

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA DE PRODUÇÃO E SISTEMAS

UM SISTEMA DE TREINAMENTO PARA OPERADORES DE  
RESERVATÓRIOS BASEADO EM TÉCNICAS DE  
INTELIGÊNCIA ARTIFICIAL

DISSERTAÇÃO SUBMETIDA A UNIVERSIDADE FEDERAL  
DE SANTA CATARINA PARA A OBTENÇÃO DO GRAU DE  
MESTRE EM ENGENHARIA



0.192.419-2

UFSC-BU

Jessé Sotto Maior Lagos Neto Thé

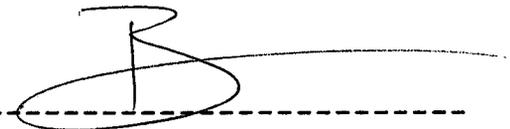
Florianópolis, Julho de 1988

DISSERTAÇÃO SUBMETIDA A UNIVERSIDADE FEDERAL DE SANTA CATARINA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA

Jessé Sotto Maior Lagos Neto Thé

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA OBTENÇÃO DE TÍTULO DE MESTRE EM ENGENHARIA

ESPECIALIDADE ENGENHARIA DE PRODUÇÃO E SISTEMAS E APROVADA NA SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO



-----  
Prof. Ricardo Miranda Barcia  
Coordenador

BANCA EXAMINADORA



-----  
Prof. Ricardo Miranda Barcia, PhD  
Presidente



-----  
Prof. Sérgio Mayerle, M.Eng



-----  
Prof. Alvaro Lezana, M.Eng



-----  
Engº Ricardo Cherem de Abreu, MSc

**A Nilbio e Lygia Thé pela Formação**

## AGRADECIMENTOS

A Cristiane P. Lagos pela paciência e compreensão ao longo destes dez anos de convivência

Ao Engenheiro Ricardo Cherem de Abreu pelo suporte dado as minhas atividades científicas e pelos estímulos.

Ao Prof. Ricardo Barcia pelo apoio e dedicação demonstrada durante este período.

A toda equipe da Divisão de Hidrologia Operacional, com especial destaque a Liu Ming, Carlos Moraes e Ricardo Neto pela incansável repetição das explicações de seus conhecimentos nesta fase de aprendizado.

Aos amigos Nilbio Filho e Amarílio Cavalcante Jr. pela torcida e fé nos meus esforços.

## ÍNDICE

Página

Lista de Tabelas.....	iv
Lista de Figuras.....	iv
Resumo.....	v
Abstract.....	vi

### Capítulos

#### 1. Introdução

1.1 Generalidades.....	1
1.2 Escopo do Trabalho.....	3
1.3 Revisão Bibliográfica.....	5
1.4 Importância do Trabalho.....	8
1.5 Organização do Trabalho.....	9

#### 2. Fundamentos de Operação de Reservatório de Usinas

2.1 Introdução.....	10
2.2 Parâmetros de Restrição.....	10
2.2.1 Nível de Restrição.....	10
2.2.2 Vazão de Restrição.....	11
2.3 Volume de Espera.....	11
2.4 Diagrama de Operação.....	13
2.5 Escala de valores de Referência dos Estados.....	15
2.5.1 Graduação de Situações.....	15
2.5.2 Compatibilidade de Decisões.....	16
2.5.3 Perda de Confiabilidade.....	17
2.5.4 Limitações nas Descargas.....	17
2.6 Procedimento de Abertura de Comportas.....	18
2.7 Procedimentos de Operação com Perda de Comunicações.....	20
2.8 Conclusão.....	22

#### 3. Fundamentos Teóricos

3.1 Introdução.....	24
3.2 Representação do Conhecimento.....	24
3.2.1 Representação do Conhecimento por Lógica.....	26

3.2.2	Estruturação do Conhecimento.....	27
3.2.3	Sistemas de Produção.....	30
3.3	Linguagem Natural.....	31
3.4	Linguagem Natural Restrita.....	33
3.4.1	Gramática G1.....	34
3.4.2	Parser Estado Máquina.....	35
3.4.3	Parser Recursivo Descendente Livre de Contexto.....	37
3.4.4	Parser Eliminador de Ruído.....	39
3.4.5	Parser Selecionado.....	40
3.5	Escolha da Ferramenta de Trabalho.....	40
3.5.1	Lisp.....	42
3.5.2	Prolog.....	43
3.5.3	SmallTalk.....	46
3.5.4	Aspectos Relevantes na Seleção da Ferramenta de Trabalho.....	46
3.5.5	Dialetos de Prolog.....	51
3.7	Conclusão.....	51

#### 4. Implementação do Modelo

4.1	Introdução.....	53
4.2	Estrutura e Modalidade do Sistema Computacional.....	53
4.3	Modelo Matemático.....	56
4.3.1	Equação da Continuidade.....	57
4.3.2	Relação Cota X Decarga do Vertedouro.....	58
4.3.2.1	Cálculo da Vazão em Lâmina Livre.....	58
4.3.2.2	Cálculo de Vazão com Abertura Parcial de Comportas.....	59
4.3.3	Relação Cota X Volume.....	61
4.3.4	Relação Volume X Cota.....	62
4.3.5	Relação Cota X Descarga do Canal de Fuga.....	63
4.3.6	Vazão Turbinada.....	65
4.3.7	Defluência Indicada.....	67
4.3.8	Modelo de Abertura das Comportas.....	68
4.4	Operação do Reservatório com Perda de Comunicações.....	71
4.5	Interface com o Usuário.....	73
4.5.1	Telas de Introdução de Novas Usinas.....	74
4.5.2	Menus de Seleção.....	74
4.5.3	Telas de Ajuda.....	76
4.5.4	Linguagem Natural Restrita.....	77
4.6	Regras Gerais do Sistema.....	77
4.7	Arquivos Auxiliares do Sistema.....	78
4.8	Conclusão.....	79

#### 5. Resultados e Discussão

5.1	Introdução.....	80
5.2	Resultados dos Testes com o Modelo Matemático.....	80
5.2.1	Resultados dos Testes com o Modelo de Vazão a Vertice.....	80
5.2.2	Resultados dos Testes com o Modelo e Vazão Turbinada.....	81
5.2.3	Resultados dos Testes com o Determinação da Cota e	

do Volume.....	83
5.2.4 Resultados dos Testes com o Ajustador de Curvas.....	86
5.3 Resultados dos Testes com o Modelo de Aberturas de Comportas.....	86
5.4 Resultados dos Testes com o Modelo de Controle de Vazão...92	
5.5 Resultados dos Testes com o Modelo de Linguagem Natural...92	
5.6 Análise do Protótipo.....	97
5.7 Análise da Ferramenta Computacional.....	99
5.7.1 Características Favoráveis da Ferramenta Adotada.....	99
5.7.2 Características Desfavoráveis da Ferramenta Adotada.....	99
5.8 Conclusão.....	101

## 6. Conclusões e Recomendações

6.1 Conclusões.....	102
6.2 Recomendações.....	104

Referências Bibliográficas.....	106
---------------------------------	-----

## Anexo

I - Dados Referentes a Usina adotada como Protótipo.....	A1
--	----

## Apêndice

I - Fonte do Sistema de Treinamento.....	Api
--	-----

## Lista de Tabelas

Tabela 2.1	- Sequência de Abertura de Comportas.....	19
Tabela 3.1	- Regras de Produção para a Gramática G1.....	35
Tabela 4.1	- Janelas de Ajuda.....	76
Tabela 4.2	- Arquivos Auxiliares do Sistema de Treinamento.....	78
Tabela 5.1	- Valores da Vazão Turbinada.....	82
Tabela 5.2	- Resultados de Cota e Volume.....	84
Tabela 5.3	- Tentativa de Melhor Ajuste para o Volume.....	87
Tabela 5.4	- Configuração das Comportas.....	89

## Lista de Figuras

Figura 2.1	- Atenuação de Uma Onda de Cheia.....	13
Figura 2.2	- Volumes de Espera de UHSS.....	13
Figura 2.3	- Quadro de Referência para Operação.....	15
Figura 2.4	- Conjunto de Regras de UHSS.....	22
Figura 3.1	- Transições Válidas no Parser Estado Máquina.....	36
Figura 3.2	- Arvore do Mecanismo do PRDLC.....	38
Figura 3.3	- Histórico de Linguagens de Inteligência Artificial.....	43
Figura 3.4	- Evolução de Custo Relativo do Software.....	48
Figura 4.1	- Esquema de Utilização do Rio Iguaçu.....	54
Figura 4.2	- Estrutura de Uma Casca.....	54
Figura 4.3	- Estrutura Esquemática do Conhecimento.....	56
Figura 4.4	- Corte de Uma Comporta de Vertedouro.....	61
Figura 4.5	- Curvas de Colina de Uma Turbina Hidrelétrica.....	66
Figura 4.6	- Diagrama de Operação Normal e em Emergência.....	69
Figura 4.7	- Divisão do Problema de Operação em Subproblemas.....	72
Figura 5.1	- Curvas Potência X Vazão Turbinada.....	83
Figura 5.2	- Curva da Cota Dado o Volume.....	85
Figura 5.3	- Curva do Volume Dado a Cota.....	85
Figura 5.4	- Curvas Obtidas da Tabela 5.3.....	88
Figura 5.5	- Desvios de Cada Tipo de Ajuste.....	88
Figura 5.6	- Exemplo da Decomposição de Sentenças pelo Parser Selecionado.....	94
Figura 5.7	- Decomposição Recursiva Executada Pelo PRDLC.....	95
Figura 5.8	- Exemplo de Estrutura Incorreta que Satisfaz a Gramática G1.....	96
Figura 5.9	- Resultado Final de Um Treinamento na Onda de Cheia de 1983.....	97

## SIMBOLOGIA

UHSS	= Usina Hidreletrica Salto Santiago
UHSO	= Usina Hidreletrica Salto Osório
UHJMF	= Usina Hidreletrica Júlio Mesquita Fº
SS	= Usina Hidreletrica Salto Santiago
SO	= Usina Hidreletrica Salto Osório
JMF	= Usina Hidreletrica Júlio Mesquita Fº
VE	= Volume de Espera
QR	= Vazão de Restrição
QN1	= Vazão de Descarga Normal 1
QN2	= Vazão de Descarga Normal 2

## RESUMO

Este trabalho apresenta um modelo para treinamento e apoio à tomada de decisão em situações de emergência no controle de reservatórios de usinas hidrelétricas. O modelo emprega técnicas de inteligência artificial para efetuar buscas no conjunto de regras de operação de um determinado aproveitamento hidráulico. É também, proposto, uma casca modular com um subconjunto de linguagem natural relacionada com operação de vertedouros.

Técnicas de representação do conhecimento que adequadamente descrevem este problema, bem como o esquema a ser adotado para linguagem natural e suas alternativas, são também abordadas.

O modelo desenvolvido foi implementado, em forma de protótipo, para a Usina Hidrelétrica de Salto Santiago. Os resultados dos testes realizados são apresentados e analisados.

## ABSTRACT

This work presents a model for training and decision support for emergency events in hydroelectric reservoir control. The model applies artificial intelligence techniques to search a operation rule set of a given reservoir. It is also proposed a modular shell with a natural language subset related to spillway operation.

The knowledge representation and the mathematical model which describes this problem, as well as the scheme for the natural language and its alternatives are also shown.

The developed model was implemented, as a prototype, for the Salto Santiago Hydroelectric Power Station. The results are presented and analysed.

## CAPÍTULO I

### INTRODUÇÃO

#### 1.1 Generalidades

A produção de energia elétrica no Brasil é baseada principalmente em aproveitamentos hidráulicos. O controle das bacias hidrográficas é uma preocupação constante, especialmente no que concerne às cheias.

Em situações de emergência a solução adotada atualmente constitui-se em um conjunto de regras de operação. Estas regras descrevem o que deverá ser executado para proteger as estruturas do barramento e a integridade da casa de força, bem como evitar danos às propriedades privadas ribeirinhas. Tal conjunto de regras estabelece um procedimento padrão a ser seguido por operadores, e está disponível em manuais e tabelas.

O controle de reservatórios exige perícia humana na tomada de decisões mesmo no caso de operação diária. Os operadores necessitam treinamento nas regras de operação do aproveitamento, especialmente no que diz respeito a situações de emergência.

No que se refere ao treinamento de operadores, estas regras são algumas vezes programadas em linguagens procedurais. Para este fim, dezenas de programas foram desenvolvidos. Entretanto, eles não são adaptáveis às diversas usinas existentes. Desta forma, os custos envolvidos no desenvolvimento de sistemas de treinamento é diretamente proporcional ao número de usinas para as quais estes venham a ser aplicados. Isto se deve à indissociabilidade entre conhecimento e controle na programação convencional.

O desenvolvimento de um sistema que simule a operação de um reservatório é extremamente interessante. Na formulação de regras de operação, a equipe técnica responsável, necessita de ferramentas para avaliação de novas regras. Logo, uma característica desejável de um sistema de simulação é a possibilidade de inserção de regras alternativas e verificação do efeito resultante. Mais especificamente, para cada nova alternativa o sistema deverá efetuar uma busca exaustiva no espaço de soluções, devendo computar o mínimo custo associado a um determinado parâmetro pré-especificado.

Existe, ainda, grande interesse em um sistema de apoio de decisão que recomende medidas e que preveja o que um elenco de atuações sobre o controle do reservatório acarretaria. Esta necessidade é decorrente do fato que os

diversos estados de riscos de operações impõe muita tensão sobre os operadores, podendo vir a diminuir sua eficácia.

Esta dissertação apresenta um protótipo de um sistema de simulação de reservatórios, para treinamento de operadores, que incorpora as características citadas, quais sejam, transportabilidades para diversas usinas, possibilidade de inserção de novas alternativas e verificação do efeito resultante, bem como o apoio à tomada de decisão.

## 1.2 Escopo do Trabalho

A simulação de reservatórios de hidrelétricas, visando tanto o treinamento como o apoio a tomada de decisão requer um alto grau de iteração com o usuário bem como flexibilidade nas regras de operação e controle do barramento, para que possa ser reaproveitado em outras hidrelétricas com diferentes exigências.

Uma análise prévia indentificou os seguintes requisitos para a satisfação destes objetivos:

- Busca em grafos com parâmetros simbólicos associados;

- Interface com linguagem natural, mesmo que seja de forma rudimentar;
- Troca de informações com o usuário, incluindo justificativa de perguntas e decisões tomadas pelo sistema;
- Lógica claramente separada de dados para permitir a utilização do sistema desenvolvido em outros barramentos;
- Modelagem matemática simples para calcular vazões ( afluentes e defluentes em função de cota).

Os itens acima, a exceção do último, são típicos de aplicações que envolvem técnicas de inteligência artificial. Por isso optou-se pelo uso desta técnica.

Uma vantagem adicional do emprego de técnicas de Inteligência Artificial, em sistemas de treinamento, é a possibilidade de se implantar mecanismos de explicações, pelo qual o sistema interage com o usuário explicando como chegou a determinada conclusão.

Buscou-se desenvolver uma "casca" altamente modular, de forma que conjuntos de regras alternativas, para cada reservatório e bacia hidrográfica, possam ser incorporadas facilmente sem interferência com a lógica do sistema. Os objetivos do sistemas exigem respostas a perguntas de diversas natureza sobre o assunto em questão. O sistema provê uma interface homem-máquina com janelas, telas de ajuda e fácil navegação entre modos de operação.

A "casca" foi desenvolvida em forma de protótipo para testar esta proposta, utilizando a Usina de Salto Santiago, situada na Bacia do Paraná-Iguaçu.

Um subconjunto de linguagem natural para este campo foi desenvolvido para facilitar a troca de informações com operadores com poucos conhecimentos em microinformática. Isto se torna crucial em operações de emergência (tanto real como simulada), quando diferentes análises de uma dada situação e de combinações de ações corretivas deverão ser geradas e a melhor alternativa selecionada e recomendada pelo sistema.

Uma abordagem baseada em uma sub-área de inteligência artificial denominada "sistemas peritos" ou "especialistas" permite que se satisfaçam todos os quesitos mencionados.

### 1.3 Revisão Bibliográfica

A estruturação da base de conhecimento deste trabalho seguiu recomendações de Keller [12] e Hayes-Roth [9] que, embora proponham alternativas conflitantes (Arquitetura Top-Down X Prototyping Scale-up), fornecem os rudimentos para que um iniciante planeje "seu curto horizonte" e crie protótipos daí por diante [9].

A extração do conhecimento do especialista, como preconizado por Barr&Feigenbaum [1] e Genaro [29], é a fase mais desgatante do processo, pois requer uma intensa interação com os especialistas da área e a organização do conhecimento. Usualmente o perito desconhece as próprias regras que adota nos momentos em que sua perícia é exigida. No caso estudado muitos destes conhecimentos estavam compilados em formas de relatórios internos [30], e detalhes gerais de hidrelétricas puderam ser revistos em Schreiber [27].

A representação do conhecimento seguiu orientações de Rich [5], Bratko [13] com formalismos do cálculo dos predicados. Procurou-se testar a base de conhecimento seguindo-se recomendações de Marcot [25] e de Salzberg [24], evitando-se, desta forma, surpresas ao longo do desenvolvimento devido à ferramenta adotada não verificar

automaticamente a consistência de regras.

Thompson [22] e Covinton&Vellino [19] são boas referências básicas na análise de ferramentas para implementação de sistemas peritos. Utilizou-se em seguida "Why List ?" da Lucid inc., Lazarev [23], Lechner [15] e Rich [5] para o selecionamento do instrumento adequado a implementação do modelo. A seleção está descrita no Capítulo 3 - Fundamentos Teóricos. Escolheu-se a linguagem Prolog para tal.

O artigo de Brittain [34] e Barr [1] apresentam genericamente esquemas para tratamento de linguagem natural e problemas e barreiras a vencer de cada um.

O manual "Turbo Prolog the Natural language of Artificial Intelligence" [26] da Borland, foi a primeira aproximação com esta linguagem. É recomendável, no entanto, para um melhor entendimento da lógica e controle do Prolog Kluzniak&Szpakowicz [8], Bratko [13] e Townsend [10]. Detalhes para programação de linguagem natural estão contidas em Bratko [13], Schildt [7] e no Geobase, e para técnicas de explanação em Prolog apenas no SANAL [26].

As tendências e aplicações de sistemas peritos são extensamente discutidos em Coombs [11] e Keller [12]. Os temas relacionados com a geração de potencia e controle de

processos por técnicas de Inteligência Artificial estão contidos em Osborne [17], Wittig [18], Kumara [20] e Milavic [21]. Os objetivos de controle deste processo (operação de reservatórios) foram revisados desde o princípio com suporte teórico destes últimos.

#### 1.4 Importância do Trabalho

Vertedouros de barragens de hidrelétricas são dimensionados para permitirem a passagem de uma enchente decamilenar. Este evento corresponde a uma enchente máxima prevista em estudos hidrológicos que poderá ocorrer a cada dez mil anos. Fica evidente a baixa probabilidade deste sinistro. Como exemplo para comparação, a grande enchente de 1983/84 na região Sul, atingiu pesadamente a bacia do Rio Iguaçu, causando um fluxo de 16000 m<sup>3</sup>/s, na Usina Hidrelétrica de Salto Santiago. Esta mesma Usina pode defluir de seu reservatório 27000 m<sup>3</sup>/s aproximadamente.

Esta explanação pode produzir a impressão de que a operação de reservatórios de hidrelétricas é algo fácil de ser executado e bastante seguro. Isto não é verdade devido a problemas operacionais e a restrições físicas. Falhas operacionais por perdas de comunicação, e defeitos mecânicos podem causar prejuízos as populações ribeirinhas a montante e a jusante, danos a outras usinas e a própria barragem.

Falhas operacionais, além do fator econômico, podem causar sérios riscos a aglomerações urbanas e, quando em cascata, podem literalmente varrer barragens a jusante, com efeito em cadeia. Tal fato já ocorreu em diversas localidades.

Estes fatores tem estimulado as empresas concessionárias de energia elétrica, responsáveis pela operação de reservatórios, a tomarem cuidados especiais. Por exemplo, a ELETROSUL através de sua Divisão de Hidrologia Operacional (DHIO), do Departamento de Operação do Sistema (DOS), vem procurando ferramentas para treinar e dar apoio de decisão aos operadores de suas Usinas.

A proposição aqui apresentada visa atender esta demanda.

## 1.5 ORGANIZAÇÃO DO TRABALHO

No capítulo 2 são apresentados os fundamentos básicos de operação de reservatório. Os conceitos principais que devem ser observados, bem como os procedimentos de operação, em forma de regras, também são explanados.

O capítulo 3 constitui a base e estrutura para construção do modelo aqui desenvolvido. São descritas técnicas de representação procedurais e declarativas, por lógica dos predicados. A disponibilidade de ferramentas computacionais, vantagens e desvantagens de cada uma e sua relação são apresentados. A linguagem de inteligência artificial escolhida como ferramenta de trabalho é discutida apenas com profundidade suficiente para compreensão dos mecanismos de inferência presentes no modelo.

No capítulo seguinte o problema em estudo é esquematizado em termos de requisitos desejáveis, operações a realizar e restrições que devem ser atendidas. Um modelo matemático para processamento numérico do contínuo fluxos afluyente-defluente, armazenamento, vazão turbinada, regras de abertura de comportas, entre outros, são também expostos.

No quinto capítulo o modelo implementado é testado e seus resultados discutidos. Sob diversos estados meteorológicos foram simulados casos onde há ausência de comunicação da usina com o resto do sistema elétrico.

No último capítulo são destacadas as contribuições e limitações desta dissertação além de apresentadas sugestões e recomendações para futuros trabalhos.

## CAPÍTULO II

### FUNDAMENTOS DE OPERAÇÃO DE RESERVATÓRIOS DE USINAS

#### 2.1 INTRODUÇÃO

Neste capítulo são introduzidos os conceitos básicos de operação de reservatórios de usinas hidrelétricas. Faz-se observações dos cuidados operativos incorporados as regras gerais de operação do reservatório no contexto de um aproveitamento em cascata.

São apresentados os estados de operação que definem os procedimentos a serem seguidos por operadores do reservatório sob determinado estado de risco, seja este mecânico ou hidrológico.

#### 2.2 Parâmetros de Restrição

##### 2.2.1 Nível de Restrição

Um nível de restrição corresponde a uma cota que, caso atingida no enchimento do reservatório causará, prejuízos ou riscos. Estes danos podem ser na forma de inundação a

---

A filosofia e regras aqui apresentadas são parte do acervo técnico da operação da ELETROSUL contidos em 1301 e 1371.

montante de cidades ou da segurança física do barramento.

### 2.2.2 Vazões de Restrição

Vazões de restrição são defluências máximas que caso ultrapassadas podem danificar barramentos subsequentes, bloquear cursos d'água (riachos ou até mesmo o canal de fuga da própria usina) e causar inundações. O conjunto de estados de operação pode possuir várias restrições que serão rompidas em sequência ordenada de possíveis danos materiais, sendo o mais sério o que impõe risco a integridade física da barragem. Neste último caso a descarga do reservatório é efetuada em lâmina livre, ou seja permite defluir sem acréscimo de nível a máxima cheia teórica citada no ítem 1.3.

### 2.3 Volumes de Espera

O amortecimento de ondas de cheias, com a finalidade de evitar o rompimento de restrições de vazões, é alcançado pelo artifício do volume de espera. Este representa diferença de cotas não preenchidas nos reservatórios. A figura 2.1 apresenta o efeito de atenuação de uma onda de cheia pelo volume de espera. A figura 2.2 apresenta os dois volumes de espera da Usina de Salto Santiago, utilizada como protótipo.

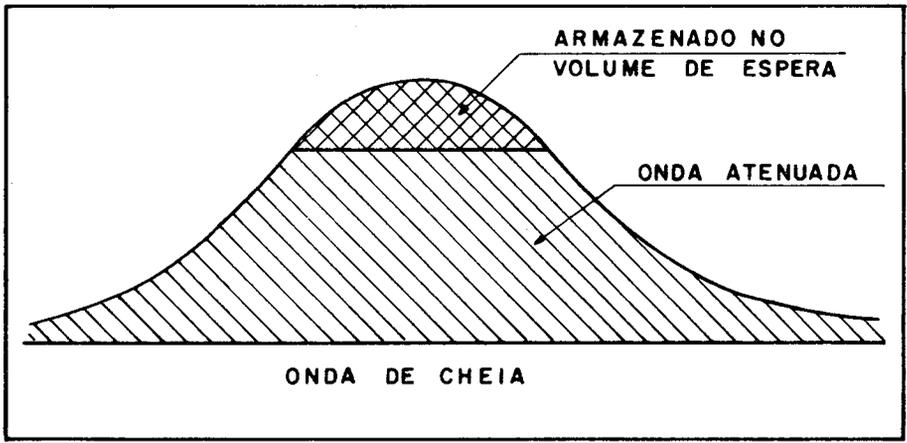


Figura 2.1 Atenuação de Uma Onda de Cheia

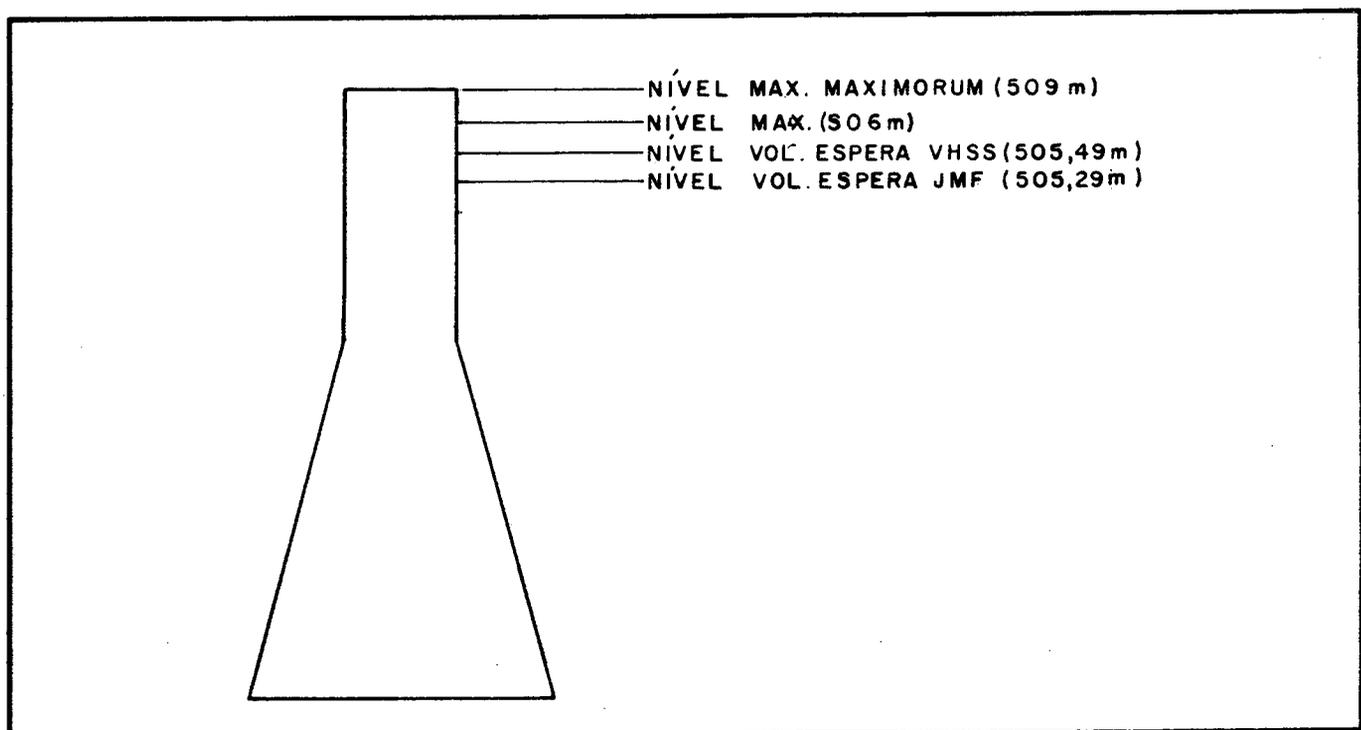


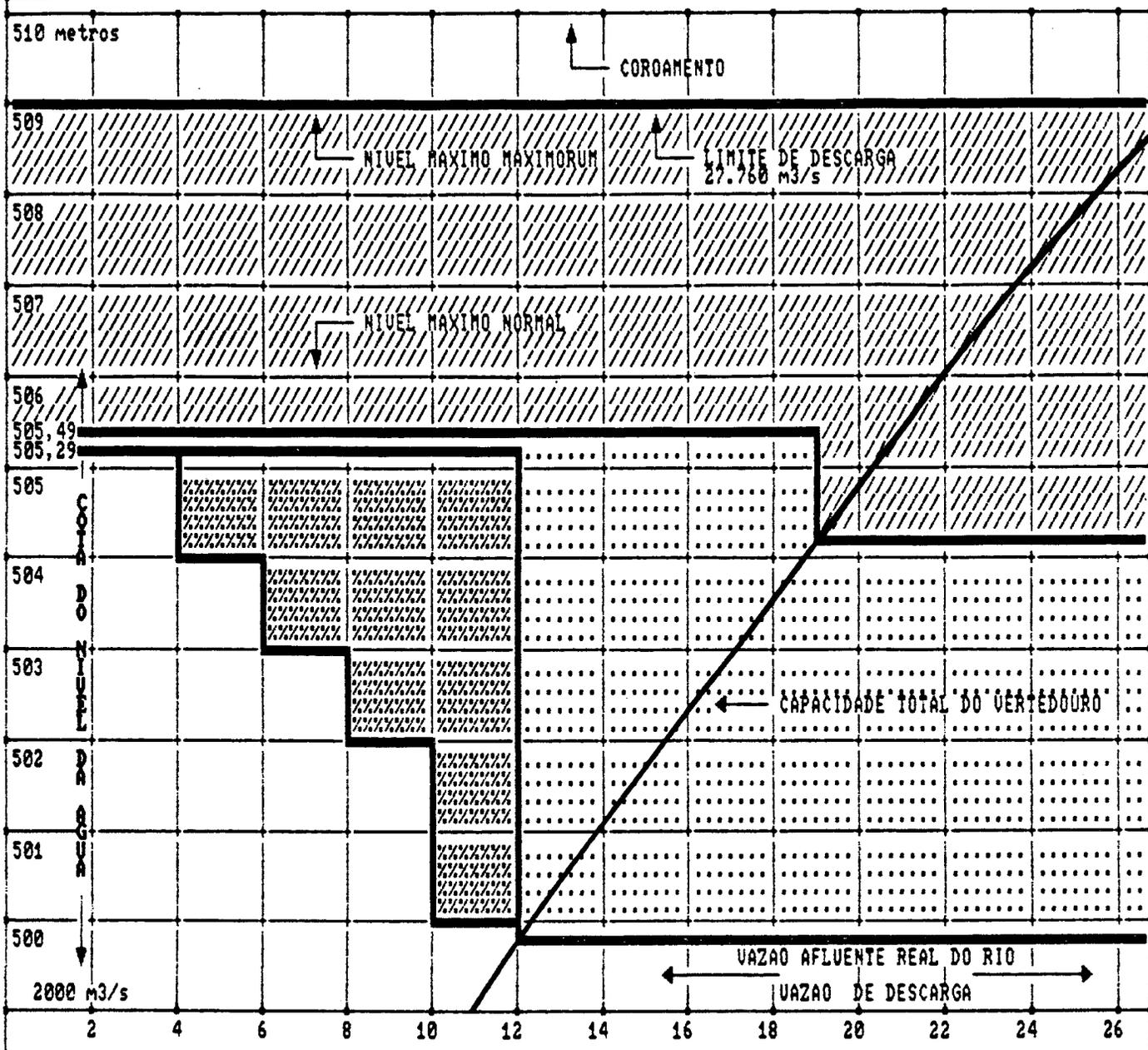
Figura 2.2 - Cotas dos Volumes de Espera de UHSS

## 2.4 Diagramas de Operação

A utilização de Diagramas de Operação proporciona meios de anteceder a tomada de decisão de se aumentar a defluência, em situações nas quais o reservatório se encontra operando abaixo dos níveis do volume de espera ou abaixo do máximo normal.

Essa antecedência faz com que ao serem atingidos aqueles níveis, a vazão defluente esteja em valores próximos ao da afluência, evitando o emprego de taxas elevadas de variação na defluência ou a violação dos volumes de espera, mantendo descargas abaixo dos valores de restrição.

As vazões defluentes, indicadas nos Diagramas de Operação, são função dos níveis e afluências ao reservatório. São valores mínimos a serem descarregados servindo como orientação básica nos casos de não se dispor de informações mais confiáveis, tais como dados hidrometeorológicos e previsões de vazões afluentes ao aproveitamento em questão. A figura 2.3 apresenta um Quadro de Referência para Operação de uma usina de concessão da ELETROSUL. Estes quadros de referência, servem para localizar o estado hidrológico na Escala de Valores de Referência.



<b>RESTRICOES</b>	QRssJMF ~ 12.000 m³/s.	Taxa de Descarga: 600 m³/s X 1 hora.
	QRss (canal de fuga) ~ 19.000 m³/s.	Nivel Operativo: 505,29 m; variacao ate -19 cm.

FAIXA	ESTADO	DECISAO	INSPECOES	RECEBER INFORMACOES DE:			
				INFORMACOES	U N G B M	U H J M F	PREVISOES
				COSI (UNSO/COPEL)	AFLUENCIA EM FOZ DO AREIA	AFLUENCIA EM RIO CHOPIN	AFLUENCIA NA BACIA
BRANCA	NORMAL	TURN DO COSI	8 EM 8 HORAS	Q < 4000 m³/s 1 vez ao dia	Q < 2000 m³/s 1 vez ao dia	Q < 2000 m³/s 1 vez ao dia	Q < 4000 m³/s 1 vez ao dia
//////	ATENCAO	OPERACAO DA USINA	8 EM 8 HORAS	4000 < Q < 6000 8 em 8 horas	2000 < Q < 4000 4 em 4 horas	2000 < Q < 3000 8 em 8 horas	4000 < Q < 9000 2 vezes ao dia
.....	ALERTA	USINA	4 EM 4 HORAS	6000 < Q < 10000 4 em 4 horas	4000 < Q < 6000 2 em 2 horas	3000 < Q < 4000 4 em 4 horas	9000 < Q < 14000 3 vezes ao dia
//////	EMERGENCIA	COMISSAO DE EMERGENCIA	1 EM 1 HORA	Q > 16000 m³/s 1 em 1 hora	Q > 6000 m³/s 1 em 1 hora	Q > 4000 m³/s 1 em 1 hora	Q > 14000 m³/s TEMPO REAL

**QUADRO DE REFERENCIA PARA OPERACAO**

## 2.5 Escala de Valores de Referência

A regularização e ações preventivas de segurança são dependentes do estado hidráulico do reservatório, que pode ser perfeitamente definido por :

- Nível de Água no Reservatório
  
- Vazões Afluentes.

### 2.5.1 Graduação de Situações

Condições da Escala de Valores de Referência são peculiares da bacia e de um aproveitamento particular estabelecendo uma graduação de situações, abaixo descritas :

- Normal :

Sem verter ou vertendo por conveniências operativas. As operações de comportas, em geral, são uma função das necessidades de geração econômica. Os Riscos de danos são nulos.

- Atenção :

Acionamento dos mecanismos de descargas do aproveitamento. Caracteriza-se pela existência de riscos de danos a jusante. Existe riscos de quebra de restrição que envolvam terceiros.

- Alerta :

Neste estado já ocorreram danos a terceiros e poderá consumir quebra de outras restrições a jusante.

- Emergência :

Esta escala tem início quando surgem ameaças às instalações da usina ou grandes responsabilidades específicas a resguardar a jusante (patrimônio de terceiros, vidas humanas). A operação é voltada para a proteção desses encargos e mais remotamente à preservação da própria barragem.

#### 2.5.2 Compatibilização de Decisões

Para casos de aproveitamentos em cascata, sempre que um reservatório passar da situação Normal para Atenção, o reservatório imediatamente a jusante, por motivos de coerência operativa e de decisões, deverá acompanhar o

reservatório de montante na ascensão da escala de graduação, passando também de Normal para Atenção, mesmo ainda se encontrando dentro dos limites da faixa Normal.

### 2.5.3 Perda de Confiabilidade

Havendo redução de confiabilidade na operação do reservatório, em qualquer situação, a classificação deverá, em função do estado hidráulico, subir na escala de gravidade. A perda de confiabilidade poderá advir de :

- Redução de recursos humanos;
- Perda de fonte principal para alimentação dos motores das comportas;
- Defeito no acionamento de comporta;
- Falta de comunicação da Usina com órgãos externos;
- Restrições imprevistas que reduzam a plena utilização dos vertedouros.

### 2.5.4 Limitações nas Descargas

Embora a graduação de situação seja sempre definida em função de vazões afluentes, a responsabilidade por descargas também acompanha os limites dessas escalas, não podendo os responsáveis de cada faixa descarregar além da afluência limite da faixa correspondente (a cada aproveitamento). Este procedimento visa resguardar usinas e propriedades a jusante

de uma onda de cheia muito elevada.

## 2.6 Procedimento de Abertura das Comportas

Operar um reservatório é, em última instância, manter um comando efetivo sobre as comportas do vertedouro. Estas deverão ser manobradas de maneira a evitar prejuízos no equipamento e em sua estrutura suporte. As recomendações principais tem haver com o afogamento mínimo das comportas e sua sequência de manobra. A sequência de abertura de comportas tem por finalidade defluir uma lâmina d'água estável. Fluxos instáveis no vertedouro causa esforços a montante, no vertedouro e a jusante do barramento. Como exemplo, a tabela 2.1 apresenta a sequência de manobra da Usina Hidrelétrica de Salto Santiago.

Tabela 2.1 - Sequência de Abertura de Comportas

I - Sequência de Operação do Vertedor com uma Comporta Impedida.

Comporta impedida

Sequência de Abertura

1

8 7 6 5 4 3 2

2	1	8	7	6	5	4	3
3	8	7	6	5	4	1	2
4	8	7	6	5	1	2	3
5	1	2	3	4	8	7	6
6	1	2	3	4	5	8	7
7	8	1	2	3	4	5	6
8	1	2	3	4	5	6	7

II - Sequência de Operação do Vertedor com Duas Comportas Impedidas

Comportas Impedidas		Sequência de Abertura					
1	2	8	7	6	5	4	3
1	3	8	7	6	5	4	2
1	4	8	7	6	5	3	2
1	5	8	7	6	4	3	2
1	6	8	7	5	4	3	2
1	7	8	5	4	6	3	2
1	8	5	4	6	3	7	2
2	3	1	8	7	6	5	4
2	4	1	8	7	6	5	3
2	5	1	8	7	6	4	3
2	6	1	8	7	5	4	3
2	7	1	8	5	4	6	3
2	8	1	5	4	3	6	7

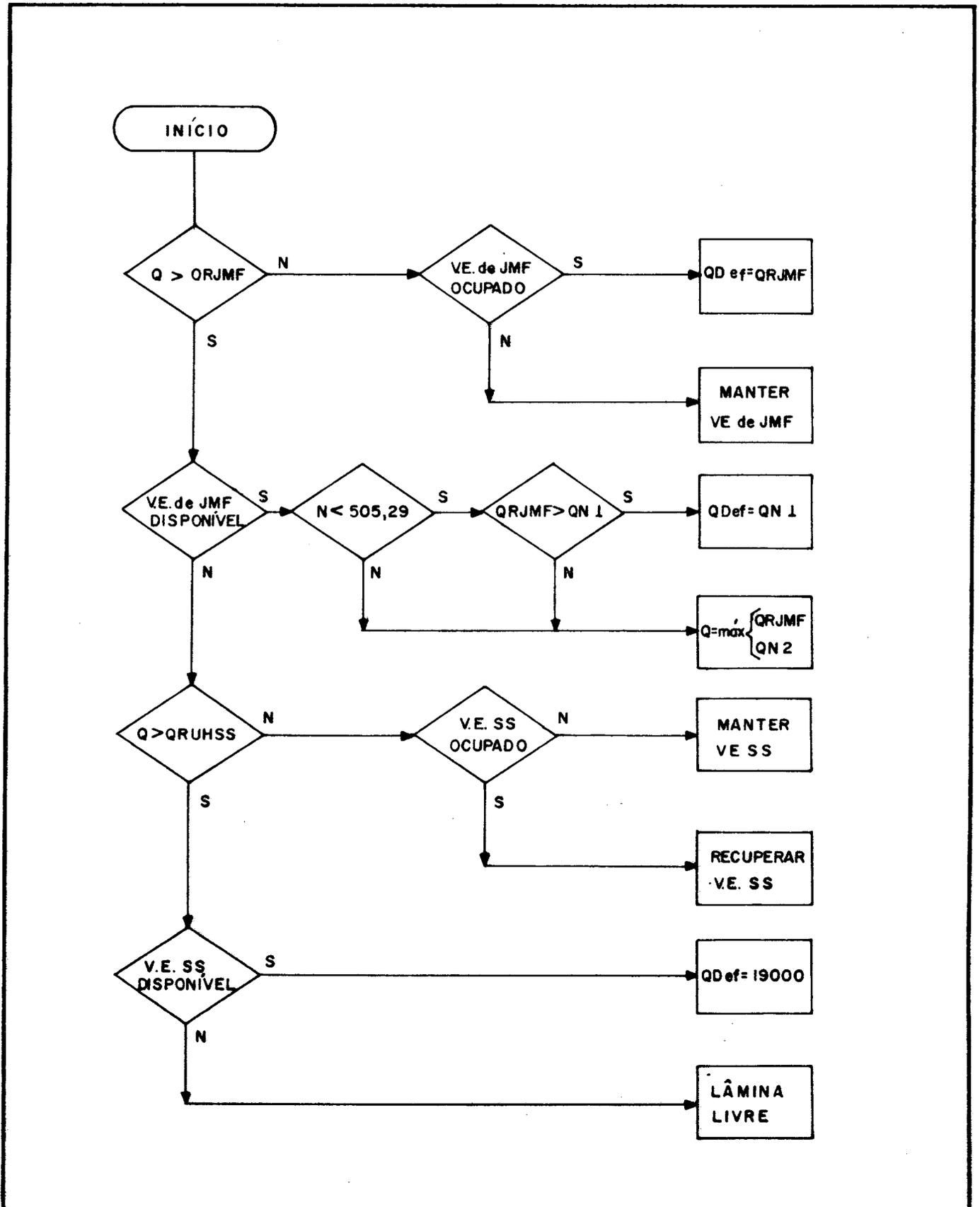
3	4	1	8	2	7	6	5
3	5	1	8	2	7	6	4
3	6	1	8	2	7	5	4
3	7	1	8	2	4	5	6
3	8	1	2	4	5	6	7
4	5	1	8	2	7	3	6
4	6	1	8	2	7	3	5
4	7	1	8	2	3	5	6
4	8	1	2	3	5	6	7
5	6	1	8	2	7	3	4
5	7	1	8	2	3	4	6
5	8	1	2	3	4	6	7
6	7	8	1	2	3	4	5
6	8	1	2	3	4	5	7
7	8	1	2	3	4	5	6

OBS.: Para o fechamento seguir sequência inversa.

## 2.7 Procedimentos de Operação com Perda de Comunicações

A perda de comunicação da Usina com o restante do sistema é a ocorrência que mais exige da perícia de um operador. A operação do reservatório será efetuada tendo por base apenas a vazão de descarga, a de afluência medida como a variação do nível em um intervalo de tempo e pelo próprio nível presente no reservatório. A figura 2.4 esquematiza um conjunto de regras para a Usina Hidrelétrica de Salto Santiago (UHSS).

Figura 2.4 - Conjunto de Regras de UHSS



## 2.8 Conclusões

Neste capítulo apresentou-se os conceitos básicos e terminologia necessários ao entendimento de operação de reservatórios. No próximo capítulo serão apresentados os fundamentos teóricos para a compreensão da implementação do modelo.

## CAPÍTULO III

### FUNDAMENTOS TEÓRICOS

#### 3.1 Introdução

Neste capítulo é apresentada a fundamentação teórica que viabiliza a formulação de um sistema para treinamento de operadores de hidrelétricas. A linguagem natural, um componente importante na fase de treinamento de operadores, é apresentada, bem como alternativas para sua implementação. Ferramentas, para um melhor desenvolvimento do sistema, são analisadas e explicações dos mecanismos da linguagem selecionada são realizadas no final do capítulo.

#### 3.2 Representação do Conhecimento

A representação do conhecimento visa mapear o espaço de problemas ao espaço de soluções. Este processo transforma objetos do mundo real em representações abstratas. A adequação de uma representação de entidades do espaço de problemas objetiva aproximá-lo, conceitualmente, ao espaço solução, facilitando este mapeamento.

A aplicação de métodos de representação do conhecimento coloca o "mundo real reduzido" em formas de estruturas de dados que possam ser logicamente consistentes

e processáveis por ferramentas computacionais.

A ausência de métodos para representação dos conhecimentos, que se deseje codificar, obtem como produto final explosões combinatoriais ou envolvimento de buscas em laços infinitos. A partir do momento que o engenheiro de conhecimento compreende os objetivos definidos, se inicia a busca de fatos, dados e regras para que se possa atingir uma conclusão possível.

A representação do conhecimento deverá ser estruturada visando sua inserção no sistema perito a ser desenvolvido. Esta é uma tarefa do engenheiro de conhecimento, que se assemelha a do analista de sistemas em processamento de dados convencional.

A solução de problemas complexos encontrados no mundo real, necessita uma grande massa de conhecimentos e alguns mecanismos para manipulá-los e criar soluções para diversas situações. A manipulação de conhecimentos necessita um formalismo adequado a cada problema inserido em um determinado domínio. Para que seja funcional é interessante que existam funções mapeando fatos a suas representações e, reciprocamente, representações a fatos. Ressalta-se que estas funções não levam ponto a ponto, mas regiões de um espaço de problemas a uma região no espaço solução.

### 3.2.1 Representação do Conhecimento por Lógica

A bibliografia consultada, especialmente Barr [1] e Rich [5], são unânimes ao apresentar as inúmeras vantagens da representação em forma de lógica. A mais importante é a possibilidade de se obter novos conhecimentos a partir daqueles existentes, assumidos como verdadeiros. Este é o mecanismo de Dedução Matemática. Kowalski [33] sugeriu a aplicação de lógica, argumentando que matemática não é diferente de tarefas complexas, as quais requerem mecanismos confiáveis de dedução e uma base heurística para controlar uma busca, que de outra forma seria completamente impossível. Esta impossibilidade é advinda de uma explosão combinatorial das alternativas a serem exploradas.

Entre os formalismos lógicos, o cálculo dos predicados, ou lógica dos predicados ou lógica de primeira ordem, tem sido preferido para representação por ser de fácil compreensão e utilização, e por existir procedimentos próprios de decisão. Uma vantagem adicional de relevância é que o cálculo dos predicados permite a representação de fatos e objetos por simples assertivas.

O procedimento de decisão, neste caso, é denominado resolução. A resolução consiste em um procedimento de prova de fatos ou associação de fatos, por refutação. A sua eficiência está associada ao fato de que as assertivas foram convertidas a uma forma conveniente, isto é, um padrão do cálculo dos predicados. As principais formas de controle da resolução, serão vistas na seção 3.5.2, e são a unificação e o caminhamento retrospectivo.

### 3.2.2 Estruturação do Conhecimento

Até mesmo uma boa representação poderá levar a uma explosão combinatorial quando uma busca for realizada em sua base de regras. Um sistema perito não poderá permitir que este fato venha prejudicar sua eficiência. A estruturação do conhecimento obedece técnicas que minimizam, ou mesmo eliminam este fenômeno inconveniente. A técnica, indicada por Richi5i, agrupa estruturas complexas de conhecimento, em um domínio particular, de forma que estes atendam as seguintes características desejáveis :

- Adequação representacional - habilidade para representar os tipos de conhecimento no domínio;

- Adequação inferencial : habilidade para manipular as estruturas de representação de forma a obter novos fatos a partir de sua base de conhecimento;

- Eficiência inferencial : habilidade de incorporar à estrutura do conhecimento, informações que direcionem os mecanismos de inferência a caminhos promissores evitando, também, procuras desnecessárias.

Técnicas para atender estas características podem ser divididas em dois grandes grupos: os declarativos e os procedurais. Os declarativos, da qual faz parte o cálculo dos predicados, separa conhecimento dos procedimentos de controle, e tem as seguintes vantagens :

- Cada fato necessita ser expresso somente uma vez, independentemente dos diferentes modos que venha a ser utilizado;

- Em decorrência, fica fácil adicionar ou modificar regras ao sistema, pois não há alterações nos demais fatos e no respectivo controle.

Por sua vez as técnicas procedurais possuem a grande parte de seu conhecimento na forma de rotinas para a sua manipulação. As principais vantagens deste grupo estão

listadas abaixo:

- Fácil de representar raciocínios probabilísticos;
- Fácil de representar heurísticas.

A decisão do esquema representativo é realizado em conjunto com a escolha da ferramenta de trabalho, pois a cada esquema corresponde um instrumento distinto. Os principais esquemas representativos são segundo Barr [1] :

- Sistemas de Produção;
- Redes Semânticas;
- Primitivas Semânticas;
- Representações Diretas ( analógicas );
- Quadros e Textos ( Frames e Scripts ).

Entre estes esquemas, apenas o "Sistemas de Produção" e o "Quadros e Textos" são de vasta aplicação em sistemas especialistas. Este trabalho descreverá apenas o primeiro esquema por dele fazer uso. Redes semânticas foram analisadas para uso em linguagem natural. No entanto esta última alternativa foi descartada em favor de um esquema mais simples e eficaz. Descrições sucintas sobre os demais esquemas poderão ser encontrados em Barr [1], Rich [5] e Townsend [10].

### 3.2.3 Sistemas de Produção

O sistema de produção é composto por regras de produção, as quais são simples assertivas do tipo

: IF (condição) : THEN (ação) !.

O processamento é cíclico, verificando qual regra, ou conjunto de regras, são aplicáveis. Caso ocorra conflito ou empate entre regras o sistema deverá saber como resolvê-lo.

Sistemas de produção possuem diversas vantagens, tais como modularidade e uniformidade de regras, e algumas desvantagens, como dificuldade para depuração. Devido a esta desvantagem torna-se difícil trabalhar com regras de produção à medida que a base de conhecimentos cresce. No entanto, linguagens computacionais, organização no desenvolvimento do programa e persistência têm permitido o emprego de regras de produção em sistemas peritos históricos, como o MYCIN, TEIRESISAS, STUDENT, GPS e SHRDLU, entre outros.

### 3.3 Linguagem Natural

Em 1949, Warren Weaver propôs o uso de computadores em problemas de tradução. A proposta consistia em armazenar-se um dicionário bilingue e regras de sintaxe dos idiomas em sua área de memória. Assim a tradução consistiria na leitura de frases em uma língua e após um simples processamento de troca de palavras por seus respectivos sinônimos com a aplicação de regras na língua destinatária. Esperava-se que uma tradução perfeita pudesse assim ser concluída. Este conceito foi abandonado quando experimentos neste campo falharam drasticamente.

Atualmente reconhece-se que o problema de linguagem natural é um dos mais complexos no campo da inteligência artificial. Diversos tipos de conhecimentos estão envolvidos nesta tarefa, tais como estrutura de sentenças, o significado das palavras, modelagem de recepção de informação, regras de conversação e um extenso campo de conhecimentos gerais sobre o mundo. Como ilustração de problemas de contexto é aqui reproduzido um exemplo de Brittain [34] sobre um dos empecilhos de linguagem natural, a ele associado, a ambiguidade :

" João viu uma mulher no parque com um telescópio "

Existem cinco interpretações para esta frase. Entre elas " João que estava no parque viu uma mulher andando com um telescópio (estando ela no parque ou não)" ou " João viu, por intermédio de um telescópio, uma mulher no parque (podendo ele estar ou não no parque) ".

A estrutura de uma linguagem natural é analisada tomando-se sua gramática e um processo de codificação para captar o significado pretendido, a análise sintática (ou parsing). A gramática, como entendida em inteligência artificial, é um esquema para combinação de palavras, de acordo com regras de sintaxe para formação correta de frases. Existem, atualmente, uma infinidade destas gramáticas em experiência no processamento natural Peat:44:.

O parsing, por sua vez, utilizando uma dada gramática, separa a estrutura da frase em elementos básicos, na tentativa de "compreender" o significado da expressão idiomática. O parser implantado neste trabalho tem sua teoria explicada a seguir e sua implementação descrita no quarto capítulo.

### 3.4. Linguagem Natural Restrita

Devido às dificuldades da linguagem natural, os sistemas peritos fazem uso do que se convencionou chamar linguagem natural restrita ou sub-linguagem natural. Esta é um subconjunto, em domínio específico, da linguagem natural, e é extremamente mais fácil de implementar e, se realizado corretamente pouco deixa a desejar. Este recurso foi desenvolvido com sucesso por diversos programas já na década de 60. Entre os mais conhecidos cita-se o ELIZA, o STUDENT e o SAD-SAM, entre outros.

Existem diversas proposições para a consecução deste objetivo, quais sejam :

- Parser Estado-Máquina de LN (State-Machine Parser);
- Parser Recursivo-Descendente Livre de Contexto (Context-Free Recursive-Descendent NLP Parser);
- Parser Limpa-Ruído ( Noise Disposal Parser).

Todas as proposições representam regras que de alguma forma captam a informação que se deseja obter, ou o comando a ser realizado. Adicionalmente a estas regras, é essencial um vocabulário condicionando palavras a seu contexto, o que dependerá do esquema adotado.

### 3.4.1 Gramática G1

A gramática G1 é uma adaptação da "G1 Grammar", em inglês. Visando sua adequação ao português ocorrerão inversões na sequência original legal em que as palavras aparecem. Este é o caso, por exemplo, dos adjetivos. Esta gramática é constituída de uma sintaxe simples e é empregada como regra básica nos parsers para verificação de associação válida de palavras contidas em frases. Todos os parser, apresentados neste trabalho, seguem a idéia central desta gramática.

A seguir serão listadas as regras da gramática G1 :

- Todos os adjetivos deverão vir após um substantivo ao qual dão qualificações;
- Todos os advérbios deverão seguir um verbo, a qual modificam;
- Todas as frases deverão ser finalizadas com um ponto.

Esta gramática necessita também um vocabulário associando cada palavra a seu tipo, isto é, se substantivo, verbo, advérbio, etc..

A tabela 3.1 apresenta as regras de produção para esta gramática.

Tabela 3.1 Regras de Produção para Gramática G1.

Frase = NP + VP

NP --> Determinante + Substantivo

NP --> Determinante + Adjetivo + Substantivo

NP --> Preposição + NP

VP --> Verbo + NP

VP --> Verbo + Adverbo + NP

VP --> Verbo + Adverbo

VP --> Verbo

#### 3.4.2 Parser Estado-Máquina

A mudança de estado, que se refere este ítem, é a da passagem do estado de uma sentença a outro pela adição de mais uma palavra. Este parser se utiliza de um grafo direcionado que verifica transições válidas de estado a estado como mostra a figura 3.1 (Schildt:71).

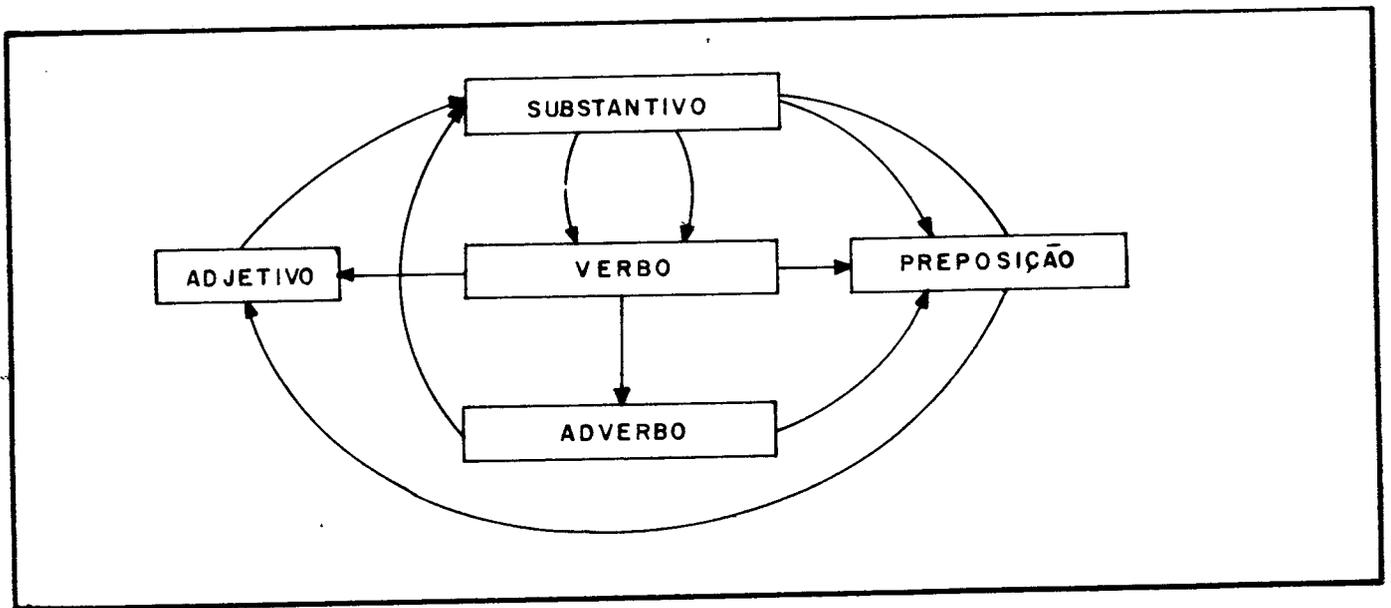


Figura 3.1 - Transições Válidas no Parser Estado-Máquina

O parser Estado-Máquina é ideal para algumas aplicações de recuperação de informações em base de dados e JCLs (Job-Control Languages), devido aos poucos tipos de frases utilizados e poucas variações de estados a analisar.

No entanto há desvantagens nesta alternativa tais como:

- Necessita muitas cláusulas para efetuar uma transição de estado;
- Não registra como atingiu um estado particular;
- Serve apenas para verificar se um comando ou frase foi fornecido corretamente, e se as palavras empregadas estão em sua base de dados.

#### 3.4.3 Parser Recursivo-Descentente Livre de Contexto (PRDLC)

O PRDLC usa regras de produção para analisar uma sentença, a qual é composta por vários itens, os quais por sua vez são compostos por outros itens. A sentença é quebrada pelo PRDLC até que se atinja seus elementos atômicos : substantivos, adjetivos, advérbios, verbos, preposições e artigos. A figura 3.2, adaptada de Schildt [7], apresenta esquematicamente este mecanismo.

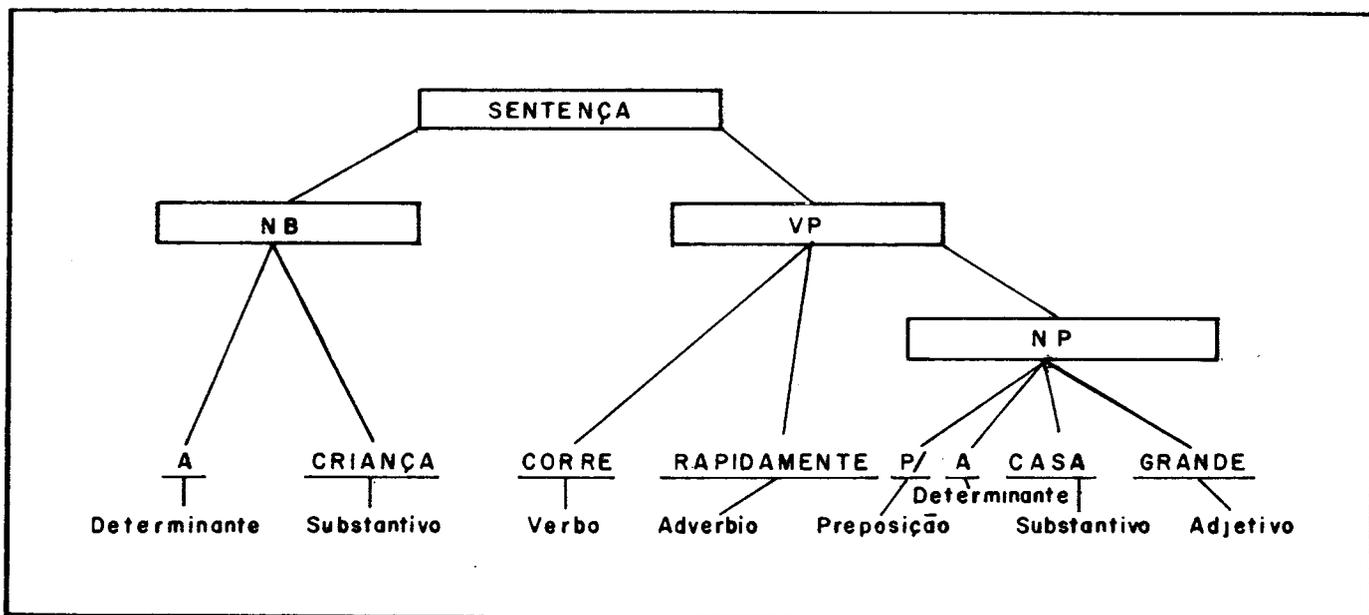


Figura 3.2 - Arvore do Mecanismo do PRDLC

A coleção de regras de produção é vasculhada por rotinas (meta-regras) recursivamente até que o objetivo final, mencionado acima seja atingido.

As principais vantagens do PRDLC são :

- Fácil de implementar em PROLOG;
- Manipula a sentença em todos os níveis ( palavra --> frase);
- Registra histórico de dissecação.

Possui como maior desvantagem o fato de ser difícil sua construção à medida que o subconjunto da linguagem desejada cresce.

#### 3.4.4 Parser Eliminador de Ruído

O parser Eliminador de Ruído (ou Limpa Ruído) é uma variante do PRDLC e, embora mais simples, procura minimizar sua maior deficiência. Neste caso o parser procura palavras chaves contidas em uma frase e todas as demais palavras que não estejam contidas na base de dados (vocabulário) são simplesmente ignoradas. Apresenta a seguinte vantagem e desvantagem :

- Mais fácil de implementar quando comparado aos demais parser e captura o significado da mensagem rapidamente;

- Aceita frases completamente esdrúxulas, desde que conttenham as palavras chaves de seu vocabulário.

#### 3.4.5 Parser Selecionado

O parser, ou analisador sintático, escolhido para este trabalho foi o Parser Recursivo Descendente Livre de Contexto. Esta seleção foi motivada pelas vantagens já expostas e a existência de implementações (171 e 1261).

#### 3.5 Escolha da Ferramenta de Trabalho

Rich 151 acredita que o processo para programar aplicando técnicas de inteligencia artificial, em teoria, independente da linguagem computacional, mas que na prática a facilidade se torna maior dependendo da seleção deste recurso. Visando esta facilidade é que se procura uma ferramenta de trabalho a qual corresponde a uma representação de conhecimento e seu respectivo esquema. A análise deverá ser feita considerando-se vários quesitos. Infelizmente não existem parâmetros numéricos que permitam julgar objetivamente cada alternativa e critérios subjetivos, de foro pessoal, entram no julgamento, influenciando o resultado final.

O problema de treinamento de operadores, bem como o de apoio à decisão, requer, como apresentado no segundo capítulo, um ambiente do tipo produção e que tenha facilidades matemáticas e gráficas. Esta restrição elimina todos os pacotes apresentados por Barstow [35], pela ausência de recursos gráficos. Entre estes estão o EMYCIN (Empty MYCIN), KAS/PROSPECTOR, EXPERT, ROSIE, OPS5, HEARSEY III e AGE. É interessante salientar que estes pacotes são importantes e já produziram grandes sistemas peritos.

O EXSYS , apresentado por Rabuske [28] foi o único pacote comercial testado entre os muitos existente tais como M1, Personal Consultant e o INSIGHT 2+. O EXSYS é considerado uma casca no qual a entrada de regras e controlada pelo próprio pacote, com auxílio de menus e checagem de consistência de regras. Sua estratégia de procura do objetivo no espaço do problema é por caminhamento retrospectivo (backward chaining). No entanto é possível, opcionalmente , realizar um caminhamento progressivo (forward chaining). Uma característica na codificação do conhecimento empregando-se o EXSYS é o efeito colateral automático da obtenção do mecanismo de explicações.

A seleção do ambiente de trabalho resume-se assim a três linguagens (e seus dialetos), o Lisp, o Prolog e o SmallTalk V, a seguir descritas.

### 3.5.1 Lisp

A primeira linguagem para manipulação de símbolos e processamento de listas foi o IPL (Information Processing Language) desenvolvida por Newell e Simon em 1957. Os programas pioneiros em inteligência Artificial foram desenvolvidos nesta linguagem. Como exemplo cita-se o "Logic Theorist", o "General Problem Solver" (GPS) e SAD-SAM [5].

O IPL não possui certas facilidades como o gerenciamento automático e recuperação de memórias em listas não mais utilizadas. Para suprir estas necessidades, John McCarthy inventou o primeiro interpretador LISP, em 1958. Atualmente o LISP é a segunda linguagem mais antiga em utilização, após o FORTRAN [2]. Este fato, de per se, é suficiente para garantir sua utilização por muito tempo, pois as principais aplicações mundiais foram nele desenvolvidos. Muitos dialetos e mesmo linguagens foram geradas a partir do LISP como mostra a figura (3.3), de [2].

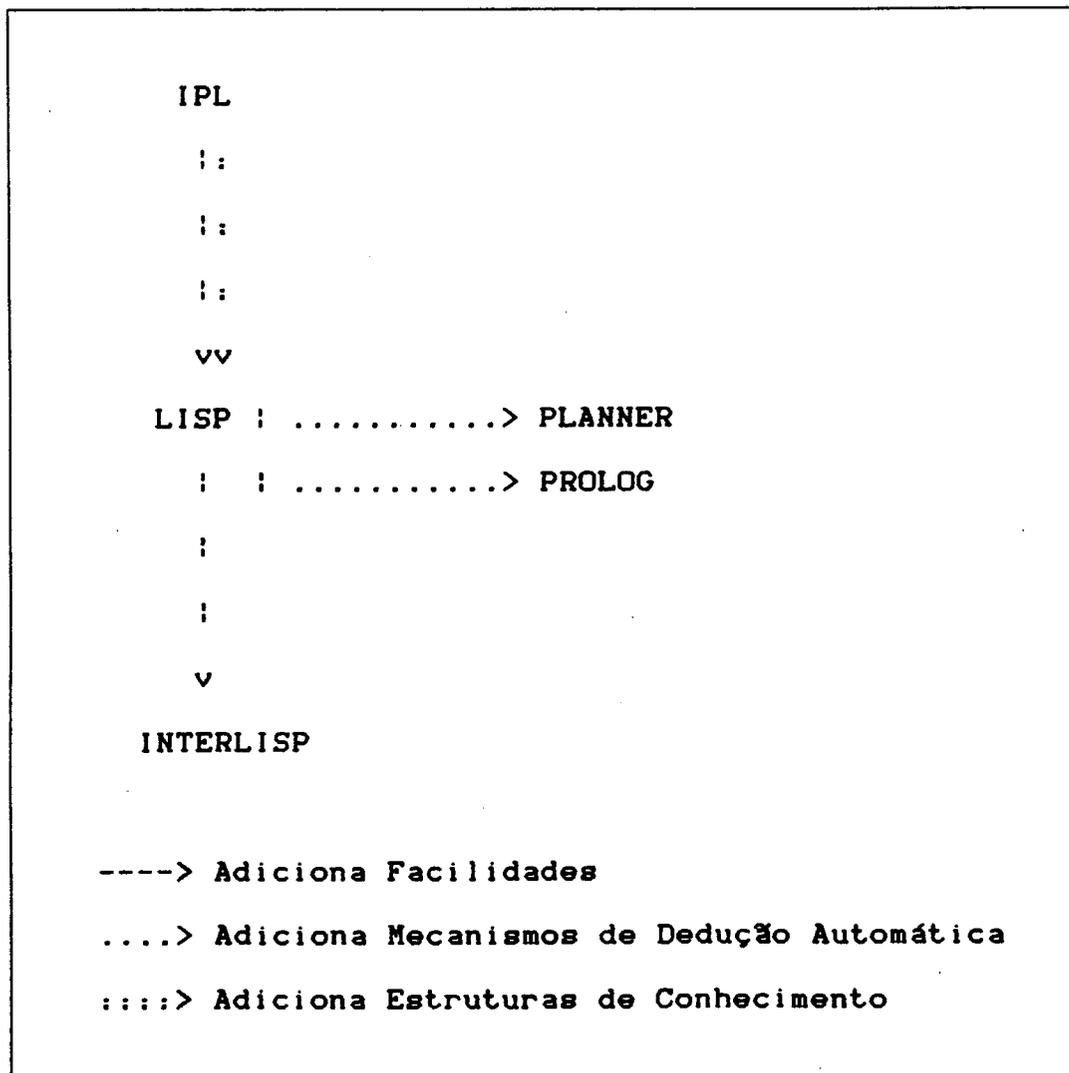


Figura 3.3 - Evolução das Linguagens de Inteligência Artificial

### 3.5.2 Prolog

O Prolog, devido a sua sintaxe não-procedural, não é simples de ser aprendido por aqueles que já programam em linguagens convencionais. Townsend [10] afirma ser

necessário uma fase de "desaprendizado" no ciclo de obtenção de proficiência nesta linguagem.

Apenas como ilustração à afirmativa contida em [10], cita-se o fato de ser o Prolog adequado a regras de produção sem no entanto possuir a estrutura característica de IF-THEN-ELSE. Este também é o motivo pelo qual muitos iniciantes em inteligência artificial escolhem o LISP como sua nova linguagem, pois esta possui estrutura de controle e fluxo de processamento semelhante às linguagens convencionais.

O Prolog possui mecanismos próprios que estabelecem uma estratégia de provas de teoremas, tendo por base o cálculo de predicados. Isto significa que ele procura demonstrar que um objetivo ou é verdadeiro ou é uma consequência lógica de fatos contidos na base de conhecimentos.

Em termos matemáticos [13], o Prolog aceita fatos e regras como um conjunto de axiomas e o objetivo como um "teorema" a ser provado. Assim ele procura provar a veracidade deste teorema ou demonstrar que ele pode ser derivado dos axiomas. A ordem sequencial da execução da busca começa com o objetivo. Aplicando-se regras substitui-se o objetivo corrente com novos objetivos até que se ache fatos primitivos (simples e verdadeiros).

O processo de obtenção de soluções em Prolog pode ser expresso na forma de uma busca em grafos. Aplica-se, para tal, o conceito de Espaço-Estado, no qual as situações (problemas) são representados por nós e ações possíveis por arcos. A metodologia de solução é a de se procurar o caminho ao objetivo, primeiramente sem preocupação com otimalidade.

A busca vertical é de fácil implementação no Prolog, visto ser o método intrínseco desta linguagem. É uma maneira muito rápida de se atingir uma solução, desde que se tome cuidados para que não ocorra "cycling". A busca horizontal é de implantação mais trabalhosa por necessitar manter uma lista de candidatos a expandir, o mesmo acontecido com o A\*.

No processo de busca, o mecanismo de inferência poderá encontrar ramos que fracassem. O procedimento automático de retorno a uma cláusula anterior e busca em outro ramo é denominado de "backtracking".

A unificação, no cálculo dos predicados, corresponde em Prolog ao processo de "matching". A diferença destes dois processos reside na eficiência computacional deste último.

### 3.5.3 SmallTalk V

A programação orientada a objetos envolve técnicas, notações e ferramentas que estudam como disciplinar o processo de abstração para a construção de sistemas com características estruturais e dinâmicas bem definidas [42]. Esta filosofia também é a mais indicada para iniciantes em inteligência artificial, pois evita inúmeros vícios de programação. Linguagens que permitam Programação Orientada a Objetos são eminentemente linguagens de processamento paralelo; embora sejam processadas sequencialmente.

O SmallTalk é uma linguagem declarativa, com ambiente de programação próprio. O SmallTalk V é uma versão do Smalltalk 80 (da Xerox) comercializado pela DigiTalk. Este produto trabalha com Quadros e Frames, programação orientada a objetos, linguagem Prolog e possui excelente interfaceamento gráfico com o usuário.

### 3.5.4 Aspectos Relevantes na Seleção da Ferramenta de Trabalho

Um aspecto relevante verificado na seleção da ferramenta de trabalho é apresentado no gráfico da figura 3.4 (Peled [36]), que mostra o crescimento da capacidade de

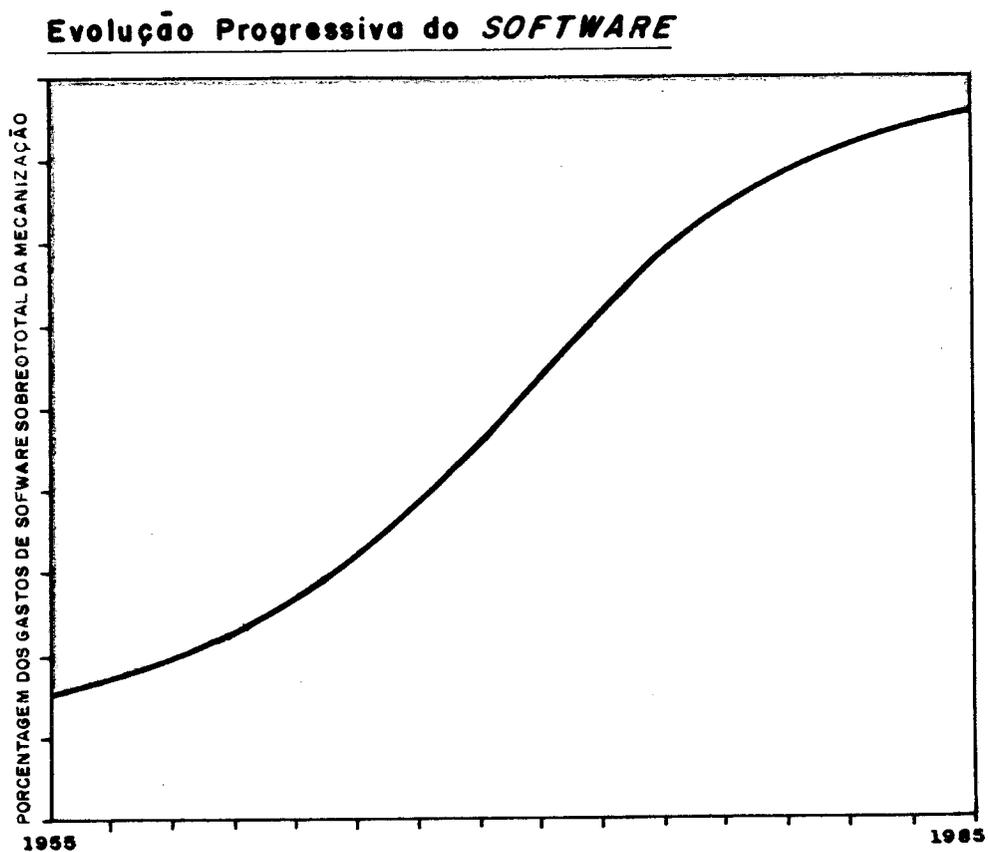
processamento dos computadores e do custo relativo do item software.

A programação convencional é um espelho da arquitetura existentes nos computadores atuais [23]. Um programa deverá conter, então, todos os passos de como executar uma tarefa. Peled [36] sugere a adoção de linguagens declarativas como forma de reduzir o custo relativo ao software no custo total de processamento de informações.

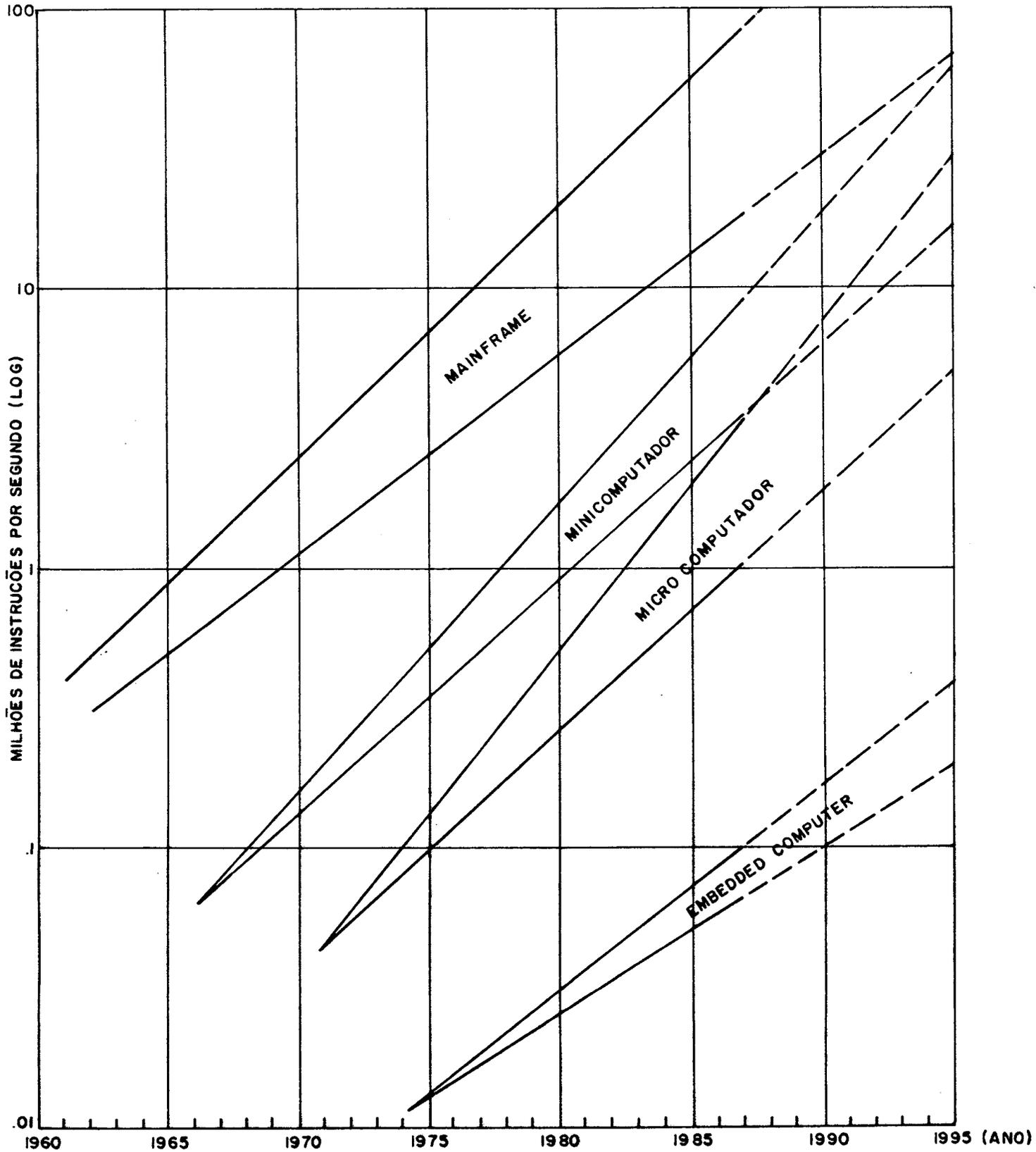
É justificado em [36] por que programas elaborados em linguagens procedurais deverão sofrer modificações para adaptar-se as novas arquiteturas (Paralelismo, etc.). As linguagens declarativas se adaptam a novas tecnologias de processamento e minimizam esforços nas fases de desenvolvimento (fluxogramas, árvores) e na documentação

Figura 3.4 - Evolução dos Custos de Software e da Capacidade de Processamento

a) Software



b) Hardware



Entre as ferramentas analisadas optou-se por linguagens básicas. O benefício esperado é o estabelecimento de experiências com os critérios e a organização de idéias com conhecimento de seus efeitos e implicações. Desta forma, não se corre risco de utilizar um produto sem o conhecimento de seus princípios básicos. Este seria o caso, se a seleção recaísse em linguagens de propósito especial, entre elas o OPS83, o Exysis e outras.

A linguagem selecionada, foi o PROLOG. Embora o Lisp também possa atender aos requisitos mencionados, há um maior potencial tecnológico no Prolog (paralelismo, etc.). Adicionalmente, dois fatores foram considerados a favor do Prolog [43]:

- A máquina de inferência já faz parte desta linguagem;
- A facilidade com que se constroi a base de conhecimento, quando comparado a outras linguagens.

O dialeto de Prolog adotado será descrito a seguir.

### 3.5.5 Dialeto Selecionado de Prolog

O Prolog foi concebido originalmente como um interpretador (assim como o Lisp) para permitir que o programa pudesse alterar o seu próprio código. O Prolog, considerado padrão "de facto", é o desenvolvido por Cloksin e Mellish para computadores DEC-10. Os dialetos desta linguagem, aplicáveis a microcomputadores, são na sua maioria compilados para obter ganho de eficiência. Nesta categoria apenas dois compiladores foram avaliados, por serem considerados tão rápidos quanto interpretadores Prolog em Mainframes: o Arity e o Turbo.

O Arity é a melhor versão, por ser um superconjunto da versão de Cloksin e Mellish. No entanto, por problemas de regulamentação de software o Turbo Prolog foi a ferramenta para construção do protótipo.

Não utilizou-se o Prolog-V, embutido no Smalltalk V, pelo fato deste produto não estar disponível para testes.

### 3.6 Conclusões

Este capítulo apresentou os conceitos básicos necessários na fase de implementação do sistema de treinamento de operadores. No próximo capítulo serão

apresentados a esquematização do problema, o modelo matemático e a estruturação de regras que fazem parte da etapa de implementação do modelo.

## CAPÍTULO IV

### IMPLEMENTAÇÃO DO MODELO

#### 4.1 Introdução

Neste capítulo apresenta-se o modelo desenvolvido, para testar-se a proposta de aplicação de técnicas de inteligência artificial no treinamento e apoio de decisão de operadores de hidrelétricas. Este modelo teve como protótipo a Usina Hidrelétrica de Salto Santiago, situada no Rio Iguaçu, a qual forma parte de um aproveitamento em cascata, como mostra a figura 4.1. A manipulação das regras de operação foi realizada com técnicas de processamento simbólico, enquanto a determinação de acumulação no reservatório, vazões turbinadas e descargas pelas comportas foi realizada com processamento numérico convencional.

A modularidade do sistema, visando seu aproveitamento em barramentos diferenciados, foi obtido adotando-se o conceito de casca.

#### 4.2. Estrutura e Modularidade do Sistema Computacional

Como forma de permitir a re-utilização deste sistema, adotou-se, neste trabalho, a proposta de criar-se uma casca. A casca (expert system shell) é composta basicamente

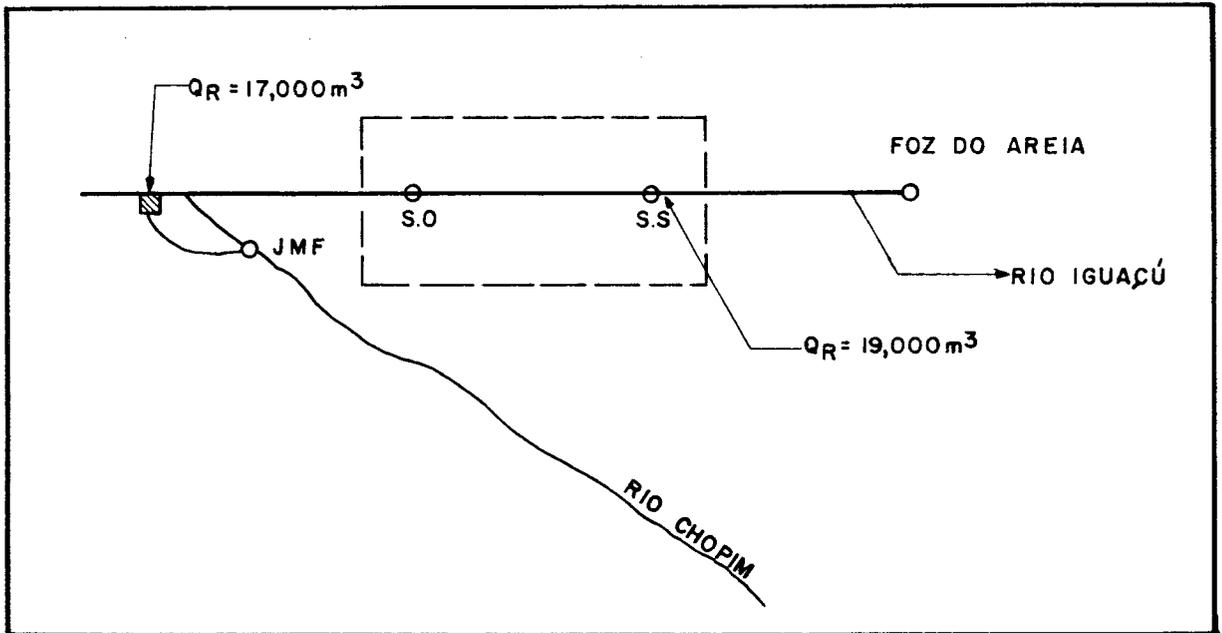


Figura 4.1 Esquema de Utilização do Rio Iguaçu

de uma associação da máquina de inferência e da interface com o usuário. A figura 4.2, extraída de [13], serve para visualizar a idéia.

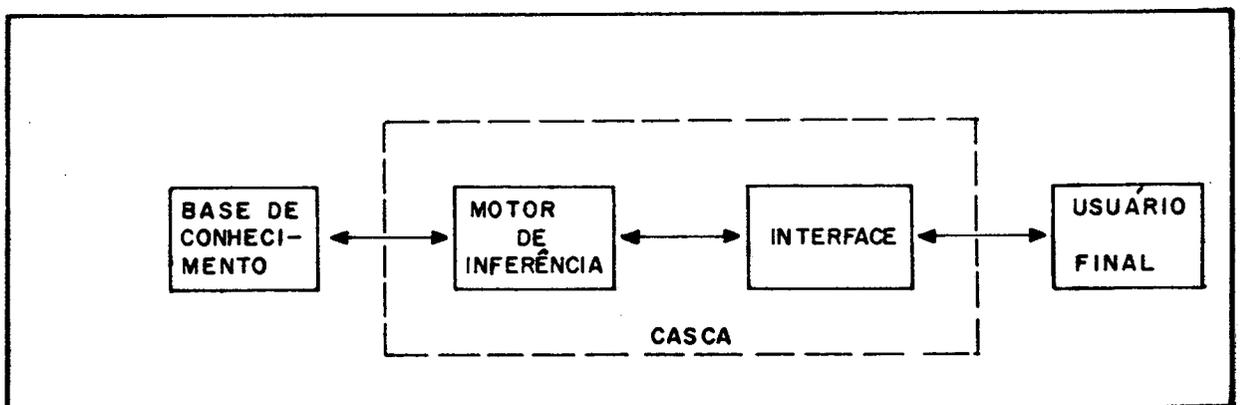


Figura 4.2 Estrutura de Uma Casca

O esquema apresentado na figura 4.2 tem como vantagem a separação completa dos procedimentos de processamento com a base de conhecimentos. Desta maneira, torna-se necessário a modificação apenas deste último elemento em cada nova aplicação. Os demais elementos que compõe a casca não são afetados, ou pouco são, pela mudança do ambiente. Este é o motivo da adoção desta proposta no protótipo deste sistema perito. As modificações necessárias na base de conhecimento, são a inclusão de novas regras de operação da cascata, de operação de comportas e detalhes construtivos. Buscou-se com isto o benefício de um produto mais apropriado à tarefa de apoio à decisão e treinamento de operadores, associado a um esquema mais econômico em recursos humanos e financeiros por toda a vida útil deste software.

A figura 4.3 mostra como todos os conceitos foram agrupados para obter uma estrutura esquemática do conhecimento (EEC). Esta estrutura previne a acumulação de erros em iterações sucessivas. O laço entre "Operação Manobra de Comporta" e o "Status Operativo" não é uma iteração de convergência, mas de incremento no tempo. Neste incremento temporal, novas medidas são tomadas para corrigir controle inapropriado ou modificações nas variáveis de controle ocorridas no intervalo.

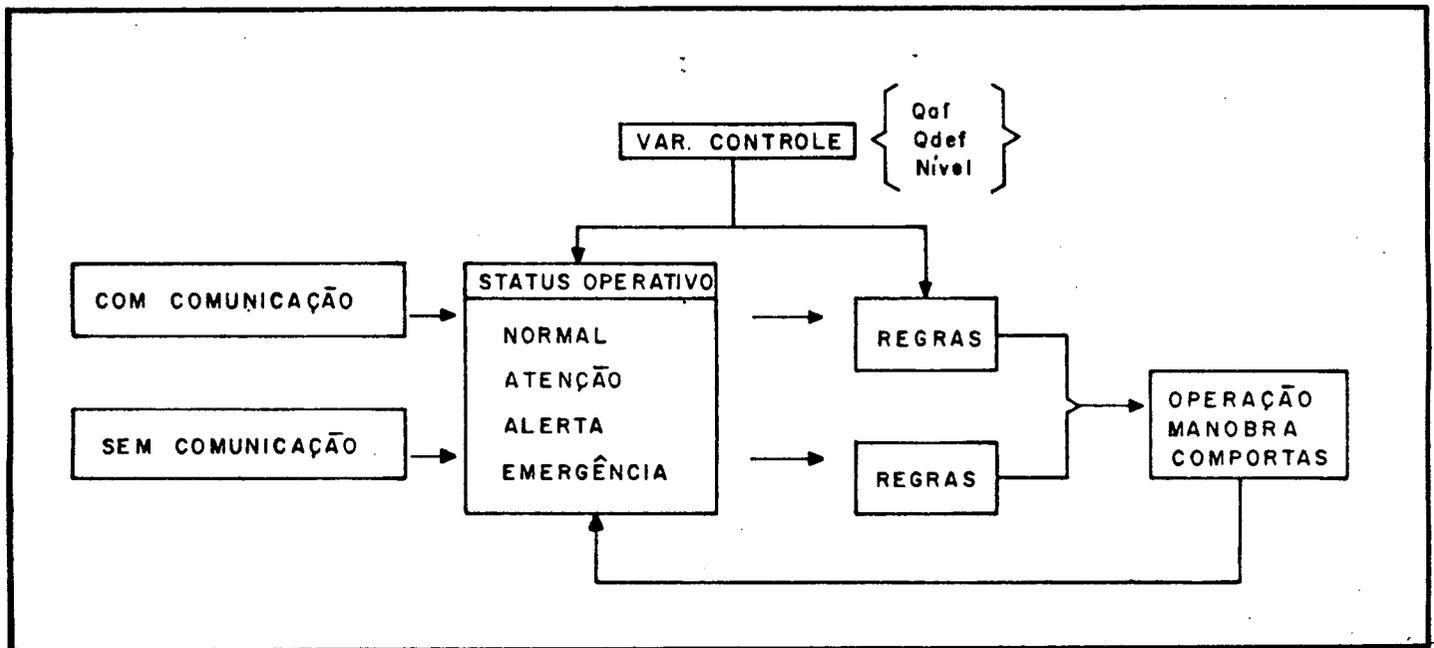


Figura 4.3 Estrutura Esquemática do Conhecimento

#### 4.3 Modelo Matemático

A representação do comportamento do reservatório pode ser feita de duas maneiras: através de conhecimento ou através de modelo matemático. São cinco as relações principais necessárias a um simulador hidráulico, quais sejam:

- Equação da continuidade;
- Relação Cota x Descarga do Vertedouro;

- Relação Cota x Volume do Reservatório;
- Relação Cota x Descarga do Canal de Fuga;
- Vazão Turbinada.

A representação através de conhecimento, como será mostrado, não é adequada para o primeiro e último itens relacionados acima. Por esta razão adotou-se a representação através de um modelo matemático, selecionando-se sempre a alternativa que facilite sua generalização a outros reservatórios.

#### 4.3.1 - Equação da Continuidade

A equação da continuidade, que possibilita o balanço de massa no reservatório, pode ser expressa como

$$I - O = \frac{dS}{dT} \quad (4.1)$$

Onde, para o problema

- I = afluência regularizada
- O = defluência total
- dS = variação de armazenamento

- dT = variação de tempo

#### 4.3.2 - Relação Cota x Descarga do Vertedor

O vertedouro pode operar com abertura parcial de suas comportas ou em lâmina livre, quando estas se encontram completamente abertas.

##### 4.3.2.1 Cálculo da Vazão em Lâmina Livre

O cálculo da vazão em lâmina livre pode ser feito por interpolação de cinquenta valores dados ou através da solução da equação abaixo

$$Q_v = C_q L H^{(3/2)} \quad [m^3/s] \quad (4.2)$$

onde:

- $C_q$  = coeficiente de descarga (obtido por interpolação / extrapolação em Tabela N.A. x  $C_q$ , onde N.A é o nível de água do reservatório);
- $L$  = largura do vão do vertedor;
- $H$  = altura de água acima da crista do vertedor.

Usando-se a equação (4.2) necessita-se apenas duas constantes,  $C_q$  e  $L$ .

#### 4.3.2.2 Cálculo de Vazão com Abertura Parcial de Comportas

O cálculo da vazão volumétrica em operação do vertedouro com abertura parcial é mais complexa que o da vazão em lâmina livre. As comportas podem estar com diferentes aberturas assim como cada abertura pode produzir diferentes afastamentos do perfil do vertedouro. Usualmente emprega-se uma das duas alternativas abaixo (37) :

$$Q_v = (\text{Valor da Tabela Abertura}) \times N.A. \times Q_v \quad (4.3)$$

ou

$$Q_v = C_q L G_0 (2gH)^{(1/2)} \text{ [m}^3\text{/s]} \quad (4.4)$$

onde:

- N.A = Nivel d'água no reservatório;
- $C_q$  = coeficiente de descarga, obtido por interpolação em tabela  $G_0 \times N.A. \times C_q$  (100 triplas);
- L = largura do vão do vertedouro;
- $G_0$  = mínima distância entre a comporta e o perfil do vertedouro;
- H = altura de água acima do centro da abertura efetiva da comporta (na secção  $G_0$ );

A valor da tabela necessária para utilização da equação (4.3) exige duzentas triplas de dados. Assim para que se generalize esta abordagem faz-se necessário a

substituição destes valores para cada aplicação distinta. O coeficiente de descaraga (Cq) e a mínima distância entre a comporta e o perfil do vertedouro (GO) são função da abertura da comporta e do nível do reservatório. Estes coeficientes podem ser aproximados, dentro da precisão requerida por um polinômio de segundo grau.

Para este protótipo estes coeficientes podem ser expressos por

$$GO = Ag0 + Ag1 \cdot A_{comp} + Ag2 \cdot A_{comp}^2 \quad (4.5)$$

Onde

$$Ag0 = 0.579;$$

$$Ag1 = 0.868;$$

$$Ag2 = 7.85E-03;$$

Acomp = Abertura da Comporta.

$$Cq = A0 + A1 \cdot Z + A2 \cdot Z^2 \quad (4.6)$$

Onde

$$A0 = 158,487;$$

$$A1 = -6.38E-01;$$

$$A2 = 6.4486E-04$$

Z = Cota do Bordo da Comporta, como mostrado na Figura 4.4.

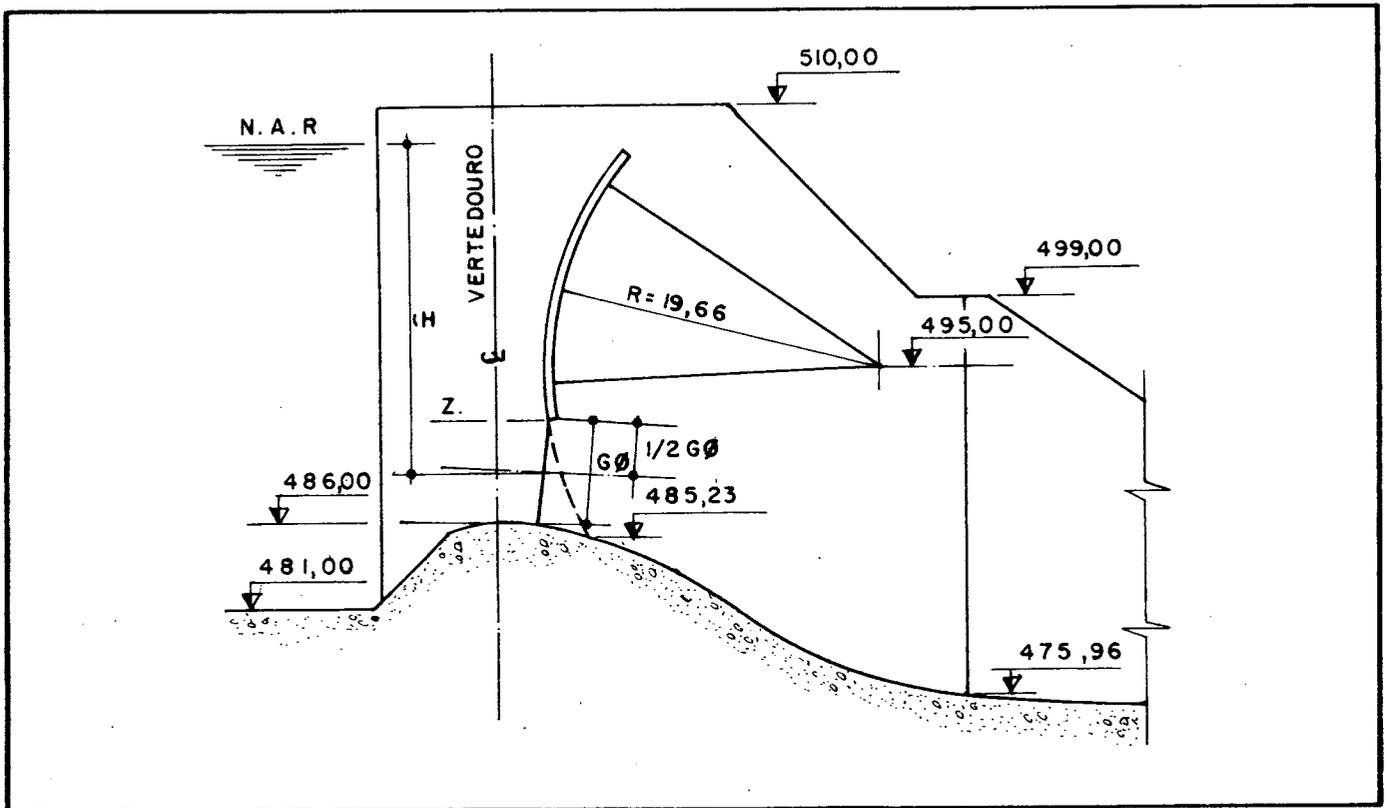


Figura 4.4 Corte de uma Comporta de Vertedouro

#### 4.3.3 Relação Cota x Volume

A relação Cota x Volume visa determinar o volume de água armazenado no reservatório para uma determinada cota. As formas usuais de obter-se estes valores são !37! :

- Tabela NA. x V (50 pares)
- Equações
- Polinomial (até grau 5)
- Potência ( $v = a^{(c-c_0)}$ )

A alternativa adotada pelos peritos da ELETROSUL faz uso de um polinômio de quarto grau. O polinômio adotado no protótipo pode ser expresso por

$$Vol = A_0 + A_1 \cdot N + A_2 \cdot N^2 + A_3 \cdot N^3 + A_4 \cdot N^4 \text{ (Hm}^3\text{)} \quad (4.7)$$

Onde

$$A_0 = 9.34636093495846E11;$$

$$A_1 = -7.51601954817769E09;$$

$$A_2 = 2.34784794096565E07;$$

$$A_3 = -3.41966810869656E04;$$

$$A_4 = 1.97428638372551E01;$$

#### 4.3.4 Relação Volume x Cota

A relação Volume x Cota é utilizada na determinação da variação do nível do reservatório. Para este fim, emprega-se a equação (4.1), a qual requer a vazão de afluência e defluência. A cota a ser prevista poderá ser

determinada por cálculos iterativos ou por uma equação polinomial com cinco termos. Esta última alternativa é de mais fácil manipulação com cálculos mais rápidos e precisos. Por estes motivos, esta foi a alternativa selecionada. A equação desenvolvida para o protótipo pode ser expressa por:

$$\text{NovaCota} = A0 + A1 \cdot X + A2 \cdot X^2 + A3 \cdot X^3 + A4 \cdot X^4 \quad (4.8)$$

Onde

$$A0 = 4.52443882205206E+02;$$

$$A1 = 1.41267315058721E-08;$$

$$A2 = -1.59306303675594E-18;$$

$$A3 = 1.31248375919341E-28;$$

$$A4 = -4.67247087654466E-39;$$

e

$$X = \text{Novo Volume} = (\text{Volume Atual}) + (\text{Variação de Volume})$$

#### 4.3.5 Relação Cota x Descarga do Canal de Fuga

O nível do canal de fuga é um fator importante por impor riscos a casa de força da usina. É usual o emprego, pelos peritos do domínio, de dois procedimentos para sua

estimativa, quais sejam :

- Tabela com 80 pares de pontos;
- Equação interpolada.

Geralmente se dispõe de várias curvas cota x descarga do canal de fuga, as quais são função do nível a jusante. Por exemplo, para Salto Santiago dispõe-se de duas: uma para vazões altas e outra para vazões baixas.

Foi adotado neste protótipo uma equação interpolada destas duas curvas. Desta forma, a equação interpolada minimiza erros devido a variação de nível a jusante, e pode ser definida por

$$Ccf = A0 + A1 \times Vd + A2 \times Vd^2 \quad (m) \quad (4.9)$$

Onde

Ccf = Cota canal de fuga;

A0 = 3.96457E+02;

A1 = 1.37877E-03;

A2 = -1.04861E-08;

#### 4.3.6 Vazão Turbinada

A vazão turbinada pode ser obtida pela aplicação da expressão abaixo (27) :

$$QT = K \cdot PT / (HL \times NT) \quad (\text{m}^3/\text{s}) \quad (4.10)$$

Onde

- HL = queda líquida (queda bruta - perda no circuito hidráulico, que é função da vazão turbinada);
- PT = potência (potência elétrica lida/rendimento da parte elétrica);
- QT = vazão turbinada total;
- NT = rendimento da turbina (obtida da interpolação em tabela HL x NT);
- K = constante.

Este cálculo é realizado iterativamente, visto ser a queda líquida dependente de perdas no circuito hidráulico. Estas por sua vez dependem do volume turbinado. Desta forma calcula-se a queda líquida por :

$$HL = H_b - K_1 \times QT^2 \quad (4.11)$$

onde :

$H_b$  = Queda bruta (m)

$K_1$  = Constante relativa ao circuito hidráulico

No protótipo os valores das constantes foram definidos na fase de projeto e são :

$K = 10197,0;$

$K_1 = 1.5043E-05;$

O rendimento da turbina depende da queda líquida existente e da potencia gerada. Sua determinação pode ser obtida com o emprego de curvas de colina, apresentadas na figura 4.5 abaixo:

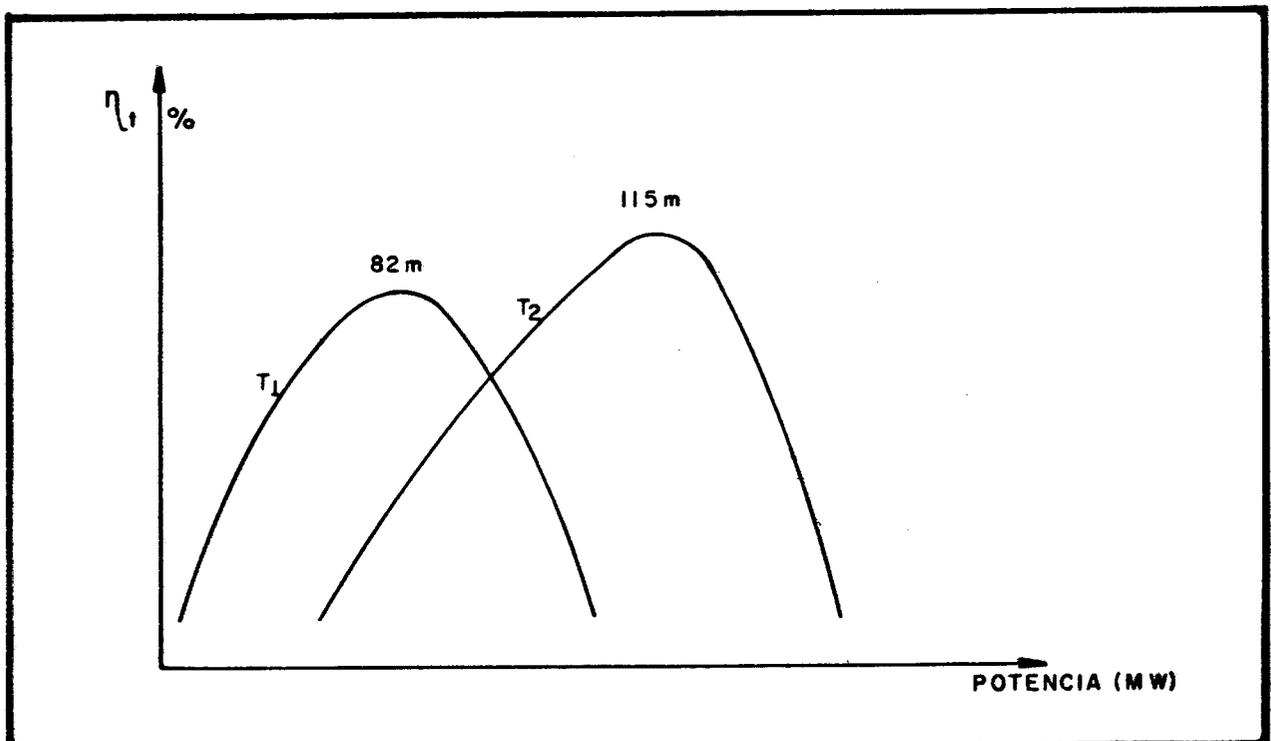


Figura 4.5 Curvas de Colina de Uma Unidade em UHSS

As curvas T1 e T2 são a mínima e máxima quedas admissíveis na operação da turbina. Calcula-se os rendimentos em T1 e T2 e interpola-se com a queda líquida através da expressão :

$$N_t = (HL - H_{min}) * (T_2 - T_1) / (Q_{max} - Q_{min}) + T_1 \quad (4.12)$$

As curvas T1 e T2 são aproximadas numericamente, pelos peritos do domínio, por um conjunto de retas. No entanto, este trabalho adota duas parábolas para permitir uma generalização. Uma avaliação no procedimento para cálculo de vazões turbinadas existente, apontou simplificações possíveis de equações e de critérios de convergência. As modificações realizadas resultaram em um acréscimo na velocidade de processamento de 72%, mantendo-se a mesma precisão.

#### 4.3.7 Defluência Indicada

Estudos com modelos hidráulicos e com modelos matemáticos permitem a previsão de uma defluência mínima que previna o rompimento de um nível de restrição. A operação utiliza estas previsões organizadas em um "Diagram de Operação Normal e em Emergência". Neste protótipo adota-se o diagrama da figura 4.6. As curvas de vazão

afluente podem ser expressas por uma única equação. Assumindo-se, para tal, que representem projeções em uma superfície em tres dimensões, isto é, que sejam curvas de nível. Esta expressão foi obtida por métodos apresentados por Stoecker [40] e é apresentada abaixo:

$$\begin{aligned}
 V_d = & (A_{00} + A_{01} \times Q + A_{02} \times Q^2) + \\
 & (A_{10} + A_{11} \times Q + A_{12} \times Q^2) \times \text{Cota} + \\
 & (A_{20} + A_{21} \times Q + A_{22} \times Q^2) \times \text{Cota}^2 \qquad (4.13)
 \end{aligned}$$

Onde para o protótipo

$$A_{00} = 1.681844209E+08;$$

$$A_{01} = -1.283064083E+04;$$

$$A_{02} = 2.650151410E-01;$$

$$A_{10} = -6.679861687E+05;$$

$$A_{11} = 5.069882363E+01;$$

$$A_{12} = -1.045060218E-03;$$

$$A_{20} = 6.632454971E+02;$$

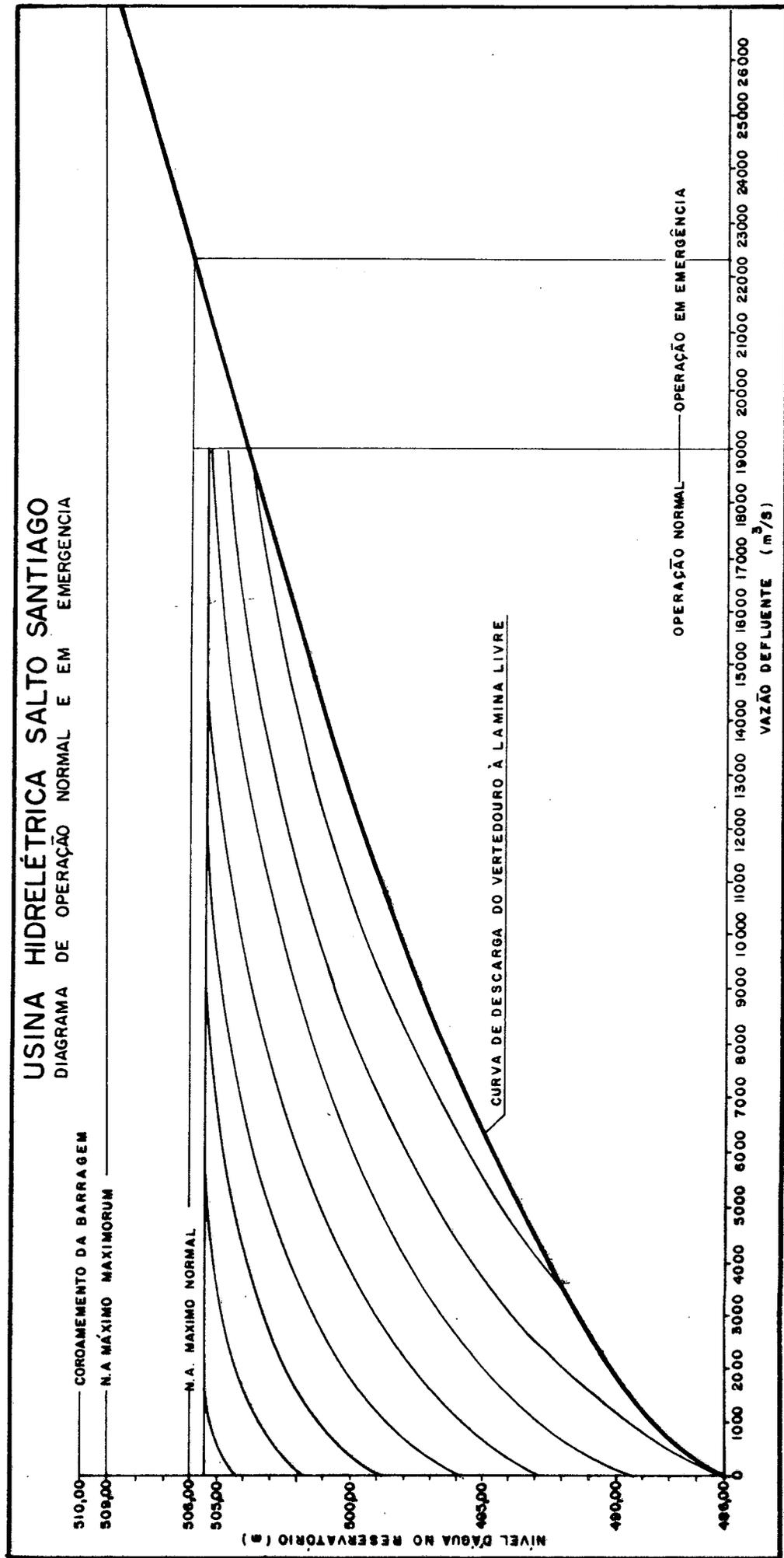
$$A_{21} = -5.007676102E-02;$$

$$A_{22} = 1.030161953E-06;$$

#### 4.3.8 Modelo de Abertura das Comportas

A abertura apropriada das comportas, pode ser obtida dado um nível d'água e uma vazão de defluência requerida. Emprega-se então cálculos de vazões vertidas (seções 4.3.2.1 e 4.3.2.2), conhecimento do procedimento de abertura das comportas (seção 2.6) e um método de procura.

Figura 4.6 - Diagrama de Operação Normal e em Emergência



Inicialmente, utilizou-se o Método de Busca Vertical, embutido no Prolog, associando-se ao processo cláusulas recursivas. Este esquema funciona com eficácia e rapidez. No entanto, a pequena memória reservada para a Pilha (Stack de 64kbytes) é rapidamente ocupada causando erros de overflow de Stack. Este problema ocorre sempre que o estado inicial estiver afastado mais que 12 passos do estado final.

A estruturação da busca em formato repetição-falha requereu a adoção de uma base de dados, atualizada a cada falha, requerendo, assim, muito tempo de processamento.

Adicionou-se mais conhecimento à busca, como forma de solucionar este problema, de maneira que o estado inicial sempre está a uma distância máxima menor que 10 passos do objetivo. Reduziu-se também o espaço de busca pela introdução de um ordenamento da direção de busca.

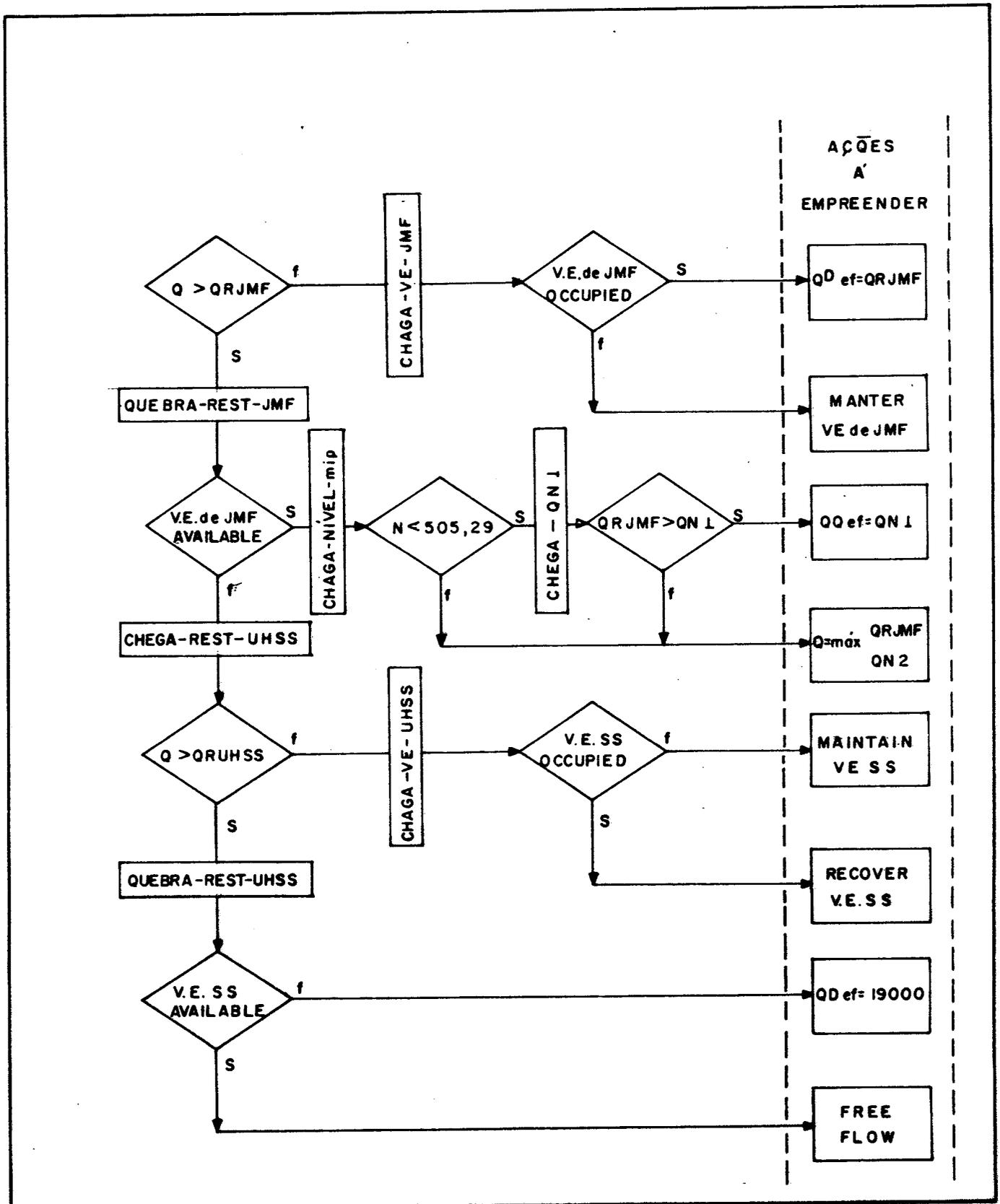
Cabe salientar que a solução do problema de abertura apropriada de comportas atenderá aos peritos do domínio e operadores de reservatório. Estes estão, hoje, confinados a volumosas listagens, com tabelas indicativas da configuração adequada das comportas.

#### 4.4 Operação do Reservatório com Perda de Comunicações

Como já visto na seção 2.6, existe risco de perda de comunicação com o sistema. O sub-modelo implementado para o controle da vazão, neste evento, está apresentado no apêndice B. Adotou-se o esquema da figura 2.2 para dividir o problema em pequenos subproblemas, como mostra a figura 4.7. Os quadros com nomes na figura 4.7 são esquemas de início de predicados, tal como implementado no protótipo em Prolog (Thé 145!).

O controle do volume a ser defluído utiliza o "Quadro de Referência para Operação", apresentado na figura 2.3 e o "Diagrama de Operação e Controle" (figura 2.4) para definir o Estado Hidrológico na Escala de Valores de Referência. As regras implementadas no protótipo definem o Estado e a ação recomendada, apresentando opcionalmente as medidas a serem tomadas contidas no manual de operação.

Figura 4.7 - Divisão do Problema de Operação com Perda de Comunicações em Subproblemas



#### 4.5 Interface com o Usuário

As interfaces implementadas procuram evitar que o usuário não dispenda seu tempo com o sistema de treinamento, mas sim com o treinamento da operação do reservatório. Os principais elementos da interface são:

- Interface Gráfica;
- Telas de Introdução de Novas Usinas;
- Menus;
- Telas de Ajuda;
- Linguagem Natural Restrita;

A interface gráfica do sistema de treinamento é constituída pelos seguintes esquemas:

- Esquemas de Abertura de Comportas;
- Visualização do Barramento em Corte com níveis de restrição assinalados;
- Gráficos com Curvas Ajustadas pelo Método de Mínimos Quadrados;
- Gráficos com Afluências e Defluências contra o Período do Evento em que Ocorreram.

A implementação da interface gráfica foi facilitada pela adoção de uma caixa de ferramentas [39] e [41].

#### 4.5.1 Telas de Introdução de Novas Usinas

O sistema possui um módulo para facilitar a introdução de dados necessários ao treinamento de novos aproveitamentos. Este módulo solicita informações por telas contendo campos pré-programados. Após a crítica de dados inicia-se o ajuste de curvas, para obtenção dos coeficientes requeridos pelo sistema de treinamento. A determinação dos coeficientes é feita utilizando-se o Método dos Mínimos Quadrados. Existem quatro opções para este ajuste, a exponencial, a logarítmica, a de potência e a polinomial. Esta última opção é a única de significado para o sistema de treinamento, servindo o conjunto inteiro como ferramenta na área dos peritos do domínio. Como facilidades adicionais implantou-se também um histograma com os desvios padrões acumulados de cada ajuste, impressão de gráficos em dois tamanhos e facilidades para o direcionamento dos resultados para impressora, tela ou um arquivo.

#### 4.5.2 Menus de Seleção

O menu é a principal ferramenta utilizada para capturar o direcionamento do fluxo de processamento. Estes menus foram envolvidos por cláusulas de fracasso e

repetição para permitir sua presença sempre que um comando fracassar ou no retorno de sua execução.

Neste protótipo existem seleções, por menu para diversas alternativas, tais como :

- Carregar dados de uma Usina;
- Treinar em Evento de Cheia;
- Listar Resultados;
- Plotar Resultados;
- Editar Arquivos;
- Shell para o Sistema Operacional

O treinamento propriamente dito é o de operação de um evento de cheia. No caso de não haver proficiência em algum aspecto desta operação, o usuário é orientado a desenvolver suas habilidades em um dos seguintes modos de treinamento, também selecionado por menu:

- Controle de Vazão;
- Abertura de Comportas;
- Calculos de Parâmetros de Operação.

Finalmente, neste protótipo é possível que o usuário obtenha, por intermédio de menus ajuda, recomendações e gráficos.

### 4.5.3 Telas de Ajuda

O sistema emprega vinte janelas que têm o propósito de tornar a operação de treinamento mais eficaz pela introdução de avisos, sempre que erros venham a ocorrer, ou caso informações sejam solicitadas. Como ilustração, a tabela 4.1 relaciona as janelas presentes no protótipo desenvolvido:

Tabela 4.1 Janelas de Ajuda

No. da Janela	Finalidade
1	Principal no treinamento de cheia
2	Regras de controle de vazão
3	Abertura de comportas
4	Cálculo de parâmetros de operação
5	Status operativo da usina
6	Entrada de dados da abertura de comportas
8	A definir
9	A definir
10	Seleção de reservatório de usina
11	Várias janelas alertam em tom informal sobre erros causados por desatenção
12	Entrada de dados sobre defluência

13	Escolha de arquivo para edição
14	Recomendações de procedimentos p/ operação
15	Janelas com instruções sobre os estados hidrológicos possíveis
20	Tela principal de todo o sistema
30	Reservada para apresentação de menus

#### 4.5.4 Linguagem Natural Restrita

A linguagem natural restrita foi implementada no protótipo como apresentada no capítulo 3. Não obstante o esforço empreendido sempre ocorreram falhas que comprometiam a confiança no sistema como um todo. Por esta razão esta facilidade encontra-se de forma isolada (analisador sintático) e sem atuação no sistema. Espera-se que uma versão futura venha a adotá-la de uma forma melhorada.

#### 4.6 Regras Gerais do Sistema

Adicionalmente às regras da seção 4.3, contidas no Apêndice "Operação e Controle do Reservatório", diversas regras tiveram de ser implementadas no protótipo. Estas regras servem para que o sistema possua conhecimentos que evitem ocorrências infactíveis. Pode-se citar, como

exemplo, regras que não permitem níveis de jusante maiores que os de montante, ou valores fora da faixa operacional.

#### 4.7 Arquivos Auxiliares do Sistema

Devido a limitação de espaço de armazenamento real, implementou-se uma estrutura de overlay e de arquivos de dados em discos. Esta estrutura apresenta a vantagem de não necessitar todo o programa residindo em memória simultaneamente. O processamento tem sua velocidade reduzida sempre que se recorre a dados em discos, mas este problema não chega a ser um incômodo ao usuário. Os arquivos utilizados pelo sistema estão listados na tabela 4.2, abaixo.

Tabela 4.2 Arquivos Auxiliares do Sistema de Treinamento

Arquivo	Finalidade
NomeUsina.DNU	Contem todos os fatos necessários ao processamento do treinamento;
Onda.DAD	Contem a onda de cheia em forma de lista.
Onda.XY	Contem o resultado do treinamento simulado;

AJC.Com	Ajustador de Curvas pelo método dos mínimos quadrados, com gráficos e histogramas;
IndUsina.EXE	Introduz dados de uma nova usina no sistema de treinamento;
Leonda.EXE	Transforma pares X, Y em listas compostas e vice-versa;
Listar.Com	Lista arquivos contidos em disco, possuindo diversos formatos de impressão;
Resegraf.Exe	Plota um corte transversal do reservatório que se encontra em simulação de operação;
Resul.Com	Apresenta as curvas das variáveis de controle do início do evento até o momento solicitado;
NomeUsina.Hal	Tela de ajuda no estado de alerta;
NomeUsina.Hat	Tela de ajuda no estado de atenção;
NomeUsina.Hem	Tela de ajuda no estado de emergência;
NomeUsina.Hno	Tela de ajuda no estado normal.

#### 4.8 Conclusão

Neste capítulo foram apresentados os esquemas de regras e estratégias adotadas para a implementação do modelo. No próximo capítulo serão vistos os casos testados e seus resultados serão comentados.

## CAPÍTULO V

### RESULTADOS E DISCUSSÃO

#### 5.1 Introdução

Neste Capítulo apresentam-se os resultados de testes do protótipo e discute-se suas vantagens e limitações. Analisa-se, também, a ferramenta computacional adotada e os recursos disponíveis para a estação de treinamento de operadores de reservatórios de hidrelétrica.

#### 5.2 Resultados dos Testes Com o Modelo Matemático

Todos os subsistemas foram testados exaustivamente antes de sua inclusão no sistema de treinamento. Apresenta-se os resultados de suas partes isoladas e de sua atuação em conjunto.

##### 5.2.1 Resultados dos Testes Com o Modelo de Defluência Indicada

Os testes no Modelo de Defluência Indicada foram realizados para avaliar a precisão da equação contida na secção 4.3.7. Os primeiros resultados indicaram erros, fora dos limites toleráveis, em certas faixas. Este problema foi

contornado introduzindo-se um maior numero de informação ao ajustador de curvas. O modelo presente não apresenta desvios maiores que 50 m<sup>3</sup>/s dos valores obtidos no gráfico da figura 4.6, o que representa erros máximos de 5%.

### 5.2.2 Resultados dos Testes Com o Modelo de Vazão Turbinada

O modelo para predição dos valores da vazão turbinada, dado uma determinada potência gerada e cota, foi igualmente testado. A precisão exigida foi facilmente obtida e a preocupação recaiu sobre a estabilidade do modelo em toda a faixa operativa. Devido a padronização das turbinas em uma usina hidrelétrica, é suficiente a representação de uma delas.

A tabela 5.1 apresenta os valores da vazão turbinada para as cotas e potência na faixa operativa. Observe que para cotas mais altas a vazão necessária, para uma mesma geração, é menor. Este fato pode ser visualizado na figura 5.1.

Tabela 5.1 Valores da Vazão Turbinada

Cota = 490(m)		Cota = 500(m)	
(Nve)	(m <sup>3</sup> /s)	(Nve)	(m <sup>3</sup> /s)
0	0.00000	0	0.00000
30	109.73400	30	72.84620
60	139.53500	60	111.00000
90	160.84300	90	138.66900
120	181.37400	120	162.82600
150	203.82200	150	186.54100
180	230.11700	180	211.74100
210	262.51700	210	240.14200
240	304.49200	240	273.80600
270	362.47200	270	315.82800
300	451.47200	300	371.86300
330	630.68700	330	455.25000

Cota = 495		Cota = 505	
(Nve)	(m <sup>3</sup> /s)	(Nve)	(m <sup>3</sup> /s)
0	0.00000	0	0.00000
30	88.21710	30	61.37800
60	124.02300	60	99.94990
90	149.08900	90	129.37400
120	171.59800	120	154.90900
150	194.69600	150	179.21400
180	220.39400	180	203.99800
210	250.68700	210	230.68700
240	288.27500	240	260.80700
270	337.68400	270	296.39100
300	408.25200	300	340.70300
330	527.32600	330	400.15200

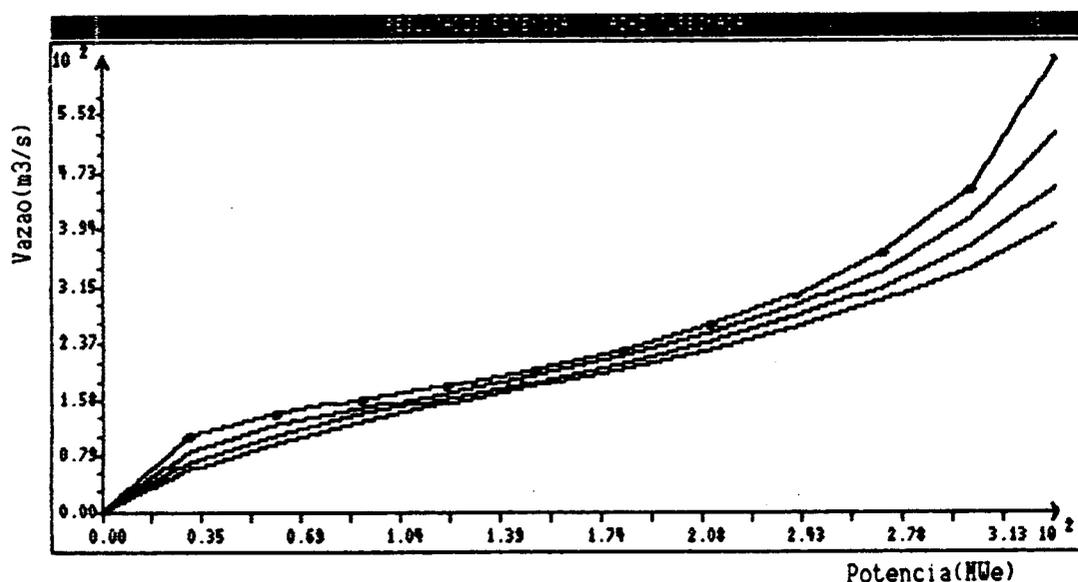


Figura 5.1 Curvas Potência X Vazão Turbinada p/ Várias Cotas

### 5.2.3 Resultados dos Testes Com a Determinação da Cota e do Volume

A representação deste processo não requer a máxima precisão no cálculo da cota e do volume do reservatório. Este fato se deve a leitura da cota a cada iteração. No entanto, para obter a confiança do usuário do sistema, buscou-se ajustes que retendo a simplicidade produzisse resultados precisos.

Testou-se, em conjunto, a precisão dos dois esquemas, o que preve o volume dado a cota e o que, dado o volume, fornece a cota. A tabela 5.2 apresenta os resultados obtidos nestes testes. Os volumes medidos por levantamentos planialtimétricos não possui as casas decimais contidas na tabela.

Tabela 5.2 - Resultados de Teste da Cota e do Volume

Cota Dada (m)	Volume Obtido (m <sup>3</sup> )	Cota Obtida (m)
485.00	3.1819548013E+09	485.01
486.00	3.3192972614E+09	486.01
487.00	3.4598374234E+09	487.01
488.00	3.6036006249E+09	488.02
489.00	3.7506126772E+09	489.02
490.00	3.9008998656E+09	490.02
491.00	4.0544889492E+09	491.01
492.00	4.2114071610E+09	492.01
493.00	4.3716822076E+09	493.01
494.00	4.5353422695E+09	494.01
495.00	4.7024160013E+09	495.01
496.00	4.8729325311E+09	496.01
497.00	5.0469214610E+09	497.00
498.00	5.2244128668E+09	498.00
499.00	5.4054372982E+09	499.00
500.00	5.5900257789E+09	500.00
501.00	5.7782098061E+09	500.99
502.00	5.9700213510E+09	501.99
503.00	6.1654928588E+09	502.99
504.00	6.3646572481E+09	503.99
505.00	6.5675479118E+09	505.00
506.00	6.7741987162E+09	506.00
507.00	6.9846440019E+09	507.00
508.00	7.1989185828E+09	508.00
509.00	7.4170577470E+09	509.00
510.00	7.6390972564E+09	509.99

A variação de 0.01(m) na cota resultante foi considerada excelente pelos peritos do domínio. As figuras 5.2 e 5.3 apresentam as curvas geradas pelo ajustador de curvas do sistema de treinamento.

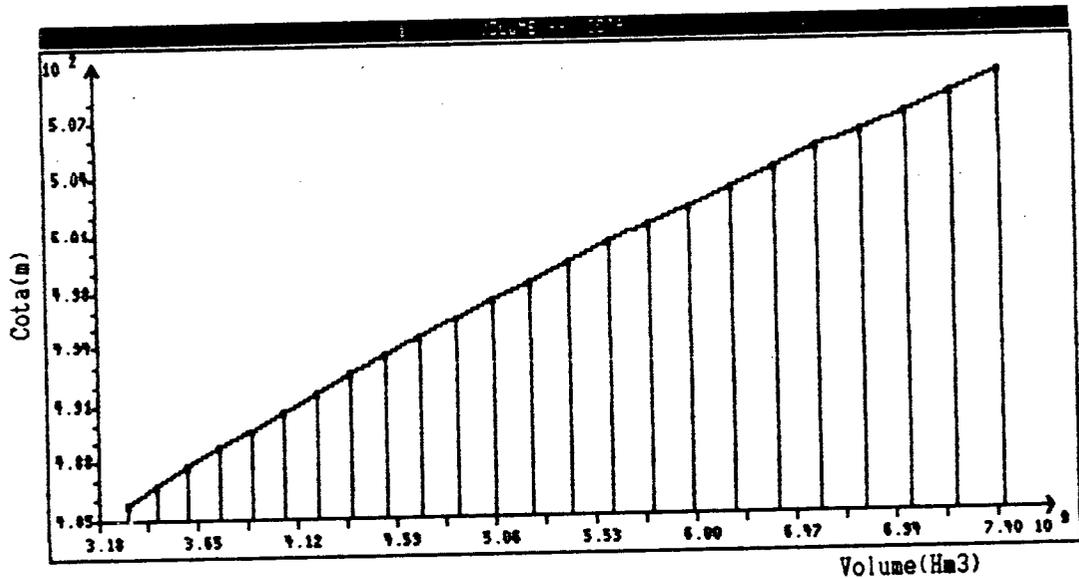


Figura 5.2 Curva da Cota Dado o Volume

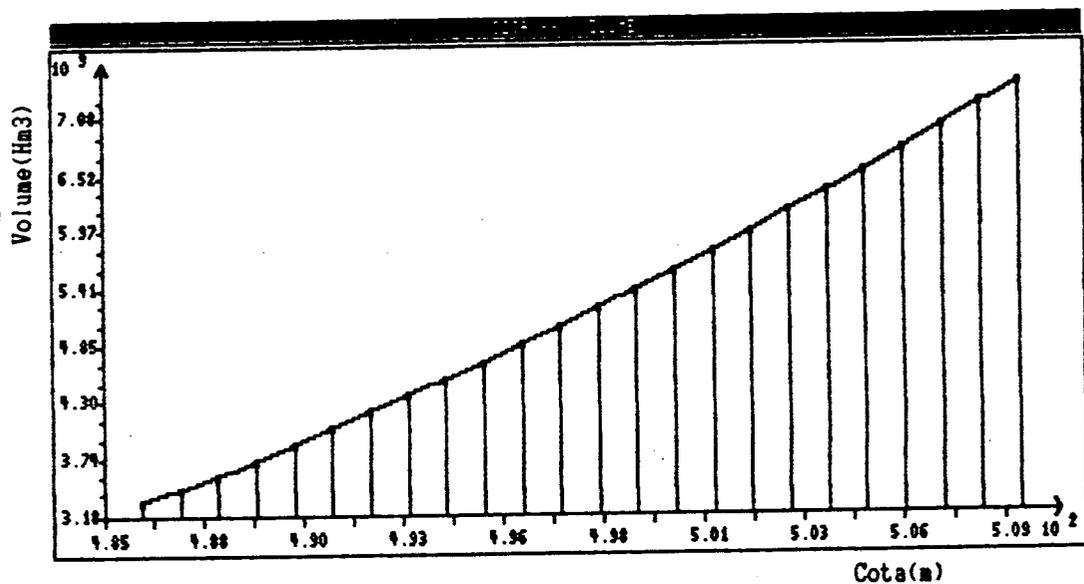


Figura 5.3 Curva do Volume Dado a Cota

#### 5.2.4 Resultados dos Testes Com o Ajustador de Curvas

O módulo cuja finalidade é ajustar curvas a pontos dados foi utilizado em todas as fases deste trabalho. O ajuste polinomial foi o que melhor ajustou coeficientes em todos os fenômenos analisados. Entre estes é possível citar o volume do reservatório como função da cota, o rendimento mecânico das turbinas, parâmetros na abertura das comportas, entre outros.

A tabela 5.3 apresenta os resultados de uma tentativa na busca de melhor precisão na predição do volume do reservatório. A figura 5.4 apresenta as curvas obtidas, aplicando-se os coeficientes da tabela 5.3. A figura 5.5 tem como finalidade a comparação qualitativa dos desvios de cada ajuste.

#### 5.3 Resultados dos Testes Com o Modelo de Abertura de Comportas

Testando-se o modelo isolado do restante do sistema o módulo de determinação da abertura de comportas forneceu os resultados esperados.

Tabela 5.3 Exemplo de Saida do Ajustador de Curvas

Pontos Dados :

X	Y
50.000	1.9793500
60.000	3.1811500
70.000	4.7037000
80.000	6.5668500
90.000	8.8087500
100.000	11.4642500
105.000	12.9578500

Polinomial  $Y=A_0+\dots+A_n \cdot X^n$

Coefficientes na Aproximacao dos Min. Quad. :

Coefficiente 0: 1.1717707345E-01  
 Coefficiente 1: -2.7345487157E-02  
 Coefficiente 2: 1.2740397117E-03  
 Coefficiente 3: -6.3381260164E-07  
 Coefficiente 4: 1.9742863960E-08

Desvio Padrao = 1.3483E-03

---

Potencia  $Y=A_0 \cdot X^{A_1}$

Coefficientes na Aproximacao dos Min. Quad. :

Coefficiente 0: 1.0219667365E-04  
 Coefficiente 1: 2.5255884336E+00

Desvio Padrao = 3.3401E-02

---

Exponencial  $Y=A_0 \cdot e^{(A_1 \cdot X)}$

Coefficientes na Aproximacao dos Min. Quad. :

Coefficiente 0: 4.1700668205E-01  
 Coefficiente 1: 3.3458823479E-02

Desvio Padrao = 5.9609E-01

---

Logaritimica  $Y=A_0 \cdot \ln(A_1 \cdot X)$

Coefficientes na Aproximacao dos Min. Quad. :

Coefficiente 0: 1.4776888758E+01  
 Coefficiente 1: 2.1022758682E-02

Desvio Padrao = 1.1003E+00

---

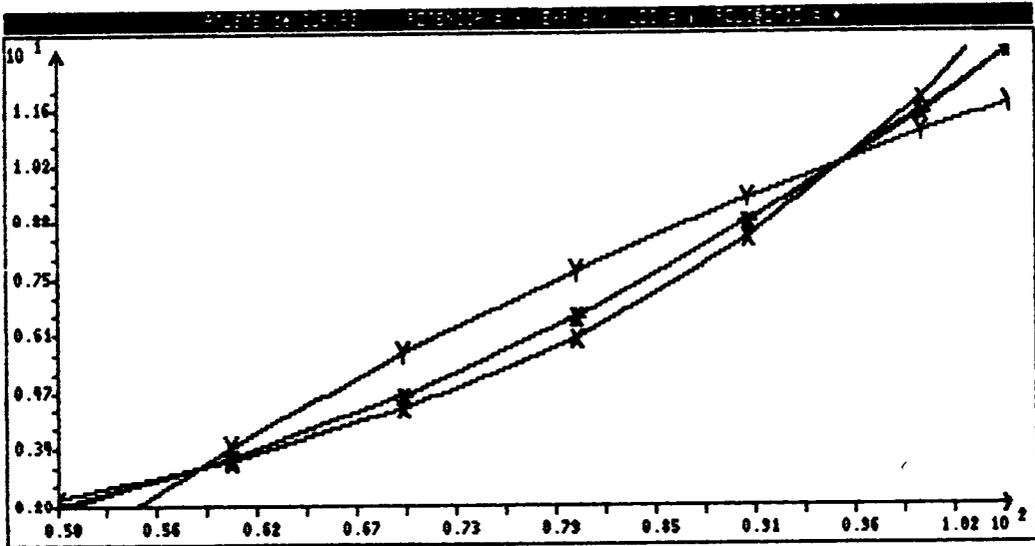


Figura 5.4 Curvas Obtidas da Tabela 5.3

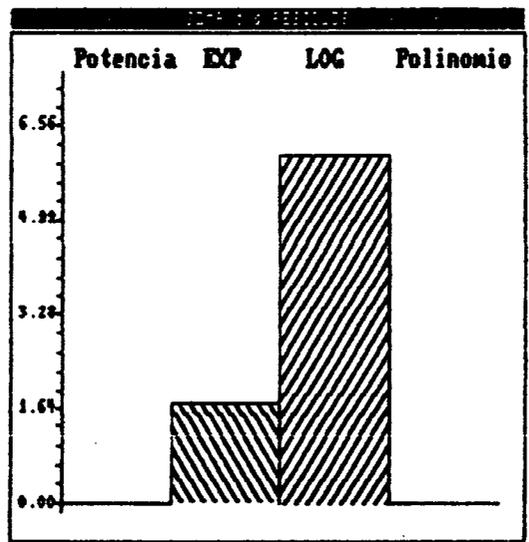


Figura 5.5 Desvios de Cada Tipo de Ajuste

Usando este modelo encontrou-se, em todas as análises, configurações que produziram afastamentos mínimos do objetivo final. Estas diferenças se devem exclusivamente à natureza discreta do passo da abertura das comportas (20 cm no protótipo) quando contrastada com a continuidade do fluxo vertido.

A tabela 5.4 apresenta resultados de alguns dos casos testados variando-se a vazão requerida, o nível do reservatório e o erro admissível.

Tabela 5.4 - Configuração das Comportas

Configuracao das Comportas $N_{ar} = 500m$		
No. Comp	Abertura(m)	Vazao (m <sup>3</sup> /s)
1	0.20	140.00
8	0.20	140.00
2	0.20	140.00
7	0.20	140.00
3	0.20	140.00
6	0.20	140.00
4	0.20	140.00
5	0.00	0.00
Vaz. Req. = 1000 ; Vaz. Obtida = 981 (m <sup>3</sup> /s)		

Tabela 5.4 - Continuação

Configuracao das Comportas Nar = 500m		
No. Comp	Abertura(m)	Vazao (m3/s)
1	1.00	263.00
8	1.00	263.00
2	1.00	263.00
7	1.00	263.00
3	1.00	263.00
6	1.00	263.00
4	1.00	263.00
5	0.80	233.00

Vaz. Req. = 2000 ; Vaz. Obtida = 2077 (m3/s)

Configuracao das Comportas Nar = 500m		
No. Comp	Abertura(m)	Vazao (m3/s)
1	1.80	382.00
8	1.80	382.00
2	1.80	382.00
7	1.80	382.00
3	1.60	353.00
6	1.60	353.00
4	1.60	353.00
5	1.60	353.00

Vaz. Req. = 3000 ; Vaz. Obtida = 2937 (m3/s)

Configuracao das Comportas Nar = 505m		
No. Comp	Abertura(m)	Vazao (m3/s)
1	0.80	272.00
8	0.80	272.00
2	0.80	272.00
7	0.80	272.00
3	0.80	272.00
6	0.60	236.00
4	0.60	236.00
5	0.60	236.00

Vaz. Req. = 2000 ; Vaz. Obtida = 2068 (m3/s)

Tabela 5.4 - Continuação

-----  
 Configuracao das Comportas Nar = 505m  
 -----

No. Comp	Abertura(m)	Vazao (m3/s)
1	1.40	378.00
8	1.40	378.00
2	1.40	378.00
7	1.40	378.00
3	1.40	378.00
6	1.20	343.00
4	1.20	343.00
5	1.20	343.00

-----  
 Vaz. Req. = 3000 ; Vaz. Obtida = 2921 (m3/s)  
 -----

-----  
 Configuracao das Comportas Nar = 505m  
 -----

No. Comp	Abertura(m)	Vazao (m3/s)
1	0.80	272.00
8	0.80	272.00
2	0.80	272.00
7	0.80	272.00
3	0.60	236.00
6	0.60	236.00
4	0.60	236.00
5	0.60	236.00

-----  
 Vaz. Req. = 2000 ; Vaz. Obtida = 2032 (m3/s)  
 -----

-----  
 Configuracao das Comportas Nar = 505m  
 -----

No. Comp	Abertura(m)	Vazao (m3/s)
1	1.40	378.00
8	1.40	378.00
2	1.40	378.00
7	1.40	378.00
3	1.40	378.00
6	1.40	378.00
4	1.20	343.00
5	1.20	343.00

-----  
 Vaz. Req. = 3000 ; Vaz. Obtida = 2957 (m3/s)  
 -----

#### 5.4 Resultados dos Testes Com o Modelo de Controle de Vazão

O Modelo de Controle de Vazão, associado ao Quadro de Referência de Operação, foi exaustivamente testado. O protótipo utiliza este Modelo para responder às solicitações de ações a empreender. Por sua importância na operação real, existe uma alternativa no sistema para treinamento apenas deste módulo. Atuando em conjunto ou isolado, é necessário apenas a informação das variáveis de controle. Não ocorreu, nos testes executados, qualquer divergência entre a implementação e o esquema atualmente em uso.

#### 5.5 Resultados dos Testes Com o Modelo de Linguagem Natural

O Modelo de Linguagem Natural foi testado de duas formas. O primeiro teste foi realizado utilizando-se o proposto em [7]. Este modelo permite a análise de frases contendo substantivos, adjetivos, pronomes, preposições, verbos e advérbios. A experimentação obteve igualmente os resultados desejados para esta alternativa.

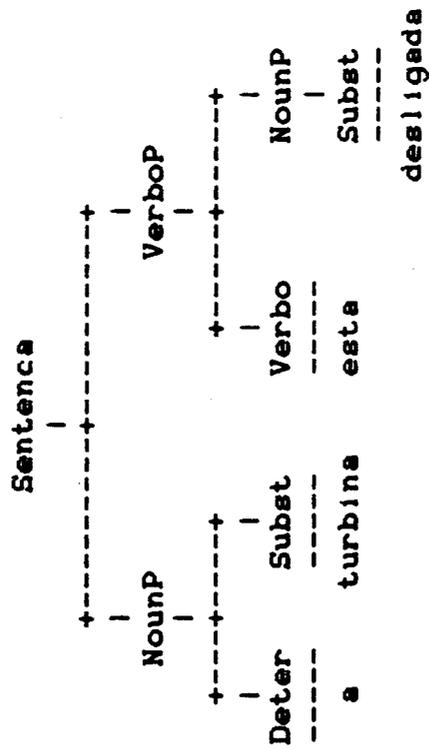
O segundo modelo testado é mais limitado que o primeiro, por não permitir o uso de adjetivos e preposições. Esta limitação pode facilmente ser eliminada.

Assim como o primeiro modelo, este também é recursivo descendente. No entanto, a sua finalidade é apenas ilustrativa, já que permite a decomposição de sentenças com a sua representação gráfica.

A figura 5.6 apresenta um resultado obtido com este último modelo. O emprego deste esquema facilita a visualização da implementação de comandos. Isto é possível, já que em um estado o sujeito é entendido como o substantivo de NOUNP. A ação a empreender é o verbo de VERBOP, recursivamente como mostra a figura 5.7.

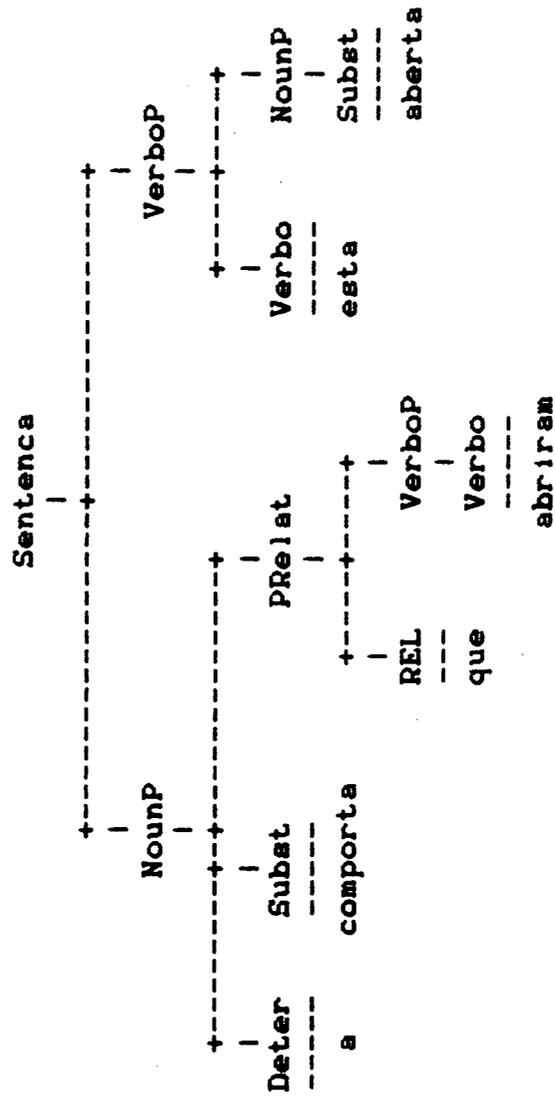
Os dois modelos testados não aceitam sentenças que estejam de acordo com a gramática G1. Todas as simplificações impostas ainda permitem uma mímica de uma linguagem natural completa. Ressalta-se que erros graves podem vir a satisfazer esta gramática. Esta possibilidade é verificada na figura 5.8.

Figura 5.6 - Exemplo da Decomposição de Sentenças Pelo Parser Selecionado



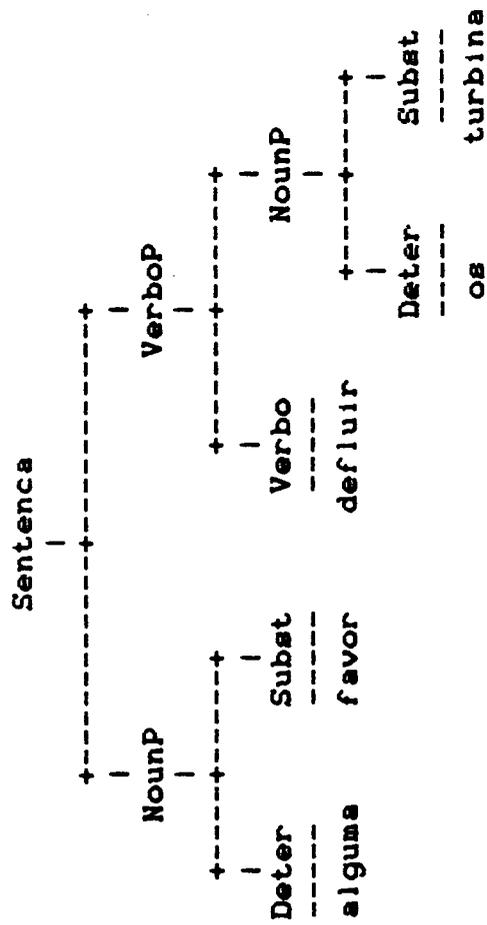
Sentença = a turbina esta desligada  
 Frase Parseada = frase(nounp(deter("a"), "turbina", sem), verbop("esta", nounp(sem),  
 desligada", sem)))

Figura 5.7 - Decomposição Recursiva Executada Pelo PRDLC



Sentenca = a comporta que abriam esta aberta  
 Frase Parceada = frase(nounp(deter("a"), "comporta", relcl("que", verb("abriam"))),  
 verbep("esta", nounp(sem, "aberta", sem)))

Figura 5.8 - Exemplo de Estrutura Incorreta que Satisfaz a Gramática G1



Sentenca = alguma favor defluir os turbina  
 Frase Parseada = frase(nounp(deter("alguma"), "favor", sem), verbop("defluir", nounp(deter("os"), "turbina", sem)))

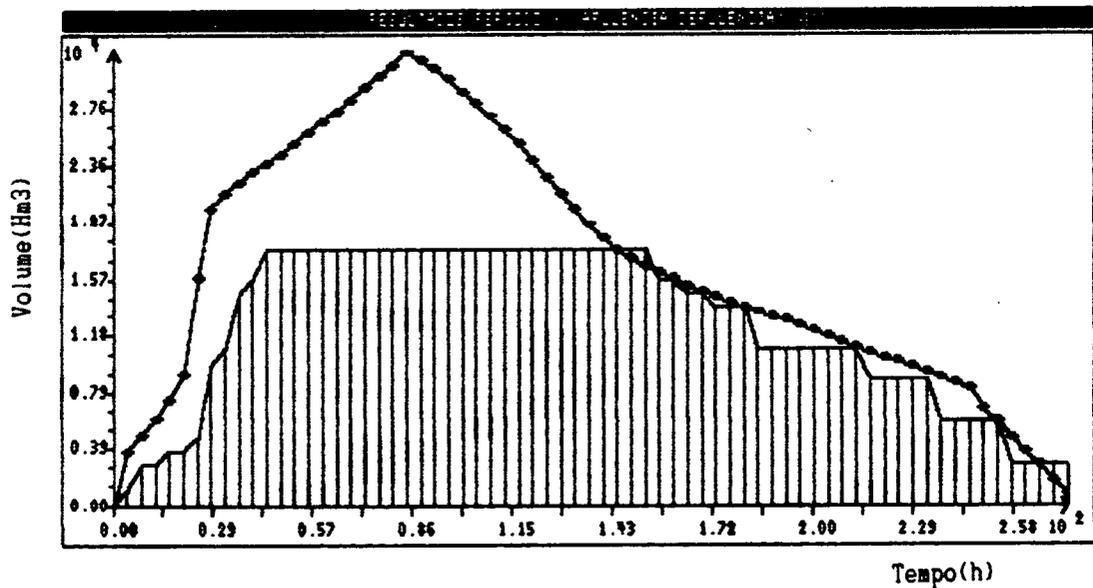


Figura 5.9 - Resultado Final de Um Treinamento na Onda de Cheia de 1983

## 5.6 Análise do Protótipo

O protótipo teve como objetivo inicial avaliar a validade da proposta de aplicação de técnicas de inteligência artificial no treinamento de operadores de hidrelétricas.

O protótipo também serviu para a realização da etapa de captura de conhecimento, do Perito do Domínio pelo Engenheiro de Conhecimento. Conforme [44], entre outros autores, esta é a etapa mais árdua na elaboração do sistema

perito.

A figura 5.9 apresenta, de forma gráfica, os resultados finais do treinamento da onda de cheia de 1983. Este mesmo gráfico pode ser apresentado com os valores intermediários da operação simulada.

Existiram, neste trabalho, fatores limitantes pela ferramenta computacional e pelo tempo disponível.

Este protótipo permitiu, adicionalmente, verificar-se que o problema do modelo matemático está praticamente solucionado e que outros devem ser pesquisados em próximos estudos.

Adotando-se a terminologia da engenharia de software este sistema é transportável apenas a equipamentos operando sob o MS-DOS. Muitas modificações seriam necessárias para que fosse possível sua utilização em sistemas maiores. A usabilidade e manutenibilidade, no entanto, são facilitadas, pelas interfaces e pelo código modulado e compacto.

## 5.7 Análise da Ferramenta Computacional

A ferramenta computacional adotada, um compilador Prolog [26], possui diferenças com relação ao Prolog de Edinburgo. As implicações de suas características serão comentadas nas seções a seguir.

### 5.7.1 Características Favoráveis da Ferramenta Adotada

A ferramenta adotada, no que se refere ao desenvolvimento de protótipos, pode ser considerada como adequada. Isto porque é eficiente em termos de tempo de compilação e de processamento. Facilita, desta forma as modificações frequentes nesta fase de desenvolvimento de um sistema perito.

### 5.7.2 Características Desfavoráveis da Ferramenta Adotada

A ferramenta selecionada, por ser um compilador, não permite a modificação de suas regras. Por este motivo a introdução de novas usinas requer a recompilação das regras do módulo de controle de vazão. O módulo de Abertura de Comportas, por sua vez, necessitou ser reformulado, no que diz respeito a representação do conhecimento, de maneira a permitir modificações apenas nos fatos e não nas regras. Fatos são armazenáveis em arquivos, podendo ser modificados

por qualquer processador de texto.

Adicionalmente, este compilador não possui facilidades para base de dados externos, com suporte para acesso por métodos que utilizem arvores B+. Isto implica na limitação do tamanho máximo das bases de dados do sistema. O gerador de interfaces gráficas é mal desenvolvido e requer a construção de muitas primitivas, induzindo desta forma ao aparecimento de muitos erros no processamento.

A memória disponível para a Pilha (Stack), máximo de 64Kb, é insignificante para aplicações com representações recursivas. Observe que um dos pilares da representação do conhecimento está apoiado nesta facilidade. Como exemplo, cita-se que até a tabela gerada na secção 5.3 chegou a ultrapassar este limite.

Outra característica desejável de um sistema computacional está a habilidade de corrigir erros de execução via aprendizado. Este aprendizado não foi considerado pela adoção de um compilador e pela falta de conhecimento da fronteira do domínio. No entanto espera-se que o sistema só venha a falhar apenas onde os peritos humanos vierem a falhar. Outro recurso que os sistemas existentes carecem, em relação a instrutores é o de explicar como se identificou-se que determinada alternativa seria mais apropriada que outra em certas circunstâncias. A

versão 2.0 deste compilador possui meios para eliminar esta deficiência.

## 5.8 Conclusões

Neste capítulo foram apresentados os resultados de testes no protótipo, que visam avaliar a validade da proposta deste trabalho.

No próximo capítulo serão apresentados as conclusões finais e recomendações para futuros trabalhos.

**CAPÍTULO VI**  
**CONCLUSÕES E RECOMENDAÇÕES**

**6.1 Conclusões**

Esta dissertação introduz uma proposta diferente na abordagem de um treinamento de operadores de hidrelétricas. Cabe salientar que não foram encontrados na literatura trabalhos para treinamento de operadores de reservatórios que utilizem técnicas de inteligência artificial.

Após experimentações os peritos do domínio, concluíram que existem as seguintes diferenças entre os sistemas convencionais e o proposto neste trabalho :

**Sistemas Convencionais :**

- Respostas mais rápidas na simulação de operação de cheia;
- Possibilidade de simulação de uma cascata em microcomputador;

**Sistema Proposto :**

- Mecanismos de auxílio ao usuário favorável ao modelo que utiliza técnicas de inteligência artificial;
- Facilidades para operação de apenas segmentos do modelo do sistema de treinamento e troca de informações por meio de linguagem natural inexistentes nos modelos convencionais.

No que se refere a interface gráficas e precisão numérica dos resultados, é opinião dos peritos que os resultados são equivalentes nas duas concepções;

Observe que os modelos convencionais comparados pelos peritos são produtos comerciais plenamente desenvolvidos. Por sua vez, o modelo implementado neste trabalho é apenas um protótipo, existindo nele um potencial ainda a ser desenvolvido.

Outras contribuições decorrentes desta proposta surgiram durante sua implementação e podem ser citadas:

- A utilização de partes do modelo no auxílio de tarefas dos peritos do domínio, tais como o procedimento de determinação adequada de comportas e o ajuste de curvas, entre outros;
- Introdução de conceitos de inteligência artificial em empresas concessionárias de energia elétrica, como uma ferramenta alternativa para treinamento e apoio a operação.

## 6.2 Recomendações

Existem diversos problemas a serem abordados a partir deste trabalho. Entre eles, pode-se citar:

- A análise da proposta realizada com ferramentas de programação orientada a objeto, como proposto no capítulo 5;
- o desenvolvimento da interface de linguagem natural restrita, que aumentaria a usabilidade do produto final;
- A aplicação de mecanismos de explicações, que se faz necessária a medida que o sistema "cresce";
- A introdução de interfaces de comando que permitam o acionamento de dispositivos de controle de reservatório, visto que neste processo o tempo real é maior que o de respostas do sistema;
- A experimentação desta proposta em estações de trabalho mais poderosas em termos de memória e capacidade de processamento, que o equipamento utilizado neste trabalho, (microcomputador IBM-PC-Xt);
- Uma vez que a limitação imposta à recursividade tenha sido contornada, várias usinas poderão ser incluídas, permitindo que uma cascata seja simulada.

O modelo como implementado não emprega regras probabilística , com fatores de incerteza e recomenda-se quando necessário os trabalhos de Sugeno!31! e Zimmermann !32! que tratam de aplicações de conjuntos difusos.

## REFERÊNCIAS BIBLIOGRÁFICAS

1. Barr, A., Feigenbaum, E., "The Handbook of Artificial Intelligence", Vol 1, Pitman, London, Inglaterra, 1981.
2. Barr, A., Feigenbaum, E., "The Handbook of Artificial Intelligence", Vol 2, Pitman, London, Inglaterra, 1981.
3. Hegenberg, L., "Logica : O Cálculo dos Predicados", USP, SP, 1973.
4. Casanova, M., Giorno, F., Furtado, A., "Programação em Lógica e a Linguagem Prolog", Blucher, SP, 1987.
5. Rich, E., "Artificial Intelligence", McGraw-Hill, NY, EUA, 1983.
6. Robinson, P., "Using Turbo Prolog", Osborne McGraw-Hill, Berkeley, Ca, EUA, 1987.
7. Schildt, H., "Advanced Turbo Prolog", Osborne McGraw-Hill, Berkeley, Ca, EUA, 1987.
8. Kluzniak, F., Szpakowicz, S., "Prolog for Programmers", Academic Press, London, Inglaterra, 1985.
9. Hayes-Roth, F., Waterman, D., Lenat, D., "Building Expert Systems", Addison-Wesley, Massachusetts, EUA, 1983.
10. Townsend, C., "Introduction to Turbo Prolog", Sybex, Berkeley, Ca, EUA, 1987.
11. Coombs, M., "Developments in Expert Systems", Academic Press, London, Inglaterra, 1984.
12. Keller, R., "Expert System Technology Development & Application", Prentice-Hall, NJ, EUA, 1987.
13. Bratko, I., "Prolog Programming for Artificial Intelligence", Addison-Wesley, Massachusetts, EUA, 1986.
14. Lucid, "Why LISP?", Menlo Park, Ca, EUA, 1987.

15. Lechner, H., "Quintus Prolog", Quintus, Mountain View, Ca, EUA, 1987.
16. Hindin, H., "Intelligent Tools Automate High-Level Language Programming", Computer Desing, May 15, 1986, pg 45-56.
17. Osborne, R., "Centralized Diagnostics Uses Artificial Intelligence", Modern Power Systems, Feb, 1987, pg 53-57.
18. Wittig, T., "Power Distribution Falls Under KRITCIS's Eye", Scada Systems, Jan, 1987, pg 49-55.
19. Covington, M., Vellino, A., "Prolog Arrives", PC Tech Journal, Nov, 1986, pg 53-69.
20. Kumara, S., et alii, "Expert Systems in Industrial Engineering", Internat. Journal Prod. Research, Vol 24, No. 5, 1986, Pg 1107-1125.
21. Milacic, V., Miler, A., "Artificial Intelligence-Morphological Approach as a New Technological Forecasting Technique", Internat. Journal Prod. Research, Vol 24, No. 6, 1986, pg 1409-1425.
21. Doukidis, G., Ray, P., "Research into Expert Systems to Aid Simulation Model Formulation", Journal Opl Res. Soc., Vol 36, No. 4, 1985, pp 319-325.
22. Thompson, B.&B., "Taxing the Expert System Shells", AI Expert, Jun 1987, pp 23-29.
23. Lazarev, G., "Solving Problems With Prolog", AI Expert, Jul 1987, Vol 2, No. 7, pp 58-68.
24. Salzberg, S., "Knowledge Representation in the Real World", AI Expert, Aug 1987, Vol 2, No. 8, pp 32-39.
25. Marcot, B., "Testing Your Knowledge Base", AI Expert, Aug 1987, Vol 2, No. 8, pp 42-47.
26. Borland, "Turbo Prolog the Natural Language of Artificial Intelligence", Manual, Scotts Valley, Ca, EUA, 1986.
27. Schreiber, G., "Usinas Hidrelétricas", Edgar Blucher, SP, 1978.
28. Rabuske, R., "Inteligência Artificial", Apostila do curso de mesmo nome, 1987.
29. Genaro, S., "Sistemas Especialistas - O Conhecimento Artificial", LTC, SP, 1987.

30. ELETROSUL, "Instruções Técnicas - Hidrologia - U.H. Salto Osório", Relatório Interno, SC, 1985.
31. Sugeno, M., "Industrial Applications of Fuzzy Control", North-Holland, Amsterdam, Holanda, 1985.
32. Zimmermann, H., "Fuzzy Set Theory and Its Applications", Kluwer-Nijhoff, Massachusetts, EUA, 1984.
33. Kowalski, R.A., "Logic for Problem Solving", North Holland, Amsterdam, Holanda, 1979.
34. Brittain, S., "Understanding Natural Languages", AI Expert, Vol 2, No. 5, May 1987, pg 31-38.
35. Barstow, D., et alli, "Language Tools for Knowledge Engineering", Building Expert Systems, Addison-Wesley, Massachusetts, EUA, 1983.
36. Peled, A., "The Next Computer Revolution", Scientific American, Vol 257, No.4, 1987.
37. Krushopf Neto, R., "Simulador Hidráulico 'K'scata'", Relatório Final DHIO-001/87, Centrais Elétrica do Sul do Brasil S/A, SC, Jan/87.
38. Canjani, A., Moraes, Carlos, "Controle de Vazões Defluentes das Usinas: Aferição de Vertedouros e Medidas de Vazões Turbinadas", V SNPTEE, PE, Brazil 1979.
39. Borland, "Turbo Prolog Toolbox", Manual, Scotts Valley, Ca, EUA, 1986.
40. Stoecker, "Desing of Thermal System", McGraw-Hill / Kogakusha, 2ª Ed., Tokio, Japão, 1980.
41. Borland, "Turbo Graphix Toolbox", Manual, Scotts Valley, Ca, EUA, 1986.
42. Takahashi, T., "Introdução à Programação Orientada a Objetos", III EBAI, Ctba, Jan 1988.
43. Barker, P., "Expert Systems in Engineering Education", Eng. Applications of Artificial Intelligence, Vol 1, Nº 1, Mar 1988.
44. Peat, F. D., "Artificial Intelligence: How Machine Think", Bean, Ny, EUA, 1988.
45. Thé, J., Barcia, R., "Expert System and Decision Support for Hydroelectric Reservoir Control", 36th ISMM International Conference, Barcelona, Espanha, 1988.

**ANEXO 1**  
**Dados Referentes a Usina Hidrelétrica de Salto Santiago**

**DADOS TÉCNICOS DA UHSS**

**a) Condições a Montante**

- Área da bacia hidrográfica	43.900 km <sup>2</sup>
- Área da bacia intermediária(Areia/UHSS)	13.900 km <sup>2</sup>
- Área inundada ( cota 506,0 m)	208 km <sup>2</sup>
- Relação da influência da bacia intermediária	48 %
- Volume Morto	2.658E06 m <sup>3</sup>
- Volume útil	4.094E06 m <sup>3</sup>
- Cota Crista (barragens e auxiliares)	510,00 m
- Nível Operativo Min.	481,00 m
- Nível Max. Normal	506,00 m
- Nível Max. Maximorum	509,00 m
- Nível Operat. Normal ou de Volume de Espera	503,88 m
- Vazão Max. Observada	18.000 m <sup>3</sup> /s
- Variação Max. Observada em 24 horas	7.000 m <sup>3</sup> /s
- Evaporação Média	500E06 m <sup>3</sup> /dia
- Deslocamento de Onda(Areia/UHSS)	7 / 8 horas
- Barragens de Terra Auxiliares	3

**b) Unidades Geradoras**

- Geração Total Projetada (6 Unidades)	1998 MWe
- Geração Total Instalada (4 Unidades)	1332 MWe
- Engolimento Unitário	348,0 m <sup>3</sup> /s
- Queda Máxima (506,00 - 393,00)	113,00 m
- Queda Mínima Operativa	82,00 m

**c) Orgãos de Descarga**

- Número de Comportas	8
- Largura da Comporta	15,30 m
- Descarga por Comporta (NA. Max. Normal)	2800 m <sup>3</sup> /s
- Capacidade de Descarga do Vertedor (NA. Max. Maximorum)	27830 m <sup>3</sup> /s
- Cota do Ponto de Toque da Comporta	485,23 m
- Tipo de Acionamento	ELÉTRICO
- Tipo de Controle	Manual/Local
- Tipos de Recursos de Alimentação de Energia	
. Normal	
Fonte Principal - Serviço Aux. da Usina	CM

Fonte Alternativa - Serviço Aux. da Substação SE

. Emergência  
Grupo Motor Diesel/Gerador 1  
Capacidade de Acionamento Simultâneo 1 Comporta

d) Condição de Jusante

- NA. Mínimo 393,00 m  
- NA. Máximo 400,00 m  
- NA. Máx. Maximorum 419,00  
- Restrição de Defluência  
. Inundação da Casa de Força da UHSS  
Vazão de Restrição Variável em  
Função das Vazões Incrementais  
do Rio Chopim e Salto Santiago/  
Salto Osório -->  
QRSS/JMF = 17000 - (Q1Chopin - QISS/SO) 12000 m3

ANEXO 2  
Procedimentos de Operação nos Diversos Estados

Situação Normal  
-----

- Efetuar manobras de Comportas de conformidade com o solicitado pelo COSI.
- Efetuar inspeções e os levantamentos dos dados hidrológicos de acordo com periodicidades fixadas.
- Zelar pela manutenção do reservatório em nível pré-estabelecido.
- Alertar COSI sobre desvios ou outros problemas.
- Aferir os dispositivos de supervisão, providenciando para seu correto desempenho.
- Registrar em formulários próprios as operações hidráulicas do reservatório.
- Zelar para que os escalonamentos de descargas sigam as leis de manobras estabelecidas.
- Diligenciar operações corretas e adequadas de comportas, como sequência, abertura, submersão, etc.
- Manter os equipamentos em pleno e perfeito estado de funcionamento.
- Intercâmbio de informações com COSI, passando e recebendo dados hidrológicos nas horas aprasadas.
- Zelar pelo treinamento do pessoal envolvido, tanto operadores como de manutenção e certificar a sua habilitação.
- Manter os operadores instruídos, a par das instruções e com conhecimento de todos detalhes que trate de segurança das barragens.
- Promover a reavaliação de dados para o aprimoramento das instruções sobre o controle de reservatório.
- Assumir o controle do reservatório em caso de perda de comunicação.

( Extraído do Volume 62 O 07 Num. 01 folhas 09-a e 10-a de 17.01.85)

---

### Situação Atenção

---

- Repetem-se as atribuições da Situação Normal( exceto o primeiro item).
- Comandar as operações hidráulicas.
- Proceder o controle do reservatório mantendo o nível na Cota fixada na instrução específica da Usina.
- Manter as descargas dentro de normas fixadas.
- Efetuar registro de dados oriundos do despacho.
- Transmitir ao COSI Informações sobre a situação da Usina.
- Aquilatar a possibilidade de danos a jusante.
- Manter-se a par das medidas tomadas no reservatório a montante.
- Procurar detetar eventuais indícios que possam indicar agravamento da situação.
- Aproveitar as informações e orientações recebidas de outras áreas em especial do COSI, para subsidiar as decisões.
- Avisar a usina a jusante das operações rotineiras de comportas
- Notificar antecipadamente a jusante e COSI de qualquer operação de comporta fora de rotina.
- Decidir pela antecipação na escala de situação no caso de perda de confiabilidade, comunicando ao COSI e DGH/SEDE.
- RESPONSABILIDADE DE DECISÃO : Chefe de Operação com o apoio imediato do Chefe de Turno.

( Extraído do Volume 62 0 07 Num. 01 folhas 10-a e 11-a de 17.01.85)

---

## Situação de Alerta

---

- Repetem-se as atribuições da Situação Atenção(exceto o ultimo ítem).
- Manter-se informado da evolução dos acontecimento na bacia onde se localiza.
- Detetar e comunicar, em devido tempo, ao DGH/SEDE a entrada em sobreaviso de pré-emergência.
- Promover contatos com órgãos de divulgação da região, alertando moradores ribeirinhos quando o evento se aproximar dos pontos de restrição.
- Tomar medidas de segurança na Usina e nos locais de restrições, quando for o caso.
- Alertar para imediata ativação da Comissão de Emergência para Enchente.
- RESPONSABILIDADE DE DE DECISÃO : Chefe da Usina com assessoramento do Chefe de Operação.

( Extraído do Volume 62 0 07 Num. 01 folhas 10-a e 11-a de 17.01.85)

---

## Situação de Emergência

---

- Repetem-se as atribuições da Situação Atenção(exceto o último segundo e terceiro item).
- Repetem-se as atribuições da Situação Alerta(exceto o último e primeiro item).
- Registrar no Livro de Ocorrências a criação da Comissão de Emergência para Enchente.
- Promover a execução de manobras das comportas de forma a atender a descargas determinada pela Comissão de Emergência.
- Informar constantemente a Comissão sobre a evolução das situações em sua area, prestando-lhe subsidios para decisões.
- Estabelecer turnos de manutenção elétrica/mecânica.
- Tomar providências para mobilizar pessoal e recursos da Usina para enfrentar a nova situação.
- Assumir as diretrizes e a responsabilidade das decisões em caso de isolamento ( perda de comunicações).

( Extraído do Volume 62 0 07 Num. 01 folhas 11-a e 12-a de 17.01.85).

---

## APÊNDICE 1

### FONTE DO SISTEMA DE TREINAMENTO

Code = 5000

/\* Definicoes Preliminares Obrigatorias no Turbo-Prolog \*/

Domains

ListInt = Integer\*

ListSymb= Symbol\*

Dcomp = d(Integer,Real,Real)

LDcomp = Dcomp\*

file = user;sist;ondaxy;extra

bond = b(Integer,Real)

lbond = bond\*

Var\_Control = symbol

Num\_Comporta = integer

Lista\_Comporta = integer\*

/\*\*\*\*\*\*\*/

Database

/\*\*\*\*\*\*\*/

/\* ConfigVert = Database p/ Atualizacao das Comportas\*/

ConfigVert(Ldcomp)

/\* Numero da Comporta Impedida \*/

Comporta\_Impedida(Integer,Integer)

/\* Dados Temporarios :

No Período, VazAf., Vaz.Vert., Vaz.Tur., Cota \*/

DadoTemp(Integer,Real,Real,Real,Real)

/\* Dir = 0 --> S1 > Vreq e Lista Nao Invertida \*/

/\* = 1 --> S1 < Vreq e Lista Invertida \*/

Direcao(Integer)

/\* Flag de Parada p/ RepeatEsp (0 = ok, 1 = PARE)\*/

Flag(Integer)

/\* Valores das Variaveis de Controle e Restricao \*/

Info(Symbol,Real)

/\* qmaxnor = Vazao Afluente Max. Normal \*/

/\* qrUHSS = Vazao de Restricao de UHSS \*/

```

/* qrJMF = Vazao de Restricao p/ JMF */
/* Cota Inf. do Vol. de Espera da UH JMF */
/* Cota Sup. do Vol. de Espera da UH JMF */
/* Cota Inf. do Vol. de Espera da UHSS */
/* Cota Sup. do Vol. de Espera da UHSS */
/* qn1 = Descarga Normal 1 */
/* qn2 = Descarga Normal 2 */
/* defind = Defluencia Ind. P/ o Resev. */
/* deltad = Step na Abert. das Comp. */
/* nar = Nivel d'agua */
/* numturb = Numero de Turb. Intaladas */
/* qaf = Vazao Afluente */
/* volres = Vol.do Reservatorio */
/* vlamlivre = Vazao Lamina Livre */
/* vparcial = vazao parcial de 1 comporta*/
/* vtur = vazao de uma turbina */
/* Vazvert = Vazao Inst.no Vertedouro */
/* Vtotturb = Vazao Total Turbinada */

```

InfoI(Symbol,Integer)

```

/* Nome da Usina em Treinamento */

```

NomeUsina(String)

```

/* Contem Uma Lista (bond) de uma Onda de Cheia */

```

Onda(lbond)

```

/* Sequencia Especificada da Abertura de Comportas */

```

Seq\_Esp\_Abert\_Comp(Integer,Integer,ListInt)

```

/* Status Contem o Estado Hirologico Presente */

```

Status(String)

```

/* Restricao contem todas as restricoes a serem observadas */

```

Restricao(Symbol,Real)

```

/* Configuracao da Turbina no Periodo Integer */

```

ListaTurb(ListInt)

Turb(Integer,Ldcomp)

```

/* Vertedouro = Database da Configuracao de Comp*/

```

Vertedouro(Ldcomp)

include "MenuO.Pro"

```

/*****/

```

Predicates

```

/*****/

```

## Depura

```
Abert_Aprox(Real,Real)
Abertura_Comporta(ListInt)
Ajuste_Contador(Integer)
/* Adiciona o Fato Comporta Impedida na Base de Dados */
Assemble_Comp(integer)
/* Atualiza o Valor Real Contido em Info Sob Symbol */
AtualDir(Integer)
AtualInfo(Symbol,Real)
AtualInfoI(Symbol,Integer)
AtualFlag(Integer)
AtualStatus(String)
Calc_Ab(ListInt,Ldcomp,Real,Real)
Clear_Comp_Imp
Checa_Abertura_Max(Real)
Checa_Def_Max(Real)
Checa_Cota(Real)
Checa_Cota_Max(Real)
Checa_Cota_Min(Real)
Checa_Pot_Max(Real)
Checa_Status(Real,Real)
Checa_Jus(Real,Real)
Checa_Jus_Min(Real)
Checa_Jus_Val(Real,Real)
Checa_Val_Pos(Real)
/* Seleciona o Arquivo de Dados Para Consult */
Escolha_Arquivo_BD
Escolha_Arquivo_Onda
Escolha_Arquivo_Qualquer
Execute_Gravacao(String,String)
Fechamento_Comporta(ListInt,ListInt)
Gravar(Char,String,String)
ImpResult(ListInt)
/* Seleciona um Elemento de uma Lista */
index(LISTInt,INTEGER,Integer)
Ler_Comp_Imp
/* Verifica Pertinencia de um Elemento em uma Lista */
Member(symbol,List symb)
Member(Integer,ListInt)
/* Eh o Goal do Sistema */
Objetivo(Integer)
/* Executa Selecao do Menu #1 */
Processe(Integer)
/* Executa Selecao do Menu #2 */
Processar(Integer)
/* Le Novos Dados da Abertura das Comportas */
Renovar_Dados_Ab_Comp
Reverse(Ldcomp,Ldcomp,Ldcomp)
```

```

Reverse(ListInt,ListInt,ListInt)
Reverse_List(Ldcomp,Ldcomp)
Reverse_List(ListInt,ListInt)
Repeat
RepeatEsp
/* Determina a Sequencia de Manobra das Comportas */
Sequencia_Abertura(ListInt)
T_Int(Integer,Real)

/* Elimina Duplicatas em Uma Lista */
Unik(ListInt,ListInt)
Unik(ListSymb,ListSymb)
/* Appenda Duas Listas */
append(ListInt,ListInt,ListInt)
append(ListSymb,ListSymb,ListSymb)

Acha_status(Real,Real,String)
Checa_Nivel_Min(Real,String)
Checa_QN1(String)
/* Checa_Rest_X Verifica se a Restricao de X Foi Violada */
Checa_Rest_JMF(Real,Real,String)
Checa_Rest_UHSS(Real,Real,String)
/* Checa_VE_X Verifica Se o Vol.de Espera de X esta Ocupado*/
Checa_VE_JMF(Real,String)
Checa_VE_UHSS(Real,String)
Erestricao(Symbol,Real)

/* Ler Cota e Vazao do Reservatorio */
Ler_Cota_Vazao
Quebra_Rest_JMF(Real,Real,String)
Quebra_Rest_UHSS(Real,String)

/* ----- */
/* Predicates do Modelo Matematico */
/* ----- */

AtualTurb(Integer,Ldcomp)
/* Calcula Volume do Reservatorio Dado Cota */
Calculo_Cota_Volume(Real)
/* Calcula a Variacao do Volume Dados Intervalo de */
/* Tempo, Vazoes Aflente,Vertida e Turbinadas */
Calculo_Var_Vol(Integer,Real,Real)
/* Calcula Cota Dado Volume do Reservatorio */
Calculo_Vol_Cota(Real)
ChecaVd(Real,Real,Real)
/* Determina Vazao Defluente dado a Afluencia p/ que */
/* Nao se Atinja Certa Cota */
Defluencia_Indicada(Real,Real)
/* Le Potencia Gerada por Cada Turbina */

```

```

Leia_Dados_Turb
/* Rotina Auxiliar de Leia_Dados_Turb */
Ler_Ger_Turb(ListInt,Ldcomp,Real,Real)
Lvt(real,real,real,real)
/* Poli Calcula o Valor de Um Polinomio de Grau 2 */
Poli(Real,Real,Real,Real,Real)
/* Calcula o Rendimento de Uma Turbina */
Rend(Real,Real,Real)
/* Calcula Vazao em Lamina Livre */
Vazao_Lamina_Livre(Real)
Vazao_Parcial(Real,Real)
Vazao_Turbinada(Real,Real)

/* ----- */
/* Predicates da Abertura Automatica de Comportas */
/* ----- */
Predicates

Abertura_Adequada(Real,Real,ListInt)
AtualConf(Ldcomp)
AtualVert(Ldcomp)
Calcula_Novas_Aberturas(Ldcomp,Ldcomp,Real,Real,Real)
/* Calcula Valor da Vazao Vertida */
Calcula_Valor_Vaz_Vert(Ldcomp,Real)
/* Verifica Direcao de Manobra ( Abre ou Fecha ) */
Checa_Dir(Real,Real,Integer)
ImpListComp(Ldcomp)
Ler_Abertura_Comporta(ListInt,Ldcomp,Real)
ListaFinal
Recalculo(Real)
Status_Comp(Integer)
Testa_Conv(Real)

/* Leitura da Onda de Cheia */

AtualOnda(lbond)
Dado_Prox_Periodo(Integer,Real)
Processi(Integer)
Proc(Integer)

/* ----- */
/* Modelo Treinamento de Cheia */
/* ----- */

AtualDadoTemp(Integer,Real,Real,Real,Real)
Execute_Impressao(Integer,Real,Real,Integer)
Edit_Help_Status(String)
Help_Status
Imprimir_Tabela(Integer,Real,Real)

```

```
Inicio_Onda
Loop_Onda
Mostra(Integer)
MostraS(String)
ProcAb(Integer,Real,Real,Real)
```

```
/* ===== INICIO do SISTEMA ===== */
```

```
Goal Objetivo(0).
```

```
Clauses
```

```
Objetivo(0) :-
  Trace(off),
  OpenWrite(ondaxy,"Onda.XY"),
  MakeWindow(20,121,14," Treinamento de Operadores de Hidreletrica ",
             0,0,25,80),
  Repeat,
  ClearWindow,
  Menu(10,20,["Carregar Dados de Uma Usina",
             "Treinamento",
             "Resultados",
             "Listar",
             "Editar",
             "Shell p/ o DOS",
             "Encerrar"],
       Escolha),
  Proccesse(Escolha),
  Fail.
```

```
Objetivo(1) :-
  Trace(off),
  MakeWindow(20,121,14," Finalizacao do Treinamento ",
             0,0,25,80),
  Repeat,
  ClearWindow,
  Menu(10,20,["Resultados",
             "Listar",
             "Editar",
             "Shell p/ o DOS",
             "Encerrar"],
       Escolha),
  Escolha2 = Escolha + 2,
  Proccesse(Escolha2),
  Fail.
```

```
Proccesse(1) :- Escolha_Arquivo_BD,
                Escolha_Arquivo_Onda,!.
```

```

Processe(1) :- Write(" Erro na Leitura dos Dados "), nl.
Processe(1) :- Write(" Falhou Leitura do Bancos de Dados "),
              ReadChar(_).

Processe(2) :-
              Menu(10,20,["Operacao de Um Evento de Cheia",
                          "Regras para o Controle de Vazao",
                          "Modelo de Abertura das Comportas",
                          "Modelo Matematico da Usina"],
                  Escolha2),Processar(Escolha2).

Processe(3) :- CloseFile(ondaxy),!,System("Resul"),
              OpenAppend(ondaxy,"Onda.XY").

Processe(4) :- System("ListAR").

Processe(5) :- Escolha_Arquivo_Qualquer.

Processe(6) :- MakeWindow(18,7,0,"",0,0,25,80),
              System(""),
              RemoveWindow.
Processe(6) :- Write(" # Problema com o Command.Com."),
              Write(" Pressione Qualquer Tecla."),
              ReadChar(_),
              REmoveWindow.

Processe(7) :- Flush(ondaxy),
              CloseFile(ondaxy),
              ClearWindow, RemoveWindow,
              Exit,!.

Processe(7) :-
              Write(" Problema no Fechamento do Arquivo de Resultados."),
              nl,Write(" Pressione Qualquer Tecla."),
              ReadChar(_),
              ClearWindow,
              RemoveWindow,
              Exit.

Processar(1) :- Inicio_Onda,!,
              RepeatEsp,
              Loop_Onda,
              Fail.

Processar(2) :-
              Makewindow(2,121,14,"Regras para o Controle de Vazao",0,0,25,80),
              Ler_Cota_Vazao,
              Info(nar,Cota),Info(qaf,Q),
              Help_Status,

```

```
Acha_Status(Cota,Q,_),
ReadChar(_), RemoveWindow.
```

```
Processar(3) :-
  MakeWindow(3,121,14,"Modelo de Abertura das Comportas",0,0,25,80),
  Repeat,
  Ler_Cota_Vazao,
  Info(nar,Cota),Info(qaf,Qaf),
  Ler_Comp_Imp,
  Abertura_Comporta(Lista_Comporta),
  nl,Write("Seq. Abertura de Comportas Apropriada "),
  ImpResult(Lista_Comporta),readchar(_),
  Fechamento_Comporta(Lista_Comporta,Lista_Fecha),nl,
  nl,Write(" E do Fechamento "),
  ImpResult(Lista_Fecha),Readchar(_),!,
  Defluencia_Indicada(Cota,Qaf),Info(defind,Vreq),
  nl,Write(" Defluencia Indicada = ",Vreq),nl,
  Abertura_Adequada(Vreq,Cota,Lista_Comporta),
  RemoveWindow.
```

```
Processar(4) :-
  MakeWindow(4,7,7," Teste Modelo Matematico ",0,0,24,80),nl,
  Ler_Cota_Vazao,
  Info(nar,Cota),Info(qaf,Qaf),
  Write(" Potencia Lida de uma Turbina (MWe) --> "),
  Readreal(Pg),nl,
  Write(" Abertura de Uma Comporta (m) --> "),
  Readreal(Acomp),

  Calculo_Cota_Volume(Cota),!,
  Vazao_Lamina_Livre(Cota),
  Vazao_Parcial(Cota,Acomp),
  Vazao_Turbinada(Cota,Pg),
  Defluencia_Indicada(Cota,Qaf),

  info(volres,Volume),
  info(vlamlivre,V_LLivre),
  info(vparcial,Qp),
  info(vlamlivre,V_LLivre),
  info(vtur,Qtur),
  info(defind,Qdef),!,

  MakeWindow(5,7,7," Status Operativo da Usina ",12,12,9,55),nl,
  Writef(" Volume = %16.12g",Volume),nl,
  Writef(" Vazao Lamina Livre = %8.2f",V_LLivre),nl,
  Writef(" Vazao Parcial por Comporta = %8.2f",Qp," m3/s "),nl,
  Writef(" Vazao Turbinada = %8.2f",Qtur," m3/s"),nl,
  Writef(" Defluencia Indicada = %8.2f",Qdef," m3/s"),nl,
  readchar(_),
```

Removewindow, RemoveWindow.

```
/* ----- */
/* Modelo de Regras de Controle de Vazao */
/* ----- */
```

Ler\_Cota\_Vazao:-

```
Clearwindow, nl,
Write(" Entre Com a Vazao Afluente (m3/s) --> "),
ReadReal(Qaf),nl,
Write(" Entre Com a Cota Atual (m) --> "),
ReadReal(Cota),
AtualInfo(nar,Cota),
AtualInfo(qaf,Qaf).
```

Acha\_Status(Cota,Q,Mensagem) :-

```
Checa_Rest_JMF(Cota,Q,Mensagem),
Writef(" Cota = %7.2f (m) Vazao = %8.2f (m3/s)\n",Cota,Q),
Write(" Recomenda-se ",Mensagem).
```

```
/* Checa Qual o Estado Hidraulico Dados Cota e Vaz Af. */
```

```
Checa_Status(Cota,_):- Cota < 498.5, AtualStatus("Normal"),!.
Checa_Status(Cota,_):- Cota>=505,AtualStatus("Emergencia"),!.
Checa_Status(Cota,Qaf) :- Cota < 503.88, Qaf < 2.200,
    AtualStatus("Normal"),!.
Checa_Status(Cota,Qaf) :- Cota < 503.88, Qaf >= 2.200,
    Qaf< 12000,AtualStatus("Atencao"),!.
Checa_Status(Cota,Qaf) :- Cota < 505, Cota >= 503.88,
    Qaf < 12000,AtualStatus("Alerta"),!.
Checa_Status(Cota,Qaf) :- Cota < 503.88, Qaf >= 12000,
    Qaf < 16000,AtualStatus("Alerta"),!.
Checa_Status(Cota,Qaf) :- Cota < 501,Qaf >= 16000,
    Qaf < 25200,AtualStatus("Alerta"),!.
Checa_Status(Cota,Qaf) :- Cota >= 503.88, Qaf >= 12000,
    AtualStatus("Emergencia"),!.
Checa_Status(Cota,Qaf) :- Cota >= 501, Qaf >= 16000,
    AtualStatus("Emergencia"),!.
Checa_Status(_,_) :- Write(" Problemas na Verificacao"),nl,
    Write(" do Estado Hidrologico."),!.
```

Errestricao(A,B) :- Restricao(A,B).

Help\_Status :-

```
/* Asserta(nomeusina("UHSS")),*/
ClearWindow,
Menu(10,20,["Normal","Atencao","Alerta","Emergencia"],
    Estado),
Mostra(Estado).
```

```

/* Mostra(n) Processa Menu do Help_Status */
Mostra(1) :- Ft = ".Hno",AtualStatus("Normal"),
            Edit_Help_Status(Ft).
Mostra(2) :- Ft = ".Hat",AtualStatus("Atencao"),
            Edit_Help_Status(Ft).
Mostra(3) :- Ft = ".Hal",AtualStatus("Alerta"),
            Edit_Help_Status(Ft).
Mostra(4) :- Ft = ".Hem",AtualStatus("Emergencia"),
            Edit_Help_Status(Ft).

Edit_Help_Status(Ft) :- NomeUsina(NU),Concat(Nu,Ft,Arq),
                       Status(St),
                       Concat("Estado Hidrologico : ",St,Aviso),
                       MakeWindow(16,3,7,Aviso,0,0,24,80),
                       File_Str(Arq,StArq),
                       Display(Starq),
                       RemoveWindow.

MostraS("Normal")      :- Mostra(1).
MostraS("Atencao")     :- Mostra(2).
MostraS("Alerta")      :- Mostra(3).
MostraS("Emergencia") :- Mostra(4).

Checa_Rest_JMF(Cota,Q,Mensagem) :-
    Erestricao(qrJMF,QRJMF),
    Q > QRJMF,
    Quebra_Rest_JMF(Cota,Q,Mensagem).
Checa_Rest_JMF(Cota,_,Mensagem) :-
    Checa_VE_JMF(Cota,Mensagem).

/* Verifica se o Volume de Espera de UHJMF Esta Ocupado */
Checa_VE_JMF(Cota,Mensagem) :-
    Erestricao(cota_ve_JMF_Inf,CINF),
    Cota > CINF,
    Mensagem="Manter Vaz de Descarga(Qdef) = Vaz de Restricao de JMF".
Checa_VE_JMF(_,Mensagem) :-
    Mensagem="Manter o Volume de Espera de JMF".

Quebra_Rest_JMF(Cota,_,Mensagem) :-
    Erestricao(cota_ve_JMF_Sup,CSUP),
    Cota < CSUP,
    Checa_Nivel_Min(Cota,Mensagem).
Quebra_Rest_JMF(Cota,Q,Mensagem) :-
    Checa_Rest_UHSS(Cota,Q,Mensagem).

Checa_Nivel_Min(Cota,Mensagem) :-
    Erestricao(cota_ve_JMF_Inf,CINF),
    Cota < CINF,
    Checa_QN1(Mensagem).

```

```

Checa_Nivel_Min(, Mensagem) :-
    Mensagem = " Defluir o Max. Entre a Restricao de JMF e QN2".

Checa_QN1(Mensagem) :-
    Erestricao(qrJMF, QRJMF), Erestricao(qn1, QN1),
    QRJMF > QN1,
    Mensagem = " Max Defluencia seja Igual a QN1".
Checa_QN1(Mensagem) :-
    Mensagem = " Defluir o Max. Entre a Restricao de JMF e QN2".

/* Verifica se a Vazo Afluente Rompe a Restrio de UHSS */
Checa_Rest_UHSS(Cota, Q, Mensagem) :-
    Erestricao(qrUHSS, QRUHSS),
    Q > QRUHSS,
    Quebra_Rest_UHSS(Cota, Mensagem).
Checa_Rest_UHSS(Cota, , Mensagem) :-
    Checa_VE_UHSS(Cota, Mensagem).
/*Ativada se a Vazo Afluente for Maior que a Restrio de UHSS*/
Quebra_Rest_UHSS(Cota, Mensagem) :-
    Erestricao(cota_ve_UHSS_Sup, CSUP),
    Cota < CSUP,
    Mensagem=" Defluir Restricao de UHSS ".
Quebra_Rest_UHSS(, Mensagem) :-
    Mensagem=" Operar em Lamina Livre ".

/* Verifica se o Volume de Espera de UHSS Est Ocupado */
Checa_VE_UHSS(Cota, Mensagem) :-
    Erestricao(cota_ve_UHSS_Inf, CINF),
    Cota > CINF,
    Mensagem = " Recuperar o Volume de Espera de UHSS ".
Checa_VE_UHSS(, Mensagem) :-
    Mensagem = " Manter Volume de Espera de UHSS ".

/* ----- */
/****** Modelo de Abertura de Comportas *****/
/* ----- */

Abertura_Comporta(Lista_Comporta) :-
    Sequencia_Abertura(Lista_Comporta), !.
Abertura_Comporta([]) :-
    Write("Abertura da Comporta Falhou"), ReadChar(_).

/* Sequencia_Abertura Contem a Seq. Apropriada de Abertura das */
/* Comportas, Observando o Numero de Comportas Impedidas */

Sequencia_Abertura(Lista_Comporta) :-
    Comporta_Impedida(Comp1, Comp2),

```

```

        seq_Esp_Abert_Comp(Comp1,Comp2,Lista_Comporta).
Sequencia_Abertura(Lista_Comporta) :-
    Comporta_Impedida(Comp1,Comp2),
    seq_Esp_Abert_Comp(Comp2,Comp1,Lista_Comporta).

/* Fechamento_Comporta Obtem a partir da lista de Seq. de abertura */
/* das comportas a sequencia de fechamento que e a lista invertida */

Fechamento_Comporta(Lista_Comporta,Lista_Fecha):-
    Reverse_List(Lista_Comporta,Lista_Fecha).

/* Ler_Comp_Imp Le o Numero de Comportas Impedidas (0 , 1 , 2) */
/* e Limpa a Base de Dados para Nova Rodada */

Ler_Comp_Imp :-
    ClearWindow,
    Write(" Numero de Comportas Impedidas "),
    ReadInt(N),
    Clear_Comp_Imp,
    Assemble_Comp(N).

Clear_Comp_Imp :- Retract(comporta_Impedida(,_)),fail.
Clear_Comp_Imp .

/* Assemble Adiciona o fato Comporta Impedida na Base de Dados */
/* Ler os Dados necessarios para tal */
Assemble_Comp(0):-
    asserta( comporta_Impedida(0,0) ).

Assemble_Comp(1) :-
    ClearWindow,
    Write(" Numero da Comporta Impedida --> "),
    ReadInt(N1),
    Asserta(comporta_Impedida(N1,0)).

Assemble_Comp(2) :-
    ClearWindow,
    Write(" Numero da Primeira Comporta Impedida --> "),
    ReadInt(N1),n1,
    Write(" Numero da Segunda Comporta Impedida --> "),
    ReadInt(N2),
    Asserta(comporta_Impedida(N1,N2)).

/* ----- */
/* Implementacao do Modelo Matematico */
/* ----- */

/* Defind = Defluencia Indicada */

```

```

/* Volres = Volume do Reservatorio */
/* Vlamlivre = Vazao Lamina Livre */
/* Vparcial = Vaz Abert. Parcial */
/* Vtur = Vazao Turbinada */

/* Acq# = Coef. de Cq na Vaz. Abert. Parcial*/
/* Ag# = Coef. de GO na Vaz. Abert. Parcial*/
/* Ac# = Coef. de CCO na Vaz. Abert. Parcial*/
/* Acomp = Abertura da comporta em Metros */
/* GO = Min. Dist. entre a Comp. e o Verted.*/

/* Calcula o Volume de Agua no Reservatorio dado a Cota */
Calculo_Cota_Volume(Cota) :-
  Info(cva0,A0),Info(cva1,A1),Info(cva2,A2),
  Info(cva3,A3),Info(cva4,A4),
  X = Cota,
  V = A0 + A1*X + A2*X*X + A3*X*X*X + A4*X*X*X*X,
  AtualInfo(volres,V).

/* Calcula a Cota no Reservatorio dado o Volume */
Calculo_Vol_Cota(Vol) :-
  Info(ccv0,A0),Info(ccv1,A1),
  Info(ccv2,A2),Info(ccv3,A3),Info(ccv4,A4),
  X = Vol,
  Cota = A0 + A1*X + A2*X*X + A3*X*X*X + A4*X*X*X*X,
  Checa_Cota(Cota),
  AtualInfo(nar,Cota),!.

Calculo_Vol_Cota(_) :- Write(" Fim da Simulacao "),
  Prozesse(7).

/* Calcula a Vazao em Lamina Livre dado uma Certa Cota */
Vazao_Lamina_Livre(Cota) :-
  Info(cota_crista,CC), Cota >= CC ,
  Info(lcomp,L), Info(cq,Cq),
/* H = Col. d'agua Acima do Centro da Abert. Efetiva(m) */
  H = Cota - CC,
  Q = L * Cq * Sqrt(H) * H,
  AtualInfo(vlamlivre,Q).

/* Nao Ha Defluencia */
Vazao_Lamina_Livre(_) :-
  AtualInfo(vlamlivre,0),!.

/* Vazao Abertura Parcial */
Vazao_Parcial(_,Acomp) :-
  Abs(Acomp) <= 0.1,AtualInfo(vparcial,0),!.

```

```

Vazao_Parcial(Cota,Acomp) :-
  Info(cota_crista,CC), Cota >= CC, Z = CC + Acomp,
  /* Carregamento dos Coef. p/ as Eqs. de Cq,GO CCO */
  Info(acq0,Acq0),Info(acq1,Acq1),Info(acq2,Acq2),
  Info(ag0,Ag0) ,Info(ag1,Ag1) ,Info(ag2,Ag2) ,
  Info(ac0,Ac0) ,Info(ac1,Ac1) ,Info(ac2,Ac2) ,
  /* Calculo dos valores de Cq, GO, CCO */
  Poli(Acq0,Acq1,Acq2,Z,Cq) ,
  Poli(Ag0,Ag1,Ag2,Acomp,GO) ,
  Poli(Ac0,Ac1,Ac2,Acomp,CCO),
  Cota >= CCO,
  H = Cota - CCO,
  Info("lcomp",Lcomp),
  Q = Cq * Lcomp * GO * Sqrt(19.62 * H),
  AtualInfo(vparcial,Q).

```

```

Vazao_Parcial(,_) :-
  AtualInfo(vparcial,0),!.

```

```

/* Calculo da Variacao do Volume */
Calculo_Var_Vol(Np,Vaf,Deltav) :-
  DadoTemp(NpAnt,VafAnt,VVert,Vturb,_),
  Deltas = (Np - NpAnt) * 3600,
  Deltav = ( (Vaf + VafAnt)/2 - Vvert - Vturb ) * Deltas.

```

```

/* Modelo para Calculo de Vazao Turbinada */
/* Pg = Potencia Gerada (Eletrica) */
/* Vazao_Turbinada Calcula a Vazao que passa por uma Turbina */
/* Dados Hb, Pt */

```

```

/* Leitura da Potencia Gerada em Cada Turbina */

```

```

Leia_Dados_Turb :- MakeWindow(17,7,7,
  " Entre Com as Potencias Lidas das Turbinas ",10,0,8,80)
  Info(nar,Cota),
  ListaTurb(Lnt),
  Ler_Ger_Turb(LNt,ListaTurb,Cota,0),
  DadoTemp(Periodo,_,_,_,_),
  AtualTurb(Periodo,ListaTurb),
  RemoveWindow,!.

```

```

Ler_Ger_Turb([],[],_,Vtt) :- AtualInfo(vtotturb,Vtt).

```

```

Ler_Ger_Turb([Nt:T], [d(Nt,Gt,Vt):Resto],Cota,Vtt) :-
  Repeat,
  Cursor(Nt,20),
  Write(" Turbina No. ",Nt," --> "),
  ReadReal(G),
  Checa_Pot_Max(G),
  Gt = G,

```

```

Vazao_Turbinada(Cota,Gt),
Info(vtur,Vt),
Vtt2 = Vtt + Vt,
Ler_Ger_Turb(T,Resto,Cota,Vtt2),!.
Ler_Ger_Turb(,[,_,_) :-
Write(" Problema na Leitura de Dados da Turbina."),nl,
Write(" Pressione Qualquer Tecla."),
ReadChar(_).

/* Calculo da Vazao Turbinada Dados Cota e Geracao Eletrica */
Vazao_Turbinada(Cota,Pg) :-
    Difq = 1000.0,
    Info("rendelettrico",Re),Info("vtur",Qt),
    Info("na_jus",NMJ),
    Hb = Cota - NMJ,
    Pt = Pg / Re,
    Lvt(Hb,Pt,Difq,Qt).

/* Laco da Vazao Turbinada */
Lvt(,_,Difqx,Qt) :- Difqx <= 0.01,
    AtualInfo("vtur",Qt),!.

Lvt(Hb,Pt,_,Qt) :-
    H1 = Hb - 1.5043E-05 * Qt * Qt,
    Rend(H1,Nt,Pt),
    Qtf = (10197 * Pt)/(H1 * Nt),
    Difq = Abs(Qt - Qtf),
    Lvt(Hb,Pt,Difq,Qt).

/* Calculo do Rend. da Turb. Dados Queda Liquida e Pot. Mec. */
Rend( H1, Nt, Pt) :-
/* rt1# e rt2# = Coef. dos Rend. na Min e Max. Quedas */
Info(rt10,Rt10),Info(rt11,Rt11),Info(rt12,Rt12),
Info(rt20,Rt20),Info(rt21,Rt21),Info(rt22,Rt22),
/* T1 e T2 = Rendimentos na Min. e Max. Quedas */
Poli(Rt10,Rt11,Rt12,Pt,T1),
Poli(Rt20,Rt21,Rt22,Pt,T2),
Info(max_hb,Hmax),Info(min_hb,Hmin),
Nt = (H1 - Hmin) * ( T2 - T1 )/(Hmax - Hmin) + T1.

/* Defluencia Indicada Dado Cota e Vazao Afluente */
Defluencia_Indicada(Cota,Qaf) :-
Info(ad00,Ad00),Info(ad01,Ad01),Info(ad02,Ad02),
Info(ad10,Ad10),Info(ad11,Ad11),Info(ad12,Ad12),
Info(ad20,Ad20),Info(ad21,Ad21),Info(ad22,Ad22),
Poli(Ad00,Ad01,Ad02,Qaf,A0),
Poli(Ad10,Ad11,Ad12,Qaf,A1),
Poli(Ad20,Ad21,Ad22,Qaf,A2),
Poli(A0,A1,A2,Cota,Vd),

```

```

ChecaVd(Cota,Qaf,Vd).

ChecaVd(Cota,_,_) :-
    Info(cota_crista,CC),
    Cota <= CC,
    AtualInfo(defind,0),!.

ChecaVd(Cota,Qaf,_) :-
    Info(cz0,Cz0), Info(cz1,Cz1), Info(cz2,Cz2),
    Poli(Cz0,Cz1,Cz2,Qaf,Cza),
    Cota < Cza,
    AtualInfo(defind,0), !.

ChecaVd(,_,Vd) :- AtualInfo(defind,Vd).

/* Calculo da Abertura Apropriada das Comportas */

/* ConfigVert = Database p/ Atualizacao das Comp*/
/* Vertedouro = Database da Configuracao de Comp*/
/* Vazvert = Vazao Instantanea pelo Vertedouro */
/* Dir = 0 --> S1 > Vreq e Lista Nao Invertida */
/*      = 1 --> S1 < Vreq e Lista      Invertida */
/* Dcomp = Obj. Composto = (No.Comp.,Abert.,Vaz)*/

Abert_Aprox(Ab,0)      :- Ab = 0,!.
Abert_Aprox(Ab,Vreq) :- Vreqc = Vreq/8,
                        T_Int(Z,Vreqc),
                        Ab = Z.

T_Int(Z,Vreqc):- Info("ap0",Ap0), Info("ap1",Ap1),
                 Z = Ap0 + Ap1 * Vreqc.

/*****/
Depura :- Sequencia_Abertura(Lista0),nl,
           Write("Vaz Req "),Readreal(Vaz),nl,
           AtualInfo(vreq,Vaz),
           Write("Cota "),ReadReal(Ct),
           Abertura_Adequada(Vaz,Ct,Lista0),
           Status_Comp(2).
/*****/

Abertura_Adequada(Vreq,Cota,Lista0) :-
    Calc_Ab(Lista0,Lista1,Cota,Vreq),
    Calcula_Valor_Vaz_Vert(Lista1,0),
    AtualConf(Lista1),
    AtualInfo(nar,Cota),
    Testa_Conv(Vreq),
    ListaFinal,!.

```

```

/* Le Dados de Abertura de Comportas e Calcula o Valor da Vazao */

Renovar_Dados_Ab_Comp :-
  MakeWindow(18,7,7,
    " Entre Com as Aberturas das Comportas ",10,0,12,80),
  Sequencia_Abertura(Lista0),nl,
  Info(nar,Cota),
  Ler_Abertura_Comporta(Lista0,Listal,Cota),
  AtualConf(Listal),
  Calcula_Valor_Vaz_Vert(Listal,0),
  RemoveWindow.

/* Le Abertura Comporta Dado Uma Seq.(Lista de Inteiros) */
/* Obtem a Vazao e Abertura e Deve Ser Dado a Cota */
Ler_Abertura_Comporta([],[],_).
Ler_Abertura_Comporta([Nc:Te],[d(Nc,Ab,Vp):Ts],Cota) :-
  Repeat,
/*   Cursor(Nc,20),*/
  Write("      Comporta No. ",Nc," --> "),
  ReadReal(Ab),
  Checa_Abertura_Max(Ab),
  Vazao_Parcial(Cota,Ab),
  Info(vparcial,Vp),
  Ler_Abertura_Comporta(Te,Ts,Cota).
Ler_Abertura_Comporta(Lista0,Listal,Cota):-
  Ler_Abertura_Comporta(Lista0,Listal,Cota).

Calc_Ab([],[],_,_).
Calc_Ab([Nc:Te],[d(Nc,Ab,Vp):Ts],Cota,Vreq) :-
  Abert_Aprox(Ab,Vreq),
  Vazao_Parcial(Cota,Ab),
  Info(vparcial,Vp),
  Calc_Ab(Te,Ts,Cota,Vreq).

/* Calcula o Valor Total de Vazao Vertida Dado as Aberts. das Comportas*/
Calcula_Valor_Vaz_Vert([],S1) :- AtualInfo(vazvert,S1).
Calcula_Valor_Vaz_Vert([d(_,_,Vp):T],S1) :-
  S2 = S1 + Vp,
  Calcula_Valor_Vaz_Vert(T,S2).

/* Testa_Conv Verifica se Ja Existe uma Solucao */
Testa_Conv(Vreq) :- info(vazvert,S1),
  Abs(Vreq - S1) < 50,!.

Testa_Conv(Vreq) :- info(vazvert,S1),Direcao(Dir),
  Checa_Dir(Vreq,S1,Dir),
  Recalculo(Vreq),
  Testa_Conv(Vreq).

```

```

/* Recalculo Procura Uma Nova Abertura */
Recalculo(Vreq) :- Trace(on),
    ConfigVert(Lista1), Trace(off),
    Info(nar,Cota), Info(vazvert,S1),
    Calcula_Novas_Aberturas(Lista1,Lista2,Vreq,S1,Cota),!,
    Atualconf(Lista2).

/* Busca de uma Abertura Adequada p/ Defluir Vreq */
Calcula_Novas_Aberturas([],[],_,S2,_):-
    AtualInfo(vazvert,S2).

Calcula_Novas_Aberturas(T,T,Vreq,S1,_):-
    Abs(S1 - Vreq) <= 50, AtualInfo(vazvert,S1).

Calcula_Novas_Aberturas([d(NoC,Ab,Q):T],[H2:T2],Vreq,S1,Cota):-
    S1 <= Vreq,
    Info(deltab,Deltab),
    Ab2 = Ab + Deltab,
    Vazao_Parcial(Cota,Ab2),
    Info(vparcial,Vp),
    H2 = d(NoC,Ab2,Vp),
    S2 = S1 - Q + Vp,
    Calcula_Novas_Aberturas(T,T2,Vreq,S2,Cota).

Calcula_Novas_Aberturas([d(NoC,Ab,Q):T],[H2:T2],Vreq,S1,Cota) :-
    S1 >= Vreq,
    Info(deltab,Deltab),
    Ab2 = Ab - Deltab,
    Vazao_Parcial(Cota,Ab2),
    Info(vparcial,Vp),
    H2 = d(NoC,Ab2,Vp),
    S2 = S1 - Q + Vp,
    Calcula_Novas_Aberturas(T,T2,Vreq,S2,Cota).

Calcula_Novas_Aberturas(_,L,_,_,_) :- L=[],
    Write(" Falhou Calcula_Novas_Aberturas "),!.

/* Checa Ordem da Abertura ou Fechamento das Comportas */
/* Dir = 0 => Abertura / Dir = 1 => Fechamento */
Checa_Dir(Vreq,S1,Dir) :- Dir = 0, S1 < Vreq,!.
Checa_Dir(Vreq,S1,Dir) :- Dir = 1, S1 > Vreq,!.
Checa_Dir(Vreq,S1,Dir) :- Dir = 1, S1 < Vreq,
    AtualDir(0),
    ConfigVert(Lista1),
    Reverse_List(Lista1,Lista2),
    AtualConf(Lista2).

```

```

Checa_Dir(Vreq,S1,Dir) :-
    Dir = 0, S1 > Vreq,
    AtualDir(1),
    ConfigVert(Lista1),
    Reverse_List(Lista1,Lista2),
    AtualConf(Lista2).

/* ListaFinal Realiza Armazenamento da Informacao Final */
ListaFinal :- Direcao(Dir), Dir = 0,
    ConfigVert(Lista1),
    AtualVert(Lista1).
ListaFinal :- Direcao(Dir), Dir = 1,
    ConfigVert(Lista1),
    Reverse_List(Lista1,Lista2),
    AtualConf(Lista2),
    AtualDir(0).

/* Imprime Lista de Comporta com (#,Abertura,Vazao) */
ImpListComp([]).
ImpListComp([d(Nc,Abc,Vc)!Tc]) :-
    Writef("\n          %2 %12.2g          %7.2g",Nc,Abc,Vc),
    ImpListComp(Tc).

/* Imprime Lista de Comporta "in a Fancy Way" */
/* Status_Comp(1) --> Resultados Enviados para a Tela */
/* Status_Comp(2) --> Resultados Enviados para a Impressora */

Status_Comp(1) :-
    MakeWindow(6,7,7," Status das Comportas ",6,12,12,56),
    nl,Write("          No. Comp      Abertura(m)      Vazao (m3/s) "),
    ConfigVert(Lista_Comp),!,
    ImpListComp(Lista_Comp),
    ReadChar(_),
    RemoveWindow.

Status_Comp(2) :-
    MakeWindow(6,7,7," Status das Comportas ",6,12,12,56),
    Writedevice(printer),
    Info(nar,Cota),
    nl,Write("-----"),
    nl,Write("          Configuracao das Comportas  Nar = ",Cota,"m"),
    nl,Write("-----"),
    nl,Write("          No. Comp      Abertura(m)      Vazao (m3/s) "),
    ConfigVert(Lista_Comp),!,
    ImpListComp(Lista_Comp),
    nl,Write("-----"),
    Info(vreq,Vreq),Info(vazvert,VazVert),
    nl,Writef("Vaz. Req. = %6.0f ; Vaz. Obtida = %6.0f (m3/s)",Vreq,VazVert),
    nl,Write("-----"),

```

```
Flush(printer),
Writedevic(screen),
ReadChar(_),
RemoveWindow.
```

```
/* ----- */
/* Clausulas para o Treinamento de Onda de Cheia */
/* ----- */
```

```
Inicio_Onda :-
  MakeWindow(1,7,7," Treinamento de Uma Cheia ",0,0,24,80),

  Repeat,
  Cursor(3,5),
  Write(" Entre com o Nivel Presente do Reservatorio --> "),
  ReadReal(Cota),
  Checa_Cota(Cota),
  AtualInfo(nar,Cota),!,

  Repeat,
  Cursor(5,5),
  Write(" Entre com o Nivel a Jusante do Reservatorio --> "),
  ReadReal(NJ),
  Checa_Jus_Val(Cota,NJ),
  AtualInfo(na_jus,NJ),nl,!,

  AtualFlag(0),
  Leia_Dados_Turb,nl,
  Renovar_Dados_Ab_Comp,
  Dado_prox_Periodo(Npi,Nqafi),
  Checa_Status(Cota,Nqafi),
  Imprimir_Tabela(Npi,Nqafi,Cota).
```

```
Loop_Onda :- Dado_Prox_Periodo(Np,Nqaf),
  Not(Np = -1),
  Calculo_Var_Vol(Np,Nqaf,Deltav),
  Info(nar,Cota),
  Calculo_Cota_Volume(Cota),
  Info(volres,V1),
  Volume = V1 + Deltav,
  Calculo_Vol_Cota(Volume),
  Info(nar,Cota2),
  Checa_Status(Cota2,Nqaf),
  Imprimir_Tabela(Np,Nqaf,Cota2).
```

```
/* Obtencao do Dado do Proximo Período */
/* Np = No. do Período */
/* Nqaf = Vazao Afluente no Período */
/* Np = -1 Indica Fim da Lista de Onda */
```

```

Dado_prox_Periodo(Np,Nqaf) :- Onda([b(Np,Nqaf);T]),
                               AtualOnda(T), AtualInfo(qaf,Nqaf),!.
Dado_Prox_Periodo(Np,Nqaf) :- Np = -1, Nqaf = -1,AtualFlag(Np).

/* Imprime a Tabela de Resultados da Operacao de Cheia */

Imprimir_Tabela(Np,Nqaf,Cota) :-
    InfoI(contimp,M),
    M = 0,
    ClearWindow,
    Write(" Período Vaz.Aflu. Vaz.Vert. Vaz.Turb. Cota"),nl,
    Write(" (Hora) (m3/s) (m3/s) (m3/s) (m)"),
    Execute_Impressao(Np,Nqaf,Cota,M),!.

Imprimir_Tabela(Np,Nqaf,Cota) :-
    InfoI(contimp,M),
    Execute_Impressao(Np,Nqaf,Cota,M),!.

Execute_Impressao(Np,Nqaf,Cota,M) :-
    Ajuste_Contador(M),
    Info(vazvert,VV), Info(vtotturb,Qt),
/* DadoTemp(,_,VV,Qt,_,_) */
    M3 = M + 3,
    Cursor(M3,2) ,Writef("%3",Np),
    Cursor(M3,9),Writef("%8.2f",Nqaf),
    Cursor(M3,21),Writef("%8.2f",VV),
    Cursor(M3,32),Writef("%8.2f",Qt),
    Cursor(M3,42),Writef("%7.2f",Cota),

    MakeWindow(12,7,7,"",20,5,3,45),
    Repeat,
    Cursor(0,0),
    Write(" Entre Com a Vazao a Defluir --> "),
    ReadReal(Vadef1),
    Checa_Def_Max(Vadef1),
    Checa_Val_Pos(Vadef1),
    RemoveWindow,
    AtualInfo(vazvert,Vadef1),
    AtualDadoTemp(Np,Nqaf,Vadef1,Qt,Cota),
    WriteDevice(ondaxy),
    Write(Np," ",Nqaf," ",Vadef1),nl,
    WriteDevice(screen),!,
Repeat,
    Menu(10,20,["Nao Preciso Ajuda",
                "Vazao Recomendada",
                "Abertura Requerida",
                "Dados Geraao","Abertura Comporta",
                "Sugestoes","Encerrar"],Escolha),

```

Processi(Escolha).

Processi(1):-!.

Processi(2) :-

```
    Info(nar,Cota),Info(qaf,Qaf),
    MakeWindow(14,7,7," Janela de Auxilio ",
               10,10,10,55),
    Defluencia_Indicada(Cota,Qaf),
    Info(defind,Deflu),
    Cursor(3,3),
    Writef("Para Uma Afluencia de %8.2f (m3/s)",Qaf),
    Cursor(4,3),
    Write("e Uma Cota de ",Cota," (m)"),
    Cursor(5,3),
    Writef("Defluir no MINIMO %8.2f (m3/s).",Deflu),
    ReadChar(_),
    RemoveWindow,!.
```

Processi(3) :- Info(nar,Cota),Info(qaf,Qaf),  
Menu(10,10,  
["Abrir Comporta P/ Defluencia Dada",  
"Use o Diag. de Op. Normal e Emergencia"],  
Selec),  
ProcAb(Selec,Cota,Qaf,Deflu),  
Sequencia\_Abertura(Lista0),  
Abertura\_Adequada(Deflu,Cota,Lista0),  
Status\_Comp(1),!.

Processi(4) :- Leia\_Dados\_Turb.

Processi(5) :- Renovar\_Dados\_Ab\_Comp.

Processi(6) :- Menu(10,25,["Estado Hidrologico",  
"Grafico Evento","Recomendacoes"],Opcao),  
Proc(Opcao).

Processi(7) :- ClearWindow,Objetivo(1).

ProcAb(1,\_,\_,Deflu) :- Info(vazvert,Deflu).

ProcAb(2,Cota,Qaf,Deflu) :-Defluencia\_Indicada(Cota,Qaf),  
Info(defind,Deflu).

Proc(1) :- Status(St),MostraS(St),!.

Proc(2) :- Processe(3),!.

Proc(3) :- Info(nar,Cota),  
Info(qaf,Qaf),  
MakeWindow(14,7,7," Janela de Auxilio ",  
10,10,10,55),

```

Checa_Rest_JMF(Cota,Qaf,Mensagem),
Cursor(3,5),
Writef("Cota = %7.2 (m)",Cota),
Cursor(5,5),
Writef("Vazao Afluente = %8.2 (m3/s)",Qaf),
Cursor(7,5),
Write("Recomenda-se ",Mensagem),
ReadChar(_),
RemoveWindow.

```

```

Ajuste_Contador(15) :- AtualInfoI(contimp,0).
Ajuste_Contador(M) :- M1 = M + 1, AtualInfoI(contimp,M1).

```

```

/* ----- */
/* Rotinas de Suporte */
/* ----- */

```

```

/* ImpResult Imprime um Lista */
ImpResult([]).
ImpResult([Head:Tail]) if
    write(Head," "),
    ImpResult(Tail).

```

```

AtualFlag(_) :- Retract(flag(_)),Fail.
AtualFlag(Integer) :- Asserta(flag(Integer)).

```

```

index([X:_,_],1,X):-!.
index([_:L],N,X):-N>1,N1=N-1,index(L,N1,X).

```

```

Member(X,[X:_,_]).
Member(X,[_:L]) :- Member(X,L).

```

```

unik([],[]).
unik([H:T],L):-member(H,T),!,unik(T,L).
unik([H:T],[H:L]):-unik(T,L).

```

```

append([],L,L).
append([A:At],B,[A:C]):-append(At,B,C).

```

```

AtualDir(_) :- Retract(direcao(_)),Fail.
AtualDir(Int) :- Asserta(direcao(Int)).

```

```

AtualInfo(Symbol,_) :- Retract(info(Symbol,_)),Fail.
AtualInfo(Symbol,Real) :- Asserta(info(Symbol,Real)).

```

```

AtualInfoI(Symbol,_) :- Retract(foi(Symbol,_)),Fail.
AtualInfoI(Symbol,Real) :- Asserta(foi(Symbol,Real)).

```

```

AtualConf(_)      :- Retract(configvert(_)),Fail.
AtualConf(Lista) :- Asserta(configvert(Lista)).

AtualStatus(_)   :- Retract(status(_)),Fail.
AtualStatus(St) :- Asserta(status(St)).

AtualVert(Lista) :- Retract(vertedouro(Lista)),Fail.
AtualVert(Lista) :- Asserta(vertedouro(Lista)).

AtualOnda(_)      :- Retract(onda(_)),Fail.
AtualOnda(Lista) :- Asserta(onda(Lista)).

AtualTurb(Int,_) :- Retract(turb(Int,_)),Fail.
AtualTurb(Int,List) :- Asserta(turb(Int,List)).

AtualDadoTemp(_____) :- Retract(dadotemp(_____,_____,_____,_____,_____),Fail.
AtualDadoTemp(Int,R1,R2,R3,R4) :-
    Asserta(dadotemp(Int,R1,R2,R3,R4)).

Checa_Cota(Nivel) :- Checa_Cota_Max(Nivel),Checa_Cota_Min(Nivel).

Checa_Cota_Max(Nivel) :- Info(na_max_maxi,NMM),
    Nivel < NMM,
    Nivel >= 0,!.

Checa_Cota_Max(Nivel) :-
    Checa_Val_Pos(Nivel),
    MakeWindow(11,7,7," Cuidado !!! ",5,0,10,80),
    Cursor(2,5),
    Write(" Voce Perdeu a Barragem !!! "),
    Cursor(3,5),Info(na_max_maxi,NMM),
    Write(" O Nivel Maximo Permissivel eh ",NMM," (m) "),
    Cursor(4,5),
    Write(" O Nivel por Voce Fornecido Foi de ",Nivel," (m)"),
    ReadChar(_),
    RemoveWindow,
    Fail.

Checa_Cota_Min(Nivel) :- Info(nar_min,CMin), Nivel >= CMin,!.
Checa_Cota_Min(Nivel) :- Info(nar_min,Cmin),
    MakeWindow(11,7,7," Cuidado !!! ",5,0,10,80),
    Cursor(2,5),
    Write(" Barragem Secou (Baita Seca) !!! "),
    Cursor(3,5),
    Write(" O Nivel Minimo Permissivel eh ",Cmin," (m) "),
    Cursor(4,5),
    Write(" O Nivel por Voce Fornecido Foi de ",Nivel," (m)"),
    ReadChar(_),
    RemoveWindow,
    Fail.

```

```
Checa_Abertura_Max(Abertura) :-  
    Info(max_ab_comp,NMM),  
    Abertura <= NMM,  
    Abertura > -0.01,!.  
.
```

```
Checa_Abertura_Max(Abertura) :-  
    Checa_Val_Pos(Abertura),  
    MakeWindow(11,7,7," Cuidado !!! ",5,0,10,80),  
    Cursor(2,5),  
    Write(" Voce Destruiu a Comporta !!! "),  
    Cursor(3,5),Info(max_ab_comp,NMM),  
    Write(" A Abertura Maxima Permissivel eh ",NMM," (m) "),  
    Cursor(4,5),  
    Write(" A Abertura por Voce Obtida Foi de ",Abertura," (m)"),  
    ReadChar(_),  
    RemoveWindow,  
    Fail.  
.
```

```
Checa_Pot_Max(Pot) :-  
    Info(max_pot,NMM),  
    Pot < NMM,  
    Pot > -0.01,!.  
.
```

```
Checa_Pot_Max(Pot) :-  
    Checa_Val_Pos(Pot),  
    MakeWindow(11,7,7," Cuidado !!! ",5,0,10,80),  
    Cursor(2,5),  
    Write(" A Turbina Deve Estar Torrando o Gerador !!! "),  
    Cursor(3,5),Info(max_pot,NMM),  
    Write(" A Potencia Maxima Permissivel eh ",NMM," (MWe) "),  
    Cursor(4,5),  
    Write(" A Potencia por Voce Lida Foi de ",Pot," (MWe)"),  
    ReadChar(_),  
    RemoveWindow,  
    Fail.  
.
```

```
Checa_Def_Max(Vdef) :-  
    Info(max_vaz_def,NMM),  
    Vdef < NMM, Vdef > -0.01,!.  
.
```

```
Checa_Def_Max(Vdef) :-  
    Checa_Val_Pos(Vdef),  
    MakeWindow(11,7,7," Cuidado !!! ",5,0,10,80),  
    Cursor(2,5),  
    Write(" Sai Correndo, La Vem Agua na Casa de Forca !!! "),  
    Cursor(3,5),Info(max_vaz_def,NMM),  
    Write(" A Vazao Maxima Permissivel eh ",NMM," (m3/s) "),  
    Cursor(4,5),  
    Write(" A Vazao de Defluencia por Voce Seleccionada Foi de ",  
        Vdef," (m3/s)"),  
    Fail.  
.
```

```
ReadChar(_),
RemoveWindow,
Fail.
```

```
Checa_Jus(Cota,NJ) :- Checa_Jus_Val(Cota,NJ),Checa_Jus_Min(Nj).
```

```
Checa_Jus_Val(Cota,NJ) :- NJ < Cota, NJ > 0,!. 
```

```
Checa_Jus_Val(Cota,NJ) :-
    Checa_Val_Pos(NJ),
    MakeWindow(11,7,7," Cuidado !!! ",5,0,10,80),
    Cursor(2,5),
    Write(" Nivel de Jusante Esta Maior que o de Montante !!! "),
    Cursor(3,5),
    Write(" O Nivel do Reservatorio Fornecido eh ",Cota," (m) "),
    Cursor(4,5),
    Write(" O Nivel de Jusante Fornecido eh      ",NJ," (m)"),
    ReadChar(_),
    RemoveWindow,
    Fail.
```

```
Checa_Jus_Min(Nj) :- Info(nar_jus_min,Nmin), Nj >= Nmin,!. 
```

```
Checa_Jus_Min(Nj) :- Info(nar_jus_min,Nmin),
MakeWindow(11,7,7," Cuidado !!! ",5,0,10,80),
    Cursor(2,5),
    Write(" Barragem a Jusante Secou !!! "),
    Cursor(3,5),
    Write(" O Nivel Minimo Permissivel eh ",Nmin," (m) "),
    Cursor(4,5),
    Write(" O Nivel por Voce Fornecido Foi de ",Nj," (m)"),
    ReadChar(_),
    RemoveWindow,
    Fail.
```

```
/* Verifica Ocorrencia de Valores Negativos */
```

```
Checa_Val_Pos(N) :- N >= 0, !.
```

```
Checa_Val_Pos(_) :- MakeWindow(11,7,7," Cuidado !!! ",
                                5,0,10,80),
    Cursor(2,5),
    Write(" Voce Entrou Com Valor Negativo ! "),
    Cursor(4,5),
    Write(" Favor Corrigi-lo "),
    ReadChar(_),
    RemoveWindow,
    Fail.
```

```
Escolha_Arquivo_BD :-
```

```
    MakeWindow(10,7,7," Escolha Arquivo com Sigla da Usina ",
```

```

        10,10,10,60),
Dir("//", "*.DNU", Dat),
Consult(Dat),
RemoveWindow.
Escolha_Arquivo_BD :-
Write(" Problema no Arquivo que Contem a Base de Dados."),nl,
Write(" Pressione Qualquer Tecla."),
ReadChar(_),
RemoveWindow.

Escolha_Arquivo_Onda :-
MakeWindow(10,7,7," Escolha Arquivo com Onda de Cheia ",
        10,10,10,60),
Dir("//", "Onda*.Dad", Dat),
Consult(Dat),
RemoveWindow.

Escolha_Arquivo_Onda :-
Write(" Problema no Arquivo que Contem a Onda de Cheia."),nl,
Write(" Pressione Qualquer Tecla."),
ReadChar(_),
RemoveWindow.

Escolha_Arquivo_Qualquer :-
MakeWindow(13,7,7," Escolha o Arquivo e Tecle Enter ",
        0,0,24,80),
Dir("//", "*.x", Dat),
File_Str(Dat, Sdat),
Edit(SDat, NovoSDat),
Repeat,
ClearWindow,
Write(" Salvar Arquivo ", Dat, " (S/N) ? "),
ReadChar(Ans),
Gravar(Ans, NovoSDat, Dat),!,
RemoveWindow.

Escolha_Arquivo_Qualquer :-
Write(" Problema no Arquivo Selecionado."),nl,
Write(" Pressione Qualquer Tecla."),
ReadChar(_),
RemoveWindow.

Gravar(Ans, Ns, Dat) :- Ans = 's', Execute_Gravacao(Ns, Dat).
Gravar(Ans, Ns, Dat) :- Ans = 'S', Execute_Gravacao(Ns, Dat).
Gravar(Ans, _, _) :- Ans = 'n'; Ans = 'N'.

Execute_Gravacao(Ns, Dat) :-
OpenWrite(extra, Dat),
WriteDevice(extra),

```

```
Write(Ns),
CloseFile(extra),!.
Execute_Gravacao(_,Dat) :- Write(" Nao foi Possivel Gravar :"),
Write(Dat,"."),nl,
Write(" Pressione Qualquer Tecla."),
ReadChar(_).
```

```
/* Poli Calcula o Valor do Polinomio de Grau 2 Dados Coef. e X */
Poli(A0,A1,A2,X,R) :- R = A0 + A1*X + A2*X*X.
```

```
/* Reverse_List Inverte a Sequencia de Uma Lista */
```

```
Reverse_List(Lista,Inversa) :-
Reverse([],Lista,Inversa).
```

```
Reverse(Inversa,[],Inversa).
```

```
Reverse(Lista1,[Head:Tail],Inversa) :-
Reverse([Head:Lista1],Tail,Inversa).
```

```
Repeat.
```

```
Repeat :- Repeat.
```

```
RepeatEsp :- Flag(N),Not(N = -1).
```

```
RepeatEsp :- Flag(N),Not(N = -1),RepeatEsp.
```