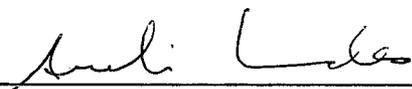


METODOLOGIA DE DESENVOLVIMENTO DE UM CONFIGURADOR DE TESTES
AUTOMÁTICOS COM TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

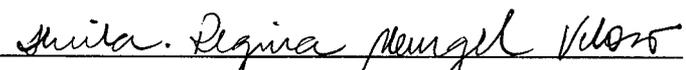
Mauricio Gonzalo Solar Fuentes

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

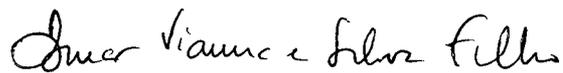
Aprovada por:



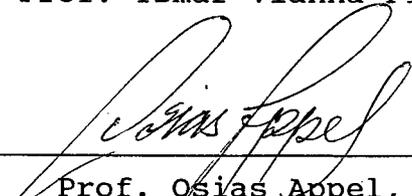
Prof^a. Sueli B. Teixeira Mendes, Ph.D.
(Presidente)



Prof^a. Sheila R. Murgel Veloso, D.Sc.



Prof. Ysmar Vianna Filho, Ph.D.



Prof. Osias Appel, M.Sc.

SOLAR F., MAURICIO G.

Metodologia de Desenvolvimento de um
Configurador de Testes Automáticos com
Técnicas de Inteligência Artificial
[Rio de Janeiro] 1989

XII, 153 p. 29,7 cm (COPPE/UFRJ, M.Sc.,
Engenharia em Sistemas e Computação,
1989)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1. Inteligência Artificial. I.
COPPE/UFRJ. II. Título (série).

a MARIELA

a la memória de mi TATA

AGRADECIMENTOS

Ao PROGRAMA DE ENGENHARIA DE SISTEMAS E COMPUTAÇÃO, à COPPE, à UFRJ e ao POVO DO BRASIL pela possibilidade de realizar a pós-graduação.

Desejo expressar meus mais sinceros agradecimentos a SUELI MENDES, pela sua confiança, apoio e orientação acadêmica.

Aos meus professores da COPPE, que me ensinaram o caminho da pesquisa.

Ao CEPEL, em especial OSIAS APPEL pelo apoio e oportunidade de me permitir desenvolver um trabalho de pesquisa na área do meu interesse.

Aos meus amigos da COPPE pelas convivências e amizade que é indispensável no estudo.

Ao pessoal do laboratório do CEPEL pelo excelente ambiente de trabalho para o desenvolvimento desta tese.

Em especial, a MARIA FERNANDA e CLÁUDIA pela sua ajuda e esforçada compilação do "portunhol" ao português na elaboração do texto.

Em geral, a todas as pessoas que me apoiaram e ajudaram desinteressadamente.

Ao Word 4.0 da MicroSoft, à impressora LaserJet serie II da Hewlett Packard e ao Flow Chart 2.42 que permitiram a edição desta tese.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências (M.Sc.).

METODOLOGIA DE DESENVOLVIMENTO DE UM CONFIGURADOR DE TESTES
AUTOMÁTICOS COM TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Mauricio Gonzalo Solar Fuentes

Agosto de 1989

Orientador: Ph.D. Sueli B. T. Mendes

Programa: Programa de Engenharia de Sistemas e Computação

Esta tese apresenta uma Metodologia de desenvolvimento aplicando técnicas de Inteligência Artificial para um sistema em configuração de processos em tempo real. Esta Metodologia propõe o uso de quadros para representar o conhecimento envolvido neste tipo de processos, para o que foi desenvolvida uma ferramenta para editar quadros, facilitando desta forma a tarefa de estruturar o conhecimento.

A Metodologia proposta foi implementada em um projeto do Centro de Pesquisas de Energia Elétrica (CEPEL), que objetiva a automação de Ensaios Climáticos. O primeiro protótipo desenvolvido em Prolog em um microcomputador compatível com o PC/XT da IBM, viabilizou o sistema de automação como uma solução econômica.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M.Sc.).

DEVELOPMENT METHODOLOGY OF A CONFIGURATOR OF AUTOMATIC TESTS USING ARTIFICIAL INTELLIGENCE TECHNIQUES

Mauricio Gonzalo Solar Fuentes

August of 1989

Thesis Supervisor: Ph.D. Sueli B. T. Mendes

Department: Programa de Engenharia de Sistemas e Computação

This thesis presents a development methodology using artificial intelligence techniques for a configuration system of real time processes. This methodology proposes the use of frames for the knowledge representation involved in this kind of processes, whereby it presents a tool for frames edition, that facilitates in this way the knowledge structure task.

The proposed methodology was implemented for developing a project of "Centro de Pesquisas de Energia Elétrica (CEPEL)", where the objective is the automation of climacteric tests. The first prototype was developed in Prolog for a compatible IBM PC, and it confirmed that the automation system is a satisfactory economic solution.

ÍNDICE

CAPÍTULO I

INTRODUÇÃO	1
------------------	---

CAPÍTULO II

DESCRIÇÃO DO PROBLEMA	4
II.1 Automação dos Ensaios Climáticos	4
II.2 Automação da Fase de Operação	7
II.2.1 Subsistema Gerente	9
II.2.1.1 Interface Homem-Máquina	9
II.2.1.2 Gerenciador do Processo	10
II.2.2 Módulo Identificador	12
II.2.3 Subsistemas TAC e GPIB	12
II.3 Observações do Sistema de Automação	13
II.4 Por que Inteligência Artificial ?	14

CAPÍTULO III

LEVANTAMENTO DAS TÉCNICAS DISPONÍVEIS NO DESENVOLVIMENTO DE SISTEMAS ESPECIALISTA	16
III.1 Uma Metodologia e uma Aplicação Complexa	16
III.2 Uma Metodologia sem Exemplo de Aplicação	17
III.3 Uma Metodologia para Sistemas de Produção	19
III.4 Uma Metodologia e muitos Exemplos de Aplicação .	20

CAPÍTULO IV

METODOLOGIA DE SOLUÇÃO	23
IV.1 Atividades da Metodologia de Desenvolvimento	23
IV.2 Estudo	26
IV.3 Aquisição do Conhecimento	27
IV.3.1 Componentes de um SE	29
IV.3.2 Adquirindo o Conhecimento do Especialista ...	31
IV.4 Análise do "Hardware"	32
IV.5 Representação do Conhecimento	34
IV.5.1 Estruturando o Conhecimento	36

IV.5.1.1	Esquema de Representação	37
IV.5.2	Base de Conhecimento Física	39
IV.5.2.1	Editor de Quadros	39
IV.5.2.2	Operações com os Quadros	41
IV.5.3	Conclusões da Representação do Conhecimento .	45
IV.6	Projeto	46
IV.7	Interface Homem-Máquina	47
IV.8	Implementação	49
IV.9	Teste do Desempenho	54

CAPÍTULO V

IMPLEMENTAÇÃO DA METODOLOGIA	57	
V.1	Estudo	57
V.1.1	Descrição do Problema	57
V.1.2	Necessidade de aplicar Técnicas de IA	59
V.1.3	Definição do Domínio e da BC inicial	59
V.1.4	Custos e Benefícios	61
V.1.5	Objetivos Gerais	61
V.1.6	Plano de Desenvolvimento	61
V.2	Aquisição do Conhecimento	62
V.2.1	Saída do Sistema Configurador	62
V.2.1.1	Restrições dos Parâmetros	63
V.2.1.2	Codificação dos Termos de Configuração	64
V.2.2	Definição do Domínio da BC	65
V.2.2.1	Normas de Especificação	65
V.2.2.2	Características dos Instrumentos	67
V.2.2.3	Características dos Equipamentos	69
V.2.3	Especificação Estruturada do Projeto	70
V.2.4	Especificação da IHM	74
V.2.5	Necessidade do "Hardware"	76
V.3	Análise do "Hardware"	78
V.4	Representação do Conhecimento	79
V.4.1	Base de Conhecimento Lógica	79
V.4.2	Base de Conhecimento Física	82
V.4.2.1	Hierarquia dos quadros	82
V.4.2.2	Seleção da Norma de Especificação	84
V.4.2.3	Norma 6796	85
V.4.2.4	Representação dos Instrumentos	92

V.5	Projeto	94
V.5.1	BC	94
V.5.2	MI	95
V.5.3	IHM	96
V.5.4	Plano de Teste	98
V.6	Implementação	99
V.7	Interface Homem-Máquina	100
V.8	Teste do Desempenho	110

CAPÍTULO VI

RESULTADOS OBTIDOS	116
---------------------------------	------------

CAPÍTULO VII

CONCLUSÕES	120
-------------------------	------------

REFERÊNCIAS BIBLIOGRÁFICAS	123
---	------------

APÊNDICE A: RELAÇÃO DOS COMANDOS DA AGENDA	127
---	------------

APÊNDICE B: SINTAXE DAS OPERAÇÕES DA AGENDA	133
--	------------

APÊNDICE C: PARÂMETROS DO OSCILOSCÓPIO	140
---	------------

APÊNDICE D: IMPLEMENTAÇÃO DO PROTÓTIPO EM PROLOG	142
---	------------

ÍNDICE DE FIGURAS

FIGURA	PAG.
II.1 "Hardware" envolvido na ATE	6
II.2 Esquema do "Software" da fase de Operação	8
II.3 Subsistema Gerente	9
IV.1 Etapas de criação de um SE	24
IV.2 Fase de ESTUDO	27
IV.3 Fase de AQUISIÇÃO DO CONHECIMENTO	28
IV.4 Esquema de um SE	30
IV.5 Fase de ANÁLISE DO "HARDWARE"	33
IV.6 Fase de REPRESENTAÇÃO DO CONHECIMENTO	34
IV.7 Estrutura do Editor de quadros	40
IV.8 Fase de PROJETO	46
V.1 Relação entre as fases de Operação e Configuração	58
V.2 Etapas de um EC	65
V.3 Esquema do "Software" do Configurador	70
V.4 Seleção do quadro dentro da Hierarquia	71
V.5 Variáveis envolvidas no EC de calor seco	73
V.6 Conceitos básicos dos testes	74
V.7 Esquema da BC do Configurador	81
V.8 Hierarquia do Conhecimento	82
V.9 Informação no Quadro ENSAIOS	83
V.10 Representação da Norma 6817	84
V.11 Representação da Norma 6796	86
V.12 Representação da etapa PRECONDICIONAMENTO	88
V.13 Representação da etapa MEDIÇÕES INICIAIS	89
V.14 Representação da etapa CONDICIONAMENTO	90
V.15 Representação da etapa ESTABILIZAÇÃO	91
V.16 Informação representada do Osciloscópio	92
V.17 Representação dos Comandos do Osciloscópio	93
V.18 Módulos do Configurador	97
V.19 Configuração de um EC para teste	98
V.20 Tela Principal	101
V.21 Tela de Seleção	103
V.22 Tela de Diálogo	104
V.23 Tela de Descrição da Norma	105
V.24 Etapas para configurar	106
V.25 Menu com valor "Default"	108
V.26 Uso da Ferramenta FAIXA	109

ÍNDICE DE TABELAS

TABELA	PAG.
IV.1 Ferramentas para desenvolver SE's em IBM PC	51
VI.1 Tempo consumido em cada fase da metodologia	116

NOMENCLATURAS

ATE:	Automação de Teste de Equipamentos
BC:	Base de Conhecimento
BD:	Base de Dados
CEPEL:	Centro de Pesquisas de Energia Elétrica
DOS:	Disk Operating System
EC:	Ensaio Climático
GBD:	Gerenciador de Bases de Dados
GPIB:	General Purpose Interface Bus
IA:	Inteligência Artificial
IBM:	International Business Machines Corp.
IHM:	Interface Homem-Máquina
MI:	Mecanismo de Inferência
NSO:	Núcleo do Sistema Operacional
PC:	Personal Computer
RC:	Representação do Conhecimento
SE:	Sistema Especialista
SBC:	Sistema Baseado no Conhecimento
TAC:	Terminal de Aquisição e Controle

CAPÍTULO I

INTRODUÇÃO

Este trabalho mostra uma metodologia de desenvolvimento de um sistema para configurar processos de controle em tempo real. D'AMBROSIO et alii (1987), LEINWEBER (1987), PERKUSICH et alii (1988) e BABB (1989) têm apresentado trabalhos que propõem o uso de técnicas de Inteligência Artificial*¹ (IA) para representar o conhecimento envolvido no problema e para desenvolver os Mecanismos de Inferência (MI) necessários para lidar com esse conhecimento, mas não mostram uma metodologia para desenvolver um sistema de configuração desses processos.

A metodologia proposta foi aplicada no desenvolvimento de um Sistema Especialista*² (SE) em configuração de Testes Automáticos em Ensaio Climáticos (EC's), propondo o uso de quadros ("frames")*³ para representar os objetos do domínio, surgindo assim a necessidade de desenvolver uma ferramenta para editar conhecimento baseado em quadros. Esta ferramenta é apresentada neste trabalho.

O objetivo dos EC's é determinar ou verificar as especificações ou características de funcionamento de um equipamento ou componente eletrônico. Para tanto é necessário submetê-lo a um teste sob diferentes condições climáticas, tais como temperatura, umidade e pressão. Um teste pode ser composto de uma série de EC's, onde cada um deles é determinado por normas de especificação (ABNT, 1981), as quais indicam também o rigor do teste. Um EC é composto de várias etapas, as quais são determinadas pelas normas de especificação, bem como as condições climáticas

*Mais detalhes sobre estes conceitos encontram-se em:

1. **Inteligência Artificial:** (BARR et alii, 1982); (FEIGENBAUM, 1977) e (RICH, 1988);
2. **Sistemas Especialistas:** (BUCHANAN e SHORTLIFFE, 1984); (NAYLOR, 1985) e (JACKSON, 1986);
3. **Quadros:** (MINSKY, 1975); (FIKES et alia, 1985) e (CHANDRASEKARAN, 1986).

que o objeto deve suportar e o tempo de duração de cada etapa.

Com as exigências climáticas indicadas pelas normas, deve-se selecionar os equipamentos condicionadores climáticos segundo suas características, tais como estufa, câmara climática ou congelador, que permitam atingir as condições do teste e os limites exigidos.

Antes, durante, e logo após executar cada etapa do EC, deve-se verificar o comportamento do objeto testado. Para isso é necessário selecionar os instrumentos para gerar e medir sinais, tais como gerador de funções, osciloscópio ou multímetro.

No Laboratório de Ensaios do Centro de Pesquisas de Energia Elétrica (CEPEL) são efetuados ensaios desse tipo, que inicialmente eram desenvolvidos totalmente por operadores humanos. Embora se tenha um grande conhecimento dos instrumentos e equipamentos usados, isto provoca muitos problemas de diferentes tipos, como, por exemplo, o estudo das muitas normas que podem ser usadas e o tempo gasto para especificar totalmente um EC, cuja a ordem de grandeza pode ser até de algumas semanas.

Cada vez que se deseja executar um novo EC é necessário configurar o processo completo para iniciar o seu controle. Esta é uma tarefa muito trabalhosa para o operador do sistema, que deve se preocupar com todas as ações que envolvem as operações a serem seguidas na evolução do EC, tornando o processo não muito confiável por ser longo e, conseqüentemente, monótono, tornando possível a omissão de algum detalhe. Pelos erros que se podem cometer num processo desta natureza, é possível que se tenha que invalidar semanas de trabalho.

Um fator importante na decisão de automatizar um sistema desta natureza é a duração dos EC's, os quais levam algumas semanas ou meses. Isto indica que os operadores

devem trabalhar em turnos fazendo as anotações da evolução do EC, sendo esta tarefa rotineira e muito cansativa, além de não manter a uniformidade.

Dados os problemas envolvidos em um EC, o CEPTEL decidiu estudar a viabilidade de um projeto de Automação de Testes de Equipamentos (ATE) como uma solução econômica, com o objetivo de auxiliar o operador nas tarefas de configuração, controle e avaliação dos EC's.

Estas razões motivaram o desenvolvimento do sistema Configurador de EC's. Este sistema é basicamente uma ferramenta de auxílio na configuração de processos de EC's. Pelas características apresentadas pelo Configurador foram estudadas as técnicas de IA, para verificar a viabilidade de desenvolver um SE em Configuração de Testes de EC's. Deste estudo obteve-se a base que motivou-nos a propor uma metodologia de desenvolvimento de um SE, que é o objetivo deste trabalho, no qual são envolvidas tarefas típicas de Configuração.

No CAPÍTULO II é apresentada a descrição do problema, onde se detalha o projeto ATE e o sistema de operação desenvolvido como base de suporte para o desenvolvimento do Configurador. Mostra-se também a motivação para estudar as técnicas existentes para desenvolver um SE desta natureza.

O CAPÍTULO III mostra o estudo bibliográfico feito para o levantamento das técnicas de IA disponíveis. A metodologia de solução proposta é apresentada no CAPÍTULO IV e sua implementação é mostrada no CAPÍTULO V.

No CAPÍTULO VI são apresentados os resultados obtidos e no CAPÍTULO VII são mostradas as conclusões do trabalho.

CAPÍTULO II

DESCRIÇÃO DO PROBLEMA

Neste capítulo detalha-se o projeto do sistema de automação dos EC's no CEPTEL. A automação do sistema tem como objetivo desenvolver um sistema inteligente que auxilie o operador na configuração dos EC's, para que depois eles sejam executados e controlados automaticamente, sem a intervenção humana. O sistema de operação mostrado neste capítulo foi desenvolvido como um primeiro estágio do projeto para se obter uma base sob a qual se apoia o desenvolvimento do Configurador de processos. Ao final, são mostradas as motivações de estudar as técnicas de IA para desenvolver um SE em Configuração.

II.1 Automação dos Ensaio Climáticos

O EC tem basicamente três fases principais: *Configuração*, *Operação* e *Avaliação*. A fase de *Configuração* tem como objetivo principal a especificação total do EC, no qual o especialista estuda e seleciona as normas que melhor se adaptam ao teste, segundo as características de funcionamento do objeto. Uma vez escolhidas as normas, deve-se selecionar os equipamentos climáticos e instrumentos de medição que sejam mais adequados ao teste. Em seguida é necessário configurar os parâmetros dos equipamentos climáticos e dos instrumentos envolvidos em cada etapa do ensaio, bem como a sequência e a característica destas etapas.

Uma vez especificado o EC com todas as suas etapas, é possível iniciar a segunda fase. A fase de *Operação* tem como objetivo principal adquirir os dados, acompanhar e gerenciar o desenvolvimento do EC, no qual deve-se monitorar, supervisionar e controlar o processo, que

normalmente durará algumas semanas, mas podendo durar até meses.

A terceira e última fase é a *Avaliação* do objeto testado. Nesta fase o objetivo é avaliar o comportamento do objeto durante o EC, verificando se os dados adquiridos estão dentro da faixa de aceitação das especificações do objeto.

O projeto de Automação de Testes de Equipamentos (ATE) engloba as três fases envolvidas num EC, ou seja, *Configuração, Operação e Avaliação*. O objetivo principal a ser alcançado com a ATE é diminuir, tanto quanto possível, a "interferência física do homem" dentro de um sistema totalmente desenvolvido e integrado, onde a atuação humana passa a ser mais de supervisão.

A automação da fase de *Operação* dos EC's foi desenvolvida como um primeiro estágio da automação total do sistema. O "hardware" envolvido neste sistema é mostrado na figura II.1, cuja descrição detalhada se encontra em ALBUQUERQUE e MENDONÇA (1989).

Nesta fase foi implementada uma Interface Homem-Máquina (IHM) simples com a qual o operador pode interagir para acompanhar o andamento do processo. Este módulo de *Operação* interpreta e executa as operações lidas de uma Base de Dados (BD) de entrada. A BD contém o código em alto nível das operações que devem ser efetuadas na execução do ensaio. Cada operação em alto nível é traduzida em comandos de baixo nível que são enviados para os equipamentos climáticos ou para os instrumentos de medição. Estes comandos consistem na configuração dos equipamentos climáticos e na configuração dos instrumentos de medição. Os equipamentos climáticos são os encarregados de manter as condições climáticas de temperatura, pressão e umidade do EC para permitir o controle. Os instrumentos de medição envolvidos em cada etapa permitem a monitoração dos sinais de interesse do objeto sob teste.

No fim da execução da fase de *Operação* tem-se todas as informações referentes à evolução do EC, tais como monitoração dos equipamentos (temperatura, pressão e umidade), monitoração dos instrumentos (tensão, corrente, frequência, etc.), alarmes ocorridos e comandos executados.

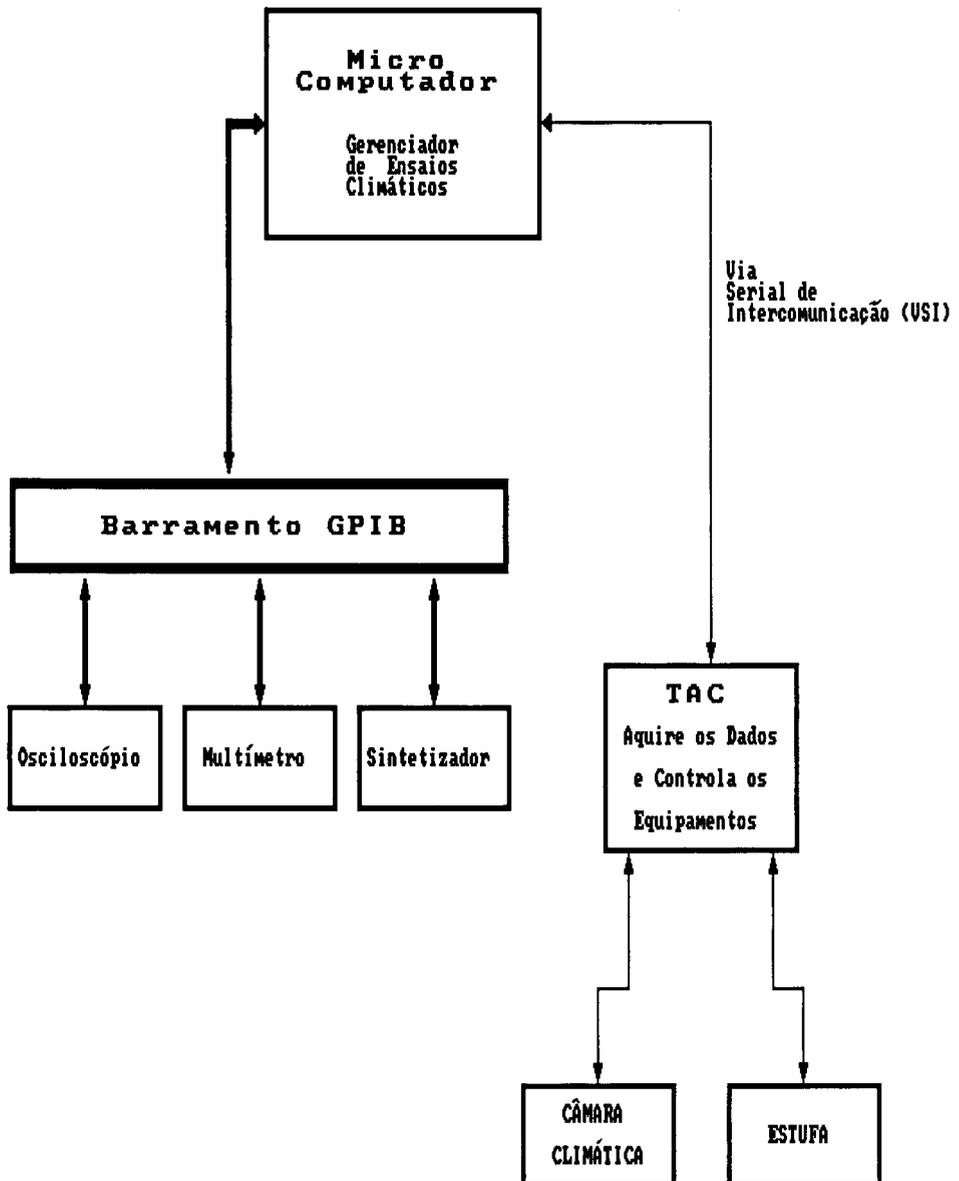


FIGURA II.1: Hardware envolvido na ATE.

II.2 Automação da Fase de Operação

O sistema de automação pode ser visto, em linhas gerais, como um sistema de supervisão e controle em tempo real que manipula uma grande quantidade de informações. O "software" envolvido na fase de *Operação* é estruturado em módulos que se relacionam em diversos níveis hierárquicos. O nível hierarquicamente mais alto é aquele onde todo o "hardware" do sistema é transparente para o usuário. O nível mais baixo lida com as instruções que atuam diretamente sobre o "hardware" do sistema.

A figura II.1 mostra o "hardware" envolvido no sistema, esclarecendo melhor a modularidade da estrutura do "software". Este "software" foi organizado em três subsistemas independentes:

- .**Subsistema Gerente:** gerenciador dos processos. Encarregado de ler os comandos, interpretá-los e enviá-los aos outros subsistemas através de um módulo chamado **Identificador**;
- .**Subsistema TAC** (Terminal de Aquisição e Controle): usado como interface com os equipamentos. O TAC permite controlar de forma independente os equipamentos condicionadores climáticos (CEPEL, 1988a);
- .**Subsistema GPIB** (General Purpose Interface Bus): usado como interface com os instrumentos. Esses instrumentos respondem ao padrão GPIB especificado pela Hewlett-Packard, também conhecido como barramento IEEE-488, amplamente usado na indústria.

Os três subsistemas estão ligados por um módulo de interface denominado módulo **Identificador** (figura II.2). A descrição detalhada deste módulo, bem como dos módulos TAC e GPIB, se encontram em ALBUQUERQUE e MENDONÇA (1989).

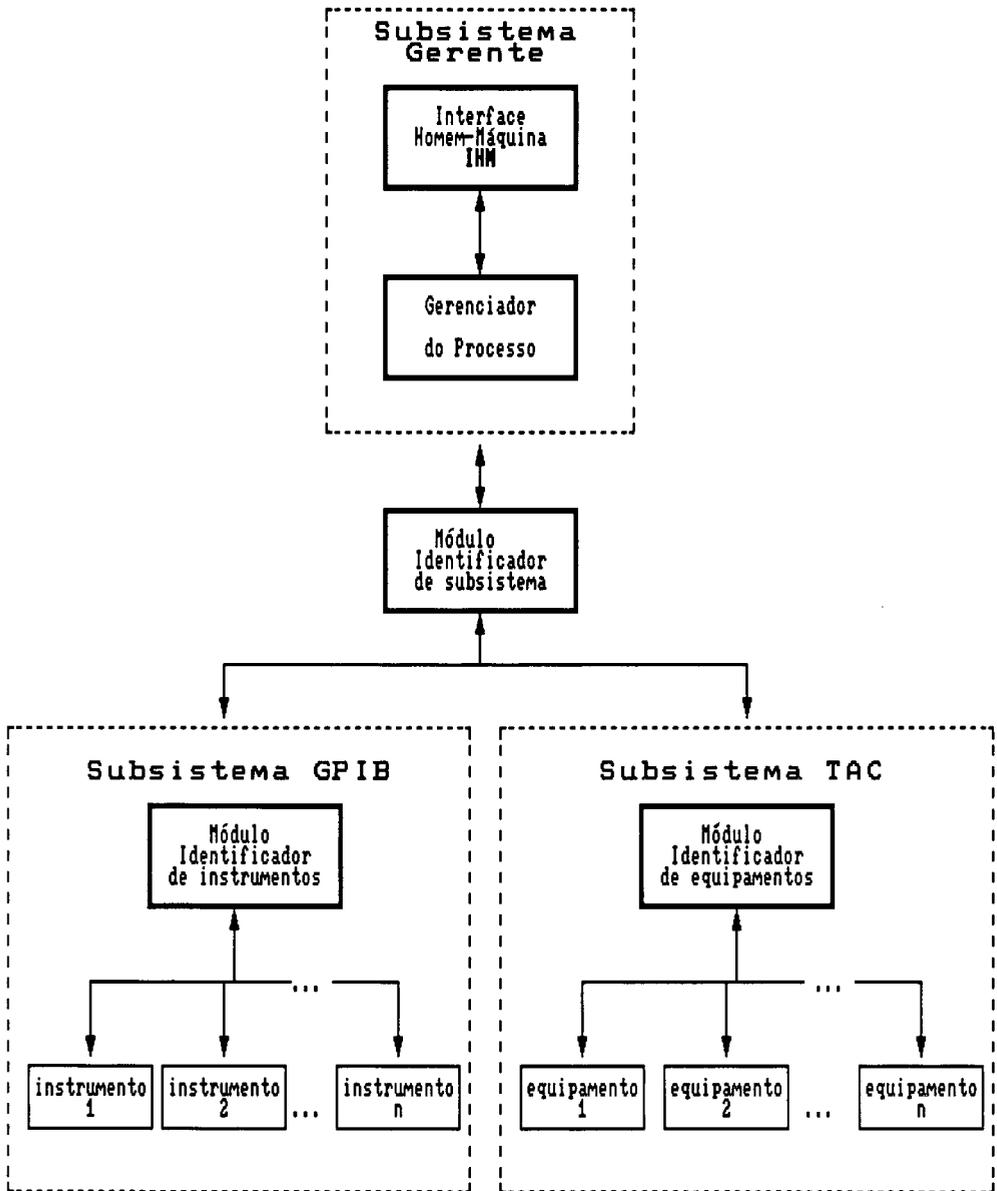


FIGURA II.2: Esquema do Software da fase de Operação da Automação.

II.2.1 Subsistema Gerente

O nível hierarquicamente mais alto corresponde ao subsistema Gerente, o qual é composto por uma IHM e um Gerenciador do Processo, como mostra a figura II.3.

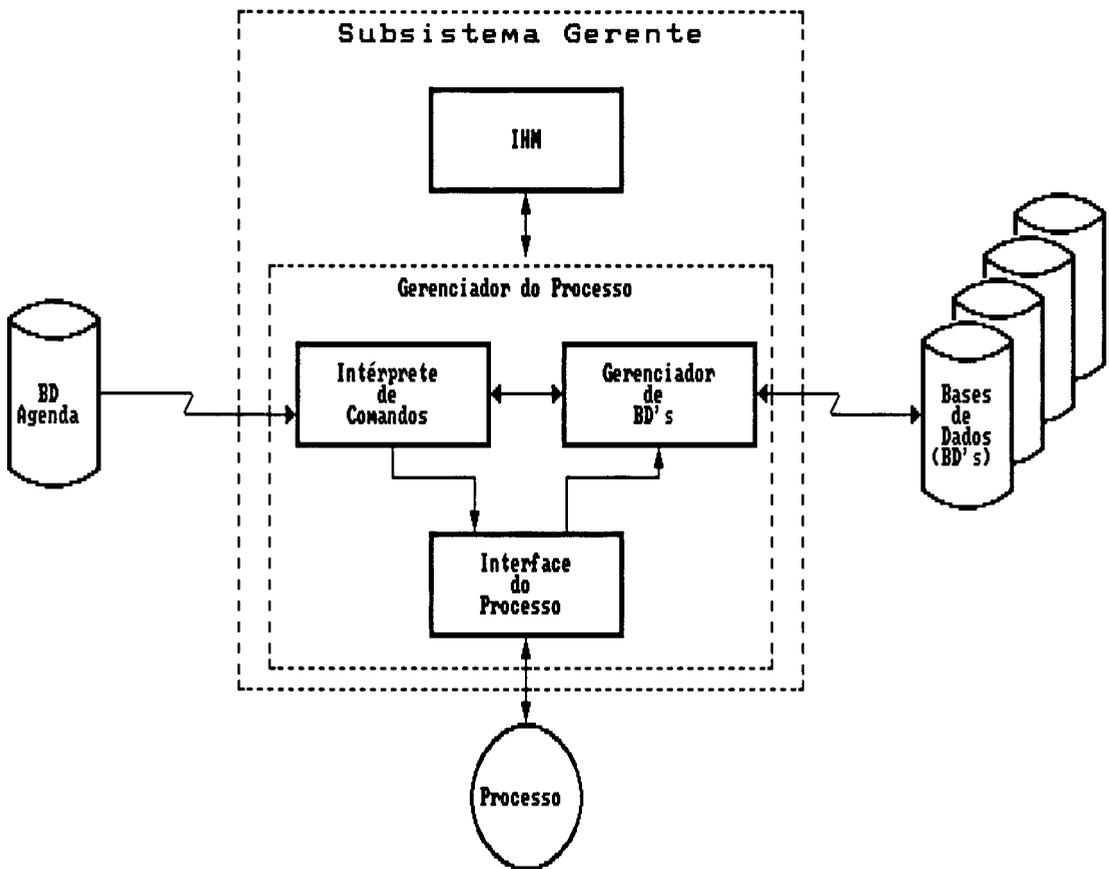


FIGURA II.3: Subsistema Gerente.

II.2.1.1 IHM

A IHM permite ao operador acompanhar o andamento do processo, como, por exemplo, observar medidas obtidas, comandos executados, gráficos de acompanhamento, alarmes ocorridos, etc. Todas as informações apresentadas pela IHM são obtidas das diferentes BD's, as quais são atualizadas em tempo real, ou seja, o operador do sistema sempre tem disponíveis as últimas informações do processo.

II.2.1.2 Gerenciador do Processo

O nível seguinte diretamente relacionado com a IHM corresponde ao Gerenciador do Processo. O Gerenciador é quem determina, a partir da BD chamada Agenda, que contém a configuração completa do ensaio, as operações que devem ser efetuadas no andamento do processo.

A função do Gerenciador é:

- interpretar os comandos de configuração;
- servir de interface entre o Gerente e os subsistemas GPIB e TAC, construindo, a partir desses comandos, as mensagens a serem enviadas a estes subsistemas e recebendo as respectivas mensagens retornadas;
- gerenciar as BD's que armazenam as informações obtidas dos subsistemas.

Intérprete de Comandos

A interpretação dos comandos consiste em três fases: ler um comando da BD da Configuração (Agenda), verificar se não contém erros e preparar a mensagem para enviar aos subsistemas TAC ou GPIB. Estes comandos podem ser do tipo:

- Configurar um instrumento para fazer uma medida;
- Configurar um equipamento climático para atingir uma temperatura determinada;
- Iniciar uma operação no equipamento climático;
- Cancelar uma operação no equipamento climático;
- Perguntar pelo status atual em um equipamento climático;
- Perguntar qual o valor de uma medida em um instrumento;
- Indicar as ligações entre os instrumentos que se deve efetuar.

Interface com o Processo

A Interface com o Processo constrói uma estrutura de dados para ser enviada aos subsistemas TAC ou GPIB. As mensagens a serem enviadas têm uma estrutura única que identifica para qual subsistema elas se destinam, indicando o instrumento ou equipamento correspondente, com os dados que são necessários.

Esta estrutura contém os seguintes dados:

- Identificador GPIB ou TAC;
- Identificação de instrumento ou equipamento;
- Tipo de operação que se deseja efetuar;
- Parâmetros da operação a realizar no instrumento ou equipamento;
- Início, duração e/ou término da operação.

Gerenciador de Bases de Dados (GBD)

A IHM exterioriza as informações provenientes das BD's, as quais são preenchidas dinamicamente em tempo real pelo GBD. O GBD se comunica com os dois subsistemas de aquisição de dados (TAC e GPIB) através da Interface com o Processo. O subsistema TAC é usado para monitorar e controlar os equipamentos envolvidos no processo e o subsistema GPIB é usado como interface dos instrumentos com o sistema.

A cada vez que é enviada uma mensagem para qualquer um dos subsistemas, é retornada uma mensagem, na qual se tem informação a respeito do andamento do processo. O GBD armazena esta informação na BD correspondente. Existem quatro BD's, cada uma contendo informações específicas sobre o processo, que são as seguintes:

- **ALARMES:** armazena todos os alarmes ocorridos no transcurso do processo;

- TAC: armazena as medidas relativas aos equipamentos;
- IEEE: armazena as medidas obtidas dos instrumentos;
- ROTEIRO: armazena os comandos executados no processo.

II.2.2 Módulo Identificador

O módulo Gerente da *Operação* interpreta instruções de operações lidas de uma BD de entrada e as executa. Estas instruções contidas na BD são códigos em alto nível das operações a serem realizadas no ensaio. Cada operação codificada em alto nível é traduzida em comandos de baixo nível que são enviados para os subsistemas TAC ou GPIB. Estes comandos constituem a configuração dos equipamentos e dos instrumentos envolvidos em cada etapa, para permitir a monitoração e controle do EC.

O módulo Identificador recebe a mensagem enviada pelo Gerente e identifica se o comando de operação ou instrução pertence ao subsistema GPIB ou TAC, passando o comando para o subsistema correspondente.

II.2.3 Subsistemas TAC e GPIB

O nível hierárquico imediatamente inferior ao do módulo Identificador contém os subsistemas GPIB e TAC, que determinam, a partir da mensagem recebida do módulo Identificador, o equipamento ou instrumento ao qual se destina o comando.

O subsistema TAC, coordenado pelo subsistema Gerente, é responsável, basicamente, pelo controle dos equipamentos condicionadores climáticos.

O desenvolvimento deste sistema de automação está sendo realizado em etapas. Primeiramente, foi implementada

a parte correspondente à automação de um único equipamento condicionador climático, no caso, uma Estufa. Atualmente está sendo desenvolvida a parte correspondente à automação de uma Câmara Climática, e assim progressivamente, até a integração completa de todos os equipamentos do laboratório.

O subsistema GPIB é o responsável pela recepção das mensagens destinadas aos instrumentos ligados ao barramento GPIB, pela identificação do instrumento ao qual a mensagem se destina, pelo envio das mensagens aos módulos correspondentes e pela recepção das respectivas respostas, encaminhando-as ao módulo Identificador. Nesta fase do projeto estão sendo utilizados três instrumentos: um Osciloscópio, um Multímetro e um Sintetizador.

Cada instrumento e equipamento tem características próprias e, para fazer a interface adequada, existe um módulo de "software" correspondente a cada um, conforme indicado na figura II.2 (instrumento 1, instrumento 2, ..., equipamento 1, equipamento 2, ...). Estes módulos recebem as instruções e as coloca em forma compatível com os parâmetros de cada instrumento ou equipamento, sendo este o nível hierarquicamente mais baixo.

II.3 Observações sobre o Sistema de Automação

O sistema de automação descrito neste capítulo é uma ferramenta disponível para o operador que lhe proporciona facilidades para verificação do andamento do processo.

O objetivo de desenvolver primeiro a fase de *operação* do sistema de automação é ter uma base sob a qual pode-se apoiar o desenvolvimento do *configurador* dos EC's. Para o desenvolvimento do *configurador* é importante conhecer qual é a saída que deve ser gerada, logo, conhecendo a estrutura dos comandos que podem ser executados na fase de

operação, é simples definir qual deve ser a saída do *configurador* para realizar o ensaio.

Baseado neste sistema de *operação* em funcionamento, o objetivo do *configurador* é gerar a BD com a configuração dos equipamentos e dos instrumentos para executar EC's.

II.4 Por que Inteligência Artificial ?

No desenvolvimento da fase de *Configuração* existe a necessidade de um grande conhecimento das normas nacionais e internacionais de especificação dos EC's. Além da aplicação de normas, também podem ser executados testes específicos requisitados por clientes, com a finalidade, por exemplo, de analisar o comportamento da amostra em situações especiais onde se supõe existir algum problema.

Outro conhecimento especializado que deve estar presente nesta fase corresponde às características dos equipamentos condicionadores climáticos e às características dos instrumentos usados para a aquisição de dados.

Durante a fase de *Configuração* deve-se selecionar as normas, equipamentos e instrumentos que permitirão a operação do EC. Isto indica que o *Configurador* deve possuir certos Mecanismos de Inferência (MI's) baseados nos conhecimentos acima mencionados para:

- selecionar normas de especificação de EC;
- selecionar equipamentos que cumpram as exigências das normas;
- selecionar os instrumentos para adquirir os dados do processo.

Detecta-se aqui a existência de um domínio de conhecimento muito raro e de alto custo. Raro, pelo fato de que os especialistas humanos são escassos e, quando

disponíveis, têm pouco tempo livre, dado que cada ensaio tem uma grande duração no tempo. Embora os especialistas sejam muito eficientes em seu trabalho, a possibilidade de confusão causada por muitas solicitações conflitantes os torna mais vulneráveis a erros do que um sistema baseado em computador. O alto custo é uma questão de tecnologia, dado que, com o advento dos microcomputadores pessoais, é possível implementar SE's com um custo relativamente baixo.

Com o *Configurador* pretende-se atingir o objetivo de diminuir o tempo de configuração de um ensaio, deixando ao especialista mais tempo para outras tarefas, e evitando erros que podem invalidar um ensaio depois de ter sido executado durante algumas semanas.

Este *Configurador* é uma ferramenta de auxílio para o especialista, que pode gerar automaticamente, e com uma confiabilidade muito maior, a BD das operações que serão executadas na fase de *Operação*.

CAPÍTULO III

LEVANTAMENTO DAS TÉCNICAS DISPONÍVEIS NO DESENVOLVIMENTO DE SISTEMAS ESPECIALISTAS

Inteligência Artificial (IA) é uma área de pesquisa muito abrangente, tratando de problemas que parecem ter pouco em comum, mas, diversos desses problemas, entretanto, podem ser resolvidos por sistemas conhecidos como Sistemas Especialistas (SE's) (FEIGENBAUM, 1977). Um SE pode ser uma alternativa de solução para um problema quando se mostra adequada a ele.

Quando se estuda um projeto para desenvolver um sistema computacional, um dos primeiros problemas que surgem é a metodologia a seguir, além de quais as atividades que devem ser realizadas, e, também, a decisão de aplicar ou não técnicas de IA. No levantamento bibliográfico feito para o estudo das técnicas disponíveis, foram pesquisados muitos trabalhos, resgatando, no nosso entender, os mais interessantes. Neste capítulo são resumidos os trabalhos de BUCHANAN e SHORTLIFFE (1984), HARMON e KING (1985), KELLER (1987) e WEISS e KULIKOWSKI (1988). Este levantamento é a base da proposta de metodologia apresentada no CAPÍTULO IV.

III.1 Uma Metodologia e uma Aplicação Complexa

BUCHANAN e SHORTLIFFE (1984) discutem no seu livro as idéias do projeto e implementação do sistema MYCIN, um dos primeiros SE's baseado em regras. Eles apresentam os temas envolvidos no desenvolvimento de um SE, que são: a construção da Base de Conhecimento (BC), a aquisição do conhecimento e o modelo de fator de incerteza para o raciocínio incerto. São mostradas as vantagens da Representação do Conhecimento (RC) na etapa experimental, com as regras e os quadros como uma solução de

representação. No fim, são apresentadas formas de avaliação do rendimento do sistema.

Estas são as principais atividades a serem executadas no desenvolvimento de um SE. Mas, pelo fato do MYCIN ser um sistema muito complexo, não é recomendável usá-lo como modelo no desenvolvimento de outros SE's.

III.2 Uma Metodologia sem Exemplo de Aplicação

HARMON e KING (1985) descrevem uma metodologia de desenvolvimento para SE's que envolve as seis atividades a seguir:

Atividade 1:

Objetivo: equacionar o problema, sendo esta uma tarefa que leva de um até três meses para completar os seguintes passos:

- Identificar o domínio do problema e a tarefa específica;
- Identificar o especialista que possa contribuir com o seu conhecimento;
- Identificar uma tentativa de solução para o problema;
- Analisar os custos e benefícios do esforço;
- Preparar um plano de desenvolvimento específico.

Atividade 2:

Objetivo: desenvolver um protótipo do sistema. A duração desta atividade pode levar de seis até nove meses, onde devem ser seguidos os passos:

- Estudar o domínio e a tarefa;
- Especificar critérios do rendimento;

- Selecionar as ferramentas para construir o SE;
- Desenvolver uma implementação inicial;
- Testar a implementação com o estudo de casos;
- Desenvolver um modelo detalhado para um SE completo.

Atividade 3:

Objetivo: construção de um SE completo, podendo ter uma duração de doze até dezoito meses. Os passos nesta atividade são:

- Implementar a parte central da estrutura do sistema completo;
- Expandir a BC;
- Adaptar a IHM do usuário;
- Monitorar o desempenho do sistema.

Atividade 4:

Objetivo: avaliação do sistema, onde se deve verificar o desempenho de estudo dos casos feitos na atividade 2.

Atividade 5:

Objetivo: integração do sistema, envolvendo os seguinte passos:

- Organizar a transferência de tecnologia;
- Desenvolver a interface do sistema com outras BD's, instrumentos, ou outro possível "hardware" para obter velocidade ou um sistema com interface mais simples para o usuário.

Atividade 6: é a última das atividades, onde se deve preocupar com a manutenção do sistema.

Para o desenvolvimento de um SE pode-se estudar esta metodologia, porém não foi dado um exemplo de aplicação, como no caso do MYCIN, que acompanha a evolução do sistema.

III.3 Uma Metodologia para Sistemas de Produção

WEISS e KULIKOWSKI (1988) apresentam uma metodologia que pode ser resumida nas três atividades a seguir:

Atividade A: Etapa inicial do projeto da BC, que compreende três fases principais:

- *Definição do Problema:* especificação dos objetivos, imposições, recursos, participantes e suas respectivas funções;
- *Conceituação:* descrição do problema e de como dividi-lo em sub-problemas;
- *Representação do problema em Computador:* escolha específica de uma representação para os elementos identificados durante a fase de conceituação e considerações sobre o fluxo das informações.

Atividade B: Etapa de Desenvolvimento e Teste do Protótipo. Uma vez escolhida a representação, é iniciada a implementação do protótipo de um subconjunto do conhecimento necessário para o sistema completo. Esta amostra deve ser representativa do conhecimento que seja típico de todo o modelo, embora deva compreender subtarefas e raciocínios que sejam suficientemente simples de testar.

Atividade C: Refinamento e Generalização da BC. Esta atividade pode consumir um tempo consideravelmente grande quando se deseja alcançar níveis muito altos de desempenho.

A apresentação desta metodologia tem muitos exemplos de aplicação, embora todos eles sejam baseados nos Sistemas de Produção, onde o conhecimento é representado por regras de produção.

III.4 Uma Metodologia e muitos Exemplos de Aplicação

KELLER (1987) propõe nove atividades a serem seguidas no desenvolvimento de um SE com o uso de técnicas de IA, as quais são resumidas a seguir:

Atividade 1: A linha principal do trabalho técnico no projeto é iniciado com o *Estudo*, ou a possibilidade do estudo, cuja principal tarefa é decidir se o projeto tem ou não a importância que justifique o desenvolvimento. Uma vez o projeto aprovado, então deve-se iniciar a atividade de *Análise Estruturada*.

Atividade 2: Análise Estruturada, onde são especificadas as necessidades do usuário em termos de funções a executar e os dados relacionados com estas funções. A especificação estruturada que resulta desta análise é usada na atividade 4, chamada *Projeto*.

Atividade 3: Projeto da BC. Durante a *Análise Estruturada*, a informação lógica requerida pelo *Projeto* (conhecimento e dados) deve ser identificada sem a preocupação de como será implementada no mundo físico. Essa descrição da BC Lógica é usada no Projeto Físico da BC para especificar os detalhes da implementação.

Atividade 4: Projeto, onde se especifica o sistema a ser desenvolvido para atender às necessidades do usuário.

Atividade 5 e 8: Integração do Sistema e Análise de "Hardware". Estas atividades certamente são importantes para o ciclo de desenvolvimento, mas nem sempre constituem parte direta do ciclo de especificação técnica (KELLER, 1987).

Atividade 6: Implementação - é dirigida à construção do sistema.

Atividade 7: Avaliação - corresponde aos testes para verificação do rendimento do SE desenvolvido.

Atividade 9: Aquisição do Conhecimento. Esta atividade se desenvolve ao longo de todas as fases do projeto, embora exista uma ênfase especial para esta atividade durante o *Estudo* e a *Análise Estruturada*. Depois da análise, as atividades do *Projeto* estão mais orientadas ao projeto do "software", e não à *Aquisição do Conhecimento*, embora este processo continue independentemente do resto. Em resumo, a evolução da BC, para se tornar um especialista, pode se estender além do desenvolvimento do protótipo. A *Aquisição do Conhecimento* é uma atividade independente onde a BC obtida está disponível para todas as outras partes do processo de desenvolvimento. Na prática é melhor ter uma cooperação mútua da BC com o *Estudo* e a *Análise*, para se tornar mais independente depois da *Análise*. Para isso sugere-se que a *Aquisição do Conhecimento* seja realizada pelo pessoal envolvido nas atividades de *Estudo* e *Análise* em forma paralela (KELLER, 1987).

Existem outros trabalhos onde são apresentadas metodologias de desenvolvimento de SE. Um dos trabalhos mais interessantes se refere às opiniões de um grupo relativamente grande de pesquisadores que compareceram, em 1980, a uma reunião sobre a criação de SE's, e que foram sintetizadas em um livro editado por HAYES-ROTH, WATERMAN e LENAT (1983), um dos primeiros livros que apresentam o tema.

As referências acima citadas foram tomadas como base para propor a metodologia de desenvolvimento que é o tema deste trabalho, a qual está sendo apresentada de forma

simples e clara para se tornar acessível a pessoas com pouca experiência na área de IA. A idéia é mostrar, através de um exemplo de aplicação, apresentado no CAPÍTULO V, a simplicidade no entendimento e na aplicação da metodologia proposta no CAPÍTULO IV. Esta metodologia tem uma forte influência do trabalho de KELLER (1987), mas foram feitas várias modificações.

CAPÍTULO IV

METODOLOGIA DE SOLUÇÃO

Um SE pode ser muito útil nas diversas áreas de aplicação, porém o seu projeto normalmente é complexo, quanto ao planejamento e ao gerenciamento. O objetivo deste capítulo é propor uma metodologia de desenvolvimento de SE's que diminua estas dificuldades de projeto. Esta metodologia tem como base o trabalho de KELLER (1987), com modificações baseadas principalmente nos trabalhos de HAYES-ROTH, WATERMAN e LENAT (1983), HARMON e KING (1985) e WEISS e KULIKOWSKI (1988).

IV.1 Atividades da Metodologia de Desenvolvimento

Alguns autores (HARMON e KING, 1985) afirmam que a chave do sucesso na criação de um SE é começar de maneira simples e progredir, através de incrementos, até um sistema consistente e significativo, cuja validação, que pode ser feita empiricamente, deve ser consumada à medida em que as várias etapas de aperfeiçoamento prosseguem. É importante mencionar, que para se alcançar níveis muito altos de desempenho, esta forma de abordar o problema pode consumir um tempo consideravelmente grande.

Esta é uma das razões pela qual outros autores (D'AMBROSIO et alii, 1987 e LEINWEBER, 1987) acham mais importante desenvolver um protótipo mais complexo, no qual se possa testar uma amostra representativa do conhecimento. Como veremos, isso envolve um trabalho mais intenso nas primeiras etapas do projeto, Estudo e Aquisição do Conhecimento, mas as vantagens obtidas são, sem dúvida, de grandes proporções para o futuro crescimento do SE.

Construir um protótipo o mais cedo possível, pode causar alterações drásticas na representação do

conhecimento, implicando um maior esforço no meio do projeto. Uma vez que já foi estudado o problema global, é bom dividir o problema em parcelas (o mais independente possível) e construir um protótipo de uma parcela do SE para testar os conceitos. Esta abordagem dificilmente vai provocar uma alteração drástica no modelo, por isso a metodologia apresentada adota esta filosofia.

A figura IV.1 mostra, em linhas gerais, o esquema da metodologia proposta.

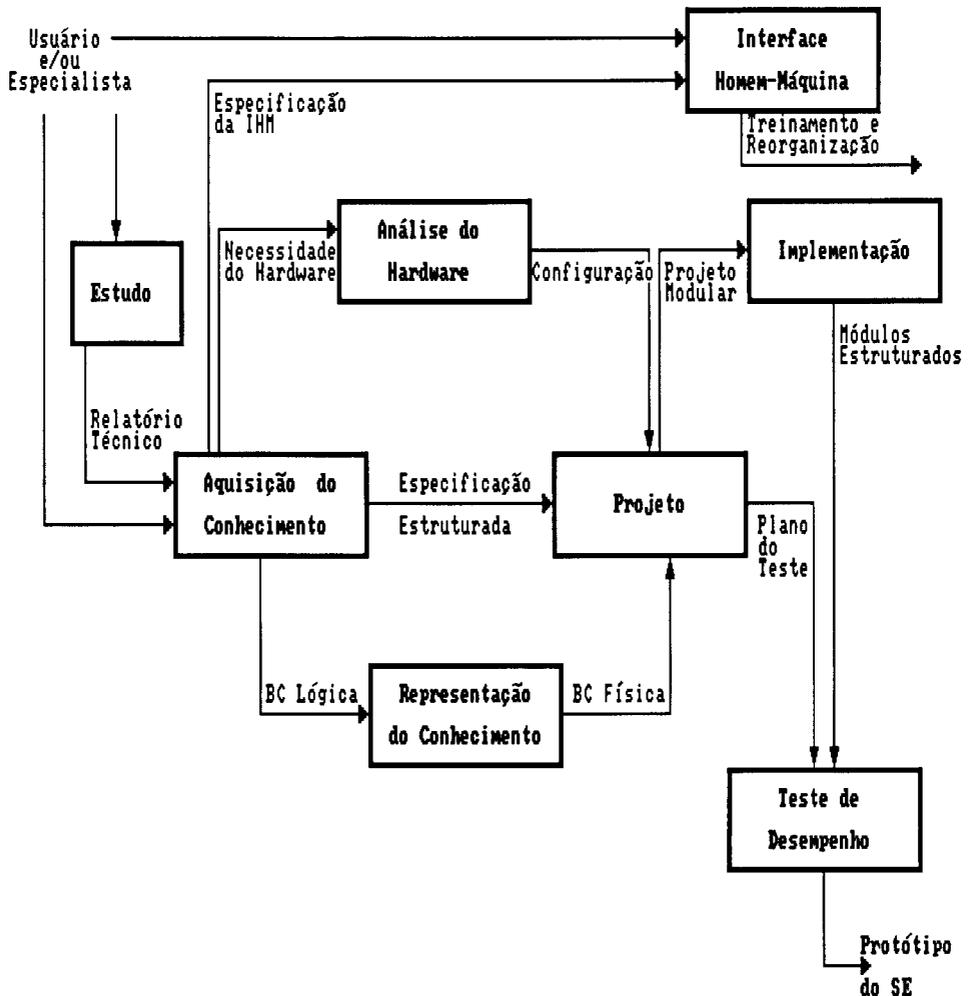


FIGURA IV.1: Etapas de criação de um SE.

Como se vê na figura IV.1, o usuário* tem grande participação na especificação do sistema. Segundo a solicitação deste, é feito um *Estudo* para decidir se o projeto é, ou não, adequado para ser desenvolvido com técnicas de IA. O especialista participa na atividade da *Aquisição do Conhecimento*, para determinar a nível de estruturação de conhecimento as especificações do sistema. E, no fim, o usuário deve se incorporar ao desenvolvimento da IHM, para ser treinado no sistema integrado.

Uma vez feito o *Estudo* do projeto e sendo este aprovado, então deve-se passar para a *Aquisição do Conhecimento*, onde são identificadas as necessidades do usuário em termos de funções que devem ser executadas e dados relacionados com estas funções. Desta atividade obtém-se dados muitos importantes para a definição do projeto, que são a BC lógica do domínio, a especificação estruturada do projeto, a especificação da IHM e as necessidades do "hardware".

Na fase de *Interface Homem-Máquina* é usada a especificação da IHM obtida da atividade de *Aquisição do Conhecimento* para treinar o usuário com o novo sistema e preparar o plano de reorganização.

Durante a *Aquisição do Conhecimento* as informações lógicas requeridas pelo projeto (conhecimento e dados) devem ser identificadas sem a preocupação de como serão implementadas no mundo físico. Essa descrição da BC lógica é usada na *Representação do Conhecimento* para especificar os detalhes da implementação da BC física.

A especificação estruturada resultante é usada na atividade chamada *Projeto* para especificar como as necessidades do usuário serão atendidas.

* O usuário pode ser considerado um especialista no domínio, porém, nem sempre é assim. O cliente geralmente solicita o projeto, sendo que o usuário será a pessoa que usará o sistema desenvolvido.

A necessidade do "hardware" é estudada na *Análise do Hardware* para determinar a configuração requerida pelo *Projeto*. Para a atividade *Projeto* é necessária a configuração do "hardware", a especificação estruturada do "software" e a BC física, para que se desenvolva um projeto modular do "software", o qual vai ser utilizado na atividade *Implementação*. Observa-se que o desenvolvimento de sistemas de forma modular é uma técnica que possui características apropriadas para lidar com IA, por isso são utilizados sistemas modulares em conjunto com técnicas de processamento de dados, obtendo-se, então, uma ferramenta que pode trazer excelentes benefícios.

Da atividade *Projeto* obtém-se também um plano do teste para avaliar o desempenho do SE depois da *Implementação*.

Ao final, depois da *Implementação*, deve-se realizar o *Teste de Desempenho* dos módulos estruturados do SE, para avaliar o seu rendimento.

IV.2 Estudo

A primeira atividade (figura IV.2) geralmente consome pouco tempo, na faixa de dias até semanas, no máximo. O objetivo principal é gerar um relatório técnico cujo conteúdo depende essencialmente do domínio potencial da aplicação, mas deve conter basicamente os seguintes pontos:

- uma demonstração da importância do Sistema Baseado no Conhecimento (SBC). Em alguns casos, é melhor mostrar em um diagrama de fluxo de dados do SBC e sua relação com as interfaces do meio ambiente onde será implantado;
- decidir se é adequado desenvolver o sistema com técnicas de IA;

- decidir qual é o sistema mais conveniente, ou seja, analisar vantagens e desvantagens de aplicar técnicas de IA;
- dar estimativas de custo para o projeto;
- prover alguns objetivos gerais;
- preparar um plano de desenvolvimento específico.



FIGURA IV.2: Fase de ESTUDO.

O relatório técnico para avaliar as vantagens de se aplicar técnicas de IA para desenvolver um SBC deve conter, além dos pontos acima, três pontos muito importantes para a segunda atividade (*Aquisição do Conhecimento*):

- explicação do sistema atual, com os seus respectivos problemas, e planejamento de uma solução;
- definição do escopo do domínio, para começar a definição da BC;
- definição da saída do sistema.

Com a ajuda deste *Estudo*, pode-se determinar quais são os domínios mais apropriados para se aplicar técnicas de IA.

IV.3 Aquisição do Conhecimento

Esta atividade (figura IV.3) é, na realidade, o começo da construção de um SE. O ponto de partida é adquirir o domínio do conhecimento de um ou mais especialistas. Também é importante consultar outras fontes

de informação, tais como livros e manuais, mas, geralmente, o conhecimento virá de um especialista que tem muita experiência no campo de interesse. Aqui é necessário o trabalho de um Engenheiro de Conhecimento, que organiza o conhecimento na forma apropriada para um SE.

Nesta atividade deve ser definida a especificação funcional do sistema. Em resumo, deve-se apresentar um fluxo dos dados e da estrutura da informação, detalhando as funções do "software", definindo as interfaces entre elas e estabelecendo as restrições do projeto. Como resultado obtém-se um projeto preliminar, com definições dos módulos que formam a estrutura do "software" e das relações entre estes.

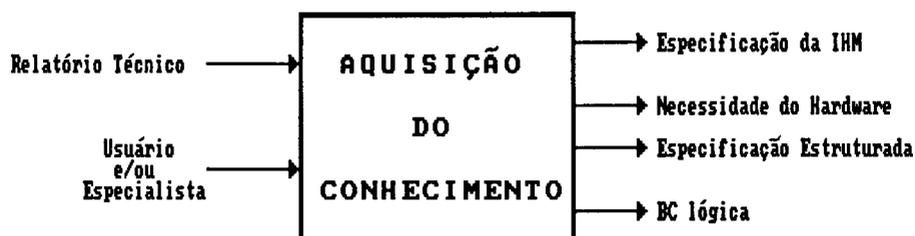


FIGURA IV.3: Fase de AQUISIÇÃO DO CONHECIMENTO.

Para adquirir o conhecimento em um projeto de configuração é importantíssima a informação do especialista. Esta informação, em conjunto com a contida no relatório técnico, é a base para especificar o protótipo inicial do projeto. Como objetivo geral da *Aquisição do Conhecimento* deve-se obter as seguintes informações (figura IV.3):

- a BC lógica;
- a especificação da estrutura do "software";
- a especificação da IHM;
- os requisitos do "hardware".

Estas informações estão diretamente relacionadas com os componentes de um SE, por isso, a seguir, é detalhado o esquema de um SE.

IV.3.1 Componentes de um SE

Na *Aquisição do Conhecimento* foi feita uma divisão que é compatível com muitas descrições dos esquemas mostrados para SE's. A figura IV.4 mostra o esquema típico de um SE, cujos componentes são:

IHM: é a comunicação entre o sistema e o usuário, com linguagem orientada ao problema: perguntas, comandos, dados e respostas adequadas que explicarão o comportamento do sistema.

BC: é onde se encontra o conhecimento do sistema sob algum dos esquemas de representação, tais como regras de produção, redes semânticas, quadros, cálculo de predicado de primeira ordem e outros.

MI: processa a BC para inferir soluções, gerar hipóteses ou propor diretrizes para a solução do problema. Os elementos do MI (HAYES-ROTH et alii, 1983):

Quadro Negro: retém os resultados intermediários;

Sequenciador: decide a ordem de processamento das regras pendentes;

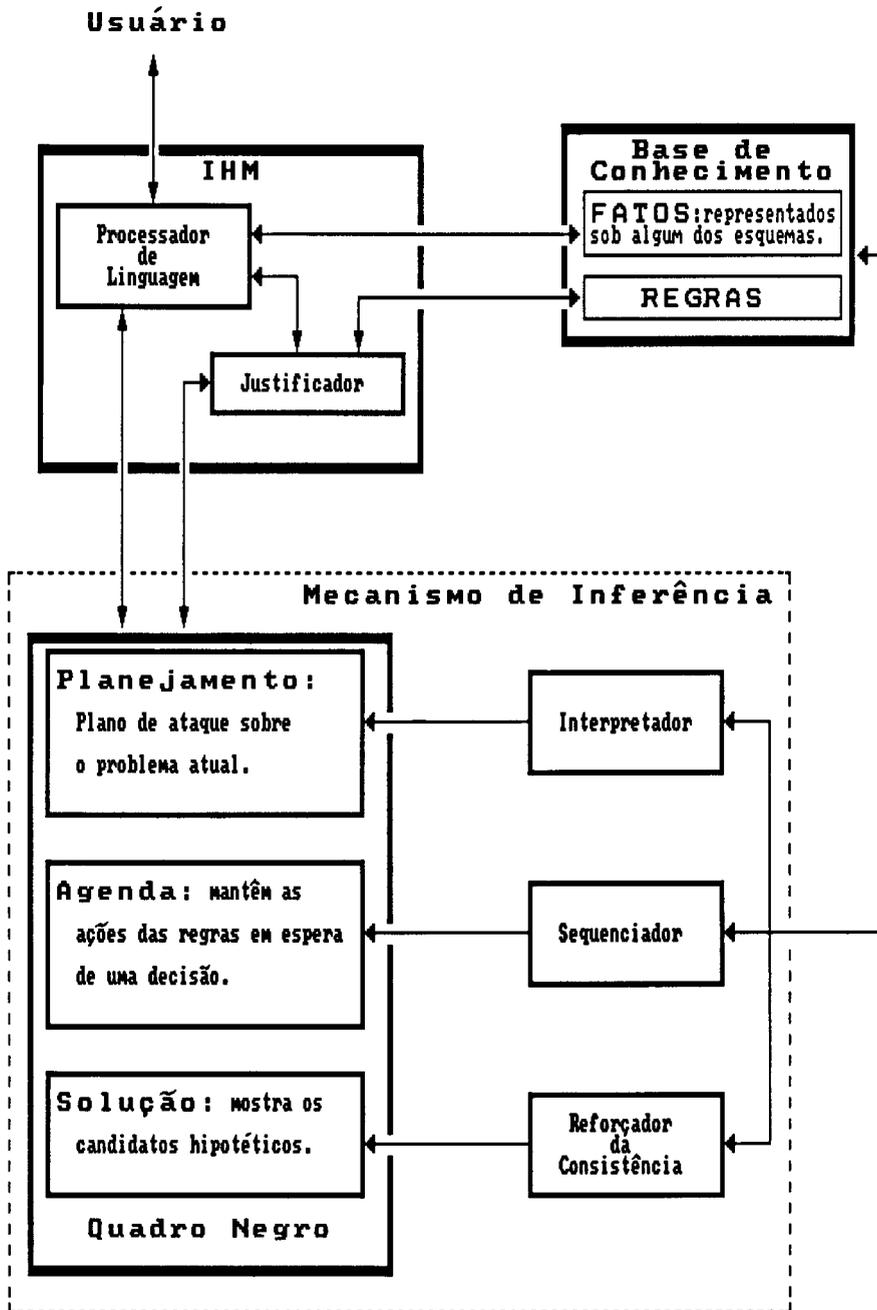


FIGURA IU.4: Esquema de um SE.

Interpretador: decide como aplicar as regras, segundo o que lhe indica o Sequenciador, para inferir novos conhecimentos;

Reforçador da Consistência: ajusta as conclusões prévias quando um novo dado (ou conhecimento) altera sua base de suporte.

IV.3.2 Adquirindo o Conhecimento do(s) Especialista(s)

O primeiro trabalho é selecionar um ou mais especialistas qualificados para atuar como fonte no domínio. Não vamos entrar em detalhes sobre este tópico. Uma vez selecionado o especialista, o processo de extração do conhecimento pode começar, definindo o escopo do conhecimento, identificando qual o objetivo a ser atingido como solução dos problemas a serem resolvidos, e determinando o conhecimento e esquemas requeridos para resolvê-los.

Primeiro, deve-se definir todas as possíveis soluções e caminhos. Esta é a saída do SE que o usuário procura e, portanto, o objetivo do sistema. Com esta informação na mão, pode-se trabalhar no sentido inverso para identificar o método de resolver o problema para obter essa saída.

O processo de trabalhar no sentido inverso acaba quando são identificadas as entradas do sistema. Esta identificação consiste em listar todos os fatos, tabelas, figuras e outras informações que o especialista precisa para resolver o problema. Neste ponto deve-se preocupar com a informação que o usuário deve fornecer ao SE para obter a solução. Todos os SE's precisam de um ponto de partida, e este ponto de partida é dado pelo usuário quando o SE pergunta por algum dado específico. Este dado será usado pelo MI para iniciar a busca da solução.

Portanto, uma vez identificado o método de obter a saída do SE, o engenheiro do conhecimento pode formalizar a BC, baseado nas entradas identificadas durante o processo. O método obtido deve ser, então, especificado. Esta especificação deve ser o mais estruturada possível, permitindo, desta forma, tornar mais independentes as soluções do problema.

Uma vez concluída a especificação do sistema, deve-se especificar a IHM. Neste ponto é importante identificar os dados de entrada que o usuário deve fornecer, e os dados de saída que o sistema deve apresentar, sem se preocupar com a forma em que será implementada.

No fim, deve-se indicar quais as necessidades básicas do "hardware" do sistema, para determinar, depois, na atividade de *Análise do "Hardware"*, qual é a configuração que melhor se adequa ao sistema.

IV.4 Análise do "Hardware"

Um SE, como se pode ver na figura IV.4, é formado por componentes de "software", ou seja, um programa de aplicação específico o qual envolve conceitos e técnicas de IA. Embora um SE seja constituído somente de módulos de "software", é importante uma *Análise de "Hardware"*, dado que o custo de desenvolvimento de um SE para um microcomputador do tipo IBM PC ou compatível é muito mais baixo do que o do desenvolvido para um computador de grande porte. É claro que, às vezes, existem limitações de velocidade de execução, espaço de memória, e até, em alguns casos, problemas de arquitetura. Mas, com o recente surgimento das novas tecnologias para os computadores pessoais, estes problemas podem não existir mais, embora toda tecnologia nova tenha um custo mais elevado no início.

Em um SE *Configurador* de processos é importante gerar um código que possa ser executado eficientemente na fase de

controle do processo. É por isso que, neste caso, é importante esta *Análise do "Hardware"*.

Para fazer um levantamento das vantagens que podem ser aproveitadas na geração do código, deve ser feita uma análise do "hardware" requerido na fase de controle do processo. Por exemplo, saber quando é possível gerar comandos que podem ser executados em paralelo, isto é, comandos que são enviados a diferentes pontos do processo simultaneamente, e, no entanto, não geram concorrência no processador central.

Todo processo sempre deve ser monitorado na fase de controle, e esses comandos de monitoração devem também ser configurados. A monitoração requer impreterivelmente o envolvimento de interfaces de "hardware" de comunicação do computador com o processo. A existência dessas interfaces físicas deve ser conhecida na fase de *Configuração* do processo para que possa ser determinada qual interface deve ser usada em cada caso.

No fim desta fase deve-se obter o "hardware" requerido pela fase de controle do processo (figura IV.5), assim como o "hardware" requerido pelo *Configurador*, que, em geral, utiliza o "hardware" da *Operação*.



FIGURA IV.5: Fase de ANÁLISE DO HARDWARE.

IV.5 Representação do Conhecimento

Aquisição e Representação do Conhecimento (figura IV.6) são as atividades que envolvem a maior parte do tempo no desenvolvimento de um SBC. O projetista da BC deve realizar constantemente operações para adicionar, modificar ou excluir informações referentes à modelagem do conhecimento.

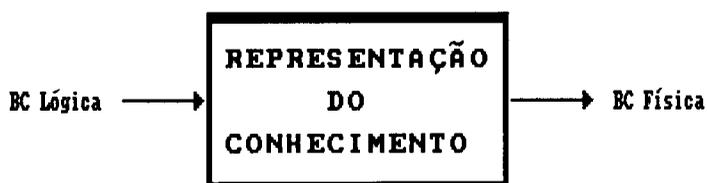


FIGURA IV.6: Fase de REPRESENTAÇÃO DO CONHECIMENTO.

Dado que um SBC deve usar conhecimento específico de um domínio, deve-se escolher uma forma eficiente de representá-lo. Os critérios básicos para a representação de conhecimento são os seguintes:

- *Expressividade* : o especialista deve poder comunicar seus conhecimentos eficientemente ao sistema;
- *Inteligibilidade* : o especialista deve entender o que o sistema conhece (ou sabe);
- *Acessibilidade* : o sistema deve saber usar o conhecimento que lhe foi dado.

Embora ainda não exista uma forma de representar o conhecimento segundo todos estes critérios (FIKES e KEHLER, 1985), o objetivo é tentar satisfazê-los. Existem muitas formas de representar o conhecimento (LEE, 1986):

Cálculo de predicado de primeira ordem: o conhecimento é representado como proposições

lógicas escritas em fórmulas bem definidas (wff's);

Rede semântica: consiste em um conjunto de nodos marcados (que representam os objetos ou os eventos), interligados por um conjunto de arcos marcados (que representam as relações entre esses objetos);

Quadros: representa fatos sobre objetos e eventos, onde os detalhes se encontram nos "slots" (MINSKY, 1975);

Regra de produção: são dados e fatos condicionais na forma *IF condição THEN ação*. As regras usam fatos ou dados conhecidos como base para deduções de novos fatos ou ações.

Não devemos esquecer que o presente trabalho trata de SE's para *Configuração*, e, como se pode notar, em geral este é um tema conhecido como "orientado ou voltado a objetos".

A tendência de orientação a objetos em diversas áreas tem notadamente influenciado as pesquisas em BC e linguagens de programação. Isto se deve ao fato de que cada entidade percebida no nível de abstração da aplicação, por mais complexa que seja, é representada por apenas um objeto definível e manipulável como uma unidade lógica num modelo voltado para objetos. Neste contexto não é necessário decompor a entidade em conceitos mais simples, como é o caso dos sistemas convencionais orientados para registros, alcançando-se, assim, uma representação da realidade mais natural.

Um elemento importante a favor da *expressividade* dos **quadros** reside nas facilidades oferecidas por eles para a descrição dos atributos, pois permitem definir restrições sobre os mesmos, o que ajuda a manter a consistência semântica da BC (SALERNO DE AQUINO et alii, 1987).

Os **quadros** têm sido considerados uma possível construção para representar conhecimento, enquanto que o enfoque baseado em regras de produção tem sido utilizado como um dos melhores métodos disponíveis para capturar o conhecimento especializado na solução de problemas. **Quadros** e regras de produção são construções autônomas e podem ser usados independentemente, embora haja sistemas que adotem um enfoque híbrido e altamente poderoso na representação do conhecimento.

Os **quadros** são, portanto, bastante úteis em situações onde o processamento do conhecimento é classificatório por natureza. Se o processamento envolver processamento de dados, então as linguagens orientadas a **quadros** usarão conceitos como os "valores ativos" e "métodos" que, para serem implementados, em geral requerem conhecimento mais interno da linguagem de implementação do sistema. Desta forma os **quadros** são mais adequados para representação de conhecimento no nível do projetista do sistema do que no nível do usuário final (MELO, 1988).

Além disso, uma grande vantagem das linguagens baseadas em **quadros** é a naturalidade com que são compreendidas pelo projetista de uma BC.

As linguagens baseadas em **quadros** provêm uma representação estruturada de objetos ou classes de objetos, e já incorporam alguns princípios organizacionais necessários, como os de generalização, classificação, agregação, projeção, além de possibilitar a associação de comportamento a objetos do domínio.

IV.5.1 Estruturando o Conhecimento

Neste ponto é importante destacar que a definição do conhecimento não deve se preocupar com a implementação, e sim deve-se esquematizar o conhecimento de forma

independente, obtendo como resultado a BC mínima (inicialmente) do sistema.

Uma linguagem baseada em **quadros** para a edição de conhecimento é uma ferramenta cujo uso facilita a tarefa de estruturar a BC. A linguagem faz uso das operações sobre a estrutura para criar novos **quadros**, "slots", facetas e valores, como também para apagá-los ou alterá-los. Desta forma o engenheiro do conhecimento pode selecionar as várias operações aqui apresentadas, até obter a estrutura desejada. A linguagem permite ainda especificar os mecanismos de controle, tais como obter informações do tipo "DEFAULT", herdadas ou executáveis (Daemons), para testar a consistência entre os **quadros** dentro da taxonomia.

IV.5.1.1 Esquema de Representação

Uma das várias formas de representar os **quadros** é como associação de listas encadeadas (WINSTON et alia, 1984). No nível mais alto, a estrutura de **quadros** consiste em uma lista de "slots" do tipo:

```
[<NomeQuadro>,[<NomeSlot1>,...],[<NomeSlot2>,...],...]
```

O primeiro elemento da lista é o nome do **quadro**, os elementos restantes são listas que representam os "slots". Em um nível mais abaixo, os elementos das listas dos "slots" possuem uma estrutura similar:

```
[<NomeSlot>,[<NomeFaceta1>,...],[<NomeFaceta2>,...],...]
```

Neste caso o primeiro elemento da lista é o nome do "slot". Os outros elementos da lista são facetas deste "slot". No último nível os elementos da lista das facetas são:

```
[<NomeFaceta>,<Valor 1>,<Valor 2>,...]
```

Pode-se ver que os **quadros** podem incluir tantos "slots", facetas e valores quanto for preciso. A representação final dos **quadros** fica da seguinte forma:

```
quadro([[NomeQuadro]], [[Slot 1], [Faceta 1, Valor 1,
                                Valor 2,
                                . . . ]],
        [Faceta 2, Valor 1,
          Valor 2,
          . . . ]],
        . . .
        ],
        [[Slot 2], [Faceta 1, Valor 1,
                    Valor 2,
                    . . . ]],
        [Faceta 2, Valor 1,
          Valor 2,
          . . . ]],
        . . .
        ],
        . . .
    ]).
```

Cada objeto ou classe é representado por um **quadro**. Os **quadros** podem ser organizados em taxonomias usando dois construtores que representam relações entre os **quadros**:

- ligações "**MEMBRO DE**" : representando elementos das classes;
- ligações "**SUBCLASSES**": representando subconjuntos das classes, que são especializações.

Os **quadros** incorporam conjuntos de descrições de atributos chamados "slots". Cada "slot" pode conter múltiplos valores e um conjunto de propriedades chamadas facetas, podendo adotar os seguintes tipos definidos na versão atual (entre outros):

- "**VALOR**" : indicando que é um dado fixo;
- "**DEFAULT**" : indicando um valor típico da variável, podendo ser mudado se for necessário;
- "**FAIXA**" : indicando que tem que ser um valor dentro desta faixa [min..max];
- "**OPÇÕES**" : é uma lista que contém valores válidos para as facetas "**DEFAULT**" e "**FAIXA**";

- "*SE PRECISA*", "*SE APAGA*" : são procedimentos ou conjuntos de regras, os quais são invocados quando o valor do "slot" é acessado ou armazenado (Daemons).

IV.5.2 BC Física

Uma linguagem baseada em **quadros** é adequada quando o conhecimento envolvido no domínio é fortemente estruturado, tanto a nível da descrição dos objetos como das relações que estes mantêm entre si. Uma linguagem baseada em **quadros** dá ao projetista da BC um meio simples de descrever os tipos de objetos do domínio que o sistema deve modelar.

IV.5.2.1 Editor de Quadros

O editor de **quadros** proposto neste trabalho inclui as facilidades consideradas mais importantes na operação com os **quadros** (figura IV.7), tais como criar, apagar, alterar e obter informação.

A primeira opção permite *processar a BC*, isto é:

- carregar uma BC existente para alterar o conteúdo;
- iniciar uma nova BC, indicando o nome do arquivo onde será armazenada;
- fazer um "backup" da BC corrente.

A segunda opção (*Editar BC*) dá ao projetista as facilidades para manipular a BC, selecionando as operações *criar, apagar, alterar* ou *obter*. A operação *criar* permite:

- *criar* um novo **quadro**;
- *criar* um novo "slot" dentro de um **quadro**;
- *criar* uma nova faceta dentro de um "slot";
- *criar* um novo valor em uma faceta.

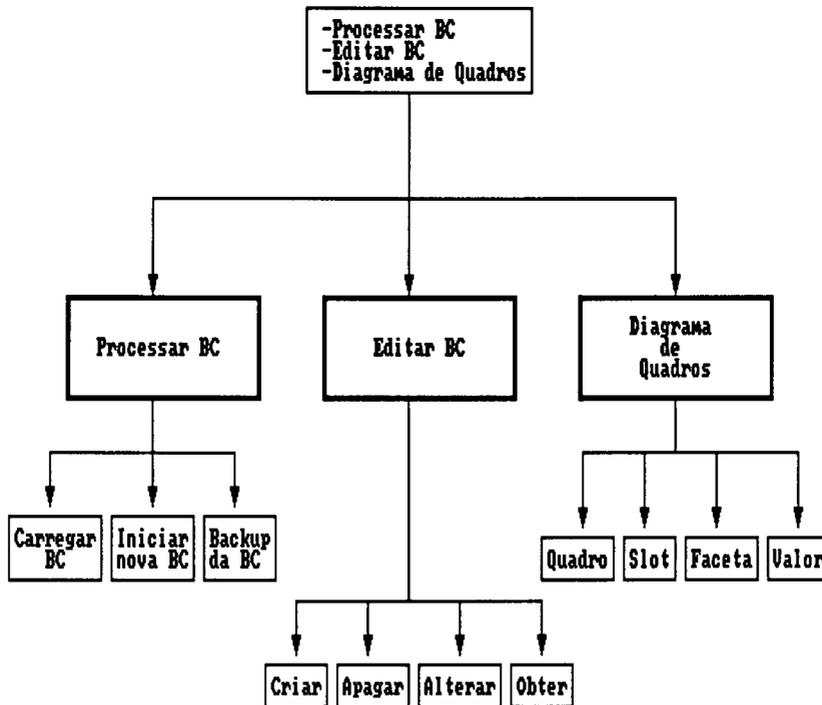


FIGURA IV.7: Estrutura do Editor de Quadros.

O editor vai consultando pelos dados, indicando qualquer tipo de erro na tentativa de criar uma nova informação.

Assim, além da operação *criar*, dispõe-se das operações *apagar*, *alterar* e *obter*. A operação *apagar* permite:

- *apagar* um **quadro**;
- *apagar* um "slot" dentro de um **quadro**;
- *apagar* uma faceta de um "slot";
- *apagar* um valor de uma faceta.

Obter dá a informação sobre a estrutura que se deseja. Primeiro mostra os **quadros**, onde se deve selecionar

aquele que é de interesse. Depois mostra os "slots" do **quadro** selecionado, e assim sucessivamente até chegar à informação desejada. A operação *alterar* permite trocar nomes dos **quadros**, "slots", facetas ou valores já criados.

Tem-se a opção *Diagrama de quadros*, que provê as mesmas operações acima mencionadas, porém, neste caso, a manipulação da informação é feita de forma gráfica, o que permite visualizar o esquema da hierarquia dos **quadros** em uma estrutura de árvore.

IV.5.2.2 Operações com os Quadros

Dado o esquema de representação dos **quadros**, apresenta-se a implementação em Prolog, das operações sobre essa estrutura, tais como *criar*, *apagar* ou *obter* um dado do tipo **quadro**, "slot", faceta ou valor. Estas operações são transparentes ao projetista, mas é interessante conhecê-las detalhadamente, dado que são as mesmas que o MI do SBC irá usar para a manipulação do conhecimento (KELLER, 1987).

Obter Informação

É possível obter a estrutura completa de um **quadro** fornecendo o seu nome como entrada para a operação *obtem*. A operação *obtem* consiste em verificar se o **quadro** existe na BC, o qual é retornado se existir:

```
obtem(RetFr, NomeFr)
    if quadros([NomeFr|Tail])
    and RetFr = [NomeFr|Tail].
```

Quando se deseja obter informação de um "slot", além do nome do **quadro**, deve-se passar o nome do "slot" para o operador *obtem*:

```
obtem(RetornaSl, NomeFr, NomeSl)
    if slot de([Nome|ValorDoSlot], NomeFr)
    and Nome = NomeSl
    and RetornaSl = [NomeSl|ValorDoSlot] and !.
```

A operação *slot_de* fornece sucessivamente (por causa do "backtracking") cada "slot" do **quadro**. Se o nome do "slot" é o nome procurado (NomeSl), então o operador *obtem* retorna o "slot" desejado.

No caso de obter uma faceta, precisa-se também do nome da faceta procurada. A operação *faceta_de* obtém cada faceta do "slot" ("backtracking"), até achar aquela com o nome procurado (NomeFa), então o operador *obtem* retorna o "slot" desejado:

```
obtem(RetornaFaceta, NomeFr, NomeSl, NomeFa)
  if faceta_de([Nome|Valor], NomeFr, NomeSl)
    and cabeça(NomeSl, Sl)
    and Sl >< Nome
    and Nome = NomeFa
    and RetornaFaceta = [NomeFa|Valor].
```

Para obter um valor, ou para verificar se existe, primeiro obtém-se a faceta, que é uma lista que contém todos os valores desta faceta, e logo pode-se verificar se o valor procurado pertence à lista da faceta:

```
obtem(LstData, NomeFr, NomeSl, NomeFa, Dado)
  if obtem([NomDd|LstData], NomeFr, NomeSl, NomeFa)
    and membro(Dado, LstData).
```

Criar Informação

Quando se quer acrescentar uma informação, existem quatro situações:

- o **quadro** não existe;
- o **quadro** existe, mas não o "slot" que se deseja;
- existem **quadro** e "slot", mas não a faceta;
- existem **quadro**, "Slot" e faceta.

O primeiro caso é simples de resolver, pois dado que o **quadro** não existe, cria-se a estrutura do **quadro**, o qual será acrescentado na BC:

```

cria(Frcriado, NomeFr, NomeSl, NomeFa, Dado)
  if not(quadros([NomeFr|_]))
    and Faceta = [NomeFa| Dado]
    and Sl = [NomeSl| [Faceta]]
    and Frcriado = [NomeFr| [Sl]]
    and insere(quadros(Frcriado)).

```

Se o **quadro** existir, a cláusula anterior falha, portanto tenta-se a segunda cláusula, a qual verifica se o "slot" existe. Caso não exista, obtém-se o **quadro**, forma-se a estrutura do novo **quadro**, acrescentando o novo "slot", logo coloca-se o novo **quadro** na BC e retira-se o antigo:

```

cria(Fracres, NomeFr, NomeSl, NomeFa, Dado)
  if not(obtem(_, NomeFr, NomeSl))
    and obtem(ModifQuadro, NomeFr)
    and append([NomeSl], [[NomeFa|Dado]], SlotNovo)
    and append(ModifQuadro, [SlotNovo], Fracres)
    and !
    and substituiQuadro(ModifQuadro, Fracres).

```

A operação *cria* possui mais duas cláusulas, as quais são tentadas no caso de falhar a segunda cláusula. Neste caso verifica-se a existência da faceta, a qual é criada se não existir. Se existir, tenta-se a última cláusula, que verifica se não existe o valor. Se este for o caso então forma-se a nova estrutura que substitui a antiga.

Apagar Informação

A operação de apagar um **quadro** (*Apaga*) é simples. Primeiro verifica a existência do **quadro** na BC, e logo, se existir, é apagado:

```

apaga([NomeFr|RestFr], NomeFr)
  if quadros([NomeFr|RestFr])
    and retira(quadros([NomeFr|RestFr])).

```

A operação *apagar* um "slot" verifica a existência do "slot", e logo *obtem* o **quadro** para verificar seu comprimento. Se o comprimento do **quadro** é dois, quer dizer que o **quadro** possui só aquele "slot" (o outro é o nome do **quadro**), então retira o **quadro** completo da BC. Caso o

quadro possua um comprimento maior, a primeira cláusula falha, tentando-se a segunda. Esta segunda cláusula *obtem* o "slot" e o **quadro** ao qual pertence o "slot", em seguida faz a operação *limpar*, a qual forma no seu terceiro argumento uma lista idêntica ao segundo argumento, porém sem o elemento indicado no primeiro argumento. Logo substitui-se na BC o novo **quadro**, obtido da operação *limpar*, pelo antigo:

```

apaga([NomeSl|RestoSl], NomeFr, NomeSl)
  if obtem([NomeSl|RestoSl], NomeFr, NomeSl)
    and obtem(QuadroNovo, NomeFr)
    and comprimento(QuadroNovo, 2)
    and retira(quadros(QuadroNovo)).
apaga([NomeSl|RestoSl], NomeFr, NomeSl)
  if obtem([NomeSl|RestoSl], NomeFr, NomeSl)
    and SlotApagar = [NomeSl|RestoSl]
    and obtem(QuadroMod, NomeFr)
    and limpar(SlotApagar, QuadroMod, Frp)
    and substituiQuadro(QuadroMod, Frp).

```

As operações *apagar* uma faceta e *apagar* um valor têm uma lógica similar à *apagar* um "slot".

Consistência da Taxonomia

Até agora foram detalhadas algumas das operações que são usadas na edição para a construção da BC. O conhecimento armazenado em **quadros** sempre é tratado como a BD de um SBC, onde o controle do raciocínio é levado para outra parte do sistema. Portanto, é preciso destacar que essas operações básicas serão sempre usadas no MI para o controle do raciocínio.

Uma dessas operações é *obtemDaemon*, que atua da seguinte forma:

- Quando se deseja obter um valor de um "slot", a primeira tentativa é verificar o seu valor na faceta "VALOR";
- Caso não haja valor na faceta "VALOR", então tenta-se obter o seu valor "DEFAULT";

- Caso não haja valor "DEFAULT", então tenta-se obter o valor sob demanda ativando-se uma operação (ou regra) indicada na faceta "SE PRECISA", a qual determina o valor do "slot".

```
obtemDaemon(V1,Fr,S1) if obtem(V1,Fr,S1,"VALOR").
obtemDaemon(V1,Fr,S1) if obtem(V1,Fr,S1,"DEFAULT").
obtemDaemon(V1,Fr,S1)
    if obtem([X1|[X2|X3]],Fr,S1,"SE PRECISA")
        and cabeça(X3,C)
        and relação(X1,X2,C).
```

Assim, além desta operação, estão implementadas as operações para obter valores herdados, sendo possível herdar valores subindo na hierarquia dos **quadros** pelas ligações "MEMBRO DE", ou partindo do primeiro nível até obter o valor descendo na hierarquia, pelas ligações "SUBCLASSES". Estas operações dependem da aplicação, as quais estão detalhadas no CAPÍTULO V.

IV.5.3 Conclusões da Representação do Conhecimento

Um editor de **quadros** é uma ferramenta de fácil interação com o usuário e confiável, que permite uma importante redução de tempo na implementação da estruturação do conhecimento, num escopo grande de aplicações, onde o conhecimento pode ser facilmente acrescentado ou alterado, deixando ao projetista mais tempo para outras tarefas.

Na construção do controle do raciocínio, que, no caso dos sistemas baseados em **quadros**, é sempre levado para outra parte do sistema, deve-se dispor das operações para acessar **quadros**, o que torna mais simples o desenvolvimento dessa etapa.

IV.6 Projeto

Esta fase (figura IV.8) começa quando se tem disponível a especificação estruturada (saída da fase de *Aquisição do Conhecimento*), a BC física (saída da fase de *Representação do Conhecimento*) e a configuração do "hardware" (saída da fase de *Análise do "Hardware"*). Com estas informações deve-se fazer um esquema da distribuição dos dados e programas em módulos de "software" e BD's.

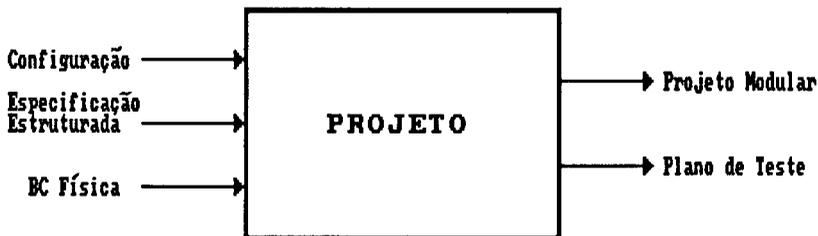


FIGURA IV.8: Fase de PROJETO.

Deve ser feita uma descrição do comportamento interno de cada módulo da estrutura de "software" e indicar interfaces entre estes módulos.

Nesta parte do projeto é necessário estudar os métodos de solução do problema, existindo basicamente dois métodos: Algorítmico e Dirigido pelos dados.

Na solução algorítmica a sequência de passos é bem definida com alguns saltos no programa, também, bem definidos. O caminho até a solução depende remota e unicamente dos dados do problema. Neste tipo de solução todos os saltos são conhecidos antes da execução. A estrutura de controle do programa é fixa e independe dos dados. Sendo este controle estático, na implementação a tendência é otimizar o rendimento do sistema.

Em uma solução dirigida pelos dados, como o nome diz, os dados controlam fortemente os passos seguidos para obter a solução. O conhecimento pode ser aplicado em qualquer ponto do processo para tomar uma decisão, enquanto novos conhecimentos são somados ou invalidados na BC, o controle cresce ou muda com o tempo. Muitos dos saltos do programa são conhecidos só na execução do programa, sendo muito difícil de resolver algoritmicamente. Este tipo de solução trabalha bem com tarefas onde a informação é difícil de organizar.

Na elaboração do projeto de um SE é importante notar que a solução geralmente é do tipo dirigida pelos dados, mais difícil de ser implementada.

Toda a informação gerada nesta fase é de muita importância para a fase de *Implementação*, na qual vão ser desenvolvidos os diferentes módulos.

Uma vez definido o projeto modular do sistema deve-se elaborar um plano de teste para avaliar o comportamento da implementação do sistema.

IV.7 Interface Homem-Máquina (IHM)

A IHM é a comunicação entre o usuário e o sistema. Aqui deve ser examinada a forma na qual o SE adquire os dados do usuário e como os apresenta. São usados três métodos básicos. O primeiro é a forma de pergunta-resposta (processamento de linguagem natural), onde a IHM apresenta uma pergunta à qual o usuário deve responder escrevendo a informação da resposta no teclado. Outros sistemas utilizam menus. Neste caso a pergunta é feita e a IHM apresenta alternativas de múltipla escolha, onde o usuário deve selecionar a resposta apropriada.

No caso de um SE em *Configuração*, uma interface com processamento de linguagem natural não é a mais adequada.

Para este caso uma interface gráfica, que apresenta informações visuais, facilita a tarefa do operador que está configurando o processo.

As interfaces textuais para o usuário deram lugar às interfaces gráficas, onde o estilo de interação pode ser a manipulação direta de objetos. Por exemplo, pode-se deslocar um objeto, ou fazer uma rotação neste objeto, abstraindo-se dos seus detalhes. Também, esta é uma área de proveitosa aplicação da "orientação a objetos". Ou seja, mais uma razão para que sejam usados **quadros** na representação dos objetos dentro do domínio em SE's de *Configuração*.

Neste ponto deve-se estudar também o uso de outras formas para trocar informações com o usuário. Por exemplo, algumas IHM's incorporam janelas, que são pequenas telas de informação dentro da tela principal do monitor. Estes blocos de informação, correspondentes a diferentes partes da configuração, permitem visualizar e manipular mais de uma idéia ao mesmo tempo.

Outro fator que facilita a interação na IHM é a cor. Alguns sistemas provêm uma tela de informação totalmente colorida. A cor pode contribuir bastante na velocidade de operação do sistema e na simplicidade de uso, por exemplo no processo de busca de solução para problemas complexos.

A organização da informação deve ter:

- Organização hierárquica das telas;
- Posicionamento de mensagens em pontos fixos da tela, preferencialmente na primeira linha.

Outro ponto importante é a codificação da informação na tela, resumindo:

- Devem ser usados termos simples;

- Correspondência única entre termos e símbolos para um determinado objeto;
- Para variáveis cujo valor deve ser observado de forma exata devem ser usados valores digitais, porém não excedendo quatro dígitos e evitando-se notações de engenharia. Devem ser utilizadas unidades de engenharia (mV, °C, KHz, etc.);
- Somente usar campo piscante para eventos que requerem ação rápida do operador, por exemplo reconhecimento de um erro;
- Utilizar letras maiúsculas para termos, nomes, mensagens de status, ou descrições, para melhorar a identificação;
- Utilização de abreviaturas somente para termos muitos conhecidos, senão é preferível escrever por extenso;
- Os símbolos usados devem ser de formatos bastante diferentes.

IV.8 Implementação

Nesta fase existem três módulos que devem ser implementados: a BC, o MI e a IHM. A implementação destes módulos não é independente, considerando-se que depois devem ser integrados para formar o SE.

Para isso é preciso fazer um estudo das ferramentas disponíveis para a implementação dos módulos do projeto. No início existem duas opções:

- selecionar uma ferramenta já disponível;
- desenvolver uma ferramenta nova.

Como foi mencionado na fase de *Análise do "Hardware"* (seção IV.4), uma solução econômica para o desenvolvimento de um SE é implementá-lo em um microcomputador do tipo IBM PC ou compatível. O sistema operacional mais popular de um microcomputador IBM PC ou compatível é o Disk Operating

System (DOS)**. Este sistema operacional permite executar as primitivas em tempo real do Núcleo do Sistema Operacional (NSO) (CEPEL, 1988b). O NSO provê um ambiente de multiprogramação, onde vários programas aplicativos, denominados tarefas, podem ser executados independentemente. É baseado no conceito de programação concorrente, onde funções concorrentes são executadas por objetos, que são módulos de programa que se comunicam uns com os outros exclusivamente através de intercâmbios pela troca de mensagens.

Um sistema operacional que execute primitivas em tempo real, como o NSO, é importante, dado que em geral um *Configurador* de processos deve gerar um código que possa ser executado em um ambiente em tempo real.

GIORNO (1987) relacionou como vantagens do uso das ferramentas o seguinte:

- menor tempo de desenvolvimento do projeto;
- menor trabalho para o engenheiro do conhecimento;
- maior envolvimento do especialista no desenvolvimento do sistema;
- maior entendimento do domínio do problema através da experimentação.

Isso indica que a escolha das ferramentas está, então, limitada para aquelas desenvolvidas para IBM PC ou compatível sob sistema operacional DOS. A TABELA IV.1 (FRENZEL, 1987) mostra as ferramentas mais conhecidas.

Observa-se que num projeto de um *Configurador* de processos é importante o uso de quadros para representar o conhecimento. Segundo a TABELA IV.1 existem poucas ferramentas com este enfoque, o que limita ainda mais a escolha.

** DOS é marca registrada da Microsoft.

TABELA IV.1: Ferramentas para desenvolver SE's em IBM PC.

Ferramenta	Publicação	Tipo
Expert Edge	Human Edge Software Inc.	Regras
Expert System	Potomac Pacific Engineering	Regras
EXSYS	Exsys, Inc.	Regras
INSIGHT	Level 5 Research	Regras
KES	Software A&E, Inc.	Regras
M.1	Tecknowledge	Regras
OPS5+	Artelligence	Regras
Personal Consultant	Texas Instruments	Regras
DAISY	Lithp Systems	Regras, Quadros e Rede semântica
CxPERT	Software Plus	Regras Quadros

Entretanto, com o uso dessas ferramentas o engenheiro do conhecimento terá menores opções de:

- projeto, limitado pela estrutura da ferramenta;
- organização e acesso ao conhecimento;
- inserção, no MI, de conhecimento relativo à solução do problema;
- além do treinamento no uso da ferramenta.

Pode-se acrescentar também que o custo para aquisição de uma dessas ferramentas é muito maior que o custo para adquirir uma linguagem dirigida para desenvolver sistemas com técnicas de IA.

Segundo GIORNO (1987), as linguagens dirigidas a IA implicam em um grande esforço de desenvolvimento, mas elas propiciam flexibilidade de projeto para o engenheiro de conhecimento e algumas estruturas básicas que ele pode utilizar para:

- representar conhecimento;
- desenvolver MI's para processar BC's;
- primitivas gráficas para desenvolver a IHM.

No caso de desenvolver uma nova ferramenta, então deve-se escolher uma linguagem adequada ao sistema. Deve-se preocupar com as formas de representar o conhecimento, com os MI's a serem utilizados e com as facilidades oferecidas para desenvolver a IHM.

Na fase de *Representação de Conhecimento* (seção IV.5) foi apresentada uma ferramenta para estruturar o conhecimento, a qual foi desenvolvida em Prolog. Na escolha de uma linguagem de desenvolvimento na área de IA tem-se várias opções, sendo as mais usadas:

- .**LISP** (LIST Processing), uma das linguagens de programação mais usada na área de IA criada na década de 1950 por John Mc Carthy no MIT (WINSTON et alia, 1984);
- .**Prolog** (PROgramming in LOGic), uma linguagem de programação muito popular na área de IA, desenvolvida na França e baseada nos conceitos do cálculo de predicados (CLOCKSIN e MELLISH, 1983);
- .**Smalltalk**, uma linguagem desenvolvida pelo "Learning Research Group" de Xerox PARC, onde a programação é vista como uma coleção de objetos que se comunicam com outros pela troca de mensagens (HALBERT et alia, 1987).

As características principais da linguagem Prolog são:

- orientada para processamento simbólico;
- permite recuperação dedutiva de informação;
- apresenta uma semântica declarativa inerente à lógica em adição à semântica operacional usual das linguagens de programação tradicionais;
- permite a definição de programas invertíveis, ou seja, programas que não distinguem entre argumentos de entrada e saída. Como consequência, permite a definição de programas com mais de uma finalidade, que podem ser chamados com formas diferentes de entrada/saída;
- permite a obtenção de respostas alternativas;
- suporta definições recursivas para descrição de processos e problemas, dispensando os mecanismos tradicionais de controle, como "goto" e "do while";
- permite descrever a especificação e a codificação de programas em uma mesma linguagem;
- representa programas e dados através do mesmo formalismo (cláusulas).

Prolog soluciona o problema executando uma busca em profundidade ("Depth-First") no espaço do problema. Faz uma busca automática para trás ("backtracking"), mas o programador pode controlar a extensão da busca por meio do uso do operador de corte ("cut"). Resumindo, a linguagem Prolog proporciona muitas características do MI de um SE, tais como casamento de padrões, busca em profundidade e encadeamento para trás (LEE, 1986).

Existem muitas implementações da linguagem Prolog para IBM PC. Para a escolha de uma implementação da linguagem Prolog adequada às necessidades de um projeto, recomenda-se o trabalho de WONG (1985), que mostra as vantagens e desvantagens das diferentes implementações da

linguagem, tais como Micro-Prolog, Arity, Prolog V, Turbo Prolog e Prolog ADA.

LISP é, simultaneamente, uma linguagem matemática formal e uma linguagem interativa de programação, para representação e manipulação de dados numéricos e principalmente simbólicos (como palavras e sentenças). LISP reconhece expressões simbólicas de complexidade arbitrária e permite dinamicamente descartar expressões existentes e construir expressões adicionais. Ou seja, LISP representa e manipula programas e dados como expressões simbólicas, uma característica única e poderosa desta linguagem. Isto significa que programas em linguagens quaisquer (inclusive em LISP), entendidos como símbolos, podem ser tratados como dados, e vice-versa: dados podem ser tratados como programas. Deste modo, pela separação própria de contexto, LISP permite que programas produzam outros programas ou modifiquem a si mesmos, podendo aprender e alterar seu comportamento com a experiência.

LISP e Prolog são as linguagens de programação mais populares na área de IA. Mas, a velocidade de execução e a quantidade de memória demandada pelos programas desenvolvidos nestas linguagens são os maiores problemas encontrados pelos pesquisadores da área (FALK, 1988).

IV.9 Teste de Desempenho

Muitas definições de SE's mencionam sua habilidade de executar tarefas a níveis muito próximos do especialista humano. Entretanto a validação dos SE's - isto é, testar sistemas para verificar o quanto está executando a um nível aceitável de rendimento - tem sido "ad hoc" (com algumas exceções), informal e de valores duvidosos.

Tipicamente, os engenheiros têm verificado o rendimento dos SE's executando testes de casos e comparando os resultados com as opiniões dos especialistas. Estes

engenheiros calculam a porcentagem de sucessos do sistema e usam critérios subjetivos para analisar e explicar as falhas. Exemplos deste tipo de verificação são o MYCIN (BUCHANAN et alia, 1984) e o sistema Emerge (sistema de diagnóstico de doenças do torax).

Ainda que estes simples esquemas apresentem muitos problemas, tem-se outros casos nos quais se compara o SE com o especialista do qual foram obtidos os conhecimentos, como foi o caso do Prospector, o que indica uma verificação muito duvidosa.

A verificação quase sempre tem sido feita qualitativamente, como em Internist-I. Enquanto os SE's a serem usados em bases regulares - particularmente em áreas críticas - devem ser validados cuidadosamente, o que motivou o desenvolvimento de métodos de verificação mais formais. O sistema de configuração do VAX, conhecido como R1/XCon, tem alguma verificação formal.

Os seguintes problemas aparecem na verificação do rendimento dos SE:

- i. O que verificar ?
- ii. Com que verificar ?
- iii. Quando verificar ?
- iv. Como controlar o custo da verificação ?

A questão mais importante é a iii. Por exemplo, com sistemas não-críticos, as aplicações como o R1/XCon podem ser verificadas no campo de operação enquanto é usado. Nas aplicações críticas onde a vida está em jogo ou grandes fortunas correm risco, o teste no campo de aplicação é usualmente impossível de ser realizado (embora sempre seja possível executar um SE em paralelo com o sistema tradicional).

Pelos motivos acima têm sido desenvolvidas teorias para verificação dos SE's antes de serem usados no campo de

aplicação. Existem teorias formais para combinar as opiniões dos especialistas, que é um problema de alto interesse e de considerável dificuldade. Considera-se a formulação do problema no qual cada especialista estima a probabilidade de certos eventos, e o objetivo é produzir uma distribuição de probabilidade que resuma as variadas estimações. O estudo de tais procedimentos para fazer um acordo é conhecido como teoria de consenso.

O acordo das opiniões dos especialistas é um problema básico no processo de tomada de decisões. Um modelo deste processo mostra eventos de interesse como eventos em um espaço de probabilidades, e cada opinião de um especialista consiste em dar uma probabilidade para cada evento. O problema do consenso consiste em formar uma única distribuição de probabilidade fornecidas pelos especialistas para um conjunto de eventos. A teoria do consenso é usada para combinar as opiniões, a qual é aplicada em conjunto com técnicas relacionadas, tais como teoria da evidência, que foi proposta para o seu uso em SE's.

No desenvolvimento de um SE em *Configuração* (como XCon) não é necessário aplicar teorias desse tipo, uma vez que é sempre possível testar o sistema no campo de aplicação. Com isso tudo, a fase de verificação do sistema é simples. Devem ser executados alguns exemplos bem definidos e analisado o comportamento do sistema.

CAPÍTULO V

IMPLEMENTAÇÃO DA METODOLOGIA

A metodologia de solução proposta no CAPÍTULO IV foi resultado da pesquisa feita para ser aplicada ao projeto ATE na fase de *Configuração* de EC's. Esta implementação é apresentada neste capítulo, detalhando os passos seguidos no desenvolvimento do primeiro protótipo do SE.

V.1 Estudo

Como foi dito no CAPÍTULO IV, o objetivo desta atividade é apresentar um relatório técnico que identifique o problema, para determinar se o projeto é viável, ou não. Alguns itens que compõem este relatório foram detalhados nos CAPÍTULOS I e II, mas a seguir é apresentado um resumo:

V.1.1 Descrição do Problema

A fase de *Operação* da ATE pode ser considerada como um "software" que lê um programa. Este programa de entrada é um código que contém todas as operações para controlar um EC. Na *Operação*, então, este código é traduzido em instruções que são executadas em baixo nível. Uma vez que os EC's a serem executados são similares na forma, mas diferentes no conteúdo (eventualmente diferentes na forma também), é necessário gerar um código do programa de controle do processo para cada EC, acarretando um custo elevado. Daí a necessidade de desenvolver um sistema *Configurador* de EC's, o qual deve gerar como saída o programa de controle do EC.

Um dos objetivos da fase de *Configuração* do projeto de automação visa diminuir o tempo de configuração de um

EC. Com este sistema *Configurador* pode-se gerar automaticamente a BD, onde se encontra a configuração do processo sem erros, dando uma maior confiabilidade ao sistema de automação na fase de *Operação*.

A figura V.1 ilustra a relação entre o *Configurador*, o qual gera uma BD (Agenda.dba) contendo o código que será executado, e a *Operação* do processo.

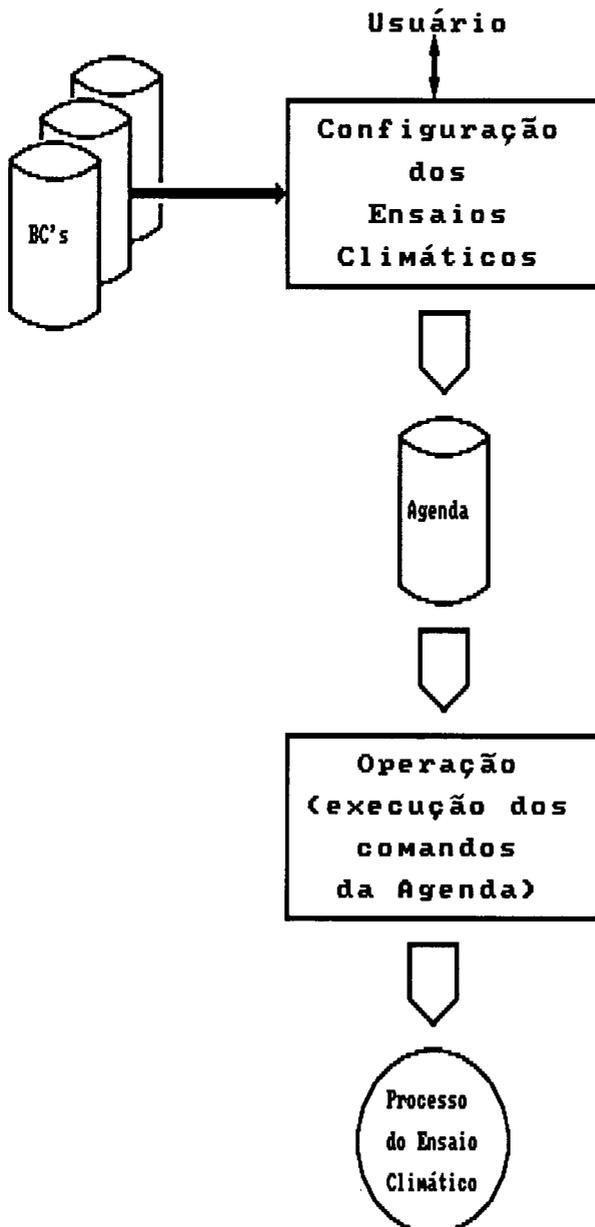


FIGURA U.1: Relação entre as fases de Operação e Configuração.

V.1.2 Necessidade de aplicar Técnicas de IA

O conhecimento envolvido neste sistema *Configurador* corresponde essencialmente à seleção de normas de especificação de EC's, seleção de equipamentos condicionadores climáticos que cumpram as exigências das normas selecionadas e seleção dos instrumentos que permitam a aquisição de dados do andamento do processo na fase de *Operação*. Estes conhecimentos são muito restritos dentro de um domínio puramente técnico. A necessidade de se aplicar técnicas de IA deve-se ao fato de que o que se quer obter como resultado é um *Configurador* inteligente, que auxilie o operador na configuração dos EC's. Isto é, o sistema deve dar alternativas de solução ao operador e explicar as decisões por ele tomadas na seleção de normas, equipamentos e instrumentos, como também gerar automaticamente, a partir dos parâmetros do EC, a configuração apropriada para estes equipamentos e instrumentos.

V.1.3 Definição do Domínio e da BC inicial

O domínio envolvido neste problema pode ser dividido em quatro subconjuntos:

- Normas Técnicas de Especificação dos EC's;
- Características de funcionamento dos equipamentos condicionadores climáticos;
- Características de funcionamento dos instrumentos de medição;
- Características de funcionamento da amostra que se deseja testar.

Os EC's são classificados conforme sua natureza como: calor sêco, frio, calor úmido prolongado e calor úmido acelerado.

Em geral, para a avaliação do comportamento de uma amostra (objeto do teste), é necessário estabelecer uma

sequência de EC's, que são combinados entre si de forma a atingir o grau de rigor especificado. Segue, então, o detalhamento dos parâmetros avaliados em cada um dos tipos de EC's:

Ensaio a Calor Seco: Objetivam determinar se a amostra é adequada para ser utilizada ou armazenada a uma alta temperatura e observar os efeitos produzidos sobre ela;

Ensaio Frios: Objetivam determinar se uma amostra é adequada para ser utilizada a uma baixa temperatura e observar os efeitos produzidos sobre ela;

Ensaio a Calor Úmido Prolongado: Objetivam determinar a capacidade das amostras serem usadas ou armazenadas sob condições de alta umidade relativa, observando-se os efeitos a temperaturas constantes por um período de tempo pré-determinado;

Ensaio a Calor Úmido Acelerado: Objetivam determinar se uma amostra é adequada para ser utilizada ou armazenada sob condições de uma alta umidade relativa, quando combinada com grandes variações de temperatura.

Na versão atual deste sistema ATE em desenvolvimento tem-se como equipamento condicionador climático uma estufa, que permite somente a realização dos Ensaio a Calor Seco. Portanto, no domínio da BC do primeiro protótipo devem ser consideradas as Normas Técnicas de especificação deste tipo de ensaios. O domínio do conhecimento envolvido com os equipamentos é, então, limitado a uma estufa.

Atualmente o sistema possui três instrumentos de medição: um osciloscópio, um multímetro e um sintetizador. Portanto, estes instrumentos devem estar presentes no escopo do domínio do primeiro protótipo.

V.1.4 Custos e Benefícios

O custo de um sistema deste tipo pode ser considerado baixo, dado que o sistema será desenvolvido para um microcomputador compatível com IBM PC, para obter uma solução econômica. Foram usadas ferramentas desenvolvidas pelo CEPTEL, tais como o NSO para permitir um ambiente em tempo real, o qual é importante na fase de *Operação*; e outras ferramentas de baixo custo como Turbo Prolog.

Os benefícios do sistema *Configurador* são grandes, considerando, entre outros, a poupança de tempo para o especialista em configurar um EC e a simplicidade para realizá-lo, gerando desta forma um código de controle mais confiável.

V.1.5 Objetivos Gerais

Um dos objetivos do projeto ATE é desenvolver um protótipo de um SE para auxiliar na tarefa de *Configuração* de EC's para calor seco, onde, por enquanto, existem os três instrumentos acima mencionados e uma estufa como equipamento condicionador climático.

V.1.6 Plano de Desenvolvimento

O plano de desenvolvimento é a aplicação da metodologia proposta no CAPÍTULO IV, seguindo os passos indicados. Isto é, iniciar pela tarefa de *Aquisição do Conhecimento*, para obter a informação necessária para o projeto todo. Logo após fazer a *Análise do "Hardware"*, deve-se *Representar o Conhecimento*, tarefa com a qual se tem a informação necessária para iniciar o *Projeto* em si. Depois do *Projeto* deve-se *Implementar* os módulos especificados e a *IHM*, para, no fim, fazer um *Teste do Desempenho* do primeiro protótipo.

V.2 Aquisição do Conhecimento

Para começar a *Aquisição do Conhecimento* acompanhamos o especialista em EC's nas atividades e nas rotinas de trabalho do Laboratório de Ensaio, visando com isso assimilar conceitos para a elaboração da especificação do projeto.

Como primeiro objetivo deve ser determinada a saída do sistema que se deseja desenvolver. Isto é, a BD com a *configuração* de um EC, que é interesse do usuário.

V.2.1 Saída do Sistema Configurador

A BD, denominada Agenda.dba, contém as operações envolvidas na configuração de um EC no seguinte esquema:

```
agenda ([[NúmeroDaOperação]],[[Operação],[Parâmetros]]])
```

O *NúmeroDaOperação* corresponde ao número correlativo da operação que se deve executar. A *Operação* é o nome da operação que se deseja realizar, onde cada operação tem certos *Parâmetros* necessários para configurar um equipamento condicionador climático ou um instrumento para adquirir dados. Essa *Operação* com os seus respectivos *Parâmetros* pode ser um dos seguintes tipos (a relação dos comandos aparece no APÊNDICE A):

Operações com equipamento condicionador climático:

- **Atingir:** configura um equipamento para atingir uma certa temperatura;
- **Manter:** configura um equipamento para manter uma certa temperatura;
- **Início:** início da operação no equipamento indicado;
- **Fim:** fim da operação no equipamento indicado;
- **Status:** operação de status no equipamento indicado.

Operações com os instrumentos:

- **SEtar:** indica que será enviada uma série de comandos iniciais específicos para o instrumento indicado;
- **Medida:** solicita a realização de uma medida no instrumento indicado;
- **Clear:** apaga os dados armazenados no instrumento indicado;
- **Status:** verifica "byte" de estado do instrumento indicado.

Outras operações:

- **Inspeção Visual:** realizar uma inspeção visual no sistema;
- **Ligar:** ligar algum instrumento ou equipamento.

V.2.1.1 Restrições dos Parâmetros

Cada *Operação* possui os parâmetros necessários correspondentes à operação a realizar, sendo que, para serem validados, estes parâmetros devem cumprir certas restrições. Vejamos as restrições de cada parâmetro:

equipamento: pode ter um único valor (na versão atual do sistema em funcionamento), que corresponde à estufa;

operação: deve indicar uma das operações válidas para um equipamento climático, que são: **Configuração**, **Início**, **Fim** ou **Status**;

ciclo: deve ser um número entre 0 e 255;

temperatura: um número que deve estar entre -50 e 200 °C;

tolerância: um número entre 0 e 50 °C;

duração: é um número entre 1 e 32767 que indica a duração da operação em minutos;

tempo máximo: é um número entre 1 e 32767 que indica o tempo máximo da operação em minutos;

- monitoração:** é um número entre 4 e 32767 que indica os segundos transcorridos entre as medidas do equipamento climático;
- instrumento:** deve indicar um dos instrumentos do sistema, podendo ser (na versão atual do sistema em funcionamento): Osciloscópio, Multímetro ou Sintetizador;
- setup:** é uma concatenação dos comandos válidos para a configuração de cada instrumento;
- tipo de medida:** deve indicar um tipo de medida válido para o instrumento em questão. As medidas válidas para o Osciloscópio são: Valor RMS da Tensão, Frequência e Período. As medidas válidas para o Multímetro são: Tensão (AC, DC e AC+DC), Corrente (AC, DC e AC+DC), Resistência ôhmica, Período e Frequência;
- tempo:** é um número entre 1 e 32767 que indica a duração da operação em minutos;
- período:** é um número entre 4 e 32767 que indica os segundos transcorridos entre as medidas no instrumento indicado;
- canal:** é um número entre 0 e 9 que indica o canal através do qual será feita a medida. O Osciloscópio tem dois canais, enquanto o Multímetro possui nove canais.

V.2.1.2 Codificação dos Termos de Configuração

As operações na BD Agenda.dba estão codificadas para diminuir o espaço requerido na memória pelo programa.

No APÊNDICE A aparecem todos os códigos que identificam cada termo (Operação, parâmetros e outros).

As figuras do APÊNDICE B mostram a sintaxe das operações que podem ser formadas.

V.2.2 Definição do Domínio da BC

A seguir é mostrado o domínio do conhecimento que o especialista aplica na solução do problema para obter a saída, que deve estar de acordo com as normas de especificação.

V.2.2.1 Normas de Especificação

Um ensaio é constituído de uma série completa de operações que objetivam avaliar o comportamento de um componente ou equipamento eletrônico, também usualmente chamado de amostra, em condições de uso similares às que estarão sujeitos na prática. Os ensaios são elaborados para fornecer informações sobre as propriedades das amostras, avaliar seu comportamento sob limites de temperatura, umidade e pressão, e sua capacidade de resistir à armazenagem e transporte. Para sua concepção são estudadas normas (ABNT, 1981), que determinam o grau de rigor e as operações a serem cumpridas, que consistem normalmente nas seguintes etapas (figura V.2):

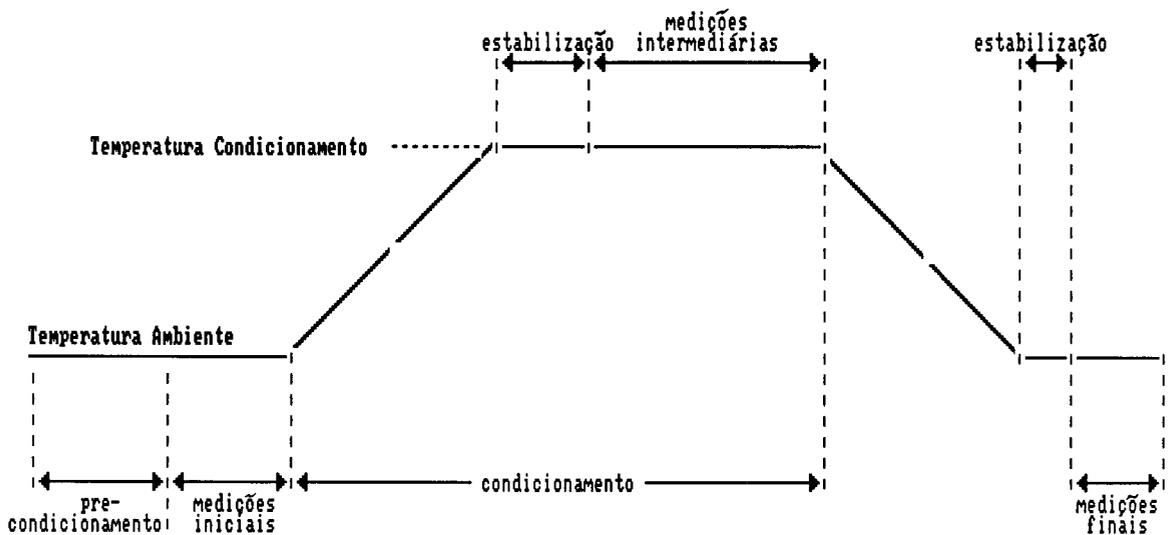


FIGURA U.2: Etapas de um EC.

- **Pré-Condicionamento:** é o tratamento de uma amostra com a finalidade de remover ou neutralizar em parte os efeitos da sua vida anterior. Quando necessária esta é a primeira operação do ensaio;
- **Medições Iniciais:** são aquelas efetuadas para determinar as condições iniciais da amostra. Nesta fase a amostra é inspecionada visualmente, submetida a medições e verificações mecânicas como exigido pela especificação;
- **Condicionamento:** é a exposição de uma amostra a uma determinada condição climática, para determinar seus efeitos sobre esta. A fim de realizar o condicionamento climático, o equipamento que gera as condições do teste (câmara) deverá ser capaz de manter, em qualquer região, uma temperatura específica com uma tolerância determinada. Os procedimentos para a condução desta fase variam com o tipo de avaliação que se deseja fazer: EC para armazenagem e transporte ou EC para uso e operação, da seguinte forma:

Condicionamento para armazenagem e transporte: a amostra, desembalada e desligada, é introduzida na câmara à temperatura ambiente; a partir daí, a temperatura deve ser ajustada para o valor especificado, sendo a taxa de subida ou descida da temperatura calibrada de forma a não exceder $1^{\circ}\text{C}/\text{min}$. Após ser alcançado o valor desejado da temperatura, este deverá ser mantido constante por um período mínimo de tempo especificado, durante o qual a amostra deve permanecer desligada para a avaliação das suas condições de armazenamento e transporte;

Condicionamento para operação: o mesmo que o anterior, exceto que a amostra deve ser ligada, e assim permanecer, submetida a medições e verificações mecânicas para a avaliação das suas condições de funcionamento;

- **Estabilização:** é o tratamento da amostra após o condicionamento, a fim de que as propriedades da amostra voltem a uma estabilização antes das medições finais. Esta fase se inicia após o término do período de condicionamento, quando a câmara é desligada. A amostra deve permanecer dentro da câmara até que a temperatura atinja o valor da temperatura ambiente;

- **Medições finais:** a amostra deve ser inspecionada visualmente e submetida a medições.

V.2.2.2 Características dos Instrumentos

Os instrumentos ligados ao sistema na fase de *Operação* cumprem a função de gerar os sinais e monitorar o comportamento da amostra. Os três instrumentos usados são: Osciloscópio HP 54200AD (HP, 1985), Sintetizador HP 3325A (HP, 1981) e Multímetro HP 3457A (HP, 1986).

Osciloscópio HP-54200AD

Através deste instrumento pode-se fazer a aquisição das respostas geradas pelos circuitos sob teste (amostra), sendo as mais requisitadas: Frequência, Período e Valor RMS de Tensão. É possível selecionar uma determinada forma de onda apresentada na tela do Osciloscópio e guardá-la, codificada digitalmente em um arquivo.

As características mais importantes e mais usadas do Osciloscópio são (os parâmetros são mostrados no APÊNDICE C):

"CHANnel" (Canal): o Osciloscópio possui dois canais (1,2). Em cada canal deve-se indicar as características de tensão do sinal que se deseja ver;

"TIME base" (Base de Tempo): corresponde à base de tempo do Osciloscópio, onde deve-se determinar o modo de varredura do sinal ("Sweep Mode"), o escopo do sinal ("Range"), o atraso para o sincronismo ("Delay") e a escala do sincronismo ("Scale");

"TRIGger" (Sincronismo): corresponde ao sincronismo do sinal. Escolhe-se a fonte do sincronismo e as características do sinal de sincronismo.

Multímetro HP-3457

O Multímetro, assim como o Osciloscópio, é utilizado para fazer a aquisição das respostas geradas pelos circuitos sob teste da amostra. O Multímetro permite medir: Frequência, Período, Resistência ôhmica, Corrente (AC, DC e AC+DC) e Tensão (AC, DC e AC+DC). Cada vez que se deseja fazer uma aquisição de dados deve-se configurar adequadamente o Multímetro, segundo o seguinte esquema:

Medida, Escopo, Precisão;

onde *Medida* pode ter um dos seguintes valores: **FREQUência**, **PERíodo**, **OHM** ou **CURrent**;

o *Escopo* determina os limites da medida que se deseja fazer e a *Precisão* indica os dígitos decimais da medição.

Sintetizador HP 3325A

O Sintetizador tem a função de gerar sinais elétricos padronizados para excitar os circuitos eletrônicos sob teste. Estes sinais, em geral, simulam os estímulos a que a amostra será submetida em funcionamento normal ou condições limites, sendo que a resposta a estes

estímulos pode ser obtida através do Osciloscópio e do Multímetro, conforme mencionado anteriormente.

As características que devem ser determinadas na configuração deste instrumento são:

"Function": tipo de sinal que se deseja gerar, podendo ser do tipo quadrado, triangular ou senoidal. O código é: "F(um dígito)".

"FReQuency": é a frequência do sinal que se deseja gerar. O código é: "FRQ (11 dígitos) (HZ/KH/MH)".

"AMPlitude": é a amplitude do sinal que se deseja gerar. O código é: "AMP (4 dígitos) (V/MV/VR)".

"DC Offset": indica o offset do sinal, cujo código é: "OF (4 dígitos) (VO/MV)".

"Output Enabled": uma vez que se deseja que o Multímetro apresente o sinal na saída, deve-se enviar o código de "OE", habilitando sua saída.

V.2.2.3 Características dos Equipamentos

Os equipamentos ligados ao sistema cumprem a função de atingir, manter e controlar as temperaturas desejadas. Os equipamentos que irão compor o sistema como um todo são: Câmara Climática Tenney BTH 0200 (CC I), Câmara Climática Tenney T14RS (CC II) e Estufa Fanem 320 SE.

Estes equipamentos serão ligados ao sistema através do Terminal de Aquisição e Controle (TAC), que pode controlar simultaneamente vários equipamentos sem a intervenção do microcomputador. O TAC está ligado ao microcomputador central para receber as configurações de cada etapa do processo via interface serial RS-232. Atualmente o domínio destes equipamentos se restringe à Estufa Fanem 320 SE, por ser o único equipamento integrado ao sistema nesta primeira versão implementada.

Estufa Fanem 320 SE

Este equipamento possui um mecanismo autônomo de controle para temperatura, que pode variar desde a temperatura ambiente (geralmente 25 °C) até 120 °C, com uma taxa de variação positiva de temperatura padronizada de 1 °C/min. A taxa de variação negativa da temperatura é variável e depende tanto da temperatura da Estufa, quanto da temperatura que se deseja atingir.

V.2.3 Especificação Estruturada do Projeto

No processo utilizado na configuração dos EC's, foram identificadas basicamente quatro fases (figura V.3):

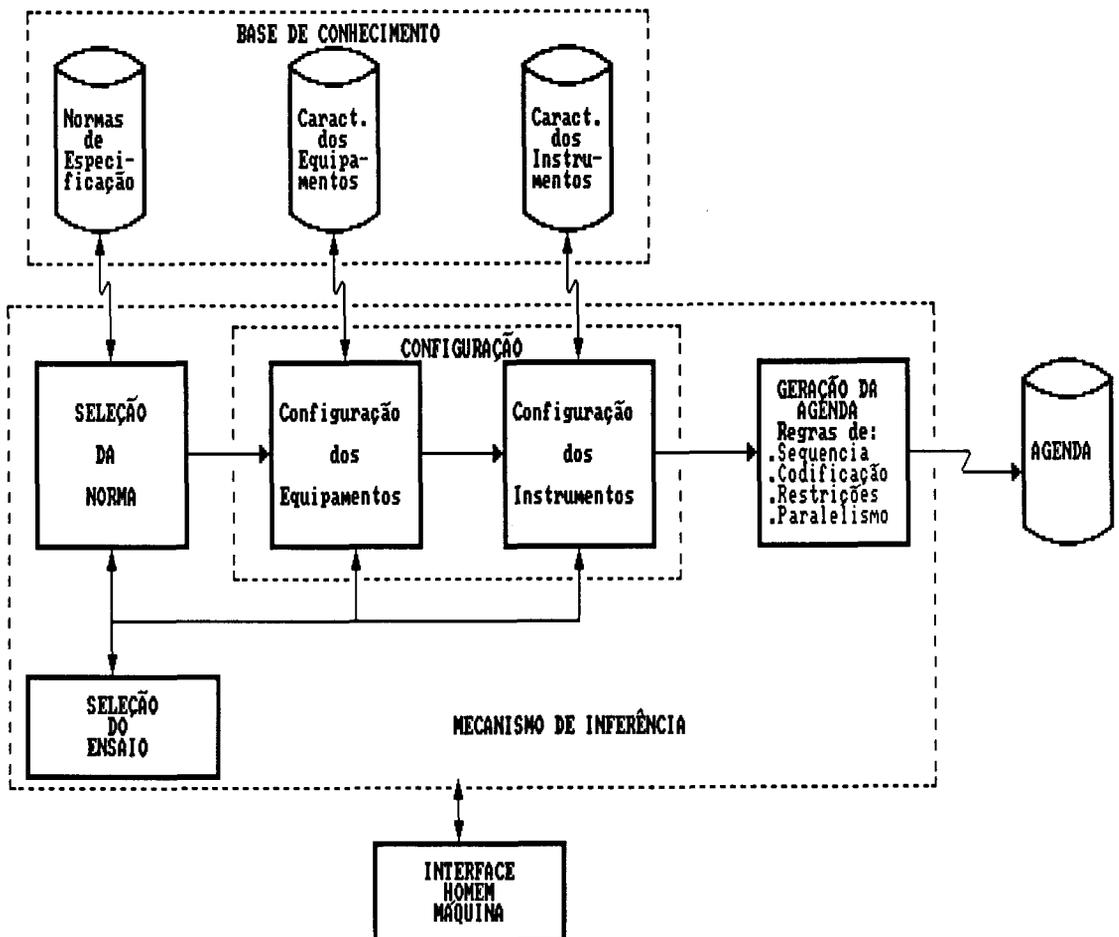


FIGURA V.3: Esquema do Software do Configurador.

Seleção do Ensaio: nesta fase escolhe-se qual vai ser a norma geral, que contém as características que devem ser determinadas para selecionar a norma correspondente ao ensaio desejado;

Seleção da Norma: onde é selecionada a norma específica que deve ser aplicada ao ensaio que se deseja executar;

Início da Configuração: onde são obtidos todos os parâmetros indicados na norma específica do ensaio;

Geração da Agenda: uma vez que o ensaio está completamente definido, então é automaticamente gerada a BD, que contém as operações que devem ser executadas na fase de *Operação*.

Na Seleção do Ensaio deve-se escolher qual é o ensaio que se deseja configurar. O protótipo inicial permite somente uma opção, que é o ensaio de calor seco.

Os EC's possuem uma norma de Generalidades. No caso do ensaio de calor seco, a norma correspondente é 6817 (ABNT, 1981). Quando selecionado o ensaio de calor seco, pode-se começar a busca pelas características do ensaio que se deseja, para fazer a Seleção da Norma específica do ensaio. A figura V.4 mostra o processo de seleção da norma de especificação.

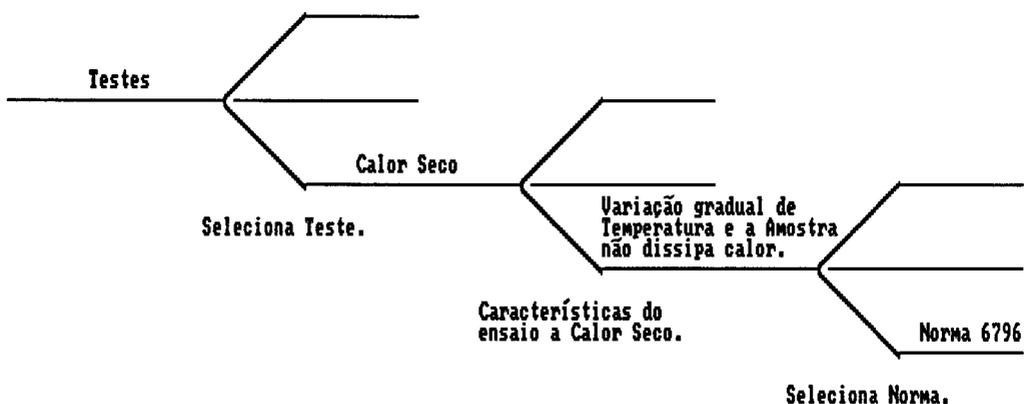


FIGURA V.4: Seleção do Quadro dentro da Hierarquia.

No caso da norma 6817 existem duas características que determinam o tipo do ensaio: a variação da temperatura do ensaio e a dissipação de calor da amostra. A combinação destas características geram quatro possíveis ensaios a serem efetuados, os quais são especificados pelas normas a seguir:

Norma 6819: Ensaio de Calor Seco com **variação rápida** de temperatura para amostras que **não dissipam** calor;

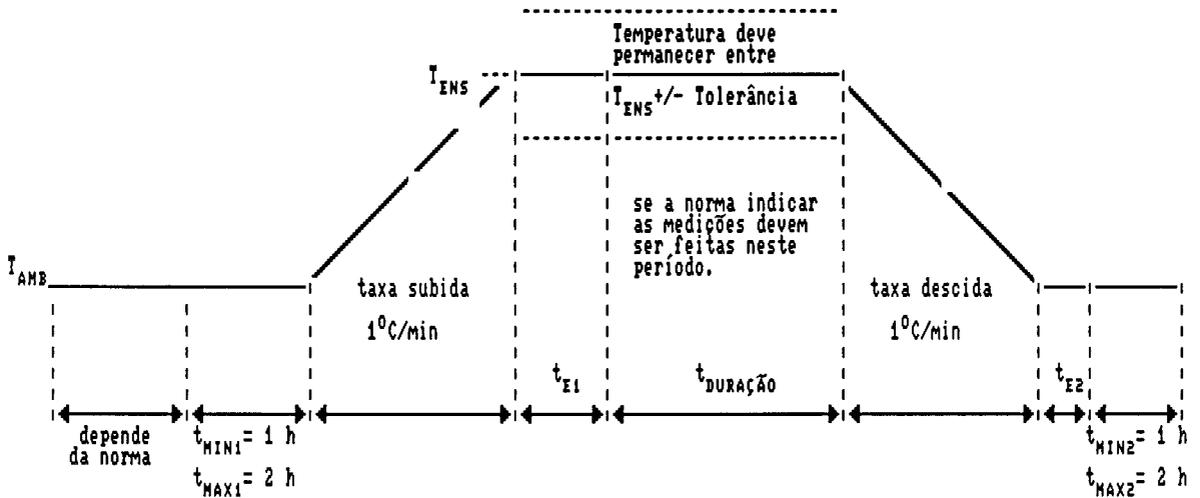
Norma 6796: Ensaio de Calor Seco com **variação gradual** de temperatura para amostras que **não dissipam** calor;

Norma 6797: Ensaio de Calor Seco com **variação rápida** de temperatura para amostras que **dissipam** calor;

Norma 6796: Ensaio de Calor Seco com **variação gradual** de temperatura para amostras que **dissipam** calor.

Uma vez especificadas as características do ensaio de calor seco, então é possível iniciar a Configuração deste. A Configuração propriamente dita consiste em determinar os parâmetros do EC mostrados na figura V.5.

Uma vez configurado completamente o ensaio, gera-se automaticamente a Agenda de comandos, a qual será usada na fase de *Operação* de EC's. O processo de geração da Agenda deve cumprir as regras de codificação antes mencionadas, como também algumas regras de paralelismo e sequencialismo dos comandos, mostrados na seção V.2.5.



T_{ENS} = Temperatura do Ensaio, podendo ser uma das seguintes (30,40,55,70,85,100,125,155,175,200) °C.

T_{AMB} = Temperatura Ambiente, geralmente na faixa de 23 a 30 °C.

t_{MIN} = Tempo Mínimo, por "default" e 1 hora.

t_{MAX} = Tempo Máximo, por "default" e 2 horas.

t_E = Tempo de Estabilização, que normalmente depende da amostra.

$t_{DURAÇÃO}$ = tempo de exposição, podendo ser (4,16,72,96) horas.

Tolerância = por "default" e 2 °C.

FIGURA U.5: Variáveis envolvidas no EC de calor seco.

V.2.4 Especificação da IHM

Das entrevistas com os especialistas conseguiu-se obter uma metodologia de trabalho para configurar um teste de EC. Os parágrafos seguintes mostram a especificação da IHM, onde são essenciais alguns dados relativos aos ensaios que se deseja configurar.

Para a especificação da IHM foram definidos alguns conceitos de relativa importância para permitir uma comunicação correta entre o sistema e o usuário. Sendo assim, o EC foi dividido em etapas, que são organizadas conforme indicado na figura V.6, e são definidas da seguinte forma:

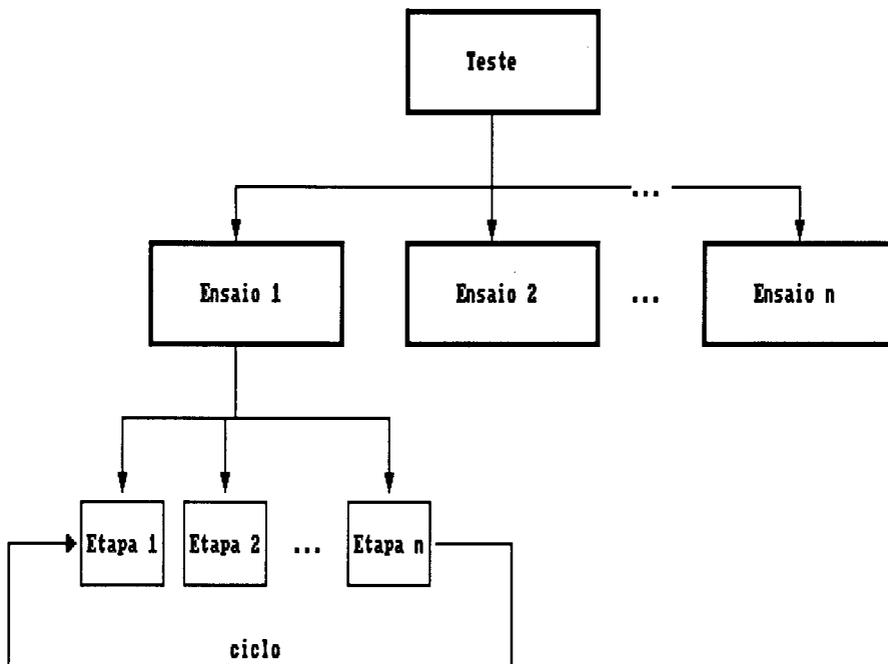


FIGURA V.6: Conceitos básicos dos Testes.

- **Teste:** é um conjunto de ensaios;
- **Ensaio:** é o tipo de prova a que se deseja submeter a amostra. É constituído de diversas etapas;
- **Etapa:** é uma parte do ensaio. Possui características próprias, e, assim sendo, deve ser configurada uma após outra;
- **Ciclo:** é o número de vezes que se repete o mesmo ensaio;
- **Fase:** é o conjunto de passos necessários para a realização de um teste. Para o sistema de testes climáticos automatizado foram identificadas três fases distintas, denominadas: *Pré-Operação*, que é a fase de *Configuração*; *Operação*, que é o controle do processo e *Pós-Operação* que é a fase de *Avaliação*.

Dados de Entrada e Dados de Saída

Segundo o processo de configuração identificado na seção V.2.3, o usuário do sistema deve entrar com as características do EC que deseja configurar para que o *Configurador* possa selecionar a norma adequada. Uma vez que o sistema selecionou a norma, o usuário deve escolher a variação de ensaio que deseja (sendo que todas as normas possuem variações, dependendo do objetivo). Outros dados que o usuário deve fornecer ao sistema são os parâmetros do EC, isto é, temperaturas limites de operação e duração das diferentes etapas (figura V.5).

Uma vez configurado o processo com os parâmetros da operação, devem-se configurar os instrumentos de medição e geração de sinais, com todos estes dados que devem ser fornecidos pelo usuário. Para isso é necessário dispor de recursos para:

- .selecionar um dado de uma lista de opções;
- .validar um dado que tem limite mínimo e limite máximo, não permitindo sair desse escopo;
- .mostrar os valores "default" de uma variável, quando existir este valor.

V.2.5 Necessidade do "Hardware"

Na fase de *Configuração* o "hardware" requerido é mínimo, mas para a fase de *Operação* é necessário "hardware" adicional, o qual permite controlar vários equipamentos simultaneamente e permite obter as medições dos instrumentos de forma independente ao controle do processo. Isto quer dizer que, durante a configuração dos EC's, deve-se distinguir quais operações podem ser realizadas em paralelo e quais devem ser executadas de forma sequencial.

Sequencialidade e Paralelismo

Os comandos podem ser executados sequencialmente ou em paralelo, dependendo da situação. Por exemplo, no caso de uma operação com a estufa: primeiro ela deve ser Configurada, e, logo em seguida, deve-se indicar Início de operação. Quando a estufa está em funcionamento pode-se consultar o seu Status. A operação de Status tem uma duração igual à última etapa configurada, diferentemente das operações de Configuração, Início e Fim, que têm uma unidade de tempo de duração, isto é, a operação é lida, e, logo após, são enviadas as instruções correspondentes à estufa. Após o pedido de Status deve ser enviada uma instrução de Fim da operação à estufa, para parar o seu funcionamento. Esta seqüência pode ser feita parcialmente em paralelo, de várias formas:

```
Sequencial:  1 --> Configuração
              2 --> Início
              3 --> Status
              4 --> Fim
              5 --> outro comando.
```

Esta é a forma mais simples de executar todos os comandos em forma sequencial, mas não aproveita o paralelismo permitido pelo "hardware". Ou seja, primeiro lê o comando 1, envia a Configuração para a estufa. Depois lê o comando 2, envia o comando de Início para a estufa. Logo

lê o comando 3, este comando é um comando de Status na estufa, sendo que a duração deste comando é o tempo enviado para a estufa na última configuração. Quando o tempo se esgotar o comando 3 termina, e, então, lê o comando 4, o qual envia a operação de Fim para a estufa, passando para o outro comando.

```
Paralelo 1:  1 --> ((Configuração) (Início))
             2 --> Status
             3 --> Fim
             4 --> outro comando.
```

Esta forma executa dois dos comandos em forma paralela, aproveitando algum paralelismo permitido pelo "hardware". Primeiro lê o comando 1 (detectando um comando paralelo), envia primeiro a Configuração para a estufa, imediatamente depois envia o comando de Início para a estufa. Logo lê o comando 2, que é um comando de Status na estufa, tendo a duração do tempo enviado para a estufa na última configuração. Quando o tempo se esgotar, então lê o comando 3, o qual envia a operação de Fim para a estufa, passando para o outro comando.

```
Paralelo 2:  1 --> Configuração
             2 --> ((Início) (Status))
             3 --> Fim
             4 --> outro comando.
```

Outra forma de executar dois dos comandos em forma paralela, aproveitando algum paralelismo permitido pelo "hardware". Primeiro lê o comando 1, envia a configuração para a estufa. Logo lê o comando 2 (detectando um comando paralelo), envia primeiro o comando de Início para a estufa, e logo após começa a operação de Status na estufa, tendo a duração do tempo enviado para a estufa na última configuração. Quando o tempo se esgotar, então lê o comando 3, o qual envia a operação de Fim para a estufa, passando para o outro comando.

```
Paralelo 3:  1 --> ((Configuração) (Início) (Status))
             2 --> Fim
             3 --> outro comando.
```

Esta é a forma de executar três dos comandos em forma paralela, aproveitando todo o paralelismo permitido pelo "hardware". Primeiro lê o comando 1 (detectando um comando paralelo), então envia primeiro a Configuração para a estufa, logo envia o comando de Início para a estufa, e logo após começa a operação de Status na estufa, tendo a duração do tempo enviado para a estufa nesta última configuração. Quando o tempo se esgotar, então lê o comando 2, o qual envia a operação de Fim para a estufa, passando para o outro comando.

O paralelismo é muito importante no caso de se ter que fazer uma medida em um instrumento, e, ao mesmo tempo, fazer uma consulta através do status do processo. Isto é simples de resolver colocando em um comando só as duas operações a serem feitas, da seguinte forma:

```
1 --> ((Medir) (Status))
```

Dado que a operação de Status tem uma duração que depende da última configuração, então esse comando vai ser executado durante todo esse tempo. Enquanto a operação de Medir tem explicitamente (como um parâmetro dela) a duração do tempo da medição, ela vai ser executada até se esgotar o tempo de medição. Qualquer uma das duas cujo tempo previsto terminar primeiro não é mais executada, permanecendo ativa a outra até se esgotar o tempo das duas, então é lido o comando seguinte.

V.3 Análise do "Hardware"

Com a informação da *Aquisição do Conhecimento*, obtém-se como conclusão que, na configuração dos EC's, deve-se levar em consideração o "hardware" necessário na fase de *Operação*, o qual foi mostrado na figura II.1.

V.4 Representação do Conhecimento

A chave do bom rendimento na eficiência de qualquer sistema é a *Representação do Conhecimento*. Este conhecimento consiste em dados (fatos) e MI's para a busca da solução apropriada na *configuração* do processo. Nas seções a seguir é detalhada a forma de representar o conhecimento envolvido no *Configurador*.

V.4.1 BC Lógica

A Agenda é a BD que vai ser preenchida pelo sistema em forma "off-line". Esta BD possui a informação necessária para configurar o processo, para gerenciar o processo durante a *operação* e para ter um acesso rápido ao "status" do sistema em qualquer tempo, ou seja, antes, durante ou depois de efetuar os testes.

Feita uma análise do conhecimento que deve estar presente na BC, obtém-se o esquema mostrado na figura V.7, isto é:

.Situações Típicas: em geral existem muitas situações típicas, as quais devem ser consideradas para poupar tempo e espaço. Neste protótipo uma destas situações típicas é o caso das medições iniciais, as quais, na maioria das vezes, são as mesmas que as medições finais. Portanto o operador não precisa perder tempo configurando duas vezes a mesma situação;

.Definição do Vocabulário: é a tradução dos parâmetros configurados no vocabulário compreendido pela fase de *Operação*. Este conhecimento deve estar explicitamente presente na BC;

- .Objetos e Relações:** representam as relações entre o ensaio configurado e os instrumentos e equipamentos envolvidos no processo. A Agenda deve ser gerada a partir destes dados;
- .Regras de Decisão:** a seleção de ensaios, normas, instrumentos e equipamentos deve ser feita de acordo com algumas regras de decisão. A figura V.4 exemplifica algumas decisões que devem ser tomadas na seleção de uma norma;
- .Regras dos Processos:** indicam a prioridade dos dados a serem selecionados. Como será detalhado posteriormente, os comandos devem obedecer a uma sequência que não pode ser modificada, então na BC deve-se indicar a sequência correta desses comandos;
- .Heurísticas de Configuração:** como dados de muita importância heurística tem-se os valores "default" dos parâmetros de configuração. Estes dados são fornecidos pelos especialistas, segundo a sua vasta experiência nos processos em questão;
- .Normas:** cada norma deve ter a sua representação, indicando todas as suas características, e quais os valores que devem ser adquiridos;
- .Instrumentos:** cada instrumento possui características específicas e próprias. Na representação destes instrumentos deve estar também a codificação dos comandos que são necessários para configurá-los corretamente;
- .Equipamentos:** no primeiro protótipo somente a estufa está presente como equipamento condicionador climático. Então esta parte do conhecimento deve indicar as características de funcionamento da estufa;

.Sistema de Entrada e Saída de Dados: este é um ponto importante para a aquisição de dados. Deve estar clara a forma em que deve ser consultado um dado, podendo ser uma escolha entre várias opções, um dado com valor "default", ou um dado que está restrito dentro de um escopo definido.

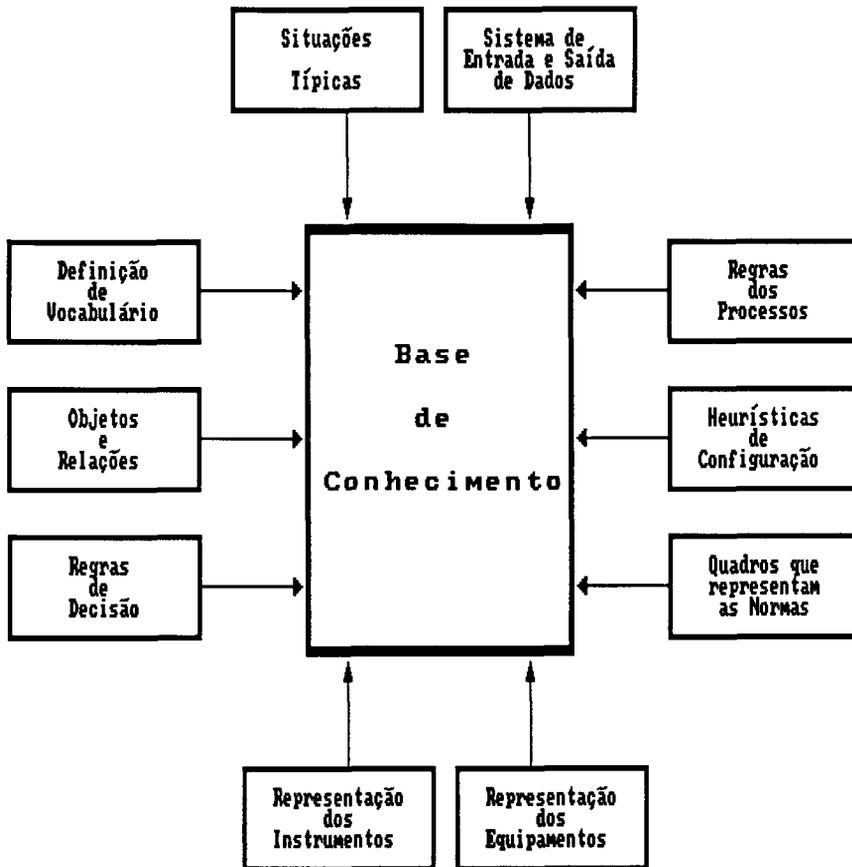


FIGURA U.7: Esquema da BC do Configurador.

V.4.2 BC Física

O conhecimento envolvido neste projeto é fortemente estruturado tanto a nível da descrição dos objetos como das relações que estes mantêm entre si. Isto motivou-nos a desenvolver uma linguagem baseada em quadros (apresentada no CAPÍTULO IV), que dá ao projetista da BC um meio simples de descrever os tipos de objetos do domínio que o sistema deve modelar. No CAPÍTULO IV foram mostradas as operações que lidam com o conhecimento representado em quadros. Tais operações são as mesmas que o MI do *Configurador* vai usar para acessar a informação. Por isso, a seguir é somente detalhada a forma de representar o conhecimento nos quadros.

V.4.2.1 Hierarquia dos quadros

Na figura V.8 é mostrada a hierarquia dos quadros dentro da taxonomia. Cada um dos quadros mostrados possui informações referentes ao quadro em si, os quais são detalhados nas seções seguintes.

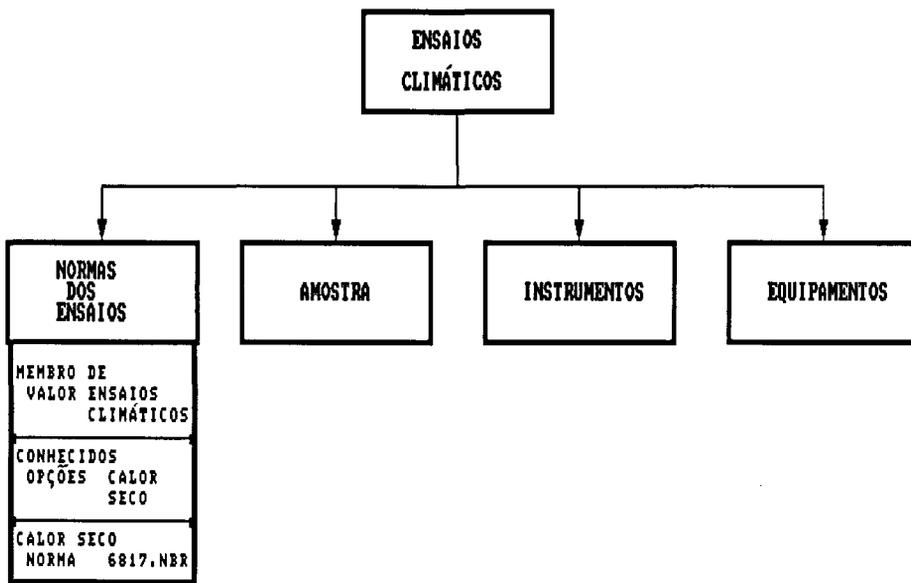


FIGURA U.8: Hierarquia do Conhecimento.

É importante notar que a estrutura dos quadros é dinâmica, e os dados armazenados no início podem ser muito diferentes dos dados que o sistema vai inferir e colocar dentro de cada quadro. Por exemplo, veja a informação necessária que o quadro "ENSAIOS" deve conter após a execução da seleção do ensaio e da seleção da norma de especificação, mostrado na figura V.9.

NOME QUADRO: ENSAIOS
MEMBRO DE: ENSAIOS CLIMÁTICOS
SLOT: CONHECIDOS OPÇÕES: calor seco VALOR: calor seco
SLOT: calor seco NORMA : 6817.NBR
SLOT: NORMA SELECIONADA VALOR: 6796.NBR
SLOT: CONFIGURADA ETAPAS VALOR: PRECONDICIONAMENTO MEDIÇÕES INICIAIS CONDICIONAMENTO ESTABILIZAÇÃO MEDIÇÕES FINAIS
SLOT: CONFIGURADO ENSAIO VALOR: calor seco

FIGURA U.9: Informação no Quadro ENSAIOS.

V.4.2.2 Seleção das Normas de Especificação

O processo de seleção de uma norma de especificação pode ser considerado como um processo de classificação para a solução do problema. Os EC's de calor seco são especificados pela norma de Generalidades 6817 da ABNT. A figura V.10 detalha a representação desta norma.

NOME QUADRO: 6817
SUPERCLASSE: NORMAS SUBCLASSES: 6796 6797 6818 6798
SLOT: nome VALOR: calor seco
SLOT: objetivo VALOR: " texto de objetivo da norma"
SLOT: ensaio VALOR: B
SLOT: COMENTÁRIO dissipa: As normas são classificadas para amostras que dissipam e que não dissipam calor. variação: A variação de temperatura se refere ao momento no qual a amostra se introduze no equipamento
SLOT: divisão dissipa: sim, não, não sei DEFAULT: não
SLOT: sim variação: gradual, rápida DEFAULT: gradual
SLOT: não variação: gradual, rápida DEFAULT: gradual
SLOT: não sei DEMON: 5390
SLOT: gradual sim: 6798.NBR não: 6796.NBR
SLOT: rápida sim: 6797.NBR não: 6819.NBR
SLOT: tipo DEFAULT: armazenagem OPÇÕES: armazenagem, operação, armazenagem e operação.

FIGURA U.10: Representação da Norma 6817.

O "slot" 'divisão' (figura V.10) mostra as alternativas que podem ser selecionadas para determinar se a amostra a ser testada dissipa ou não calor, sendo que, para ambas as respostas ('sim' e 'não'), mais uma característica do EC deve ser determinada, a variação. O tipo de variação da temperatura pode ser rápida ou gradual, característica com a qual se determina o momento de introduzir a amostra dentro do equipamento condicionador climático.

Uma vez determinadas as características do EC, o sistema pode selecionar a norma pelos "slots" 'gradual' ou 'rápida' em conjunto com a faceta 'sim' ou 'não'. Desta forma obtém-se o nome da norma que se adequa ao EC desejado. Outra característica que deve ser determinada é o tipo de EC que se deseja efetuar, isto é armazenagem, operação ou ambas. Toda a informação adquirida neste processo deve ser armazenada no quadro "ENSAIOS", como mostra a figura V.9.

V.4.2.3 Norma 6796

A figura V.11 mostra a representação da norma 6796, a qual corresponde ao ensaio de calor seco com variação gradual da temperatura para espécimens que não dissipam calor.

É importante notar que a forma em que deve ser consultado um dado de que o MI necessita, está armazenada na característica do dado, em geral com um valor típico "default". Por exemplo o caso da 'temperatura' (figura V.11), a qual possui um valor "default" de 40 °C, e tem uma lista das opções válidas para esta norma, sob o nome de 'OPÇÕES'. Então, quando é necessário adquirir esta informação, o MI obtém as opções da lista e as apresenta ao usuário na forma "escolha uma opção". De outra forma opera no caso de ser uma "FAIXA", como na 'tolerância', onde indica ao usuário quais os limites do dado consultado.

NOME QUADRO: 6796	
SUPERCLASSE: NORMAS	
SUBCLASSES: PRE CONDICIONAMENTO MEDICÕES INICIAIS CONDICIONAMENTO ESTABILIZAÇÃO MEDICÕES FINAIS	
SLOT: nome	VALOR: calor seco com variação gradual de temperatura para espécimens que não dissipam calor.
SLOT: ensaio	VALOR: Bb
SLOT: objetivo	HERDAR: 6817
SLOT: descrição	VALOR: " texto da descrição do ensaio "
SLOT: monitor	FAIXA: min=5; max=600 UNIDADE: segundos DEFAULT: 30
SLOT: temperatura	OPÇÕES: 30,40,55,70,85,100,125,155,175,200 UNIDADE: °C DEFAULT: 40
SLOT: tolerância	FAIXA: min=1; max=20 UNIDADE: °C DEFAULT: 2
SLOT: duração	OPÇÕES: 2,16,72,96 UNIDADE: horas DEFAULT: 2

FIGURA U.11: Representação da Norma 6796.

Pode-se ver na figura V.11 que este quadro tem uma ligação para mais cinco quadros, que são: PRECONDICIONAMENTO, MEDIÇÕES INICIAIS, CONDICIONAMENTO, ESTABILIZAÇÃO e MEDIÇÕES FINAIS, detalhados a seguir.

Na figura V.12 está representada a etapa de PRE-CONDICIONAMENTO. Nota-se que um dos "slots" indica os comandos que devem ser configurados para esta etapa, o "slot" 'NÚMERO COMANDOS'. Neste caso são três comandos: 'INSPEÇÃO VISUAL', 'COLOCAR' e 'ATINGIR'. Todos os comandos possuem diferentes parâmetros, os quais são indicados dentro do "slot" do comando respectivo com o nome de 'PARÂMETROS'.

Neste quadro (figura V.12) é possível ver o processo de recuperação da informação (MITTAL et alii, 1984), chamado herança. Este é o maior benefício da teoria de quadros, onde um quadro pode "HERDAR" um valor de uma faceta (ou atributo), ou um quadro pode inferir um valor a partir de quadros de maior nível na hierarquia. Com os quadros é possível aproveitar o passo de informação, fazendo-a migrar para baixo, para cima ou para os lados na hierarquia, mas os valores só podem passar de uma classe aos seus membros. Isto acarreta uma grande economia de espaço no armazenamento do conhecimento.

No caso do comando 'ATINGIR' (figura V.12) existe a herança dos valores 'tolerância' e 'equipamento', cujos valores devem ser obtidos do quadro pai, indicado no "slot" 'MEMBRO DE', neste caso '6796'.

No quadro da etapa de PRE-CONDICIONAMENTO, tem-se que os valores para 'temperatura' e 'duração' devem ser obtidos do mesmo quadro, no slot do mesmo nome. Este procedimento de herança é o chamado 'OBTER' e não precisa de parâmetros.

Outro recurso importante da teoria dos quadros é a ativação de procedimentos denominados "DAEMONS". No caso do

quadro PRE-CONDICIONAMENTO é possível ver o valor 'tempo máximo' do "slot" '3' (comando 'ATINGIR'), o qual, quando é consultado, ativa um procedimento, denominado 'CALCMAX', que calcula o tempo máximo deste comando.

NOME QUADRO: PRE-CONDICIONAMENTO
MEMBRO DE: 6796
SLOT: PRECISA OPÇÕES: sim, não
SLOT: temperatura OPÇÕES: 30,40,55,70,85,100,125,155,175,200 UNIDADE: °C DEFAULT: 40
SLOT: duração FAIXA: min=60 ; max=120 UNIDADE: minutos DEFAULT: 60
SLOT: NÚMERO COMANDOS VALOR: 1, 2, 3
SLOT: 1 CMD: INSPEÇÃO VISUAL PARÂMETROS: 0
SLOT: 2 CMD: COLOCAR PARÂMETROS: desde, para desde: Amostra pronta para uso para: Equipamento
SLOT: 3 CMD: ATINGIR PARÂMETROS: temperatura, tolerância, duração, equipamento, ciclo, operação, tempo máximo. temperatura: OBTER tolerância: HERDAR equipamento: HERDAR ciclo: CONFIG tempo máximo: CALCMAX duração: OBTER operação: configuração

FIGURA U.12: Representação da etapa PRE-CONDICIONAMENTO.

Na etapa de MEDIÇÕES INICIAIS (figura V.13) é utilizado um outro procedimento para herdar informação, chamado 'OBTEM'. Este procedimento difere do 'OBTER' por possuir um parâmetro. Este parâmetro indica o nome do "slot" deste quadro onde se deve obter o dado solicitado sob a faceta com o mesmo nome que a faceta que tem este tipo de herança. Por exemplo, o valor do 'instrumento' do "slot" '2', deve ser obtido do slot 'GERAR'. O processo de seleção do instrumento cria o valor da faceta 'instrumento' dentro do "slot" 'GERAR'. Este valor criado é o que deve ser passado como valor do parâmetro.

NOME QUADRO: MEDIÇÕES INICIAIS	
MEMBRO DE: 6796	
SLOT: PRECISA	OPÇÕES: sim, não
SLOT: GERAR	OPÇÕES: sim, não
SLOT: período	FAIXA: min=10 ; max=300. unidade: segundos DEFAULT: 10
SLOT: tempo	FAIXA: min=60 ; max=120. unidade: minutos DEFAULT: 60
SLOT: NÚMERO COMANDOS	VALOR: 1, 2, 3
SLOT: 1	CMD: SETAR PARÂMETROS: instrumento, setup instrumento: OBTER setup: OBTEM instrumento
SLOT: 2	CMD: SETAR PARÂMETROS: instrumento, setup instrumento: OBTEM GERAR setup: OBTEM GERAR
SLOT: 3	CMD: MEDIR PARÂMETROS: instrumento, tempo, tipo medida, período, canal. instrumento: OBTER tempo: OBTER período: OBTER tipo medida: OBTEM instrumento canal: OBTEM instrumento

FIGURA U.13: Representação da etapa MEDIÇÕES INICIAIS.

Na etapa de **CONDICIONAMENTO** (figura V.14) existe um outro procedimento para herdar informação, chamado 'IRMÃO'. Este procedimento possui um parâmetro. Este parâmetro indica o nome do quadro onde se deve obter o dado solicitado sob o "slot" com o mesmo nome que a faceta que tem este tipo de herança. Por exemplo, o valor da 'duração' do "slot" '2', deve ser obtido do "slot" 'duração' do quadro 'ESTABILIZAÇÃO'. O processo de configuração da etapa de ESTABILIZAÇÃO adquire o valor do "slot" 'duração', sendo este o valor passado.

NOME QUADRO: CONDICIONAMENTO
MEMBRO DE: 6796
SLOT: PRECISA OPÇÕES: sim, não
SLOT: NÚMERO COMANDOS VALOR: 1, 2, 3.
SLOT: 1 CMD: ATINGIR PARÂMETROS: temperatura, tolerância, duração, equipamento, ciclo, operação, tempo máximo. temperatura: HERDAR tolerância: HERDAR equipamento: HERDAR ciclo: CONFIG tempo máximo: CALCMAX duração: CALCULAR SUBIDA, PRECONDICIONAMENTO, 6796, temperatura. operação: configuração
SLOT: 2 CMD: MANTER PARÂMETROS: temperatura, tolerância, duração, equipamento, ciclo, operação, tempo máximo. temperatura: HERDAR tolerância: HERDAR equipamento: HERDAR ciclo: CONFIG tempo máximo: CALCMAX duração: IRMAO, ESTABILIZACAO. operação: configuração
SLOT: 3 CMD: MANTER PARÂMETROS: temperatura, tolerância, duração, equipamento, ciclo, operação, tempo máximo. temperatura: HERDAR tolerância: HERDAR equipamento: HERDAR ciclo: CONFIG tempo máximo: CALCMAX duração: HERDAR operação: configuração

FIGURA U.14: Representação da etapa de CONDICIONAMENTO.

Tanto na etapa de CONDICIONAMENTO, quanto na etapa de ESTABILIZAÇÃO (figura V.15), aparecem DAEMONS. No caso do CONDICIONAMENTO, no "slot" '1' (comando 'ATINGIR'), tem-se a faceta 'duração' que ativa um procedimento chamado 'CALCULAR SUBIDA'. Este procedimento possui três parâmetros:

- . 'PRECONDICIONAMENTO': sendo este o nome de um quadro;
- . '6796': sendo o nome de outro quadro;
- . 'temperatura': sendo este o nome de um "slot".

O procedimento "CALCULAR SUBIDA" obtém o valor do "slot" com nome 'temperatura' de ambos os quadros ('PRECONDICIONAMENTO' e '6796'), e logo calcula, a partir da diferença de temperatura, qual a 'duração' do tempo que o equipamento precisa para atingir a temperatura desejada. Este procedimento é mostrado no APÊNDICE D.

NOME QUADRO: ESTABILIZAÇÃO
MEMBRO DE: 6796
SLOT: PRECISA OPÇÕES: sim, não
SLOT: GERAR OPÇÕES: sim, não
SLOT: temperatura OPÇÕES: 30,40,55,70,85,100,125,155,175,200 UNIDADE: °C DEFAULT: 70
SLOT: duração FAIXA: min=60 ; max=120 UNIDADE: minutos DEFAULT: 60
SLOT: NÚMERO COMANDOS VALOR: 1
SLOT: 1 CMD: ATINGIR PARÂMETROS: temperatura, tolerância, duração, equipamento, ciclo, operação, tempo máximo. temperatura: OBTER tolerância: HERDAR equipamento: HERDAR ciclo: CONFIG tempo máximo: CALCMAX duração: CALCULAR DESCIDA, 6796, ESTABILIZAÇÃO, temperatura. operação: configuração

FIGURA U.15: Representação da etapa ESTABILIZAÇÃO.

V.4.2.4 Representação dos Instrumentos

A hierarquia dos quadros na representação dos instrumentos é mostrada na figura V.16. O quadro 'INSTRUMENTOS' tem todas as opções das medidas que podem ser obtidas dos diferentes instrumentos, indicando para cada medida que instrumento(s) pode(m) realizar a medição que se deseja efetuar.

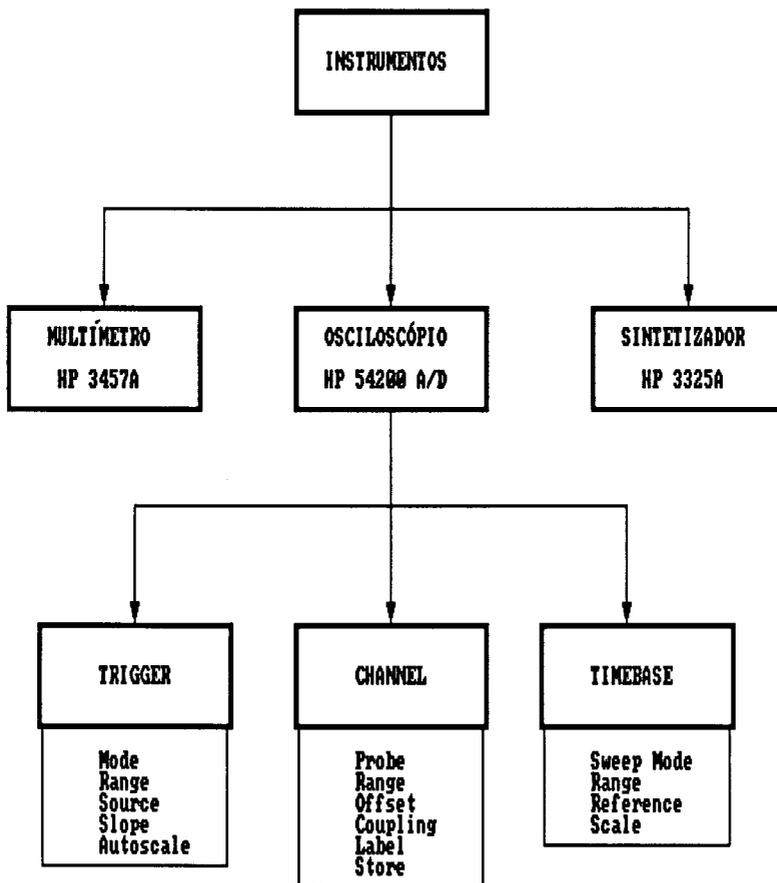


FIGURA U.16: Informação representada do Osciloscópio.

Na figura V.17 se vê a forma na qual é representada uma das codificações que deve ser gerada para a configuração do osciloscópio. É importante notar que esta parte do conhecimento é usada na geração da agenda para formar a sequência correta dos comandos que devem ser enviados para o instrumento na fase de *operação*.

NOME QUADRO: TIMEBASE
MEMBRO DE: osciloscópio HP 54200 A/D
SLOT: PARÂMETROS VALOR: 0
SLOT: SWEEP MODE OPÇÕES: AUTO, TRIGGERED, SINGLED CÓDIGO: MODE DEFAULT: AUTO
SLOT: RANGE FAIXA: min=50 E-9, max=10 CÓDIGO: RANG DEFAULT: 1 unidade: segundos
SLOT: REFERENCE OPÇÕES: LEFT, CENTER, RIGHT CÓDIGO: REF DEFAULT: CENTER
SLOT: SCALE OPÇÕES: DISABLED, POS-PULSE, NEG-PULSE, RISE, FALL CÓDIGO: SCAL DEFAULT: PERIOD

FIGURA U.17: Representação dos Comandos do Osciloscópio.

V.5 Projeto

Considerando os dados obtidos da *Aquisição do Conhecimento*, da *Representação do Conhecimento* e da *Análise do "Hardware"*, foi projetado um "software" que possui os três módulos que constituem um SE: a BC, o MI e a IHM.

V.5.1 BC

A BC tem como função o armazenamento do conhecimento, representado em quadros, como foi detalhado no item V.4. Esta estrutura, distribuída em seis BD's, permite o acesso independente às diferentes partes do conhecimento:

- .**Ensaio Climáticos:** onde é armazenada a estrutura hierárquica dos quadros, mostrada na figura V.8. Esta parte do conhecimento deve estar sempre presente no processo de configuração, portanto é carregada no início;
- .**Instrumentos:** onde é armazenada a representação em quadros dos instrumentos deste primeiro protótipo. Esta parte do conhecimento é carregada cada vez que é configurado um instrumento. Após a configuração do instrumento é retirada da memória interna de trabalho;
- .**Equipamentos:** onde é armazenada a representação dos equipamentos deste protótipo (estufa). É carregada somente quando é necessário configurar o equipamento, sendo retirada da memória interna de trabalho uma vez configurado o equipamento;

- .Regras de Configuração:** armazena as regras para configurar os ensaios, os instrumentos e os equipamentos. Cumpre a função de interface entre o MI e a BC;
- .Normas:** armazena os quadros que representam as normas de especificação dos EC's. Este conhecimento é carregado quando o MI seleciona a norma adequada às características do ensaio.
- .Codificação:** armazena a codificação dos comandos que compõem a Agenda. É carregada no fim, quando é iniciado o processo da geração da Agenda.

V.5.2 MI

O MI corresponde à parte do SE que lida com o conhecimento, fazendo as inferências necessárias para obter uma solução para o problema. Neste protótipo o MI é dividido em quatro subsistemas:

- .Operações com os quadros:** cuja função é realizar todas as operações com os quadros, tais como obter, criar ou apagar um quadro, "slot", faceta ou valor, detalhados no CAPÍTULO IV;
- .Seleção de Ensaio, Características e Normas:** a função deste subsistema é orientar a busca para seleção dos ensaios, seleção das características do ensaio, e, segundo estas características, iniciar o processo de seleção da norma adequada ao ensaio desejado;
- .Configuração:** a função deste módulo é realizar a configuração do EC, segundo a norma selecionada. Como esta fase do projeto opera na forma "off-line",

possui facilidades de configuração para fixar as regras que vão gerenciar o processo;

.Gerador da Agenda: cuja função é gerar automaticamente a Agenda dos comandos a serem executados na fase de *Operação*. Esta Agenda é gerada a partir da informação adquirida durante a fase de *Configuração*, e se encarrega de colocar os comandos na sequência correspondente, aproveitando o paralelismo permitido pelo "hardware".

V.5.3 IHM

A IHM é o módulo encarregado de apresentar os dados para o usuário, usando as ferramentas disponíveis para fazer a aquisição do dado solicitado pelo MI. É dividido em três subsistemas:

.Ferramentas: cuja função é gerar menus, janelas, linhas para adquirir dados, linhas de estado e menus tipo "pulldown";

.Aquisição dos Dados: cuja função é adquirir o dado solicitado pelo MI, segundo a forma que indica a BC. Isto é, se for uma escolha de opção, então deve usar a ferramenta que gera um menu para indicar ao usuário que deve escolher uma das opções apresentadas;

.Leitura/Escrita de arquivos: cuja função é fazer a interface entre a BC e o conhecimento solicitado pelo MI. Ou seja, cada vez que o MI precisar de um conhecimento, e este não estiver presente na memória interna de

trabalho do sistema, então deve carregar a BC que o contém, indicando ao usuário que está realizando uma operação de leitura/escrita em arquivo.

A implementação do SE, detalhada no APÊNDICE D, foi feita segundo a estrutura modular mostrada na figura V.18.

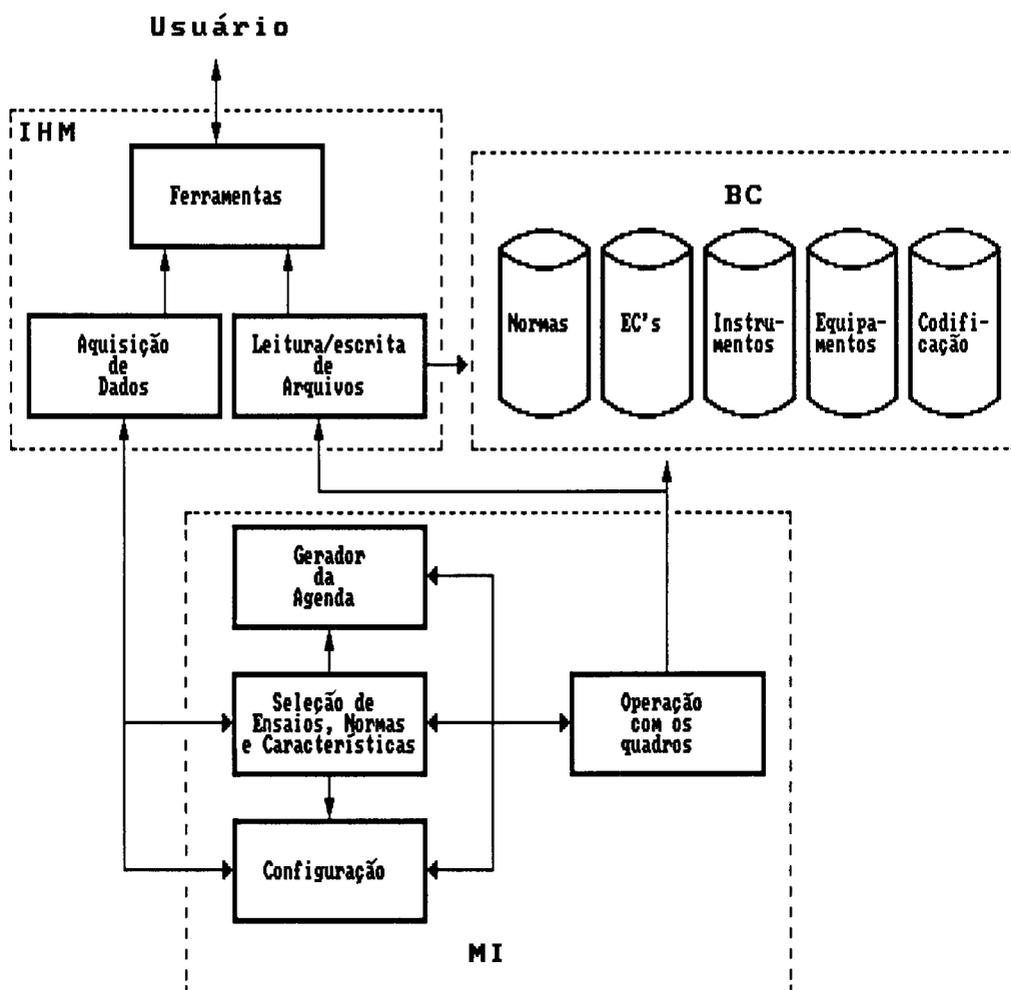


FIGURA U.18: Módulos do Configurador.

V.5.4 Plano do Teste

Para verificação do comportamento do sistema utilizando os procedimentos descritos neste trabalho foi elaborado um teste contendo todas as fases típicas de um sistema completo. Através do *Configurador* é gerada uma Agenda que contém os comandos para gerenciar o sistema na fase de *Operação*.

A figura V.19 detalha as etapas do teste. O EC selecionado é um ensaio de calor seco com variação gradual de temperatura para amostras que não dissipam calor. O tipo do EC é de armazenagem, sendo que este tipo de ensaio não precisa de medições intermediárias.

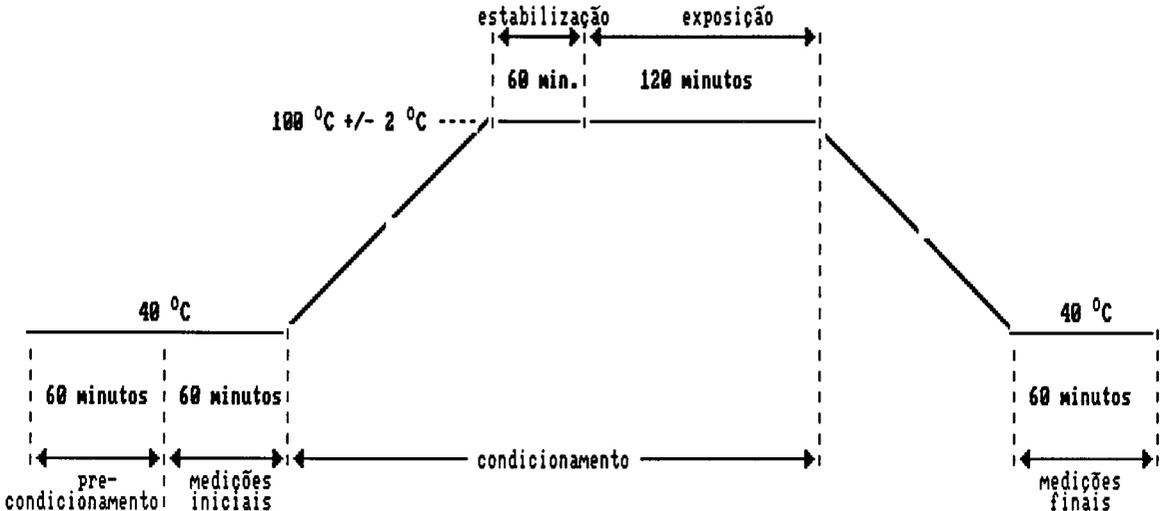


FIGURA U.19: Configuração de um EC para teste.

As etapas que compõem o teste são:

PRE-CONDICIONAMENTO: configurada para atingir uma temperatura de 40 °C em 60 minutos.

MEDIÇÕES INICIAIS: o sintetizador é configurado para gerar um sinal senoidal de 1 Volt de amplitude com uma frequência de 1 KHz. O osciloscópio é configurado para fazer uma medição periódica, através do canal 1 a cada 10 segundos, da frequência gerada pelo sintetizador, durante um tempo de 60 minutos.

CONDICIONAMENTO: configurada para atingir uma temperatura de 100 °C. Quando esta temperatura é atingida, deve ser mantida por um tempo de duas horas (120 minutos).

ESTABILIZAÇÃO: configurada para um tempo de 60 minutos.

MEDIÇÕES FINAIS: a configuração desta etapa é a mesma que a da etapa MEDIÇÕES INICIAIS.

V.6 Implementação

O primeiro passo é indicar quais são as alternativas para selecionar a linguagem de implementação.

LEE (1986) mostra no seu trabalho as facilidades do Prolog para representar conhecimento. LANE (1988) mostra as facilidades gráficas do Turbo Prolog para gerenciamento de janelas, menus e posição do cursor, como também as novas facilidades para gerenciar BD's multidimensionais na versão 2.0. WONG (1986), no seu trabalho, faz uma avaliação das diferentes implementações da linguagem Prolog para IBM PC,

sendo que a mais eficiente em termos de velocidade de operação é o Turbo Prolog.

Além destas vantagens do Turbo Prolog, pode-se operar com BD's dinâmicas, e está operacional uma interface de comunicação com o Núcleo do Sistema Operacional (NSO) em Tempo Real (CEPEL, 1981).

Estas razões motivaram a escolha do Turbo Prolog versão 2.0 (BORLAND INTERNATIONAL, 1988a, 1988b) como linguagem de implementação do primeiro protótipo. A fase da *Configuração* foi desenvolvida em sua totalidade em Turbo Prolog versão 2.0. O APÊNDICE D mostra os detalhes de implementação dos módulos apresentados na seção V.5.

V.7 Interface Homem-Máquina (IHM)

A IHM interage com o usuário através de menus, janelas e gráficos, oferecendo também uma opção de ajuda. Cada alternativa selecionada, através do posicionamento correto do cursor, ou dado fornecido pelo usuário, indica ao sistema a informação solicitada pelo MI para continuar o processo de configuração, até preencher toda a informação necessária para a configuração completa do EC.

A seguir são apresentadas as telas usadas pelo *Configurador* para obter os dados de que necessita através de consulta ao usuário.

A figura V.20 mostra a tela pela qual deve ser selecionado o ensaio que se deseja configurar. A lista dos ensaios conhecidos pelo sistema é obtida do quadro que contém as informações correspondentes ao ensaio selecionado. Neste caso existe uma única opção de ensaio, o de calor seco.

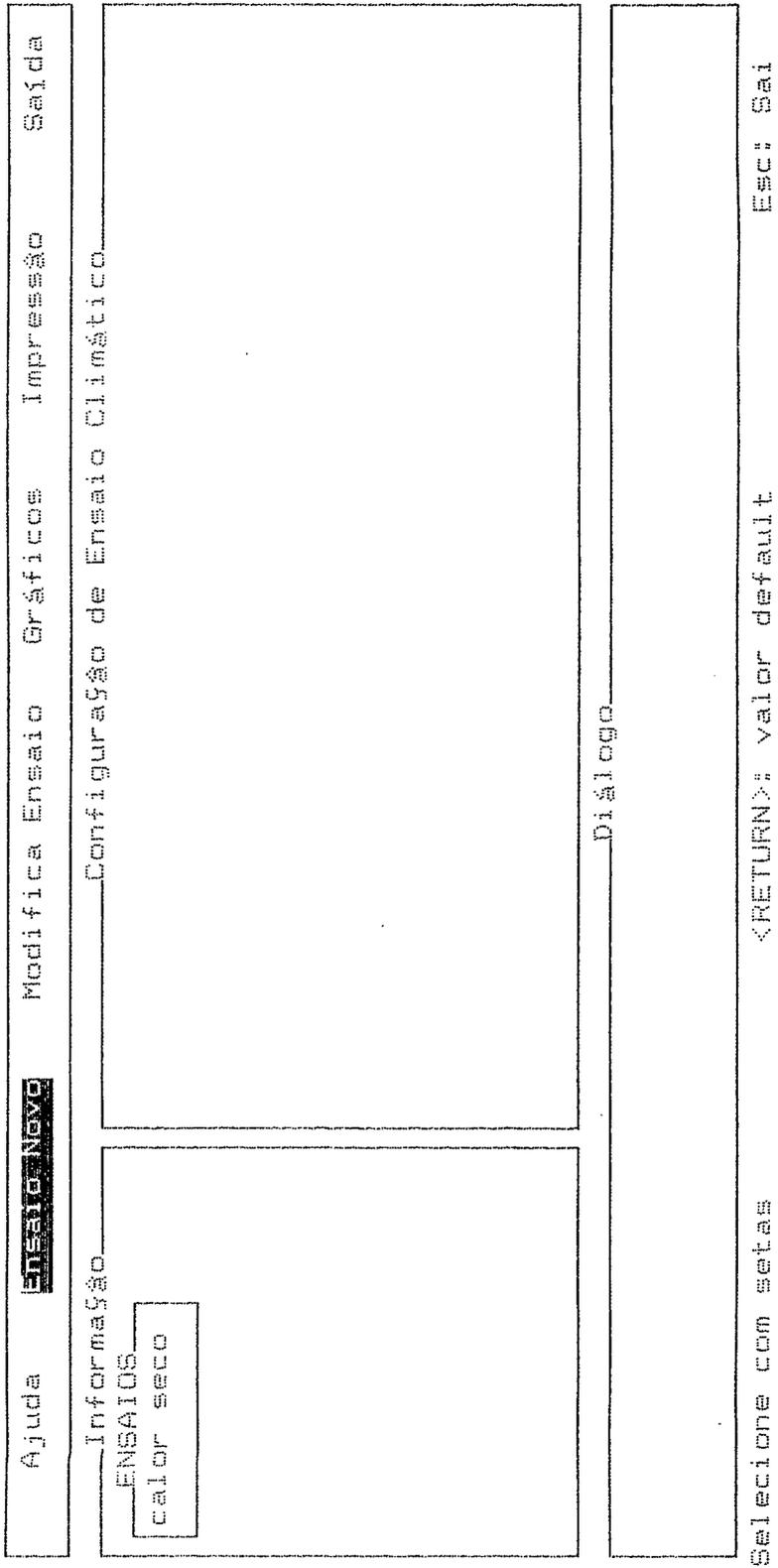


FIGURA U.20: Tela principal -

Na escolha de ensaio a calor seco o sistema seleciona a norma 6817, que corresponde às generalidades do ensaio de calor seco. Dentre as informações contidas neste quadro (6817) estão as características que devem ser definidas para a seleção da norma específica do ensaio que se deseja executar, que são: dissipação de calor da amostra, variação da temperatura do ensaio e o tipo de ensaio.

A característica que corresponde à dissipação de calor da amostra que se deseja testar é selecionada como mostra a figura V.21. Existem três opções possíveis: dissipa calor (sim), não dissipa calor (não) e não é sabido se dissipa (não sei). Geralmente, o valor mais usado é não dissipa calor, por isso o valor "default" assume esta opção.

É importante destacar que o sistema deve ter bem definidas as três características do ensaio. A parte "Diálogo" esclarece o objetivo de cada característica (figura V.22). Este objetivo mostrado também é obtido do quadro 6817.

Uma vez que as características do ensaio de calor seco estão definidas, como mostram as figuras V.20, V.21 e V.22, então o sistema seleciona a norma adequada ao ensaio desejado, no caso a norma 6796. Como mostra a figura V.23, é mostrada uma descrição da norma selecionada.

Uma vez selecionada a norma específica do ensaio, então a configuração pode ser iniciada, como mostra a figura V.24. Neste caso as etapas que devem ser configuradas foram obtidas do quadro 6796. O usuário pode ir selecionando as etapas que ele deseja configurar primeiro, até configurar todas as etapas.

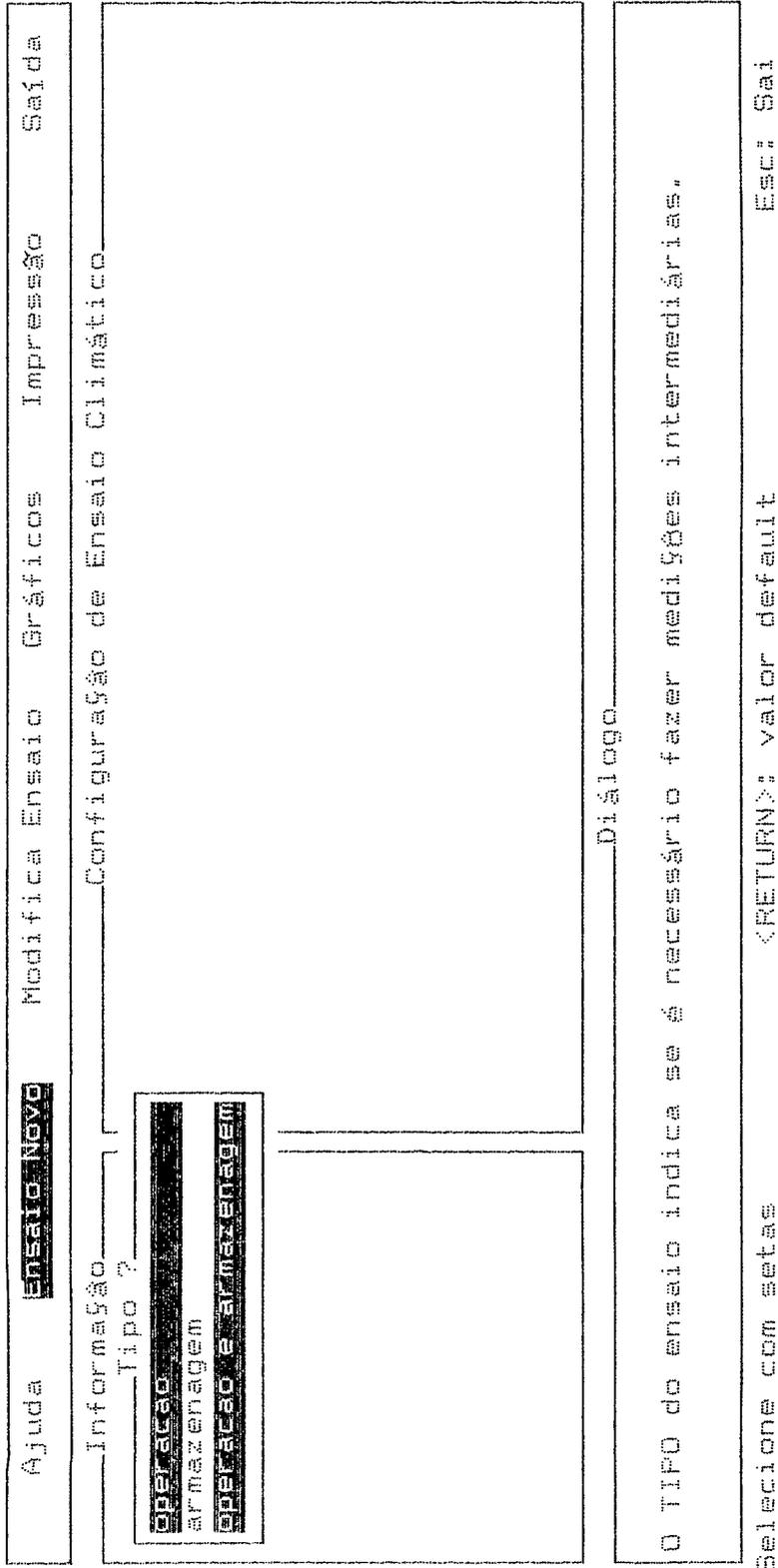


FIGURA U - 21: Tela de Seleção -

Ajuda	Ensaio Novo	Modifica Ensaio	Gráficos	Impressão	Saída
-------	--------------------	-----------------	----------	-----------	-------

Informação	Configuração de Ensaio Climático
Dissipa ?	
<input checked="" type="checkbox"/> sim <input type="checkbox"/> não <input type="checkbox"/> não sei	

Diálogo

As normas para Calor Seco são distintas para amostras que dissipam (sim) e que não dissipam calor (não) para determinar a característica do ensaio.

Selecione com setas <RETURN>: valor default Esc: Sai

FIGURA U - 22: Tela de diálogo -

Ajuda	ENSAIO NOVO	Modifica Ensaio	Gráficos	Impressão	Saída	
<p>Informação</p>	<p>Configuração de Ensaio Climático</p> <p>DESCRICAÇÃO DA NORMA 6796.NBR</p> <p>*****</p> <p>A amostra, estando à temperatura ambiente do laboratório, é introduzida na câmara, que também deve estar na temperatura do laboratório. A temperatura da câmara é então ajustada para o valor correspondente ao grau de severidade indicado na norma selecionada.</p> <p>Após a estabilidade térmica da amostra ter sido alcançada, a mesma deve ficar exposta a estas condições durante o tempo especificado.</p> <p>As amostras sob ensaio devem estar normalmente nas condições 'pronta para uso'. A circulação forçada de ar é normalmente usada para este ensaio.</p>				<p>Diálogo</p>	<p>A característica variação de temperatura determina o momento em que a amostra deve ser introduzida no equipamento.</p> <p>APERTE UMA TECLA ...</p>

FIGURA U - 23: Tela de descrição da Norma -

Ajuda	Ensaio Novo	Modifica Ensaio	Gráficos	Impressão	Saída
Informação Etapas PRECONDICIONAMENTO MEDICIONES INICIAIS CONDICIONAMENTO ESTABILIZACAO MEDICIONES FINAIS		Configuração de Ensaio Climático			
Diálogo					
Seleccione com setas <RETURN>: valor default Esc: Sai					

FIGURA U.24: Etapas para configurar.

As figuras V.25 e V.26 mostram os dois recursos usados para a obtenção de dados do usuário:

- menu com valor "default": indica o tipo do dado requisitado e as unidades na qual está expresso. No caso apresentado na figura V.25, o tipo é a temperatura, expressa em °C, onde o valor típico desta etapa é de 40 °C, podendo ser escolhida entre uma das temperaturas mostradas no menu;

- aquisição de um dado que deve estar dentro de um escopo: é indicado o valor mínimo e o valor máximo, assim como as unidades do dado requerido. Neste caso (figura V.26) trata-se da duração da etapa, expressa em minutos, onde o valor mínimo é 60 minutos e o valor máximo é 120 minutos, o valor "default" é 60 minutos.

Ajuda	Ensaio Novo	Modifica Ensaio	Gráficos	Impressão	Saída
-------	--------------------	-----------------	----------	-----------	-------

Informação

temperatura (°C)

30
40
55
70
95
100
125
155
175
200

CONFIGURANDO PRECONDICIONAMENTO

é o tratamento de uma amostra com a finalidade de remover ou neutralizar parcialmente os efeitos da sua vida anterior.

Quando exigido, é a primeira operação do ensaio.

Indicar Temperatura e Duração do condicionamento.

Diálogo

Selecione com setas <RETURN>; valor default Esc: Sai

FIGURA U - 25: Menu com valor "default".

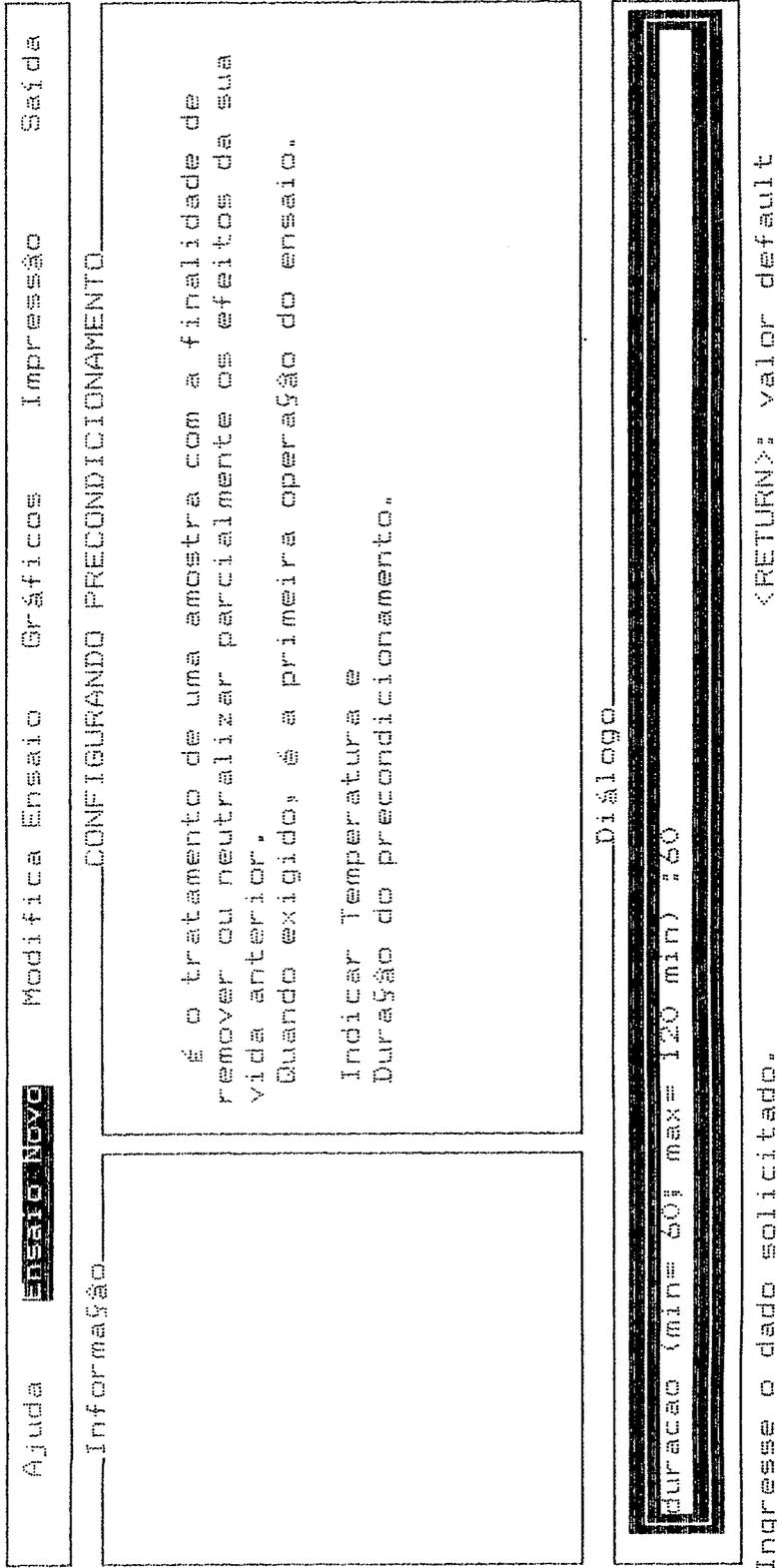


FIGURA U-26: Uso da Ferramenta Faixa.

V.8 Teste de Desempenho

Para realizar o Teste de Desempenho deste primeiro protótipo do SE em configuração de EC's, foi configurado um EC com as características detalhadas na seção V.5.4.

A partir desta configuração o sistema gerou a Agenda com os seguintes comandos:

ETAPA 1:

Esta etapa tem como objetivo realizar o PRECONDICIONAMENTO do EC. Este condicionamento consiste em atingir uma temperatura de 40 °C em 60 minutos. Para isso existem dois comandos em sequência. O primeiro é um comando paralelo e o segundo é um comando simples:

1: ATINGIR, INÍCIO e STATUS;

2: FIM.

1: ATINGIR, INÍCIO, STATUS.

ATINGIR: temperatura = 40 °C;
 tolerância = 2 °C;
 duração = 60 minutos;
 equipamento = Estufa;
 operação = configuração;
 tempo máximo= 70 minutos;

INÍCIO: equipamento = Estufa;
 operação = início;

STATUS: equipamento = Estufa;
 operação = status;

2: FIM

FIM: equipamento = Estufa;
 operação = fim.

ETAPA 2:

O objetivo desta etapa é executar as MEDIÇÕES INICIAIS. Neste caso em questão deve ser configurado o canal 1 do osciloscópio para fazer uma medição de frequência. Também deve ser configurado o sintetizador para gerar um sinal senoidal de 1 KHz com uma amplitude de 1 Volt. A medida deve ser realizada durante 60 minutos. São utilizados quatro comandos: três comandos simples e um comando paralelo:

3: SETAR;

4: SETAR;

5: ATINGIR, INÍCIO, STATUS, MEDIR;

6: FIM.

3: SETAR.

```
SETAR: instrumento = osciloscópio;
      setup= "TIM;MODE TRIGGERED;RANG 0.1;REF
              CENT;SCAL PERIOD;CHAN 1;PROB 1;
              RANG 1;OFFS 0;COUP DC";
```

4: SETAR.

```
SETAR: instrumento = sintetizador;
      setup          = "F1;AM 1 V;FR 1000HZ";
```

5: ATINGIR, INÍCIO, STATUS, MEDIR.

```
ATINGIR: temperatura = 40 °C;
          tolerância  = 2 °C;
          duração     = 60 minutos;
          equipamento = Estufa;
          operação    = configuração;
          tempo máximo= 60 minutos;
```

```
INÍCIO: equipamento = Estufa;
        operação    = início;
```

```
STATUS: equipamento = Estufa;
        operação    = status;
```

```
MEDIR: instrumento = osciloscópio;
        tipo medida = frequência;
        tempo       = 60 minutos;
        período     = 10 segundos;
        canal       = 1;
```

6: FIM.

FIM: equipamento = Estufa;
 operação = fim.

ETAPA 3:

O objetivo desta etapa é atingir as condições para realizar o condicionamento propriamente dito, ou seja, deve atingir a temperatura de 100°C para submeter a amostra durante um determinado tempo. Para isso são dados dois comandos em sequência:

7: ATINGIR, INÍCIO, STATUS;**8: FIM.****7: ATINGIR, INÍCIO, STATUS.**

ATINGIR: temperatura = 100 °C;
 tolerância = 2 °C;
 duração = 60 minutos;
 equipamento = Estufa;
 operação = configuração;
 tempo máximo= 70 minutos;

INÍCIO: equipamento = Estufa;
 operação = início;

STATUS: equipamento = Estufa;
 operação = status;

8: FIM.

FIM: equipamento = Estufa;
 operação = fim.

ETAPA 4:

O objetivo desta etapa é realizar a ESTABILIZAÇÃO do ensaio na fase de condicionamento. Para isso, existem dois comandos em sequência, o primeiro é um comando paralelo e o segundo é um comando simples:

9: ATINGIR, INÍCIO, STATUS;**10: FIM.**

9: ATINGIR, INÍCIO, STATUS.

ATINGIR: temperatura = 100 °C;
 tolerância = 2 °C;
 duração = 60 minutos;
 equipamento = Estufa;
 operação = configuração;
 tempo máximo= 60 minutos;
INÍCIO: equipamento = Estufa;
 operação = início;
STATUS: equipamento = Estufa;
 operação = status;

10: FIM.

FIM: equipamento = Estufa;
 operação = fim.

ETAPA 5:

O objetivo desta etapa é realizar o **CONDICIONAMENTO** propriamente dito, que consiste em manter a temperatura do ensaio durante o tempo de 120 minutos. Para isso são dados dois comandos em sequência: o primeiro é um comando paralelo e o segundo é um comando simples:

11: ATINGIR, INÍCIO, STATUS;

12: FIM.

11: ATINGIR, INÍCIO, STATUS.

ATINGIR: temperatura = 100 °C;
 tolerância = 2 °C;
 duração = 120 minutos;
 equipamento = Estufa;
 operação = configuração;
 tempo máximo= 120 minutos;
INÍCIO: equipamento = Estufa;
 operação = início;
STATUS: equipamento = Estufa;
 operação = status;

12: FIM.

FIM: equipamento = Estufa;
 operação = fim.

ETAPA 6:

O objetivo desta etapa é realizar a ESTABILIZAÇÃO após o condicionamento, ou seja, deve atingir a temperatura para iniciar as medições finais. São gerados dois comandos em sequência: o primeiro é um comando paralelo e o segundo é um comando simples:

13: ATINGIR, INÍCIO, STATUS;

14: FIM.

13: ATINGIR, INÍCIO, STATUS.

ATINGIR: temperatura = 40 °C;
 tolerância = 2 °C;
 duração = 120 minutos;
 equipamento = Estufa;
 operação = configuração;
 tempo máximo = 130 minutos;

INÍCIO: equipamento = Estufa;
 operação = início;

STATUS: equipamento = Estufa;
 operação = status;

14: FIM.

FIM: equipamento = Estufa;
 operação = fim.

ETAPA 7:

O objetivo desta etapa é realizar as MEDIÇÕES FINAIS. Os quatro comandos desta etapa são os mesmos da etapa MEDIÇÕES INICIAIS:

15: SETAR;

16: SETAR;

17: ATINGIR, INÍCIO, STATUS, MEDIR;

18: FIM.

15: SETAR.

```

SETAR: instrumento = osciloscópio;
      setup= " TIM;MODE TRIG;RANG 0.1;REF
              CENT; SCAL PER;CHAN 1;PROB 1;
              RANG 1;OFFS 0;COUP DC";

```

16: SETAR.

```

SETAR: instrumento = sintetizador;
      setup          = "F1;AM 1 V;FR 1000HZ";

```

17: ATINGIR, INÍCIO, STATUS, MEDIR.

```

ATINGIR: temperatura = 40 °C;
         tolerância   = 2 °C;
         duração      = 60 minutos;
         equipamento  = Estufa;
         operação     = configuração;
         tempo máximo= 60 minutos;

INÍCIO:  equipamento = Estufa;
         operação    = início;

STATUS:  equipamento = Estufa;
         operação    = status;

MEDIR:   instrumento = osciloscópio;
         tipo medida = frequência;
         tempo       = 60 minutos;
         período     = 10 segundos;
         canal       = 1;

```

18: FIM.

```

FIM:    equipamento = Estufa;
        operação    = fim.

```

Avaliação do Comportamento da Agenda no Sistema Operação

A Agenda gerada foi executada no sistema de *Operação*. Observou-se que o sistema realizou, ao longo do teste, o controle da temperatura da Estufa e o controle dos instrumentos conforme a programação elaborada na fase de *Configuração*, demonstrando que o protótipo desenvolvido, gera uma Agenda confiável.

CAPÍTULO VI

RESULTADOS OBTIDOS

A implementação do primeiro protótipo do Sistema de Automação de Ensaio Climáticos aplicando a metodologia proposta neste trabalho permitiu a avaliação do tempo consumido em cada fase do desenvolvimento, conforme indicado na TABELA VI.1. Observa-se que o tempo total de 16 meses é considerado aceitável em função da complexidade do projeto.

TABELA VI.1: Tempo consumido em cada fase da metodologia.

Fase	duração em meses
ESTUDO	1
AQUISIÇÃO DO CONHECIMENTO	7
ANÁLISE DO HARDWARE	2
REPRESENTAÇÃO DO CONHECIMENTO	2
PROJETO	1
IMPLEMENTAÇÃO E IHM	2
TESTE DO DESEMPENHO	1
TOTAL	16 meses

A fase de ESTUDO se confunde com o estudo para desenvolver a metodologia, logo o tempo de quatro meses contém um tempo de pesquisa além do projeto em si. O tempo para o ESTUDO específico relativo ao projeto deste

protótipo pode ser estimado em cerca de um mês. Esta fase teve uma duração de cerca de quatro meses para verificar a viabilidade do projeto e realizar um plano de desenvolvimento. Na realização deste plano surgiu a necessidade de dispor de uma metodologia de desenvolvimento, para o qual foram pesquisados livros e artigos, onde, no fim, foi obtida a metodologia proposta neste trabalho.

HARMON e KING (1985) indicam que, depois de equacionar o problema, deve ser desenvolvido um protótipo do sistema, não propondo nenhuma fase de AQUISIÇÃO DO CONHECIMENTO. Esta fase de AQUISIÇÃO DO CONHECIMENTO é indispensável para a especificação correta do sistema, que comcorde com os critérios dos especialistas no domínio.

Na fase de AQUISIÇÃO DO CONHECIMENTO a maior parte do tempo foi consumido na extração do conhecimento (especificação da BC) e na especificação inicial do projeto. A definição da IHM e a necessidade do "hardware", que também formam parte desta fase, consumiram um tempo menor.

WEISS e KULIKOWSKI (1988) não fazem uma diferença entre AQUISIÇÃO DO CONHECIMENTO e REPRESENTAÇÃO DO CONHECIMENTO, sendo que na metodologia apresentada neste trabalho, a fase de AQUISIÇÃO DO CONHECIMENTO deve ser realizada independentemente da forma na qual será representado o conhecimento. A tarefa de representação do problema em computador, como é chamada por WEISS e KULIKOWSKI, na metodologia proposta é deixada para a fase de REPRESENTAÇÃO DO CONHECIMENTO, onde o engenheiro do conhecimento estuda os esquemas mais adequados de representação.

A fase de REPRESENTAÇÃO DO CONHECIMENTO teve uma duração de dois meses. Deve-se ressaltar que o uso do editor de quadros apresentado no CAPÍTULO IV proporcionou grande contribuição. Com este editor foi possível fazer as

modificações na BC sem a necessidade de alterar um arquivo codificado com terminologia não acessível ao usuário. Isto poupou muito tempo de processamento mental do engenheiro do conhecimento.

Na fase de ANÁLISE DO "HARDWARE" foi determinado qual o hardware apropriado para o desenvolvimento deste projeto. É importante notar que inicialmente não foi considerado relevante o custo dos equipamentos, pois foram utilizados os equipamentos e instrumentos disponíveis no Laboratório de Ensaios do CEPEL. Para outras aplicações este fator é de vital importância.

KELLER (1987) no seu livro não dá a importância devida à ANÁLISE DO "HARDWARE" durante a especificação técnica do sistema. Isto não é válido em sistemas de automação, onde o "hardware" é uma parte fundamental que pode ser o sucesso ou o fracasso do sistema.

O PROJETO em si teve uma duração de um mês, dado que nas fases de AQUISIÇÃO e REPRESENTAÇÃO DO CONHECIMENTO já é feita uma definição preliminar do projeto.

Deve-se destacar que o desenvolvimento modular do sistema foi de grande vantagem nesta fase, uma vez que cada módulo pode ser definido de forma independente, sem influir na definição dos outros módulos.

As fases de IMPLEMENTAÇÃO e IHM foram desenvolvidas paralelamente, sendo que a IHM deste primeiro protótipo foi desenvolvida somente para testar os conceitos do sistema, não possuindo grandes facilidades gráficas.

Sobre a IMPLEMENTAÇÃO na linguagem Turbo Prolog observa-se que é uma excelente linguagem para desenvolver protótipos. A velocidade do compilador e a capacidade de indicar os erros permitem um ambiente e um desenvolvimento interativo simples, rápido e eficiente.

O TESTE DE DESEMPENHO teve uma duração de um mês, onde foi avaliada a confiabilidade das Agendas geradas pelo *Configurador* de acordo com os parâmetros de Ensaio a calor seco com variação gradual da temperatura para uma amostra que não dissipa calor.

CAPÍTULO VII

CONCLUSÕES E RECOMENDAÇÕES

A metodologia proposta para desenvolver sistemas em configuração de processos em tempo real, aplicada no projeto da automação de Ensaios Climáticos no Laboratório de Ensaios do CEPTEL, permitiu o desenvolvimento de um protótipo no qual foi possível testar uma amostra representativa do conhecimento envolvido neste sistema. Esta aplicação comprovou que as idéias propostas na metodologia têm grandes vantagens em sistemas desta natureza.

Sobre as propostas da metodologia pode-se concluir que:

- o uso de quadros para representar o conhecimento envolvido neste tipo de processos é benéfico. Observa-se a simplicidade com que foi estruturado e gerenciado o conhecimento envolvido neste protótipo;
- não defendemos os quadros como a melhor forma de representar conhecimento, mas podemos dizer que para este tipo de problemas é muito eficiente em termos de representação;
- o desenvolvimento em módulos do sistema permitiu uma maior independência na definição dos mesmos, permitindo uma maior flexibilidade na sua alteração. Por exemplo, a IHM pode ser substituída por uma outra IHM com recursos de computação gráfica, sem necessidade de mudar outros módulos do sistema. Esta é uma recomendação para a futura evolução do protótipo do sistema desenvolvido;

- uma solução dirigida pelos dados, ao invés de algorítmica, é o tipo de abordagem que deve ter o desenvolvimento de um SE. Isto porque geralmente, neste tipo de problemas, é difícil encontrar uma solução que responda a um algoritmo, dado que a solução depende fortemente dos dados inferidos pelo sistema na execução;
- a implementação de um protótipo complexo, onde se possa testar uma grande parte dos conceitos do conhecimento, é uma excelente forma de resolver o problema, porque para o futuro crescimento do sistema sabe-se que não é necessário fazer grandes mudanças no protótipo desenvolvido;
- o uso de um editor de quadros para estruturar o conhecimento facilita muito a tarefa de representação, dado que é possível alterar o esquema da BC sem maiores problemas, mantendo a integridade desta.

Destacam-se como aspectos importantes na aplicação desta metodologia ao protótipo desenvolvido, o fato de que, para o futuro crescimento do sistema, é necessário representar o conhecimento de novas normas, novos instrumentos e novos equipamentos na forma de quadros. Este novo conhecimento não compromete os outros módulos do sistema, dado que a BC não altera sua estrutura. Para isso sugere-se o uso do editor de quadros, o qual comprovou ser uma ferramenta muito eficiente para este propósito.

Na evolução do sistema, ao acrescentar mais equipamentos, instrumentos ou normas, é necessário representar as novas características, indicando os parâmetros que devem ser configurados, como também a forma de serem adquiridos, isto é, na forma de opções ou dados

que têm um escopo definido, ou bem um dado que deve ser inferido pelo MI através de um procedimento "DAEMON" ou herdado de um outro quadro dentro da hierarquia.

Uma fase que não foi mencionada na metodologia apresentada é a Manutenção do Sistema, pelo fato de que uma boa documentação do sistema desenvolvido é uma recomendação importante para a manutenção posterior.

Concluindo pode-se afirmar que o protótipo respondeu satisfatoriamente às necessidades do projeto do CEPEL. Nesta afirmação está inserida uma grande implicação: a viabilidade da aplicação de técnicas de IA na configuração de processos.

Finalmente, conclui-se que esta metodologia não se restringe ao âmbito desta aplicação. Este trabalho é uma contribuição para o desenvolvimento de SE's mais complexos e completos, permitindo sofisticções de recursos, tais como o aprimoramento da IHM com conceitos de computação gráfica, a interação da Operação com a Configuração para permitir configurar e executar EC's dentro de um ambiente totalmente automatizado.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALBUQUERQUE, A. e MENDONÇA, E., (1989), Sistema de Automação do Laboratório de Ensaio Climáticos do Centro de Pesquisas de Energia Elétrica, *Escola de Engenharia, Departamento de Eletrônica, UFRJ*, Rio de Janeiro.
- Associação Brasileira de Normas Técnicas (ABNT), (1981), *Ensaio Básico Climático e Mecânico*, Sistema Nacional de Metrologia, Normalização e Qualidade Industrial, Brasil.
- BABB, M. (1989), New CAD Applications Aid Control System Design, *Control Engineering*, January, pag. 63-64.
- BARR, A., FEIGENBAUM, E. e COHEN, D., (1982), *The Handbook of Artificial Intelligence*, vol. I, II, III, HeurisTech Press, Stanford, Calif.
- BORLAND INTERNATIONAL, (1988a), *Turbo Prolog version 2.0 User's Guide*.
- BORLAND INTERNATIONAL, (1988b), *Turbo Prolog version 2.0 Reference Guide*.
- BRATKO, I. (1986), *Prolog Programming for Artificial Intelligence*, Addison-Wesley.
- BUCHANAN, B. e SHORTLIFFE, E., (1984), *Rule-Based Expert Systems*, Addison-Wesley, Reading, Mass.
- Centro de Pesquisas de Energia Elétrica (CEPEL), (1988a), *Sistemas Digitais Distribuídos para a Automação de Usinas e Subestações, Vol. I - Manual do Hardware*.
- Centro de Pesquisas de Energia Elétrica (CEPEL), (1988b), *Sistemas Digitais Distribuídos para a Automação de Usinas e Subestações, Vol. III - Manual do Software Básico*.
- CHANDRASEKARAN, B. (1986), Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design, *IEEE Expert*, Fall, pag. 23-30.
- CLOCKSIN, W. e MELLISH, C. (1984), *Programming in Prolog*, 2nd edition, Springer-Verlag, Berlin.
- D'AMBROSIO, B., FEHLING, M., FORREST, S., RAULEFS, P. e WILBER, B., (1987), Real-Time Process Management for Materials Composition in Chemical Manufacturing, *IEEE Expert*, Summer, pag. 80-93.

- FALK, H., (1988), AI Techniques enter the realm of Conventional Languages, *Computer Design*, October 15, pag. 45-49.
- FEIGENBAUM, E. (1977), The Art of Artificial Intelligence: Themes and Case Studies in Knowledge Engineering, in *Proc. IJCAI* 5.
- FIKES, R. e KEHLER, T. (1985), The Role of Frame-Based representation in reasoning, *Communication of the ACM*, Setembro, Vol. 28, Num. 9, pag. 904-920.
- FRENZEL, L. (1987), *Crash Course in Artificial Intelligence and Expert Systems*, Howard W. Sams & Co, Indianapolis.
- HALBERT, D. C. e O'BRIEN, P. D. (1987), Using types and Inheritance in Object-Oriented Programming, *IEEE Software*, September, pag. 71-79.
- HARMON, P. e KING, D., (1985), *Expert Systems*, John Wiley&Sons Inc, New York.
- HAYES-ROTH, F., WATERMAN, D. e LENAT, D., (1983), *Building Expert Systems*, vol.1, Addison-Wesley, London.
- HAYES-ROTH, F. (1984a), Knowledge-Based Expert Systems: A Tutorial, *Computer*, IEEE, September.
- HAYES-ROTH, F. (1984b), Knowledge-Based Expert Systems, *Computer*, IEEE, October.
- HEWLETT-PACKARD COMPANY (1981), *Operating and Service Manual, Model, 8165A Programmable Signal Source*, HP-Alemanha.
- HEWLETT-PACKARD COMPANY (1985), *Operating, Programming and Service Manual, Model 54200A/D Digitizing Oscilloscope*, Colorado Spring Division.
- HEWLETT-PACKARD COMPANY (1986), *HP 3457A Multimeter, Operating Manual*, Colorado Spring Division.
- JACKSON, P. (1986), *Introduction to Expert Systems*, Addison-Wesley, London.
- KAISER, G.E. et alia, (1988), Intelligent Assistance for Software Development and Maintenance, *IEEE Software*, May, pag. 40-49.
- KELLER, R. (1987), *Expert System Technology: Development & Application*, Prentice-Hall, New Jersey.
- LANE, A. (1988), Turbo Prolog Revisited, *Byte*, October.
- LEE, N. (1986), Programming with P-Shell. *IEEE Expert*, Summer.

- LEIBSON, S. (1988), Laboratory Automation Software, *Electronic Design*, June.
- LEINWEBER, D. (1987), Expert Systems in Space, *IEEE Expert*, Spring.
- MELO, R. (1988), *Bancos de Dados Não Convencionais: A Tecnologia do BD e suas novas áreas de Aplicação*, VI Escola de Computação, Campinas-SP.
- MINSKY, M. (1975), *A framework for representing knowledge*. In *The Psychology of Computer Vision*. P.Winston, Mc Graw-Hill.
- MITTAL, S., B. CHANDRASEKARAN e J. STICKLEN (1984), Patrec: A Knowledge-Directed Database for a Diagnostic Expert System, *Computer IEEE*, September.
- NAYLOR, C. (1985), *Build your own Expert Systems*, Halsted Press, New York.
- PERKUSICH, A., DE MORAIS, M. e PINTO, F. (1988), Interface Homem-Máquina para aplicação em Automação e Controle de Processos, 7^o CBA, ITA, São José dos Campos-SP.
- RICH, E. (1988), *Inteligência Artificial*, (tradução do original: Artificial Intelligence), Mc Graw-Hill Ltda, São Paulo.
- SALERNO DE AQUINO, M., MONGIOVI, G. e DE MENEZES, H., (1987), EDICON: Um Editor de Conhecimento para SE, 4^o Simposio Brasileiro de Inteligência Artificial, Rio de Janeiro-RJ.
- SIBLEY, E. (1988), Managing Prototype Knowledge/Expert System Projects, *Communication of the ACM*, May, Volume 31, Number 5.
- STEFIK, M. e BOBROW, D. G., (1986), Object Oriented Programming - Themes and Variations, *The AI Magazine*, Vol. 6, num. 4, Winter, pag. 40-62.
- WALKER, A. (1986), Knowledge Systems: Principles and Practice, *IBM Journal Research & Development*, Volume 30, N^o 1, January.
- WATERMAN, D. (1986), *A Guide to Expert Systems*, Addison-Wesley, Reading, Mass.
- WEISS, S. e KULIKOWSKI, C., (1988), *Guia prático para projetar Sistemas Especialistas* (tradução do original: A

practical guide to designing Expert Systems), Livros Técnicos e Científicos Editora S.A., Rio de Janeiro.

WINSTON, P. e HORN, P. (1984), *LISP*, Addison-Wesley, 2^a edição, Reading, Mass.

WONG, W. (1986), Prolog: A Language for Artificial Intelligence, *PC Magazine*, October.

APÊNDICE A: RELAÇÃO DOS COMANDOS DA AGENDA**TABELA A.1: Operação Atingir.**

<i>Parâmetro</i>	<i>Comentário</i>
equipamento	indica o equipamento envolvido
operação	indica ao equipamento que a operação a realizar é de configuração
ciclo	indica o número do ciclo em configuração
temperatura	temperatura que se deseja atingir
tolerância	tolerância permitida para a temperatura exigida
duração	tempo em que se deve atingir a temperatura especificada
tempo máximo	tempo máximo para atingir a temperatura desejada

TABELA A.2: Operação Manter.

<i>Parâmetro</i>	<i>Comentário</i>
equipamento	indica o equipamento envolvido
operação	indica ao equipamento que a operação a realizar é de configuração
ciclo	indica o número do ciclo em configuração
temperatura	temperatura que se deseja manter
tolerância	tolerância permitida, dentro da qual deve-se manter a temperatura
duração	tempo durante o qual se deve manter a temperatura
tempo máximo	é igual à duração, indicando que se deseja manter uma temperatura por um tempo fixo

TABELA A.3: Operação Início.

<i>Parâmetro</i>	<i>Comentário</i>
equipamento	indica o equipamento envolvido
operação	indica ao equipamento início da operação

TABELA A.4: Operação Fim.

<i>Parâmetro</i>	<i>Comentário</i>
equipamento	indica o equipamento envolvido
operação	indica ao equipamento fim da operação

TABELA A.5: Operação SStatus com equipamento climático.

<i>Parâmetro</i>	<i>Comentário</i>
equipamento	indica o equipamento envolvido
operação	indica ao equipamento que se deseja realizar uma operação de status
monitoração	tempo transcorrido entre cada leitura no equipamento

TABELA A.6: Operação SETar.

<i>Parâmetro</i>	<i>Comentário</i>
instrumento	indica o instrumento que se deseja configurar
setup	série de comandos para configurar o instrumento

TABELA A.7: Operação MEDIDA.

<i>Parâmetro</i>	<i>Comentário</i>
instrumento	indica o instrumento que deve realizar a medição
tipo de medida	identifica um tipo de medida válida para os instrumentos
tempo	tempo durante o qual se deve realizar a medida no instrumento
período	tempo transcorrido entre cada medida
canal	identifica o canal do instrumento a ser utilizado na aquisição do dado

TABELA A.8: Operação Clear.

<i>Parâmetro</i>	<i>Comentário</i>
instrumento	indica o instrumento que se deseja limpar

TABELA A.9: Operação SStatus com instrumento.

<i>Parâmetro</i>	<i>Comentário</i>
instrumento	indica o instrumento que se deve verificar

TABELA A.10: Operação Inspeção Visual.

sem parâmetros

TABELA A.11: Operação Ligar.

<i>Parâmetro</i>	<i>Comentário</i>
desde	indica o que se deve ligar
para	indica onde deve ligar

TABELA A.12: Código das Operações.

Operação	Código
Atingir	"A"
Manter	"A"
Início	"I"
Fim	"F"
STatus equipamento	"ST"
SEtar	"SE"
MEdir	"ME"
cleaR	"R"
Status instrumento	"S"
inspeção Visual	"V"
Ligar	"L"
Colocar	"C"

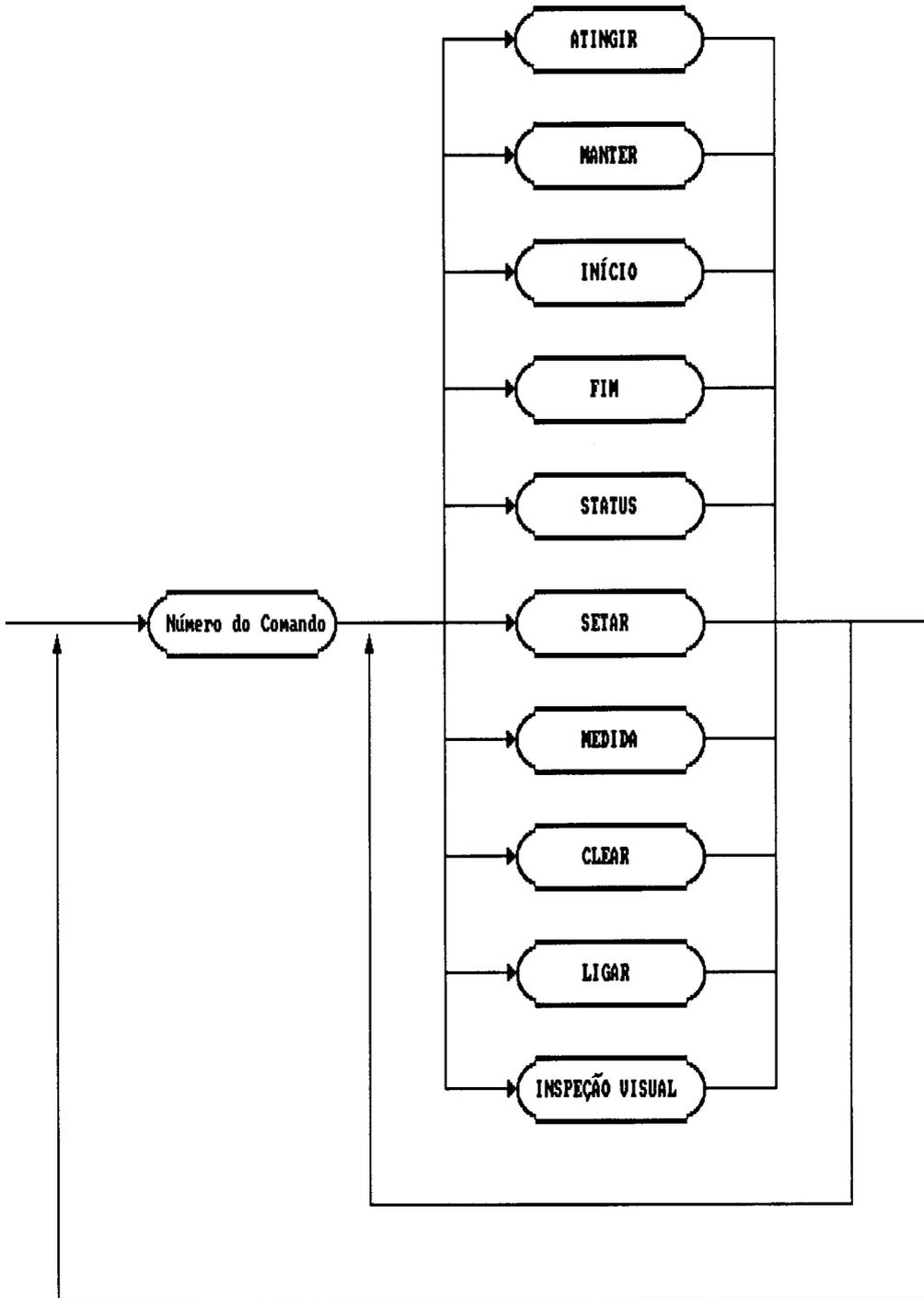
TABELA A.13: Código dos Parâmetros.

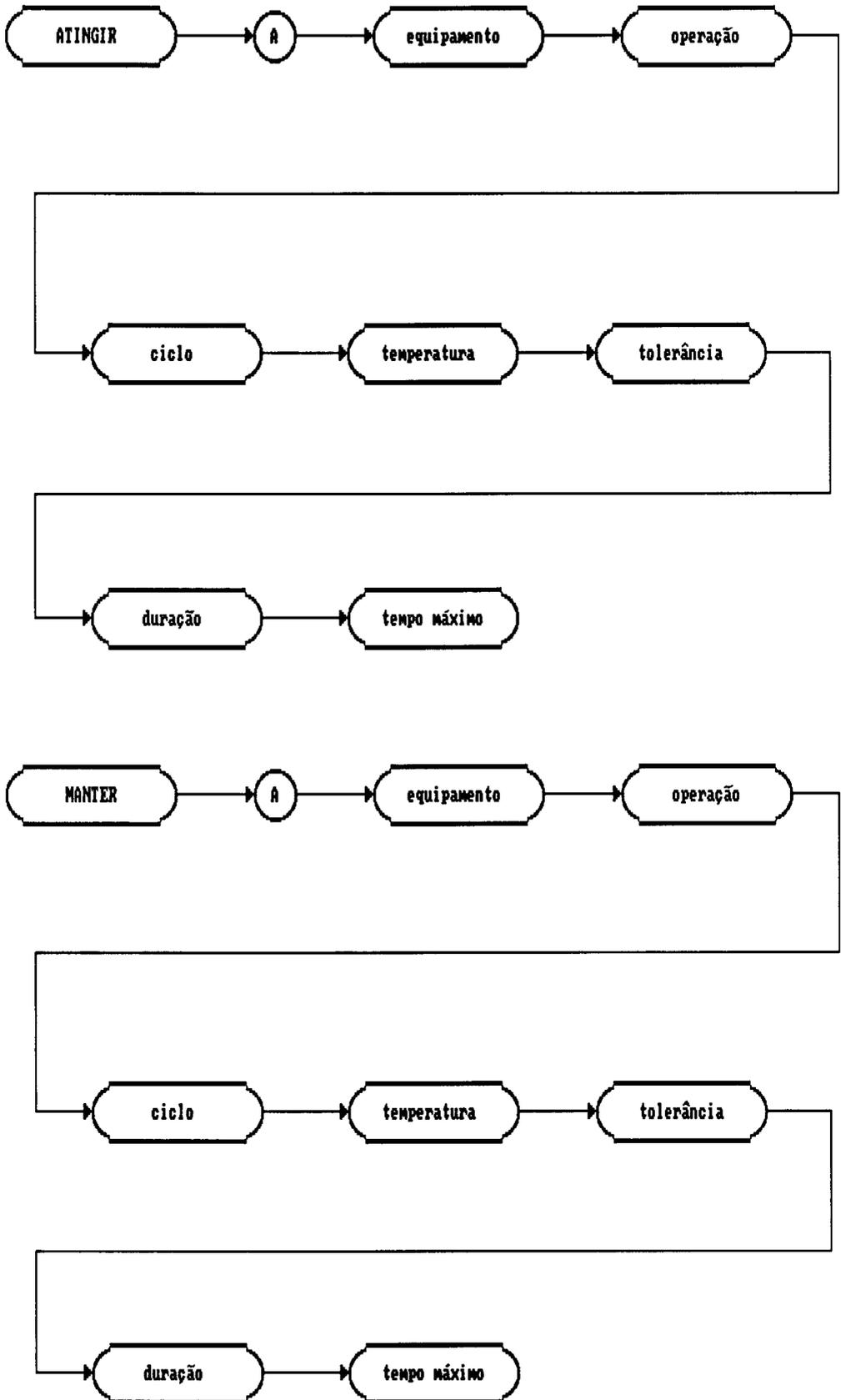
Parâmetro	Código
equipamento	"eq"
operação	"op"
ciclo	"ci"
temperatura	"te"
tolerância	"to"
duração	"du"
tempo máximo	"tx"
monitoração	"mo"
instrumento	"in"
setup	"sp"
tipo medida	"tm"
tempo	"tp"
período	"pe"
canal	"ca"
desde	"dd"
para	"pa"

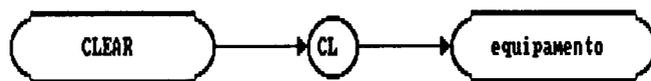
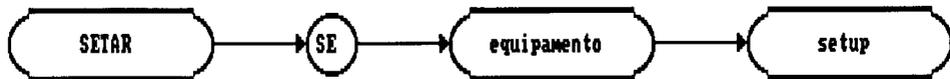
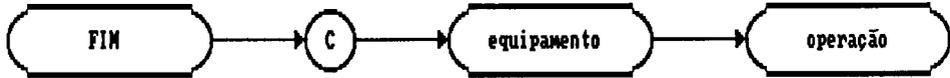
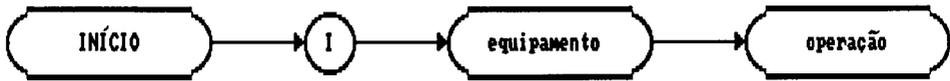
TABELA A.14: Código dos Valores.

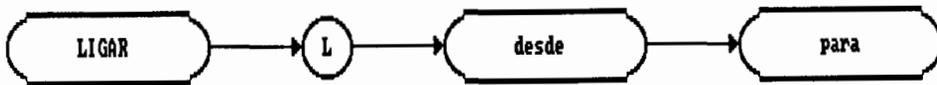
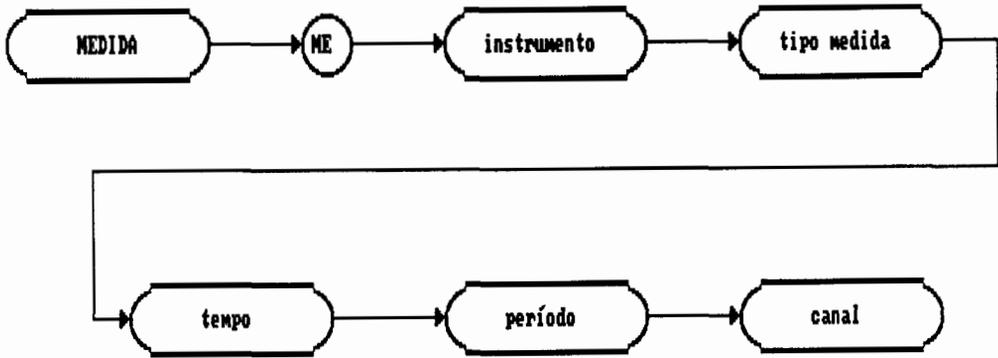
Parâmetro	Valores	Código
equipamento	estufa	"0"
operação	Configuração	"C"
	Status	"S"
	Início	"I"
	Fim	"F"
instrumento	Osciloscópio	"O"
	Multímetro	"M"
	Sintetizador	"S"
tipo medida	Frequência	"F"
	Tensão	"V"
	período	"S"
	resistência	"O"
	corrente	"A"
	valor Rms	"R"
	temperatura	"C"
	foto	"P"
revelação	"W"	
canal	Trigger	"0"
	canal 1	"1"
	canal 2	"2"
	canal 3	"3"
	canal 4	"4"
	canal 5	"5"
	canal 6	"6"
	canal 7	"7"
	canal 8	"8"
	canal 9	"9"

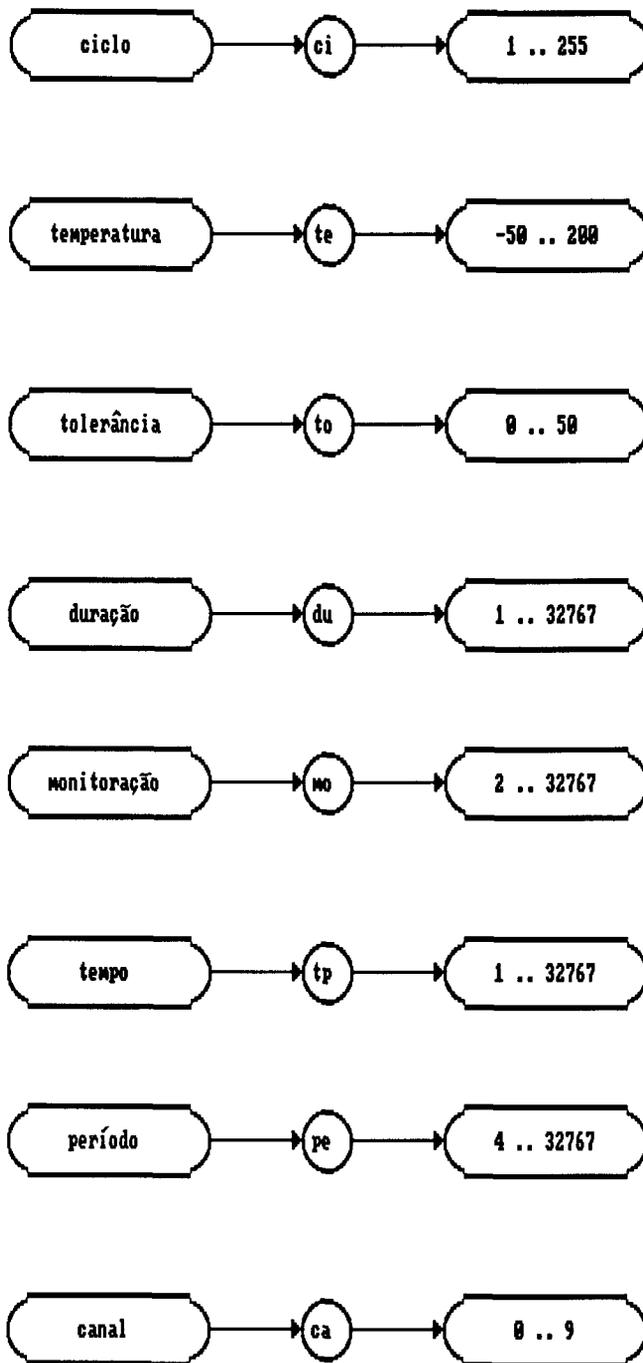
APÊNDICE B: SINTAXE DOS COMANDOS DA AGENDA

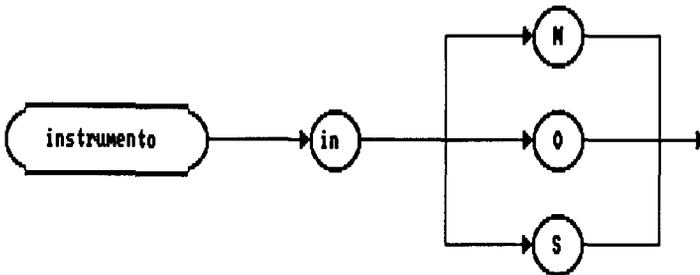
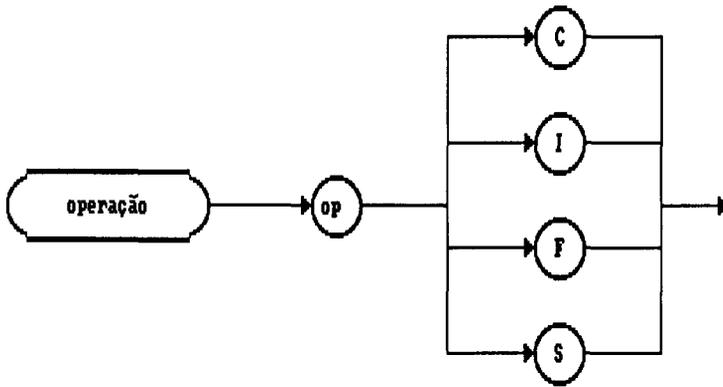
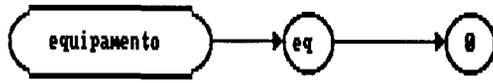


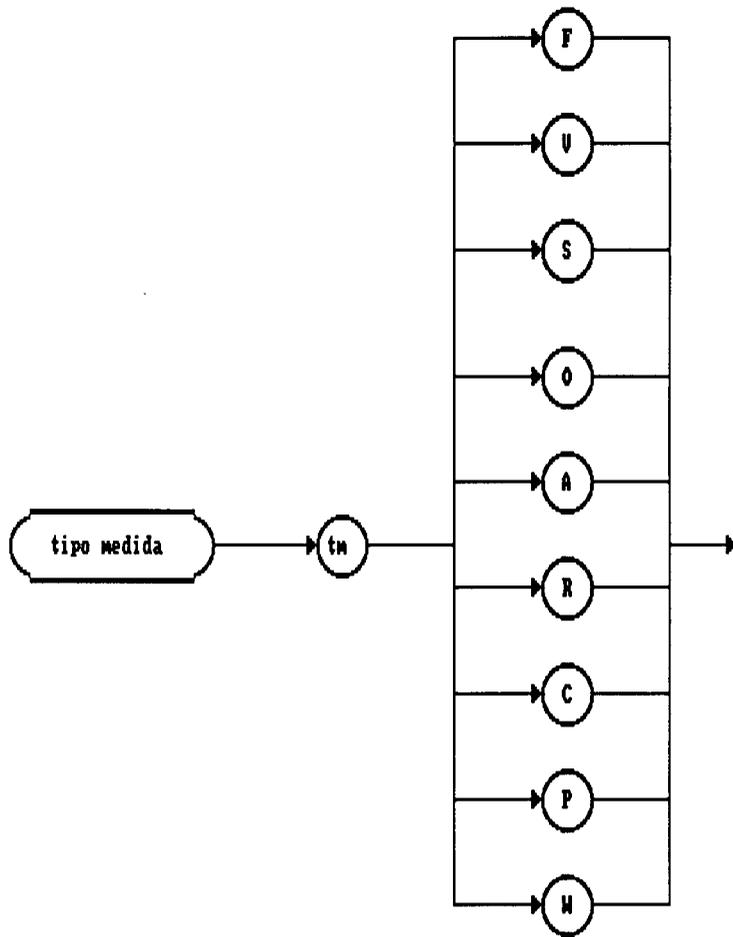












APÊNDICE C: PARÂMETROS DO OSCILOSCÓPIO

TABELA C.1: CHANNEL do osciloscópio.

PROBe: (1,2,5,10,20,50,100)		
RANGe:	PROBe	RANGe p/divisão
	1	40 mV a 40 V
	2	80 mV a 80 V
	5	200 mV a 200 V
	10	400 mV a 400 V
	20	800 mV a 800 V
	50	2 V a 2000 V
	100	4 V a 4000 V
OFFSet:	RANGe	OFFSet
	40 mV .. 390 mV	± 2 V
	400 mV .. 40 V	±20 V
COUPling: (AC,DC)		
LABel: ("10 caracteres")		
STORe: (NORmal, AVERage, ENVelope)		
	.AVERage: é preciso mais um parâmetro que pode ser (4,16,64,256)	
	.ENVelope: mais um parâmetro de (10 a 10000)	
	.NORmal: parâmetro=1	
SCALE: (DISabled, ENABled).		

TABELA C.2: TIME base do osciloscópio.

Sweep MODE : (AUTO, TRIGger, SINGled)		
RANGe: (50E-9 .. 10 seg)		
DELay:	RIGHT	LEFT
RANGe	Pre Trigger	Pós Trigger
50 pseg..5micseg	até 5micseg	até 1 mseg
10E-6..10	até 10	até 260
SCALE: DISabled, PERiod, POSitive pulse, NEGative pulse, RISE, FALL		
REFerence: (LEFT, RIGHT, CENTER) (vide DELay)		
ALIAStest: (ON, OFF)		

TABELA C.3: TRIGger do osciloscópio.

SOURce: (CHANnel (1,2), EXTernal)		
PROBe: (1,2,5,10,20,50,100)		
RANGe:	PROBe	RANGe p/divisão
	1	40 mV a 40 V
	2	80 mV a 80 V
	5	200 mV a 200 V
	10	400 mV a 400 V
	20	800 mV a 800 V
	50	2 V a 2000 V
	100	4 V a 4000 V
OFFSet:	RANGe	OFFSet
	40 mV .. 390 mV	± 2 V
	400 mV .. 40 V	±20 V
SLOPe: (POSitive, NEGative)		
COUPling: (AC,DC)		
LABel: ("10 caracteres")		
STORe: (NORmal, AVERage, ENVelope)		
	.AVERage: é preciso mais um parâmetro que pode ser (4,16,64,256)	
	.ENVelope: mais um parâmetro de (10 a 10000)	
	.NORmal: parâmetro=1	
SCALE: (DISabled, ENABled).		

APÊNDICE D: IMPLEMENTAÇÃO DO PROTÓTIPO EM PROLOG

D.1 Definição do Domínio

A seguinte definição em Turbo Prolog corresponde à estrutura dos quadros:

```
% domínio da representação do conhecimento
FRAME   = SLOTS*    % é uma lista de "slots".
SLOTS   = STRINGLIST* % é uma lista de stringlist.
STRINGLIST = STRING* % é uma lista de "string".
```

onde "string" por definição interna do compilador é uma lista de caracteres.

D.2 Mecanismo de Inferência

O MI implementado em Turbo Prolog tem a seguinte estrutura, onde `select_ensaio` seleciona o ensaio e `go_config` inicia a configuração:

```
go_config:- select_ensaio(ENSAIO),!,go_config(ENSAIO),!.
go_config:- !.
```

`select_ensaio` apresenta os ensaios climáticos conhecidos, para selecionar a sua norma.

```
select_ensaio(ENSAIO):-
    verifica no quadro "ENSAIOS" se o ensaio já foi configurado pelo sistema. Esse
    conhecimento está no slot "CONFIGURAR ENSAIOS", !.
select_ensaio(ENSAIO):-
    obtém do quadro "ENSAIOS" os ensaios conhecidos pelo sistema. Esses ensaios estão no
    slot "CONHECIDOS", !,
    apresenta na forma de menu os ensaios que o usuário pode selecionar. Retorna o ensaio
    selecionado, ou ABORT em caso de aborto,
    ENSAIO<>"ERRO", !,
    cria no quadro "ENSAIOS" um slot "CONFIGURAR ENSAIO" com o valor do ensaio
    selecionado.
select_ensaio("ABORT"):- !.
```

go_config(String) seleciona a norma do ensaio selecionado, seleciona o tipo do ensaio e logo inicia a configuração do ensaio:

```
go_config("ABORT):-!, fail.
go_config(ENSAIO):-
  obtem do quadro "ENSAIO" no slot "CONFIGURAR ENSAIO" o ensaio selecionado pelo
  usuário,
  select_normas(ENSAIO,NORMA),
  NORMA<>"ABORT",
  select_tipo(NORMA,Tipo),
  Tipo<>"ERRO",!,
  config_ensaio(NORMA),!.
go_config(_):-!.
```

A seguinte corresponde à regras de seleção da norma, segundo as características definidas pelo usuário:

```
if (ensaio="calor seco")and(amostra nao dissipa)and
  (variacao gradual de temperatura)
then (seleciona NBR 6796).
```

```
if (ensaio="calor seco")and(amostra dissipa)and
  (variacao gradual de temperatura)
then (seleciona NBR 6798).
```

```
if (ensaio="calor seco")and(amostra dissipa)and
  (variacao rapida de temperatura)
then (seleciona NBR 6797).
```

```
if (ensaio="calor seco")and(amostra nao dissipa)and
  (variacao rapida de temperatura)
then (seleciona NBR 6819).
```

select_tipo pergunta pelo tipo de ensaios que a Norma selecionada aceita:

```
select_tipo(_,Tipo):-
  obtem(["VALOR" | [Tipo]],["ENSAIOS"],["tipo"],"VALOR"),!.
select_tipo(NORMA,Tipo):-
  splitfilename(NORMA,Nm,_),
  obtem(["OPCOES" | Dados],[Nm],["tipo"],"OPCOES"),
  ihm_msg(3,Nm,"tipo"),
  ihm_menu(Dados,"Tipo ?",Nm,"tipo",Tipo),!,
  cria_([Nm],["tipo"],"VALOR",[Tipo]),
  cria_(["ENSAIOS"],["tipo"],"VALOR",[Tipo]),!.
select_tipo(_,"ERRO):-!.
```

config_ensaio pergunta pelas características do ensaio para selecionar o tipo do ensaio:

```

config_ensaio():-
  obtem(["VALOR" | _],[["ENSAIOS"],["CONFIGURADOENSAIO"],"VALOR"),!.
config_ensaio(NORMA):-
  splitfilename(NORMA,Nm,_),
  obtem(["VALOR" | ListaEtapas],[[Nm]],["SUBCLASSES"],"VALOR"),
  cria(_,[["ENSAIOS"],["CONFIGURAR ETAPAS"],"VALOR",ListaEtapas),!,
  config_etapa(ListaEtapas),!.
config_ensaio(NORMA):-
  splitfilename(NORMA,Nm,_),
  obtem(["VALOR" | ListaEtapas],[[Nm]],["SUBCLASSES"],"VALOR"),
  apaga(_,[["ENSAIOS"],["CONFIGURAR ETAPAS"]]),!,
  cria(_,[["ENSAIOS"],["SUBCLASSES"],"VALOR",ListaEtapas),!,
  cria(_,[["ENSAIOS"],["CONFIGURAR ETAPAS"],"VALOR",ListaEtapas),!,
  config_etapa(ListaEtapas),!.
config_ensaio("ERRO):-!.

```

Cada norma possui diferentes etapas a serem configuradas. Em cada etapa devem ser adquiridos os dados que são necessários para completar a configuração:

```

config_etapa():-
  repeat,
  obtem(["VALOR" | EtapasLst],[["ENSAIOS"],["CONFIGURAR ETAPAS"],"VALOR"),
  ihm_menu(EtapasLst,"Etapas","ENSAIOS","CONFIGURAR ETAPAS",ETAPA),
  ETAPA < > "ERRO", configura(ETAPA),
  seguinte_etapa(ETAPA,Rest), Rest = [],!,
  obtem(["VALOR" | ENSAIO],[["ENSAIOS"],["CONFIGURAR ENSAIO"],"VALOR"),!,
  cria(_,[["ENSAIOS"],["CONFIGURADO ENSAIO"],"VALOR",ENSAIO),!,
  apaga(_,[["ENSAIOS"],["CONFIGURAR ENSAIO"],"VALOR"),!.

```

```

seguinte_etapa(ETAPA,Rest):-
  apaga(_,[["ENSAIOS"],["CONFIGURAR ETAPAS"],"VALOR",ETAPA),
  obtem(["VALOR" | Rest],[["ENSAIOS"],["CONFIGURAR ETAPAS"],"VALOR"),!,
  cria(_,[["ENSAIOS"],["CONFIGURADAETAPAS"],"VALOR",[ETAPA]),
  apaga(_,[["6796"],["SUBCLASSES"],"VALOR",ETAPA),!.

```

```

seguinte_etapa(ETAPA,[]):-!,
  cria(_,[["ENSAIOS"],["CONFIGURADA ETAPAS"],"VALOR",[ETAPA]),!,
  apaga(_,[["6796"],["SUBCLASSES"],"VALOR"),!.

```

```

caracteristicas("6817.NBR",NORMA):-
  not(obtem(["VALOR" | _],[["6817"],["DISSIPACAO"],"VALOR"]),
  ihm_msg(3,"6817","dissipacao"),
  obtem(["dissipa" | Lista],[["6817"],["divisao"],"dissipa"),
  ihm_menu(Lista,"Dissipa ?","6817","divisao",Dissipacao),
  cria(_,[["6817"],["DISSIPACAO"],"VALOR",[Dissipacao]),!,
  caracteristicas("6817.NBR",NORMA).

```

```

caracteristicas("6817.NBR",NORMA):-
  not(obtem(["VALOR" | _],["6817"],["VARIACAO"],"VALOR")),
  obtem(["VALOR" | Dissipacao],["6817"],["DISSIPACAO"],"VALOR"),
  obtem(["variacao" | Dados],["6817"],[Dissipacao],"variacao"),
  ihm_msg(3,"6817","variacao"),
  ihm_menu(Dados,"Variacao ?","6817",Dissipacao,Variacao),!,
  cria_([["6817"],["VARIACAO"],"VALOR",[Variacao]]),
  carrega_norma("calor seco",Dissipacao,Variacao,NORMA),!.
caracteristicas("6817.NBR","ABORT"):-!.

```

Conhecendo o nome do ensaio e suas características, `carrega_norma` carrega o arquivo que contém a norma que mais se adequa:

```

carrega_norma("calor seco", Dissipa, Variacao,NBR):-
  obtem_([["6817"],[Variacao],Dissipa,NBR]),
  concat(norma_path,NBR,Arquivo),
  trap(consult(Arquivo,frames_dba),_,fail),!,
  mostra_descricao(NBR).
carrega_norma("calor seco",_,_, "ABORT"):-!,
  msg_wait(1,"nNÃO CARREGOU NORMA\n").

```

Uma vez realizada a configuração do ensaio é iniciada a tradução da Agenda gerando o código do ensaio a ser executado:

```

traduz_agenda:-
  obtem([_,Nor],["ENSAIOS"],["NORMA SELECIONADA"],"VALOR"),
  splitfilename(Nor,NORMA,_),
  obtem([_ | LST],[[NORMA]],["ETAPAS"],"VALOR"),
  cria_([["ENSAIOS"],["ETAPAS"],"VALOR",LST]),
  botasequencia(LST),
  trap(consult(codigo_bc,codigo_dba),_,fail),
  geralIntermedio,!,
  trap(save("c:\inter.dba",frames_dba),_,fail),!,
  retractall(_ ,codigo_dba),
traduz_agenda:- !, msg_wait(1,"nAGENDA ainda não gerada\n").

```

```

botasequencia({}):-!.
botasequencia([H | T]):-
  obtem([_, "nao"],[[H]],["PRECISA"],"VALOR"),
  apaga_([H]),
  apaga_([["ENSAIOS"],["ETAPAS"],"VALOR",H]),!,
  botasequencia(T).
botasequencia([_ | T]):- !, botasequencia(T).

```

D.3 Interface Homem-Máquina.

O módulo de Aquisição dos dados foi implementado da seguinte forma:

```

index1([H|_],N,N,H):-!.
index1([_|T],N,N1,X):- N1 = N + 1, index1(T,N1,N2,X).

ihm_menu([],_,_,_, "ERRO"):-!.
ihm_menu(.,_,Fr,Sl,Valor):- obter(["VALOR" | [Valor]], [[Fr]], [Sl], "VALOR"),!.
ihm_menu(DadLst,TITULO,Fr,Sl,SELECT):-
  obter(., [[Fr]], [Sl], "DEFAULT", STR),
  index1(DadLst, 1, INDEX, STR),
  menu(4, 1, 32, 107, DadLst, TITULO, INDEX, CHOICE),
  index(DadLst, CHOICE, SELECT),!.
ihm_menu(Lst,TIT,_,_,SL):-
  menu(4, 1, 32, 107, Lst, TIT, 1, CH), index(Lst, CH, SL),!.
ihm_menu(.,_,_,_, "ERRO"):-!.

ihm_opcoes(.,Fr,Sl,Valor):- obter(["VALOR" | [Valor]], [[Fr]], [Sl], "VALOR"),!.
ihm_opcoes(OPCOES,Fr,Sl,SELECT):-
  obter(., [[Fr]], [Sl], "DEFAULT", STR),
  index1(OPCOES, 1, INDEX, STR), !,
  obter(., [[Fr]], [Sl], "unidade", BASE),
  concatlist([Sl, " (" ,BASE, ")"], TITULO),
  menu(4, 1, 32, 107, OPCOES, TITULO, INDEX, CHOICE),
  index(OPCOES, CHOICE, SELECT),!.
ihm_opcoes(OPCOES,Fr,Sl,SELECT):-
  obter(., [[Fr]], [Sl], "unidade", BASE),
  concatlist([Sl, " (" ,BASE, ")"], TITULO),
  menu(4, 1, 32, 107, OPCOES, TITULO, 1, CHOICE),
  index(OPCOES, CHOICE, SELECT),!.
ihm_opcoes(Lst,_,TIT,SL):-
  menu(4, 1, 32, 107, Lst, TIT, 1, CH), index(Lst, CH, SL),!.
ihm_opcoes(.,_,_, "ERROR"):-!.

ihm_faixa(.,_,Fr,Sl,Valor):- obter(["VALOR" | [Valor]], [[Fr]], [Sl], "VALOR"),!.
ihm_faixa(INF,SUP,Fr,Sl,SELECT):-
  obter(., [[Fr]], [Sl], "DEFAULT", STR),
  obter(., [[Fr]], [Sl], "unidade", BASE),
  concatlist([Sl, "(min = ", INF, "; max = ", SUP, " ", BASE, "):"], TIT),
  lineinput(20, 2, 75, 25, 40, TIT, STR, Result),
  valido(INF, SUP, Result, SELECT),!.
ihm_faixa(INF,SUP,Fr,Sl,SELECT):- obter(., [[Fr]], [Sl], "unidade", BASE),!.
  concatlist([Sl, "(min = ", INF, "; max = ", SUP, " ", BASE, "):"], TIT),
  lineinput(20, 2, 75, 25, 40, TIT, "", Result),
  valido(INF, SUP, Result, SELECT),!.
ihm_faixa(INF,SUP,_,Sl,SELECT):-
  concatlist([Sl, "(min = ", INF, "; max = ", SUP, " ) :"], TITULO),
  lineinput(20, 2, 75, 25, 40, TITULO, "", Result),
  valido(INF, SUP, Result, SELECT),!.

```

```

valido(INF,SUP,Result,Result1):-
  str_real(INF,Inf1),
  str_real(SUP,Sup1),
  str_real(Result,Result0),
  Result0 >= Inf1, Result0 <= Sup1,
  str_real(Result1,Result0),!.
valido(,,):-!,
  msg_wait(1,"\nDados fora dos\nLimites Max e Min.\n"),
  fail.

```

```

mostra_descricao(NBR):-
  splitfilename(NBR,Nm,_),
  obtem(,[[Nm]],["descricao"],"VALOR",TEXT),!,
  concatlist(["DESCRICAÇÃO",Nm,TEXT],TEXT),
  msg_wait(2,TEXT).

```

```

ihm_select(1,[Instrumento],_,_Instrumento):-!.
ihm_select(2,InstLst,Fr,Sl,Instrum):-
  ihm_opcoes(InstLst,Fr,Sl,Instrum),!.

```

```

ihm_window(WINDOW,TITULO):-
  shiftwindow(OLD), shiftwindow(WINDOW),
  makewindow(,Fr,Cor,_,X,Y,X1,Y1), removewindow,
  makewindow(WINDOW,Fr,Cor,TITULO,X,Y,X1,Y1),
  shiftwindow(OLD).

```

```

ihm_msg(Win,Fr,Tipo):-
  obtem(,[[Fr]],["COMENTÁRIO"],Tipo,Msg),!,
  msg(Win,Msg).

```

D.4 Geração da Agenda

O módulo Gerador da Agenda tem a seguinte estrutura:

```

replace(numCiclo(_)):- retract(numCiclo(_)),fail.
replace(numComando(_)):- retract(numComando(_)), fail.
replace(ultimaCon(_)):- retract(ultimaCon(_)), fail.
replace(T):- asserta(T),!.

```

```

incrementa(numCiclo(_,NStr1):-
  numCiclo(N),!,
  N1 = N + 1, str_int(NStr1,N1),
  replace(numCiclo(N1)).
incrementa(numComando(_,NStr1):-
  numComando(N),!, N1 = N + 1, str_int(NStr1,N1),
  replace(numComando(N1)).

```

geralIntermedio:-

```

asserta(numComando(2)), asserta(numCiclo(1)),
cria(_,[["1"],["F"],"eq",["0"]],
ihm_window(2,"*** GERANDO AGENDA ***"),
repeat,
obtem(_ | LstEtapas],[["ENSAIOS"],["ETAPAS"],"VALOR"),
LstEtapas=[Etapa | _],
geraSlot(Etapa),
apaga(_,[["ENSAIOS"],["ETAPAS"],"VALOR",Etapa],
seguinte("ENSAIOS","ETAPAS","VALOR",LstRest),
incrementa(numCiclo(_),_),
LstRest=[],!.

```

geraSlot(Etapa):-

```

repeat,
obtem(["VALOR" | NumComand],[[Etapa]],["NUMERO COMANDOS"],"VALOR"),
NumComand=[Num | Rest],
obtem(["cmd",CMD],[[Etapa]],[Num],"cmd"),
geraPre(Etapa,CMD), geraDado(Etapa,Num,CMD), geraPos(Etapa,CMD),
apaga(_,[[Etapa]],["NUMERO COMANDOS"],"VALOR",Num),
apaga(_,[[Etapa]],[Num]),
seguinte(Etapa,"NUMERO COMANDOS","VALOR",RestComand),
incrementa(numComando(_),_),
RestComand=[],!.

```

geraDado(Etapa,Num,CMD):-

```

repeat,
obtem(["PARAMETROS" | LstParam],[[Etapa]],[Num],"PARAMETROS"),
LstParam=[Param | Rest],
obtem([Param | Tipo],[[Etapa]],[Num],Param),
obtemTipo(Etapa,Num,Param,Tipo,CMD),
apaga(_,[[Etapa]],[Num],"PARAMETROS",Param),
seguinte(Etapa,Num,"PARAMETROS",RestParam),
RestParam=[],!.

```

obtemTipo(Etapa,Num,Param,["OBTER" | _],CMD):-

```

obtemdefault(Etapa,Param,Dado),!,
codslot(Param,Par), codslot(CMD,CMDCOD),
numComando(N), str_int(NS,N),
codDado(Dado,DadCod),
cria(_,[["NS"]],[CMDCOD],Par,[DadCod]),!.

```

obtemTipo(Etapa,Num,Param,["OBTEM",NomeSlot],CMD):-

```

obtem(_ | Dado],[[Etapa]],[NomeSlot],Param),!,
codslot(Param,Par),
codslot(CMD,CMDCOD),
numComando(N), str_int(NS,N),
codDado(Dado,DadCod),
cria(_,[["NS"]],[CMDCOD],Par,[DadCod]),!.

```

obtemTipo(Etapa,Num,Param,["IRMAO",NomeEtapa],CMD):-

```

obtemdefault(NomeEtapa,Param,Dado),!,
codslot(Param,Par),
codslot(CMD,CMDCOD),
numComando(N), str_int(NS,N),
codDado(Dado,DadCod),
cria(_,[["NS"]],[CMDCOD],Par,[DadCod]),!.

```

```

obtemTipo(Etapa,Num,Param,["HERDAR" | _],CMD):-!,
  obtem([_,Pai],[[Etapa]],["MEMBRO DE"],"VALOR"),!,
  obtemdefault(Pai,Param,Dado),
  codslot(Param,Par),
  codslot(CMD,CMDCOD),
  numComando(N), str_int(NS,N),
  codDado(Dado,DadCod),
  cria([_[NS]], [CMDCOD],Par,[DadCod]).
obtemTipo(Etapa,Num,Param,["CALCULAR SUBIDA",ABAIXO,ACIMA,TEMP],CMD):-!,
  obtemdefault(ABAIXO,TEMP,DAD1), str_int(DAD1,TMP1), !,
  obtemdefault(ACIMA,TEMP,DAD2), str_int(DAD2,TMP2),
  TEMPO=TMP2-TMP1, str_int(TEMPS,TEMPO),
  codslot(Param,Par),
  codslot(CMD,CMDCOD),
  numComando(N), str_int(NS,N),
  cria([_[NS]], [CMDCOD],Par,[TEMPS]).
obtemTipo(Etapa,Num,Param,["CALCULAR DESCIDA",ACIMA,ABAIXO,TEMP],CMD):-!,
  obtemdefault(ACIMA,TEMP,DAD1), str_int(DAD1,TMP1), !,
  obtemdefault(ABAIXO,TEMP,DAD2), str_int(DAD2,TMP2),
  TEMPO=(TMP1-TMP2)*4, str_int(TEMPS,TEMPO),
  codslot(Param,Par),
  codslot(CMD,CMDCOD),
  numComando(N), str_int(NS,N),
  cria([_[NS]], [CMDCOD],Par,[TEMPS]).
obtemTipo(Etapa,Num,Param,["CONFIG" | _],CMD):-!,
  codslot(Param,Par),
  codslot(CMD,CMDCOD),
  numCiclo(CIC), str_int(C,CIC),
  numComando(N), str_int(NS,N),
  cria([_[NS]], [CMDCOD],Par,[C]).
obtemTipo(Etapa,Num,Param,["CALCMAX" | _],"ATINGIR"):-
  numComando(N), str_int(NS,N),!,
  obtem(["du",Temp],[[NS]],["A"],"du"), str_int(Temp,Tempo),
  Tempo10 = Tempo + 10, str_int(TempoS,Tempo10),
  cria([_[NS]],["A"],"bx",[TempoS]).
obtemTipo(Etapa,Num,Param,["CALCMAX" | _],"MANTER"):-
  numComando(N), str_int(NS,N),!,
  obtem(["du",Temp],[[NS]],["A"],"du"),
  cria([_[NS]],["A"],"bx",[Temp]).
obtemTipo(Etapa,Num,Param,["0" | _],CMD):-
  codslot(CMD,CMDCOD),
  numComando(N), str_int(NS,N),
  assertz(frames([[_[NS]], [CMDCOD]])),!.
obtemTipo(Etapa,Num,Param,_,CMD):-
  obtem([Param,Dad],[[Etapa]], [Num],Param),!,
  codslot(Param,Par),
  codslot(CMD,CMDCOD),
  numComando(N), str_int(NS,N),
  codDado(Dad,DadCod),
  cria([_[NS]], [CMDCOD],Par,[DadCod]),!.
obtemTipo(.,.,.,[.],.):-!.

```

```
codDado(D,Cod):- codslot(D,Cod),!.
```

```
codDado(D,D):-!.
```

geraPre(Etapa,"MEDIR):-

% OBTÉM O FRAME COM A ÚLTIMA CONFIGURAÇÃO FEITA NO EQUIPAMENTO
% PARA OBTER OS DADOS PARA O COMANDO DE MEDIÇÃO

numComando(N), str_int(NS,N),

ultimaCon([H|FR]), assertz(frames([[NS]|FR])),

% INSERE OS COMANDOS DE INÍCIO DE OPERAÇÃO NO EQUIPAMENTO

% E COMANDO DE STATUS, AMBOS COMO COMANDOS EM PARALELO COM

% O COMANDO DE CONFIGURAÇÃO.

obtem(["eq",EQ],[[NS]],["A"],"eq"),!,

cria_([[NS]],["I"],"eq",[EQ]), cria_([[NS]],["I"],"op",["I"]),

cria_([[NS]],["ST"],"eq",[EQ]), cria_([[NS]],["ST"],"op",["S"]),

obtemdefault(Etapa,"monitor",MO),!,

cria_([[NS]],["ST"],"mo",[MO]).

geraPre(_):-!.

geraPos(Etapa,"MANTER):-!, geraPos(Etapa,"ATINGIR"),!.

geraPos(Etapa,"ATINGIR):-

% CRIA O FRAME COM A ÚLTIMA CONFIGURAÇÃO FEITA NO EQUIPAMENTO

% PARA OBTER OS DADOS EM CASO DE TER APÓS UM COMANDO DE MEDIÇÃO

numComando(N), str_int(NS,N),

obtem([[NS]|Rest],[[NS]],!, asserta(frames([[["ULTIMO"]]|Rest])),

apaga_([[["ULTIMO"]]],["A"],"du"), apaga_([[["ULTIMO"]]],["A"],"ci"),

apaga_([[["ULTIMO"]]],["A"],"bx"),

obtem(FR,[["ULTIMO"]]),!, apaga_([[["ULTIMO"]]]),

replace(ultimaCon(FR)),

% INSERE OS COMANDOS DE INÍCIO DE OPERAÇÃO NO EQUIPAMENTO

% E COMANDO DE STATUS, AMBOS COMO COMANDOS EM PARALELO COM

% O COMANDO DE CONFIGURAÇÃO.

obtem(["eq",EQ],[[NS]],["A"],"eq"),!,

cria_([[NS]],["I"],"eq",[EQ]), cria_([[NS]],["I"],"op",["I"]),

cria_([[NS]],["ST"],"eq",[EQ]), cria_([[NS]],["ST"],"op",["S"]),

obtemdefault(Etapa,"monitor",MO),!, cria_([[NS]],["ST"],"mo",[MO]),

% INSERE O COMANDO DE FIM DE OPERAÇÃO NO EQUIPAMENTO

% DEPOIS DO COMANDO STATUS

incrementa(numComando(_),NS1),

cria_([[NS1]],["F"],"op",["F"]),!, cria_([[NS1]],["F"],"eq",[EQ]).

geraPos(_,"MEDIR):-

% OBTÉM O TEMPO DE MEDIÇÃO PARA INSERIR ESSE TEMPO NO COMANDO DE STATUS

numComando(N), str_int(NS,N), obtem(["tp",TP],[[NS]],["ME"],"tp"),!,

cria_([[NS]],["A"],"du",[TP]), cria_([[NS]],["A"],"bx",[TP]),

numCiclo(CIC), str_int(CI,CIC),

cria_([[NS]],["A"],"ci",[CI]),

% OBTÉM O EQUIPAMENTO ONDE FOI FEITA A ÚLTIMA CONFIGURAÇÃO PARA

% INSERIR O COMANDO DE FIM APÓS DO COMANDO EM PARALELO DE MEDIÇÃO

% E STATUS NO EQUIPAMENTO.

incrementa(numComando(_),NS2),

obtem(["eq",EQ],[[NS]],["A"],"eq"),!,

cria_([[NS2]],["F"],"eq",[EQ]), cria_([[NS2]],["F"],"op",["F"]).

geraPos(_):-!.

seguinte(Fr,Sl,Tip,Rest):- obtem([Tip|Rest],[[Fr]],[Sl],Tip),!.

seguinte(Fr,Sl,Tip,[]):-!.

geragend:-

frames([[NS]|Rest]), apaga([[NS]|Rest],[[NS]]), str_int(NS,N),

assertz(agenda([[NS]|Rest]),agenda_dba), not(frames(_)),!.

geragend:-!.

D.5 Base de Conhecimento

```

configura(Etapa):- obtem(_,[["ENSAIOS"]],["CONFIGURADA ETAPAS"],"VALOR",Etapa),!.
configura(Etapa):-
  not(obtem(["VALOR" | _],[[Etapa]],["PRECISA"],"VALOR")),
  obtem(Lst,[[Etapa]],["PRECISA"],"OPCOES",_), ihm_opcoes(Lst,Etapa,"PRECISA",DECISAO),
  cria(_,[[Etapa]],["PRECISA"],"VALOR",[DECISAO]), DECISAO="sim",!,
  cria(_,[["6796"]],["ETAPAS"],"VALOR",[Etapa]),!, configura(Etapa).
configura("PRECONDICIONAMENTO"):-
  obtem(["VALOR",DECISAO],[["PRECONDICIONAMENTO"]],["PRECISA"],"VALOR"),
  DECISAO="sim",!,
  obtem(TempeLst,[["PRECONDICIONAMENTO"]],["temperatura"],"OPCOES",_),!,
  ihm_opcoes(TempeLst,"PRECONDICIONAMENTO","temperatura",TEMPERATURA),
  cria(_,[["PRECONDICIONAMENTO"]],["temperatura"],"VALOR",[TEMPERATURA]),!,
  obtem((INF,SUP),[["PRECONDICIONAMENTO"]],["duracao"],"FAIXA",_),!,
  ihm_faixa(INF,SUP,"PRECONDICIONAMENTO","duracao",TEMPO),!,
  cria(_,[["PRECONDICIONAMENTO"]],["duracao"],"VALOR",[TEMPO]),!.
configura("MEDICOES INICIAIS"):-
  obtem(["VALOR",DECISAO],[["MEDICOES INICIAIS"]],["PRECISA"],"VALOR"),
  DECISAO="sim",!, configuraInstrumento(SETUP,MEDIDA,Instrumento,Canal),!,
  cria(_,[["MEDICOES INICIAIS"]],["instrumento"],"canal",[Canal]),
  cria(_,[["MEDICOES INICIAIS"]],["instrumento"],"setup",[SETUP]),!,
  cria(_,[["MEDICOES INICIAIS"]],["instrumento"],"VALOR",[Instrumento]),!,
  cria(_,[["MEDICOES INICIAIS"]],["instrumento"],"tipo medida",[MEDIDA]),!,
  configuraGerador("MEDICOES INICIAIS","sintetizador HP 3325A").
configura("CONDICIONAMENTO"):-
  cria(_,[["6796"]],["ETAPAS"],"VALOR",["CONDICIONAMENTO"]),!,
  obtem(TempeLst,[["6796"]],["temperatura"],"OPCOES",_),!,
  ihm_opcoes(TempeLst,"6796","temperatura",TEMPERATURA),
  cria(_,[["6796"]],["temperatura"],"VALOR",[TEMPERATURA]),!,
  select equipamento(TEMPERATURA,EQUIPAMENTO),
  cria(_,[["6796"]],["equipamento"],"VALOR",[EQUIPAMENTO]),!,
  obtem(TempoLst,[["6796"]],["duracao"],"OPCOES",_),!,
  ihm_opcoes(TempoLst,"6796","duracao",TempMinS), str_int(TempMinS,TempMin),
  TempoMin=TempMin*60, str_int(TempoMinS,TempoMin),
  cria(_,[["6796"]],["duracao"],"VALOR",[TempoMinS]),!,
  obtem((INF,SUP),[["6796"]],["tolerancia"],"FAIXA",_),!,
  ihm_faixa(INF,SUP,"6796","tolerancia",TOLER),!,
  cria(_,[["6796"]],["tolerancia"],"VALOR",[TOLER]),!.
configura("ESTABILIZACAO"):-
  obtem(["VALOR",DECISAO],[["ESTABILIZACAO"]],["PRECISA"],"VALOR"),
  DECISAO="sim",!,
  obtem((INF,SUP),[["ESTABILIZACAO"]],["duracao"],"FAIXA",_),!,
  ihm_faixa(INF,SUP,"ESTABILIZACAO","duracao",TEMPO),
  cria(_,[["ESTABILIZACAO"]],["duracao"],"VALOR",[TEMPO]),!.
configura("MEDICOES FINAIS"):-
  obtem(["VALOR",DECISAO],[["MEDICOESFINAIS"]],["PRECISA"],"VALOR"),
  DECISAO="sim",!,
  configuraInstrumento(SETUP,MEDIDA,Instrumento,Canal),
  cria(_,[["MEDICOES FINAIS"]],["instrumento"],"canal",[Canal]),
  cria(_,[["MEDICOES FINAIS"]],["instrumento"],"setup",[SETUP]),!,
  cria(_,[["MEDICOES FINAIS"]],["instrumento"],"VALOR",[Instrumento]),!,
  cria(_,[["MEDICOES FINAIS"]],["instrumento"],"tipo medida",[MEDIDA]),!,
  configuraGerador("MEDICOES FINAIS","sintetizador HP 3325A").
configura(Etapa):-
  obtem(["VALOR" | [DECISAO]],[[Etapa]],["PRECISA"],"VALOR"),
  DECISAO="nao",!,
  apaga(_,[["6796"]],["ETAPAS"],"VALOR",Etapa),!.

```

% Regras para configurar um Instrumento:

```
configurarInstrumento(SETUP,MEDIDA,Instrumento,Canal):-
  trap(consult(instrums_bc,frames_dba),_,fail),
  tipoMedida(MEDIDA,Instrumento),
  configurarInstrumento(Instrumento,MEDIDA),!, geraSETUP(Instrumento,SETUP),
  obterdefault("CHANNEL","PARAMETROS",Canal),
  apagaClasse("INSTRUMENTOS"),!.
configuraGerador(Etapa,Instrumento):-
  obter(Lst,[[Etapa]],["GERAR"],"OPCOES",_), ihm_opcoes(Lst,Etapa,"GERAR",DECISAO),
  cria_,[[Etapa]],["GERAR"],"VALOR",[DECISAO]],
  DECISAO="sim",!,
  trap(consult(instrums_bc,frames_dba),_,fail),
  configurarInstrumento("sintetizador HP 3325A",Etapa),!,
  geraSETUP("sintetizador HP 3325A",SETUP),
  apagaClasse("INSTRUMENTOS"),!,
  cria_,[[Etapa]],["GERAR"],"setup",[SETUP]],!,
  cria_,[[Etapa]],["GERAR"],"instrumento",[Instrumento]],!.
configuraGerador(Etapa,_):- obter_,[[Etapa]],["GERAR"],"VALOR","nao"),!.
```

configurarInstrumento("ABORT","ERRO):-!.

```
configurarInstrumento("osciloscopio HP 54200 A/D","Frequencia):-
  obter(Lst1,[[TIMEBASE]],["SWEEP MODE"],"OPCOES",_),!,
  ihm_menu(Lst1,"OSCILOSCOPIO","TIMEBASE","SWEEP MODE",MODE),!,
  cria_,[[TIMEBASE]],["SWEEP MODE"],"VALOR",[MODE]],!,
  obter([INF2,SUP2],[TIMEBASE]],["RANGE"],"FAIXA",_),
  ihm_faixa(INF2,SUP2,"TIMEBASE","RANGE",AMPL),!,
  cria_,[[TIMEBASE]],["RANGE"],"VALOR",[AMPL]],!.
configurarInstrumento("multimetro HP 3457A","Frequencia):-
  obter([INF1,SUP1],[multimetro HP 3457A]],["Frequencia"],"FAIXA",_),!,
  ihm_faixa(INF1,SUP1,"multimetroHP 3457A","Frequencia",FREQ),!,
  cria_,[[multimetro HP 3457A]],["Frequencia"],"VALOR",[FREQ]],!,
  obter([INF2,SUP2],[multimetro HP 3457A]],["Amplitude"],"FAIXA",_),!,
  ihm_faixa(INF2,SUP2,"multimetro HP 3457A","Amplitude",AMPL),!,
  cria_,[[multimetro HP 3457A]],["Amplitude"],"VALOR",[AMPL]],!.
configurarInstrumento("sintetizador HP 3325A",Etapa):-
  obter(["VALOR","sim"],[[Etapa]],["GERAR"],"VALOR"),!,
  obter(["OPCOES"|Lst],[sintetizador HP 3325A]],["Funcao"],"OPCOES"),!,
  ihm_opcoes(Lst,"sintetizador","Funcao",FUN),!,
  cria_,[[sintetizador HP 3325A]],["Funcao"],"VALOR",[FUN]],!,
  obter([INF1,SUP1],[sintetizador HP 3325A]],["Frequencia"],"FAIXA",_),!,
  ihm_faixa(INF1,SUP1,"sintetizador HP 3325A","Frequencia",FREQ),!,
  cria_,[[sintetizador HP 3325A]],["Frequencia"],"VALOR",[FREQ]],!,
  obter([INF2,SUP2],[sintetizador HP 3325A]],["Amplitude"],"FAIXA",_),!,
  ihm_faixa(INF2,SUP2,"sintetizador HP 3325A","Amplitude",AMPL),!,
  cria_,[[sintetizador HP 3325A]],["Amplitude"],"VALOR",[AMPL]],!.
configurarInstrumento("sintetizador HP 3325A",_):-!.
```

tipoMedida(MEDIDA,Instrumento):-

```
free(MEDIDA), free(Instrumento),
obter(Dad1,[[INSTRUMENTOS]],["MEDIDA"],"OPCOES",_),
ihm_menu(Dad1,"MEDIDA","INSTRUMENTOS","MEDIDA",MEDIDA),!,
cria_,[[INSTRUMENTOS]],["MEDICAO"],"VALOR",[MEDIDA]],!,
consultamedida(Dad2,MEDIDA,Quantidade),
ihm_select(Quantidade,Dad2,"INSTRUMENTOS",MEDIDA,Instrumento),!,
cria_,[[INSTRUMENTOS]],["MEDIDA POR"],"VALOR",[Instrumento]],!.
tipoMedida("ABORT","ERRO):-!.
```

```
consultamedida(Dad,MED,2):-obtem(Dad,[[["INSTRUMENTOS"]],[MED],"OPCOES",_],!).
consultamedida(Dad,MED,1):-obtem(Dad,[[["INSTRUMENTOS"]],[MED],"MEDIDA",_],!).
```

```
geraSETUP("ABORT","ERRO"):-!.
geraSETUP("osciloscopio HP 54200 A/D",SETUP):-
  timebase(SETUP1), channel(SETUP2), %graph(SETUP3),
  concatlist([SETUP1,SETUP2],SETUP).
geraSETUP("sintetizador HP 3325A",SETUP):-
  setar("sintetizador HP 3325A","Funcao",SETUP1),
  setar("sintetizador HP 3325A","Amplitude",SETUP2),
  setar("sintetizador HP 3325A","Frequencia",SETUP3),
  concatlist([SETUP1,SETUP2,SETUP3],SETUP).
```

```
timebase(SETUP):-
  setar("TIMEBASE","SWEEP MODE",SETUP1),
  setar("TIMEBASE","RANGE",SETUP2),
  setar("TIMEBASE","REFERENCE",SETUP3),
  setar("TIMEBASE","SCALE",SETUP4),
  concatlist(["TIM;",SETUP1,SETUP2,SETUP3,SETUP4],SETUP).
```

```
channel(SETUP):-
  obtemdefault("CHANNEL","PARAMETROS",SETUP1),
  setar("CHANNEL","PROBE",SP1),
  setar("CHANNEL","RANGE",SP2),
  setar("CHANNEL","OFFSET",SP3),
  setar("CHANNEL","COUPLING",SP4),
  concatlist(["CHAN ",SETUP1,";",SP1,SP2,SP3,SP4],SETUP).
```

```
setar(Fr,SI,SETUP):-
  obtemdefault(Fr,SI,SETUP1),
  obtem(_,[Fr],[SI],"CODIGO",COD),!,
  concatlist([COD,SETUP1,";"],SETUP).
```