecimento correta, já que uma instrução operacional errônea pode ocasionar sérios problemas ao sistema interconectado.

### **II.5.3. Sistema Especialista de 60 Hz - IOI60**

Com o objetivo de manter a integridade do sistema interligado ITAIPU/FURNAS/ELETROSUL durante contingências, foram implantados vários esquemas especiais de proteção, denominados “Esquemas de Controle de Emergência”, alguns objetivando o corte de geração na Usina de ITAIPU, no setor de 60 Hz.

Desta forma, os despachos de carga de ITAIPU e FURNAS são permanentemente

acionados, no sentido de ligar ou desligar esses esquemas, dependendo de condições específicas de operação do sistema interligado. Estas manobras são realizadas por operadores, após solicitação do despacho de carga, e implicam o acionamento de elevado número de combinações de chaves de controle. Além destas manobras, os despachantes da Usina também são encarregados de coordenar todos os procedimentos operativos estabelecidos pelas IOIs, o que exige maior concentração e dispêndio de tempo para que as manobras sejam feitas com confiabilidade [El Khouri 93].

Na busca de soluções práticas viáveis para auxiliar o operador deste sistema complexo, reduzindo a quantidade de decisões a serem tomadas, desenvolveu-se um Sistema Especialista para interpretação das instruções operacionais ITAIPU/FURNAS/Eletrosul - o IOI60. Sua arquitetura e funcionalidades serão detalhadas a seguir.

### **II.5.3.1. Objetivo e Funcionalidade**

O diagrama unifilar do sistema interligado ITAIPU/FURNAS-60 Hz, ilustrado na figura II.2, engloba as informações essenciais abordadas nas IOIs do setor de 60 Hz.

Além das informações sobre os estados de linhas, compensadores séries, reatores de linha e de barra, transformadores, fluxos e geração de ITAIPU, são necessários dados sobre queda bruta, estado do Painel de Corte de Geração e de suas respectivas chaves, período de carga, recebimentos pelo Sul e Sudeste, entre outros.

Assim, o IOI60 objetiva, paralelamente ao IOI50, auxiliar o operador na manipulação destas informações, aumentando a qualidade dos serviços prestados.

Os estados das variáveis são obtidos por entrada manual, ou via Sistema de Supervisão da Operação, que provê dados em tempo real sobre a geração.

Com as vantagens proporcionadas por esta ferramenta (qualidade, confiabilidade), observou-se a necessidade de tornar disponível ao despachante um sistema que estimulasse seu uso. Isto foi obtido através de uma Interface Homem-Máquina amigável, com as



informações dispostas ergonomicamente, fazendo com que o usuário pudesse extrair todo o potencial presente no sistema.

### **II.5.3.2. Arquitetura do Sistema**

A construção do Sistema Especialista de 60 Hz consistiu em formalizar as informações contidas nas IOIs, através de um conjunto de regras de produção que resultaram na formação da Base de Conhecimento. Além disso, utilizou-se como Mecanismo de Inferência a linguagem OPS5, orientada para regras de produção.

Antes de ir direto à edição das regras, foi necessário construir árvores de decisão das informações contidas nas IOIs.

A Base de Regras foi particionada em blocos, a fim de obter-se um projeto modular e, conseqüentemente, minimizar o tempo de processamento. Estes blocos de regras se encadeiam no momento da inferência, sendo que dois blocos consecutivos interagem entre si através da memória de trabalho.

Como o IOI50, o IOI60 é constituído basicamente por três componentes principais:

- **Interface Homem-Máquina** - proporciona três serviços ao despachante: a entrada de dados com a configuração do sistema interligado, a saída de dados com os procedimentos operativos a serem conduzidos e a elaboração do Relatório Diário de Ocorrências.

Os dados sobre as unidades de 60 Hz e a queda bruta chegam via comunicação serial, provenientes do SSO. A IHM embute um pré-processamento dos dados de entrada, verificando a plausibilidade da informação que está sendo fornecida. Isto previne a submissão de configurações inconsistentes ao Sistema Especialista, proporcionando maior robustez à ferramenta. A linguagem usada para desenvolver a Interface Homem-Máquina foi o Turbo Pascal, da Borland.

- **Mecanismo de Inferência** - é o elemento que torna a Memória de Trabalho e a Base de Conhecimento ativas, sendo o seu funcionamento sintetizado nas seguintes etapas:
  1. **Verificação das regras aptas ao disparo** - todas as premissas devem casar com pelo menos um elemento da memória de trabalho, e nenhum elemento da memória de trabalho deve casar com as premissas negativas;
  2. **Seleção de uma das regras** - esta resolução de conflito está baseada principalmente na seleção da regra que referencia o elemento de memória com a marca de tempo mais recente;
  3. **Execução das ações indicadas.**

Estas etapas caracterizam um ciclo de execução. O processo de inferência encerra-se quando não houverem mais regras aptas ao disparo. Neste momento, a memória de trabalho conterá as informações desejadas resultantes da inferência.

- **Base de Conhecimento** - fornece as lógicas do esquema de proteção que devem estar ativas ou inativas para uma dada configuração do sistema interligado ITAIPU/FURNAS 60 Hz; contém, aproximadamente, duzentos e cinquenta regras.

O Sistema Especialista do setor de 60 Hz foi operacionalizado em 1991, sendo submetido a modificações, decorrentes das constantes atualizações das IOIs. Em julho de 1997, este sistema foi substituído por CLPs - Controladores Lógicos Programáveis - visando facilitar o trabalho dos operadores, frente às ações finais do Esquema de Controle de Emergência do Sistema de Transmissão em 750 kV, o que permitiu a automatização destas ações através da aquisição automática de dados em tempo real.

A aceitação do IOI60 por parte do usuário final, além de comprovar a eficiência desta tecnologia, é um estímulo ao início de estudos detalhados no sentido de viabilizar a utilização dos SE em outros domínios de aplicação, tais como tratamento de alarmes, planejamento da operação, treinamento e outros.

## **II.6. Conclusão**

Este capítulo apresentou o complexo da Hidrelétrica de ITAIPU Binacional, destacando os principais sistemas que a compõem.

Aqui, apresentou-se o Sistema de Supervisão da Operação da Usina, destacando os principais módulos que o compõem e a sua importância junto aos operadores do sistema elétrico. Fez-se, também, uma breve descrição do Sistema SCADA, previsto para a supervisão final na Usina, face à necessidade de uma ferramenta de grande porte para apoio à operação, que permita a monitoração de um grande número de variáveis e mantenha a confiabilidade e qualidade dos serviços.

Devido à urgência em evoluir a supervisão e o controle local, otimizando os processos operativos sob o ponto de vista funcional, tem-se a oportunidade de desenvolver um Sistema Especialista capaz de auxiliar o operador da usina a desempenhar suas tarefas, indicando a decisão/ação humana a ser executada.

Assim, com o Sistema ITAIPU entendido claramente, as etapas, descritas nos capítulos seguintes, têm o propósito de apresentar a proposta de um ambiente para construção de Sistemas Especialistas em tempo real.

A partir deste ponto de detalhamento, parte-se para a descrição de aspectos da implementação do R-TESE, onde os elementos descritos são inseridos em uma estrutura modular, baseada em Engenharia de Programas.

## **Capítulo III**

# **Ambiente para Construção de Sistemas Especialistas**

Este capítulo apresenta a proposta de um ambiente para construção de Sistemas Especialistas em tempo real (R-TESE - *Real Time Expert System Environment*), tema principal deste trabalho.

Os sistemas, desenvolvidos a partir desta ferramenta, provêem assistência ao despachante de carga na tomada de decisões rápidas e precisas, diminuindo a complexidade frente a manobras operacionais.

A estrutura, aqui descrita, detalha as funcionalidades e os procedimentos necessários à inferência, enfatizando seus aspectos de projeto.

### **III.1. Introdução**

Nos últimos anos tem crescido o número de trabalhos e pesquisas sobre ferramentas computacionais, em Inteligência Artificial, que procuram capturar e simular o comportamento de especialistas humanos.

Visando uma maior viabilidade econômica na implementação de um Sistema Especialista, e considerando-se que diversos sistemas compartilham uma *máquina de inferência* e outras características comuns, foram criadas ferramentas - *shells* - aptas a realizar o trabalho necessário para transpor um ambiente inteligente para um computador. Essas ferramentas permitem que se represente apenas o conhecimento do especialista, deixando para a *shell* a tarefa de interpretação do conhecimento.

Apresenta-se, aqui, a estrutura do ambiente de construção de Sistemas Especialistas em tempo real R-TESE (*Real Time Expert System Environment*), uma *shell* voltada ao setor de geração e transmissão de energia, que permite o desenvolvimento modular de Bases de Conhecimento através de uma interface de fácil manipulação.

O R-TESE é uma ferramenta de criação de SEs, implementada na linguagem de programação Delphi, visando a plataforma PC nos ambientes Windows 95 e NT.

Além disso, pode-se destacar características inerentes a Sistemas Especialistas, já salientadas no capítulo II (seção II.5.1), como desenvolvimento orientado a objetos, ambiente de desenvolvimento interativo, linguagem de alto nível e habilidades de comunicação.

As restrições temporais impostas ao processo caracterizam-no como um ambiente para desenvolvimento de SEs em tempo real, cujos limites de resposta e peculiaridades do processos serão descritos no decorrer do capítulo.

Os Sistemas Especialistas, implementados a partir do R-TESE, recebem informações de outros sistemas em operação na Usina, sendo capazes de analisar conjuntos de dados, com informações sobre os pontos digitais e analógicos supervisionados, aplicando o raciocínio baseado em regras e apresentando aos usuários desta ferramenta (operadores do sistema elétrico da ITAIPU Binacional) instruções pertinentes para solucionar eventuais problemas ocorridos.

A seguir, são detalhados os processos que o compõem, deixando clara a sua importância para se obter respostas confiáveis, dentro das exigências de manutenibilidade e qualidade exigidas.

## III.2. Especificação dos Requisitos

Os requisitos do R-TESE definem as características básicas deste ambiente. Estes requisitos, funcionais, de desempenho, e de banco de dados, são descritos nos próximos itens.

### III.2.1. Requisitos Funcionais

O R-TESE incorpora funções que definem a abrangência deste produto, determinando o comportamento do software que rege o processo.

Seus requisitos funcionais são os seguintes:

- Apresentação de instruções aos operadores do Sistema ITAIPU. Isto é feito através do envio de pacotes de informações, contendo dados sobre o estado do sistema interligado.
- Disponibilização da Base de Conhecimento em estudo, na forma de regras *se/então*, permitindo a edição das mesmas e a verificação de consistência de seus dados.
- Acesso à Base de Fatos do sistema (variáveis que o compõem), permitindo inclusão de novos atributos e valores.
- Verificação de consistência dos dados provenientes de ambientes externos, através de informações sobre o estado dos mesmos.
- Impressão, para análise, das regras e variáveis que compõem o processo.

### III.2.2. Requisitos de Desempenho

Os requisitos de desempenho envolvem atributos de performance do sistema.

O software deve apresentar uma estrutura eficiente, que minimize seu tempo de resposta, isto é, o intervalo entre a requisição das funções e o resultado final. Assim,

determinados critérios são adotados com o intuito de otimizar seu funcionamento em tempo-real. São eles:

- Recebimento de dados de sistemas externos: o intervalo de tempo, que define a chegada de novos pacotes de informações, não deve exceder três (3) segundos.
- Análise de dados e execução de funções de inferência: o tempo para conclusão destas tarefas deve apresentar periodicidade de três (3) segundos. Neste intervalo, o sistema deve ser capaz de analisar os dados do sistema em estudo (atualização e verificação das variáveis que compõem a memória de trabalho), aplicar o conhecimento baseado em regras, e retornar instruções pertinentes ao operador do processo em questão.

As restrições destacadas estão diretamente relacionadas ao tempo entre a aquisição de grandezas analógicas e digitais nas Unidades Terminais Remotas e a atualização destas no SSO, pois o R-TESE foi desenvolvido e testado a partir de um protótipo direcionado à área de supervisão, tendo como objetivo primordial a geração de um *Sistema Especialista* para suporte a este processo.

Na utilização do R-TESE em outras aplicações, estes parâmetros de tempo podem variar, dependendo das necessidades dos sistemas externos que provêm e/ou recebem dados deste ambiente.

### **III.2.3. Requisitos de Banco de Dados**

A estrutura básica do R-TESE é formada por oito bancos de dados, ilustrados no **Apêndice 1**.

Os Bancos de Dados do R-TESE são arquivos que contêm informações sobre os componentes do sistema elétrico da ITAIPU. Também definem a maneira como são estruturados os atributos que compõem as Bases de Fatos e de Regras do sistema.

Os registros de banco de dados são definidos através do *tipo*, *tamanho* e *significado* dos predicados necessários à estruturação da memória de trabalho do R-TESE, para o funcionamento correto do mecanismo de inferência proposto.

Além dos requisitos aqui descritos, o projeto do sistema deve incluir aspectos como a completa utilização de suas funções e facilidades, consistência e coerência nos resultados apresentados, e flexibilidade para aceitar futuras mudanças e melhorias.

### III.3. Projeto da Arquitetura

O R-TESE foi criado de modo a permitir, ao próprio projetista do conhecimento, a implementação das bases desejadas. Sua arquitetura é apresentada na figura III.1, sendo um ambiente típico para desenvolvimento de Sistemas Especialistas.

Como em SEs típicos, aqueles implementados utilizando o R-TESE constituem-se de uma Base de Fatos, uma Base de Conhecimento e um Mecanismo de Inferência, onde:

- *Base de Conhecimento e Base de Fatos*: armazenam as informações (regras e fatos) que um especialista utiliza, representadas formalmente;
- *Máquina de Inferência*: parte do *Sistema Especialista* responsável pelas deduções sobre a Base de Conhecimento;
- *Editor de Regras*: meio pelo qual a *shell* permite a implementação das bases desejadas.



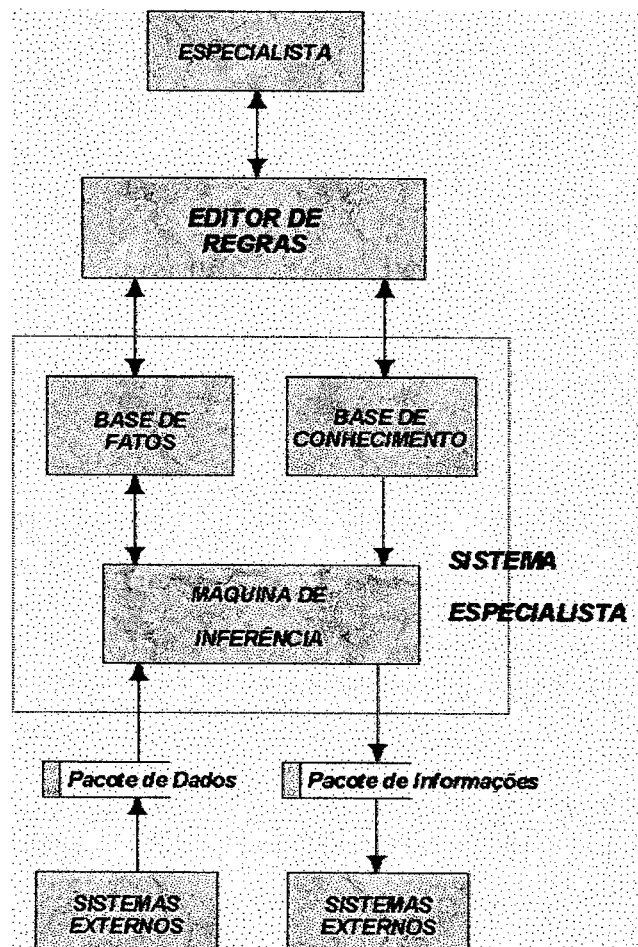


Figura III.1 - Arquitetura Típica do Ambiente Especialista

Sistemas Externos enviam pacotes de dados ao ambiente, atualizando sua Base de Fatos. Com o auxílio de um perito, responsável pela apresentação de lógicas consistentes que caracterizam o projeto em questão, forma-se uma Base de Conhecimento consistente.

O Mecanismo de Inferência, por sua vez, utiliza as informações contidas nas Bases de Fatos e Conhecimento para gerar informações pertinentes ao problema a ser solucionado. Por fim, pacotes contendo estas informações são enviados a Sistemas Externos, que usam-nos para fins específicos.

O fluxo de dados, representativo do R-TESE, pode ser visto na figura III.2, cuja descrição sintetiza seu funcionamento.

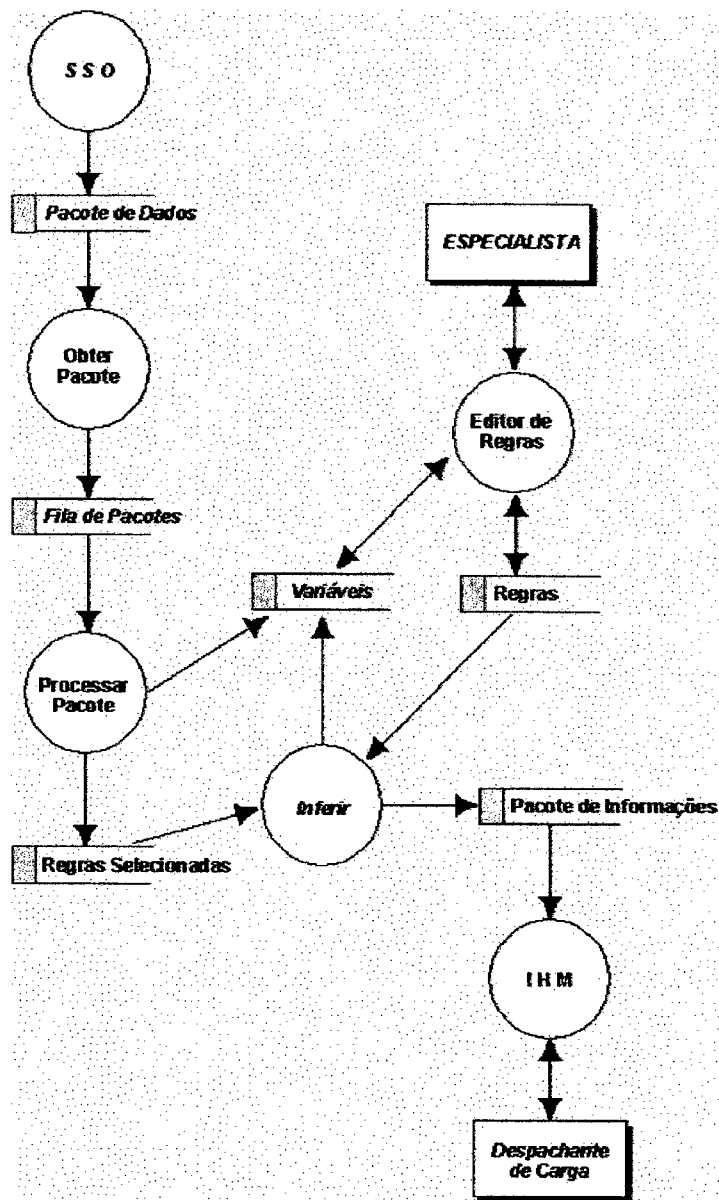


Figura III.2 - Fluxo de Dados do R-TESE

É importante salientar que, como afirmado na seção III.2.2, a arquitetura do R-TESE foi definida visando obter um Sistema Especialista que provesse auxílio ao operador do Sistema Elétrico da Usina de ITAIPU e, sendo assim, direcionada ao Sistema de Supervisão da Operação.

A partir deste ponto, os detalhes de implementação sempre serão descritos com base no funcionamento, em paralelo, destes dois sistemas.

Detalhando-se a figura III.2, o R-TESE executa os seguintes procedimentos:

### 1. *Processamento dos pacotes de dados vindos do SSO, atualizando a Base de Fatos do sistema.*

O SSO envia, a cada 3 segundos, um *pacote de dados* ao R-TESE, contendo informações sobre os pontos analógicos e digitais supervisionados pela ITAIPU.

O R-TESE, através de um mecanismo de *thread*, armazena este pacote em um banco de dados de memória (método *Obter Pacote*). Cada novo pacote disponibilizado ao sistema é inserido em uma *fila de pacotes*.

Outra *thread*, paralela à esta, acessa cada pacote recebido, retirando-o da fila, e disponibilizando seus dados ao sistema. Os dados de cada pacote são processados (método *Processar Pacote*), atualizando a Base de Fatos do sistema, que contém o valor e o estado de todas as variáveis envolvidas na supervisão da operação.

Esta configuração produtor-consumidor garante que não haja perda de pacotes, fazendo com que todos os dados enviados pelo SSO sejam acessados pelo ambiente inteligente. A definição deste modo de funcionamento foi necessária, pois considerando-se alguma interferência de processos externos, ou uma inferência acima do limite de tempo definido (causada por um problema eventual), poderiam ocorrer perdas de pacotes, comprometendo a análise do estado do sistema interligado.

### 2. *Verificação das regras aptas ao disparo.*

As regras aptas ao disparo são aquelas cujas variáveis, relacionadas às suas premissas, sofreram alguma alteração em seu estado ou valor, quando analisadas as informações provenientes do SSO. Estas regras são armazenadas em uma coleção na memória (*Regras Seleccionadas*), obedecendo a ordem crescente de prioridade de cada regra.

### 3. *Processamento das regras seleccionadas.*

São disparadas todas as regras habilitadas. As árvores de objetos, relativas a cada regra seleccionada, são processadas (método *Inferir*). Avalia-se o estado das premissas e

executa-se a ação correspondente.

A interpretação das regras também pode ocasionar a mudança do estado ou valor de outras variáveis que compõem o processo, ativando novas regras e criando, assim, o encadeamento necessário.

Estas etapas caracterizam um ciclo de execução. O processo de inferência encerra-se quando não houverem mais regras aptas ao disparo. Neste momento, a memória de trabalho conterà as informações desejadas, resultantes desta inferência. Estas são estruturadas em *pacotes*, e enviadas a uma Interface Homem-Máquina, na forma de instruções operacionais, disponibilizando ao despachante de carga o acesso às mesmas.

O especialista conta, ainda, com um *Editor de Regras* para a composição das regras pertinentes à Base de Conhecimento do sistema em estudo. Detalhes de sua estrutura são ressaltados no capítulo VI.

Nos próximos itens, far-se-á uma descrição geral das interfaces que compõem o R-TESE, cujos módulos serão detalhados nos capítulos seguintes.

### **III.3.1. Interfaces Internas**

Os processos e os depósitos que compõem o R-TESE estão encapsulados nos seguintes objetos:

- Objeto Gerenciador do R-TESE;
- Objeto Gerenciador de Variáveis;
- Objeto Gerenciador de Regras;
- Objeto Gerenciador de Pacotes.

A descrição detalhada destes objetos sintetiza o funcionamento do R-TESE, apresentando suas características de implementação, que garantem a confiabilidade das informações disponibilizadas.

### **III.3.1.1. Objeto Gerenciador do R-TESE**

O gerenciador do R-TESE é o responsável pela administração da Interface Homem-Máquina, e pelo controle da operação dos demais objetos.

Este elemento sincroniza a ativação dos mecanismos de *thread*, que ativam os métodos apresentados na figura III.2, além executar as funções de inicialização do sistema.

#### **Inicialização dos Objetos**

- *Criação de bancos de dados de memória*

Esta rotina cria coleções de pontos na memória, contendo informações relevantes para o funcionamento do mecanismo de inferência.

Os bancos de dados de memória são relativos à Base de Conhecimento (lógicas operacionais), à Base de Fatos (estrutura das variáveis que compõem o processo) e ao Banco de Dados do SSO, que contém os dados relativos aos pontos analógicos e digitais do sistema interligado.

Permite-se, desta maneira, o acesso a estas estruturas e o processamento de suas informações.

- *Criação das demais estruturas do processo*

Inicialização de todos os outros objetos que compõem o sistema, entre eles: instanciação das classes que definem a estrutura do processo, alocação dinâmica de memória, leitura

de arquivos com dados de inicialização (p.ex.: definição de diretórios de leitura de armazenamento de informações).

Esta função é responsável pela ativação das funcionalidades do R-TESE.

### **Ativação dos Mecanismos de Controle (Threads)**

- Ativação das 3 *threads* que compõem o sistema. Estes mecanismos são responsáveis pela execução das seguintes funções:

#### ***Thread para controle da chegada de novos pacotes de dados ao sistema:***

- Verificação de novos pacotes de dados a serem processados, provenientes de sistemas externos, e armazenamento dos mesmos em listas para processamento;

#### ***Thread para ativação do Mecanismo de Inferência:***

- Processamento dos pacotes de dados, submetendo-os a verificações de validade, e atualização de variáveis relacionadas aos pontos analógicos e digitais analisados;
- Ativação do Mecanismo de Inferência, gerando recomendações operacionais que são enviadas a Sistemas Externos via pacotes de informações;
- Exteriorização de dados funcionais: disponibilização de dados relevantes ao conhecimento do sistema, como número de pacotes processados e tempo de CPU utilizado para o processamento das informações;

#### ***Thread para gerenciamento das variáveis temporizadas:***

- Controle das variáveis temporizadas do sistema: ativação e atualização de temporizadores.

O *Gerenciador do R-TESE* administra a ativação de todos os módulos do sistema, fornecendo a cadência de ações executadas pelo software.

### III.3.1.2. Objeto Gerenciador de Variáveis

O Gerenciador de Variáveis é responsável pelo gerenciamento da memória de trabalho, apresentando uma estrutura que engloba os atributos relativos à Base de Fatos do sistema. A aplicabilidade deste objeto, e o destaque das funcionalidades presentes são descritos no capítulo IV, devido a sua importância na composição da ferramenta especialista.

#### Objetivo

Gerenciar a Base de Fatos do processo, através da inserção de novas variáveis, atribuição de valores, e relacionamento das mesmas com as regras pertinentes ao caso estudado.

#### Estrutura

O objeto *Gerenciador de Variáveis* armazena, em uma lista de objetos, as variáveis que compõem o sistema, agregando suas características, entre elas: identificação, descrição, valor, condições de ativação.

### III.3.1.3. Objeto Gerenciador de Regras

Este objeto é responsável pelo gerenciamento das regras que compõem a Base de Conhecimento do sistema.

O Gerenciador de Regras é o componente chave do R-TESE, pois realiza o processamento das informações relativas ao Sistema Elétrico, ou seja, verificação da relação entre os dados externos e as variáveis da memória de trabalho, ativando as regras, ou lógicas, que auxiliarão o operador na tomada de decisões rápidas e precisas.

Como o Gerenciador de Variáveis, a aplicabilidade deste objeto e o destaque das funcionalidades presentes também são descritos em um capítulo próprio, devido a sua importância estratégica no funcionamento do processo.

### **Objetivo**

Gerenciar as regras pertinentes ao processo, através do processamento das mesmas, obedecendo seqüências de prioridade e ativações em intervalos de tempo.

### **Estrutura**

O objeto *Gerenciador de Regras* armazena, em uma lista de objetos, as regras que compõem a Base de Conhecimento do sistema, em forma de árvores de objetos.

Cada objeto armazenado possui as seguintes características: identificador, premissa e ação associada, tipo (regra comum, mista ou temporizada), prioridade, número de disparos.

## **III.3.1.4. Objeto Gerenciador de Pacotes**

O Gerenciador de Pacotes caracteriza o protocolo de comunicação entre R-TESE e Sistemas Externos, e seu detalhamento é descrito na seção que discute as interfaces externas apresentadas (III.3.3), destacando a estrutura de seus dados e a cadência de suas informações.

### **Objetivo**

Gerenciar os pacotes de dados que chegam e saem do sistema, provendo informações importantes para a ativação do mecanismo de inferência.

### **Estrutura**

O objeto *Gerenciador de Pacotes* monitora dois tipos diferentes de pacotes. Um deles contém dados de entrada, que atualizam a memória de trabalho antes da inferência; o outro envia ações de comando ao usuário do Sistema Elétrico.



### III.3.2. Interface de Programação

Os procedimentos necessários para a criação de uma Base de Conhecimento, na ferramenta apresentada, são ilustrados nesta seção.

Na elaboração de regras, o desenvolvedor deve preocupar-se com o formato das regras suportado pelo R-TESE. Assim, baseando-se nos estudos realizados com as Instruções de Operação (IOIs), mostra-se como é caracterizada a interface de programação utilizada na edição das premissas e ações que compõem suas lógicas operacionais.

As seguintes regras exemplificam as IOIs:

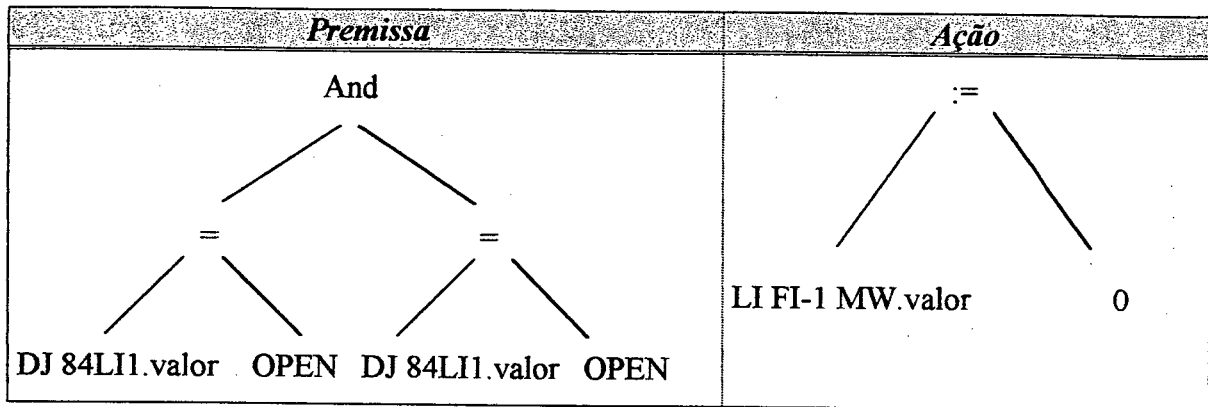
01. **Se** Disjuntor 44L11 e Disjuntor 84LI1 abertos  
**Então** Potência da LI-MD-FI-1 é zero.
  
02. **Se** Tensão na barra 500 kV da GIS acima de 550 kV  
**Então** Tensão está acima da máxima operativa. Providenciar sua redução.
  
03. **Se** Corrente de excitação da *máquina x* acima de 3600 A  
**Então** Reduzir geração da *máquina x*.

As regras são inseridas na Base de Conhecimento na seguinte forma:

01. **Se** (“DJ 44L11.valor” = ABERTO) And  
 (“DJ 84LI1.valor” = ABERTO)  
**Então** (“LI FI-1 MW.valor” := 0)
  
02. **Se** (“IPU-60 BARRA A4-KV.VALOR” > 550)  
**Então** (“Tensão da barra 500 kV da GIS acima da máxima operativa.  
Providenciar redução.” := TRUE)
  
03. **Se** (“ICAMPO UX.valor” > 3600)  
**Então** (“Reduzir geração da máquina x” := TRUE)

A partir da interpretação das regras, o R-TESE gera árvores de objetos.

Tomando-se como exemplo a **regra 1**, as árvores de objetos, correspondentes à premissa e à ação, possuem o seguinte formato:



Estas estruturas ficam armazenadas em bancos de dados de memória, e são processadas pelo Mecanismo de Inferência.

A Base de Conhecimento pode evoluir à medida que se deseja detalhar os procedimentos propostos pelo Sistema Especialista.

### III.3.3. Interfaces Externas

A estratégia de solução do protocolo de comunicação entre o SSO e o R-TESE foi definida através do envio e recebimento de pacotes como forma de comunicação de dados. Baseando-se nesta estratégia, partiu-se para a estruturação e implementação do *Objeto Gerenciador de Pacotes*. A figura III.3 mostra o diagrama de fluxo de dados representativo do processo.

A seguir, descreve-se a estrutura e o comportamento dos objetos que constituem o protocolo de comunicação aqui exposto, ou seja, os objetos responsáveis pelas interfaces SSO/R-TESE e R-TESE/SSO.

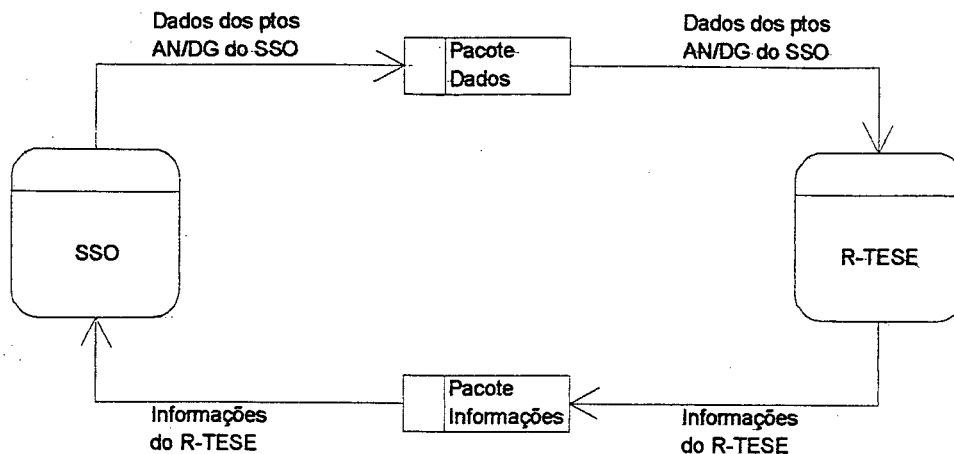


Figura III.3 - Intercâmbio de Informações entre SSO e R-TESE

### III.3.3.1. Objeto responsável pela Interface SSO/R-TESE

Os dados disponibilizados por esta estrutura estão associados, diretamente, ao recebimento dos pacotes enviados pelo SSO. A partir destes pacotes, o usuário tem acesso a informações dos pontos analógicos e digitais supervisionados na Usina, que atualizarão a Base de Fatos do ambiente especialista.

Os pacotes estruturados pelo SSO, e enviados ao R-TESE, são assim definidos:

<i>Cabeçalho</i>	<i>Vetor Pontos Analógicos</i>	<i>Vetor Pontos Digitais</i>
------------------	--------------------------------	------------------------------

#### ***Cabeçalho :***

```

Tipo Cabeçalho = Record
  Número Pontos Analógicos,
  Número Pontos Digitais : Inteiro;
  UTR1,
  UTR2 : Byte;
  Hora,
  Data : TDateTime;
Fim Record.
  
```

O cabeçalho contém o número de pontos analógicos e digitais supervisionados na Usina, o status (operante ou não) das Unidades Terminais Remotas da Casa de Força e da Margem Direita (UTR1 e UTR2), a hora e a data de envio do pacote.

Logo após o cabeçalho, estão localizados os pontos analógicos e digitais, cujas informações estão armazenadas de acordo com os *registros* apresentados a seguir, abrangendo as características necessárias à definição das grandezas supervisionadas.

### **Pontos Analógicos :**

```
Tipo Ponto Analógico = Record
  Origem : Inteiro;
  Status : Byte;
  Valor,
  Limite Superior de Razonabilidade,
  Limite Inferior de Razonabilidade,
  Limite Superior Operacional,
  Limite Inferior Operacional,
  Delta Superior,
  Delta Inferior,
  Limiar : Real;
Fim Record.
```

### **Pontos Digitais :**

```
Tipo Ponto Digital = Record
  Status,
  Valor,
  Valor Default : Byte;
Fim Record.
```

O status destes dois tipos de pontos é definido em um *byte*, estruturado conforme a tabela abaixo:

<b>Bit 0</b>	<b>Modo de Aquisição</b>	<b>Bit 1</b>	<b>Acesso aos Dados do FE</b>
0	Manual	0	Inibido
1	Automático	1	Ativo
<b>Bit 2</b>	<b>Estado do Ponto</b>	<b>Bit 3</b>	<b>Validade do Dado</b>
0	Normal	0	Inválido
1	Alarmado	1	Válido
<b>Bit 4</b>	<b>Reconhecimento do Ponto</b>	<b>Bits 5-6-7</b>	<b>Sempre 0</b>
0	Não Reconhecido		
1	Reconhecido		

Tabela III.1 - Status dos Pontos do SSO

Pacotes de dados, gerados com este modelo, são disponibilizados pelo SSO. Estes pacotes podem ser acessados por outros softwares que fazem uso de seus dados, a fim de processar aplicativos que auxiliam nas atividades do despacho e da operação.

O R-TESE, caracterizando-se como um destes aplicativos, acessa este tipo de pacote, utilizando seus dados para a inferência.

### III.3.3.1.1. Thread Associada

O R-TESE possui um mecanismo de *thread*, responsável pelo controle de chegada dos pacotes descritos anteriormente.

O funcionamento deste módulo pode ser visto na figura III.4.

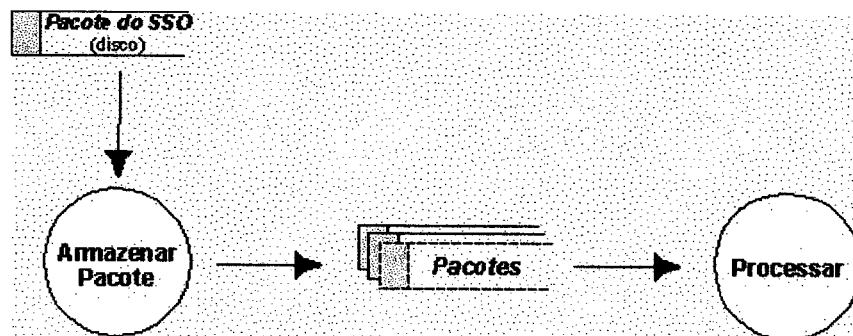


Figura III.4 - Fluxo de Dados do Mecanismo de *Thread*

No detalhamento da figura III.2, fez-se uma descrição geral desta estrutura. Aqui, são ressaltados os aspectos mais relevantes de seu comportamento.

O SSO armazena em disco, a cada 3 segundos, um novo pacote de dados. O mecanismo de *thread* verifica, de 1 em 1 segundo, a existência de um novo pacote de informações, inserindo-o (se disponível) em uma *fila de pacotes*.

Uma outra *thread* (cujo funcionamento é descrito em um capítulo próprio), paralela a esta, retira pacotes da fila, respeitando sua ordem de chegada. Os dados de cada pacote

são processados, atualizando a memória de trabalho do R-TESE e inicializando a inferência. Após processado, o pacote é eliminado.

Deste ponto em diante, o R-TESE possui todas as informações necessárias à inferência. Aspectos de seu funcionamento são destacados quando definidos os comportamentos de seus gerenciadores de regras e variáveis.

### **III.3.3.2. Objeto responsável pela Interface R-TESE/SSO**

Os dados disponibilizados por este módulo estão associados ao envio, a Interfaces Externas, do pacote contendo informações referentes às conclusões obtidas durante o processo de inferência.

Para o funcionamento correto deste protocolo de comunicação, foram definidas algumas estratégias que solucionam o escrita/leitura dos dados que são disponibilizados ao usuário final, descritas no decorrer desta seção.

O R-TESE, ao término de cada inferência, armazena em disco um arquivo (pacote), apresentando uma estrutura que permite a exteriorização das ações resultantes da inferência. Este arquivo pode ser acessado por sistemas que necessitem de suas informações.

No caso em estudo, o SSO é o responsável pela leitura deste arquivo, administrando e disponibilizando suas informações.

Os pacotes gerados pelo R-TESE possuem a seguinte estrutura:

***Vetor de Dados Informativos do Sistema Elétrico***

Cada ponto, no vetor, apresenta-se na forma:

```
Tipo Dado = Record
  Identificador do Dado : Byte;
  Contador : Word;
  Valor : Real;
  Show : Boolean;
Fim Record;
```

O atributo “*ID Dado*” indica se a informação, contida no ponto, corresponde a uma *Ação*, a um *Valor* ou a um *Estado*. Isto determina como esta informação deve ser disponibilizada na IHM, e como deve ser feita a sua atualização no sistema em estudo.

- *ID Dado* = 0 : a mensagem, relativa ao ponto em questão, corresponde a uma *Ação*, e deve ser disponibilizada ao operador em um diagrama apropriado para que este atue sobre o processo.
- *ID Dado* = 1 : corresponde a um *Valor*, cujo conteúdo deve atualizar o ponto analógico indicado.
- *ID Dado* = 2 : corresponde a um *Estado*, cujo conteúdo deve atualizar o ponto digital indicado.

O atributo *Contador* define a seqüência das mensagens, a serem disponibilizadas na IHM. O *Valor* indica o conteúdo, resultante da inferência, atribuído à variável em questão.

O atributo *Show* indica se a informação, contida no pacote, deve ser disponibilizada ao usuário. Este elemento foi utilizado para resolução de dois casos:

- 1) Como opção para disponibilização das mensagens na IHM: o atributo *Show* permite, a cada inferência, a omissão de ações já exteriorizadas na IHM e que tornaram-se sem efeito, e a ilustração de todas as mensagens válidas do pacote.
- 2) Quando o resultado da inferência é incompatível com a configuração imediata do sistema interligado: para uma determinada configuração, pode ser disparada uma regra, cuja ação afirma: “*Número mínimo de reatores em operação deve ser 5*”. Se houver 6 reatores operando, esta informação não é necessária, e o atributo *Show* é enviado, no pacote, como FALSO.

A figura III.5 ilustra como é feita a leitura destes pacotes de dados, possibilitando a sua exteriorização na interface gráfica desejada.

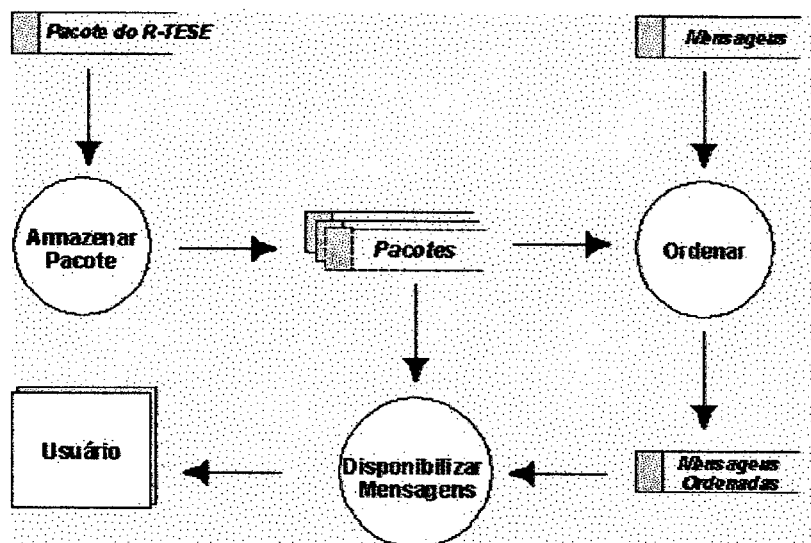


Figura III.5 - Fluxo de Dados do Protocolo de Comunicação

Descrevendo-se a figura acima (III.5), ressalta-se que a IHM Externa utiliza a mesma técnica de controle de recebimento de pacotes utilizada pelo R-TESE (seção III.3.3.1.1).

Além disso, existem dois depósitos na memória (*Pacotes* e *Mensagens*), a partir dos quais são realizadas todas as operações de análise, montagem e armazenamento dos dados, a saber:

- *Pacotes* - armazena os pacotes de dados disponibilizados pelo R-TESE, apresentando a mesma estrutura interna destes arquivos.
- *Mensagens* - disponíveis na forma de variáveis, estão armazenadas no BD que caracteriza a Base de Fatos do sistema. São carregadas para a memória, obedecendo a estrutura deste banco de dados.

A IHM Externa, com informações sobre estas duas estruturas, ordena as mensagens que serão disponibilizadas ao usuário, na seqüência ditada pelo atributo *Contador*, presente no pacote de dados.



A partir das mensagens ordenadas, o sistema analisa os outros atributos (*ID Dado*, *Show*) relacionados às mesmas, e as exterioriza corretamente na interface gráfica.

O exemplo abaixo ilustra o funcionamento do protocolo de comunicação.

*Exemplo da Aplicação:*

Nome	Valor	Pacote	ID
ABRIR DISJUNTOR 86LI4	1	1	0
DJ 86LI4.VALOR	0	2	2
U11 - MW.STATUS	27	3	0
U11 - MW.VALOR	500	4	1

*Mensagens*  
Estrutura em memória, referente à BF do sistema analisado.

	ID Dado	Contador	Valor	Show
1	0	4	1	True
2	2	1	0	True
3	0	2	27	False
4	1	3	500	True

*Pacotes de Dados*

Esta estrutura está armazenada em um vetor, e o acesso a cada ponto é feito através do atributo *Pacote*, que determina o local onde a variável está inserida no pacote de dados.

Entretanto, a localização do ponto não define a seqüência que as mensagens devem ser disponibilizadas na interface gráfica. Para isto, utiliza-se o valor do atributo *Contador*, determinado a cada inferência conforme a ordem de disparo das regras executadas, cujas variáveis correspondem às mensagens do sistema.

Nome	Valor	Show	ID
DJ 86LI4.VALOR	0	True	2
U11 - MW.STATUS	27	False	0
U11 - MW.VALOR	500	True	1
ABRIR DISJUNTOR 86LI4	1	True	0

*Mensagens Ordenadas*

*Mensagens disponibilizadas na IHM<sup>1</sup>:*

<b>DJ 86LI4.VALOR := ABERTO</b>	_____	Corresponde a um Estado (ID = 2)
<b>U11 - MW.VALOR := 500</b>	_____	Corresponde a um Valor (ID = 1)
<b>ABRIR DISJUNTOR 86LI4</b>	_____	Corresponde a uma Ação (ID = 0)

<sup>1</sup> A mensagem "U11 - MW.STATUS" := 27 não é mostrada na IHM, pois seu atributo *Show* é FALSE.

O protocolo, definido no projeto deste ambiente, determinou as estratégias acima descritas. Este baseia-se em uma estrutura estática, onde o envio e o recebimento de pacotes necessitam de informações contidas em bancos de dados fixos. Esta configuração foi escolhida pela eficiência e facilidade de implementação; outros estudos estão sendo realizados para utilização em versões posteriores.

### **III.3.3.3. Bancos de Dados Associados**

Quatro bancos de dados formam a estrutura básica dos sistemas externos, que se comunicam com o R-TESE, sendo eles: Banco de Dados do SSO, Descrição dos Pontos do SSO, Variáveis e Funções.

Estes arquivos, quando carregados na memória do R-TESE, formam uma estrutura que permite a atualização de sua Base de Fatos através da leitura dos pacotes de dados vindos de sistemas externos.

*Banco de Dados do SSO:* é um arquivo que contém informações sobre os componentes do Sistema Elétrico de ITAIPU. É constituído por, aproximadamente, 1030 pontos, referentes às grandezas analógicas e digitais supervisionadas na Usina.

*Descrição dos Pontos do SSO:* contém a descrição resumida de cada ponto do SSO, presente no banco de dados anterior. Estes pontos caracterizam as variáveis que compõem o sistema em estudo. A simplificação de seu nome facilita a edição das regras, pois alguns destes dados possuem nomes muito extensos, e qualquer erro na grafia compromete a validação das regras que os utilizam.

*Variáveis:* indica todas as variáveis que farão parte da memória de trabalho, e estão contidas no SSO.

**Funções:** disponibiliza funções, cuja execução é responsável pela atualização da memória de trabalho com o conteúdo das informações presentes nos pacotes de dados.

A figura III.6 mostra a inter-relação entre estes BDs, cujas estruturas são detalhadas no *Apêndice 1*.

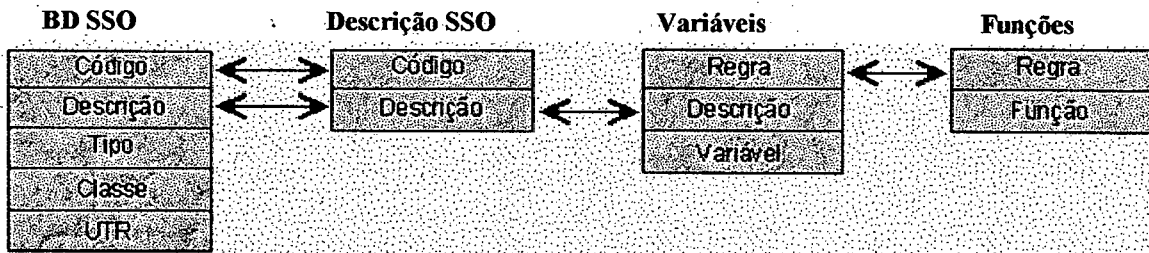
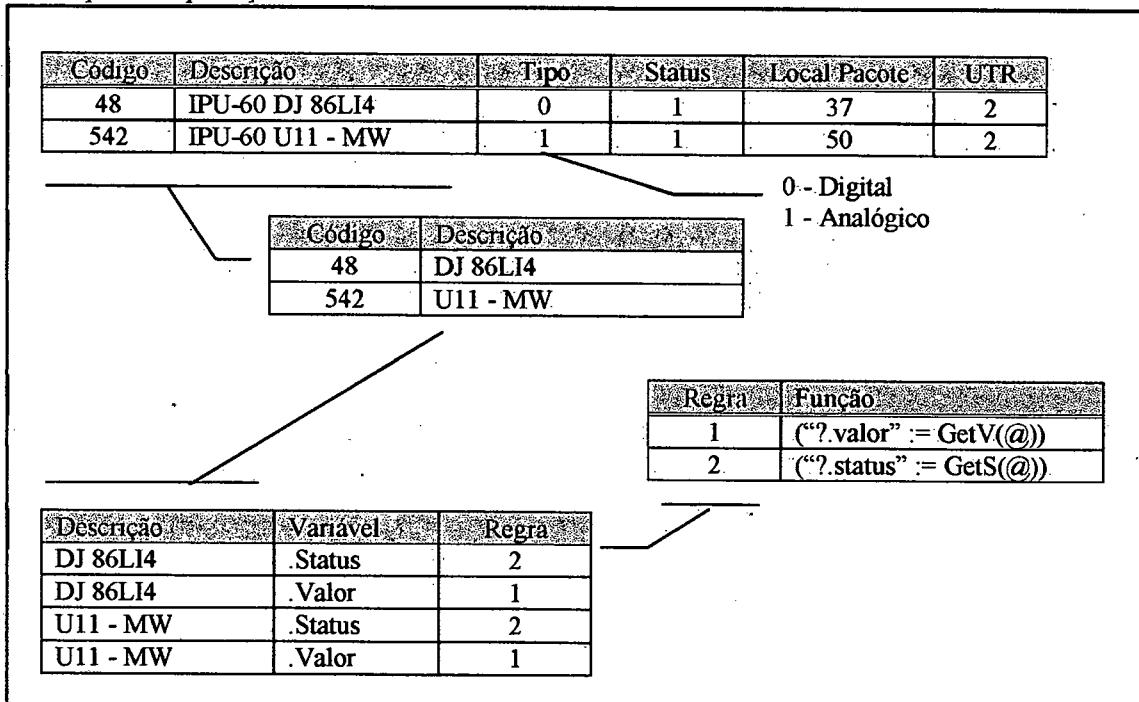


Figura III.6 - Inter-relação entre BDs Externos do R-TESE

O exemplo abaixo demonstra como as informações destes bancos de dados são armazenadas em depósitos na memória, evidenciando sua funcionalidade básica, que é extrair informações de sistemas externos e adicioná-las ao R-TESE.

*Exemplo da Aplicação:*



As tabelas apresentadas são estruturadas na memória do R-TESE, formando um vetor de pontos analógicos e digitais, do seguinte modo:

Código	48	
Nome	DJ 86LI4	
Status	1	
Local Pacote	37	
Tipo	Digital	
Funções	("?.valor" := GetV(@))	("DJ 86LI4.valor" := GetV (37))   ("DJ 86LI4.status" := GetS (37))
	("?.status" := GetS(@))	

As funções<sup>2</sup>, relativas a cada ponto, estão estruturadas em árvores de objetos, e são executadas antes de cada inferência, sendo responsáveis pela aquisição do valor (GetV) e do status (GetS) do mesmo, no local do pacote de dados em que esta informação se encontra.

A atribuição determina a atualização do valor das variáveis que compõem a memória de trabalho, permitindo ao mecanismo de inferência a execução de suas lógicas com novos conjuntos de informações.

### III.4. Estruturação do Ambiente

O diagrama representativo do R-TESE engloba a estrutura dos objetos que o compõem, conforme a figura III.7.

O modelo apresenta um SE integrado por gerenciadores, que administram Bases de Regras e de Fatos, com a seguinte arquitetura interna:

- *Base de Regras*

Uma regra, no R-TESE, envolve premissas e ações, que são tuplas formadas por variáveis que constituem o universo compreendido pela Base de Conhecimento.

<sup>2</sup> Métodos GetV e GetS descritos no Capítulo V, seção V.2.3.

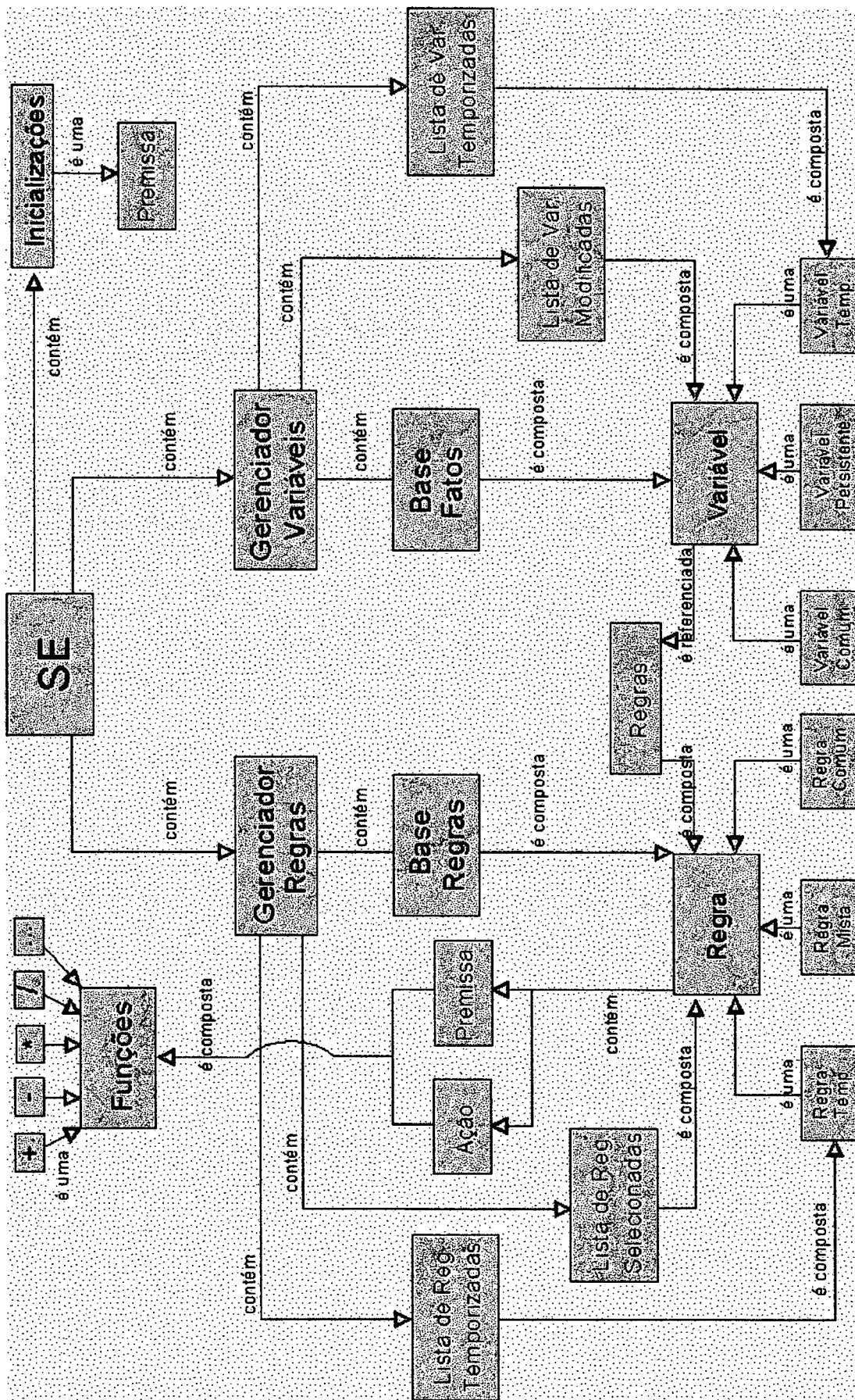


Figura III.7 - Diagrama Representativo do R-TESE

A composição das premissas e das ações é feita através de variáveis, constantes, operadores e funções, como adição, subtração, multiplicação, divisão, seno, cosseno, atribuição, entre outros, como ilustra o exemplo.

*Exemplo da aplicação:*

SE	( "ICAMPO.valor" > 3600 )
ENTÃO	( "Reduzir geração da máquina x" := TRUE )
SE	( A + B + Cos(PI) = 1 )
ENTÃO	( If ((x=5), ("x y" := 3), ("x y" := 4)) )

As regras que compõem o sistema podem ser de três tipos: comum, temporizada e mista.

São classificadas como regras comuns a maioria das regras do sistema, apresentando-se no formato [*Se Premissa / Então Ação*], e possuindo uma prioridade associada. Estas regras são selecionadas durante a resolução de conflito do mecanismo de inferência<sup>3</sup>, e ativadas pelo mesmo.

As regras temporizadas são aquelas disparadas em determinados instantes de tempo, não constituindo o escopo da resolução de conflito. Para isto, possuem um atributo de tempo relacionado às mesmas, que define o intervalo esperado para o seu disparo/redisparo. Estas regras são inseridas em uma lista própria, e a verificação de sua ativação é feita a cada ciclo de inferência.

As regras mistas apresentam-se tanto como comuns quanto como temporizadas. Podem ser ativadas por sua seleção durante o processo de inferência, ou mediante a passagem do tempo associado às mesmas.

#### • *Base de Fatos*

As variáveis que compõem o sistema podem ser de três tipos: comum, persistente e temporizada.

<sup>3</sup> Método descrito no Capítulo V, seção V.3.3.

As variáveis comuns são as que possuem ciclo de vida coincidente com o período de execução do programa, ou seja, todas aquelas pertencentes à memória de trabalho, que não são armazenadas em disco.

As persistentes, ao contrário das comuns, são armazenadas em bancos de dados próprios, após a finalização do aplicativo.

As variáveis temporizadas não são persistentes. São ativadas durante a execução de determinadas regras, e utilizam-se de um temporizador para aguardar o tempo necessário ao disparo de uma ação.

O exemplo a seguir detalha esta estrutura.

*Exemplo da aplicação:*

<b>SE</b>	("U05 MW.valor" > 300) And (t_U05 > 10) And (b = 0)	
<b>ENTÃO</b>	(t_U05 := -1)	
<i>Variável Comum</i>	-	b
<i>Variável Persistente</i>	-	U05 MW.valor
<i>Variável Temporizada</i>	-	t_U05

As variáveis temporizadas são inseridas em uma lista própria, e a verificação de sua ativação é feita por um mecanismo de *thread*, que gerencia esta lista.

A lista de variáveis modificadas engloba todas aquelas cujo valor tenha sido alterado pelo disparo de alguma regra. Esta lista indica as regras que devem ser disparadas, ou seja, aquelas que possuem, em suas premissas, as variáveis que sofreram modificações. As regras selecionadas são inseridas, a cada ciclo de inferência, em uma lista própria e ordenadas conforme sua prioridade. Isto expõe o relacionamento intrínseco regra/variável na estrutura do sistema.

Detalhes do modelo apresentado são descritos em capítulos próprios (IV e V), onde suas particularidades são enfatizadas.

### III.5. Considerações

A estruturação de um ambiente computacional, definido de acordo com os itens acima descritos, tornou-se o tema principal deste estudo.

Como suporte à implementação do software proposto, utiliza-se o ambiente de desenvolvimento Delphi, cuja sintaxe de programação é baseada na linguagem Pascal [Sawyer 86].

O produto é apresentado em Delphi 2 [Cantù 96], que gera código 32 bits e permite o desenvolvimento de programas utilizando o paradigma da orientação a objetos [Pressman 95][Rumbaugh 97].

Esta escolha, além de atender os requisitos desejados (simplicidade na utilização, rapidez de execução, produtividade, modularidade), também facilita a integração do ambiente com o SSO, também desenvolvido nesta linguagem. O sistema é compilável em Delphi 4, visando as próximas aplicações da operação que estão migrando para esta versão.

Para atender aos requisitos de desempenho em tempo real, utilizou-se um computador Pentium 200 MHz, com 32 MBytes de memória RAM. Exige-se que todo o sistema permaneça na memória durante o processamento para que não ocorra *swap* em disco, comprometendo, desta forma, o tempo de resposta.

Testes foram realizados com o propósito de validar seu funcionamento, garantindo a aquisição e a transmissão de informações confiáveis. Cada módulo foi testado separadamente, para verificação da consistência de seus dados, e em conjunto com os outros módulos presentes.

Detalhes destes testes estão especificados no capítulo VI, onde é descrito o protótipo que foi utilizado para validação do processo. A interface gráfica disponível também é ilustrada neste capítulo, junto ao detalhamento do protótipo.

A questão primordial a ser respondida é a seguinte:

- A cada varredura, o R-TESE é capaz de analisar todos os dados recebidos e enviar informações seguras aos sistemas externos?



Para responder esta pergunta, necessita-se de um bom funcionamento dos outros objetos que compõem o R-TESE, como também de uma Base de Conhecimento consistente com o Sistema Elétrico em questão.

Apresenta-se, nos próximos capítulos, a estrutura e o funcionamento dos mesmos, com o propósito de garantir os tempos de resposta exigidos pela supervisão do processo, e disponibilizando um ambiente que auxilia o operador de forma eficaz e com resultados seguros.

### **III.6. Conclusão**

Este capítulo apresentou uma visão geral do R-TESE, contribuindo com a especificação de uma estrutura importante na construção de Sistemas Especialistas.

Além deste aspecto, incorpora-se a vantagem do paradigma orientado a objetos, que tornou viável o desenvolvimento deste ambiente computacional, contribuindo com uma tecnologia que supre as necessidades imediatas da Hidrelétrica de ITAIPU.

O R-TESE atinge seu objetivo como um gerador de Sistemas Especialistas, simplificando o processo de criação dos mesmos.

Sua máquina de inferência, pronta para o uso através de encadeamento para frente, e sua representação de conhecimento baseada em regras de produção, procura deixar o especialista humano mais à vontade, preocupando-se apenas em como modelar seu conhecimento.

Com o desenvolvimento do R-TESE, deu-se início à disponibilização de um ambiente capaz de receber informações reais das grandezas digitais e analógicas da usina, e supervisioná-las de forma segura.

Desta forma, com definições claras sobre o projeto, parte-se para a descrição de seus componentes, detalhando sua arquitetura e processos.

# Capítulo IV

## Gerenciador de Variáveis

O presente capítulo descreve o funcionamento do Gerenciador de Variáveis do R-TESE, parte essencial na definição do comportamento do ambiente em estudo.

Detalha-se sua arquitetura interna, destacando a sua importância na administração da memória de trabalho, contribuindo para a obtenção de uma análise confiável das informações.

### IV.1. Introdução

A memória de trabalho é formada por variáveis definidas dentro do universo do sistema em estudo.

No ambiente R-TESE, as variáveis são administradas por um gerenciador, que executa e sincroniza a ativação dos métodos que o compõem. Controladas por este objeto, funções responsáveis pela atualização da memória de trabalho são executadas, sendo sua estrutura definida de forma a otimizar o funcionamento do aplicativo.

Através deste mecanismo, a base fica organizada, fácil de manter, e o usuário tem acesso a informações sobre as variáveis que compõem o Sistema Elétrico da ITAIPU, podendo interagir com o mesmo através das facilidades disponibilizadas.

As próximas seções evidenciam aspectos de projeto e características internas das variáveis que compõem a memória de trabalho.

## **IV.2. Estrutura das Variáveis**

A Base de Fatos do ambiente R-TESE é formada por variáveis, cuja arquitetura interna é definida de modo a satisfazer os requisitos de projeto impostos pelo sistema, bem como dispor de funcionalidades que garantam o funcionamento adequado de seu mecanismo de inferência.

Aqui, apresenta-se o detalhamento desta estrutura.

### **IV.2.1. Bancos de Dados Associados**

Bancos de dados externos são utilizados na estruturação da memória de trabalho do R-TESE. Estes arquivos estão armazenados em disco, e são inseridos no sistema, criando uma estrutura dinâmica que contém os dados das variáveis que formam o escopo do Sistema Especialista construído a partir deste ambiente.

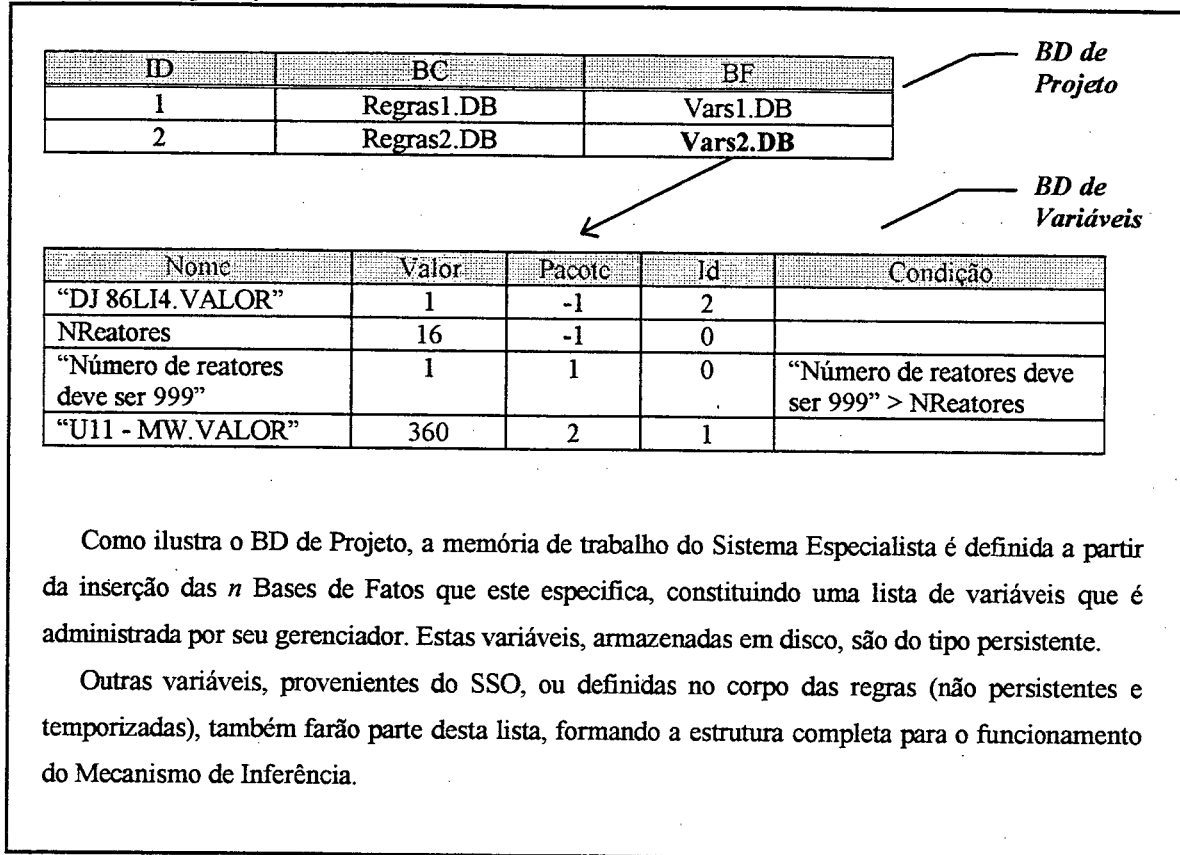
Os BDs ficam assim definidos:

**Bancos de Dados de Variáveis:** são arquivos que contêm informações sobre as variáveis que compõem o sistema em estudo.

**Banco de Dados de Projeto:** contém o nome dos arquivos que armazenam as Bases de Fatos e de Regras do sistema. Seu objetivo é a estruturação de diversas bases em bancos de dados distintos (há sempre um par Base de Fatos/Base de Regras), permitindo a inserção destas estruturas em memória.

A arquitetura destes bancos de dados está detalhada no *Apêndice 1*. O exemplo abaixo ilustra como as informações do BD de variáveis são armazenadas na memória do processo.

*Exemplo da Aplicação:*



O objeto responsável pela estruturação destes arquivos em memória é detalhado a seguir, onde são enfatizados aspectos de sua arquitetura interna.

#### IV.2.2. Arquitetura Interna

Faz-se, aqui, uma descrição detalhada das variáveis que compõem o sistema, de acordo com o diagrama representativo do mesmo, ilustrado no capítulo anterior (figura III.7), e como mostra a figura IV.1.

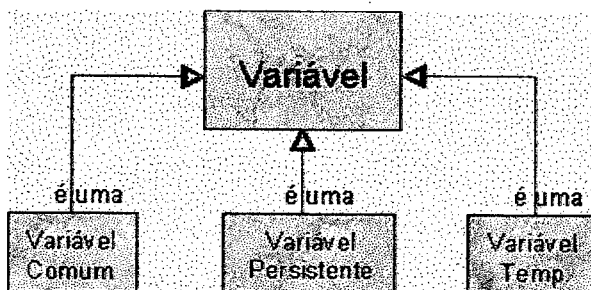


Figura IV.1 - Variáveis do R-TESE

As variáveis que compõem o R-TESE podem ser classificadas em três tipos:

- Variável Comum;
- Variável Persistente;
- Variável Temporizada.

### Variáveis Comuns

As variáveis comuns possuem ciclo de vida coincidente com o período de execução do programa, ou seja, são não persistentes, e não são armazenadas em disco no término da aplicação.

São variáveis que fazem parte da premissa de alguma regra, e criadas na inicialização do ambiente especialista, quando a regra que a utiliza é compilada e inserida em uma estrutura de objetos<sup>1</sup>.

### Variáveis Persistentes

As variáveis persistentes estão armazenadas em bancos de dados próprios, como mostrado na seção anterior. São aquelas que fazem parte do BD de variáveis, sendo inseridas na memória do R-TESE na leitura destes arquivos.

<sup>1</sup> Detalhes deste mecanismo são descritos no capítulo V.

As variáveis que fazem parte da ação de alguma regra também são consideradas persistentes pelo aplicativo (desde que não sejam temporizadas), sendo armazenadas no BD de variáveis associado ao BD de regras que as definiu, obedecendo o BD de projeto do sistema em questão.

### Variáveis Temporizadas

As variáveis temporizadas não são persistentes e são definidas no corpo das regras, sendo reconhecidas pelo sistema pelas iniciais “t\_”. São utilizadas como temporizadores, e estão associadas a variáveis que necessitam aguardar um determinado intervalo de tempo para ativação de alguma ação.

As variáveis temporizadas são inseridas em uma lista própria, permanecendo desativadas até que alguma condição relacionada às suas variáveis seja satisfeita. A verificação de sua ativação é feita por um mecanismo de *thread*, que gerencia esta lista. Detalhes de sua implementação são descritos na seção IV.3.3.

O exemplo abaixo ilustra a diferença entre os três tipos de variáveis.

#### *Exemplo da Aplicação:*

Nome	Valor	Pacote	Id	Condição
“DJ 86LI4.VALOR”	1	-1	2	xxxx
“U11 - MW.VALOR”	360	2	1	xxxx

BD de Variáveis

**SE** (“U11 - MW.VALOR” > 300) And (t\_U11 > 5) And (a = 0)  
**ENTÃO** (b := b + 1)

**SE** (a = 0) And (c < 0)  
**ENTÃO** (b := b + 1)

*Variável Comum* - a, c

*Variável Persistente* - DJ 86LI4.VALOR, U11 - MW.VALOR, b

*Variável Temporizada* - t\_U11

Ressalta-se, ainda, que não há diferença entre maiúsculas e minúsculas, e se o nome da variável contiver espaços em branco ou símbolos, coloca-se a expressão entre aspas.

Os atributos de cada variável que compõe o R-TESE estão encapsulados no **Objeto Variável**, responsável pela definição de uma estrutura que agrega as informações do banco de dados de variáveis. O **Objeto Variável** fica assim definido:

```
Tipo Variável = Classe
Nome : String;
Valor : Real;
Persistente : Boolean;
Local da Variável no Pacote : Inteiro;
Identificador da Variável : Byte;
Identificador do Projeto : Byte;
Condição Associada : Tipo Regra;
Regras Associadas : Lista;
Construtor Create (Nome; Valor; Local Pacote;
                  Id Variável; Id Projeto;
                  Persistente);

Destrutor Done;
Fim Classe.
```

A informação de cada variável está armazenada de acordo com esta classe, cuja arquitetura interna é ilustrada na figura IV.2.

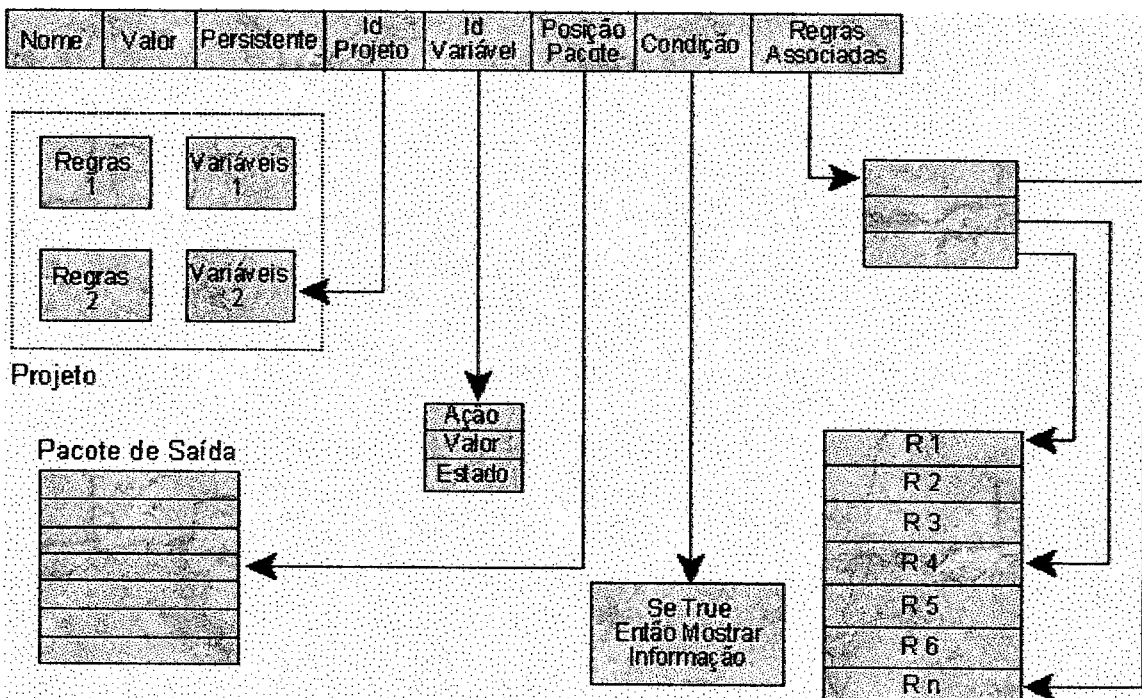


Figura IV.2 - Arquitetura Interna das Variáveis

**Descrição da variável (*Nome*):** armazena o nome da variável, para análise de seus dados no decorrer do processo de inferência.

**Valor:** armazena o valor atual da variável.

**Indicador de Variável Persistente (*Persistente*):** valor *booleano* que indica se a variável é ou não persistente. Este atributo define se os dados desta variável são armazenados em disco.

**Identificador do projeto (*Id Projeto*):** identifica a qual Banco de Dados a variável pertence (se for persistente). Este identificador possibilita o armazenamento dos atributos da variável no BD adequado, quando a execução do Sistema Especialista é finalizada.

**Identificador da variável (*Id Variável*):** este atributo indica se a variável, quando disponibilizada na IHM, corresponderá a uma *Ação*, a um *Valor* ou a um *Estado*:

- ⇒ Se  $Id = 0$ , a variável deve ser mostrada como uma *Ação*, ou seja, na forma de mensagem ao usuário.
- ⇒ Se  $Id = 1$ , a variável corresponde a um *Valor*, que deve atualizar o ponto analógico indicado pela mesma.
- ⇒ Se  $Id = 2$ , a variável corresponde a um *Estado*, que deve atualizar o ponto digital indicado pela mesma.

**Localização da variável no pacote enviado à IHM (*Posição Pacote*):** determina a localização de uma variável no pacote de informações enviado a Interfaces Externas.

Se o valor deste atributo é -1, a variável em questão não faz parte do pacote de dados, ou seja, é apenas uma variável utilizada internamente, durante o processo de inferência.



Qualquer outro valor positivo indica em que local do pacote a variável é armazenada. Esta informação é necessária para que a IHM possa acessar corretamente os dados relativos à mesma, na leitura do pacote de informações gerado pelo mecanismo de inferência.

**Condição para disponibilização da variável na IHM externa (Condição):** o atributo condição está relacionado a cada variável enviada a interfaces externas, ou seja, aquelas cuja posição no pacote de dados é um valor positivo, diferente de -1.

A condição associada à variável apresenta-se na forma de expressão lógica. Deste modo, como toda regra pertencente ao escopo do problema, a condição também é estruturada em uma árvore de objetos.

Ao fim de cada inferência, antes de ser gerado um pacote com informações sobre os resultados da mesma, são processadas todas as condições relativas às variáveis pertencentes ao pacote em questão. Se o resultado de cada processamento é verdadeiro, esta variável deve ser mostrada na IHM; caso contrário, seus dados devem ser desconsiderados.

A IHM tem acesso a esta informação através do atributo *Show* (definido no capítulo III) relativo a cada variável pertencente ao pacote. Este atributo é enviado como TRUE quando a condição relacionada à variável é válida.

Utiliza-se esta técnica para garantir que informações incoerentes com o estado do Sistema Elétrico não sejam disponibilizadas ao usuário do sistema. Por exemplo: o mecanismo de inferência gera uma mensagem (que é armazenada no sistema na forma de variável) que indica o fechamento de um determinado disjuntor. A mesma só poderá ser disponibilizada ao operador se este disjuntor estiver aberto; caso contrário, mesmo sendo uma ação válida disparada, esta não deve ser mostrada na interface gráfica pois sua informação já está atendida.

O exemplo abaixo ilustra o exposto:

*Exemplo da Aplicação:*

Nome	Valor	Pacote	Id	Condição
"Número de reatores"	5	-1	0	
"Número de reatores deve ser 999"	6	1	0	"Número de reatores deve ser 999" > "Número de reatores"

Considerando-se que alguma regra disparou a seguinte ação:

"Número de reatores deve ser 999" := 6

Antes desta mensagem ser enviada à IHM, deve-se verificar se

"Número de reatores deve ser 999" > "Número de reatores"

ou seja, se 6 > 5

Como a premissa é verdadeira, então a mensagem deve ser disponibilizada ao usuário final.

Além disso, a IHM verifica a existência de máscaras (999) nas variáveis, substituindo-as por seus valores. Assim, a mensagem final será: *Número de reatores deve ser 6.*

**Lista de regras relacionadas à variável em questão (*Regras Associadas*):** toda variável armazena, em uma lista, as regras que possuem, em sua premissa, a variável em questão.

A cada ciclo do processo de inferência, as variáveis que mudam de valor têm suas regras habilitadas para o próximo ciclo. Estas regras são inseridas em uma lista única, por ordem de prioridade (atributo associado a cada regra, definido no BD próprio a elas) e administradas pelo seu gerenciador.

*Exemplo da Aplicação:*

**Variável : A → Listas de Regras Associadas: [r1, r2]**

r1: SE ( A > 350 )  
ENTÃO ( B := 2 )

r2: SE ( A + B + Cos(PI) = 1 )  
ENTÃO ( If ((x=5), ("x y":=3), ("x y":=4)) )

Cada variável que compõe o sistema é estruturada na memória do R-TESE, e inserida em uma lista, do seguinte modo:

Nome	A
Valor	350
Persistente	TRUE
Id Projeto	2
Id Variável	0 (Ação)
Posição Pacote	3
Condição	A > 300
Regras Associadas	r1, r2

O *Gerenciador de Variáveis* administra a lista que contém todas as variáveis do R-TESE, além de ativar os métodos que fornecem a cadência de ações que atualizam seus atributos. Esta cadência é proporcionada por um mecanismo de *thread*, descrito no capítulo V, responsável pela ativação do mecanismo de inferência.

A seguir, descreve-se a estrutura na qual as variáveis estão inseridas.

### IV.3. Gerenciador de Variáveis

O Gerenciador de Variáveis é responsável pela administração da memória de trabalho, apresentando uma estrutura que engloba os atributos relativos à Base de Fatos do sistema.

A figura IV.3 ilustra o modelo escolhido para a representação do Gerenciador da Base de Fatos do sistema. No capítulo III, fez-se um resumo de sua arquitetura; aqui são descritos seus dados e métodos.

As variáveis que compõem os Sistemas Especialistas implementados a partir do R-TESE estão encapsuladas no **Objeto Gerenciador de Variáveis**, responsável pela definição de uma estrutura que gerencia a memória de trabalho, durante o processo de inferência.

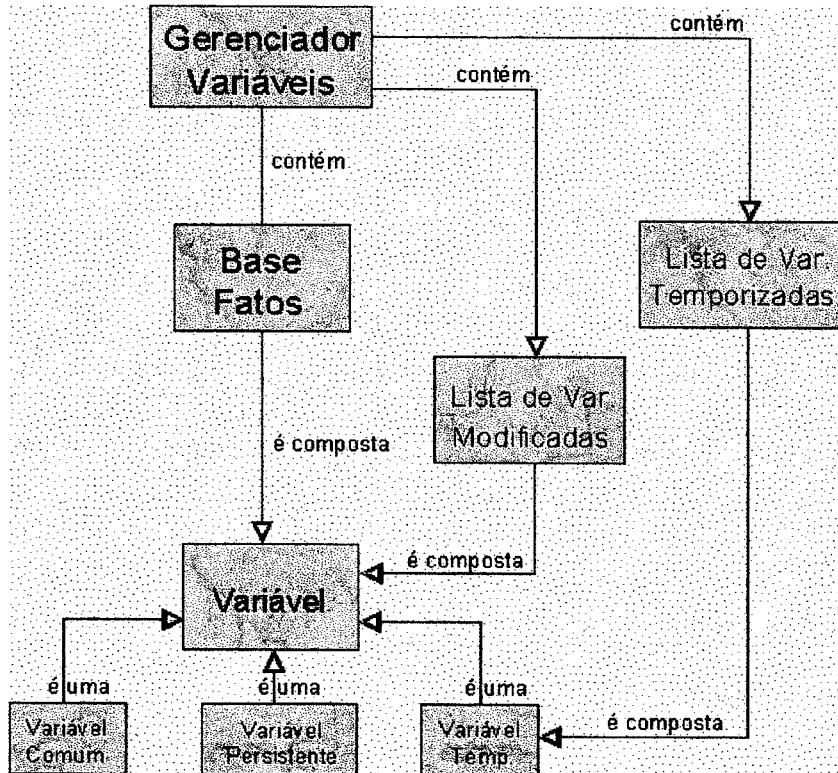


Figura IV.3 - Diagrama Representativo do Gerenciador de Variáveis

Este objeto administra a ativação dos métodos responsáveis pelas operações realizadas sobre as variáveis, como criação, atribuição de valores, e relacionamento das mesmas com as regras pertinentes ao caso estudado.

O *Gerenciador de Variáveis* armazena, em listas de objetos, as variáveis que fazem parte da memória de trabalho do sistema em estudo. É composto por três listas, necessárias para a manipulação desta memória de trabalho, de acordo com a implementação do mecanismo de inferência proposto.

**O Objeto Gerenciador de Variáveis** fica assim definido:

```
Tipo Gerenciador de Variáveis = Classe
Variáveis,
Variáveis Temporizadas,
Variáveis Modificadas : Lista;
Construtor Create;
Função Inserir Variável (Nome; Valor;
                        Local Pacote; Persistente;
                        Id Variável; Id Projeto) : Inteiro;
Função Pegar Variável (Nome) : Variável;
Função Existe (Nome) : Boolean;
Função Pegar Valor (Variável) : Real;
Procedimento Atribuir Valor (Variável; Valor);
Procedimento Carregar Tabela (Tabela; Id Projeto);
Procedimento Gravar Tabela (Tabela; Id Projeto);
Procedimento Criar Variáveis Temporizadas;
Procedimento Iniciar Pacote SSO;
Procedimento Compilar Condição (Tabela);
Procedimento Processar Condição;
Destrutor Done;
Fim Classe;
```

Os principais dados e métodos que formam a estrutura representante do *Gerenciador de Variáveis* são descritos nas próximas seções.

#### IV.3.1. Dados

*TList* é um tipo abstrato que armazena coleções de objetos e oferece uma quantidade de métodos para manipulá-los.

Além de ser uma *classe*, e possuir conceito mais geral que uma lista ou *array* tradicional, este elemento ainda apresenta duas características adicionais: seus *arrays* são polimórficos (são constituídos por objetos de diferentes tipos e tamanhos, encapsulando as características de um array dinâmico, e oferecendo rotinas de ativação de procedimentos ou funções, para cada item da coleção) e possuem tamanho variável.

Com base nas definições do *TList*, as variáveis do R-TESE são estruturadas na memória, de forma a utilizar suas facilidades de implementação.

Os seguintes dados formam a estrutura do *Gerenciador de Variáveis*:

**Dados***Variáveis**Variáveis Temporizadas**Variáveis Modificadas*

Estes dados são *TLists* que armazenam as variáveis que constituem o escopo do sistema, criadas na inicialização do R-TESE.

A lista *Variáveis* armazena todas as variáveis que compõem o Sistema Especialista em estudo, ou seja, as persistentes, não persistentes e temporizadas.

Primeiramente, o *BD de Variáveis* é carregado à memória, e as variáveis persistentes que este armazena são criadas com a estrutura do **Objeto Variável** e inseridas nesta lista. Depois, com a estruturação das regras em árvores de objetos, há a criação e inserção das variáveis não persistentes e temporizadas, que fazem parte do corpo das regras em questão.

A lista *Variáveis Temporizadas* armazena, como o próprio nome diz, as variáveis temporizadas do sistema. Esta lista é administrada por um mecanismo de *thread*, que verifica a ativação e atualização destas variáveis, utilizadas como contadores de tempo para determinação do instante de disparo de alguma ação que as utiliza<sup>2</sup>.

A lista *Variáveis Modificadas* engloba todas aquelas cujo valor tenha sido alterado pelo disparo de alguma regra durante a inferência. Como descrito anteriormente, cada variável possui uma lista de regras associadas à mesma. Assim, as regras associadas a cada variável que sofreu alteração são inseridas em uma lista própria, e habilitadas para o próximo ciclo de inferência. Neste sentido, conclui-se que a lista *Variáveis Modificadas* é a responsável pela indicação das regras que devem ser disparadas.

<sup>2</sup> Detalhes de seu funcionamento são descritos na seção IV.3.3.

### IV.3.2. Métodos

Os principais métodos que compõem o *Objeto Gerenciador de Variáveis* são os seguintes:

#### Métodos

*Carregar Tabela*  
*Gravar Tabela*  
*Criar Variáveis Temporizadas*  
*Inserir Variável*  
*Atribuir Valor*  
*Compilar Condição*  
*Processar Condição*

#### Carregar/Gravar Tabela

O método *Carregar Tabela* armazena, em memória, todos os dados estáticos da Base de Fatos (BD de Variáveis) do sistema, identificando as variáveis inseridas como **persistentes**. No encerramento do ambiente especialista, seus valores são armazenados em disco através da rotina *Gravar Tabela*.

#### Criar Variáveis Temporizadas

Esta rotina é responsável pelo armazenamento, em memória, de todas as variáveis temporizadas do sistema, formando a *Lista Variáveis Temporizadas*, descrita anteriormente.

#### Inserir Variável

Quando o R-TESE percebe uma nova variável, não pertencente ao BD de Variáveis, no corpo de uma regra, esta é inserida na *Lista Variáveis*, com os atributos referentes à mesma.

## Atribuir Valor

A cada atribuição de valor ( $:=$ ), na execução de uma regra, esta rotina é executada.

A seguinte lógica representa seu funcionamento:

**Se** Valor da Variável  $\diamond$  Valor Anterior  
**Então** Atualizar Valor da Variável;  
Inserir Variável na *Lista Variáveis Modificadas*.

Este método é o responsável pela inserção das variáveis que sofreram alterações na *Lista Variáveis Modificadas*, definindo quais regras serão executadas em cada ciclo de inferência.

## Compilar/Processar Condição

A condição associada a uma variável apresenta-se na forma de premissa de uma regra. A rotina *Compilar Condição* estrutura cada premissa em uma árvore de objetos, armazenando-a no atributo Condição (que é do *Tipo Regra*) da variável em questão.

A rotina *Processar Condição* é responsável pela verificação da veracidade de todas as condições relacionadas às variáveis, ao término de cada processo de inferência.

Se uma condição é verdadeira, deve-se indicar este fato no pacote de informações que será exteriorizado à IHM. Isto fará com que a variável (em forma de mensagem) seja disponibilizada ao usuário final. A lógica abaixo sintetiza o funcionamento desta rotina:

Obter Pacote do SSO.  
Realizar Inferência.  
Para cada Variável faça  
    **Se** Condição = TRUE  
        **Então** Indicar no Pacote de Informações que a  
                Variável deve ser disponibilizada na IHM.  
Gravar Pacote de Informações em disco.



O Gerenciador de Variáveis trabalha em conjunto com o Gerenciador de Regras, atualizando os dados da memória de trabalho no processamento das regras através dos métodos apresentados.

### IV.3.3. Thread Associada

O R-TESE possui um mecanismo de *thread*, responsável pelo controle das variáveis temporizadas, cujo funcionamento é sintetizado na figura IV.4.

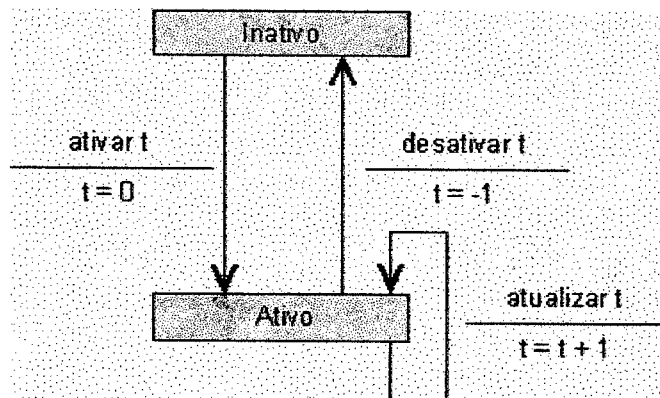


Figura IV.4 - Funcionamento do Mecanismo de *Thread*

As variáveis temporizadas são, como descrito em seções anteriores, inseridas em uma lista própria. Durante o processo de inferência, uma *thread* gerencia estas variáveis, verificando se houve alteração de seu estado *inativo* para *ativo*.

Uma variável temporizada permanece no estado inativo (valor igual a -1) até que uma determinada condição, definida na premissa de uma regra, seja satisfeita. Isto faz com que a variável passe para o estado ativo (valor igual a 0), inicializando um contador de tempo, que incrementa o valor da mesma a cada segundo. A variável volta ao estado inativo quando a condição que a ativou não é mais verdadeira, sendo atribuído a ela o valor -1.

As variáveis temporizadas estão, geralmente, associadas a uma variável não temporizada (persistente ou não), objetivando determinar quanto tempo a mesma satisfaz determinada condição.

Este mecanismo é utilizado, por exemplo, para saber quanto tempo a potência de uma máquina permanece acima de um certo valor. Se este tempo ultrapassar um limite definido, é gerada uma mensagem para o operador do sistema, informando tal ocorrência. Vários outros casos do Setor Elétrico<sup>3</sup> necessitam deste método para controle de seus dados, o que justifica a sua implementação.

As regras que utilizam o método (ou seja, contêm variáveis temporizadas no corpo de suas lógicas) devem ser definidas na Base de Conhecimento do sistema conforme o exemplo abaixo, pertencendo a um mesmo nível de prioridade.

*Exemplo da aplicação:*

<b>SE</b>	("U05 MW.valor" <= 300)	<i>prioridade = 2</i>
<b>ENTÃO</b>	(t_U05 := -1)	
<b>SE</b>	("U05 MW.valor" > 300) And (t_U05 = -1)	<i>prioridade = 2</i>
<b>ENTÃO</b>	(t_U05 := 0)	
<b>SE</b>	("U05 MW.valor" > 300) And (t_U05 > 10)	<i>prioridade = 2</i>
<b>ENTÃO</b>	("Potência da máquina 5 acima do limite permitido" = TRUE)	

A variável  $t_{U05}$  está associada à "U05 MW.valor", que é o valor da potência da máquina 5, atualizado pelo pacote de dados proveniente do SSO. Enquanto este valor estiver abaixo de 300 MW (megawatts),  $t_{U05}$  permanece inativa. Quando a potência atinge um valor maior que 300, a variável temporizada torna-se ativa, tendo seu valor incrementado a cada segundo.

Quando  $t_{U05} = 11$ , ou seja, a potência da máquina 5 está mais de 10 segundos acima do desejado, uma mensagem é enviada ao operador, que tomará as ações necessárias para que a primeira regra seja verdadeira novamente, desabilitando o contador de tempo, ou seja, retornando a variável temporizada ao valor -1.

<sup>3</sup> O detalhamento destes casos não constitui o escopo deste trabalho.

#### IV.4. Considerações

O R-TESE, em uma próxima versão, apresentará uma estrutura de variáveis ampliada. A Classe *TVariável* se dividirá em dois subtipos: variável **comum** e variável **com histórico**.

As variáveis comuns apresentarão a mesma estrutura atual. As *variáveis com histórico* terão seus  $n$  valores (valor atual e  $n-1$  valores relativos a varreduras anteriores) inseridos em listas. Isto permitirá, mediante funções de ajustes de curvas, a avaliação da tendência nos estados das variáveis analisadas, possibilitando que o Sistema Especialista faça previsões de situações emergenciais, informando ao usuário as manobras adequadas para evitá-las.

#### IV.5. Conclusão

Este capítulo apresentou o *Gerenciador de Variáveis* do R-TESE.

Sua implementação disponibiliza um objeto capaz de guardar informações sobre as variáveis pertinentes a cada processo estudado.

Com o entendimento desta estrutura, parte-se para a descrição detalhada do módulo responsável pelo gerenciamento das lógicas operacionais, com o objetivo de compreender o funcionamento completo do processo de inferência.

# Capítulo V

## Gerenciador de Regras

Este capítulo descreve aspectos de implementação do Gerenciador de Regras do R-TESE, que engloba o mecanismo de inferência deste ambiente computacional.

Este objeto gerencia as regras pertinentes ao projeto, gerando informações e sugerindo ações sobre o sistema analisado. A seguir, detalha-se sua arquitetura interna e o comportamento de seus processos.

### V.1. Introdução

A parte mais importante no projeto de um Sistema Especialista é a escolha do método de representação de conhecimento. A linguagem associada ao método escolhido deve ser suficientemente expressiva para permitir a representação do conhecimento a respeito do domínio escolhido, de maneira completa e eficiente [Bittencourt 98].

O conhecimento, no R-TESE, é armazenado na forma de regras de produção, seguindo o padrão *se/então*. Pode-se destacar algumas vantagens do método proposto:

- *Modularidade*: cada regra, por si mesma, pode ser considerada como uma peça de conhecimento independente;

- *Facilidade de edição*: novas regras podem ser acrescentadas e antigas podem ser modificadas com relativa independência;
- *Transparência do sistema*: garante maior legibilidade da Base de Conhecimento.

Portanto, é preciso ter em mente que a modularidade de um sistema baseado nessa arquitetura permite a construção passo-a-passo da Base de Conhecimento, ou seja, é possível realizar vários testes com apenas um subconjunto de regras concluído. Obviamente, sabe-se que menos regras implicam geralmente um menor número de casos abrangidos.

O objeto *Gerenciador de Regras* é o elemento chave do mecanismo de inferência proposto para o R-TESE.

Este elemento, que constitui o escopo deste capítulo, armazena as regras que compõem a Base de Conhecimento do sistema em estudo, sendo também compreendido como o interpretador e escalonador destas regras. Parte-se para o detalhamento desta estrutura, definindo o funcionamento de seu mecanismo de inferência.

## V.2. Estrutura das Regras

A Base de Conhecimento do ambiente R-TESE é formada por regras, cuja arquitetura interna é definida de modo a apresentar atributos e funcionalidades que garantam o funcionamento adequado de seu mecanismo de inferência.

Uma regra de produção típica, no R-TESE, envolve as seguintes estruturas:

- *Premissa*: é formada por variáveis, constantes, operadores e funções, que constituem o universo abrangido pela Base de Conhecimento.
- *Ação*: é análoga à premissa, sendo que sua operação implícita é sempre a atribuição.

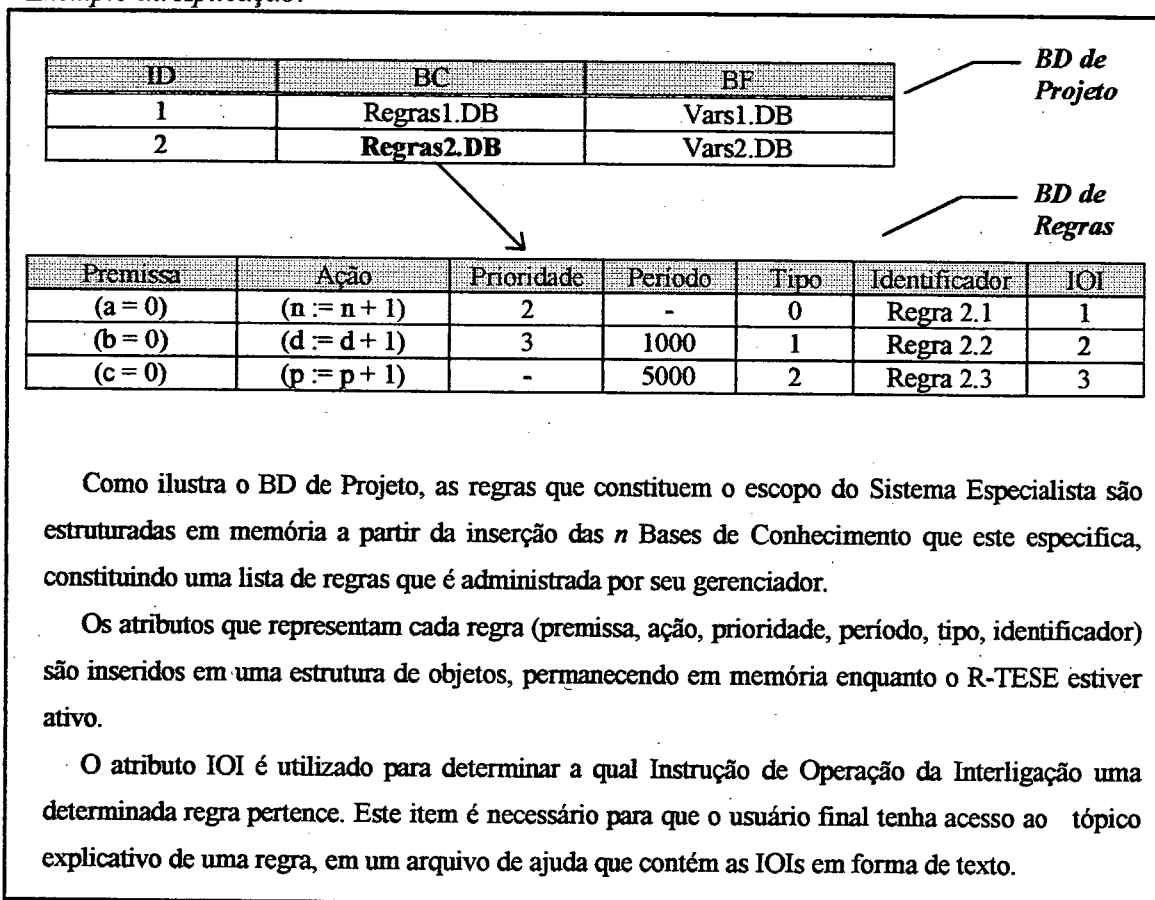
Aqui, apresenta-se o detalhamento desta estrutura.

### V.2.1. Bancos de Dados Associados

A Base de Conhecimento do R-TESE fica armazenada em disco (BD de Regras), em bancos de dados estruturados de forma a suportar os atributos relativos a cada regra, cuja arquitetura é apresentada no *Apêndice 1*.

O exemplo abaixo ilustra como as informações do BD de regras são armazenadas na memória do processo.

*Exemplo da Aplicação:*



O objeto responsável pela estruturação destes arquivos em memória é detalhado a seguir, onde são enfatizados aspectos de sua arquitetura interna.

## V.2.2. Arquitetura Interna

Faz-se, aqui, uma descrição detalhada dos tipos de regras que compõem o sistema, de acordo com o diagrama representativo, ilustrado no capítulo III (figura III.7). A figura V.1 destaca os itens relacionados a seu modelo de regras.

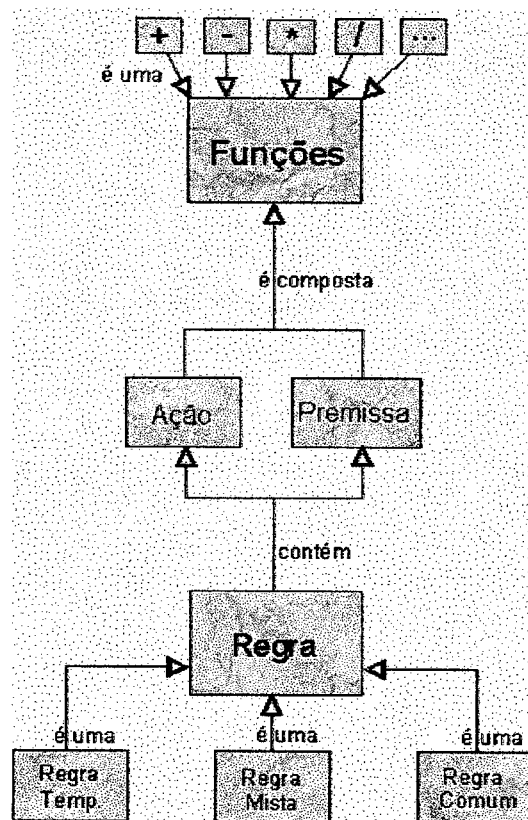


Figura V.1 - Regras do R-TESE

As regras apresentadas no R-TESE podem ser classificadas em três tipos:

- Regra Comum;
- Regra Mista;
- Regra Temporizada.

O tipo da regra é determinado na edição da mesma, ficando armazenado em um banco de dados que contém as informações pertinentes às regras em estudo.

### **Regras Comuns**

As regras comuns constituem a maioria das regras do sistema.

Apresentam-se, como as demais, no formato [*Se Premissa / Então Ação*]; possuem uma prioridade associada, que determinará sua ordem de ativação durante o processamento do mecanismo de inferência.

Estas regras estão inseridas em *Listas de Regras*, relacionadas às variáveis que compõem suas premissas. Assim, durante a resolução de conflito, as variáveis que mudam de valor determinam quais regras comuns estão habilitadas para processamento em um novo ciclo de inferência.

### **Regras Temporizadas**

As regras temporizadas possuem um atributo de tempo relacionado às mesmas, determinado em sua edição. Este atributo é definido em milisegundos.

Quando a diferença entre a última ativação de uma regra e o instante atual é maior ou igual ao tempo determinado, esta regra é habilitada, sendo processada pelo mecanismo de inferência.

As regras temporizadas não constituem o escopo da resolução de conflito, não apresentando uma prioridade associada. São inseridas em uma lista própria, percorrida a cada ciclo de inferência para verificação da passagem do tempo relacionado às mesmas, que definirá o instante do seu processamento.

### **Regras Mistas**

As regras mistas apresentam-se como comuns e temporizadas.

Estas regras possuem atributos de tempo e de prioridade associados às mesmas. Desta forma, podem ser ativadas na resolução de conflito, ou mediante a passagem do tempo associado.

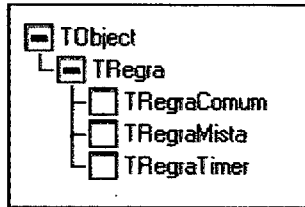


As regras são armazenadas em listas de objetos, e construídas a partir da interpretação de sua premissa e ação. Além da premissa e ação, cada regra é composta por vários atributos, que definem o funcionamento correto do mecanismo de inferência.

Os atributos de cada regra estão encapsulados no **Objeto Regra**, responsável pela definição de uma estrutura que agrega as informações do banco de dados de regras do sistema. O **Objeto Regra** fica assim definido:

```
Tipo Regra = Classe
  Premissa,
  Ação      : Tipo Elemento;
  Identificador da Regra : String;
  Tipo,
  Prioridade,
  Período,
  Número de Disparos,
  Último Disparo,
  Identificador do Projeto : Inteiro;
  Erro Premissa,
  Erro Ação      : Boolean;
  Construtor Create (Id Regra; IdProjeto; Tipo; Tokens);
  Procedimento Compile (Premissa; Ação);
  Procedimento Compile Inicialização (Start);
  Função Processar Regra : String;
  Função Processar Regra Temporizada : String;
  Destrutor Done;
Privado
  É String,
  É Premissa,
  É Ação : Boolean;
  Tokens : Lista;
  Lista de Operadores : Lista;
  Procedimento Interpretar (Início; Fim; Regra);
  Procedimento Adicionar Premissa (Regra);
  Função Identificar Meio (Início; Fim) : Inteiro;
  Procedimento Identificar Token (S);
  Procedimento Agregar Token;
  Função TokenToList (Regra) : Boolean;
  Função Parênteses OK : Boolean;
  Procedimento StringToToken (Linhas);
  Função Verificar Erro (Regra) : Boolean;
Fim Classe.
```

Este objeto é responsável pela definição de uma estrutura que agrega as informações de cada regra pertencente à Base de Conhecimento. A árvore hierárquica, ilustrada na figura V.2, indica a estruturação dos objetos que compõem este elemento.

Figura V.2 - Hierarquia do Objeto *Regra*

Seus descendentes (Regra Comum, Regra Mista, Regra Timer) possuem rotinas próprias, com características particulares de cada componente, que diferenciam o modo de processamento de cada regra durante a inferência, conforme seu tipo associado. A seguir, são definidas as características destes elementos, destacando os métodos implementados.

Os dados que compõem o **Objeto Regra** são descritos conforme sua arquitetura interna, ilustrada na figura V.3.

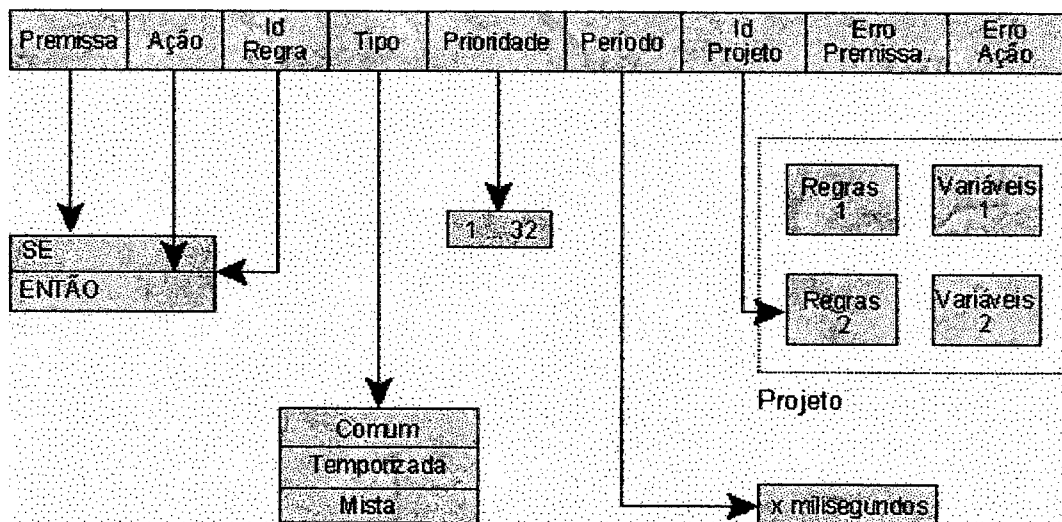


Figura V.3 - Arquitetura Interna das Regras

**Premissa:** armazena a árvore de objetos representativa da premissa de uma regra.

A estrutura de cada premissa segue o seguinte modelo:

- *variável:* é um dos elementos que compõem a Base de Fatos do sistema.

- **operador/função:** é qualquer associação (atribuição, comparação) de um valor a uma variável numérica.
- **valor:** é um atributo correspondente à variável em uso, armazenado na estrutura de objeto que a compreende.

**Ação:** armazena a árvore de objetos representativa da ação de uma regra. A estrutura de cada ação segue o modelo análogo ao da premissa.

**Identificador da Regra (*Id Regra*):** armazena o nome da regra como, por exemplo, *Regra 4.4.1*. Este atributo identifica a regra, na edição da mesma, contribuindo para uma construção adequada da Base de Conhecimento. Além disso, também auxilia a análise de seu processamento no decorrer do processo de inferência, pois uma função auxiliar do R-TESE disponibiliza a seqüência de regras disparadas a cada ciclo, utilizando-se, para isso, do identificador das mesmas.

**Tipo:** indica se a regra é comum (0), mista (1) ou temporizada (2). Este atributo é definido estaticamente, ficando armazenado em disco, no BD relativo à regra em questão.

**Prioridade:** indica a prioridade associada à regra, podendo assumir valores de 1 a 32. Uma regra só possui prioridade se for comum ou mista, pois fará parte do conjunto de conflito definido pelo mecanismo de inferência, que aninha as regras pela prioridade associada.

Quanto maior a prioridade de uma regra, menor seu valor numérico associado. Assim, uma regra com prioridade máxima armazena, neste campo, o valor 1.

**Período:** indica o período associado a uma regra. Uma regra só possui este atributo se for mista ou temporizada, sendo habilitada após a passagem do intervalo de tempo associado. O período é indicado em milisegundos, e fica armazenado em disco, junto à regra correspondente.

**Identificador do Projeto (*Id Projeto*):** identifica o BD de Regras que armazena a regra em questão. Este identificador possibilita o armazenamento dos atributos de cada regra no BD

adequado, além de definir a que BD as variáveis que compõem sua premissa e/ou ação estão associadas.

**Erro relacionado à Premissa (*Erro Premissa*):** valor *booleano* que indica se houve erro na estruturação da premissa de uma regra, quando construída sua árvore de objetos.

**Erro relacionado à Ação (*Erro Ação*):** valor *booleano* indica se houve erro na estruturação da ação de uma regra, quando construída sua árvore de objetos.

Cada regra, que compõe a Base de Conhecimento, é estruturada na memória do R-TESE, e inserida em uma lista. Por exemplo:

*Regra 1.2: Se* (A = 0)  
*Então* (B := B + 1)

Premissa	(A = 0)
Ação	(B := B + 1)
Id Regra	Regra 1.2
Tipo	0 (Comum)
Prioridade	1 (Máxima)
Período	-
Id Projeto	2
Erro Premissa	False
Erro Ação	False

Além dos atributos descritos, várias rotinas compõem o **Objeto Regra**. As ações, pertinentes a ele, são responsáveis pela estruturação das regras em árvores de objetos. Os principais métodos que fazem parte de sua arquitetura interna são os seguintes:

#### **Métodos**

*Compile*  
*Processar Regra*  
*Processar Regra Temporizada*  
*Interpretar*

O método *Compile* é o responsável pela leitura das regras contidas nos BDs relativos à Base de Conhecimento, e transformá-las em uma estrutura de objetos, em forma de árvore.

Esta rotina executa, através da chamada de outros métodos que compõem o **Objeto Regra**, o desmembramento das premissas e ações em *tokens*, a verificação do uso adequado de parênteses no corpo da regra, a verificação de erros na sua sintaxe e, finalmente, a interpretação da mesma (através do método *Interpretar*), gerando uma estrutura em árvore que é armazenada em memória e permite a execução da regra durante o processo de inferência.

O *Apêndice 4* descreve, sucintamente, o mecanismo de transformação de um conjunto de caracteres, na forma de premissa ou ação, em uma árvore de objetos.

Ressalta-se, ainda, os métodos de processamento das regras, responsáveis pela execução das regras do tipo *Comum* (*Processar Regra*), ou do tipo *Mista e Temporizada* (*Processar Regra Temporizada*). Estes são invocados durante a inferência, e analisam as árvores de objetos relativas às premissas e às ações, atualizando a memória de trabalho do sistema.

A seguir, são descritas as funções externas aceitas pelo R-TESE, na construção do corpo das lógicas que compõem sua Base de Conhecimento.

### V.2.3. Funções Externas

Quando se utiliza variáveis, atribui-se a elas valores extraídos de fórmulas matemáticas, ou compara-se as mesmas a valores equivalentes.

Para a criação de regras, deve-se saber como inserir uma sentença contendo variáveis numéricas associadas a seus valores.

No R-TESE, um objeto denominado **Elemento** é responsável pela definição de uma estrutura que agrega funções e operadores matemáticos, disponibilizando os métodos responsáveis pela estruturação das premissas e ações associadas a cada regra.

Este objeto agrega as características comuns aos demais componentes de sua árvore hierárquica, ilustrados na figura V.4.

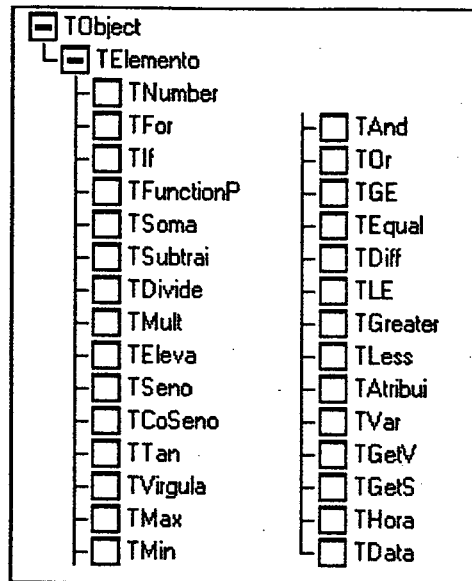


Figura V.4 - Funções Externas Associadas

A estrutura do **Objeto Elemento** fica assim definida:

```

Tipo Elemento = Classe
Operação : String;
Left,
Right : Tipo Elemento;
Erro : Inteiro;
Construtor Create;
Função Resultado : Real;
Função Processa : Real;
Procedimento Verificar Erro;
Destrutor Done;
Fim Classe.
  
```

Estrutura Recursiva

Seus descendentes diferem entre si pelo modo de processamento (*Função Processa*) de seus dados, ou seja, cada um executa o método correspondente à operação que deve ser realizada sobre um determinado conjunto de dados.

O exemplo mostra duas rotinas de processamento diferenciadas, uma para o **Objeto Atribui**, e outra para o **Objeto Soma**.

*Processamento do Objeto Atribui:*

Se (Existem Ramos Direito e Esquerdo na Árvore de Objetos) Então  
Se (Ramo Esquerdo é uma Variável) Então  
Atribuir Valor do Ramo Direito à Variável em questão;  
Senão  
Erro no Processamento.

*Processamento do Objeto Soma:*

Se (Existem Ramos Direito e Esquerdo na Árvore de Objetos) Então  
Somar Valor do Ramo Esquerdo ao Valor do Ramo Direito;  
Senão  
Erro no Processamento.

Este objeto define como as regras do R-TESE devem ser inseridas na Base de Conhecimento, ou seja, quais os operadores e funções válidos, que podem ser escritos em forma de lógicas (premissa e ação).

Para se construir uma lógica, deve-se seguir as seguintes diretrizes:

- Podem ser utilizados operadores matemáticos e funções, como: + (adição), - (subtração), \* (multiplicação), / (divisão), + (sinal unário indicando que um número é positivo), - (sinal unário indicando que um número é negativo), Min(a, b) (retorna o menor valor entre a e b), Max(a, b) (retorna o maior valor entre a e b), entre outros.
- Variáveis podem ser utilizadas nas expressões:
  - ⇒ Para atribuir uma função a uma variável, faz-se:  $a := \text{Min}(b, c)$ ;
  - ⇒ A diferença entre maiúsculas e minúsculas não é considerada;
  - ⇒ Se o nome da variável contiver espaços em branco ou símbolos, coloca-se a expressão entre aspas.
- É possível utilizar parênteses e aninhar funções.

Os seguintes operadores e funções são permitidos para a definição dos predicados que compõem a Base de Conhecimento:

Operador	Descrição	Exemplo	
>=	Maior ou igual que	A := 12 >= 15	resposta = 0 (false)
<=	Menor ou igual que	A := 12 <= 15	resposta = 1 (true)
◇	Diferença	A := 12 ◇ 15	resposta = 1 (true)
+	Soma	A := 10 + 15	resposta = 25
-	Subtração	A := 10 - 15	resposta = -5
*	Multiplicação	A := 10 * 15	resposta = 150
/	Divisão	A := 15 / 10	resposta = 1.5
^	Exponencial	A := 2^3	resposta = 8
**	Exponencial	A := 2**3	resposta = 8
>	Maior que	A := 2 > 2	resposta = 0 (false)
<	Menor que	A := 2 < 3	resposta = 1 (true)
=	Igual a	A := 2 = 3	resposta = 0 (false)
:=	Atribuição	A := 100,	resposta → A = 100
AND	Condição "E"	(5=5) AND (3=2)	resposta = 0 (false)
OR	Condição "OU"	(5=5) OR (3=2)	resposta = 1 (true)
FOR	Repetição	For ( (I:=0),(I<10),(I:=I+1),I ) For ( ((A:=0),(I:=0)), (A<=10), (I:=I+A),(A:=A+1)),I )	
SIN	Seno	A := SIN (90)	resposta = 1
COS	Coseno	A := COS (180)	resposta = -1
TAN	Tangente	A := TAN (30)	resposta = 0,577
,	Separador	ver FOR	
MAX	Maior de <i>n</i> números	A := MAX (1,2,3)	resposta = 3
MIN	Menor de <i>n</i> números	A := MIN (1,2,3)	resposta = 1
GetV	Lê o valor de um ponto na memória	A := GetV (200)	resposta = valor do ponto 200 no pacote vindo do SSO
GetS	Lê o status de um ponto na memória	A := GetS (200)	resposta = status do ponto 200 no pacote vindo do SSO
IF	Condição	If((Condição), (Ação), (Ação Contrária))	
DATA	Retorna o dia da semana	A := DATA	resposta = dia da semana
HORA	Retorna a hora atual	A := HORA	resposta = hora atual
VAR	Indicativo de variável	A := 5	A → variável
NUMBER	Indicativo de valor numérico	A := 5	5 → número
FUNÇÃO P	Retorna o valor atual da variável desejada	P(A) P(5)	resposta = valor de A resposta = 5



As operações e funções permitidas foram definidas conforme a necessidade na estruturação de regras que constituem o escopo do Sistema Elétrico da ITAIPU Binacional.

Como particularidade, salienta-se a implementação da função *IF*, definida no formato [ *Se / Então / Senão* ], utilizada para a estruturação de tabelas de dados em forma de regras (exemplos detalhados estão presentes no capítulo VI). Tem-se, ainda, as funções *HORA* e *DATA*, necessárias à construção de regras que necessitam da hora e da data atual para a verificação da validação de suas informações.

A seguir, define-se o funcionamento do *Gerenciador de Regras*, que administra os objetos apresentados até aqui.

Este gerenciador suporta as regras do R-TESE, além de definir os métodos que inicializam o mecanismo de inferência proposto, fornecendo a cadência para o processamento das regras pertinentes ao caso em estudo.

Sua estrutura é descrita, evidenciando detalhes de projeto, como estratégia de busca, resolução de conflito e funcionamento do mecanismo de inferência propriamente dito.

### **V.3. Gerenciador de Regras**

Foram descritas as estruturas e comportamentos do *Objeto Regra*, sendo este a base para projetar o próximo nível de abstração: o Gerenciador de Regras, responsável pela administração das regras que compõem o Sistema Especialista em estudo.

A figura V.5 ilustra o modelo escolhido para a representação do Gerenciador de Regras. No capítulo III, fez-se um resumo de sua arquitetura; aqui são descritos seus dados e métodos.

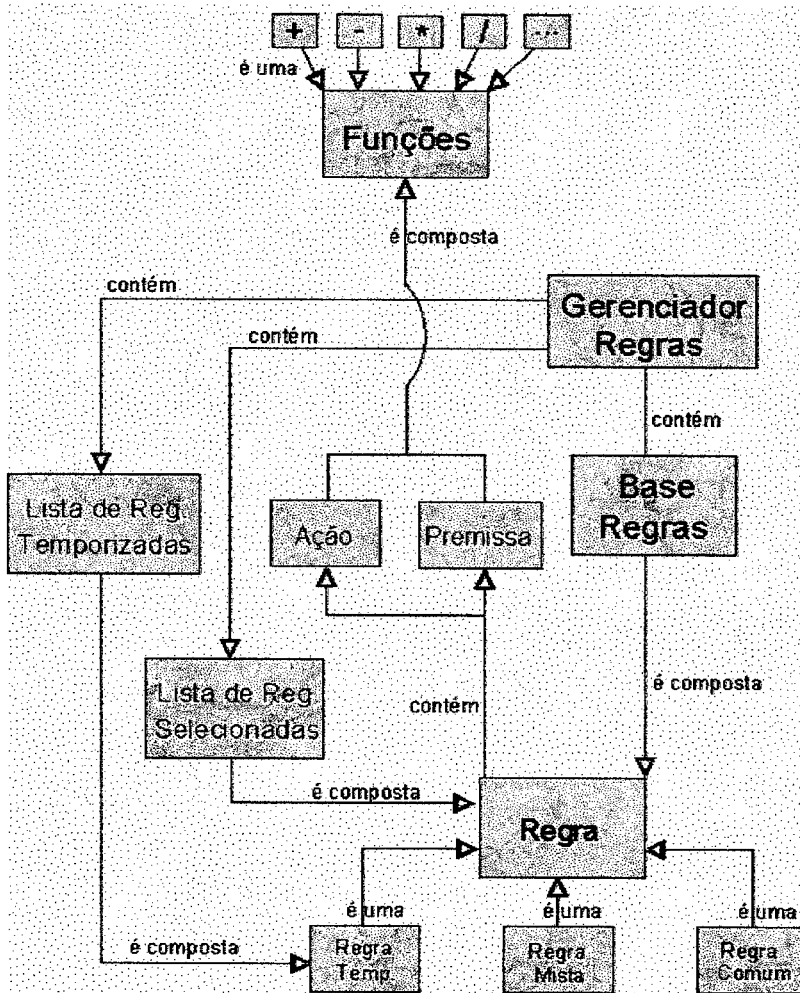


Figura V.5 - Diagrama Representativo do Gerenciador de Regras

O Objeto Gerenciador de Regras fica assim definido:

```

Tipo Gerenciador de Regras = Classe
Start : Tipo Regra;
Regras,
Regras Temporizadas : Lista;
Regras Seleccionadas : Array[1..NListas] De Listas;
Construtor Create;
Destrutor Done;
Procedimento Carregar Tabela (Tabela; Id Projeto);
Função Processar : String;
Procedimento Iniciar MI (StartStr);
Público
Tokens : Lista;
Função Processar Regra (Id Regra) : String;
Procedimento Desenhar Regra (Formulário; Id Regra);
Fim Classe.
  
```

As regras que compõem os Sistemas Especialistas implementados a partir do R-TESE estão encapsuladas no **Objeto Gerenciador de Regras**, responsável pela definição de uma estrutura que executa o processamento destas regras, durante a inferência.

O *Gerenciador de Regras* armazena, em listas de objetos, as regras que fazem parte do sistema em estudo. Estas listas são necessárias para o funcionamento do mecanismo de inferência. O comportamento do MI proposto no R-TESE é dividido em ciclos, onde cada ciclo consiste das seguintes operações:

- Seleção das regras aptas ao disparo,
- Resolução de conflito,
- Execução das regras.

Estas estratégias estão detalhadas nos próximos itens, onde são destacadas as características relacionadas às suas funcionalidades.

### **V.3.1. Modo de Raciocínio**

O modo de raciocínio utilizado no R-TESE é o encadeamento progressivo, ou encadeamento à frente.

A parte esquerda da regra é comparada com a descrição da situação atual das variáveis que a compõem, contidas na memória de trabalho. As regras que satisfazem esta descrição têm sua parte direita executada, significando, em geral, a introdução de novos fatos na memória de trabalho.

### **V.3.2. Estratégia de Busca**

A estratégia de busca é a responsável por guiar a pesquisa do mecanismo de inferência na memória de trabalho e na base de regras. Esta estratégia, conhecida como busca em espaço de estados, foi determinada com o intuito de minimizar o tempo de resposta do motor de inferência, devido às exigências de resposta em tempo real.

Ao processar-se os dados dos pacotes provenientes do SSO, atualiza-se a memória de trabalho do mecanismo inteligente.

Compara-se, então, as novas informações com seus valores anteriores, selecionando-se as regras compostas pelas variáveis que tiveram suas características alteradas. As regras são agrupadas conforme sua prioridade estática, e na ordem alfabética das variáveis que compõem o sistema em estudo.

### **V.3.3. Resolução de Conflito**

Ao finalizar o processo de busca, tem-se um conjunto de regras que satisfazem à situação atual do problema, caracterizando o conjunto de conflito do ciclo de inferência em questão.

Determina-se, então, quais as regras serão realmente executadas e em que ordem, passo denominado como “resolução de conflito”.

A ordem de disparo das regras é determinada por uma prioridade estática, delegada à regra na edição da mesma. Assim, a cada ciclo de inferência, ocorrem os seguintes passos:

- 1) São analisadas as regras aptas ao disparo, ou seja, aquelas cujas premissas estão relacionadas às variáveis que mudaram de valor;
- 2) Estas regras são separadas por níveis de prioridade e guardadas em listas próprias;
- 3) As regras são executadas;
- 4) Volta-se ao passo 1.

O R-TESE define dois métodos para a resolução de conflito. São eles:

#### **Método 1**

*Resolução de conflito:* são executadas todas as regras do conjunto de conflito em ordem de prioridade, evoluindo do nível de maior para o de menor prioridade.

## Método 2

*Resolução de conflito*: são executadas apenas as regras do conjunto de conflito que tenham prioridade máxima. Assim, a cada passagem pelo nível de mais alta prioridade, determina-se um novo conjunto de conflito para verificar se alguma regra com prioridade máxima foi habilitada.

Este método apresenta uma política “injusta”, ou seja, uma regra com prioridade alta pode permanecer no conjunto de conflito (caso as variáveis relacionadas à regra sempre mudem de valor), impedindo a execução das demais, e fazendo, conseqüentemente, com que o mecanismo de inferência entre em *loop*.

Entretanto, considera-se como verdade que a construção de uma Base de Conhecimento adequada é um problema dos especialistas e desenvolvedores que utilizarão o sistema em estudo. Uma Base de Conhecimento sólida, com dados bem estruturados é fundamental para o funcionamento correto de um sistema inteligente.

A partir daqui, são definidos os principais dados e métodos que formam a estrutura representante do *Gerenciador de Regras*, e sintetizam o seu funcionamento.

### V.3.4. Dados

Com base nas definições do *TList*, detalhado no capítulo anterior, as regras do R-TESE também são estruturadas na memória, de forma a utilizar as facilidades de implementação desta *classe*.

Os seguintes dados formam a estrutura do *Gerenciador de Regras*:

#### **Dados**

*Regras*  
*Regras Temporizadas*  
*Regras Seleccionadas*  
*Start*

*Regras*, *Regras Temporizadas* e *Regras Seleccionadas* são *TLists* que armazenam as regras que constituem o escopo da Base de Conhecimento.

A lista *Regras* armazena todas as regras que compõem o Sistema Especialista em estudo, ou seja, as comuns, as mistas e as temporizadas. Esta lista reúne informações gerais sobre a regra, como tipo, prioridade, período, sendo seu conteúdo (premissa e ação) guardado na forma de árvores de objetos.

Primeiramente, o *BD de Regras* é carregado à memória, e as regras que o compõem são criadas com a estrutura do **Objeto Regra**, compiladas em árvores de objetos e inseridas nesta lista. Também são inseridas na lista *Regras Temporizadas*, se forem do tipo que esta suporta.

Assim, a lista *Regras Temporizadas* armazena as regras mistas e/ou temporizadas do sistema. Estas regras são disparadas durante a inferência, em intervalos de tempo pré-determinados, relativos a cada regra.

A lista *Regras Seleccionadas* é uma coleção (*array*) de 32 listas de regras (dos tipos comum ou mista). Cada lista armazena as regras com prioridade condizente à sua posição no *array*, ou seja, uma regra com prioridade máxima (igual a 1) será inserida na primeira lista deste *array*; uma regra com prioridade igual a 2 será inserida na segunda lista, e assim por diante.

Como descrito no capítulo IV, a lista *Variáveis Modificadas* armazena aquelas variáveis que sofreram modificações em seu valor durante o ciclo de inferência. As regras associadas a estas variáveis são inseridas na lista *Regras Seleccionadas*, definindo aquelas que devem ser disparadas em um próximo ciclo.

O atributo *Start* apresenta-se na forma de premissa de uma regra, sendo utilizado na inicialização de dados do sistema. Assim, antecedendo a inferência, são executadas todas as condições relativas à inicialização, ou seja, é processada a árvore de objetos representativa do *Start*.

Utiliza-se esta técnica por dois motivos:

- ⇒ Para inicializar certos dados do Sistema Especialista. Por exemplo: a determinação do número de reatores presentes no tronco Foz-Tijuco Preto, que é um dado fixo (igual a 16) e é obrigatório para o cálculo do número de reatores desligados, já que o valor do número de reatores ligados é disponibilizado pelo SSO.
- ⇒ Para garantir a mudança de valor de determinadas variáveis, forçando-as a entrar na lista *Variáveis Modificadas* e, em consequência, habilitando as regras relacionadas às mesmas. Isto é feito para formar um primeiro conjunto de regras aptas ao disparo, para início do processo de inferência. Sem esta técnica, os primeiros dados vindos do SSO seriam descartados, comprometendo a análise dos dados pelo Sistema Especialista.

### V.3.5. Métodos

Os principais métodos que compõem o *Objeto Gerenciador de Regras* são os seguintes:

#### Métodos

*Carregar Tabela*  
*Iniciar MI*  
*Processar*  
*Processar Regra*  
*Desenhar Regra*

#### Carregar Tabela

A Base de Conhecimento do R-TESE é armazenada em diversas estruturas de árvore de objetos. Para isto, recorre-se, primeiramente, a um analisador sintático que trabalha as regras e as estrutura em árvores; a própria interface do programa armazena diretamente estas regras e demais dados nas citadas estruturas. A rotina responsável por esta estratégia é denominada *Carregar Tabela*.

Esta rotina é responsável pelo armazenamento, em memória, de todas as regras do sistema, formando a lista *Regras* e a lista *Regras Temporizadas*, descritas anteriormente.

A seguinte lógica representa seu funcionamento:

<p>Carregar Regra do BD correspondente; Transformar Premissa/Ação em Árvore de Objetos; Se (Tipo = Regra Comum ou Regra Mista) Então     Adicionar Regra Comum ou Mista na lista <i>Regras</i>; Se (Tipo = Regra Mista ou Regra Temporizada) Então     Adicionar Regra Mista ou Temporizada na lista <i>Regras Temp.</i></p>
--

## Iniciar MI

Este método é executado na inicialização da inferência, sendo o responsável pelo processamento do mecanismo de *Start*, descrito na seção anterior.

Seu objetivo é, justamente, inicializar as variáveis do sistema, garantindo o funcionamento correto do processo. A seguinte lógica é representativa deste método:

<p>Carregar <i>Start</i> do BD correspondente; Transformar <i>Start</i> em Árvore de Objetos; Processar <i>Start</i>.</p>
---

## Processar

Este método representa o núcleo do mecanismo de inferência proposto neste trabalho. É executado por um mecanismo de *thread*<sup>1</sup>, cuja ativação é feita pelo Gerenciador do R-TESE, na inicialização deste ambiente.

Sua função é analisar a lista *Variáveis Modificadas*, em cada ciclo do processo de inferência, e estruturar, em uma coleção (*array*) de listas, as regras aptas ao disparo, obedecendo o método de resolução de conflito utilizado.

Aqui, apresenta-se a lógica que sintetiza o funcionamento do **Método 2**.

<sup>1</sup> Detalhes do funcionamento desta *thread* são descritos na próxima seção.



```
Repetir
  Enquanto (Há Elementos na Lista Var. Modificadas) Faça
  Início
    Para cada regra relacionada a uma variável modificada Faça
    Início
      Processar Premissa;
      Se (resultado da Premissa for verdadeiro) Então
        Adicioná-la à Lista Regras Seleccionadas;
      Fim Para;
    Fim Enquanto;

  Enquanto (Há Elementos na Lista Regras Seleccionadas) Faça
  Início
    Processar Regras de Maior Prioridade;
    Para cada Regra Temporizada Faça
      Processar Regra Temporizada;
    Fim Enquanto;
  Fim Repetir.
```

Esta lógica é executada até que não haja mais variáveis na lista *Variáveis Modificadas*, culminando, então, com a geração de pacotes com as informações resultantes da inferência.

### Processar Regra

Esta rotina é responsável pelo processamento, em particular, de cada regra do sistema. É executada no corpo do mecanismo de inferência (método anterior), processando as premissas e as ações das regras habilitadas nos diferentes ciclos.

O método faz parte do corpo do **Objeto Regra**, sendo representado pelo seguinte código:

```
Se (Premissa sem Erro) Então
  Processar Premissa;
Se (Resultado da Premissa Verdadeiro) E (Ação sem Erro) Então
  Processar Ação;
  Contar número de disparos da regra;
  Armazenar o instante do último disparo;
```

## Desenhar Regra

Esta rotina disponibiliza, ao desenvolvedor do Sistema Especialista, o desenho da árvore de objetos representativa de uma determinada regra. Isto possibilita a visualização de sua forma final, na memória, auxiliando a análise da mesma e verificação de erros em sua sintaxe.

O detalhamento do Gerenciador de Regras finaliza a descrição da estrutura interna do R-TESE. A seguir, mostra-se a *thread* associada a esta classe, responsável pela ativação de seus métodos.

### V.3.6. Thread Associada

O R-TESE possui um mecanismo de *thread*, responsável pela ativação do mecanismo de inferência, cujo funcionamento é sintetizado na figura V.6.

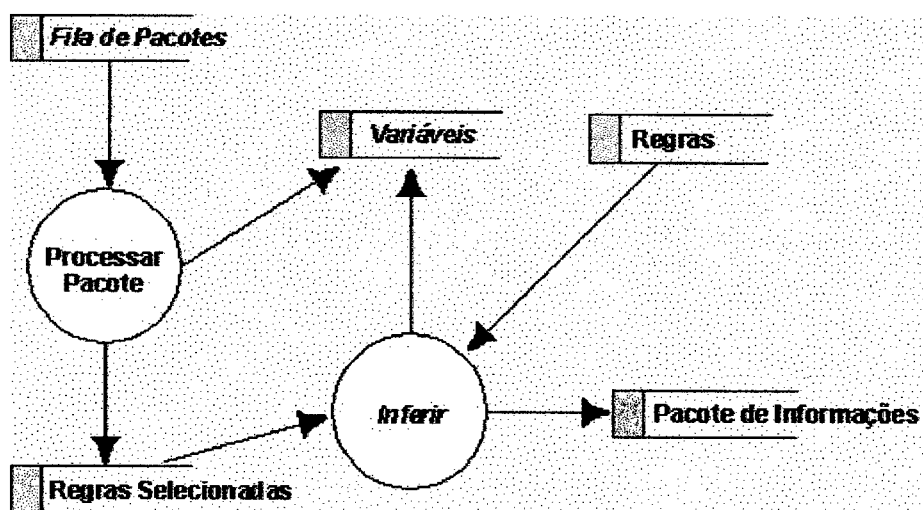


Figura V.6 - Fluxo de Dados do Mecanismo de *Thread*

Esta figura está inserida na figura III.2, que descreve a arquitetura do R-TESE. Assim, pode-se afirmar que esta *thread* é a peça chave do ambiente inteligente proposto

neste trabalho, pois executa as ações relativas à inferência, enquanto as outras duas *threads*, já apresentadas, controlam a chegada de pacotes e a ativação de temporizadores.

O detalhamento de seus métodos (processos *Processar Pacote* e *Inferir*) é feito na seção III.3.

Resumidamente, tem-se: o R-TESE acessa o primeiro pacote da *fila de pacotes*, analisando suas informações e disponibilizando-as ao sistema, atualizando a Base de Fatos local. Inicia-se, então, o processo de Inferência (método *Processar*, descrito na seção anterior), que encerra-se somente quando não houverem mais regras aptas ao disparo. A memória de trabalho conterà, então, as informações resultantes desta inferência, que são enviadas à *Interface Homem-Máquina* através de pacotes de informações.

#### V.4. Considerações

Os dois modos de funcionamento do mecanismo de inferência foram determinados mediante estudos de Sistemas Especialistas existentes, e na literatura encontrada.

A escolha do melhor método vai depender de cada aplicação, ou seja, do modo como será implementada cada Base de Conhecimento, do tempo de resposta e dos objetivos desejados.

No caso do Sistema Elétrico da ITAIPU Binacional, baseando-se em estudos de versões anteriores (IOI60), utiliza-se o método 2 como estratégia de resolução de conflito, onde redefine-se um novo conjunto de conflito após a execução das regras que fazem parte de um determinado nível de prioridade.

O próximo capítulo descreve o protótipo de um Sistema Especialista, cujos testes de inferência utilizaram este método.

## **V.5. Conclusão**

Este capítulo descreveu o *Gerenciador de Regras* do R-TESE, finalizando o detalhamento completo deste ambiente para desenvolvimento de Sistemas Especialistas.

Sua implementação disponibiliza um objeto que armazena informações sobre as regras que estruturam a Base de Conhecimentos do projeto.

Com o modelo definido, parte-se para a apresentação do protótipo de um SE, o Sistema Especialista de Apoio à Operação da Hidrelétrica de ITAIPU, objetivando a validação do funcionamento do ambiente proposto.

# **Capítulo VI**

## **Sistema Especialista de Apoio à Operação**

Este capítulo descreve os detalhes da estrutura do Sistema Especialista de Apoio à Operação, sistema inteligente aplicado à supervisão do Sistema Elétrico da Hidrelétrica de ITAIPU.

A estrutura, aqui descrita, detalha suas funcionalidades, enfatizando aspectos de projeto e operacionalização do R-TESE.

### **VI.1. Introdução**

Este capítulo tem como meta apresentar o protótipo de um SE, desenvolvido a partir do R-TESE, que provê instruções e informações aos operadores do Sistema Elétrico.

O Sistema de Transmissão de 750 kV, juntamente ao SSO, descritos no capítulo II, foram selecionados como elementos desta primeira aplicação. Esta escolha foi feita devido à sua importância para o Sistema Interligado ITAIPU - Tijuco Preto, pois o funcionamento correto de seus componentes é essencial ao abastecimento de energia para o Sul e Sudeste do país.

Estrutura-se, assim, o *Sistema Especialista de Apoio à Operação* (SEAO), apresentando-se como um ambiente inteligente, de apoio à área de operação da Usina.

O conhecimento do sistema está definido nas Instruções de Operação da Interligação (IOIs), armazenadas em um documento que relata as condições de operação do Sistema Interligado. As informações contidas nas IOIs estão presentes na Base de Conhecimento do SE, na forma de regras de produção, fornecendo um ensaio sobre seu domínio de atuação.

Nas próximas seções, são destacados aspectos de construção deste ambiente, como também os testes realizados com o mesmo, que serviram para validar o funcionamento do R-TESE, dentro das exigências de projeto impostas ao mesmo.

## **VI.2. Projeto de Sistemas Especialistas**

O R-TESE foi criado de modo a permitir ao próprio projetista do conhecimento implementar o Sistema Especialista desejado.

Inicialmente, o desenvolvedor tem a opção de criar um novo SE, ou abrir um já existente, alterando seus dados. Para isto, o R-TESE cria, como padrão, arquivos de projeto com a extensão *.PJ*. Estes arquivos, já apresentados em capítulos anteriores (seções IV.2.1 e V.2.1) apresentam-se na forma de bancos de dados, e contêm o nome dos BDs que armazenam as Bases de Fatos e de Regras do SE em estudo.

A partir do arquivo de projeto, o R-TESE identifica as bases nele especificadas, armazenando-as em memória e formando a estrutura descrita neste trabalho. A figura VI.1 mostra a tela de projeto disponibilizada pelo ambiente inteligente.

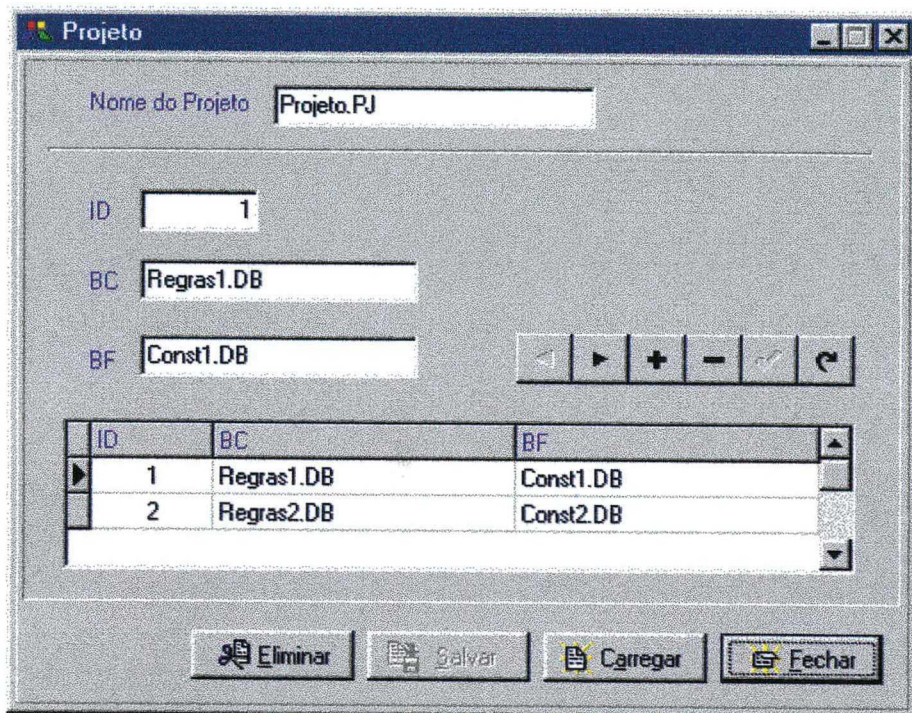


Figura VI.1 - Projeto de Sistemas Especialistas

Como indica a figura, as Bases de Conhecimento e de Fatos são inseridas sempre em pares, ou seja, para todo conjunto de regras deve haver um conjunto de variáveis relacionado às mesmas.

A partir desta tela, carrega-se o SE à memória do R-TESE. Caso os pares BC/BF não sejam arquivos já existentes (com regras e variáveis pré-definidas), o próprio ambiente cria os novos arquivos definidos pelo desenvolvedor, que se encarregará de elaborar regras e definir as variáveis que farão parte da memória de trabalho.

A seguir, são mostrados os passos necessários para o desenvolvimento de um SE a partir do R-TESE, evidenciando aspectos de implementação de sua Base de Conhecimento e de sua Base de Fatos.

### VI.2.1. Interface com o Usuário

Para a construção de um Sistema Especialista, implementado a partir do R-TESE, a *Shell* disponibiliza menus de múltipla escolha. Os elementos da interface com o usuário são apresentados na figura VI.2.

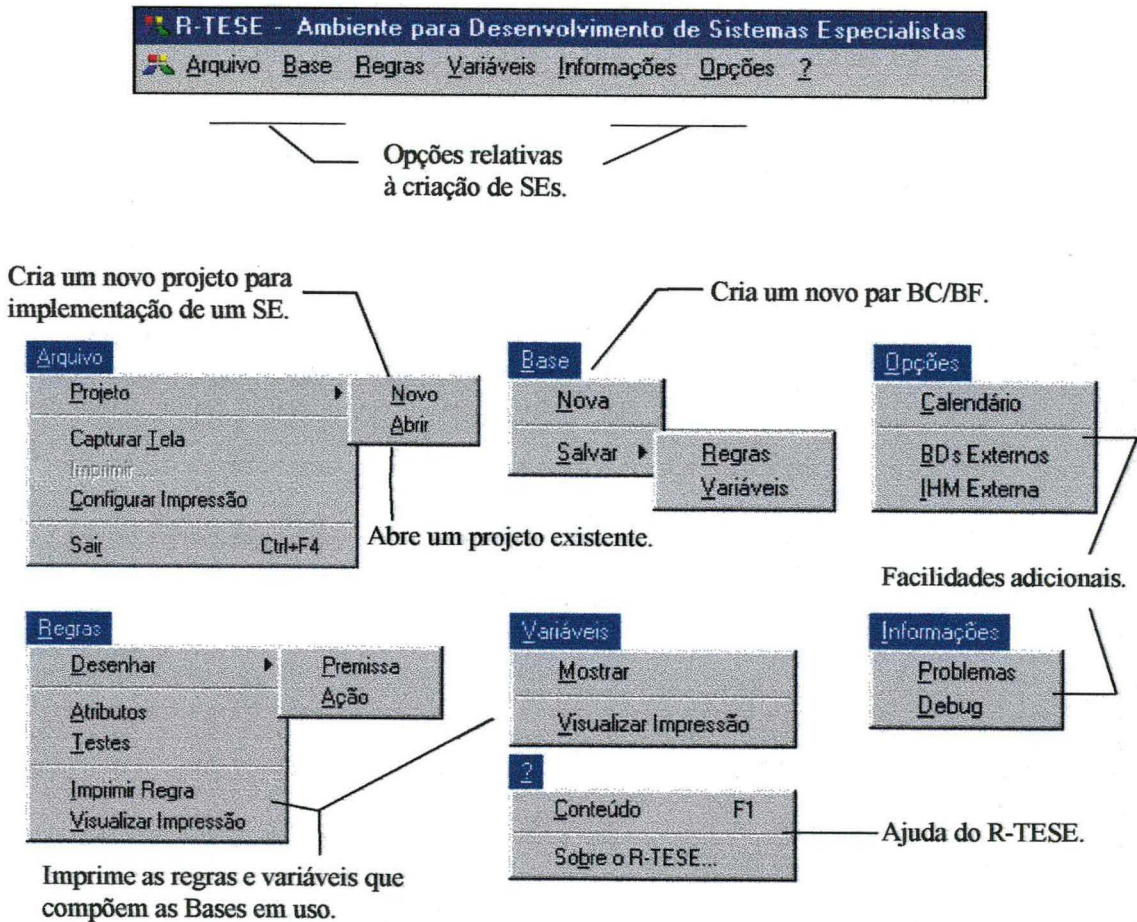


Figura VI.2 - Itens de Menu para Execução de Tarefas

Estes itens trazem todas as opções de criação e edição de bases, fornecendo funcionalidades à construção de um SE (como impressão de regras e variáveis, arquivos de ajuda, captura de telas, acesso a problemas ocorridos na estruturação das regras, visualização de interfaces externas).



Tais opções permitem o acesso às telas específicas, detalhadas nos próximos itens, e a manipulação dos dados que estas suportam.

## VI.2.2. Base de Conhecimento

Esta seção descreve a parte do sistema que incorpora as Instruções de Operação da Interligação. São aproximadamente 50 regras, relativas às IOIs 61, 62 e 63, detalhadas nos *Apêndices 2 e 3*.

O *Apêndice 2* apresenta as regras na forma como estas foram definidas pelos operadores do Sistema Elétrico, e repassadas ao desenvolvedor do SE. O *Apêndice 3*, por sua vez, apresenta estas mesmas regras na notação suportada pelo R-TESE, incluindo, ainda, algumas inicializações necessárias ao funcionamento correto do sistema.

Estas regras definem a Base de Conhecimento do SEAO, cujo estudo e análise validaram o funcionamento do R-TESE.

As figuras VI.3 e VI.4 ilustram as telas específicas do SEAO, dando ênfase às regras que o compõem e seus atributos.

Para criar regras, realiza-se os seguintes procedimentos:

- ⇒ Na janela de projeto (figura VI.1), indica-se quantos e quais são os BDs de regras que fazem parte do SE em estudo. Estes arquivos são carregados à memória do R-TESE, onde as regras são armazenadas na forma de árvores de objetos;
- ⇒ As janelas específicas do R-TESE são abertas (figuras VI.3 e VI.4), apresentando as regras pertinentes ao projeto;
- ⇒ O desenvolvedor tem a opção de editar as regras já construídas, ou criar novas regras, inserindo as premissas e ações correspondentes;

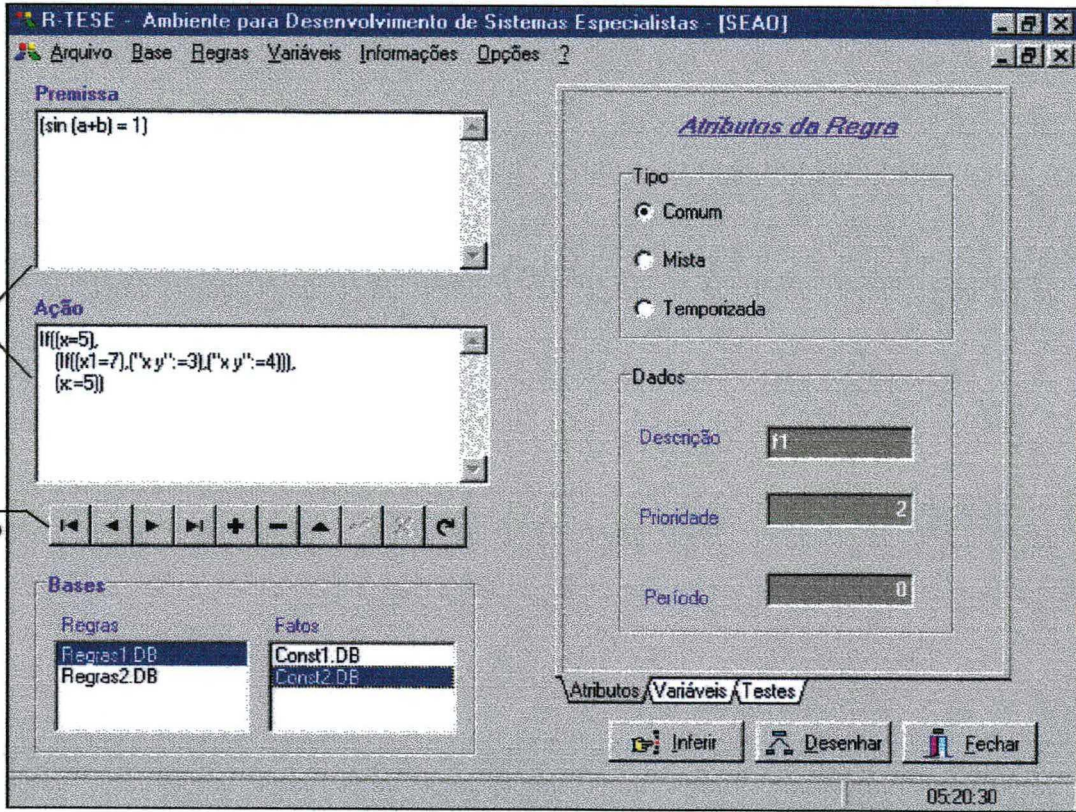


Figura VI.3 - Tela Específica do R-TESE - Atributos

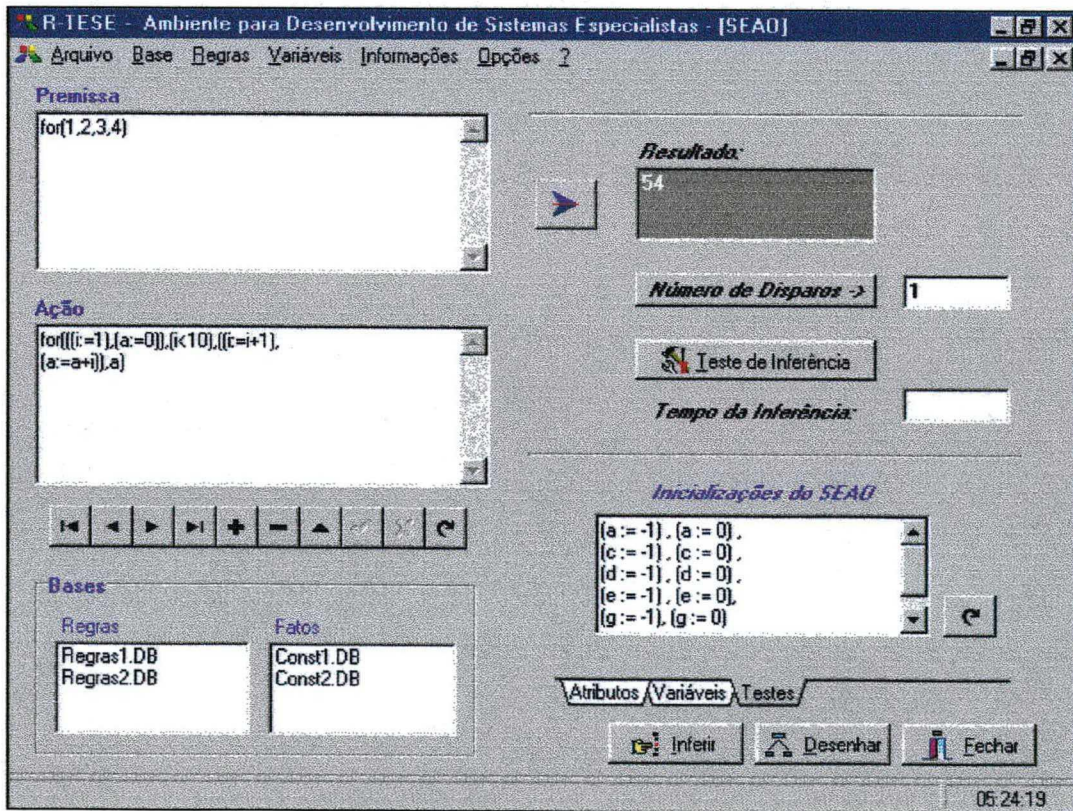


Figura VI.4 - Tela Específica do R-TESE - Testes

⇒ É permitida, ainda, a realização de testes individuais no processamento de cada regra, como também ativar seu mecanismo de inferência. Obtém-se, desta forma, dados relativos às regras executadas, como o número de ativação das mesmas e o tempo de CPU utilizado para o seu processamento.

Além do exposto, o R-TESE apresenta algumas particularidades em seu funcionamento, ou seja, artifícios utilizados para inicializar as variáveis que compõem o SE em estudo.

Há dois tipos de inicializações: as realizadas a cada ciclo do processo de inferência (a cada resolução de conflito), e aquelas executadas apenas uma vez, na ativação do mecanismo de *thread* que administra a cadência de ações do mecanismo de inferência.

As primeiras são definidas como regras de prioridade máxima (=1), sendo utilizadas quando se necessita atribuir valores a determinadas variáveis em cada ciclo de execução. Como exemplo, pode-se citar a regra 02, do apêndice 3, onde a cada ciclo deve-se verificar a alteração da hora/data atual, para determinação de um novo período de carga. O valor desta variável ("*período de carga*") é utilizado em outras regras que compõem a Base de Conhecimento, influenciando no resultado final da inferência, ou seja, nas ações indicadas ao operador do sistema.

As inicializações executadas na ativação do mecanismo de *thread* são utilizadas como um mecanismo de *start* para o processo de inferência. Considerando-se que as regras selecionadas pelo mecanismo de inferência são aquelas cujas variáveis associadas sofreram alteração, este tipo de inicialização afeta o valor inicial de certas variáveis, fazendo-as entrar no conjunto de variáveis modificadas e, em consequência, ativando as regras relacionadas às mesmas. Esta estratégia é responsável pelo início do processamento, uma vez que define um primeiro conjunto de regras aptas ao disparo.

Neste tipo de inicialização são determinados, também, valores fixos utilizados no corpo das regras. Como exemplo, cita-se a regra 01, do apêndice 3, onde são definidas algumas constantes, como o número de reatores presentes no Sistema Interligado Itaipu-Tijuco Preto.

### VI.2.3. Base de Fatos

A Base de Fatos de um SE criado a partir do R-TESE contém todas as variáveis pertinentes ao sistema em estudo.

No caso do SEAO, as variáveis que o compõem são as que estruturam as regras contidas na IOIs 61, 62 e 63, além de todas aquelas envolvidas na supervisão da operação, ou seja, presentes no pacote de dados enviado pelo SSO.

As variáveis que fazem parte do corpo das regras, definidas no protótipo em estudo, encontram-se no *Apêndice 3*. Aquelas cujos valores e estado são disponibilizados pelo SSO ficam sempre armazenadas em um dos BDs de variáveis que compõem o SE, totalizando, aproximadamente, 1030 elementos.

As seguintes variáveis exemplificam o escopo abrangido pelo sistema:

U01.VALOR	FI-IV RE 330 LI1.VALOR
U01.STATUS	FI-IV RE 330 LI1.STATUS
U02.VALOR	ICAMPO U12.VALOR
U02.STATUS	ICAMPO U12.STATUS
DJ 86LI4.VALOR	CARGA LEVE
DJ 86LI4.STATUS	CARGA MÍNIMA
U11 - MW.VALOR	Geração de ITAIPU deve ser 999 para esta configuração.
U11 - MW.STATUS	
ON	Tensão da Barra 500 kV da GIS abaixo da mínima operativa. Providenciar elevação.
OFF	
TRUE	Tensão da Barra 500 kV da GIS acima da máxima operativa. Providenciar redução.
FALSE	Providenciar redução de carga na LI-IPU-60Hz-FI-3.

A figura VI.5 mostra a tela específica do R-TESE, indicando as Bases de Fatos e as variáveis que o compõem.

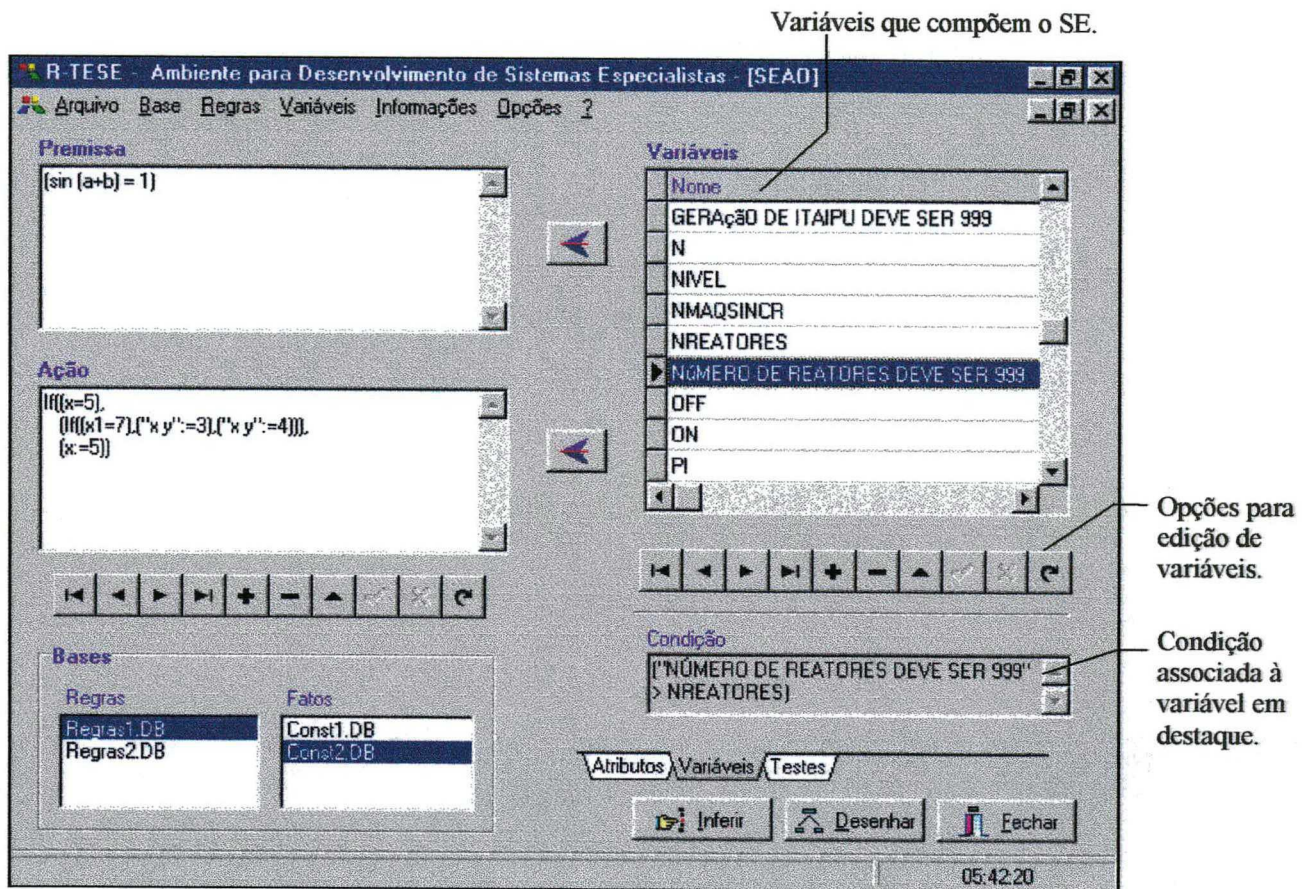


Figura VI.5 - Tela Específica do R-TESE - Variáveis

Os passos necessários para criação e edição de variáveis são os seguintes:

- ⇒ Na janela de projeto (figura VI.1), indica-se quantos e quais são os BDs de variáveis que fazem parte do SE em estudo. Estes arquivos são carregados à memória do R-TESE, formando a memória de trabalho do Sistema Especialista;
- ⇒ As janelas específicas do R-TESE são abertas e uma caixa de listagem apresenta as variáveis pertinentes ao projeto, juntamente a seus atributos;
- ⇒ Pode-se adicionar e/ou excluir variáveis e valores, através de botões específicos para este fim.

Neste ponto, tem-se conhecimento sobre a estruturação de um SE a partir do R-TESE. Parte-se, então, para a descrição dos aspectos gerais de seu Editor de Regras e para o detalhamento dos testes realizados, que serviram para validar este estudo.

### **VI.3. Aspectos Gerais**

Como demonstrado nas seções anteriores, o Editor de Regras é uma ferramenta especial para a manutenção das bases em estudo, disponibilizando uma Interface Gráfica com funções que atendem aos requisitos operacionais.

Primeiramente, o usuário tem acesso a uma tela principal, onde visualiza-se itens de menu que permitem a construção e a modificação de Sistemas Especialistas, criados a partir do R-TESE.

A partir da tela principal, tem-se acesso às telas que definem o Sistema Especialista propriamente dito, fornecendo dados mais detalhados sobre o processo.

Como ilustrado nas figuras VI.3, VI.4 e VI.5, as telas específicas do Editor de Regras apresentam as seguintes características:

- Relacionam informações referentes às regras que compõem a Base de Conhecimento em estudo, tais como: tipo da regra (comum, mista, temporizada), descrição, prioridade, período;
- Mostram as variáveis que compõem a Base de Fatos do SE, como também os dados gerais de cada uma, como nome, valor, identificador (ação, valor, estado), localização da mesma no pacote de informações gerado pelo R-TESE; além destes atributos, é indicada a condição associada a cada variável, cuja execução determina se a mesma será ou não disponibilizada ao operador do Sistema Elétrico;
- Possibilitam a realização de testes sobre a Base de Conhecimento, como processamento de uma determinada regra, e a ativação de um ciclo completo de inferência, desde o

recebimento do pacote de dados do SSO, até a exteriorização de informações a Interfaces Externas;

- Possibilitam a inicialização do mecanismo de inferência propriamente dito, através da ativação da *thread* responsável pela execução de seus métodos.

Além destas funcionalidades, o R-TESE oferece algumas funções adicionais para permitir maior flexibilidade na construção de um SE, sendo apresentadas a seguir.

### Bancos de Dados Externos

Disponibiliza a visualização dos BDs externos, associados ao R-TESE, responsáveis pela definição de uma estrutura que permite a leitura dos pacotes de dados provenientes do SSO<sup>1</sup>. A figura VI.6 ilustra estes elementos.

The screenshot shows a window titled "Bancos de Dados Externos" with four sections, each containing a table:

Principal		
CODIGO	DESCRICAO	VALOR
0	IPU-60 DJ 52-31 QP04	1
1	OBRA DJ 31-69 KV	1
2	OBRA DJ 34-69 KV	1

Descrição	
CODIGO	DESCRICAO
0	DJ 52-31 QP04
1	DJ 31-69
2	DJ 34-69

Variáveis		
REGRA	DESCRICAO	VARIAVEL
2	03T01/43TL1/83L11 AL GERAL	.Status
1	03T01/43TL1/83L11 AL GERAL	.Valor
2	03T01/43TL1/83L11 DISC FASE	.Status

Funções	
REGRA	FUNCAO
1	["?.valor" := GetV(@)]
2	["?.status" := GetS(@)]

Figura VI.6 - Bancos de Dados Externos

<sup>1</sup> Capítulo III, seção III.3.3.3

### IHM Externa

Permite o acesso à Interface Externa, responsável pela leitura dos pacotes de informações disponibilizados pelo R-TESE, ao fim de cada ciclo de inferência (detalhes desta estrutura são apresentados na próxima seção).

### Calendário

Permite a inserção de datas indicativas de feriados, em um BD correspondente, detalhado no *Apêndice 1*.

Como ilustra a figura VI.7, o botão  Feriado permite que sejam armazenadas, neste BD específico, as datas relativas aos feriados anuais.

Este banco de dados é utilizado para processamento de algumas regras, pertinentes ao *Sistema Especialista de Apoio à Operação*, que determinam o período de carga das máquinas em operação na Hidrelétrica de ITAIPU, em função da hora e do dia da semana (dia útil, sábado, domingo ou *feriado*).

Neste sentido, faz-se necessário um arquivo que indique quais são os feriados anuais, para que haja um controle efetivo da carga.

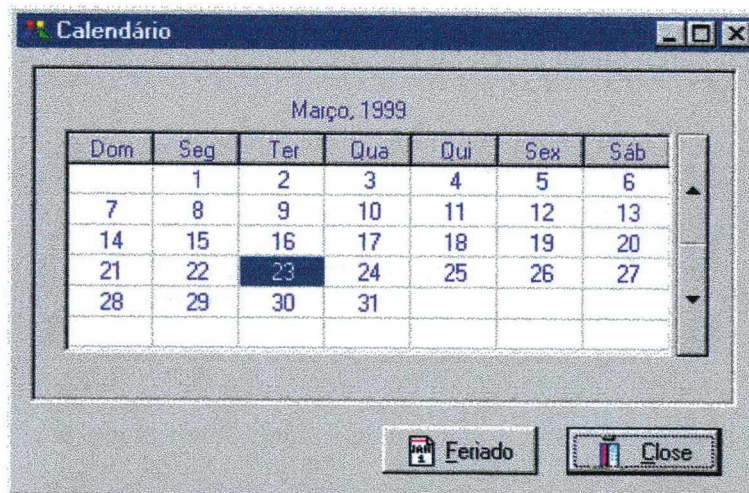


Figura VI.7 - Calendário



### Visualizador de Regras

Possibilita a visualização da premissa e da ação de cada regra na forma de árvores de objetos. Esta funcionalidade auxilia a verificação da estrutura interna da regra, indicando se esta foi editada de maneira correta em sua Base de Conhecimento.

A figura VI.8 ilustra este visualizador, mostrando a estrutura interna da seguinte regra, correspondente à função externa *IF*:

```

SE X = 5
ENTÃO
    SE X1 = 7
    ENTÃO XY := 3
    SENÃO XY := 4
    SENÃO X := 5
  
```

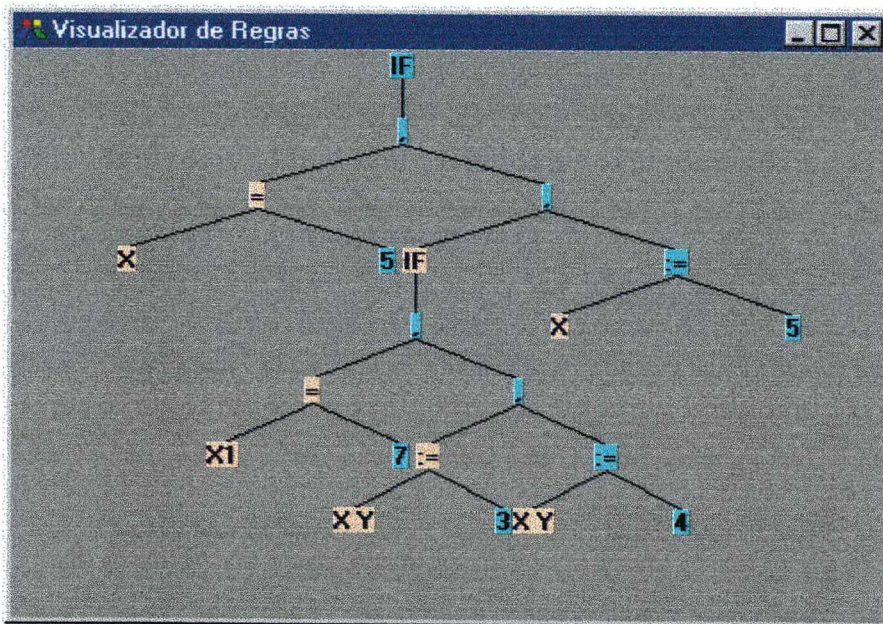


Figura VI.8 - Visualizador de Regras em Estrutura de Árvore

### Indicador de Problemas na Estrutura das Regras e Variáveis

Indica os erros encontrados na sintaxe das regras, quando estas são analisadas e estruturadas em árvores de objetos. Estes erros podem ser de utilização incorreta de parênteses, falta de aspas (“ ”) entre variáveis que apresentam símbolos ou espaços em branco, e/ou incompatibilidade com as funções externas aceitas pelo ambiente inteligente.

Indica, ainda, a existência de variáveis repetidas nos BDs de variáveis que fazem parte da estrutura do projeto em estudo. Neste caso, apenas uma variável é inserida na memória de trabalho do SE, mas avisa-se o usuário da presença da mesma em locais distintos.

Os problemas ocorridos na estruturação do Sistema Especialista não invalidam seu funcionamento, pois o próprio R-TESE desconsidera as informações relativas aos dados errôneos. A janela que disponibiliza estes dados é mostrada na figura VI.9.

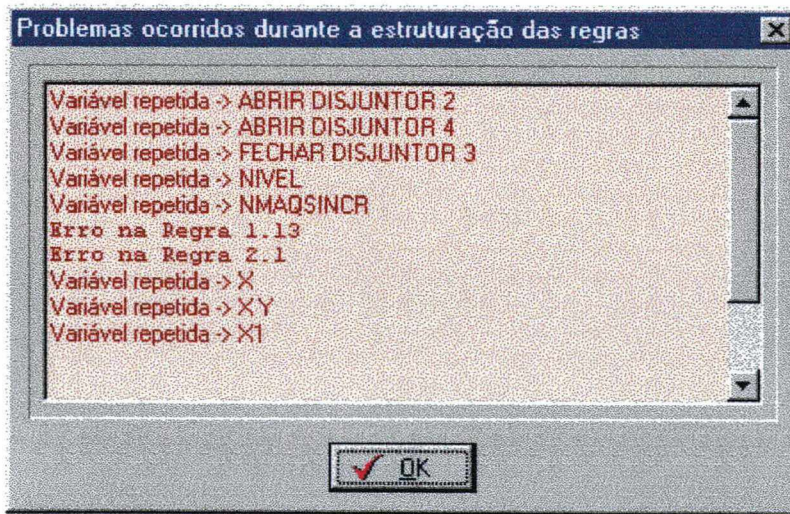


Figura VI.9 - Janela Indicativa de Problemas Ocorridos

### **Depurador do Mecanismo de Inferência**

Durante a execução do mecanismo de inferência, um depurador auxilia o usuário do sistema na verificação do disparo das regras.

Esta janela, ilustrada na figura VI.10, mostra a seqüência de ativação das regras durante cada ciclo, através do identificador da regra, da data e da hora de ativação da mesma.

Esta funcionalidade auxilia a validação do funcionamento do ambiente inteligente proposto, facilitando a análise do comportamento de seu mecanismo de inferência.

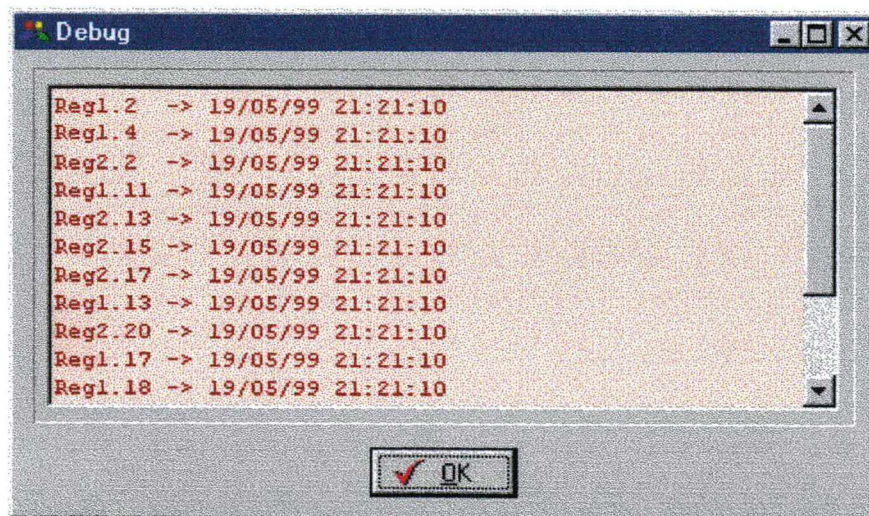


Figura VI.10 - Janela de Depuração do Mecanismo de Inferência

Na seqüência, apresenta-se a Interface Externa que recebe as informações disponibilizadas pelo R-TESE, resultantes do processamento das regras pertinentes a cada Sistema Especialista estudado.

#### VI.4. Interface com Ambientes Externos

Ao final de cada busca, o R-TESE envia um pacote de informações a ambientes externos, contendo os comandos indicados pelo Sistema Especialista. Uma Interface Homem-Máquina muito simples foi desenvolvida para visualização destes dados.

A Interface Externa disponibiliza, ao usuário, informações confiáveis para a atualização do sistema, com dados inferidos em tempo real, referentes ao pacote de informações enviado pelo R-TESE, como mostra a figura VI.11.

O funcionamento do protocolo de comunicação entre R-TESE/Interfaces Externas foi descrito na seção III.3.3.2.

Como afirmado, é utilizada a mesma técnica de controle de recebimento de pacotes utilizada pelo R-TESE, onde duas *threads* controlam a cadência de suas ações.

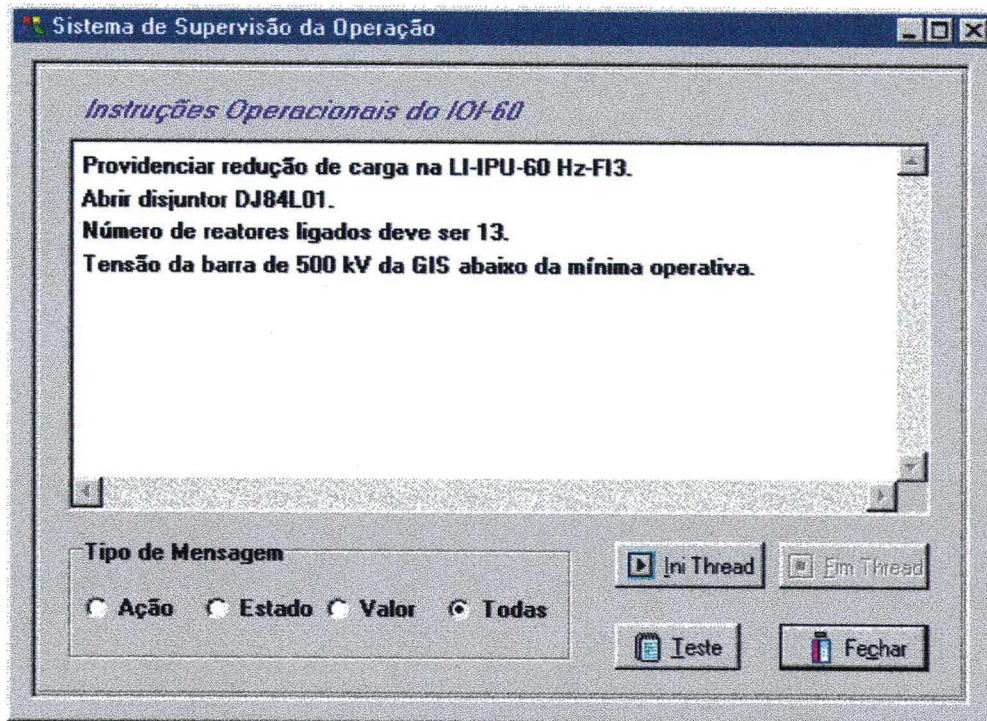


Figura VI.11 - IHM Externa

O R-TESE disponibiliza em disco, ao término de cada inferência, um novo pacote de informações. Um mecanismo de *thread* verifica, de 1 em 1 segundo, a existência de novos pacotes, inserindo-os em uma *fila de pacotes*. Uma outra *thread*, paralela a esta, retira pacotes da fila, analisa suas informações, e as disponibiliza ao operador do Sistema Elétrico.

Em próximas versões, esta IHM poderá ser incorporada ao SSO, juntamente às suas funcionalidades, onde suas informações serão disponibilizadas em janelas específicas.

## VI.5. Testes e Validação dos Resultados

Durante a implementação do R-TESE, testes foram realizados com o propósito de garantir a confiabilidade dos resultados obtidos.

A validação do SSO já foi realizada em trabalhos anteriores [Capanema 97],

ficando explícito que o sistema de supervisão atual disponibiliza dados confiáveis aos operadores, e as ações sobre estes dados são processadas adequadamente.

Para validar o funcionamento do R-TESE, foi utilizado o protótipo apresentado neste capítulo. Seus módulos foram testados separadamente, como também a integração entre eles. A partir do estudo do SEAO, destaca-se, a seguir, alguns dos principais testes realizados no decorrer deste trabalho.

### Testes Específicos

Os testes específicos detalham aqueles realizados com o propósito de validar o funcionamento de uma determinada unidade, com funcionalidade particular. São eles:

#### *Pacotes de Dados provenientes do SSO:*

Testes	Resultados
Leitura de pacotes de dados disponibilizados pelo SSO, contendo os valores das grandezas analógicas e digitais supervisionadas.	Atualização da memória de trabalho do sistema, permitindo o início de um novo ciclo de inferência.
Verificação do intervalo de tempo entre a leitura de cada pacote.	O intervalo entre a leitura de dois pacotes consecutivos é, geralmente, de 3 segundos. Caso a inferência tenha excedido este tempo, os novos pacotes disponibilizados pelo SSO são inseridos em uma fila, esperando para serem processados.
Simulação de perda de comunicação com o SSO, tornando-o inativo.	O SEAO continua ativo, verificando, através de um mecanismo de <i>thread</i> , a chegada de pacotes de dados do SSO, para atualização da memória de trabalho e início do ciclo de inferência.

#### *Pacotes de Informações enviados ao SSO:*

Testes	Resultados
Verificação, por meio de um mecanismo de <i>thread</i> , da chegada de novos pacotes para serem processados pela IHM.	Análise do pacote de informações, e disponibilização das mesmas aos operadores do Sistema Elétrico.

**Gerenciador de Variáveis:**

Testes	Resultados
Verificação da atualização da memória de trabalho, quando analisados os pacotes de dados provenientes do SSO.	Validação do funcionamento do Gerenciador de Variáveis.
Verificação da atualização da memória de trabalho, durante o processamento das regras pelo mecanismo de inferência.	
Verificação do funcionamento do mecanismo de <i>thread</i> responsável pela ativação das variáveis temporizadas, presentes no corpo das regras.	

**Gerenciador de Regras:**

Testes	Resultados
Análise da construção de árvores de objetos, a partir das regras presentes na Base do Conhecimento do SE em estudo.	Validação do funcionamento do Gerenciador de Regras.
Verificação do funcionamento do mecanismo de <i>thread</i> responsável pela ativação do MI.	
Controle da execução das regras e da resolução de conflito, realizadas pelo MI.	

**Editor de Regras:**

Testes	Resultados
Criação de novos Sistemas Especialistas, a partir da definição de BDs de projeto (.PJ), contendo um ou mais conjuntos Base de Regras/Base de Fatos.	Validação do funcionamento do Editor de Regras.
Visualização de SEs existentes, manipulando seus dados, como inserção/exclusão de regras e variáveis.	
Visualização das regras do SE em estudo, na forma de árvores de objetos, através da funcionalidade disponibilizada pelo Editor.	

**Outros:**

Testes	Resultados
Acompanhamento da ativação das regras, pelo MI, através de uma janela que indica a hora, a data e a regra disparada, na seqüência definida por cada ciclo do processamento.	Facilidade na verificação da execução das regras, auxiliando na validação do processo.
Verificação do tempo de processamento do mecanismo de inferência, para verificação da validade dos requisitos pré-estabelecidos.	O tempo <i>máximo</i> de uma inferência foi de 30 milisegundos, para um total de 1000 regras processadas.
Análise do funcionamento global, para validação do ambiente de construção de Sistemas Especialistas.	Validação dos resultados, culminando com o protótipo de um ambiente especialista que auxiliará o operador do Sistema ITAIPU na tomada de decisões.

**Testes Globais**

Os testes globais validam o funcionamento do sistema como um todo, ou seja, simulam determinadas configurações do Sistema Elétrico, analisando a veracidade das informações disponibilizadas pelo Sistema Especialista ao operador do processo.

Vários testes foram realizados para a validação do ambiente inteligente. Neste sentido, mostra-se dois testes executados sob condições adversas do sistema.

Primeiramente, foi aumentada a tensão da barra de 500 kV da Subestação Isolada a Gás (GIS). Com esta ação, o mecanismo de inferência ativou a seguinte regra (*regra 19 - apêndice 3*):

*Se* ("IPU-60 BARRA A4-KV.VALOR" > 550)  
*Então* ("Tensão da Barra 500 kV da GIS acima da máxima operativa.  
 Providenciar redução." := TRUE)

A figura VI.12 ilustra como a informação gerada pelo disparo desta regra é disponibilizada ao operador do sistema, na IHM Externa.

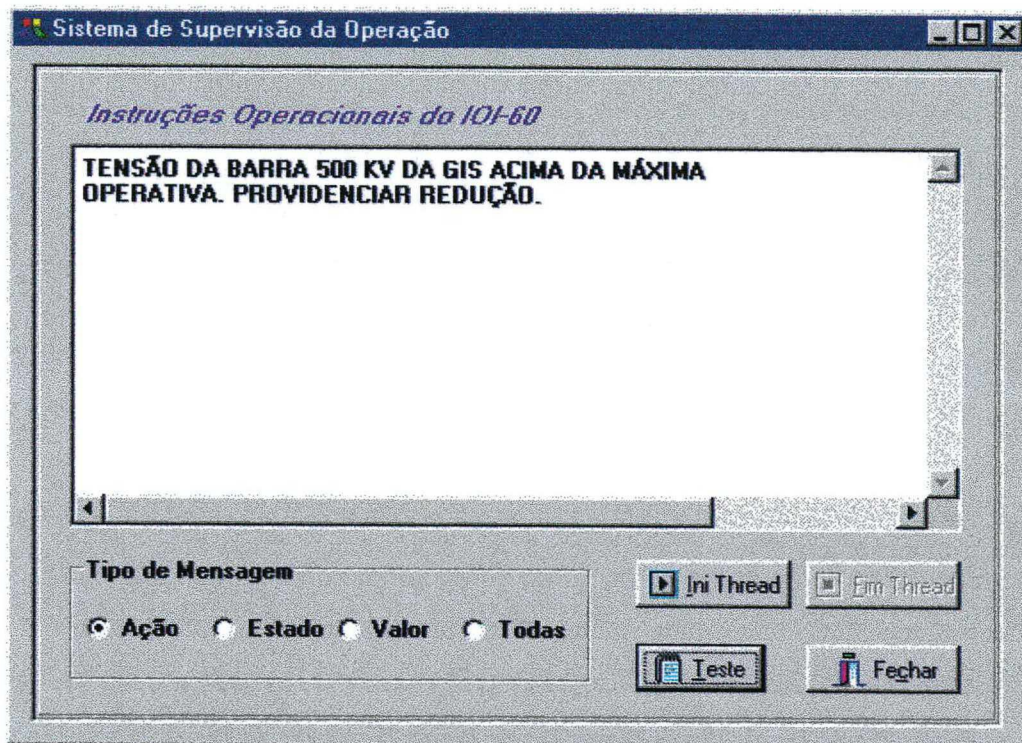


Figura VI.12 - Alteração da Tensão da Barra de 500 kV da GIS

Verificou-se, também, a determinação do período de carga das máquinas, em função da hora atual e do dia da semana. Os períodos de carga considerados são os mostrados a seguir:

PERÍODO DE CARGA	2ª FEIRA A SÁBADO	DOMINGO E FERIADOS
	DE --- ATÉ (horas)	DE --- ATÉ (horas)
MÍNIMA	00:00 -- 07:00 (segunda feira)	05:00 --- 10:00
LEVE	00:00 --- 07:00	00:00 --- 05:00 10:00 --- 17:00 22:00 --- 24:00
MÉDIA	07:00 --- 17:00 22:00 --- 24:00	17:00 --- 22:00
PESADA	17:00 --- 22:00	-----

A tabela acima é escrita na forma da função externa *IF* (regra 02 - apêndice 3), que apresenta a estrutura [ *Se / Então / Senão* ], e é utilizada para transformar tabelas de dados em regras de produção. Esta funcionalidade é disponibilizada pelo R-TESE através de um Editor de Tabelas<sup>2</sup>, que converte automaticamente tabelas em regras, em um formato suportado pelo ambiente inteligente. Esta particularidade auxilia a manutenção da Base de



Conhecimento do SE em estudo, devido às mudanças constantes nas informações pertinentes ao Setor Elétrico.

Utiliza-se, ainda, para definição da regra representativa da tabela de período de carga, as funções externas *HORA* e *DATA*, pois estas disponibilizam a hora e data atual, sendo necessárias à verificação dos limites impostos.

Com a mudança de horário, em um domingo (como ilustra a figura VI.13), a regra relativa à tabela descrita é executada, e a carga do sistema passa de média a leve.

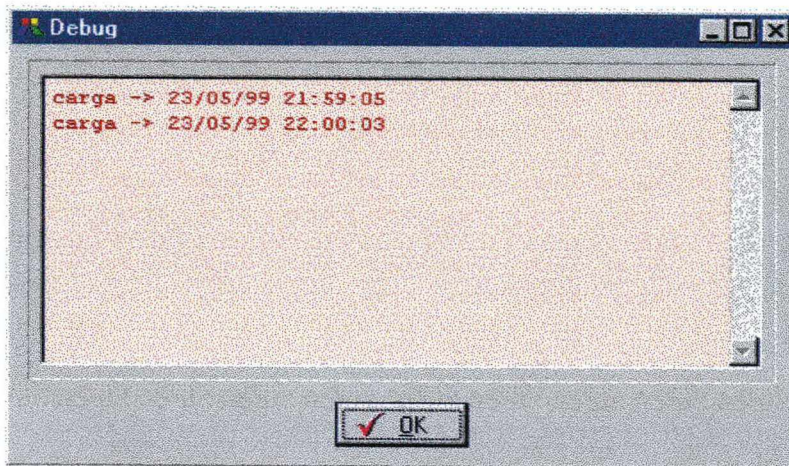


Figura VI.13 - Disparo da Regra Relativa ao Período de Carga

A partir desta ação, outra regra é habilitada (*regra 27 - apêndice 3*):

**Se** ("periodo carga" = "carga leve") Or  
 ("periodo carga" = "carga minima") And  
 ("RSE.VALOR" <= 3000) And  
 ("nmaqsincr" = 8) Or ("nmaqsincr" = 9)) And  
 ("nreatores330 OFF" >= 1)  
**Então** ("Geração de ITAIPU deve ser <= 5100 MW" := TRUE)

A figura VI.14 ilustra como a informação gerada pelo disparo desta regra é exteriorizada ao operador.

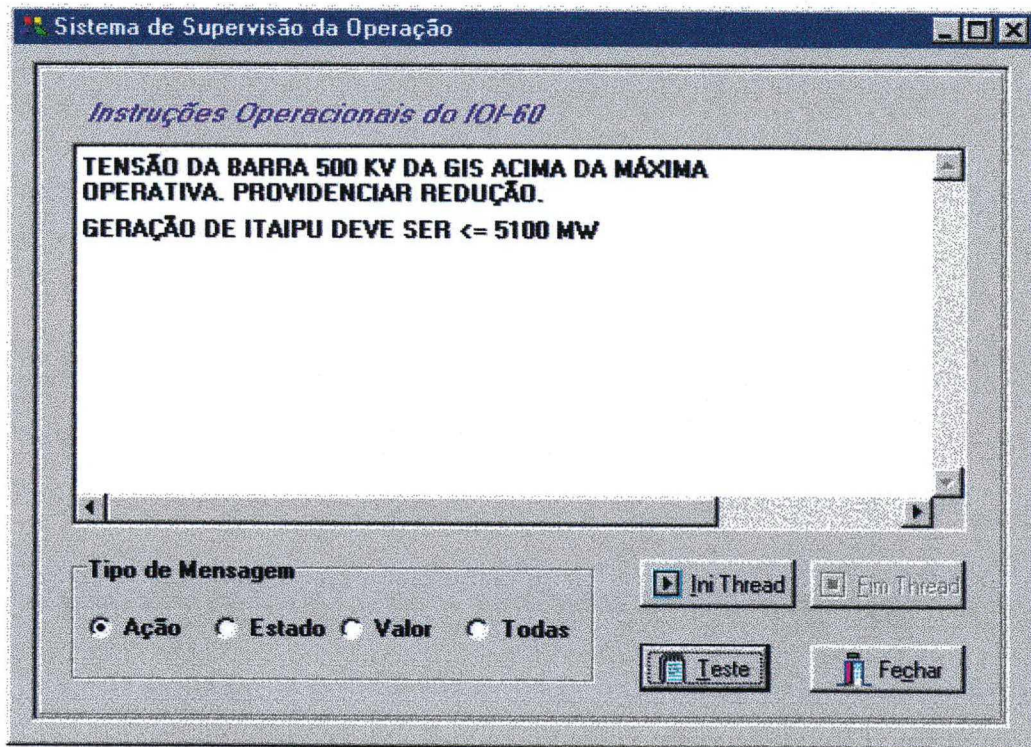


Figura VI.14 - Alteração do Período de Carga

Esta fase de experimentação foi executada desde o início da construção do software. Os testes destinaram-se a detectar erros no programa executável, e em achar falhas na sua concepção.

Realizou-se a verificação dos módulos detalhados neste trabalho, sendo seu código analisado sintaticamente antes da sua execução. Verificou-se, também, sua conformidade às especificações globais, funcionais e de desempenho.

## VI.6. Considerações

Para auxiliar o usuário na operação deste novo ambiente, foi implementada uma ferramenta de ajuda ao operador (*Help*), que pode ser consultada em tempo de execução.

Este *Help* incorpora as Instruções de Operação da Interligação, permitindo ao operador uma maior interação com o ambiente.

A idéia desses textos não é somente ativar uma ajuda on-line quando o usuário estiver utilizando o sistema, mas ir diretamente ao tópico explicativo de uma regra. Quando o usuário final ativa a ajuda on-line, o *Help* disponibiliza a IOI correspondente à regra mostrada na IHM no momento, auxiliando o usuário na tomada de decisões e na compreensão do processo como um todo. Se nenhuma regra estiver exposta, o arquivo de ajuda ativa sua tela principal.

A figura VI.12 ilustra a janela de ajuda disponibilizada quando uma regra pertencente à IOI 61 está sendo mostrada na interface gráfica.

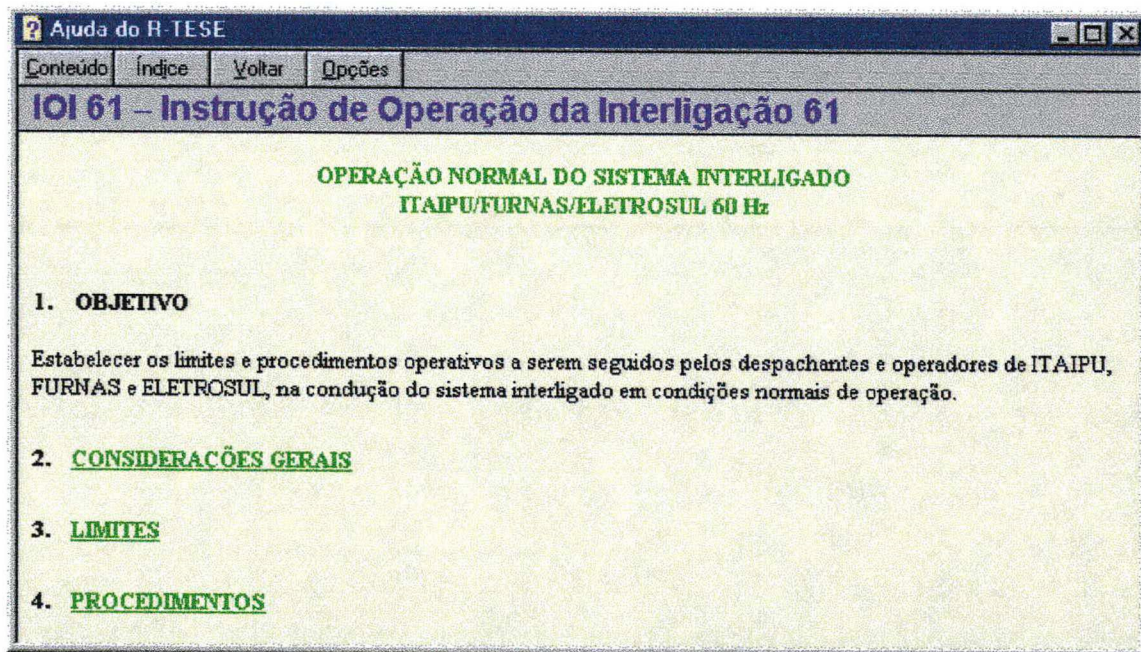


Figura VI.15 - Ajuda do R-TESE

Esta opção torna o SEAO mais robusto, já que auxilia a verificação da consistência das informações contidas em sua Base de Conhecimento.

## VI.7. Conclusão

Este capítulo definiu o funcionamento do *Sistema Especialista de Apoio à Operação* do Sistema Elétrico da ITAIPU Binacional, finalizando a estruturação do ambiente R-TESE.

A incorporação de funções estratégicas à monitoração do sistema conduzem a um aumento da qualidade dos serviços disponibilizados. Seus usuários utilizam-se das funcionalidades apresentadas para obter informações confiáveis sobre a operação do processo interligado.

Além destes aspectos, incorpora-se a vantagem do paradigma orientado a objetos, que tornou viável o desenvolvimento deste ambiente computacional, contribuindo com uma tecnologia atualizada, que proporciona a evolução dos sistemas que suprem as necessidades da empresa.

Conclui-se o trabalho proposto, onde se procurou utilizar conhecimentos em Inteligência Artificial para criar um ambiente seguro, dentro dos critérios de qualidade exigidos.

# Capítulo VII

## Conclusão

Este trabalho abordou um ambiente para desenvolvimento de Sistemas Especialistas, intitulado R-TESE, orientado à área de operação, com enfoque para o setor de geração e transmissão de energia elétrica.

Frente ao aumento da complexidade dos sistemas elétricos a serem supervisionados, os sistemas digitais apresentam benefícios que tornam a sua utilização incontestável. Desta forma, a Usina Hidrelétrica de ITAIPU investe na automatização de seus sistemas, através de mecanismos avançados no auxílio à tomada de decisões.

A construção de um software para a desenvolvimento de Sistemas Especialistas não é trivial, tendo em vista que este deve ser capaz de tratar problemas complexos do mundo real, que capturem e simulem o comportamento de especialistas humanos.

A necessidade de utilização deste tipo de ferramenta deve-se a diversos fatores tecnológicos e econômicos, dentre os quais tem-se: armazenamento e formalização do conhecimento de vários especialistas e a disponibilidade de uma ferramenta de apoio à tomada de decisões.

Especificamente, este programa foi desenvolvido para simplificar a implementação de Sistemas Especialistas em tempo real, na área de operação de plantas de geração de energia, apresentando uma máquina de inferência com alta velocidade de resposta.

O R-TESE, através de mecanismos de *thread*, administra o encadeamento das ações necessárias ao funcionamento dos sistemas inteligentes. O software processa pacotes de dados provenientes de ambientes externos, atualizando a Base de Fatos do sistema, ativa o mecanismo de inferência, além de controlar as restrições temporais impostas pelo processo em estudo.

O produto final apresenta as seguintes características:

- Armazena conhecimento sobre os passos operacionais necessários para executar uma determinada operação;
- Disponibiliza uma interface amigável aos usuários, tanto os operadores do Sistema Elétrico, quanto àqueles que garantem a manutenibilidade das Bases de Conhecimento. A interface oferece um conjunto de facilidades (menus, janelas, apresentação da estrutura das regras) que torna bastante simples a utilização da ferramenta;
- Permite alta velocidade de resposta, sendo capaz de obter dados do Sistema de Supervisão da Operação, analisá-los e formular um conjunto de ações para ajustar o sistema à situação desejada; tudo isto ocorre antes que novas informações sejam adicionadas ao sistema, modificando o estado do mesmo.

A confiabilidade de uma ferramenta como o R-TESE está intrinsecamente relacionada à sua operacionalização, sendo imprescindível a implementação de Bases de Conhecimento para garantir sua aceitação tanto a nível acadêmico quanto comercial.

Assim, para validação do R-TESE, construiu-se o protótipo de um Sistema de Apoio à Operação, apresentando uma abordagem em tempo real.

A necessidade deste tipo de sistema, no apoio à operação, é prover operadores/despachantes de carga com informações relevantes no caso de situações emergenciais, sugerindo, de forma rápida e segura, alternativas de otimização da operação do sistema elétrico.

O protótipo foi elaborado a partir de uma análise geral das Instruções de Operação da Interligação. Deste estudo, resultou a definição das características do Sistema Especialista, onde foram detectadas situações que exigiram a agregação de novas funcionalidades ao ambiente de desenvolvimento.

A análise do desempenho do protótipo permitiu verificar a importância do uso de um Sistema Especialista no diagnóstico e na tomada de decisões na área de operação do Sistema ITAIPU, validando as técnicas de representação de conhecimento e de procura adotadas.

Por outro lado, o especialista conta, ainda, com um Editor de Regras para entrada das informações referentes ao funcionamento adequado do sistema interligado, compondo as regras pertinentes à Base de Conhecimento do R-TESE. Esta ferramenta está em fase de refinamento, e seu objetivo é disponibilizar ao operador o acesso às instruções operacionais, como também prevenir a inclusão de configurações inconsistentes ao Sistema Especialista.

O produto gerado pelo R-TESE apresenta-se em estágio final de desenvolvimento, e entrará em operação no presente ano. Frente ao estado atual do trabalho, considera-se como perspectivas imediatas deste projeto:

- Aperfeiçoamento da ferramenta de edição de regras, com a inclusão de novas funcionalidades, como, por exemplo, a análise sintática das regras que constituem a Base de Conhecimento; isto permitirá que o usuário (especialista encarregado da manutenção do sistema inteligente) interaja mais facilmente com o ambiente, diminuindo o tempo de construção de bases consistentes;
- Construção de novos módulos de funções, em decorrência de validações e testes já realizados. Como proposta imediata, tem-se a avaliação histórica da tendência nos estados das variáveis analisadas, possibilitando que o Sistema Especialista faça previsões de situações emergenciais, em um tempo adequado para contorná-las;
- Construção de uma Base de Conhecimento real ampliada, com o auxílio dos operadores do Sistema ITAIPU, englobando os setores de 50 e 60 Hz da Usina. Esta base será

constituída pelo total das Instruções de Operação da Interligação, disponibilizando um sistema de informações completo, que permita a utilização da ferramenta especialista de forma mais efetiva;

- Geração de Bases de Conhecimento pré-compiladas, minimizando as etapas de construção das árvores de objetos, relativas a cada regra componente do sistema em estudo;
- Implementação de ações de controle, a fim de complementar o sistema desenvolvido, fazendo com que as ações, aplicadas atualmente pelo operador, sejam executadas automaticamente.

Baseando-se nos resultados obtidos e nos aspectos descritos, conclui-se que o projeto é estratégico no sentido de gerar uma tecnologia que agrega valor às condições operacionais da empresa em questão.

A supervisão do processo industrial da ITAIPU está se consolidando em um ambiente tecnologicamente atualizado, mesclando desenvolvimento “in-house” e parcerias com Universidades.

O ambiente proposto neste trabalho amplia o Sistema de Supervisão local, tornando-o mais funcional, e apresentando atributos de escalabilidade e usabilidade.



# Apêndice 1

## Bancos de Dados Utilizados na Estruturação do R-TESE

### Projeto do Sistema Especialista

	Tipo	Tamanho	Chave Primária
ID	Real		*
BC	String	30	
BF	String	30	

### Base de Regras

	Tipo	Tamanho	Chave Primária <sup>1</sup>
Premissa	Memo	240	
Ação	Memo	240	
Prioridade	Real		
Período	Real		
Tipo	String	1	
Identificador	String	10	
IOI	String	2	

<sup>1</sup> Os Bancos de Dados “Base de Regras” e “Feriados” não possuem chave primária.

**Base de Fatos**

	<b>Tipo</b>	<b>Tamanho</b>	<b>Chave Primária</b>
Nome	String	255	*
Valor	Real		
Pacote	Real		
Identificador	Real	240	
Condição	Memo		

**Descrição dos Pontos do SSO**

	<b>Tipo</b>	<b>Tamanho</b>	<b>Chave Primária</b>
Código	Short Integer		*
Descrição	String	40	

**Variáveis**

	<b>Tipo</b>	<b>Tamanho</b>	<b>Chave Primária</b>
Descrição	String	40	*
Variável	String	10	*
Regra	String	2	

**Funções**

	<b>Tipo</b>	<b>Tamanho</b>	<b>Chave Primária</b>
Regra	String	2	*
Função	Memo	240	

**Feriados**

	<b>Tipo</b>	<b>Tamanho</b>	<b>Chave Primária<sup>1</sup></b>
Dia	String	2	
Mês	String	2	
Ano	String	4	

**Banco de Dados do SSO**

	<b>Tipo</b>	<b>Tamanho</b>	<b>Chave Primária</b>
Código	Real	4	*
Descrição	String	40	*
Tipo	Real	3	
Classe	String	2	
UTR	Real	3	
PFE UTR	Real	3	
PCNOS	Real	3	
PPMímico	Real	3	
PFE EC	Real	3	
LimSupRaz	Real	10	
LimInfRaz	Real	10	
LimSupOp	Real	10	
LimInfOp	Real	10	
BMortaInf	Real	10	
BMortaSup	Real	10	
DefaultDig	Real	3	
Ativo	Boolean		
Multiplica	Real	3	
Valor	Real	10	
Status	Real	3	
QByte	Real	3	
QBit	Real	2	
Nível	Real	1	
Online	Boolean		
Printal	Boolean		
Porta	Real	3	
Casas	Real	1	
Digitos	Real	1	
CDDLlen	Real	1	
Limiar	Real	10	
AbaGraf	Real	3	

## Apêndice 2

### Base de Conhecimento Real

#### (notação dos operadores do Sistema ITAIPU)

#### IOI 61

##### Operação Normal do Sistema Interligado ITAIPU/FURNAS/ELETROSUL 60 Hz

01. **Se** Corrente de excitação da máquina  $U_{xx}^1 < 2000 \text{ A}$   
**Então** Coordenar com Furnas medidas para aumentar a corrente de campo, tais como: aumentar potência ativa e/ou reativa da máquina.
  
02. **Se** Tensão na barra 500 kV da GIS  $> 550 \text{ kV}$   
**Então** Tensão está acima da máxima operativa. Providenciar sua redução.
  
03. **Se** Tensão na barra 500 kV da GIS  $< 480 \text{ kV}$   
**Então** Tensão está abaixo da mínima operativa. Providenciar sua elevação.

---

<sup>1</sup>  $U_{xx} = U_{10} \text{ a } U_{18}$  - geradores do setor de 60 Hz

04. **Se** Potência da LI-IPU-60Hz-FI-1 > 1650  
**Então** Providenciar redução de carga na LI-IPU-60Hz-FI-1<sup>1</sup> U<sub>xx</sub> = U10 a U18 - geradores do setor de 60 Hz
05. **Se** Potência da LI-IPU-60Hz-FI-2 > 1650  
**Então** Providenciar redução de carga na LI-IPU-60Hz-FI-2.
06. **Se** Potência da LI-IPU-60Hz-FI-3 > 1650  
**Então** Providenciar redução de carga na LI-IPU-60Hz-FI-3.
07. **Se** Potência da LI-IPU-60Hz-FI-4 > 1650  
**Então** Providenciar redução de carga na LI-IPU-60Hz-FI-4.
08. **Se** (carga = leve ou mínima) e (RSE ≤ 3000) e (mais de um reator de 330 MVar desligado) e (8 ou 9 unidades sincronizadas)  
**Então** Geração de ITAIPU deve ser ≤ 5100 MW.
09. **Se** (desligado 1 reator em Itaberá da LT-IA-IT) e/ou (desligado 1 reator em Itaberá da LT-IT-TP)  
**Então** Se nº máq. sincronizadas em ITAIPU ≤ 6    **Então** FSE ≤ 4500 MW  
          Se nº máq. sincronizadas em ITAIPU = 7    **Então** FSE ≤ 4700 MW  
          Se nº máq. sincronizadas em ITAIPU ≥ 8    **Então** FSE ≤ 5200 MW.
10. **Se** (geração de ITAIPU/setor 60 Hz ≥ 4400) e (nº máq. sincronizadas = 7)  
**Então** tensão nas máquinas = 18,9 kV.
11. **Se** (geração de ITAIPU – setor 60 Hz ≥ 4700) e (nº máq. sincronizadas > 8)  
**Então** tensão nas máquinas = 18,9 kV.
12. **Se** tensão terminal de capacitor série de Ivaiporã < 780 kV  
**Então** limite de geração em ITAIPU = limite de geração em ITAIPU – 300, para cada 20 kV abaixo de 780.

## IOI 62

### Restabelecimento da Interligação

#### ITAIPU/FURNAS 60Hz

13. **Se** tensão na barra A3 = 0  
**Então** disjuntores 06A34, 06U12, 06U10, 06LI2 e 06LI1 devem estar desligados.
14. **Se** tensão na barra A4 = 0  
**Então** disjuntores 06A34, 06TA2, 06LI4, 06LI3 e 06U14, 06U16 e 06U18 devem estar desligados.
15. **Se** tensão na barra B3 = 0  
**Então** disjuntores 86B34, 86U13, 86U11, 86LI2 e 86LI1 devem estar desligados.
16. **Se** tensão na barra B4 = 0  
**Então** disjuntores 86B34, 86TA2, 86LI4, 86LI3, 86U15, 86U17 e 86U8A devem estar abertos.
17. **Se** disjuntores 86LI4, 86LI3, 86LI2, 86LI1, 06LI4, 06LI3, 06LI2 e 06LI1 estão abertos  
**Então** interligação ITAIPU/FURNAS - 60 Hz está aberta.
18. **Se** disjuntores 12626, 12646, 12656, 12616 da SE-FI-60Hz estão abertos  
**Então** linhas IPU-60 Hz-FI estão desenergizadas. Para energizá-las, a tensão máxima de pré-energização é de 460 kV, no sentido ITAIPU-Foz. Quando o sistema já estiver com carga, não há limitações.  
Para energizá-las no sentido Foz-ITAIPU, a tensão máxima de energização é de 530 kV.

## IOI 63

### Operação em Emergência do Sistema Interligado

#### ITAIPU/FURNAS/ELETROSUL 60 Hz

19. **Se** desligada a LT-FI-IV-1 ou a LT-FI-IV-2  
**Então** tabela do item 4.4.1.
20. **Se** desligada LT-IV-IA-1, ou LT-IV-IA-2, ou LT-IA-TP-1, ou LT-IA-TP-2  
**Então** tabela do item 4.4.2.
21. **Se** 1 reator desligado entre IV-IA e o remanescente estiver sem o reator de linha em IA  
**Então** devem estar ligados os reatores de barra da SE-IA (R1 e R2). Caso esteja ligado somente 1, o limite do fluxo entre IV-IA é de 2400 MW.
22. **Se** 1 reator desligado entre IA-TP e o remanescente estiver sem o reator  
**Então** deverão estar ligados pelo menos 3 reatores de 330 MVar em IA (de linha ou de barra) e o limite do fluxo entre IV-IA é de 2200 MW.
23. **Se** desligadas: uma linha IV-IA e uma linha IA-TP  
**Então**  $FSE \leq 1900$  e deverão estar ligados 2 reatores de barra em IA, podendo estar desligado até 1 reator de linha em IA ou em TP.
24. **Se**  $n^\circ$  de linhas de 765 kV desligadas = 0  
**Então** *Case*  $n^\circ$  máximo de reatores desligados =
- 2 :  $n^\circ$  mínimo de máquinas sincronizadas = 3
  - 3 :  $n^\circ$  mínimo de máquinas sincronizadas = 4
  - 4 :  $n^\circ$  mínimo de máquinas sincronizadas = 5
  - 5 :  $n^\circ$  mínimo de máquinas sincronizadas = 6
  - 6 :  $n^\circ$  mínimo de máquinas sincronizadas = 6 e pelo menos 1 dos reatores desligados deve ser de TP.

25. Se uma linha de FI-IV desligada

**Então** Case n° máximo de reatores desligados =

3 : n° mínimo de máquinas sincronizadas = 3

4 : n° mínimo de máquinas sincronizadas = 4

5 : n° mínimo de máquinas sincronizadas = 5

6 : n° mínimo de máquinas sincronizadas = 6 e pelo menos 1 dos reatores desligados deve ser de TP.

26. Se uma linha de IA-TP desligada

**Então** Case n° máximo de reatores desligados =

3 : n° mínimo de máquinas sincronizadas = 3

4 : n° mínimo de máquinas sincronizadas = 4

5 : n° mínimo de máquinas sincronizadas = 5

6 : n° mínimo de máquinas sincronizadas = 6 e pelo menos 1 dos reatores desligados deve ser de TP.

27. Se uma linha de IV-IA desligada

**Então** Case n° máximo de reatores desligados =

2 : n° mínimo de máquinas sincronizadas = 3

3 : n° mínimo de máquinas sincronizadas = 4

4 : n° mínimo de máquinas sincronizadas = 5

5 : n° mínimo de máquinas sincronizadas = 6

6 : n° mínimo de máquinas sincronizadas = 6 e pelo menos 1 dos reatores desligados deve ser de TP.



28. **Se** uma linha de IV-IA e uma linha de IA-TP estão desligadas  
**Então** *Case* o número máximo de reatores desligados =  
4 : número mínimo de máquinas sincronizadas = 3  
5 : número mínimo de máquinas sincronizadas = 4  
6 : número mínimo de máquinas sincronizadas = 5 e pelo menos 1 dos reatores desligados deve ser de TP.
29. **Se** uma linha de FI-IV e uma linha de IA-TP desligadas  
**Então** *Case* n° máximo de reatores desligados =  
3 : n° mínimo de máquinas sincronizadas = 2  
4 : n° mínimo de máquinas sincronizadas = 3  
5 : n° mínimo de máquinas sincronizadas = 4  
6 : n° mínimo de máquinas sincronizadas = 5 e pelo menos 1 dos reatores desligados deve ser de TP.
30. **Se** uma linha de FI-IV e uma linha de IV-IA desligadas  
**Então** *Case* n° máximo de reatores desligados =  
1 : n° mínimo de máquinas sincronizadas = 2  
2 : n° mínimo de máquinas sincronizadas = 3  
3 : n° mínimo de máquinas sincronizadas = 4  
4 : n° mínimo de máquinas sincronizadas = 5
31. **Se** uma linha de FI-IV e uma linha de IV-IA e uma linha de IA-TP desligadas  
**Então** *Case* n° máximo de reatores desligados =  
2 : n° mínimo de máquinas sincronizadas = 1 e os reatores do circuito FI-IV devem obrigatoriamente estar conectados  
4 : n° mínimo de máquinas sincronizadas = 2 e o reator da LT IA-TP deve estar em operação.

## Apêndice 3

### Base de Conhecimento Real

#### (notação no R-TESE)

01. Se (TRUE)  
Então ("carga leve" := 0) And ("carga mínima" := 1) And  
("carga média" := 2) And ("carga pesada" := 3) And  
(ntotalreatores := 16) And (ntotalre330 := 13) And (ntotalre150 := 3))
  
02. Se (TRUE Or \_Hora)  
Então If ( (( (DATA <= 7) And (DATA >= 3)) And ( (HORA >= 0) And  
(HORA < 7) ) ), ("período carga" := "carga leve"),  
( If ( (((DATA <= 7) And (DATA >= 2)) And (((HORA >= 7) And  
(HORA < 17)) Or ((HORA >= 22) And (HORA < 24))))),  
("período carga" := "carga média"),  
( If ( ((2 <= DATA <= 7) And ((HORA > 17) And  
(HORA < 22))), ("período carga" := "carga pesada"),  
(If ( ((DATA = 2) And ((HORA >= 0) And (HORA < 7))),  
("período carga" := "carga mínima"),  
(If ( (((DATA = 1) Or (DATA = 8)) And ((HORA >= 5)  
And (HORA < 10))), ("período carga" := "carga  
mínima"),  
( If ( (((DATA = 1) Or (DATA = 8)) And  
(((HORA >= 0) And (HORA < 5)) Or  
((HORA >= 10) And (HORA < 17)) Or  
((HORA >= 22) And (HORA < 24))))),  
("período carga" := "carga leve"),  
( If ( (((DATA = 1) Or (DATA = 8)) And  
((HORA >= 17) And (HORA < 22))),  
("período carga" := "carga média"),  
(Erro := TRUE)) ))))))))

03. **Se** (TRUE Or U01.VALOR Or U02.VALOR Or U03.VALOR Or U04.VALOR Or U05.VALOR Or U06.VALOR Or U07.VALOR Or U08.VALOR Or U09.VALOR Or U10.VALOR Or U11.VALOR Or U12.VALOR Or U13.VALOR Or U14.VALOR Or U15.VALOR Or U16.VALOR Or U17.VALOR Or U18.VALOR)
- Então** ("nmaqsincr" := (U01.VALOR=1) + (U02.VALOR=1) + (U03.VALOR=1) + (U04.VALOR=1) + (U05.VALOR=1) + (U06.VALOR=1) + (U07.VALOR=1) + (U08.VALOR=1) + (U09.VALOR=1) + (U10.VALOR=1) + (U11.VALOR=1) + (U12.VALOR=1) + (U13.VALOR=1) + (U14.VALOR=1) + (U15.VALOR=1) + (U16.VALOR=1) + (U17.VALOR=1) + (U18.VALOR=1))
04. **Se** (TRUE Or "FI-IV RE 330 LI1.VALOR" Or "FI-IV RE 330 LI2.VALOR" Or "FI-IV RE 330 LI3.VALOR" Or "IV RE 330 R1.VALOR" Or "IV RE 330 R7.VALOR" Or "IA RE 330 R1.VALOR" Or "IA RE 330 R2.VALOR" Or "IA-TP RE 330 LI1.VALOR" Or "IA-TP RE 330 LI2.VALOR" Or "IV-IA RE 330 1-LI1.VALOR" Or "IV-IA RE 330 1-LI2.VALOR" Or "IV-IA RE 330 2-LI1.VALOR" Or "IV-IA RE 330 2-LI2.VALOR")
- Então** ("nreatores330" := "FI-IV RE 330 LI1.VALOR" + "FI-IV RE 330 LI2.VALOR" + "FI-IV RE 330 LI3.VALOR" + "IV RE 330 R1.VALOR" + "IV RE 330 R7.VALOR" + "IA RE 330 R1.VALOR" + "IA RE 330 R2.VALOR" + "IA-TP RE 330 LI1.VALOR" + "IA-TP RE 330 LI2.VALOR" + "IV-IA RE 330 1-LI1.VALOR" + "IV-IA RE 330 1-LI2.VALOR" + "IV-IA RE 330 2-LI1.VALOR" + "IV-IA RE 330 2-LI2.VALOR")

05. **Se** (TRUE Or  
"FI-IV RE 150 LI1.VALOR" Or  
"FI-IV RE 150 LI2.VALOR" Or  
"FI-IV RE 150 LI3.VALOR")  
**Então** ("nreatores150" := "FI-IV RE 150 LI1.VALOR" +  
"FI-IV RE 150 LI2.VALOR" +  
"FI-IV RE 150 LI3.VALOR")
06. **Se** (TRUE Or "nreatores330" Or "nreatores150")  
**Então** ("nreatores" := "nreatores330" + "nreatores150")
07. **Se** (TRUE Or nreatores330)  
**Então** ("nreatores330 OFF" := ntotalre330 - nreatores330)
08. **Se** (TRUE Or nreatores150)  
**Então** ("nreatores150 OFF" := ntotalre150 - nreatores150)
09. **Se** (TRUE Or nreatores)  
**Então** ("nreatores OFF" := ntotalreatores - nreatores)
10. **Se** ("ICAMPO U10.VALOR" < 2000)  
**Então** ("Coordenar com Furnas medidas para aumentar a corrente de campo, tais como: aumentar potência ativa e/ou reativa da máquina." := TRUE)
11. **Se** ("ICAMPO U11.VALOR" < 2000)  
**Então** ("Coordenar com Furnas medidas para aumentar a corrente de campo, tais como: aumentar potência ativa e/ou reativa da máquina." := TRUE)
12. **Se** ("ICAMPO U12.VALOR" < 2000)  
**Então** ("Coordenar com Furnas medidas para aumentar a corrente de campo, tais como: aumentar potência ativa e/ou reativa da máquina." := TRUE)

13. **Se** ("ICAMPO U13.VALOR" < 2000)  
**Então** ("Coordenar com Furnas medidas para aumentar a corrente de campo, tais como: aumentar potência ativa e/ou reativa da máquina." := TRUE)
  
14. **Se** ("ICAMPO U14.VALOR" < 2000)  
**Então** ("Coordenar com Furnas medidas para aumentar a corrente de campo, tais como: aumentar potência ativa e/ou reativa da máquina." := TRUE)
  
15. **Se** ("ICAMPO U15.VALOR" < 2000)  
**Então** ("Coordenar com Furnas medidas para aumentar a corrente de campo, tais como: aumentar potência ativa e/ou reativa da máquina." := TRUE)
  
16. **Se** ("ICAMPO U16.VALOR" < 2000)  
**Então** ("Coordenar com Furnas medidas para aumentar a corrente de campo, tais como: aumentar potência ativa e/ou reativa da máquina." := TRUE)
  
17. **Se** ("ICAMPO U17.VALOR" < 2000)  
**Então** ("Coordenar com Furnas medidas para aumentar a corrente de campo, tais como: aumentar potência ativa e/ou reativa da máquina." := TRUE)
  
18. **Se** ("ICAMPO U18.VALOR" < 2000)  
**Então** ("Coordenar com Furnas medidas para aumentar a corrente de campo, tais como: aumentar potência ativa e/ou reativa da máquina." := TRUE)
  
19. **Se** ("IPU-60 BARRA A4-KV.VALOR" > 550)  
**Então** ("Tensão da Barra 500 kV da GIS acima da máxima operativa. Providenciar redução." := TRUE)
  
20. **Se** ("IPU-60 BARRA B3-KV.VALOR" > 550)  
**Então** ("Tensão da Barra 500 kV da GIS acima da máxima operativa. Providenciar redução." := TRUE)

21. **Se** ("IPU-60 BARRA A4-KV.VALOR" < 480)  
**Então** ("Tensão da Barra 500 kV da GIS abaixo da mínima operativa. Providenciar elevação." := TRUE)
22. **Se** ("IPU-60 BARRA B3-KV.VALOR" < 480)  
**Então** ("Tensão da Barra 500 kV da GIS abaixo da mínima operativa. Providenciar elevação." := TRUE)
23. **Se** ("IPU-60 LI-FI-1 MW.VALOR" >1650)  
**Então** ("Providenciar redução de carga na LI-IPU-60Hz-FI-1." := TRUE)
24. **Se** ("IPU-60 LI-FI-2 MW.VALOR" >1650)  
**Então** ("Providenciar redução de carga na LI-IPU-60Hz-FI-2." := TRUE)
25. **Se** ("IPU-60 LI-FI-3 MW.VALOR" >1650)  
**Então** ("Providenciar redução de carga na LI-IPU-60Hz-FI-3." := TRUE)
26. **Se** ("IPU-60 LI-FI-4 MW.VALOR" >1650)  
**Então** ("Providenciar redução de carga na LI-IPU-60Hz-FI-4." := TRUE)
27. **Se** (("periodo carga" = "carga leve") Or  
("periodo carga" = "carga mínima")) And  
("RSE.VALOR" <= 3000) And  
(("nmaqsincr" = 8) Or ("nmaqsincr" = 9)) And  
("nreatores330 OFF" >= 1)  
**Então** ("Geração de ITAIPU deve ser <= 5100 MW" := TRUE)

28. **Se** ((**"IV-IA RE 330 2-LI2.VALOR"** = OFF) Or  
(**"IV-IA RE 330 2-LI1.VALOR"** = OFF)) And  
(**"IA-TP RE 330 LI2.VALOR"** = OFF) Or  
(**"IA-TP RE 330 LI1.VALOR"** = OFF))  
**Então** If(**"nmaqsincr"** <= 6), (**"FSE deve ser <= 4500 MW"** := TRUE),  
(If(**"nmaqsincr"** = 7), (**"FSE deve ser <= 4700 MW"** := TRUE),  
(If(**"nmaqsincr"** >= 8), (**"FSE deve ser <= 5200 MW"** := TRUE),  
(**"Verificar número de máquinas síncronas"** := TRUE))))))
29. **Se** ((**"GER DISP 60.VALOR"** >= 4400) And (**"nmaqsincr"** = 7))  
**Então** (**"Tensão nas máquinas = 18,9 kV"** := TRUE)
30. **Se** ((**"GER DISP 60.VALOR"** >= 4700) And (**"nmaqsincr"** > 8))  
**Então** (**"Tensão nas máquinas = 18,9 kV"** := TRUE)
31. **Se** (**"IV CAPACITOR KV.VALOR"** < 780)  
**Então** (**"Limite geração Itaipu = Limite Geração Itaipu - 300 para cada 20 kV  
abaixo de 780 kV."** := TRUE)
32. **Se** (**"IPU-60 BARRA A3-KV.VALOR"** = 0)  
**Então** (**"Disjuntores 06A34, 06U10, 06U12, 06LI1 e 06LI2 devem estar  
desligados."** := TRUE)
33. **Se** (**"IPU-60 BARRA A4-KV.VALOR"** = 0)  
**Então** (**"Disjuntores 06A34, 06TA2, 06LI4, 06LI3, 06U14 e 06U16 devem estar  
desligados."** := TRUE)
34. **Se** (**"IPU-60 BARRA B3-KV.VALOR"** = 0)  
**Então** (**"Disjuntores 86B34, 86U11, 86U13, 86LI1 e 86LI2 devem estar  
desligados."** := TRUE)
35. **Se** (**"IPU-60 BARRA B4-KV.VALOR"** = 0)  
**Então** (**"Disjuntores 86B34, 86TA2, 86LI4, 86LI3, 86U15, 86U17 e 86U8A  
devem estar desligados."** := TRUE)

36. Se (( "DJ 86LI4.VALOR" = OFF) And  
 ("DJ 86LI3.VALOR" = OFF) And  
 ("DJ 86LI2.VALOR" = OFF) And  
 ("DJ 86LI1.VALOR" = OFF) And  
 ("DJ 06LI4.VALOR" = OFF) And  
 ("DJ 06LI3.VALOR" = OFF) And  
 ("DJ 06LI2.VALOR" = OFF) And  
 ("DJ 06LI1.VALOR" = OFF))

Então ("Interligação ITAIPU/FURNAS 60Hz aberta" := TRUE)

37. Se (( "DJ 12626.VALOR" = OFF) And  
 ("DJ 12646.VALOR" = OFF) And  
 ("DJ 12656.VALOR" = OFF) And  
 ("DJ 12616.VALOR" = OFF))

Então (( "Linhas IPU-60 Hz-FI estão desenergizadas." := TRUE) And ("A tensão máxima de pré-energização é de 460 kV no sentido IPU-FI." := TRUE) And ("A tensão máxima de pré-energização é de 530 kV no sentido FI-IPU." := TRUE) And ("Se o sistema já estiver com carga, não há limitações." := TRUE))

38. Se (( "LI FI-IV 1.VALOR" = OFF) Or  
 ("LI FI-IV 2.VALOR" = OFF) Or  
 ("LI FI-IV 3.VALOR" = OFF))

Então If ( (( "nreatores330 OFF" <= 3) And ("período carga" = "carga pesada")),  
 ("IPUMAX441" := 2700) And ("RSEMAX441" := 4000) ), (If (  
 ("nreatores330 OFF" <= 3) And ("período carga" = "carga média" ), (  
 ("IPUMAX441" := 2500) And ("RSEMAX441" := 4000) ), (If (  
 ("nreatores330 OFF" <= 3) And ("período carga" = "carga leve" ), (  
 ("IPUMAX441" := 1800) And ("RSEMAX441" := 4000) ), (If (  
 ("nreatores330 OFF" <= 3) And ("período carga" = "carga mínima" ), (  
 ("IPUMAX441" := 1500) And ("RSEMAX441" := 4000) ), (If (  
 ("nreatores330 OFF" >= 4) And ("período carga" = "carga pesada" ), (  
 ("IPUMAX441" := 2700) And ("RSEMAX441" := 4200) ), (If (  
 ("nreatores330 OFF" >= 4) And ("período carga" = "carga média" ), (  
 ("IPUMAX441" := 2500) And ("RSEMAX441" := 4200) ), (If (  
 ("nreatores330 OFF" >= 4) And ("período carga" = "carga leve" ), (  
 ("IPUMAX441" := 1800) And ("RSEMAX441" := 4300) ), (If (  
 ("nreatores330 OFF" >= 4) And ("período carga" = "carga mínima" ), (  
 ("IPUMAX441" := 1500) And "RSEMAX441" := 4300) ), (Erro := TRUE)  
 ))))))))



39. Se ("LI IV-IA 1.VALOR" = OFF) Or  
 ("LI IV-IA 2.VALOR" = OFF) Or  
 ("LI IA-TP 1.VALOR" = OFF) Or  
 ("LI IA-TP 2.VALOR" = OFF))

**Então** If ( ("nreatores330 OFF" <= 3) And ("período carga" = "carga pesada") ), ( ("IPUMAX442":= 4600) And ("RSEMAX442" := 2200) ), (If ( ("nreatores330 OFF" <= 3) And ("período carga" = "carga média") ), ( ("IPUMAX442":= 4600) And ("RSEMAX442" := 1800) ), (If ( ("nreatores330 OFF" <= 3) And ("período carga" = "carga leve") ), ( ("IPUMAX442":= 4400) And ("RSEMAX442" := 1400) ), (If ( ("nreatores330 OFF" <= 3) And ("período carga" = "carga mínima") ), ( ("IPUMAX442":= 4400) And ("RSEMAX442" := 1200) ), (If ( ("nreatores330 OFF" >= 4) And ("período carga" = "carga pesada") ), ( ("IPUMAX442":= 4900) And ("RSEMAX442" := 2200) ), (If ( ("nreatores330 OFF" >= 4) And ("período carga" = "carga média") ), ( ("IPUMAX442":= 4900) And ("RSEMAX442" := 1800) ), (If ( ("nreatores330 OFF" >= 4) And ("período carga" = "carga leve") ), ( ("IPUMAX442":= 4700) And ("RSEMAX442" := 1400) ), (If ( ("nreatores330 OFF" >= 4) And ("período carga" = "carga mínima") ), ( ("IPUMAX442":= 4700) And ("RSEMAX442" := 1200) ), (Erro := TRUE) ))))))))

40. Se ("LI IV-IA 1.VALOR" = OFF) And  
 ("LI IV-IA 2.VALOR" = ON) And  
 ("IV-IA RE 330 2-LI2.VALOR" = OFF)) Or  
 ("LI IA-TP 1.VALOR" = ON) And  
 ("LI IA-TP 2.VALOR" = OFF) And  
 ("IV-IA RE 330 2-LI1.VALOR" = OFF))

**Então** ("Os reatores de barra da SE-IA (R1 e R2) devem estar ligados" := TRUE)  
 And  
 ("Se apenas um estiver ligado, o limite do fluxo entre IV-IA deve ser 2400 MW" := TRUE)

41. Se ("LI IA-TP 1.VALOR" = OFF) And  
 ("LI IA-TP 2.VALOR" = ON) And  
 ("IA-TP RE 330 LI2.VALOR" = OFF)) Or  
 ("LI IA-TP 1.VALOR" = ON) And  
 ("LI IA-TP 2.VALOR" = OFF) And  
 ("IA-TP RE 330 LI1.VALOR" = OFF))

**Então** ("Devem estar ligados pelo menos 3 reatores 330 MVar em IA - de linha ou de barra" := TRUE) And  
 ("O limite do fluxo entre IV-IA deve ser 2200 MW" := TRUE)

42. Se ((“LI IV-IA 1.VALOR” = OFF) Or  
 (“LI IV-IA 2.VALOR” = OFF)) And  
 (“LI IA-TP 1.VALOR” = OFF) Or  
 (“LI IA-TP 2.VALOR” = OFF))

Então (“FSE deve ser  $\leq 1900$ ” := TRUE) And  
 (“Dois reatores deverão estar ligados em barra em IA, podendo estar  
 desligado até um reator de linha em IA ou TP” := TRUE))

43. Se ((“LI FI-IV 1.VALOR” = ON) And  
 (“LI FI-IV 2.VALOR” = ON) And  
 (“LI FI-IV 3.VALOR” = ON) And  
 (“LI IV-IA 1.VALOR” = ON) And  
 (“LI IV-IA 2.VALOR” = ON)) And  
 (“LI IA-TP 1.VALOR” = ON) And  
 (“LI IA-TP 2.VALOR” = ON))

Então If(“nreatores OFF”  $\leq 2$ ), (“Número mínimo de máquinas sincronizadas =  
 3” := TRUE),  
 (If(“nreatores OFF”  $\leq 3$ ), (“Número mínimo de máquinas  
 sincronizadas = 4” := TRUE),  
 (If(“nreatores OFF”  $\leq 4$ ), (“Número mínimo de máquinas  
 sincronizadas = 5” := TRUE),  
 (If(“nreatores OFF”  $\leq 5$ ), (“Número mínimo de máquinas  
 sincronizadas = 6” := TRUE),  
 (If(“nreatores OFF”  $\leq 6$ ), (“Número mínimo de máquinas  
 sincronizadas = 6” := TRUE) And (“Pelo menos um dos  
 reatores deve ser de Tijuco Preto” := TRUE)),  
 (“Erro” := TRUE))))))

44. Se ((“LI FI-IV 1.VALOR” = OFF) Or  
 (“LI FI-IV 2.VALOR” = OFF) Or  
 (“LI FI-IV 3.VALOR” = OFF))

Então If(“nreatores OFF”  $\leq 3$ ), (“Número mínimo de máquinas sincronizadas =  
 3” := TRUE),  
 (If(“nreatores OFF”  $\leq 4$ ), (“Número mínimo de máquinas  
 sincronizadas = 4” := TRUE),  
 (If(“nreatores OFF”  $\leq 5$ ), (“Número mínimo de máquinas  
 sincronizadas = 5” := TRUE),  
 (If(“nreatores OFF”  $\leq 6$ ), (“Número mínimo de máquinas  
 sincronizadas = 6” := TRUE) And (“Pelo menos um dos reatores  
 deve ser de Tijuco Preto” := TRUE)),  
 (“Erro” := TRUE))))))

45. Se ((“LI IA-TP 1.VALOR” = OFF) Or  
 (“LI IA-TP 2.VALOR” = OFF))

Então If(“nreatores OFF” <= 3), (“Número mínimo de máquinas sincronizadas =  
 3” := TRUE),  
 (If(“nreatores OFF” <= 4), (“Número mínimo de máquinas  
 sincronizadas = 4” := TRUE),  
 (If(“nreatores OFF” <= 5), (“Número mínimo de máquinas  
 sincronizadas = 5” := TRUE),  
 (If(“nreatores OFF” <= 6), (“Número mínimo de máquinas  
 sincronizadas = 6” := TRUE) And (“Pelo menos um dos reatores  
 deve ser de Tijuco Preto” := TRUE)),  
 (“Erro” := TRUE)))))))))

46. Se ((“LI IV-IA 1.VALOR” = OFF) Or  
 (“LI IV-IA 2.VALOR” = OFF))

Então If(“nreatores OFF” <= 2), (“Número mínimo de máquinas sincronizadas =  
 3” := TRUE),  
 (If(“nreatores OFF” <= 3), (“Número mínimo de máquinas  
 sincronizadas = 4” := TRUE),  
 (If(“nreatores OFF” <= 4), (“Número mínimo de máquinas  
 sincronizadas = 5” := TRUE),  
 (If(“nreatores OFF” <= 5), (“Número mínimo de máquinas  
 sincronizadas = 6” := TRUE),  
 (If(“nreatores OFF” <= 6), (“Número mínimo de máquinas  
 sincronizadas = 6” := TRUE) And (“Pelo menos um dos  
 reatores deve ser de Tijuco Preto” := TRUE)),  
 (“Erro” := TRUE)))))))))

47. Se ((“LI IV-IA 1.VALOR” = OFF) Or  
 (“LI IV-IA 2.VALOR” = OFF)) And  
 ((“LI IA-TP 1.VALOR” = OFF) Or  
 (“LI IA-TP 2.VALOR” = OFF))

Então If(“nreatores OFF” <= 4), (“Número mínimo de máquinas sincronizadas =  
 3” := TRUE),  
 (If(“nreatores OFF” <= 5), (“Número mínimo de máquinas  
 sincronizadas = 4” := TRUE),  
 (If(“nreatores OFF” <= 6), (“Número mínimo de máquinas  
 sincronizadas = 5” := TRUE) And (“Pelo menos um dos reatores  
 deve ser de Tijuco Preto” := TRUE)),  
 (“Erro” := TRUE)))))))))

48. **Se** ("LI FI-IV 1.VALOR" = OFF) Or  
 ("LI FI-IV 2.VALOR" = OFF) Or  
 ("LI FI-IV 3.VALOR" = OFF)) And  
 ("LI IA-TP 1.VALOR" = OFF) Or  
 ("LI IA-TP 2.VALOR" = OFF))

**Então** If(("nreatores OFF" <= 3), ("Número mínimo de máquinas sincronizadas = 2" := TRUE),  
 (If(("nreatores OFF" <= 4), ("Número mínimo de máquinas sincronizadas = 3" := TRUE),  
 (If(("nreatores OFF" <= 5), ("Número mínimo de máquinas sincronizadas = 4" := TRUE),  
 (If(("nreatores OFF" <= 6), (("Número mínimo de máquinas sincronizadas = 5" := TRUE) And ("Pelo menos um dos reatores deve ser de Tijuco Preto" := TRUE))),  
 ("Erro" := TRUE))))))

49. **Se** ("LI FI-IV 1.VALOR" = OFF) Or  
 ("LI FI-IV 2.VALOR" = OFF) Or  
 ("LI FI-IV 3.VALOR" = OFF)) And  
 ("LI IV-IA 1.VALOR" = OFF) Or  
 ("LI IV-IA 2.VALOR" = OFF))

**Então** If(("nreatores OFF" <= 1), ("Número mínimo de máquinas sincronizadas = 2" := TRUE),  
 (If(("nreatores OFF" <= 2), ("Número mínimo de máquinas sincronizadas = 3" := TRUE),  
 (If(("nreatores OFF" <= 3), ("Número mínimo de máquinas sincronizadas = 4" := TRUE),  
 (If(("nreatores OFF" <= 4), ("Número mínimo de máquinas sincronizadas = 5" := TRUE),  
 ("Erro" := TRUE))))))

50. **Se** ("LI FI-IV 1.VALOR" = OFF) Or  
 ("LI FI-IV 2.VALOR" = OFF) Or  
 ("LI FI-IV 3.VALOR" = OFF)) And  
 ("LI IV-IA 1.VALOR" = OFF) Or  
 ("LI IV-IA 2.VALOR" = OFF)) And  
 ("LI IA-TP 1.VALOR" = OFF) Or  
 ("LI IA-TP 2.VALOR" = OFF))

**Então** If(("nreatores OFF" <= 2), (("Número mínimo de máquinas sincronizadas = 1" := TRUE) And ("Reatores de FI-IV devem estar conectados" := TRUE)),  
 (If(("nreatores OFF" <= 4), (("Número mínimo de máquinas sincronizadas = 2" := TRUE) And ("reator da LT IA-TP deve estar em operação")),  
 ("Erro" := TRUE))))

## Apêndice 4

# Estruturação de Regras em Árvores de Objetos

Apresenta-se, a seguir, um exemplo de estruturação de regras em árvores de objetos, no contexto do ambiente R-TESE.

Considera-se a seguinte lógica:

$$(x + y * (A + B) / C) \geq (C - D)$$

Esta estrutura é desmembrada em *tokens* (fichas), que são inseridos em uma lista própria, a partir da qual é feita a estruturação da árvore de objetos correspondente à lógica estudada.

*Tokens* são as menores unidades significativas de um texto, em um programa, podendo ser formados por caracteres, identificadores, números, palavras reservadas, símbolos especiais, entre outros.

Quando dois *tokens* sucessivos são utilizados em uma lógica, deve-se incluir um separador entre eles. Os separadores são definidos na estrutura do programa, e correspondem a *espaços em branco* e *operadores*.

Operadores são palavras reservadas, ou símbolos, utilizados para indicar que alguma operação deve ser realizada sobre aquele determinado conjunto de dados. Os seguintes operadores fazem parte do escopo do R-TESE:

[ , , + , - , \* , / , ^ , > , < , = , ( , ) , : , " , >= , <= , < > , \*\* , := ,

And , Or , Max , Min , GetV , GetS , Sin , Cos , Tan , Data , Hora , If , For ]

Cada operador possui uma prioridade associada, que determina como deve ser estruturada sua árvore de objetos.

Retomando-se o exemplo, a lista de *tokens* apresenta-se na seguinte forma:

[ ( , x , + , y , \* , ( , A , + , B , ) , / , C , ) , > , = , ( , C , - , D , ) ]

Alguns separadores, compostos por mais de um caracter, são reconhecidos no desmembramento da lógica, e agregados em apenas um *token*. São eles:

[ >= , <= , < > , \*\* , := ]

Após a verificação da existência destes operadores, constitui-se a seguinte lista:

[ ( , x , + , y , \* , ( , A , + , B , ) , / , C , ) , ≥ , ( , C , - , D , ) ]

Analisando esta estrutura da esquerda para a direita, e respeitando os parênteses que compõem o modelo, o R-TESE gera uma árvore de objetos a partir de seus operadores. O operador que possui maior prioridade<sup>1</sup> define quais serão seus ramos.

No exemplo, tem-se:

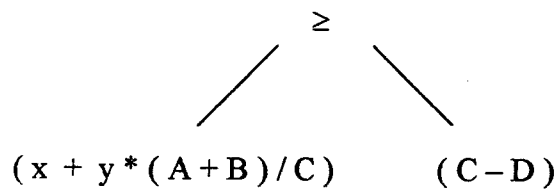
<sup>1</sup> Quanto maior a prioridade, menor o valor numérico associado.

Operador	Prioridade
+	1
-	1
$\geq$	0
/	2
*	2

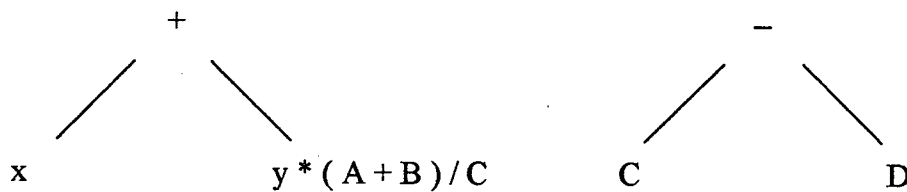
Partindo-se do operador com mais alta prioridade ( $\geq$ ), é inicializada a construção da árvore de objetos. Diferentes etapas são definidas, recursivamente, até a formação da estrutura final, onde cada *token* corresponde a um objeto da hierarquia de elementos que constituem as funções externas do ambiente especialista.

As etapas ficam assim particionadas:

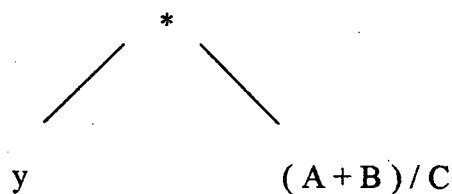
### Etapa 1: Prioridade 0



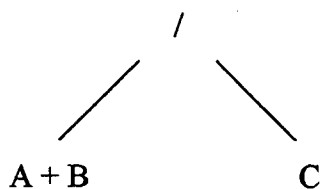
### Etapa 2: Prioridade 1



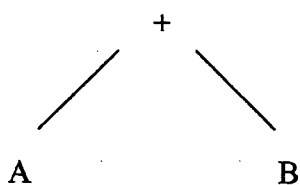
### Etapa 3: Prioridade 2



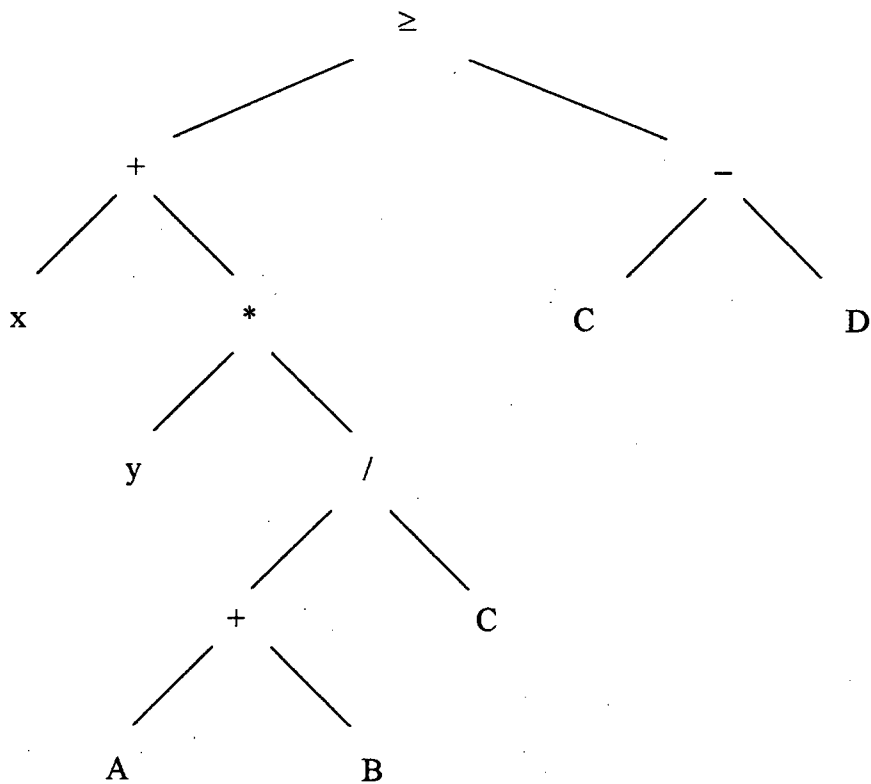
**Etapa 4: Prioridade 2**



**Etapa 5: Prioridade 1**



A árvore completa, representativa do exemplo, apresenta-se na seguinte forma:





Todas as premissas e ações das regras que compõem as Bases de Conhecimento dos Sistemas Especialistas, construídos a partir do R-TESE, são estruturadas em memória seguindo este método.

Esta estrutura permite o processamento das regras durante a inferência.

# Referências Bibliográficas

[Barr 86] Barr, Avron & Feigenbaum, Edward A. *The Handbook of Artificial Intelligence - vol. II*. Department of Computer Science Stanford University. Addison-Wesley Publishing Company, 1986.

[Beck 90] Beck Th. & Lauber, R. J. *Integration of an Expert System into a Real-Time Software System*. Institute of Control Engineering and Industrial Automation of the University of Stuttgart. Stuttgart, FRG. 11° IFAC, vol. 7, Tallinn, 8/13-17/90, USSR.

[Bittencourt 94] Bittencourt, Guilherme & Souza, Gil Fábio de. *A Rete Match Algorithm for a Multirepresentation Expert System Shell*. Proceedings of the XI Brazilian Symposium on Artificial Intelligence (SBIA 94) pp. 471-482, October, 17-21, Fortaleza, CE, 1994.

[Bittencourt 98] Bittencourt, Guilherme. *Inteligência Artificial - Ferramentas e Teorias*. Editora da UFSC, Florianópolis - SC, 1998.

[Cantù 96] Cantù, Marco. *Dominando o Delphi 2*. Makron Books, São Paulo, 1996.

[Capanema 97] Capanema, Ivana Fontanive. *Desenvolvimento de um Sistema de Supervisão em Tempo Real*. Projeto de Fim de Curso desenvolvido na Hidrelétrica de ITAIPU Binacional - Engenharia de Controle e Automação Industrial - UFSC. Florianópolis, SC, 1997.

[Charniak 85] Charniak, Eugene & McDermott, Drew. Introduction to Artificial Intelligence. Addison-Wesley Publishing Company, 1985.

[Crespo 96] Crespo, B.F.J. van Weert A. A Real-Time Expert System Shell with Progressive Reasoning, Temporal Reasoning And Fuzzy Logic. Departamento de Ingeniería de Sistemas, Computadores y Automática. Universidad Politécnica de Valencia, Spain, 1996.

[Dodhiawala 89] Dodhiawala Rajendra et alii. Real-Time AI Systems: A Definition and An Architecture. FMC Corporate Technology Center, California. IJCAI-89.

[El Khouri 91] El Khouri, Jorge Habib Hanna. Proposta de Processo de Desenvolvimento para Sistemas de Comando e Controle. Dissertação de Mestrado - Instituto Tecnológico de Aeronáutica - São José dos Campos, 1991.

[El Khouri 93] El Khouri, Jorge Habib Hanna. Sistema Especialista de Apoio ao Despacho de Carga da Usina de ITAIPU. V Encontro Regional Latino-Americano da Cigré. Ciudad del Este, Paraguay, 1993.

[Électra 97] Testing and Maintenance Procedures for Expert Systems in Power System Operation and Planning. Task Force 06.03 of Study Committee 38. Électra nº 173, August 97.

[Fernandes 95] Fernandes, Rubiara Cavalcante & Machado, Luiz Jairo B. Características e Vantagens do Sistema de Apoio à Operação de Subestações. XIII SNPTEE - Seminário Nacional de Produção e Transmissão de Energia Elétrica, Cigré-Brasil / Eletrosul, Outubro 1995, Camboriú, SC.

[Garnousset 91] Garnousset, Hervé. KHEOPS: A Real Time Oriented Expert System Generator (Language Specification). Laboratoire d'Automatique et d'Analyse des Systemes du CNRS. January 1991. Toulouse, France.

[Herreros 96] Herreros, Oscar et alii. Sistema Experto de Apoyo al Despacho de Carga. II SESEP - Cigré Paraguay, 1996.

[IEEE 93] A Tutorial Course on Knowledge-Based System Techniques with Applications to Power Systems. IEEE Power Engineering Society. Intelligent System Applications Working Group Computer and Analytical Methods Subcommittee & System Operations Subcommittee Power System Engineering Committee. October 1993.

[Ignizio 91] Ignizio, James P. Introduction to Expert Systems - The Development and Implementation of Rule-Based Expert Systems. McGraw-Hill, 1991.

[ITAIPU 89] Aspectos Técnicos do Empreendimento ITAIPU. Rio de Janeiro: ITAIPU Binacional, 1989.

[Lacerda 87] Lacerda, Paulo Neves de. Protótipo de um Sistema Especialista para Auxílio à Operação de Subestações de Alta Tensão. Dissertação de Mestrado - LCMI - UFSC. Florianópolis, SC, 1987.

[Laffey 88] Laffey, Thomas J. et alii. Real-Time Knowledge-Based Systems. AI Magazine 9(1):27-45, Spring, 1988.

[Levine 88] Levine, Robert I.; Drang, Diane E. & Edelson, Barry. Inteligência Artificial e Sistemas Especialistas. McGraw-Hill, São Paulo, 1988.

[Liempd 90] Liempd, Gidi van; Velthuijsen, Hugo & Florescu, Adriana. Blondie-III. PTT Research, Neher Laboratories. IEEE, 1990.

[Martino 98] Martino, Marcello Baptista de et alii. Expert Systems for Power System Restoration. 1998.

[Pinto 98] Pinto, Hélio B. et alii. DESPRO Expert System for Scheduled Equipment Maintenance Analysis. Eletro Evolução - pág. 46 a 50. 1998.

[Pressman 95] Pressman, Roger S. *Engenharia de Software*. Makron Books, São Paulo, 1995.

[Rabuske 95] Rabuske, Renato Antônio. *Inteligência Artificial*. Editora da UFSC, Florianópolis, 1995.

[Ribeiro 95] Ribeiro, Guilherme Moutinho; Soares, Wellington Zakhia; Torres, Germano Lambert. *Automação de Subestações Utilizando Sistemas Especialistas*. 6º ERLAC - Encontro Regional Latino-Americano da Cigré. Comitê Nacional Brasileiro/Copel - Cia. Paranaense de Energia. Foz do Iguaçu - Paraná. Junho 1995.

[Rumbaugh 97] Rumbaugh James et alii. *Modelagem e Projetos Baseados em Objetos*. Rio de Janeiro, Editora Campus, 1997.

[Sassen 90] Sassen, J.M.A. *A Modeling Technique for Real Time Knowledge Based Systems*. TNO Institute of Applied Computer Science (TNO-ITI). ESM90: European Simulation Conference Modeling & Simulation. June 10-13, 1990, Nuremberg.

[Sawyer 86] Sawyer, Brian & Foster, Dennis L. *Programming Expert Systems in Pascal*. Wiley Press, 1986.

[SCADA 95] *System of Supervision and Control by Computer (SCADA/EMS) - Technical Specifications for Elaboration of Bidding Documentation*. Foz do Iguaçu, ITAIPU, 1995.

[Sertich 97] Sertich, Antônio. *Manual de Actualizacion de la Base de Conocimientos del IOI-50 Hz*. ENES.DT, ITAIPU Binacional, Setembro 97.

[Shirai 82] Shirai, Yoshiaki & Tsujii, Jun-ichi. *Inteligência Artificial - Conceitos, Técnicas e Aplicações*. Publicações Europa-América, 1982.

[Silva 97] Silva, Victor N. A. L. da; Ribeiro, Guilherme Ferreira; Lourenço, Celia Regina S. H. Sistema Híbrido Inteligente para Diagnose no Sistema Elétrico. Artigo apresentado no VII ERLAC. Dezembro 1997.

[Tano 88] Tano, Shun'ichi; Masui, Shoichi; Nakano, Toshihiko; Mori, Kiyomi. EUREKA-II: A Programming Tool for Knowledge-Based Real Time Control Systems. Japan. International Workshop on Artificial Intelligence for Industrial Applications, 1988.

[Waterman 93] Waterman, Donald A.; Hayes-Roth, Frederick; Lenat, Douglas B. Building Expert Systems. Vol. 1. Addison-Wesley Publishing Company, 1983.

[Wright 86] Wright, M. Lattimer; Green, Milton W.; Fiegl, Gudrun & Cross, Perry F. An Expert System for Real-Time Control. SRI International, March 1986.