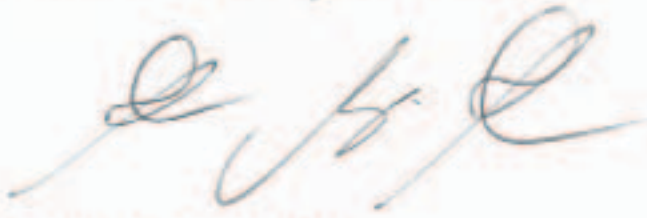
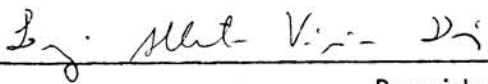


1. Publicação nº INPE-3890-TDL/822	2. Versão	3. Data Maio, 1988	5. Distribuição ☐ Interna ☒ Externa ☐ Restrita
4. Origem DRH/DIN	Programa FRH/CAP		
6. Palavras chaves - selecionadas pelo(s) autor(es) SISTEMAS ESPECIALISTAS RACIOCÍNIO INEXATO REGRAS DE DECISÃO ENCADEAMENTO RETROATIVO DE REGRAS FERRAMENTAS TIPO "ESQUELETO"			
7. C.D.U.: 681.3.019			
8. Título INPE-3890-TDL/822 SISTEMA DIAGNÓSTICO: UMA FERRAMENTA PARA A CONSTRUÇÃO DE SISTEMAS ESPECIALISTAS		10. Páginas: 256 11. Última página: E.141 12. Revisada por CRS Celso de Renna e Souza	
9. Autoria Sandra Aparecida Sandri 		13. Autorizada por  Dr. Marco Antonio Rangel - Diretor Geral -	
Assinatura responsável			
14. Resumo/Notas Este trabalho apresenta o sistema DIAGNÓSTICO, uma ferramenta para a construção de sistemas especialistas. A representação de conhecimento adotada é a de regra de decisão do tipo SE premissa ENTÃO (x) conclusão, onde x atesta a crença de que a conclusão é verdadeira se a premissa o for. O sistema consiste em quatro módulos: INICIALIZAÇÃO, AQUISIÇÃO, CONSULTA e EXPLANAÇÃO. O módulo de INICIALIZAÇÃO inicializa os arquivos do sistema e adquire palavras-chave. O módulo de AQUISIÇÃO interage com o especialista para adquirir as regras que compõem o banco de conhecimento de um dado domínio. O módulo de CONSULTA utiliza o banco de conhecimento e os dados do usuário para fornecer respostas a um problema proposto, utilizando encadeamento retroativo de regras. O módulo de EXPLANAÇÃO permite ao usuário acompanhar uma consulta através de uma linguagem de comandos. Além disto, o trabalho apresenta dois pequenos sistemas especialistas construídos pelo SISTEMA DIAGNÓSTICO: um sistema para diagnóstico do mau funcionamento de buzinhas de automóveis e um sistema para previsão de precipitação pluviométrica para a cidade de São José dos Campos.			
15. Observações: Dissertação de Mestrado em Computação Aplicada, aprovada em 25 de outubro de 1985, no Instituto de Pesquisas Espaciais - INPE.			

Aprovada pela Banca Examinadora
em cumprimento a requisito exigido
para a obtenção do Título de Mestre
em Computação Aplicada

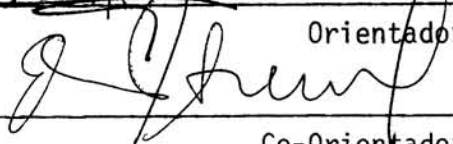
Dr. Luiz Alberto Vieira Dias


Presidente

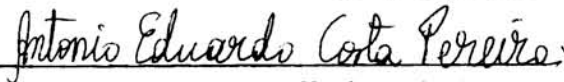
Dr. Celso de Renna e Souza


Orientador

Engº Edson Luiz França Senne, Mestre


Co-Orientador

Dr. Antonio Eduardo Costa Pereira


Membro da Banca
-convidado-

Engº Paulo Ouverá Simoni, Mestre


Membro da Banca

Candidata: Sandra Aparecida Sandri

São José dos Campos, 25 de outubro de 1985

Ao Seu Hermelindo e à Dona Geisa.

AGRADECIMENTOS

Ao Instituto de Pesquisas Espaciais por tornar possível a realização deste trabalho.

Ao Dr. Celso de Renna e Souza pela orientação e apoio.

A Edson Luís França Senne, pela orientação, disponibida de e apoio constantes.

A Iracema Fonseca de Albuquerque Cavalcanti pela participação como especialista no desenvolvimento do sistema para previsão de precipitação pluviométrica.

Aos colegas do Departamento de Informática e a todos os que direta ou indiretamente contribuíram na elaboração deste trabalho.

ABSTRACT

This work presents the DIAGNOSIS SYSTEM, a tool for building expert systems. The knowledge representation adopted is decision rules such as IF premise THEN (x) conclusion, where 'x' states the belief that the conclusion is true if the premise is true. The system consists of four modules: INICIALIZATION, ACQUISITION, CONSULTATION and EXPLANATION. The INICIALIZATION module inicializes the system files and acquires key words. The ACQUISITION module interacts with the specialist to acquire the rules which will constitute the knowledge base of a given domain. The CONSULTATION module uses the knowledge base and the user's data to give the answers to a proposed problem, utilizing backward-chaining of rules. The EXPLANATION module lets the user follow a consultation through a command language. This work also presents two small expert systems built by the DIAGNOSIS SYSTEM: a system for diagnosis of automobile horn malfunction and a system for pluviometric precipitation prevision for the city of São José dos Campos.

SUMÁRIO

	<u>Pág.</u>
LISTA DE FIGURAS	<i>xi</i>
LISTA DE TABELAS	<i>xiii</i>
<u>CAPÍTULO 1 - INTRODUÇÃO</u>	1
1.1 - Sistemas especialistas	1
1.2 - Sistema MYCIN	5
1.3 - Ferramentas para construção de sistemas especialistas	6
1.4 - Sistema DIAGNÓSTICO	8
1.5 - Organização do documento	10
<u>CAPÍTULO 2 - REPRESENTAÇÃO DE CONHECIMENTO</u>	13
2.1 - Atributos	13
2.2 - Objetos	14
2.3 - Funções	15
2.4 - Valores	16
<u>CAPÍTULO 3 - MÓDULOS DE INICIALIZAÇÃO E AQUISIÇÃO DE CONHECIMENTO</u> ..	17
3.1 - Módulo de INICIALIZAÇÃO	17
3.1.1 - Procedimento DICIONÁRIO	17
3.1.2 - Procedimento ATUALIZATIPOSCONTEXTO	18
3.2 - Módulo de AQUISIÇÃO	18
3.2.1 - Aquisição de uma regra	20
3.2.2 - Alteração de uma regra	25
3.2.2.1 - Inclusão de cláusula ou subcláusula	26
3.2.2.2 - Alteração de cláusula ou subcláusula	27
3.2.2.3 - Exclusão de cláusula ou subcláusula	28
3.2.3 - Reconhecimento de cláusulas	29
3.2.4 - Armazenamento de regras	31
3.2.5 - Exclusão de regra	32
3.2.6 - Listagem de regra	32
3.2.7 - Listagem de referências a atributos	33
3.2.8 - Manipulação de regras especiais	34

	<u>Pág.</u>
<u>CAPÍTULO 4 - MÓDULO DE CONSULTA</u>	37
4.1 - Procedimento MONITOR	40
4.2 - Procedimento BUSCAVALOR	43
4.3 - Modelo de verdade	45
4.4 - Árvore de contextos	48
4.5 - Tratamento de regras especiais	50
<u>CAPÍTULO 5 - MÓDULO DE EXPLANAÇÃO</u>	51
<u>CAPÍTULO 6 - CONCLUSÃO E EXTENSÕES FUTURAS</u>	57
REFERÊNCIAS BIBLIOGRÁFICAS	59
APÊNDICE A - DESCRIÇÃO DOS ARQUIVOS ESTÁTICOS DO SISTEMA	
APÊNDICE B - DESCRIÇÃO DOS ARQUIVOS DINÂMICOS DO SISTEMA	
APÊNDICE C - SISTEMA ESPECIALISTA PARA DIAGNÓSTICO DO MAU FUNCIONA MENTO DE BUZINAS DE AUTOMÓVEIS	
APÊNDICE D - SISTEMA DE PREVISÃO DE PRECIPITAÇÃO PLUVIOMÉTRICA	
APÊNDICE E - LISTAGEM DE PROGRAMAS	

LISTA DE FIGURAS

	<u>Pág.</u>
1.1 - Arquitetura do sistema DIAGNÓSTICO	9
3.1 - Arquitetura do módulo de AQUISIÇÃO	20
3.2 - Status da base de dados	31
4.1 - Esquema simplificado de MONITOR	42
4.2 - Esquema simplificado de BUSCAVALOR	44
4.3 - Árvore de contextos	48
4.4 - Evolução da árvore de contextos	49
A.1 - Registro de DIRDISK	A.1
A.2 - Registro de TIPOSCONTEXTO	A.3
A.3 - Registro de OBJETOETIPOS	A.5
A.4 - Registro de ATRIBUTOSREF	A.6
A.5 - Estrutura de uma regra de decisão	A.7
A.6 - Estrutura de uma cláusula <i>ou</i>	A.8
A.7 - Cláusula simples	A.9
A.8 - Registro de PROPATRIBUTOS	A.10
A.9 - Registro de OBJETIVOS	A.11
B.1 - Registro de REGRACONSULTA	B.1
B.2 - Desenvolvimento da árvore de raciocínio	B.3
B.3 - Registro de VALORCONHECIDO	B.4

LISTA DE TABELAS

Pág.

5.1 - Comandos do sistema de EXPLANAÇÃO	52
A.1 - Classes de palavras-chave	A.2
A.2 - Campo ASSOC.	A.2

CAPÍTULO 1

INTRODUÇÃO

1.1 - SISTEMAS ESPECIALISTAS

Segundo Stefik et alii (1982), *sistemas especialistas* são programas que resolvem problemas substanciais, geralmente tidos como difíceis e requerendo *perícia*. Dizem ainda que estes são chamados *baseados em conhecimento* porque seu desempenho depende criticamente do uso de fatos e heurísticas usados por especialistas.

O direcionamento para sistemas baseados em conhecimento, como aponta Davis et alii (1977), representa uma mudança das tentativas prévias de *resolvedores de problemas gerais* como, por exemplo, o GPS (Newell and Simon, 1963). Ao invés de tentar resolver problemas gerais, os sistemas baseados em conhecimento têm enfatizado a acumulação de grandes quantidades de conhecimento em um único domínio e o desenvolvimento de técnicas específicas a este domínio, de modo a desenvolver um alto nível de *perícia*. Para Feigenbaum (1977) o poder de um sistema especialista deriva do conhecimento que ele possui e não dos formalismos específicos e esquemas de inferência que ele utiliza.

Segundo Hayes-Roth et alii (1983) *perícia* consiste em conhecimento sobre um domínio específico, entendimento dos problemas do domínio e habilidade para resolver alguns desses problemas. Além de conhecimento público, disponível em publicações, um especialista dispõe de um conhecimento particular, que o permite dar palpites quando necessário, reconhecer enfoques promissores para atacar problemas e trabalhar efetivamente com dados errôneos ou incompletos. A tarefa central na construção de um sistema especialista é reproduzir tal conhecimento.

Além de resolver problemas difíceis ao nível de especialistas, uma das características dos sistemas especialistas é a de, tal como os especialistas humanos, ser capaz de justificar suas decisões.

Esta habilidade tanto dos especialistas humanos como dos sistemas faz aumentar a confiança do consultante nos resultados da consulta e, no caso específico dos sistemas, permite ao especialista que o constrói encontrar erros no raciocínio do programa, caso estes ocorram (Barr and Feigenbaun, 1981).

As tarefas genéricas desempenhadas por sistemas especialistas, segundo Stefik et alii (1982) são:

- *Interpretação* - análise de dados para determinar seu significado.
- *Diagnóstico* - processo de detecção de falhas em um sistema (ou determinação de um estado patológico em um organismo vivo).
- *Monitoramento* - interpretação contínua de sinais e alarme quando é requerida uma intervenção.
- *Planejamento* - programa das ações que podem ser desempenhadas para atingir os objetivos e criar planos.
- *Projeto* - estabelecimento das especificações para criar objetos que satisfazem requisições específicas.

Além das categorias acima, Hayes-Roth et alii (1983) citam reparo, instrução e controle.

A grande importância dos sistemas especialistas reside na disseminação de conhecimento, abundantes em alguns meios e escassos em outros e, por isto mesmo, de custo elevado. A eficiência demonstrada por estes sistemas em certas áreas determinou o seu sucesso comercial, como comprova o crescente aumento de empresas de Inteligência Artificial no mercado americano, que têm os sistemas especialistas como seu principal produto, e de grandes empresas que vêm investindo em pesquisa na área (Artificial Intelligence, 1984). O grande interesse na área cresceu sobretudo após o anúncio do uso desta tecnologia no projeto japonês de computadores de quinta geração.

O número atual de sistemas conhecidos no mundo é cerca de 400, nas áreas de Medicina (diagnóstico, tratamento e monitoramen

to), Psicofarmacologia, Diagnóstico Técnico, Projeto Assistido por Computador, Ensino Assistido por Computador, Reconhecimento da Palavra, Compreensão de Linguagem Natural, Engenharia de "Software", Configuração de Sistemas de Informática, Instruções de Comando, Biologia, Química, Física, Matemática, Engenharia Civil, Implantação de Redes de Fluidos, Cálculo de Estruturas, Burocracia, Robótica, Alocação de Espaço, Arqueologia, Agricultura, Meteorologia, Administração de Empresas, Administração de Investimentos, Contabilidade, Fiscalização, Economia, Gerenciamento de Produção, Jogos, Elaboração de Emprego de Tempo, Prospeção Petrolífera e Mineral, Ajuda ao Comando, Controle de Sistemas Complexos, Simulação, Controle de Tráfego Aéreo, Interpretação de Sinais e Tratamento de Imagens, como afirma Rault (1984). Neste mesmo artigo, são citados 240 sistemas especialistas provenientes de laboratórios universitários e industriais (com predominância das universidades) e as listas das principais empresas fornecedoras destes sistemas e de seus mais importantes usuários.

Os sistemas especialistas, na sua maioria, têm sido usados em máquinas de grande porte devido à quantidade de memória necessária para conter o *conhecimento* do sistema. Por este motivo, o uso destes sistemas em microcomputadores tem sido inexpressivo, embora já existam alguns sistemas disponíveis comercialmente (Webster, 1985). É possível afirmar que o uso destes sistemas tende a crescer com o progressivo aumento de memória nos microcomputadores.

Um dos primeiros e mais bem sucedidos sistemas especialistas, DENDRAL (Buchanan and Feigenbaum, 1978), foi desenvolvido para analisar espectrografia de massa, ressonância magnética e outros dados químicos experimentais para inferir estruturas plausíveis para compostos químicos desconhecidos. Outros sistemas importantes são: PROSPECTOR (Duda et alii, 1977), de prospecção mineral, cujo maior feito foi a descoberta de uma grande jazida de molibdênio nos EUA; R1 (McDermott, 1981), que configura pedidos de clientes para o computador VAX da DEC, tarefa considerada anteriormente impossível pelos especialistas residentes; CADUCEUS (Pople Jr., 1977) para diagnóstico em medi

cina interna, que pode diagnosticar corretamente casos complexos que frustram especialistas humanos; MACSYMA (Martin and Fateman, 1971), que supera muitos especialistas no cálculo diferencial e integral simbólicos e na simplificação de expressões simbólicas; HEARSAY-II (Erman et alii, 1980), para entendimento de linguagem natural, uma das áreas mais difíceis em Inteligência Artificial; e MYCIN (Shortliffe, 1976), para diagnóstico de doenças infecciosas, em particular de infecções bacterianas no sangue, de importância fundamental no presente trabalho por servir de modelo ao sistema DIAGNÓSTICO.

A maior diferença entre os sistemas especialistas e os outros sistemas de aplicação, segundo Nau (1983), é ter uma base de conhecimento separada da estratégia de controle, ao invés de contê-los explicitamente no código do programa. A estrutura básica de um sistema especialista é descrita por Gevarter (1982) como consistindo em:

- 1) uma *Base de conhecimento* dos fatos e heurísticas do domínio associados ao problema;
- 2) uma *Estrutura de controle* para utilizar a base de conhecimento na solução do problema;
- 3) uma *Memória de trabalho* para acompanhar o status do programa, os dados de entrada para o problema específico e o histórico relevante do que foi feito até então.

Como observa Duda e Gaschinig (1981), a maneira mais usual de codificar a base de conhecimento é com o uso de regras de decisão do tipo SE <premissa> ENTÃO <conclusão>, onde a conclusão será tomada se a premissa for verdadeira. Cada uma destas regras representa um pequeno módulo de conhecimento em um dado domínio. A maioria dos sistemas existentes usa centenas de regras obtidas de vários especialistas. Algumas das vantagens do uso de regras de decisão na formação da base de conhecimento são:

- O crescimento da base de conhecimento ao longo do tempo, ao permitir o refinamento de regras antigas e a aquisição de novas regras.

- O mesmo sistema geral (dependendo da representação) pode ser usado para uma variedade de aplicações, ao retirar um conjunto de regras e ao colocar outro no lugar.
- O mesmo conhecimento pode ser usado de várias maneiras (até mesmo para a educação) ao mudar o interpretador das regras.
- O programa pode dar explicações simples e esclarecedoras de seu comportamento, ao meramente descrever as regras que está aplicando (uma maneira eficiente de detectar regras falhas).
- A possibilidade de desenvolver sistemas introspectivos, (p.ex.: que possam checar a consistência de suas próprias regras) e evolucionários (p.ex.: que possam modificar suas próprias regras e aprender novas).

Outra das características dos sistemas especialistas é a de tentar imitar, até certo ponto, a maneira que um especialista toma decisões. Uma das maneiras de fazê-lo é codificar o quanto um especialista acredita na veracidade de uma hipótese considerando toda a evidência disponível, através da associação de valores numéricos às regras de decisão.

1.2 - SISTEMA MYCIN

O sistema MYCIN, desenvolvido em LISP na Universidade de Stanford (Shortliffe, 1976), é um exemplo clássico de sistemas que modelam seu conhecimento através de regras de decisão. Uma implementação em PROLOG deste sistema é descrito por Hammond (1982).

Este sistema é formado por um banco de conhecimento que armazena cerca de 400 regras, um mecanismo de inferência, um módulo de explanação e uma base de dados que contém os dados do paciente e os dados dinâmicos, criados ao longo de uma consulta ao sistema.

O MYCIN representa raciocínio inexato como *A sugere B ou C e D tendem a descartar E*, através da manipulação de valores que

atestam a crença numa hipótese tomando toda a evidência disponível num dado instante da consulta. Estes valores, chamados Fatores de Certeza (FC), variam entre -1 (hipótese definitivamente falsa) e 1 (hipótese definitivamente verdadeira). À conclusão de cada regra está associado um FC que atesta a crença na sua validade, caso a premissa seja verdadeira. Ao acessar uma regra, o sistema obtém um FC para a premissa que é combinado com o FC da conclusão, o que resulta na crença da hipótese representada pela conclusão, considerando-se a evidência adquirida até aquele ponto da consulta.

A hierarquização das entidades do domínio de conhecimento é descrita através de uma árvore construída ao longo do processamento.

O controle retroativo, que é o mecanismo de inferência utilizado pelo MYCIN, consiste em escolher uma proposição, selecionar as regras cujas conclusões referenciam aquela proposição e tentar provar os subobjetivos sugeridos pelas premissas destas regras. Quando não se consegue mais deduzir, o sistema pede ao usuário a informação desejada. Com a utilização deste mecanismo, o sistema ganha em eficiência (ao examinar o banco exaustivamente), o que perde em generalidade por não aceitar raciocínio progressivo (forward).

Uma das maiores potencialidades do sistema é a sua capacidade de explanação. O sistema utiliza uma linguagem de comandos que recupera as regras utilizadas e um interpretador de linguagem natural para responder perguntas sobre o status da base de dados.

1.3 - FERRAMENTAS PARA CONSTRUÇÃO DE SISTEMAS ESPECIALISTAS

O esforço dispendido na construção de um sistema especialista, particularmente na extração de conhecimento de especialistas humanos e sua posterior representação, depende em parte da ferramenta adotada para construí-lo. Estas ferramentas são classificadas por Barstow et alii (1983) como linguagens de propósito geral, sistemas do tipo *esqueleto* e ferramentas de projeto assistido por computador, cujo exemplo mais significativo é o AGE (Nii and Aiello, 1979).

As linguagens de propósito geral, construídas especificamente para engenharia de conhecimento, são menos restritas que os sistemas de *esqueleto*, por admitirem várias estruturas de controle, mas de manipulação mais difícil. Algumas destas linguagens são: ROSIE (Fain et alii, 1982), RLL (Greiner and Lenat, 1980), HEARSAY III (Erman et alii, 1981) e OPS5 (Forgy, 1981).

Os sistemas do tipo *esqueleto* utilizam uma mesma máquina de inferência para a criação de vários sistemas especialistas ao trocar as bases de conhecimento. Estes sistemas derivaram, na sua maioria, de sistemas especialistas já existentes, ao serem generalizados e dotados de um módulo de aquisição de conhecimento. Como usualmente existe conhecimento específico do antigo sistema embutido de maneiras irreconhecíveis na própria máquina de inferência, na maioria das vezes é preciso fazer *acertos* também nesta máquina para adequar o novo domínio de conhecimento. A quantidade de mudanças é maior, quanto mais distante for o domínio atual do antigo. Alguns dos sistemas deste tipo são o KAS (Knowledge Acquisition System), programa de consulta do PROSPECTOR sem seu conhecimento específico (Duda et alii, 1978); o EXPERT (Weiss e Kulikowski, 1979), usado principalmente para desenvolver modelos para diagnóstico em medicina, especificamente em oftalmologia, endocrinologia e reumatismo; e EMYCIN (van Melle, 1979), versão independente de domínio do MYCIN, que gerou entre outros sistemas, o PUFF (Feigenbaum, 1977), que diagnostica doenças pulmonares.

O EMYCIN é considerado um sistema apropriado para desenvolver sistemas de consulta que requerem dados sobre um caso e provê uma interpretação de sua análise (Barstow et alii, 1983). Ele é particularmente apropriado para problemas dedutivos, tais como diagnose de falhas, em que uma grande quantidade de dados não é confiável e onde o espaço de possíveis diagnósticos possa ser enumerado.

Hayes-Roth et alii (1983) apresentam a comparação do desempenho de sistemas especialistas construídos através de EMYCIN, KAS, EXPERT, OPS5, ROSIE e RLL.

1.4 - SISTEMA DIAGNÓSTICO

Tal como o sistema EMYCIN, o sistema DIAGNÓSTICO é uma ferramenta para construção de sistemas especialistas, baseado no sistema MYCIN.

O sistema constitui-se de quatro módulos: INICIALIZAÇÃO, AQUISIÇÃO, CONSULTA E EXPLANAÇÃO. Os dois primeiros módulos são responsáveis pela aquisição de conhecimento de um dado domínio de aplicação, o módulo de CONSULTA constitui a máquina de inferência do sistema e o módulo de EXPLANAÇÃO dá informações sobre o status de uma dada consulta.

A forma de representação adotada foi a de regras de decisão do tipo SE premissa ENTÃO (x) conclusão, onde a crença na conclusão é x, se a premissa for verdadeira. Esta forma, dita externa, difere da codificação interna, onde todas as informações contidas nas regras são distribuídas em vários arquivos estáticos.

Numa primeira fase, o banco de conhecimento (uma coleção de regras de decisão) é construído da interação de especialistas com os módulos de INICIALIZAÇÃO e AQUISIÇÃO. Posteriormente, este banco é utilizado pelos usuários do sistema através dos módulos de CONSULTA e EXPLANAÇÃO.

A arquitetura do sistema está ilustrada na Figura 1.1.

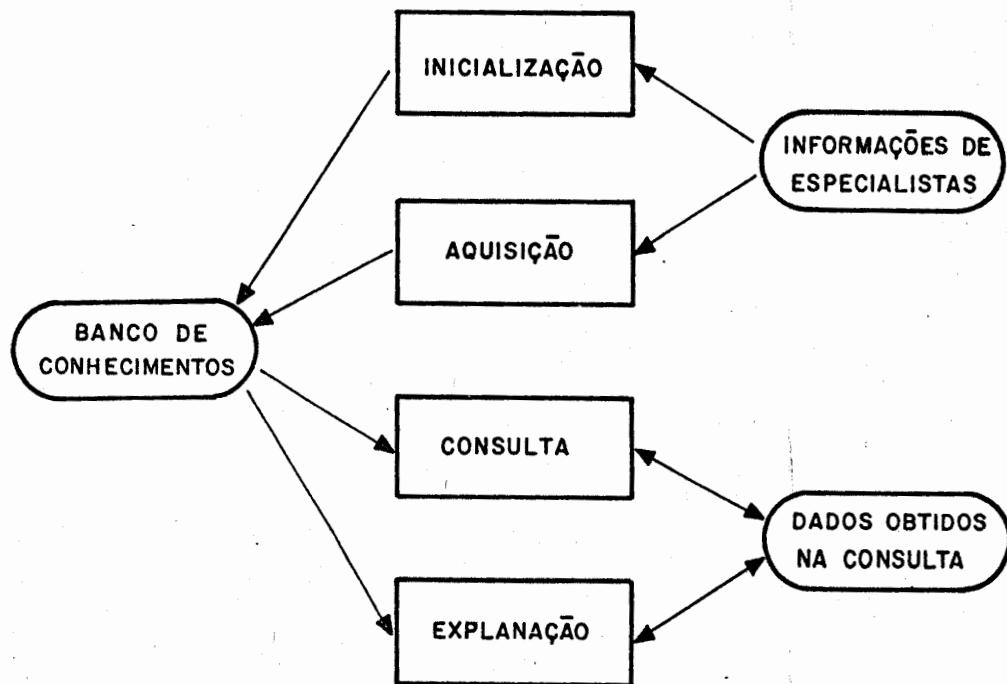


Fig. 1.1 - Arquitetura do sistema DIAGNÓSTICO.

A primeira aplicação do sistema foi a construção de um sistema especialista para diagnosticar o mau funcionamento de buzinas de automóveis. As regras utilizadas (van Melle, 1974) e os exemplos da construção e utilização deste sistema encontram-se no Apêndice C. Atualmente o sistema vem sendo utilizado na construção de um sistema especialista para previsão de precipitação pluviométrica em São José dos Campos. As regras deste sistema e comentários sobre sua aquisição encontram-se no Apêndice D.

O módulo de AQUISIÇÃO está baseado no sistema de aquisição de regras desenvolvido por Lage Filho (1978) para uso em medicina comunitária. Algumas alterações foram feitas sobre este sistema, de modo a acomodar a sintaxe das regras no formato SE/ENTÃO e de dotar o programa de estrutura flexível a ponto de poder ser usado em qualquer domínio de conhecimento, além de algumas modificações relativas à sua adequação para o processamento interativo (Senne e Simoni, 1981). Este sistema foi depois disto modificado para adequar-se ao módulo de CONSULTA. O módulo de consulta e a linguagem de comandos do módulo de EXPLA

NAÇÃO foram inicialmente idealizados por Senne e Simoni (1982 a e b).

A linguagem utilizada para a implementação do sistema DIAGNÓSTICO é o Algol do Burroughs 6800 (Segre, 1981), por ser a mais adequada disponível no INPE no início do projeto em 1977.

Os domínios de aplicação previstos para este sistema, pertinentes ao INPE, são por exemplo Meteorologia, Sensoriamento Remoto e Diagnóstico de Falhas em Sistemas Espaciais. Cabe aqui mencionar que *sistemas especialistas* para controle de voo de ônibus espaciais estão sendo implementados na NASA (Helly Jr et alii, 1984). Embora o sistema seja de cunho geral, este provavelmente sofrerá pequenas modificações para uso em um domínio específico, devido a características inerentes a cada área de conhecimento.

1.5 - ORGANIZAÇÃO DO DOCUMENTO

O Capítulo 2 traz a forma de representação de conhecimento do sistema DIAGNÓSTICO e analisa os elementos que formam as regras do banco de conhecimento.

O Capítulo 3 descreve o módulo de INICIALIZAÇÃO, responsável pela inicialização de arquivos do sistema e aquisição de palavras-chave, e seus procedimentos básicos, bem como o módulo de AQUISIÇÃO DE CONHECIMENTO que adquire, depura e exclui regras no banco de conhecimento, detalhando cada uma de suas fases.

O módulo de CONSULTA, máquina de inferência do sistema, está descrito no Capítulo 4, com o exemplo de uma consulta ao sistema especialista para previsão de precipitação criado pelo sistema DIAGNÓSTICO, seus procedimentos básicos, o modelo de verdade adotado pelo sistema, a descrição da árvore de contextos e o tratamento de regras especiais.

O Capítulo 5 contém as especificações para o módulo de EXPLANAÇÃO, a linguagem de comandos que o acessa e exemplos de sua utilização.

As conclusões e as futuras expansões encontram-se no Capítulo 6, e os Apêndices A, B, C e D trazem, respectivamente, os arquivos estáticos do sistema, os arquivos dinâmicos, as regras que formam o banco de conhecimento do sistema que diagnostica o mau funcionamento de buzinas de automóveis e sua utilização pelo sistema e as regras do sistema para previsão de precipitação pluviométrica. O Apêndice E traz as listagens dos programas que constituem o sistema.

Os exemplos do uso do sistema DIAGNÓSTICO foram feitos utilizando linguagem pseudo-natural e atendendo às conveniências do usuário e do sistema, podendo por isto eventualmente desviar-se das regras gramaticais da Língua.

CAPÍTULO 2

REPRESENTAÇÃO DE CONHECIMENTO

As regras de decisão que formam o banco de conhecimento do sistema DIAGNÓSTICO são do tipo SE <premissa> ENTÃO x <conclusão>, onde x atesta a crença do especialista de que a conclusão é válida, se as condições da premissa forem satisfeitas.

A premissa é formada por uma conjunção de cláusulas que podem ser simples, do tipo *ou* ou do tipo *ou1*. Por sua vez, uma cláusula do tipo *ou* ou *ou1* é uma disjunção de cláusulas simples, as quais neste caso também são chamadas subcláusulas. A conclusão é formada exclusivamente pela conjunção de cláusulas simples. Uma cláusula simples é basicamente representada pela quádrupla (atributo, objeto, função, valor). A frase *João tem 13 anos*, por exemplo, poderia ser codificada como (idade, João, é, 13).

O objeto de uma cláusula é uma das entidades do domínio de conhecimento do sistema, e o atributo representa uma das características deste objeto. A função relaciona o valor especificado ao atributo do objeto. Quando a cláusula é avaliada, a função causa uma ação e determina se a cláusula é verdadeira ou falsa.

Além das regras como as descritas acima, o sistema conta com regras especiais que transferem vários valores para o arquivo VALOR CONHECIDO (Apêndice B) se a premissa for verdadeira. Por exemplo, se num dado domínio obtém-se para o atributo *cidade* o valor *São José dos Campos*, automaticamente podem ser atualizados os valores para os atributos *estado*, *país*, *população*, etc..., sem precisar deduzí-los.

2.1 - ATRIBUTOS

Os atributos podem ser de três tipos:

Valores múltiplos - são aqueles que podem assumir mais de um valor. Ao tomar *aptidão* como atributo, este será de valores múltiplos, já que uma pessoa pode ter mais de uma aptidão.

Valor único - são aqueles cujos valores possíveis são mutuamente exclusivos. O atributo *idade* citado anteriormente é de valor único.

Sim/não - são atributos de valor único, cujos valores possíveis são falso ou verdadeiro. Na frase *João é corintiano*, o atributo *corintiano* é classificado como sim/não.

O arquivo PROPATRIBUTOS (Apêndice A) descreve as propriedades de cada atributo, tais como os valores que aquele atributo pode assumir e se seu valor deve ser deduzido ou perguntado ao usuário.

Caso um atributo necessite de uma unidade e esta não seja encontrada no texto da cláusula, o sistema adotará a unidade padrão fornecida pelo especialista. Se a cláusula tiver explicitamente um múltiplo ou submúltiplo da unidade padrão, adota-se esta como a unidade e normaliza-se o valor.

2.2 - OBJETOS

Nos módulos de INICIALIZAÇÃO e AQUISIÇÃO, adquirem-se para cada objeto quais os tipos de contexto a ele associados, os quais são armazenados nos arquivos TIPOSCONTEXTO (Apêndice A). Um tipo de contexto é um objeto seguido de uma especificação que depende do domínio. No domínio do MYCIN, os tipos de contexto referem-se ao fator tempo. O objeto *remédio*, por exemplo, tem associados a ele os tipos *remédio anterior* e *remédio atual*, referindo-se a remédios que o paciente tenha tomado no passado e remédios que estejam sendo ministrados no mento da consulta.

Durante a execução do módulo de CONSULTA, a cada nova utilização de um tipo de contexto, cria-se um novo contexto. No MYCIN, por

exemplo, ao *paciente-1* podem estar associados os remédios atuais *remédio-1*, *remédio-2* e *remédio-3* e os remédios anteriores *remédio-4* e *remédio-5*. No caso de haver um só tipo de contexto, este confunde-se com o objeto ao qual está associado, como é o caso de *paciente* no domínio do MYCIN.

O sistema admite uma hierarquia entre tipos de contexto, que é descrito no arquivo TIPOSCONTEXTQ. Por exemplo, *paciente* está hierarquicamente acima de *cultura atual (anterior)*, que está acima do *organismo atual (anterior)* que, por sua vez, está acima de *remédio atual (anterior)*. Cada cláusula de conclusão de uma regra é classificada de acordo com tipos de contexto mais baixos dentre os associados aos objetos desta cláusula de conclusão e os das cláusulas de premissa. Por exemplo, tomando como base a hierarquia do MYCIN, a regra fictícia:

SE:

- 1) ... paciente ...&
- 2) ... organismo ...&

ENTÃO:

- 1) ... remédio ...&
- 2) ... cultura ...&

teria a primeira cláusula da conclusão classificada como *remédio atual (anterior)* e a segunda como *organismo atual (anterior)*.

Durante a execução do módulo de CONSULTA, os contextos relacionam-se através da árvore de contextos, cuja descrição encontra-se na Seção 4.3.

2.3 - FUNÇÕES

Uma função no sistema promove uma ação especificada, e ao final do processamento de determinadas funções é atribuído a elas

um valor verdadeiro ou falso. Se uma função for falsa, a sua negação será verdadeira e vice-versa.

Durante a execução do módulo de CONSULTA, os valores de duzidos para um determinado atributo e contexto são armazenados juntamente com seus fatores de certeza. A função de premissa de uma dada cláusula compara o valor contido na cláusula com os armazenados no banco de conhecimento. A cláusula é considerada verdadeira se, além de satisfazer a condição, a certeza associada está numa determinada faixa. As funções de premissa implementadas no momento são, *igual*, *diferente*, *maior*, *menor*, *maior igual*, *menor igual* e *conhecido*, as quais serão verdadeiras se $.2 < FC \leq 1..$ A função *definitivo* é verdadeira se $FC = 1.$ e a função *pode ser*, se estiver na faixa $-.2 < FC \leq 1..$

As funções de conclusão disponíveis no momento são: *igual*, que atualiza o FC de um valor para um dado atributo e objeto, e *mostrar*, que atualiza o FC de um valor que está representado na forma de um texto e mostra este texto na tela (uma cláusula é do tipo "mostrar" se esta for a sua função).

Somente são aceitas funções definidas no sistema ou sinônimas destas. Isto por que cada função tem as ações correspondentes a ela codificadas no módulo de CONSULTA.

2.4 - VALORES

Os valores aceitos no sistema DIAGNÓSTICO são números reais ou inteiros, textos, cadeias de caracteres e valores booleanos representados por sim/não. Para cada atributo são aceitos um ou mais tipos de valores.

Durante a execução do módulo de CONSULTA, todos os valores que cada atributo assume e a crença e a descrença na sua validade são armazenados no arquivo VALORCONHECIDO (Apêndice B).

CAPÍTULO 3

MÓDULOS DE INICIALIZAÇÃO E AQUISIÇÃO DE CONHECIMENTO

3.1 - MÓDULO DE INICIALIZAÇÃO

O módulo de INICIALIZAÇÃO presta-se a duas atividades:

- 1) Inicialização de todos os arquivos do sistema (Apêndices A e B) para um dado domínio de conhecimento. Caso estes arquivos já tenham sido utilizados anteriormente, o módulo de INICIALIZAÇÃO reinicializa-os, limpando-os.
- 2) Aquisição de palavras-chave do domínio escolhido, manipulando diretamente o dicionário (arquivo DIRDISK) com o procedimento DICIONÁRIO e os arquivos TIPOSCONTEXTO e OBJETOETIPOS com o procedimento ATUALIZATIPOSCONTEXTO.

A primeira atividade garante que a cada utilização deste módulo todos os arquivos de um dado domínio estejam limpos.

Como será visto na Seção 3.2, a segunda atividade também é feita pelo módulo de AQUISIÇÃO, mas de maneira mais amigável (user-friendly). No entanto, o módulo de INICIALIZAÇÃO por ter a entrada de dados formatada torna mais rápido o armazenamento de palavras-chave e de seus atributos, além de ser a única fase em que os arquivos citados podem ter seus registros apagados ou modificados.

3.1.1 - PROCEDIMENTO DICIONÁRIO

Neste procedimento é feita a inserção, modificação e listagem de palavras-chave e de seus atributos em DIRDISK. A entrada dos dados é formatada, e a linguagem de comandos para manipulação deste módulo é a seguinte:

- a - adquire a próxima palavra-chave,
- m - escreve um novo registro sobre o registro dado,

- l - lista todos os registros de DIRDISK,
- f - termina a execução de DICIONÁRIO.

Para excluir a palavra-chave do registro basta modificar α , escrevendo nele a próxima palavra que se deseja inserir.

As funções de predicado reconhecidas pelo sistema estão nos primeiros registros do arquivo e não são aceitas novas palavras-chave que representem funções que não sejam sinônimas daquelas.

3.1.2 - PROCEDIMENTO ATUALIZATIPOSCONTEXT0

O procedimento ATUALIZATIPOSCONTEXT0 adquire, modifica e lista os tipos de contexto associados a cada objeto. Cada tipo de contexto de cada objeto tem um registro em TIPOSCONTEXT0 que descreve suas características, e cada objeto tem um registro em OBJETOETIPOS contendo todos os contextos a ele associados. Cada atualização em TIPOSCONTEXT0 atualiza automaticamente OBJETOETIPOS.

A exclusão de um registro é feita da mesma maneira que no procedimento DICIONÁRIO. A linguagem de comandos deste procedimento é:

- a - adquire um novo contexto,
- m - escreve um novo registro sobre o registro dado,
- l - lista todos os registros de TIPOSCONTEXT0,
- f - termina a execução de ATUALIZATIPOSCONTEXT0.

Um objeto poderá ser associado a um contexto vazio; neste caso, este objeto não poderá ser associado a nenhum outro contexto.

3.2 - MÓDULO DE AQUISIÇÃO

O módulo de aquisição de conhecimento é responsável pela aquisição, modificação e exclusão de regras do banco de conhecimento, atualizando todos os arquivos estáticos do banco, além de listar regras

e referências a atributos. Este módulo interage com o especialista de maneira amigável (user-friendly), orientando-o na utilização dos seus recursos e submetendo para sua aceitação todo o processamento que fizer.

A aceitação de uma regra pelo sistema será feita pela busca dos parâmetros (atributo, objeto, função, valor) através do reconhecimento das palavras-chave contidas no dicionário. Estes parâmetros podem não aparecer explicitamente na frase digitada pelo especialista.

O sistema pode ser utilizado de dois modos: através de uma linguagem de comandos ou de *menus*. Em qualquer um deles, há uma opção para continuar o processamento utilizando o outro modo. A linguagem de comandos deste módulo obedece à seguinte sintaxe:

<i>regranova</i>	- adquire uma nova regra,
<i>regraespecial</i>	- adquire uma nova regra especial,
<i>altere regra <x></i>	- altera a regra <x> ,
<i>exclua regra <x></i>	- exclui a regra <x> ,
<i>mostre regra <x></i>	- mostra a regra <x> ,
<i>referencias</i>	- lista as referências de um atributo,
<i>menu</i>	- ativa a comunicação através de <i>menu</i> ,
<i>fim</i>	- termina a sessão.

As rotinas principais deste módulo são: ADQUIREREGRA, ALTERAREGRA, MOSTRAREGRA, EXCLUIREGRA, LISTAREFERENCIAS, RECONHECECLAUSULA e ARMAZENAREGRA. A arquitetura simplificada deste módulo está ilustrada na Figura 3.1.

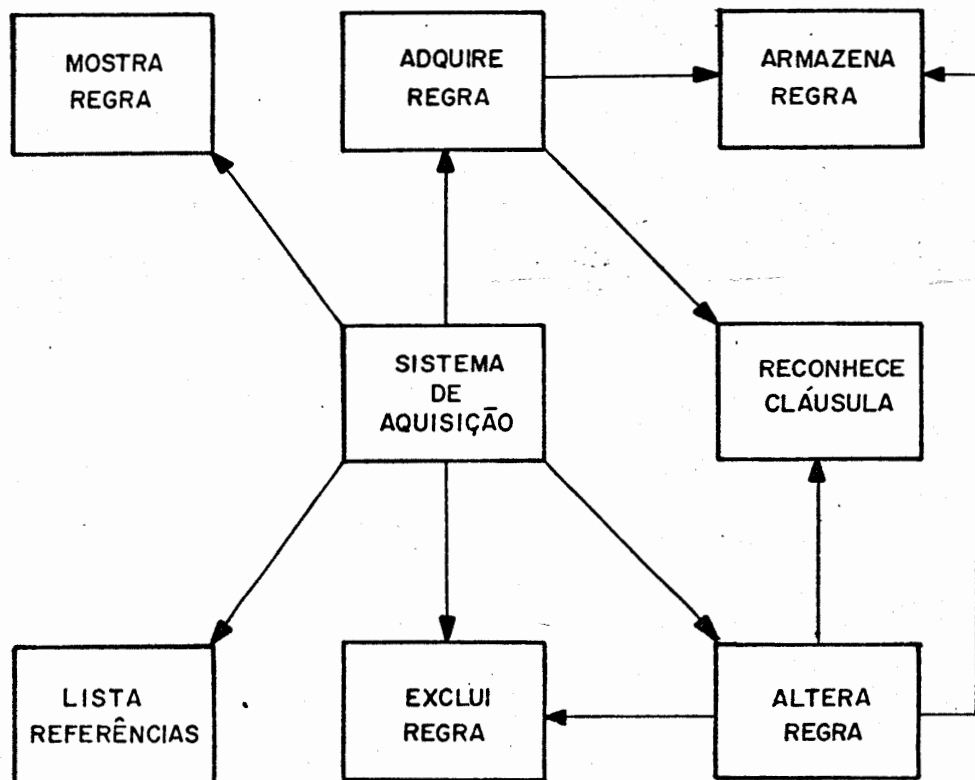


Fig. 3.1 - Arquitetura do módulo de AQUISIÇÃO.

3.2.1 - AQUISIÇÃO DE UMA REGRA

A aquisição de uma regra é feita interativamente com o especialista que submete esta nova regra ao sistema, cláusula a cláusula.

O sistema analisa uma cláusula (reconhecendo os elementos que a compõe, tais como, atributo, objeto, etc...) e devolve ao especialista o resultado desta análise. Caso o especialista a aprove, o sistema então requer e analisa a cláusula seguinte.

Durante a aquisição da premissa, o sistema reconhece o fim de uma cláusula ao encontrar as palavras *&*, *ou* ou quando o especialista coloca uma entrada vazia (que irá indicar o fim da premissa). No caso de encontrar a palavra *ou* ou *ou 1*, o sistema começa a montar uma cláusula disjuntiva que só irá terminar quando for encontrado o próximo

mo & ou a entrada vazia. Por exemplo, se o usuário entrar com:

$A \& B \text{ ou } C \text{ ou } D \& E$

o sistema entenderá esta premissa como:

$A \& (B \text{ ou } C \text{ ou } D) \& E.$

O sistema não aceita cláusulas do tipo *ou* ou *ou 1* na conclusão. Assim, as cláusulas da conclusão devem terminar com um & e a última delas com a entrada vazia.

Quando todas as cláusulas foram adquiridas, o sistema submete a regra para a aprovação. Caso o especialista a aprove, ela é incorporada ao banco. Caso contrário, é dada ao especialista a possibilidade de alterá-la ou desprezá-la.

Segue-se um exemplo de aquisição de uma regra utilizando menu:

QUE OPERAÇÃO DEVE SER FEITA:

- 1) AQUISIÇÃO DE REGRA
- 2) EXCLUSÃO DE REGRA
- 3) ALTERAÇÃO DE REGRA
- 4) LISTAGEM DE REGRA
- 5) LISTAGEM DE REFERÊNCIAS
- 6) FIM
- 7) DESATIVAÇÃO DE MENU

> 1

REGRA 4

ESTA REGRA É DO TIPO ESPECIAL?

> NÃO

ENTRE COM A CLÁUSULA 1

HA' UM /SISTEMA SINÓTICO/ SE APROXIMANDO &

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA:

ATRIBUTO APROXIMANDO

OBJETO SISTEMA SINOTICO

FUNCAO HA'

VALOR

UNIDADE

OK? (S/N) > S

ENTRE COM A CLAUSULA 2

O SISTEMA TEM POUCA INTENSIDADE

>

QUAL O ATRIBUTO?: > INTENSIDADE

PALAVRAS CONHECIDAS:

22 - HORA ATUAL

23 - DIA ATUAL

24 - MES ATUAL

56 - ATIVIDADE

57 - PROXIMIDADE

58 - ATIVA

59 - PROXIMO

60 - PROXIMA

61 - APROXIMANDO

62 - CONDICOES EM ALTOS NIVEIS

63 - CONDICOES DE GRANDE ESCALA EM 24 HORAS

64 - CONDICOES LOCAIS EM 24 HORAS

65 - CONDICOES LOCAIS PARA MENOS DE 24 HORAS

66 - CONDICOES DE PRESSAO NAS ULTIMAS 24 HORAS

67 - CONDICOES DE PRESSAO NAS ULTIMAS 3 HORAS

68 - CONVERGENCIA DE UMIDADE

69 - BLOQUEIO NO ESCOAMENTO

70 - VERAO

71 - DESENVOLVENDO-SE CONTINUAMENTE

72 - DESENVOLVENDO-SE NA PARTE DA MANHA

- 73 - JATO EM ALTOS NIVEIS
- 74 - PRESSAO ESPERADA
- 75 - PRESSAO
- 76 - LOCAL
- 77 - TENDENCIA DA VELOCIDADE
- 79 - VELOCIDADE

INTENSIDADE E' SINONIMO DE ALGUMA PALAVRA CONHECIDA?

> N

COMO ESTE ATRIBUTO PODE SER CLASSIFICADO?

- 0) ATRIBUTO SIM/NAO
- 1) ATRIBUTO DO TIPO UNICO
- 2) ATRIBUTO DO TIPO MULTIPLO

> 2

COMO OS VALORES RELACIONADOS A ESTE ATRIBUTO DEVEM SER CLASSIFICADOS?

- 0) TIPO TEXTO
- 1) TIPO NUMERICO
- 2) TIPO NAO-NUMERICO (NOMES)

> 2

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA:

ATRIBUTO INTENSIDADE
OBJETO SISTEMA
FUNCAO IGUAL
VALOR POUCA
UNIDADE

OK? (S/N) > S

ENTRE COM A CONCLUSAO 1

AS /CONDICOES DE GRANDE ESCALA EM 24 HORAS/ SAO FAVORAVEIS

>

QUAL O OBJETO? : > /SAO JOSE DOS CAMPOS/

NUMA ESCALA DE 1 A 10, QUAL A CERTEZA DEPOSITADA NESTA CONCLUSAO?

> 4

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA:

ATRIBUTO CONDICoes DE GRANDE ESCALA EM 24 HORAS

OBJETO SAO JOSE DOS CAMPOS

FUNCAO FAVORAVEIS

VALOR

UNIDADE

OK? (S/N) > S

ESTA CONCLUSAO E' DO TIPO PERGUNTAR DE NOVO?

> N

ESTA REGRA DEVE SER MOSTRADA NO ENCADEAMENTO DE 'PORQUE'?

> S

=====

REGRA 4

SE : 1) HA' UM SISTEMA SINOTICO SE APROXIMANDO, &
2) O SISTEMA TEM POUCA INTENSIDADE

ENTAO : 1) EXISTE EVIDENCIA (0.4) DE QUE AS CONDICoes DE GRANDE ESCALA
EM 24 HORAS SAO FAVORAVEIS

=====

** PROPRIEDADES DE INTENSIDADE

** ATUALIZACAO DE PROPATRIBUTOS

SINTAXE PARA PREENCHIMENTO DO CAMPO 'ESPERADO' (UNIDADE =?)

<ELEMENTOS> ::= {<ELEMENTO>}⁺

<ELEMENTO> ::= <NOME> /UM DOS <CLASSE> /

::= (CONSTANTES {<CONSTANTES>}⁺ [<UNIDADE>])

::= (<CONSTANTE>, <CONSTANTE> [<UNIDADE>])

** ENTRE COM OS VALORES ESPERADOS P/ O ATRIBUTO (MAX = 60 CARAC)

> POUCA

> MUITA

>

O VALOR DO ATRIBUTO DEVE SER PERGUNTADO (P) OU DEDUZIDO (D)?

> D

ENTRE COM A PERGUNTA A SER FEITA AO USUARIO (MAX = 90 CARAC)

> QUAL E' A INTENSIDADE DO * ?

>

** QUAL A TRADUCAO DESTE ATRIBUTO? (MAX = 48 CARAC)

> INTENSIDADE

>

** PROPRIEDADES DO ATRIBUTO INTENSIDADE:

ESPERADO = POUCA MUITA

PERGUNTA = QUAL E' A INTENSIDADE DO * ?

PD = 0 TRADUCAO = INTENSIDADE

TUDO OK ? (S/N) > S

REGRA 4 ADQUIRIDA. DESEJA VER A REGRA?

> N

QUE OPERACAO DEVE SER FEITA:

- 1) AQUISICAO DE REGRA
- 2) EXCLUSAO DE REGRA
- 3) ALTERACAO DE REGRA
- 4) LISTAGEM DE REGRA
- 5) LISTAGEM DE REFERENCIAS
- 6) FIM
- 7) DESATIVACAO DE MENU

> 7

3.2.2 - ALTERAÇÃO DE UMA REGRA

Na alteração de uma regra, o sistema interage com o especialista através de uma linguagem de comandos que inclui, exclui ou al

tera cláusulas e subcláusulas.

Um comando de alteração deve obedecer à seguinte sintaxe:

```
<comando> ::= <alteracao> /FIM
<alteracao> ::= <operacao> CLAUSULA <numero> <local> /
                <operacao> CLAUSULA <numero> SUBCLAUSULA <numero>
<operacao> ::= EXCLUIR / ALTERAR / INCLUIR
<operacao> ::= PREMISSA / CONCLUSAO
```

Após cada interação, o sistema expõe a regra alterada ao especialista para sua aprovação. Quando o especialista encerra a alteração, o sistema exclui a regra que estava no banco (como descrito na Seção 3.2.5) e armazena a regra alterada.

3.2.2.1 - INCLUSÃO DE CLÁUSULA OU SUBCLÁUSULA

A inclusão de uma cláusula na premissa ou na conclusão obedece o mesmo esquema: a cláusula é reconhecida, incluída na regra e o descritor é alterado de modo que a nova cláusula seja a primeira dentre as cláusulas.

A inclusão de uma subcláusula pode ser feita numa cláusula do tipo *e* ou numa cláusula do tipo *ou* da premissa. No primeiro caso, um descritor do tipo *ou* é criado, ligando a cláusula do tipo *e* transformada em subcláusula e a nova subcláusula. No segundo caso, a subcláusula é inserida e o descritor é alterado. Em ambos os casos, a subcláusula é inserida na primeira posição.

Segue-se um exemplo de inclusão de uma subcláusula:

```
CODIGO DE INICIALIZACAO:    >    ALTERE REGRA 4
OPERACAO:                   >    INCLUIR CLAUSULA 2 SUBCLAUSULA
ENTRE COM A NOVA CLAUSULA  (TERMINE COM ';'')
```


> HA' /CONVERGENCIA DE UMIDADE/ EM BAIXOS NIVEIS EM /SAO JOSE DOS
> CAMPOS/ ;

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA:
ATRIBUTO..... CONVERGENCIA DE UMIDADE
OBJETO SAO JOSE DOS CAMPOS
FUNCAO HA'
VALOR
UNIDADE

OK? (S/N) > S

=====

REGRA 4 MODIFICADA

SE : 1) HA' UM SISTEMA SINOTICO SE APROXIMANDO, &
2) (2.1 HA' CONVERGENCIA DE UMIDADE EM BAIXOS NIVEIS EM SAO
JOSE DOS CAMPOS, OU
2.2 O SISTEMA TEM POUCA INTENSIDADE)

ENTAO : 1) EXISTE EVIDENCIA (0.4) DE QUE AS CONDICOES DE GRANDE ESCALA
EM 24 HORAS SAO FAVORAVEIS

=====

MAIS ALGUMA ALTERACAO? (S/N) > S

3.2.2.2 - ALTERAÇÃO DE CLÁUSULA OU SUBCLÁUSULA

A alteração de uma cláusula consiste em reconhecer a no
va cláusula, reescrevendo-a sobre a cláusula indicada pelo especialis
ta. O mesmo procedimento se aplica à alteração de uma subcláusula.

Segue-se um exemplo de alteração de cláusula:

OPERACAO: > ALTERAR CLAUSULA 1 PREMISSA

ENTRE COM A NOVA CLAUSULA (TERMINE COM ';')

> HA' UM SISTEMA NO SUL DO BRASIL ;

QUAL O ATRIBUTO? : > LOCAL

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA:

ATRIBUTO LOCAL
OBJETO SISTEMA
FUNCAO HA'
VALOR SUL
UNIDADE

OK? (S/N) > S

=====

REGRA 4 MODIFICADA

SE : 1) HA' UM SISTEMA NO SUL DO BRASIL, &
2) (2.1 HA' CONVERGENCIA DE UMIDADE EM BAIXOS NIVEIS EM SAO
JOSE DOS CAMPOS, OU
2.2 O SISTEMA TEM POUCA INTENSIDADE)

ENTAO : 1) EXISTE EVIDENCIA (0.4) DE QUE AS CONDICOOES DE GRANDE ESCALA
EM 24 HORAS SAO FAVORAVEIS

=====

MAIS ALGUMA ALTERACAO? (S/N) > S

3.2.2.3 - EXCLUSÃO DE CLÁUSULA OU SUBCLÁUSULA

A exclusão de uma cláusula tanto na premissa como na conclusão consiste em rearranjar os ponteiros de modo que aquela cláusula não seja mais acessada. O mesmo se aplica à exclusão de uma subcláusula numa cláusula *ou* ou *ou1* que contenha pelo menos duas subcláusulas.

No caso da cláusula conter somente duas subcláusulas, o descritor *ou* (*ou1*) é desativado, a subcláusula indicada é excluída e a outra subcláusula é transformada em cláusula *e*.

Segue-se um exemplo de exclusão de cláusulas:

OPERACAO: > EXCLUIR CLAUSULA 2 SUBCLAUSULA 1

=====

REGRA 4 MODIFICADA

SE : 1) HA' UM SISTEMA NO SUL DO BRASIL, &

2) O SISTEMA TEM POUCA INTENSIDADE

ENTAO : 1) EXISTE EVIDENCIA (0.4) DE QUE AS CONDICÕES DE GRANDE ESCALA
EM 24 HORAS SÃO FAVORAVEIS

=====

MAIS ALGUMA ALTERACAO? (S/N) > S

OPERACAO: > EXCLUIR CLAUSULA 1 CONCLUSAO

EXCLUSAO IMPOSSIVEL. A CONCLUSAO NAO PODE SER VAZIA

MAIS ALGUMA ALTERACAO? (S/N) > N

ESTA REGRA DEVE SER MOSTRADA NO ENCADEAMENTO DE 'PORQUE'?

> S

REGRA 4 ADQUIRIDA. DESEJA VER A REGRA?

> N

3.2.3 - RECONHECIMENTO DE CLÁUSULAS

O reconhecedor de cláusulas identifica as palavras da frase digitada pelo usuário, monta a cláusula e verifica se ela está correta (por palavra entende-se um identificador, um número ou um conjunto de palavras delimitado por barras "/"). Se a cláusula pertencer à conclusão, é perguntado ao usuário quais são os tipos de contexto que podem ser associados àquela cláusula, qual a crença na sua conclusão e, caso não seja possível deduzir o valor daquela conclusão, se o sistema deve perguntar explicitamente por ele.

A frase de entrada é analisada duas vezes. Na primeira vez tenta-se conseguir todos os parâmetros da cláusula, ou seja, atributo, objeto, função, valor, modificador de função e unidade, verificando a cada palavra se esta se encontra no arquivo de palavras-chave DIRDISK. Estes parâmetros são armazenados no registro da cláusula. Caso algum dos parâmetros principais (atributo, objeto, função, valor) não

tenha sido encontrado, o sistema tenta conseguir este identificador dialogando com o usuário. Cada identificador que o usuário digita é colocado no arquivo DIRDISK com sua classe correspondente e armazenado no registro de cláusula.

Quando o sistema tem todos os parâmetros, é feita uma segunda leitura da frase de entrada e, então, o *esqueleto* da cláusula é montado. No *esqueleto*, os parâmetros são codificados pelo signo "\$" concatenado ao número de sua classe e as demais palavras, pelo número de seu registro no arquivo OUTRASPALAVRAS. Na decodificação, os símbolos são substituídos pelos identificadores apontados pelo registro de cláusula e os números, pelos identificadores referenciados por eles em OUTRASPALAVRAS. O símbolo \$4, que representa *valor*, é substituído pelo próprio número armazenado no registro de cláusula, ou pelo identificador por ele apontado, caso este não seja numérico.

Tomando como exemplo a cláusula:

A /TENDENCIA DA PRESSAO/ E' > 3 EM /SAO JOSE DOS CAMPOS/ e utilizando as informações de DIRDISK e OUTRASPALAVRAS da Figura 3.2 tem-se o seguinte *esqueleto*:

1 \$1 2 \$3 \$4 3 \$2.

Dadas as características do reconhecedor é possível que os parâmetros da cláusula não constem explicitamente no texto. Se por exemplo a frase digitada fosse:

A /TENDENCIA DA PRESSAO/ E' > 3 ,

o sistema não encontraria o objeto, perguntaria por ele e, sendo-lhe dado /SAO JOSE DOS CAMPOS/ como resposta, ao final da análise o *esqueleto* seria:

1 \$1 2 \$3 \$4 ,

sendo os arquivos DIRDISK e OUTRASPALAVRAS e o registro da cláusula como os da Figura 3.2.

DIRDISK

	Palavra	Classe
5	TENDENCIA DA PRESSAO	1 % atributo
6	SAO JOSE DOS CAMPOS	2 % objeto
7	>	3 % funcao

OUTRASPALAVRAS

1	A
2	E'
3	EM

REGISTRO DE CLÁUSULA

ATRIBUTO	OBJETO	FUNÇÃO	VALOR
5	6	7	3

Fig. 3.2 - Status da base de dados.

3.2.4 - ARMAZENAMENTO DE REGRAS

No armazenamento de uma regra, além da incorporação da regra ao banco, faz-se a atualização dos arquivos ATRIBUTOSREF e PROPATRIBUTOS. Pretende-se que futuramente seja verificada a existência de conflitos de compatibilidade entre as regras do banco, segundo os estudos realizados por Senne (1980).

O arquivo ATRIBUTOSREF é atualizado percorrendo cada cláusula e colocando o número da regra no registro correspondente ao atributo da cláusula. O arquivo PROPATRIBUTOS é atualizado para um dado atributo quando este é acessado pela primeira vez, como ilustrado no exemplo na Seção 3.2.1. A sintaxe para preenchimento do campo relativo aos valores aceitáveis para o atributo encontra-se na Seção A.8.

3.2.5 - EXCLUSÃO DE REGRA

Primeiramente, percorre-se a premissa e a conclusão da regra, retirando a referência do atributo de cada cláusula no registro correspondente em ATRIBUTOSREF (Apêndice A). A seguir, a regra é excluída do banco, ou seja, o número desta regra é colocado na pilha de regras desativadas e o registro é zerado.

Segue-se um exemplo de exclusão de regra:

CODIGO DE INICIALIZACAO: > EXCLUA REGRA 4

REGRA 4 EXCLUIDA. DESEJA VER A REGRA?

> S

=====

REGRA 4

SE : 1) HA' UM SISTEMA NO SUL BRASIL, &
2) O SISTEMA TEM POUCA INTENSIDADE

ENTAO : 1) EXISTE EVIDENCIA (0.4) DE QUE AS CONDICÕES DE GRANDE ESCALA
EM 24 HORAS SÃO FAVORAVEIS

=====

3.2.6 - LISTAGEM DE REGRA

Para mostrar uma regra, o sistema acessa o registro correspondente a ela no arquivo REGRASDISK (Apêndice A), decodifica-a e lista-a na tela.

A decodificação consiste em substituir no *esqueleto* de cada cláusula os símbolos especiais pelos parâmetros correspondentes, armazenados no registro da cláusula, e os números correspondentes a registros em OUTRASPALAVRAS pelos seus identificadores. Se uma dada cláusula for do tipo *texto*, o registro correspondente a tal texto no arquivo TEXTODISK é acessado e listado.

Segue-se um exemplo de listagem de regra:

CODIGO DE INICIALIZACAO > MOSTRE REGRA 4

```
=====
REGRA 4
SE      : 1) HA' UM SISTEMA NO SUL DO BRASIL, &
          2) O SISTEMA TEM POUCA INTENSIDADE
ENTAO   : 1) EXISTE EVIDENCIA (0.4) DE QUE AS CONDICÕES DE GRANDE ESCALA
          EM 24 HORAS SAO FAVORAVEIS
=====
```

3.2.7 - LISTAGEM DE REFERÊNCIAS A ATRIBUTOS

O sistema pergunta ao usuário o nome do atributo e acessa os registros correspondentes a ele em REGRASDISK. Primeiramente são listadas todas as regras que contêm aquele atributo na premissa e depois as que o contêm na conclusão.

Segue-se um exemplo de listagem de referências a atributos:

```
CODIGO DE INICIALIZACAO: > REFERENCIAS
REFERENCIAS AO ATRIBUTO: > INTENSIDADE
```

```
=====
ATRIBUTO INTENSIDADE NA PREMISA DE:
REGRA 3 REGRA 4 REGRA 15
=====
```

```
ATRIBUTO INTENSIDADE NA CONCLUSAO DE:
```

```
=====
> /PRECIPITACAO EM 24 HORAS/
=====
```

```
ATRIBUTO PRECIPITACAO EM 24 HORAS NA PREMISA DE:
```

```
=====
ATRIBUTO PRECIPITACAO EM 24 HORAS NA CONCLUSAO DE:
REGRA 24 REGRA 25 REGRA 28 REGRA 29
=====
```

>

3.2.8 - MANIPULAÇÃO DE REGRAS ESPECIAIS

Dada uma certa informação, muitas vezes está ligada a ela uma série de outras informações consideradas óbvias. O sistema utiliza regras especiais para, nestes casos, evitar perguntas desnecessárias. São chamadas regras especiais aquelas que transferem valores de vários atributos para o arquivo VALORCONHECIDO se a crença no valor de um da do atributo é igual a 1. Segue-se um exemplo de regra especial:

TRANSFIRA:

- 1) O Estado é São Paulo &
- 2) A população é de 350000 habitantes

SE :

- 1) O nome da cidade é São José dos Campos.

O sistema adquire regras especiais, modifica-as e as exclui como uma regra normal. Suas únicas restrições são a de somente aceitar *igual* ou um seu sinônimo como função nas cláusulas de premissa e a de não aceitar subcláusulas.

Segue-se um exemplo de aquisição de regra especial:

CODIGO DE INICIALIZACAO: > REGRAESPECIAL

REGRA 42

ENTRE COM A CLAUSULA 1

ESTAMOS NO VERA0 EM /SAO JOSE DOS CAMPOS/

>

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA:

ATRIBUTO VERA0

OBJETO SAO JOSE DOS CAMPOS

FUNCAO IGUAL

VALOR

UNIDADE

OK? (S/N) > S

ENTRE COM A CLAUSULA 2

NAO ESTAMOS NO INVERNO EM /SAO JOSE DOS CAMPOS/

>

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA

ATRIBUTO INVERNO

OBJETO SAO JOSE DOS CAMPOS

FUNCAO NAO IGUAL

VALOR

UNIDADE

OK? (S/N) > S

ENTRE COM A CONCLUSAO 1

/MES ATUAL/ = 3

>

QUAL O OBJETO?: > /SAO JOSE DOS CAMPOS/

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA

ATRIBUTO MES ATUAL

OBJETO SAO JOSE DOS CAMPOS

FUNCAO IGUAL

VALOR 3

UNIDADE

OK? (S/N) > S

=====

REGRA 42

TRANSFIRA:

1) ESTAMOS NO VERA0 EM SAO JOSE DOS CAMPOS, &

2) NAO ESTAMOS NO INVERNO EM SAO JOSE DOS CAMPOS

SE:

1) MES ATUAL = 3

=====

REGRA 42 ADQUIRIDA. DESEJA VER A REGRA?

> N

OK? (S/N) > S

CAPÍTULO 4

MÓDULO DE CONSULTA

O módulo de CONSULTA é a máquina de inferência do sistema, o qual, interagindo diretamente com o usuário, fornece as respostas ao problema dado.

Este módulo utiliza como entrada o banco de conhecimento que contém as regras relativas àquele domínio e os dados específicos daquela consulta fornecidos pelo usuário.

O encadeamento de regras é feito de modo retroativo (backward chaining), ou seja, tendo a conclusão de uma regra, o sistema tentará validar cada cláusula da premissa. Para isto, o sistema reunirá todas as regras cuja conclusão seja a cláusula em questão e avaliará a premissa destas.

O sistema utiliza dois procedimentos interrelacionados, o MONITOR e o BUSCAVALOR, para concluir o valor do *objetivo*, isto é, de um atributo e contexto. O procedimento MONITOR verifica a validade de uma regra e o BUSCAVALOR determina o valor de um dado atributo num dado contexto.

Inicialmente o sistema reúne todas as regras que concluem o objetivo da consulta e aplica o MONITOR a cada uma delas. Se o MONITOR tenta analisar uma cláusula, cujos atributo e contexto ainda não tenham valor, o procedimento BUSCAVALOR é chamado. O BUSCAVALOR então perguntará pelo valor ou reunirá todas as regras que concluem aquele atributo e objeto, aplicando o MONITOR a cada uma delas. Quando o BUSCAVALOR não encontra o valor para um atributo e contexto, a regra que estava sendo analisada é rejeitada.

Segue-se um exemplo de consulta do sistema de previsão pluviométrica.

MODULO DE CONSULTA

(SESSAO ABERTA)

QUINTA-FEIRA 17/OUT/1985 22:13:17

** ENTRE COM O DOMINIO DE APLICACAO:

> PREVISAO

** OBJETIVOS POSSIVEIS

1: PRECIPITACAO EM SAO JOSE DOS CAMPOS EM 24 HORAS

2: PRECIPITACAO EM SAO JOSE DOS CAMPOS EM MENOS DE 24 HORAS

** QUAL O OBJETIVO DA CONSULTA:

> 1

** DESEJA ACOMPANHAR A EXECUCAO? (S/N)

> N

1) QUAL O VALOR DA REGIAO NA CARTA ISALOBARICA ONDE SE LOCALIZA SAO JOSE DOS CAMPOS-1?

> -6 MB (.9)

>

2) QUAL O VALOR DA DEPRESSAO DO BULBO UMIDO (TEMP. AMBIENTE - TEMP. ORVALHO) EM SAO JOSE DOS CAMPOS-1?

> -8 (.8)

>

3) DEVO CRIAR UM NO' ZONA FRONTAL NA ARVORE DE CONTEXTOS?

> S

----- ZONA FRONTAL-1 -----

4) EXISTE ALGUMA OUTRA ZONA FRONTAL?

> NAO

5) QUAL A LOCALIZACAO DA ZONA FRONTAL-1?

> PARANA (.9)

> SAO PAULO (.8)

>

6) QUAL A VELOCIDADE DE ZONA FRONTAL-1?

> 10 KM POR HORA

>

7) ZONA FRONTAL-1 ESTA' ATIVA?

> S

8) ZONA FRONTAL-1 ESTA' DESENVOLVENDO-SE CONTINUAMENTE?

> S

9) DEVE CRIAR UM NO' CAVADO NA ARVORE DE CONTEXTOS?

> S

----- CAVADO-1 -----

10) EXISTE ALGUM OUTRO CAVADO?

> N

11) O(A) CAVADO-1 ESTA PROXIMO (OU SE APROXIMANDO) ?

> S

12) COMO ESTA' A INTENSIDADE DE CAVADO-1?

> AMPLIFICANDO

>

13) EXISTE UM JATO EM ALTOS NIVEIS EM SAO JOSE DOS CAMPOS-1?

> SIM (0.7)

14) DEVO CRIAR O NO' VORTICE CICLONICO NA ARVORE DE CONTEXTOS?

> SIM

----- VORTICE CICLONICO-1 -----

15) HA' ALGUM OUTRO VORTICE CICLONICO ?

> NAO

16) O(A) VORTICE CICLONICO-1 ESTA PROXIMO (OU SE APROXIMANDO) ?

> S

17) DEVE CRIAR O NO' SISTEMA SINOTICO?

> S

----- SISTEMA SINOTICO-1 -----

18) DEVO COLOCAR OUTRO NO' SISTEMA SINOTICO NA ARVORE?

> N

19) O(A) SISTEMA SINOTICO-1 ESTA PROXIMO (OU SE APROXIMANDO) ?

> S

20) COMO ESTA A INTENSIDADE DE SISTEMA SINOTICO-1?

> POUCA

>

21) HA' CONVERGENCIA DE UMIDADE EM BAIXOS NIVEIS?

> SIM (.7)

22) QUAL A LOCALIZACAO DE SISTEMA SINOTICO-1?

> SAO PAULO (.9)

>

23) HA' UM SISTEMA SINOTICO-1 NO SUL DO BRASIL?

> SIM

24) EXISTE UM BLOQUEIO NO ESCOAMENTO?

> SIM

CONCLUSOES:

VALOR DE PRECIPITACAO EM SAO JOSE DOS CAMPOS EM 24 HORAS =

1) VERDADEIRO (FATOR = 0.67)

4.1 - PROCEDIMENTO MONITOR

O procedimento MONITOR (Figura 4.1) tem como parâmetros uma regra e uma de suas conclusões. A regra em questão é avaliada para

todos os contextos existentes na árvore, cujos tipos sejam aplicáveis àquela conclusão. Após a avaliação da premissa, sendo esta verdadeira, o sistema tomará a ação descrita na conclusão.

A avaliação da premissa é feita tomando cada cláusula e verificando sua validade. Quando a cláusula é do tipo *ou*, ela é considerada verdadeira se pelo menos uma das subcláusulas for verdadeira. No caso de cláusula do tipo *ou1*, ela é considerada verdadeira assim que uma de suas subcláusulas for avaliada como verdadeira (sem avaliar as demais). Se uma cláusula *e* ou uma cláusula *ou* ou *ou1* for falsa, a regra é descartada.

Antes de avaliar a premissa, verifica-se qual é o tipo de regra analisada (objeto hierarquicamente mais baixo entre os objetos da premissa ou conclusão). Procura-se então na árvore se existem nós, cujos objetos sejam do tipo desta regra e cujos tipos de contexto estejam relacionados hierarquicamente com o tipo de contexto do nó para o qual a regra esteja sendo analisada. Se não existe nenhum desses nós na árvore, o sistema cria quantos o usuário quiser, utilizando de perguntas armazenadas em PROPATRIBUTOS, e para cada novo nó criado, são deduzidos os valores dos atributos principais do contexto através de BUSCAVALOR. Cada nó que satisfaça as condições acima (tenha ele sido criado naquele instante ou não) é colocado numa lista. Em seguida, o sistema analisa a regra tomando como base cada um dos nós da lista. Na análise de cada cláusula o sistema *sabe* qual o contexto que lhe corresponde, percorrendo o caminho formado pelo nó da lista e o nó raiz, até encontrar o nó que contenha o objeto da cláusula.

A avaliação de uma cláusula da premissa é feita comparando o valor de um atributo deduzido ou perguntando ao usuário e o valor que consta na cláusula. O valor deduzido é procurado no arquivo VALOR CONHECIDO através do atributo e contexto. Caso não se encontre tal valor, o procedimento BUSCAVALOR é chamado para deduzi-lo. A comparação entre os valores é feita de acordo com as funções de avaliação. As funções avaliarão as cláusulas observando os critérios descritos na Seção 2.3. Por exemplo, se a função for *igual* ou *pode-ser*, a avaliação será

verdadeira se os valores forem iguais e a crença na validade do valor deduzido for maior que .2 ou maior que -.2 respectivamente. Quando a função for *definitivo*, a crença no valor deverá ser necessariamente igual a 1.

Depois de avaliada a premissa, toma-se a cláusula da conclusão para a qual o MONITOR foi chamado. Se a função for *mostre busca* -se a mensagem em TEXTODISK, mostrando-a ao usuário, e atualiza-se a crença naquela conclusão.

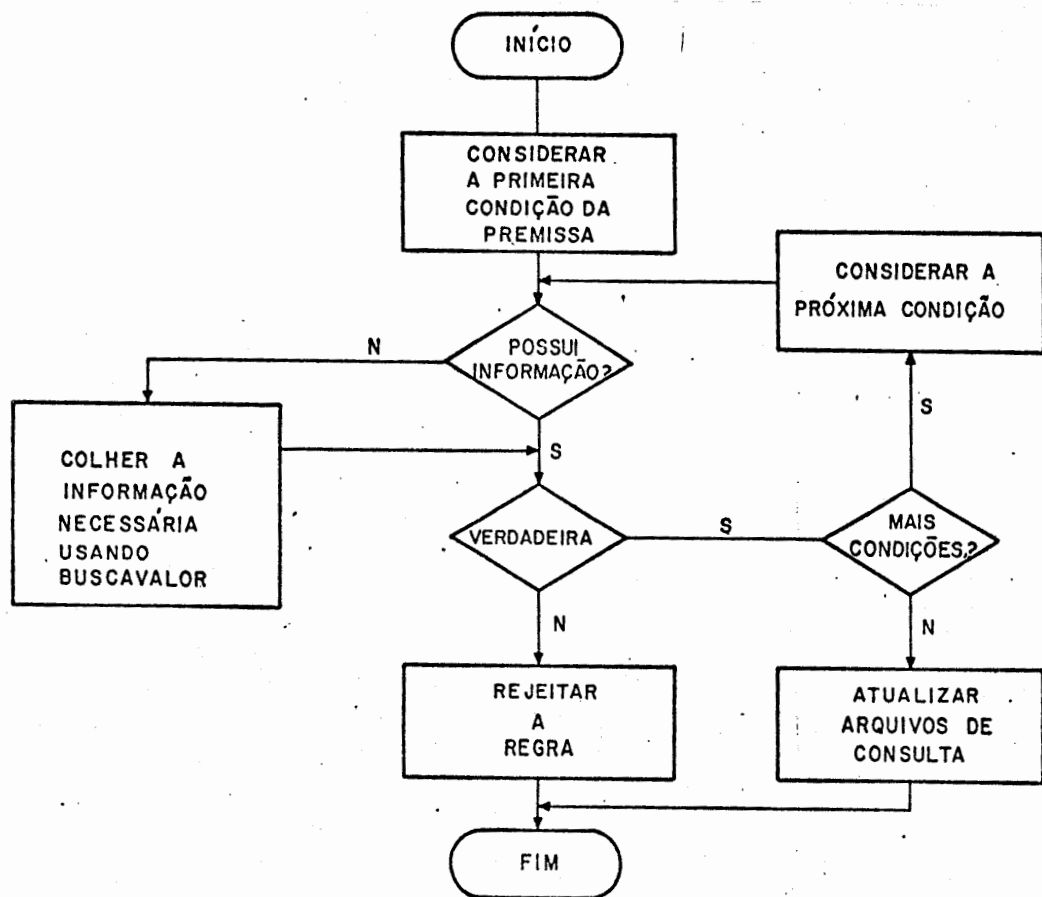


Fig. 4.1 - Esquema simplificado de MONITOR.

4.2 - PROCEDIMENTO BUSCAVALOR

O procedimento BUSCAVALOR (Figura 4.2) obtém os valores para um dado atributo num dado contexto.

Primeiramente é verificado no arquivo PROPATRIBUTOS se o atributo deve ser perguntado ao usuário ou deduzido. No primeiro caso, o valor é perguntado; se o usuário desconhecer a resposta, o sistema tenta deduzir tal valor. No segundo caso, se o sistema não consegue deduzir o valor, este é perguntado ao usuário. Se não se consegue descobrir o valor por nenhuma destas formas, o sistema atualiza o VALORCONHECIDO para aquele atributo e contexto como desconhecido.

Para deduzir os valores para um atributo, o sistema agrupa em uma lista os números de todas as regras que contêm este atributo e objeto na conclusão, armazenados em ATRIBUTOSREF. Toma-se então um elemento desta lista, recupera-se a regra, verifica-se qual conclusão da cláusula contém o atributo em questão e aplica-se o MONITOR àquela regra e conclusão. Procede-se desta forma até que todas as regras da lista tenham sido analisadas.

Para perguntar os valores de um dado atributo, a pergunta armazenada no registro de PROPATRIBUTOS referente àquele atributo é apresentada ao usuário. Em seguida, é verificada a validade da resposta do usuário comparando-a com os valores possíveis (armazenados em PROPATRIBUTOS e CLASEDISK). O usuário pode digitar quantos valores quiser, seguidos ou não pelo fator de certeza. Caso o usuário não forneça o fator de certeza, o sistema entende que este é igual a 1. O sistema não aceita, no entanto, que haja mais de um valor com o fator igual a 1 para atributos do tipo *valor único*. Caso o valor seja numérico, se ele necessitar de unidade e o usuário não fornecê-la, o sistema toma a unidade padrão associada àquele atributo em DIRDISK.

Após o processo descrito acima, o sistema toma as regras que deduzem o atributo e agrupa as do tipo *perguntar de novo*. Estas re

gras, por serem consideradas importantes, merecem uma reavaliação. Isto é feito mostrando a conclusão daquela regra ao usuário e perguntando se ela é verdadeira (caso o valor referente àquela cláusula ainda não esteja na base de dados). Este mecanismo é demonstrado no exemplo de consulta pelas perguntas 22 e 23. Não tendo sido possível concluir o valor *sul* para *sistema sinótico* na conclusão da regra 6 (Apêndice D), o sistema pergunta diretamente ao usuário pela validade da cláusula. Note-se que o uso deste mecanismo é irrelevante para cláusulas de conclusão cujo atributo seja do tipo *sim/não*.

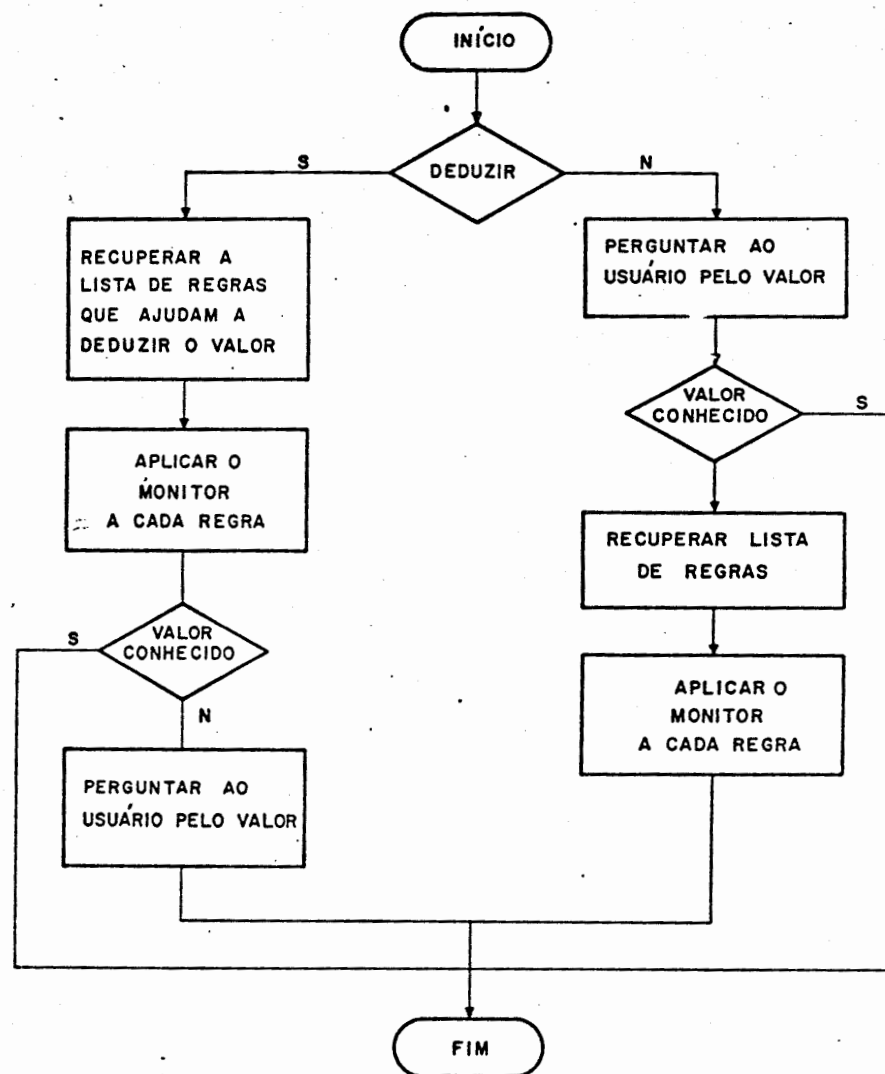


Fig. 4.2 - Esquema simplificado de BUSCAVALOR.

4.3 - MODELO DE VERDADE

O modelo de verdade adotado pelo sistema DIAGNÓSTICO baseia-se no modelo do MYCIN (Shortliffe, 1976). Embora o problema de diagnóstico médica possa ser visto como a atribuição de probabilidades a diagnósticos específicos após a análise de todos os dados relevantes ao problema, o modelo adotado pelo MYCIN não é probabilístico devido à impossibilidade de conseguir dados estatísticos completos e confiáveis.

Considerando que o mesmo problema ocorre em vários outros domínios onde o conhecimento é empírico e inexato, e dada a generalidade a que se propõe o sistema DIAGNÓSTICO, a adoção deste modelo é justificada.

O especialista ao incorporar uma regra ao banco de conhecimento associa a ela um número que caracteriza sua crença na validade da conclusão, caso a premissa seja verdadeira (FC inicial). Como a premissa, por sua vez, é formada por hipóteses, sua validade dependerá dos FC destas hipóteses. Assim, ao longo do processamento o FC de uma dada hipótese é uma função do seu FC inicial e do FC das premissas das regras que o concluem (ou do FC da resposta, caso o atributo tenha sido perguntado). Este valor é a diferença entre a crença e a descrença nesta hipótese, chamados, respectivamente, MC (medida de crença) e MD (medida de descrença).

Formalmente, para uma dada hipótese h , considerando o conjunto de evidências E , estas medidas são definidas como:

$MC(h, E) = x$ a medida de crença em h baseada em E é x

$MD(h, E) = y$ a medida de descrença em h baseada em E é y

$FC(h, E) = MC(h, E) - MD(h, E),$

onde: $0 \leq MC(h, E) \leq 1$

$0 \leq MD(h, E) \leq 1$

$-1 \leq FC(h, E) \leq 1$

Os valores de MC e MD para uma dada hipótese são independentes até que esta seja confirmada ou desconfirmada definitivamente. Neste caso, quando $MC = 1$, o sistema faz com que $MD = 0$ e vice-versa.

Sendo S_i uma evidência (por exemplo, a validade da premissa da regra que tem h como conclusão) e E o conjunto de todos os S_i , as funções que implementam o modelo são:

1) Aumento de evidência:

$$(a) \quad MC(h, S1 \& S2) = 0, \text{ se } MD(h, S1 \& S2) = 1 \\ MC(h, S1) + MC(h, S2) (1 - MC(h, S1)) \quad \text{caso contrário};$$

$$(b) \quad MD(h, S1 \& S2) = 0, \text{ se } MC(h, S1 \& S2) = 1 \\ = MD(h, S1) + MD(h, S2) (1 - MD(h, S1)) \quad \text{caso contrário}.$$

2) Conjunções de hipóteses:

$$(a) \quad MC(h1 \& h2, E) = \min (MC(h1, E) , MC(h2, E)); \\ (b) \quad MD(h1 \& h2, E) = \max (MD(h1, E) , MD(h2, E));$$

3) Disjunções de hipóteses:

$$(a) \quad MC(h1 \text{ ou } h2, E) = \max (MC(h1, E) , MC(h2, E)); \\ (b) \quad MD(h1 \text{ ou } h2, E) = \min (MD(h1, E) , MD(h2, E)).$$

4) Evidência na conclusão:

$$(a) \quad MC(h, S1) = MC' (h, S1) \cdot \max (0 , FC(S1, E)) \\ (b) \quad MD(h, S1) = MD' (h, S1) \cdot \max (0 , FC(S1, E))$$

onde MC' e MD' são os graus de crença e descrença em h quando $S1$ é verdadeiro.

O modelo também afirma que $FC(-h, e) = -FC(h, e)$, ou seja, o grau de crença na negação de uma hipótese com base na evidência e é igual ao grau de descrença na hipótese. Quando o atributo de uma hipótese é do tipo sim/não, a afirmação acima justifica o armazenamento do grau de crença na hipótese h no $MC(h, e)$ e do grau de crença da

negação da hipótese no $MD(h, e)$. O fato de $FC(-h, e) = -FC(h, e)$ também sugere que no caso do atributo relativo à hipótese ser do tipo *único*, a negação da hipótese acarreta um aumento na crença das outras hipóteses relativas àquele atributo. Por exemplo, se aumentada a crença de que *a casa de Maria não é amarela*, estar-se-ia aumentando a crença de que *a casa de Maria é de qualquer outra cor*. Levando em conta o raciocínio acima, pretende-se modificar o modelo de verdade adotado para poder comprovar a validade da afirmação ou negação de uma hipótese, caso não tenha sido possível deduzi-la.

Por exemplo: está se tentando deduzir que um atributo k não é a_4 , dado que:

k é a_1 com $FC = FC_1$,

k é a_2 com $FC = FC_2$,

k é a_3 com $FC = FC_3$.

Se não se conseguisse deduzir um FC diferente de 0 para a_4 , o sistema tomaria $FC' = f(FC_1, FC_2, FC_3)$ e perguntaria ao usuário se seria razoável tomar FC' como crença para a negação de a_4 . Independentemente da resposta do usuário, este FC' não seria armazenado, servindo apenas para avaliar ou invalidar a hipótese naquele instante.

No sistema MYCIN, o limite superior da soma dos FCs de hipóteses mutuamente excludentes não pode ultrapassar 1, ou seja, $\sum_i FC(S_i, E) \leq 1$.

O sistema DIAGNÓSTICO ignora este limite; o único procedimento tomado para modificar a relação entre os FCs de várias hipóteses excludentes é a de no caso de uma hipótese h obter $MC = 1$, além de zerar seu MD , os MCs de todas as hipóteses excludentes a ela serem zerados e fazerem os MDs destas hipóteses iguais a 1. Garante-se assim que $FC(h, E) = 1$ e $FC(H_i, E) = -1$, onde H_i é cada uma das hipóteses excludentes a h . Isto porque a entrada de cada nova regra no banco de conhecimento ocasionaria a comparação desta com todas as outras do banco, fazendo com que o especialista tivesse que se preocupar com o relacionamento de todas as hipóteses mutuamente excludentes. Mesmo que is

to fosse feito, é possível pensar que durante um processamento a soma dos FCs acumulados ultrapassasse 1, embora a soma dos FCs de várias hipóteses não o faça.

4.4 - ÁRVORE DE CONTEXTOS

Como mencionado na Seção 2.2, a árvore de contextos é uma estrutura construída durante a execução do módulo de CONSULTA que explicita a hierarquia entre os contextos criados. A criação desta árvore deve-se à necessidade do sistema saber a qual nó um objeto se refere numa cláusula. Por exemplo, seja a árvore da Figura 4.3 e a regra fictícia:

SE:

- 1) ... paciente ... &
- 2) ... organismo ...

ENTÃO:

- 1) ... cultura ...

e admita-se que, num dado instante da consulta, esta regra tenha sido invocada para o contexto cultura-2. Para a avaliação da premissa desta regra, é preciso saber a qual organismo se refere a cláusula 2, dentre os organismos conhecidos neste ponto da consulta. O sistema avaliará corretamente a cláusula percorrendo a árvore de contextos.

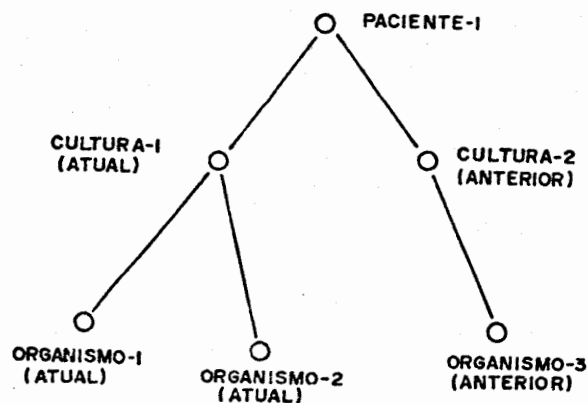


Fig. 4.3 - Árvore de contextos.

Um nó é criado explicitamente quando já existe na árvore pelo menos um nó, cujo tipo de contexto está imediatamente acima do tipo de contexto do nó em questão. Se isto não ocorre, são criados implicitamente tantos nós quantos forem necessários para ligar este nó ao nó raiz, atendendo a hierarquia. Quando um contexto é criado, a busca dos valores dos atributos principais associados ao seu tipo de contexto (armazenados em TIPOSCONTEXTO) são estabelecidos como subobjetivos. Por exemplo, durante uma consulta genérica a árvore contém somente o nó PACIENTE-1 (Figura 4.4 (a)) e a regra referencia-se a um *organismo*; o sistema cria primeiramente os nós de *cultura* (Figura 4.4 (b)) e deduz os valores de seus atributos principais para cada um destes contextos. A seguir são criados os nós de *organismo* (Figura 4.4 (c)).

A existência de atributos principais deve-se à preocupação de forçar uma ordem nas perguntas a serem feitas ao usuário, partindo das mais relevantes para as menos relevantes. É mais natural tomar informações primeiramente sobre a *cultura* onde se encontrou o *organismo* e depois obter informações sobre este *organismo*, do que na ordem inversa.

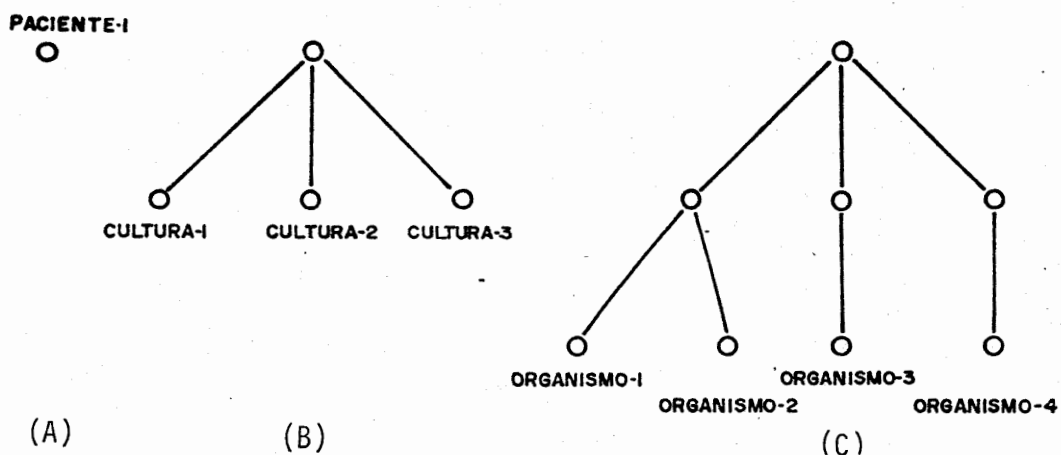


Fig. 4.4 - Evolução da árvore de contextos.

Cada contexto é descrito em um registro de TIPOSCONTEXTO que contém ponteiros para outros registros deste mesmo arquivo, indicando quais os contextos que podem anteceder-lo na árvore.

O sistema DIAGNÓSTICO aplica-se a domínios onde, durante uma consulta, todos os objetos das cláusulas de uma regra possam estar relacionados através de nós num mesmo ramo da árvore.

4.5 - TRATAMENTO DE REGRAS ESPECIAIS

Durante a execução do módulo de CONSULTA, as regras que concluem um dado atributo são agrupadas numa lista e depois analisadas. As regras especiais que têm aquele atributo na premissa são colocadas ao final da lista, garantindo que quando forem avaliadas o atributo já terá sido concluído.

Na avaliação de uma regra especial, caso a conclusão se ja verdadeira, as cláusulas da premissa são consideradas verdadeiras e seus valores são transferidos para VALORCONHECIDO.

CAPÍTULO 5

MÓDULO DE EXPLANAÇÃO

O módulo de EXPLANAÇÃO provê o especialista que constrói a base de conhecimento do sistema e o usuário deste de um mecanismo de acompanhamento do fluxo de uma consulta, justificando as ações tomadas, respondendo sobre o status da consulta em um dado instante e permitindo a alteração dos arquivos dinâmicos do sistema e a simulação de processamento com dados diferentes dos armazenados.

Talvez a maior importância deste módulo seja a de poder acompanhar os passos que levaram a uma resposta errada e quais informações no banco de conhecimento, ou nos dados fornecidos pelo usuário, que levaram o sistema a adotar o caminho errado.

Este módulo está embutido no módulo de CONSULTA e é manipulado pela linguagem de comandos descritos pela Tabela 5.1. Estes comandos recuperam informações dos arquivos estáticos do banco de conhecimento, tais como REGRASDISK(*regra*) ATRIBUTOSREF(*quando é usado e como concluir*) e PROPATRIBUTOS(*respostas*) e sobre o status da consulta com os arquivos dinâmicos VALORCONHECIDO(*valores e como concluiu*), REGRA CONSULTA(*porque e como concluiu*), além de construir cadeias de raciocínio simbólico(*se <atributo 1>...*) e mudar valores de atributos o que acarreta uma nova consulta(*mudar*).

Este módulo poderá ser acessado ao final e durante uma consulta para obter informações sobre esta. Para acessar o módulo no decorrer de uma consulta, basta o usuário digitar qualquer comando da linguagem de comandos, ao invés de responder à pergunta feita pelo sistema. O comando é tratado, suas ações são executadas e, em seguida, o controle retorna ao módulo de consulta que fica esperando a resposta à pergunta feita anteriormente ou uma nova chamada ao módulo de EXPLANAÇÃO.

Logo após a entrada no sistema e antes de iniciada uma consulta, o usuário poderá obter informações sobre a consulta anterior. O módulo de EXPLANAÇÃO utiliza os arquivos obtidos na última sessão, os quais são apagados no início da nova consulta.

TABELA 5.1

COMANDOS DO SISTEMA DE EXPLANAÇÃO

COMANDO	SIGNIFICADO E AÇÕES A TOMAR.
COMANDOS	Mostrar esta lista de comandos.
PERGUNTA ?	Repetir a pergunta feita ao usuário.
REGRA <lista>	Mostrar as regras da <lista>. Caso <lista> seja omitida, mostrar a regra corrente.
PORQUE	Percorrer o arquivo REGRACONSULTA em ordem inversa. A cada chamada consecutiva deste comando, sobe-se um nível na árvore de raciocínio. Este encadeamento termina ao invocar qualquer outro comando.
QUANDO E' USADO <atributo> ?	Mostrar todas as regras que possuem o <atributo> na premissa.
COMO CONCLUIR <atributo> ?	Mostrar todas as regras que possuem o <atributo> na conclusão.
RESPOSTAS	Mostrar os valores possíveis para o atributo corrente.
ARVORE	Mostrar toda a árvore de contextos.
ANCESTRAIS .DE <contexto>	Mostrar os contextos acima de <contexto>.
DESCENDENTES .DE <contexto>	Mostrar os contextos abaixo de <contexto>.

(continua)

Tabela 5.1 - Continuação

VALORES .DE <contexto> .PARA <lista>	Mostrar os valores de cada atributo de <lista> armazenados em VALORCONHECIDO. Caso .PARA <lista> seja omitida, exibir os valores do atributo corrente.
COMO CONCLUIU <atributo> .DE <contexto> = <valor> ?	Recuperar regra ou pergunta utilizada para concluir o <valor> do <atributo> em VALORCONHECIDO. Exibir o trecho da árvore de raciocínio que validou este atributo utilizando informações de REGRACONSULTA.
MUDAR <atributo> .DE <contexto>	Limpar os arquivos dinâmicos, colocar o novo valor de <atributo> em VALORCONHECIDO, apagar a árvore de contextos e recomençar a consulta desde o início. Os atributos que foram anteriormente perguntados e não afetados pela mudança são automaticamente atualizados quando acessados.
SE <atributo1> .DE <contexto> = <valor> .ENTAO <atributo2> ?	Montar uma cadeia de raciocínio simbólico. Todas as condições que acarretariam perguntas a serem feitas por BUSCAVALOR são consideradas verdadeiras.
TERMINE	Terminar a consulta

Segue-se um exemplo da utilização deste módulo no sistema de previsão pluviométrica:

22) QUAL A LOCALIZACAO DE SISTEMA SINOTICO-1?

> REGRA

=====

REGRA 6

SE : 1) (1.1 HA' UM SISTEMA NO PARANA, OU1

1.2 HA' UM SISTEMA EM SANTA CATARINA, OU1

1.3 HA' UM SISTEMA NO RIO GRANDE DO SUL)

ENTAO : 1) COM CERTEZA (1.0) HA' UM SISTEMA NO SUL DO BRASIL

=====

> PORQUE

ESTOU TENTANDO DEDUZIR O ATRIBUTO CONDICÕES DE GRANDE ESCALA EM 24
HORAS NA REGRA 5

SE :

- 1) HA' UM SISTEMA NO SUL DO BRASIL, &
 - 2) HA' UM BLOQUEIO NO ESCOAMENTO, &
 - 3) AS CONDIÇÕES EM ALTOS NÍVEIS NÃO SÃO FAVORÁVEIS
- EXISTE FORTE EVIDÊNCIA (0.9) DE QUE AS CONDIÇÕES DE GRANDE ESCALA EM
24 HORAS NÃO SÃO FAVORÁVEIS (REGRA 5)

> PERGUNTA

22) QUAL A LOCALIZAÇÃO DE SISTEMA SINOTICO-1?

> RESPOSTAS

** VALORES ESPERADOS PARA O ATRIBUTO LOCAL:

1. SÃO PAULO
2. PARANÁ
3. SANTA CATARINA
4. RIO GRANDE DO SUL
5. SUL

> SÃO PAULO (.9)

> VALORES .DE SISTEMA SINOTICO-1

** VALORES DO ATRIBUTO LOCAL PARA O CONTEXTO SISTEMA SINOTICO-1

1. SÃO PAULO (.9)

>

23) HA' UM SISTEMA SINOTICO-1 NO SUL DO BRASIL?

> SIM

24) EXISTE UM BLOQUEIO NO ESCOAMENTO?

> PORQUE

ESTOU TENTANDO DEDUZIR O ATRIBUTO CONDIÇÕES DE GRANDE ESCALA EM 24
HORAS NA REGRA 5

JA' SEI QUE:

HA' UM SISTEMA NO SUL DO BRASIL

ASSIM SENDO, SE :

- 1) HA' UM BLOQUEIO NO ESCOAMENTO, &
 - 2) AS CONDICÕES EM ALTOS NÍVEIS NÃO SÃO FAVORÁVEIS
- EXISTE FORTE EVIDÊNCIA (0.9) DE QUE AS CONDIÇÕES DE GRANDE ESCALA EM
24 HORAS NÃO SÃO FAVORÁVEIS (REGRAS 5)

> PORQUE

ESTOU TENTANDO DEDUZIR O ATRIBUTO PRECIPITAÇÃO EM 24 HORAS NA REGRA 28
SE:

- 1) AS CONDIÇÕES DE GRANDE ESCALA EM 24 HORAS SÃO FAVORÁVEIS, &
 - 2) ESTAMOS NO VERAO
- EXISTE FORTE EVIDÊNCIA (0.8) DE QUE HAVERÁ PRECIPITAÇÃO EM 24 HORAS
EM SÃO JOSÉ DOS CAMPOS (REGRAS 28)

FIM DE ENCADEAMENTO DE 'PORQUE'. A PRÓXIMA CHAMADA DO COMANDO
REINICIARÁ O ENCADEAMENTO.

> SIM

Como mostra o exemplo de aquisição de regra na Seção 3.2.1, o sistema pergunta se a regra deve ser mostrada no encadeamento de *porque*. Isto é feito para evitar que se mostre regras utilizadas no processamento que, embora de importância para o sistema, possam ser consideradas irrelevantes do ponto de vista do usuário. Este procedimento é visto na pergunta 22 do exemplo acima, onde a regra corrente é a de número 6 (Apêndice D). marcada anteriormente, fazendo com que o comando *porque* mostre o encadeamento da regra imediatamente acima desta.

CAPÍTULO 6

CONCLUSÃO E EXTENSÃO FUTURAS

O sistema DIAGNÓSTICO tem sido testado satisfatoriamente na construção de dois pequenos sistemas especialistas. O primeiro diagnostica o mau funcionamento de buzinas de automóveis que, por utilizar um conjunto de regras muito simples, não explora todas as potencialidades do sistema. O segundo sistema prevê a precipitação pluviométrica em São José dos Campos; seu desempenho em relação a um especialista humano ainda não foi avaliado, mas prevêem-se bons resultados pela qualidade de *expertise* utilizada na construção do banco de regras.

Acredita-se que este sistema possa vir a ser utilizado sem grandes modificações nos domínios pertinentes ao INPE, tais como Meteorologia, Sensoriamento Remoto e Detecção de Falhas em Sistemas Espaciais.

Como uma futura extensão, pretende-se dotar o sistema de uma estrutura mais flexível para a árvore de contextos, que permita a utilização, na mesma regra, de objetos que não estão na mesma linha de ascendência sobre o nó raiz.

Pretende-se também desenvolver um novo modelo de verdade, onde a avaliação da premissa de uma regra seja a média ponderada da certeza das cláusulas, tendo como pesos coeficientes de importância (CI) associados a cada cláusula, onde $\sum CI = 1$. O módulo de AQUISIÇÃO tomaria, para cada cláusula, qual o seu fator de importância e reordenaria a regra por ordem decrescente de CI.

Duas outras pequenas extensões são a classificação automática das regras pelo contexto mais baixo e a utilização de listas de valores possíveis para um atributo de acordo com cada objeto ao qual ele possa estar associado.

Outras possíveis extensões para o sistema, embora não no âmbito deste trabalho, são a verificação da compatibilidade das regras do banco de conhecimento, o uso de mecanismos mais inteligentes para a aquisição de conhecimento (Davis, 1976) e uma interface de linguagem natural para dialogar com o usuário durante uma consulta.

REFERÊNCIAS BIBLIOGRÁFICAS

- BARR, A.; FEIGENBAUM, E.A. eds. *The handbook of artificial intelligence*. Los Altos, Ca, William Kaufman, 1981, v.1.
- BARSTOW, D.R.; AIELLO, N.; DUDA, R.O.; ERMAN, L.D.; FORGY, C.L.; GORLIN, D.; GREINER, R.D.; LENAT, D.B.; LONDON, P.E.; McDERMOTT, J.; NII, H.P.; POLITAKIS, P.; REBOH, R.; ROSENSCHEIN, S.; SCOTT, A.C.; van MELLE, W.; WEISS, S.M. Languages and tools for knowledge engineering. In: HAYES-ROTH, F.; WATERMAN, D.A.; LENAT, D.B. eds. *Building expert systems*, Reading, Ma, Addison-Wesley, 1983, cap. 9, p. 283-345.
- BUCHANAN, B.G.; FEIGENBAUM, E.A. DENDRAL and META-DENDRAL: their applications dimension. *Artificial Intelligence*, 11 (1-2):5-24, 1978.
- ARTIFICIAL INTELLIGENCE IS HERE. *Businessweek*, p. 52-60, July 9, 1984.
- DAVIS, R. *Applications of meta level knowledge to the construction, maintenance and use of large knowledge bases*. Ph.D. thesis, Stanford, Computer Science Department, Stanford University, 1976 (AI MEMO 283).
- DAVIS, R.; BUCHANAN, B.G.; SHORTLIFFE, E.H. Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence*, 8 (1):15-45, 1977.
- DUDA, R.O.; GASCHNIG, J.G. Knowledge-based expert systems come of age. *Byte*, 6(9): 238-281, Sept. 1981.
- DUDA, R.O.; HART, P.E.; BARRET, P.; GASCHNIG, J.; KONOLIGE, K.; REBOH, R.; SLOCUM, J. *Development of the PROSPECTOR consultation system for mineral exploration*. Menlo Park, CA, Artificial Intelligence Center, SRI International, 1978. (Final Rept SRI Projects 5821 and 6415).
- DUDA, R.O.; GASCHNIG, J.G.; HART, P.E.; NILSSON, N.J.; REBOH, R.; SLOCUM, J.; SHUTERLAND, G.L. *Development of a computer-based consultant for mineral exploration*, annual report; Menlo Park, CA, Stanford Research Institute, 1977.

- ERMAN, L.D.; LONDON, P.E.; FICKLAS, S.F. *The design and an example use of HEARSAY-III*. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 7., Vancouver, Canada, 1981. *Proceedings*, Stanford, CA, Stanford University, 1981, p. 409-415.
- ERMAN, L.D.; HAYES-ROTH, F.; LESSEE, V.R.; REDDY, D.R. The HEARSAY-II Speech understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12(2): 213-253, 1980.
- FAIN, J.; HAYES-ROTH, F.; SOWIZRAL, H.; WATERMAN, D. *Programming in ROSIE: an introduction by means of examples*. Santa Monica, CA, Rand, 1982 (Tech. Rept. N-1646-ARPA).
- FEIGENBAUM, E.A. The art of Artificial Intelligence: Themes and case studies of knowledge engineering. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 5., Cambridge, MA, 1977. *Proceedings*, Pittsburg, PA, Carnegie-Mellon University, 1977, v.2, p. 1014-1029.
- FORGY, C.L. The OPS5 user's manual. Pittsburg, PA. Computer Science Department, Carnegie-Mellon University, 1981 (Tech.Rept. CMU-CS-81-135).
- GEVARTER, W.B. *An overview of expert systems*. Washington, DC, NASA, 1982 (NASA CR-169197).
- GREINER, R. *RLL-1: A representational language language*. Stanford, CA, Heuristic Programming Project, Computer Science Department, Stanford University, 1980 (Rep. No. HPP-80-9).
- HAMMOND, P. *Logic programming for Expert Systems*. London, Imperial College of Science and Technology, University of London, 1982 (Tech. Rept. DOC82/4).
- HAYES-ROTH, F.; WATERMAN, D.A.; LENAT, D.B. eds. *Building expert systems*. Reading, MA, Addison-Wesley, 1983.
- HELLY Jr. J.J.; BATES, W.V.; CUTLER, M.; KELEN, S. *A representational basis for the development of a distributed Expert System for Space Shuttle Flight Control*. Washington, DC, NASA, 1984 (NASA-TM-58258).
- LAGE FILHO, L. *Um sistema de aquisição de regras e inferência automática aplicado ao diagnóstico diferencial*. Dissertação de Mestrado em Computação Aplicada. São José dos Campos, INPE, 1978 (INPE-1176-TPT/078).

- McDERMOTT, J. R1: the formative years. *AI Magazine* 2(2):21-29, 1981.
- MOSES, J. *A MACSYMA primer*. Cambridge, MA, Computer Science Laboratory, Massachusetts Institute of Technology, 1975 (Mathlab Memo n. 2).
- NAU, D.S. Expert computer systems. *Computer*, 16(2): 63-85, Feb.1983.
- NEWELL, A.; SIMON, H.A. GPS, a program that simulates human thought. In: FEIGENBAUM, E.A.; FELDMAN, A. eds. *Computers and thought*, New York, McGraw-Hill, 1963; Part 2, p. 279-296.
- NII, H.P.; AIELLO, N. AGE (attempt to generalize): A knowledge-based program for building knowledge-based programs. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., Tokyo, 1979. *Proceedings*. Stanford, CA, Stanford University, 1979, v.2, p. 645-655.
- POPLE Jr., H.E. The formation of composite hypotheses in diagnostic problem solving: an exercise in synthetic reasoning. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 5., Cambridge, MA, 1977. *Proceedings*. Pittsburg, PA, Carnegie Mellon University, 1977, v.2, p. 1030-1037.
- RAULT, J.C. Les systemes experts: perspectives industrielles. *Bulletin de Liaison de la Recherche en Informatique et Automatique*, 97: 9-23, Nov. 1984.
- SEGRE, L.M. *Linguagem de programação ALGOL*. Rio de Janeiro, Campus, 1981.
- SENNE, E.L.F. *Compatibilidade de regras de produção usadas como representação de conhecimento em sistemas de consulta*. Tese de Mestrado em Ciências, São José dos Campos, ITA, 1980.
- SENNE, E.L.F.; SIMONI, P.O. *Desenvolvimento de um sistema de consulta usando regras de decisão*. São José dos Campos, INPE, 1981 (INPE-2296-PRE/068). Apresentado na 33ª Reunião Anual da SBPC, Salvador, BA, Jul. 1981.
- SENNE, E.L.F.; SIMONI, P.O. Projeto Diagnóstico: desenvolvimento de um sistema de consulta. *Reunião anual da Sociedade Brasileira para o Progresso da Ciência*, 34, Campinas, Jul. 1982a.

- SENNE, E.L.F.; SIMONI, P.O. *Sistema de consulta com representação de conhecimento por regras de decisão*. São José dos Campos, INPE, 1982b. (INPE-2378-PRE/103). Apresentado no 9º Seminário Integrado de "Software e Hardware", Ouro Preto, MG, Jul. 1982.
- SHORTLIFFE, E.H. *Computer-based medical consultation: MYCIN*. New York, American Elsevier, 1976.
- STEFIK, M.; AIKINS, J.; BALZER, R.; BENOIT, J.; BIRNBAUM, L.; HAYES-ROTH, F.; SACERDOTI, E. The organization of expert systems: a tutorial. *Artificial Intelligence*, 18(2): 135-173, 1982.
- van MELLE, W. *Do you wish advice another horn?*. Stanford, Computer Science Department, Stanford University, 1974 (Rept CS225).
- van MELLE, W. A domain-independent production rule system for consultation programs. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., Tokyo, 1979. *Proceedings*. Stanford, CA, Stanford University, 1979, v.2, p. 923-925.
- WEBSTER, R. Expert systems on microcomputers. *Computer & Electronics*, 23(3): 69-73, Mar. 1985.
- WEISS, S.M.; KULIKOWSKI, C.A. EXPERT: a system for developing consultation models. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6., Tokyo, 1979. *Proceedings*. Stanford, CA, Stanford University, 1979, v.2, p. 942-947.

APÊNDICE A

DESCRIÇÃO DOS ARQUIVOS ESTÁTICOS DO SISTEMA

A.1 - DIRECTORY

O arquivo DIRECTORY contém o dicionário de palavras-chave do sistema. A chave de acesso a este registro é o nome de uma palavra-chave, a qual poderá ter no máximo 42 caracteres, sendo o primeiro caractere necessariamente um literal. O tamanho máximo de DIRECTORY é de 1024 registros, onde cada registro é composto de 10 palavras.

0	CLASSE	SINÔNIMO	ANTÔNIMO	SUBMÚLTIPLO	ASSOC	TIPO
1	CHAVEREG					
2	VALORSUB					
3	NOME					
9						

Fig. A.1 - Registro de DIRECTORY.

- CLASSE - classe da palavra-chave (ver Tabela A.1).
SINÔNIMO - endereço em DIRECTORY do sinônimo principal.
ANTÔNIMO - endereço em DIRECTORY do antônimo principal.
SUBMÚLTIPLO - endereço em DIRECTORY da menor unidade de valor.
ASSOC - (ver Tabela A.2).
TIPO - tipo do atributo: sim/não (=0), único (=1), múltiplo (=2).
CHAVEREG - chave de acesso para ATRIBUTOSREF (classe = atributo) ou para OBJETOETIPOS (classe = objeto)

VALORSUB - valor de uma unidade em relação à menor unidade.

NOME - nome da palavra-chave.

TABELA A.1

CLASSES DE PALAVRAS-CHAVE

CLASSE	TIPO DA PALAVRA CHAVE
0	palavra sem identificação
1	atributo
2	objeto
3	função
4	valor não numérico
5	unidade de valor
6	modificador de função
7	tipo de contexto

TABELA A.2

CAMPO ASSOC.

CLASSE	ASSOC
atributo	.0 = atributo requer valor s/n ou mensagem. .1 = atributo requer um valor numérico sem unidade. .2 = atributo requer valor não-numérico. (endereço da menor unidade) = atributo requer um valor numérico com unidade.
unidade	endereço do atributo associado

A.2 - OUTRASPALAVRAS

Este arquivo contém todos os identificadores que constam das regras e não são palavras-chave do domínio. Cada registro é formado por 7 palavras e não há limite para o tamanho do arquivo.

A.3 - TIPOSCONTEXT0

Cada registro de TIPOSCONTEXT0 (Figura A.2) armazena as características de um contexto, isto é, da associação de um objeto a um tipo de contexto. O tamanho máximo de TIPOSCONTEXT0 é de 256 registros, sendo cada registro formado por 40 palavras. Cada objeto pode ter no máximo 8 contextos a ele associados.

			ASSUMIR	NUMASSOCIADOS		
0	OBJETO		ASSOCIADO(1)	ASSOCIADO(2)	ASSOCIADO(3)	ASSOCIADO(4)
1	ATRIBUTO PRINCIPAL(1)	ATRIBUTO PRINCIPAL (2)	ATRIBUTO PRINCIPAL(3)	ATRIBUTO PRINCIPAL (4)		
2	TIPOCNTXT	LENG(1)	LENG(2)	LENG(3)		
3	TRADUZ PRIMEIRO NÓ NÓ ADICIONAL					
29						

Fig. A.2 - Registro de TIPOSCONTEXT0.

- OBJETO - ponteiro para o objeto em DIRDISK
- ASSUMIR - indica se o primeiro nó deste tipo deve ser considerado como obrigatório (= 1) ou não (= 0).
- NUMASSOCIADOS - número de tipos de contexto.

- A.4 -

ASSOCIADO (x)	- lista dos números dos tipos de contextos (ponteiros para este mesmo arquivo) que podem aparecer imediatamente acima de contexto deste tipo na árvore.
ATRIBUTOPRINCIPAL(x)	- lista dos atributos (ponteiros para DIRDISK) que deverão ser buscados quando um contexto deste tipo é criado.
TIPOCNTXT	- ponteiro para o tipo de contexto em DIRDISK.
LENG(x)	- tamanho das cadeias traduz (x=1), primeiro não (x=2) e não adicional (x=3).
TRADUZ	- forma do nome do contexto que será apresentado ao usuário (p.ex.: <i>organismo atual</i> -).
PRIMEIRO NO'	- mensagem a ser mostrada ao usuário quando se tentar adicionar o primeiro não deste tipo à árvore (se assumir = 0, esta mensagem é na verdade uma pergunta que irá esperar resposta sim/não).
NO' ADICIONAL	- mensagem a ser mostrada ao usuário no caso de se tentar adicionar novos não deste tipo à árvore de contextos (é uma pergunta que espera sim/não como resposta).

A.4 - OBJETOETIPOS

Este é um arquivo auxiliar que contém para cada objeto os tipos de contextos a ele associados. O tamanho de OBJETOETIPOS (Figura A.3) é o número de objetos do domínio, e cada registro é formado por 4 palavras.

0	NULO		CONTEXTTO (1)	CONTEXTTO (2)	CONTEXTTO (3)	CONTEXTTO (4)
	LTOTAL					
1			CONTEXTTO (5)	CONTEXTTO (6)	CONTEXTTO (7)	CONTEXTTO (8)
2	TNULO		NROTIPO (1)	NROTIPO (2)	NROTIPO (3)	NROTIPO (4)
3			NROTIPO (5)	NROTIPO (6)	NROTIPO (7)	NROTIPO (8)

Fig. A.3 - Registro de OBJETOETIPOS.

- NULO - indica se há contexto associado ao objeto (=0) ou não (=1).
- LTOTAL - número de tipos de contexto associados ao objeto.
- CONTEXTTO(I) - ponteiro para o arquivo DIRDISK.
- TNULO - ponteiro para o registro em TIPOSCONTEXTTO quando não há tipos de contextos associados ao objeto.
- NROTIPO(I) - ponteiro do contexto(i) no arquivo TIPOSCONTEXTTO.

A.5 - ESQUELETOSDISK

Este arquivo armazena o *esqueleto* de uma cláusula qualquer. O *esqueleto* é uma sequência de números e símbolos, onde os números são ponteiros para o arquivo OUTRASPALAVRAS e os símbolos são formados por "\$" seguidos do número da classe da palavra-chave em questão. Este arquivo tem no máximo 1024 registros, sendo cada registro formado por 20 palavras.

A.6 - TEXTODISK

O arquivo TEXTODISK armazena textos que correspondem a valores para a função de predicado *mostrar*. O texto consiste numa cadeia de caracteres finalizados por ";". O tamanho de cada registro é de 90 palavras, e o arquivo não tem tamanho máximo.

A.7 - ATRIBUTOSREF

O arquivo ATRIBUTOSREF (Figura A.4) armazena as referências a atributos nas regras. Este arquivo é acessado através do campo *chavereg* do registro que contém um dado atributo em DIRDISK; se acessado através de *chavereg*, o registro contém as referências ao atributo na conclusão de regras e, se acessado através de *chavereg-1*, o registro refere-se ao atributo na premissa de regras. Cada registro tem 180 palavras e cada palavra refere-se a uma regra que contém aquele atributo. Os registros deste arquivo são encadeados devido à necessidade de frequente atualização. O tamanho máximo de ATRIBUTOSREF é de 1024 registros.

0	TIPO (0)	NÚMERODAREGRA(0)	NEGAÇÃO (0)	RLINK (0)
1	TIPO (1)	NÚMERODAREGRA(1)	NEGAÇÃO (1)	RLINK (1)
2				
179				

Fig. A.4 - Registro de ATRIBUTOSREF.

- TIPO(x) - indica se é regra especial (=1) ou não.
- NUMERODAREGRA(x) - número da regra em REGRASDISK que contém o atributo.
- NEGACAO(x) - indica a existência de negação (=1) ou não (=0) do atributo na regra em questão.
- RLINK(x) - aponta a próxima palavra.

A.8 - REGRASDISK

O arquivo REGRASDISK (Figura A.5) armazena as regras de decisão que formam um dado domínio de conhecimento. Além de cláusulas

simples, cada regra tem um descritor de regra, um descritor de premissa, um descritor de conclusão e tantos descritores *ou* quantas forem as cláusulas *ou* (*ou1*) existentes na regra. Um registro ocupa 180 palavras, cada descritor ocupa uma palavra, uma cláusula de premissa ocupa 3 pa lavras e uma cláusula de conclusão ocupa 5 palavras. O número total de registros deste arquivo é no máximo de 1024 registros.

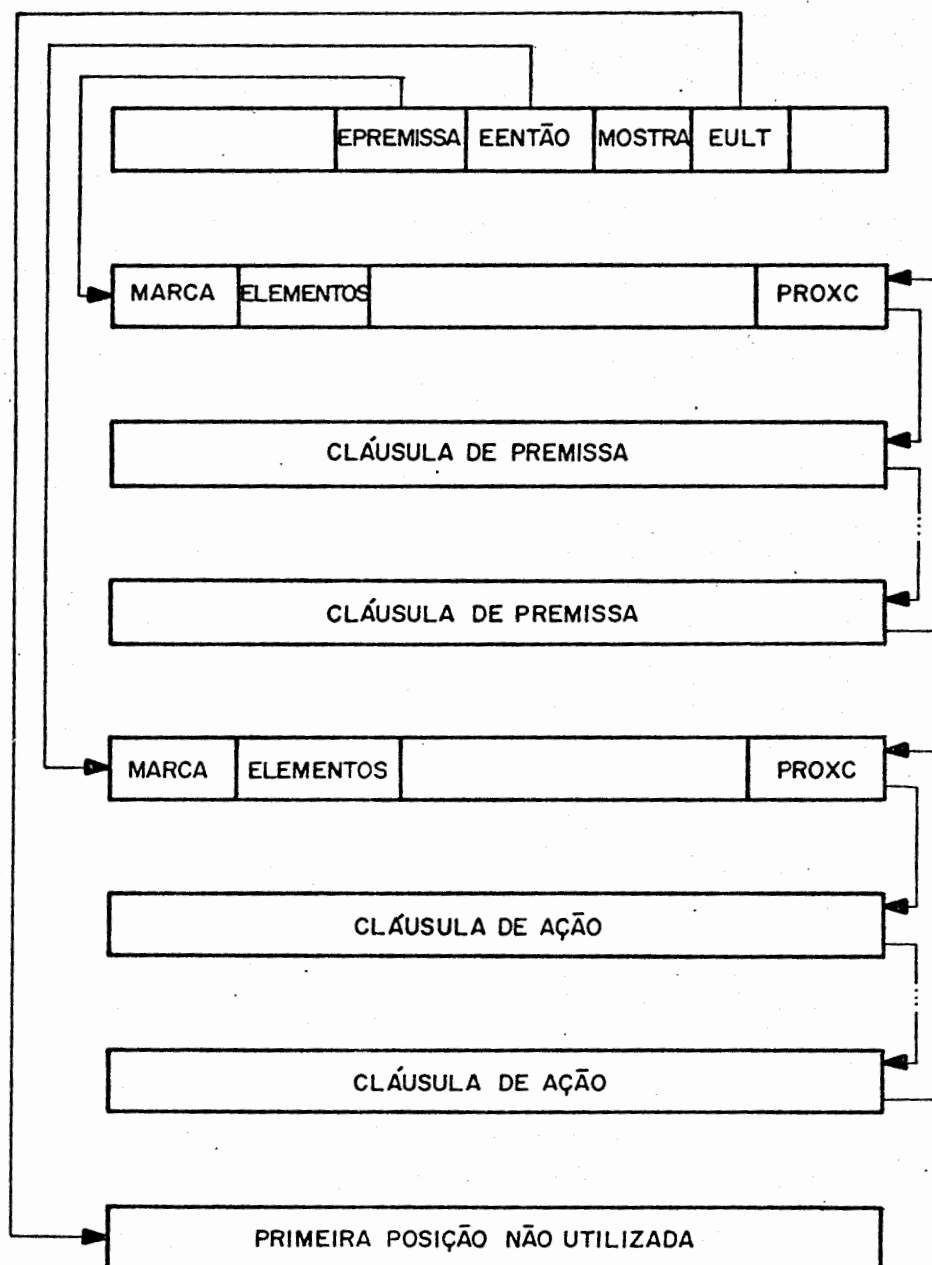


Fig. A.5 - Estrutura de uma regra de decisão.

EPREMISSA - endereços do descritor da premissa.
EENTAO - endereço do descritor da ação.
MOSTRA - indica se a regra deve ser mostrada em *porque* (=0) ou não (=1).
EULT - endereço da primeira posição não-utilizada.
MARCA - marcador de cláusula simples (=0), de premissa (=1), de ação (=2) ou de cláusula *ou* (=3).
ELEMENTOS - número de componentes da premissa ou ação.
PROXC - endereço da próxima cláusula.

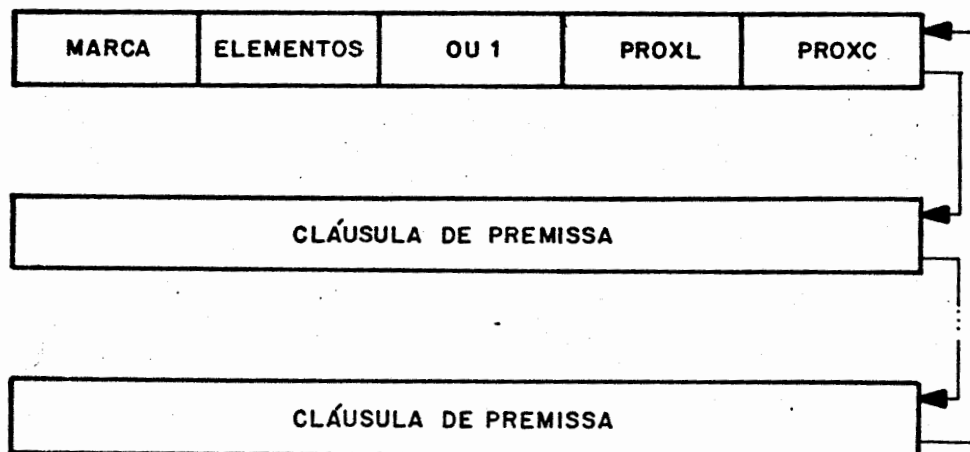


Fig. A.6 - Estrutura de uma cláusula *ou*.

OU1 - indica se a cláusula é do tipo *ou1* (=1).
PROXL - endereço da próxima cláusula.
PROXC - endereço da próxima subcláusula.

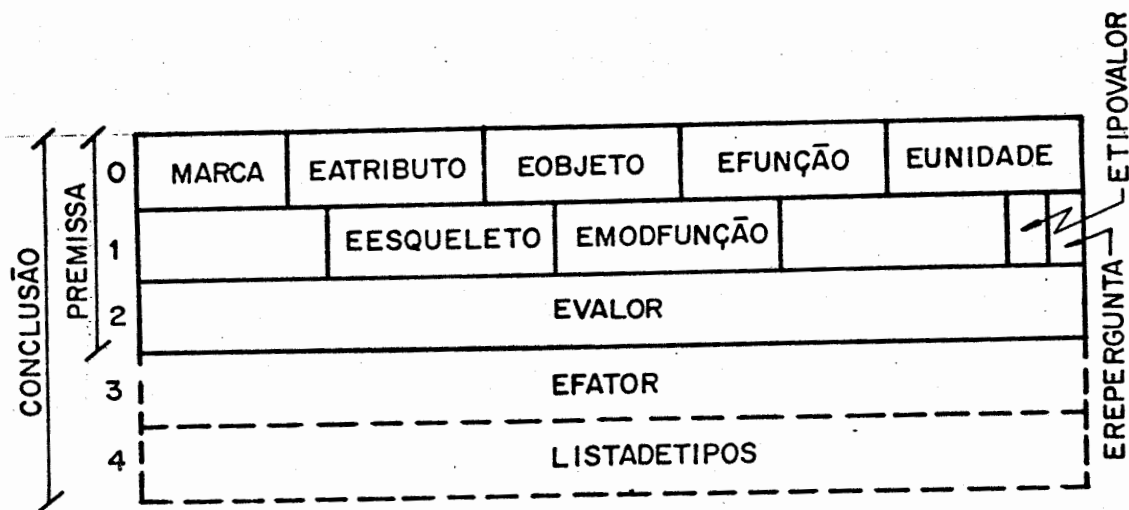


Fig. A.7 - Cláusula simples.

- EATRIBUTO - ponteiro para atributo em DIRDISK.
- EOBJETO - ponteiro para objeto em DIRDISK.
- EFUNCAO - ponteiro para função em DIRDISK.
- EUNIDADE - ponteiro para unidade em DIRDISK.
- EESQUELETO - ponteiro para *esqueleto* em ESQUELETOSDISK.
- EMODFUNCAO - ponteiro para modificador de função em DIRDISK.
- ETIPOVALOR - indica se o valor é numérico (=1) ou não (=0).
- EREPERGUNTA - indica se a condição daquela cláusula é perguntada pela segunda vez.
- EVALOR - ponteiro para valor em DIRDISK, ou ponteiro para texto em TEXTODISK (valor não-numérico), ou o próprio número.
- EFATOR - fator de certeza.
- LISTADETIPOS - tipos de contexto aplicáveis ao objeto da cláusula.

A.9 - PROPATRIBUTOS

O arquivo PROPATRIBUTOS (Figura A.8) contém as propriedades de cada atributo de um dado domínio. A chave para acesso do registro correspondente a um dado atributo é *chavereg/2*. O tamanho máximo de PROPATRIBUTOS é de 512 registros, e cada registro é composto de 72 palavras.

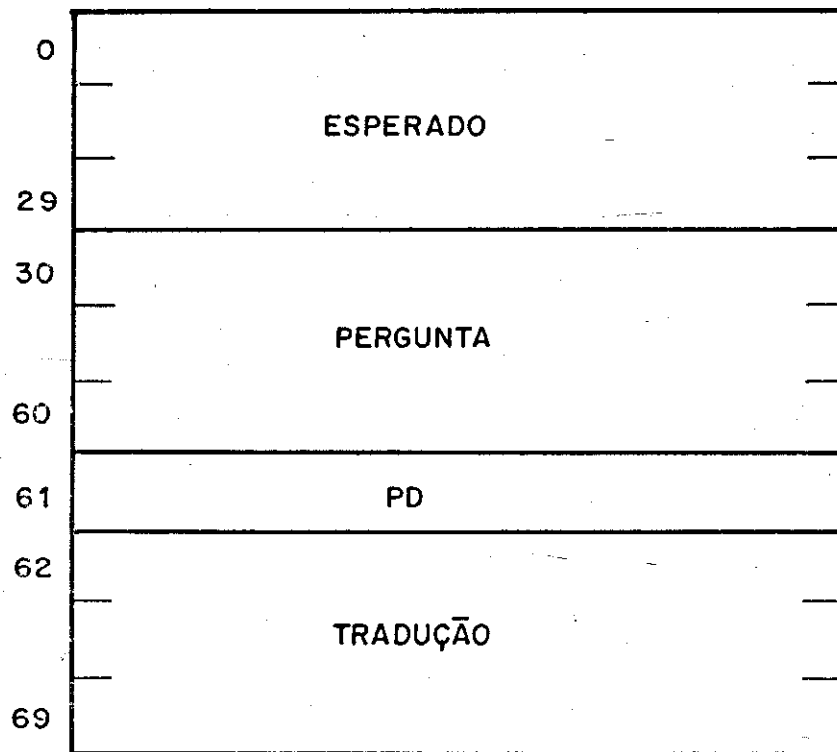


Fig. A.8 - Registro de PROPATRIBUTOS.

ESPERADO - lista dos possíveis valores para o atributo. Sintaxe:

```
<elementos> ::= {<elementos>}+  
<elemento>  ::= <não-numérico> <numérico>  
<não-numérico> ::= <nome>/UMDOS <classe>  
<numérico>    ::= (CONSTANTES{<constante>}+ [<unidade>])  
                ::= (<constante> , <constante> [<unidade>])  
<classe> ::= ver A.10
```

PERGUNTA - sentença a ser mostrada ao usuário. O nome do contexto é codificado por "*" (atributo de valor único ou sim/não) ou por "\$" (atributo de valor múltiplo) e substituído quando acessado.

PD - indica se o atributo deve ser perguntado (=0) ou deduzido (=1).

TRADUCAO - usado para comunicação com o usuário ("*" é substituído pelo contexto). Por exemplo: a identidade de "*".

A.10 - CLASEDISK

Este arquivo contém em cada registro uma classe de valores. O primeiro identificador do registro é o nome da classe, e todas as demais cadeias de caracteres são os valores a ele associados. As classes são acessadas através de PROPATRIBUTOS, e uma classe pode ser referenciada por mais de um atributo. Cada registro de CLASEDISK é formado por 20 palavras e não há número máximo de registros.

A.11 - OBJETIVOS

Este arquivo armazena os objetivos que o sistema é capaz de concluir. O atributo e objeto escolhidos pelo usuário são os primeiros parâmetros que o sistema tenta deduzir. O arquivo OBJETIVOS (Figura A.9) tem no máximo 12 registros, sendo cada registro formado por 14 palavras.

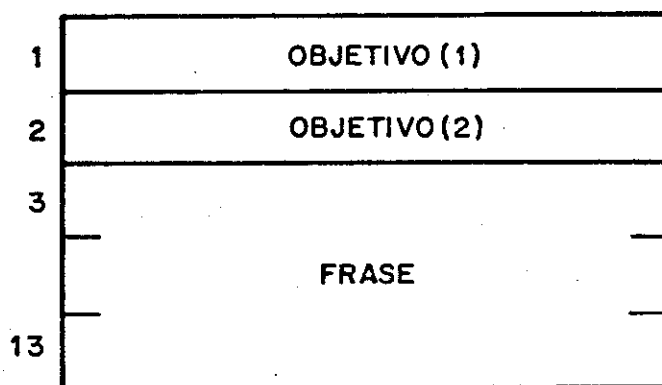


Fig. A.9 - Registro de OBJETIVOS.

OBJETIVO(1) - ponteiro para o atributo em DIRDISK.

OBJETIVO(2) - ponteiro para o objeto em DIRDISK.

FRASE - frase a ser mostrada ao usuário apresentando o objetivo numa forma mais coloquial.

APÊNDICE B

DESCRIÇÃO DOS ARQUIVOS DINÂMICOS DO SISTEMA

B.1 - REGRACONSULTA

O arquivo REGRACONSULTA (Figura B.1) documenta o fluxo de uma dada consulta. Cada registro deste arquivo refere-se à utilização de uma regra ou à criação de um contexto. A Figura B.2 ilustra parte da árvore construída durante uma sessão fictícia. Os nós C1 e C2 referem-se a contextos criados implicitamente, e C3 refere-se a um contexto criado explicitamente. Os índices das regras não correspondem necessariamente à ordem de suas aplicações. O arquivo tem no máximo 1024 registros, onde cada registro ocupa 3 palavras.

	INDICADOR	PAI	IRMÃO	FILHO	CONCLUSÃO	TIPO	SUCESSO
0	PRÓXIMO	PROXAPLICAÇÃO	CRIACONTEXTO	CONTEXTO	ATRIBUTO		
1	CLÁUSULA	SUBCLÁUSULA	CERTEZA	NEGAÇÃO	ESPECIAL		

Fig. B.1 - Registro de REGRACONSULTA.

- INDICADOR - ponteiro para REGRADISK (tipo=0) ou para a árvore de contextos (tipo=1).
- PAI - número do registro que descreve a regra ou contexto que ocasionou a criação deste registro.
- IRMAO -
- próxima premissa da regra (quando o pai do registro é do tipo *regra* e tanto o registro quanto o irmão são do tipo *regra*).
 - próxima avaliação de atributo principal (quando o pai é do tipo *contexto* e tanto o registro como o irmão são do tipo *regra*).

	● próximo contexto (quando o registro e o irmão são do tipo <i>contexto</i>, não importando o tipo do pai).
FILHO	- ● primeira cláusula da premissa (quando o registro é do tipo <i>regra</i> e o pai também). ● primeira avaliação de atributo principal (quando o registro é do tipo <i>regra</i> e o pai é do tipo <i>contexto</i>).
CONCLUSAO	- número da conclusão analisada (tipo=0).
TIPO	- indica se o registro refere-se a uma regra (=0) ou a um contexto (=1).
SUCESSO	- indica se a regra foi concluída com sucesso (=0) ou não (=1).
PROXIMO	- próxima regra aplicada ao mesmo contexto (tanto para registro com pai do tipo <i>regra</i> como para registro com pai do tipo <i>contexto</i>).
PROXAPLICACAO	- próxima aplicação da regra para outro contexto.
CRIACONTEXTTO	- primeiro contexto criado (tanto para registro com pai do tipo <i>regra</i> como para registro com pai do tipo <i>contexto</i>).
CONTEXTTO	- contexto a que a regra está sendo aplicada (ponteiro para a árvore de contextos).
ATRIBUTO	- ponteiro para atributo em DIRDISK.
CLAUSULA	- cláusula da regra que motivou a criação do nó.
SUBCLAUSULA	- subcláusula da regra que motivou a criação do nó.
CERTEZA	- crença na hipótese dada a aplicação da regra.
NEGACAO	- atributo negado (=1) ou não (=0).
ESPECIAL	- indica se a regra é do tipo especial (=1) ou não (=0).

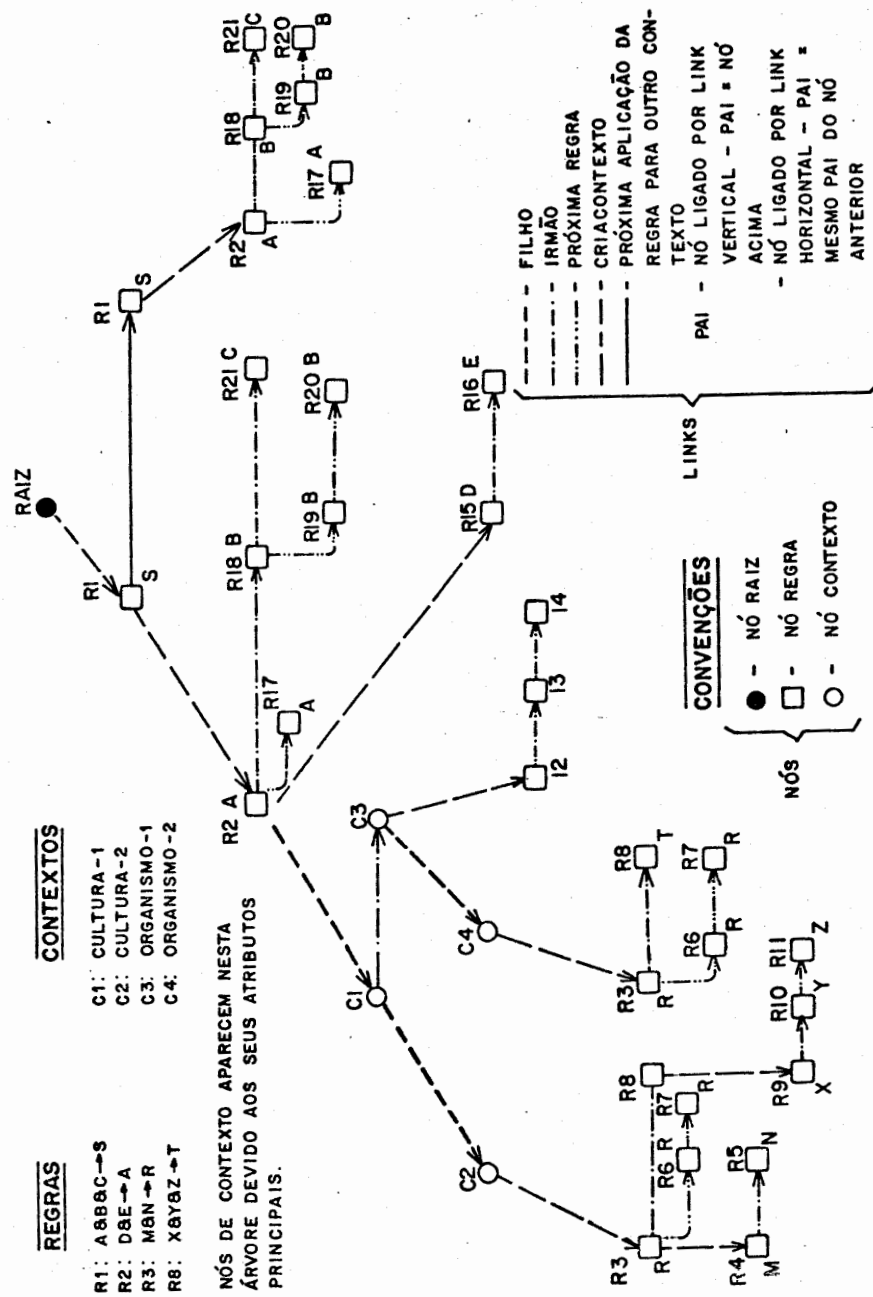


Fig. B.2 - Desenvolvimento da árvore de raciocínio.

B.2 - VALORCONHECIDO

O arquivo VALORCONHECIDO (Figura B.3) contém as listas dos valores conhecidos para cada atributo. O registro referente a um dado atributo é acessado através de *chavereg/2* (*chavereg* do registro do atributo em DIRDISK). Cada elemento da lista que forma o registro refere-se a um valor do atributo aplicado a um contexto. Se o sistema não conseguiu um valor e o usuário também o desconhece, cria-se um elemento que atesta que o valor é desconhecido para tais atributos e contexto. Cada registro ocupa 30 palavras, e cada elemento da lista é formado por 2 palavras; portanto, a lista tem no máximo 15 elementos. O número de registros é o número de atributos do domínio.

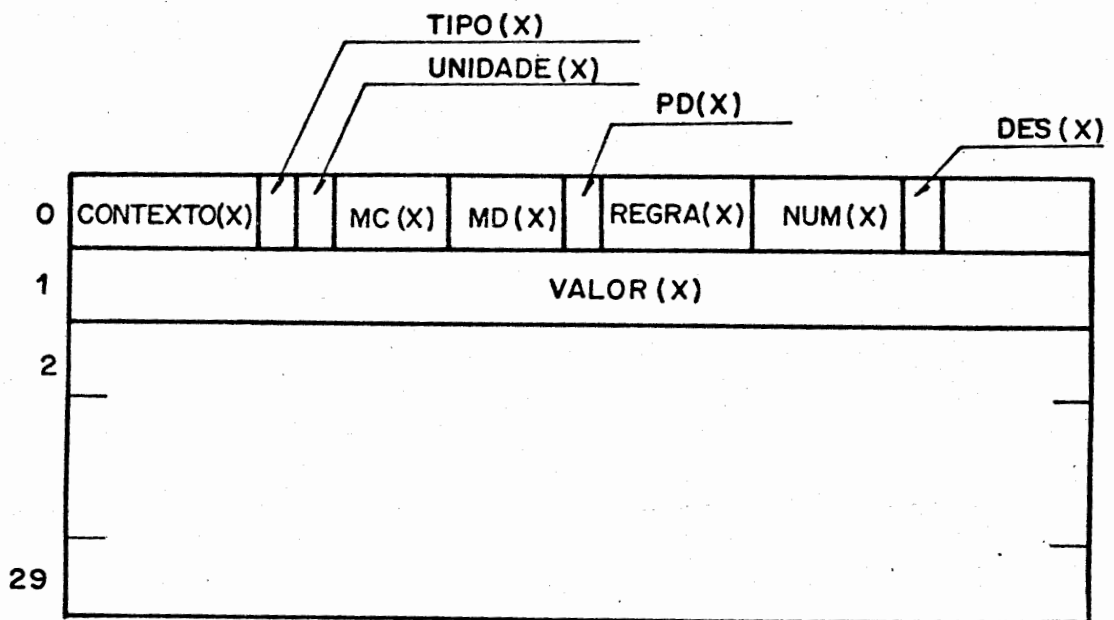


Fig. B.3 - Registro de VALORCONHECIDO.

- CONTEXTO (X) - número do contexto.
- TIPO (X) - valor não-numérico (=0) ou numérico (=1).
- UNIDADE (X) - presença (=1) ou não (=0) de unidade.
- MC (X) - fator de crença.
- MD (X) - fator de descrença.

- PD (X) - valor perguntado (=1) ou deduzido (=0).
- REGRA (X) - primeira regra que concluiu o valor (ponteiro para REGRACONSULTA).
- NUM (X) - número da conclusão da regra (pd = 0) ou número da pergunta (pd = 1).
- DESCONHECIDO (X) - indica se o valor é conhecido (=0) ou não (=1).
- VALOR (X) - valor (tipo (x)=1) ou ponteiro para o dicionário ou arquivo dos textos (tipo (x)= 0).

APÊNDICE C

SISTEMA ESPECIALISTA PARA DIAGNÓSTICO DO MAU FUNCIONAMENTO DE BUZINAS DE AUTOMÓVEIS

C.1 - REGRAS

REGRA 1:

- SE :1) A buzina não funciona é um sintoma da buzina &
2) O relê faz clique quando o botão da buzina é pressionado &
3) A lâmpada de teste acende quando uma ponta é aterrada e a
outra é conectada ao terminal do fio verde do relê enquanto
o botão da buzina é pressionado.
- ENTÃO :1) Com certeza (1.0) há uma abertura entre o fio-verde do relê
e o terra.

REGRA 2:

- SE :1) A buzina não funciona é um sintoma da buzina &
2) O relê faz clique quando o botão da buzina é pressionado &
3) A lâmpada de teste não acende quando uma ponta é aterrada e
a outra é conectada ao terminal do fio verde do relê enquan
to o botão é pressionado.
- ENTÃO :1) Com certeza (1.0) o diagnóstico para a buzina é trocar o re
lê.

REGRA 3:

- SE :1) A buzina não funciona é um sintoma da buzina &
2) O relê não faz clique quando o botão da buzina é pressiona
do &
3) A lâmpada de teste acende quando uma ponta é aterrada e a
outra é conectada ao terminal do fio verde do relê enquan
to o botão da buzina é pressionado.

ENTÃO :1) Com certeza (1.0) há uma abertura entre o fio preto do relê e o terra.

REGRA 4:

SE :1) A buzina não funciona é um sintoma da buzina &
2) O relê não faz clique quando o botão da buzina é pressionado &
3) A lâmpada de teste não acende quando uma ponta é aterrada e a outra é conectada ao terminal do fio preto do relê enquanto o botão da buzina é pressionado.

ENTÃO :1) Com certeza (1.0) o diagnóstico para a buzina é trocar o relê.

REGRA 5:

SE :1) Há uma abertura entre o fio verde do relê e o fio terra &
2) A lâmpada de teste acende quando uma ponta é aterrada e a outra é conectada ao terminal do fio no conector enquanto o botão da buzina é pressionado.

ENTÃO :1) Com certeza (1.0) há uma abertura entre o fio no conector na buzina e o fio terra.

REGRA 6:

SE :1) Há uma abertura entre o fio no conector e o terra &
2) Não há um bom-terra na buzina.

ENTÃO :1) Com certeza (1.0) o diagnóstico para a buzina é consertar o fio terra na buzina.

REGRA 7:

SE :1) Há uma abertura entre o fio verde do relê e o terra &
2) A lâmpada de teste não acende quando uma ponta é aterrada e a outra é conectada ao terminal do fio no conector enquanto o botão da buzina é pressionado.

ENTÃO : 1) Com certeza (1.0) o diagnóstico para a buzina é consertar a abertura no fio entre o relê e a buzina.

REGRA 8:

SE : 1) Hã uma abertura entre o fio no conector e o fio terra &
2) Hã um bom-terra na buzina.

ENTÃO : 1) Com certeza (1.0) o diagnóstico para a buzina é trocar a buzina.

REGRA 9:

SE : 1) Hã uma abertura entre o terminal do fio preto do relê e o do fio terra.
2) A lâmpada de teste não acende quando uma ponta é aterrada e a outra é conectada ao terminal do fio preto no turn signal conector enquanto o botão da buzina é pressionado.

ENTÃO : 1) Com certeza (1.0) o diagnóstico para a buzina é checar contatos sujos no relê, senão consertar a abertura no fio entre o turn signal conector e o relê.

REGRA 10:

SE : 1) Hã uma abertura entre o terminal do fio preto do relê e o do fio terra &
2) A lâmpada de teste acende quando uma ponta é aterrada e a outra conectada ao fio preto no turn signal conector enquanto o botão da buzina é pressionado.

ENTÃO : 1) Com certeza (1.0) hã uma abertura entre o fio preto no turn signal conector e o fio terra.

REGRA 11:

SE : 1) Hã uma abertura entre o fio preto no turn signal conector e o fio terra &
2) A lâmpada de teste acende quando uma ponta é aterrada e a outra conectada ao contact assembly da buzina.

ENTÃO :1) Com certeza (1.0) há uma abertura entre o contact assembly da buzina e o fio terra.

REGRA 12:

SE :1) Há uma abertura entre o fio preto no turn signal conector e o fio terra &

2) A lâmpada de teste não acende quando uma ponta é aterrada e a outra conectada ao contact assembly da buzina enquanto o botão da buzina é pressionado.

ENTÃO :1) Com certeza (1.0) o diagnóstico para a buzina é consertar a abertura no fio entre o turn signal conector e o contact assembly da buzina.

REGRA 13:

SE :1) Há uma abertura entre o contact assembly da buzina e o fio terra &

2) A correia do fio terra na coluna da direção está quebrada.

ENTÃO :1) Com certeza (1.0) o diagnóstico para a buzina é trocar a rosca de engate da coluna da direção.

REGRA 14:

SE :1) Há uma abertura entre o contact assembly da buzina e o fio terra &

2) A correia do fio terra na coluna da direção não está quebrada.

ENTÃO :1) Com certeza (1.0) o diagnóstico para a buzina é consertar o fio terra nos fios contatos da buzina até os botões ou borda da buzina.

C.2 - EXEMPLO DE AQUISIÇÃO

CODIGO DE INICIALIZACAO: > REGRANOVA

REGRA 4

ENTRE COM A CLAUSULA 1

> A BUZINA NAO FUNCIONA E' UM SINTOMA DA BUZINA &

EXISTE SINONIMO PARA 'NAO FUNCIONA'? (S/N)

> NAO

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA:

ATRIBUTO SINTOMA

OBJETO BUZINA

FUNCAO NAO FUNCIONA

VALOR

UNIDADE

OK? (S/N) > SIM

ENTRE COM A CLAUSULA 2

> O RELE' NAO FAZ CLIQUE QDO O BOTAO DA BUZINA E' PRESSIONADO &

EXISTE SINONIMO PARA 'NAO FAZ'? (S/N)

> NAO

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA:

ATRIBUTO CLIQUE

OBJETO BUZINA

FUNCAO NAO FAZ

VALOR

UNIDADE

OK? (S/N) > SIM

ENTRE COM A CLAUSULA 3

> A LAMPADA DE TESTE ACENDE QUANDO UMA PONTA E' ATERRADA E A OUTRA E'

> CONECTADA AO TERMINAL DO FIO PRETO DO RELE' ENQUANTO O BOTAO DA

> BUZINA E' PRESSIONADO

>

QUAL O ATRIBUTO?: > PRETO

PALAVRAS CONHECIDAS:

- 5 - DIAGNOSTICO
- 6 - SINTOMA
- 8 - CLIQUE
- 10 - VERDE
- 12 - FIO-VERDE
- 14 - FIO-PRETO

PRETO E' SINONIMO DE ALGUMA PALAVRA CONHECIDA?

> NAO

COMO ESTE ATRIBUTO PODE SER CLASSIFICADO?

- 0) ATRIBUTO SIM/NAO (VERDADEIRO OU FALSO)
- 1) ATRIBUTO DE VALOR NUMERICO (COM UNIDADE)
- 2) ATRIBUTO QUE REQUER VALOR SEM UNIDADE
- 3) ATRIBUTO DO TIPO MENSAGEM

> 0

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA:

ATRIBUTO PRETO
OBJETO BUZINA
FUNCAO ACENDE
VALOR
UNIDADE

ENTRE COM A CONCLUSAO 1

> O DIAGNOSTICO PARA A BUZINA E' TROCAR O RELE'
>

NUMA ESCALA DE 1 A 10, QUAL A CERTEZA DEPOSITADA NESTA CONCLUSAO?

> 10

ELEMENTOS IDENTIFICADOS NESTA CLAUSULA

ATRIBUTO DIAGNOSTICO
OBJETO BUZINA
FUNCAO MOSTRAR
VALOR
UNIDADE

OK? (S/N) > SIM

=====

REGRA 4

SE : 1) A BUZINA NAO FUNCIONA E' UM SINTOMA DA BUZINA, &
2) O RELE' NAO FAZ CLIQUE QDO O BOTAO DA BUZINA E' PRESSIONADO, &
3) A LAMPADA DE TESTE ACENDE QUANDO UMA PONTA E' ATERRADA E A
OUTRA E' CONECTADA AO TERMINAL DO FIO PRETO DO RELE' ENQUANTO
O BOTAO DA BUZINA E' PRESSIONADO.

ENTAO : 1) COM CERTEZA (1.0) O DIAGNOSTICO PARA A BUZINA E' TROCAR O
RELE'.

=====

** PROPRIEDADES DE PRETO

O VALOR DO ATRIBUTO DEVE SER PERGUNTADO (P) OU DEDUZIDO (D) ?

> P

ENTRE COM A PERGUNTA A SER FEITA AO USUARIO (MAX = 90 CARAC)

> A LAMPADA DE TESTE ACENDE QUANDO SE APERTA A BUZINA COM UMA PONTA
> NO FIO PRETO E A OUTRA NO FIO TERRA?

>

** QUAL A TRADUCAO DESTE ATRIBUTO (MAX = 24 CARAC)

>

** PROPRIEDADES DO ATRIBUTO PRETO:

ESPERADO = SIM NAO S N

PERGUNTA = A LAMPADA ACENDE QUANDO SE APERTA A BUZINA COM UMA PONTA NO
FIO PRETO E A OUTRA NO FIO TERRA?

PD = 1 TRADUCAO =

TUDO OK? (S/N) > SIM

REGRA 4 ADQUIRIDA. DESEJA VER A REGRA?

> NAO

CODIGO DE INICIALIZACAO: > REFERENCIAS

REFERENCIAS AO ATRIBUTO: > DIAGNOSTICO

=====

ATRIBUTO DIAGNOSTICO NA PREMISSE DE:

=====

ATRIBUTO DIAGNOSTICO NA CONCLUSAO DE:

REGRA 2 REGRA 4

=====

C.3 - EXEMPLO DE CONSULTA

1) A BUZINA ESTA'FUNCIONANDO?

> SIM

2) HA' UMA ABERTURA ENTRE O FIO VERDE DO RELE'E O FIO TERRA?

> REGRA

=====

REGRA 5

SE : 1) HA' UMA ABERTURA ENTRE O FIO-VERDE DO RELE'E O FIO TERRA, &
2) A LAMPADA DE TESTE ACENDE QDO UMA PONTA E' ATERRADA E A
OUTRA CONECTADA AO FIO NO CONECTOR ENQTO O BOTAO DA BUZINA
E' PRESSIONADO.

ENTAO : 1) COM CERTEZA (1.0) HA' UMA ABERTURA ENTRE O FIO-NO-CONECTOR
NA BUZINA E O FIO TERRA.

=====

> SIM

3) A LAMPADA ACENDE QDO SE APERTA A BUZINA COM UMA PONTA NO FIO NO
CONECTOR E A OUTRA NO FIO TERRA?

> PORQUE

ESTOU TENTANDO DEDUZIR O ATRIBUTO FIO-NO-CONECTOR NA REGRA 5

JA' SEI QUE:

HA' UMA ABERTURA ENTRE O FIO-VERDE DO RELE E O FIO TERRA

ASSIM SENDO, SE :

1) A LAMPADA DE TESTE ACENDE QDO UMA PONTA E' ATERRADA E A OUTRA
CONECTADA AO FIO NO CONECTOR ENQTO O BOTAO DA BUZINA E'PRESSIONADO

ENTAO :

COM CERTEZA (1.0) HA' UMA ABERTURA ENTRE O FIO-NO-CONECTOR NA BUZINA E FIO TERRA (REGRA 5).

> PORQUE

ESTOU TENTANDO DEDUZIR O ATRIBUTO DIAGNOSTICO NA REGRA 6

SE :

1) HA' UMA ABERTURA ENTRE O FIO-NO-CONECTOR E O FIO TERRA &

2) NAO HA' UM BOM-TERRA NA BUZINA

ENTAO:

UM DIAGNOSTICO E' CONSERTAR O FIO TERRA NA BUZINA (REGRA 6)

FIM DE ENCADEAMENTO DE 'PORQUE'. A PROXIMA CHAMADA DO COMANDO REINICIARA'O ENCADEAMENTO.

> RESPOSTAS

** VALORES ESPERADOS PARA O ATRIBUTO CONECTOR:

1. SIM

2. NAO

3. S

4. N

> PERGUNTA

3) A LAMPADA ACENDE QDO SE APERTA A BUZINA COM UMA PONTA NO FIO NO CONECTOR E A OUTRA NO FIO TERRA?

> NAO

4) HA' UMA ABERTURA ENTRE O FIO NO CONECTOR E O FIO TERRA?

> NAO

5) HA' UMA ABERTURA ENTRE O FIO PRETO DO RELE E O FIO TERRA?

> SIM

6) A LAMPADA ACENDE QUANDO SE APERTA A BUZINA COM UMA PONTA NO FIO PRETO DO TURN SIGNAL CONECTOR E A OUTRA NO FIO TERRA?

> SIM

7) A LAMPADA ACENDE QDO A BUZINA E' APERTADA COM UMA PONTA ATERRADA E A OUTRA NO CONTACT ASSEMBLY?

> NAO

8) HA'UMA ABERTURA ENTRE O CONTACT ASSEMBLY DA BUZINA E O FIO TERRA?

> SIM

9) A CORREIA DO FIO TERRA ESTA' QUEBRADA?

> NAO

CONCLUSOES :

O DIAGNOSTICO E' CONSERTAR A ABERTURA NO FIO ENTRE O RELE E A BUZINA (FATOR = 1.).

O DIAGNOSTICO E' CONSERTAR A ABERTURA NO FIO ENTRE O TURN SIGNAL CONECTOR E O FIO TERRA (FATOR = 1.).

O DIAGNOSTICO E' CONSERTAR O FIO TERRA NOS FIOS DOS CONTATOS DA BUZINA ATE'OS BOTOES (FATOR = 1.).

APÊNDICE D

SISTEMA DE PREVISÃO DE PRECIPITAÇÃO PLUVIOMÉTRICA

D.1 - COMENTÁRIOS SOBRE A AQUISIÇÃO DO BANCO DE CONHECIMENTO

A primeira versão das regras do banco de conhecimento adquiridas junto à especialista não levava em conta a representação adotada pelo sistema. Posteriormente, os elementos básicos foram extraídos e as regras adaptadas até chegarem ao presente estado. A montagem deste pequeno banco já demonstrou a necessidade da figura do engenheiro de conhecimento.

As duas condições principais na previsão pluviométrica, a saber, as condições locais e as condições de grande escala, não foram utilizadas nas mesmas regras para evitar inconsistência e para que a certeza na previsão fosse a combinação dos fatores de certeza destas condições e não o mínimo deles.

Diferentemente do sistema da buzina, onde entidades foram encaradas como atributos de um só objeto, este sistema possui uma hierarquia de 6 objetos, o que lhe possibilita aumentar sua eficiência à medida que estes objetos são quantificados.

D.2 - REGRAS

REGRA 1

- SE : 1) Há uma zona frontal no Paraná, &
2) A velocidade da zona frontal é de 10 km por hora, &
3) A zona frontal está ativa, &
4) A zona frontal está desenvolvendo-se continuamente, &
5) As condições em altos níveis são favoráveis.
- ENTÃO : 1) Existe forte evidência (0.8) de que as condições de grande escala em 24 horas são favoráveis.

REGRA 2

- SE : 1) Hã uma zona frontal no Rio Grande do Sul, &
2) A velocidade da zona frontal é de 20 km por hora, &
3) A zona frontal está desenvolvendo-se continuamente, &
4) A zona frontal está ativa, &
5) As condições em altos nı́veis são favoráveis.

ENTÃO : 1) Existe evidência (0.6) de que as condições de grande escala em 24 horas são favoráveis.

REGRA 3

- SE : 1) Hã uma zona frontal no Rio Grande do Sul, &
2) A velocidade da zona frontal é de 20 km por hora, &
3) (3.1 A intensidade da zona frontal está diminuindo continuamente, ou
3.2 A velocidade da zona frontal está diminuindo) &
4) As condições em altos nı́veis são favoráveis.

ENTÃO : 1) Existe fraca evidência (0.2) de que as condições de grande escala em 24 horas são favoráveis.

REGRA 4

- SE : 1) Hã um sistema sinótico se aproximando, &
2) O sistema sinótico tem pouca intensidade, &
3) Hã um jato em altos nı́veis, &
4) Hã convergência de umidade em baixos nı́veis

ENTÃO : 1) Existe evidência (0.7) de que as condições de grande escala em 24 horas são favoráveis.

REGRA 5

- SE : 1) Hã um sistema no sul do Brasil, &
2) Hã um bloqueio no escoamento, &
3) As condições em altos nı́veis não são favoráveis.

ENTÃO : 1) Existe forte evidência (0.9) de que as condições de grande escala em 24 horas não são favoráveis.

REGRA 6

- SE : 1) (1.1 Hã um sistema no Paraná, ou1
1.2 Hã um sistema em Santa Catarina, ou1

1.3 Hã um sistema no Rio Grande do Sul)

ENTÃO : 1) Com certeza (1.0) hã um sistema no sul do Brasil.

REGRA 7

SE : 1) As condições de pressão nas últimas 24 horas são favoráveis, &

2) A depressão do bulbo úmido $\bar{e} < 3$ graus.

ENTÃO : 1) Existe forte evidência (0.8) de que as condições locais em 24 horas são favoráveis .

REGRA 8

SE : 1) As condições de pressão nas últimas 24 horas são favoráveis , &

2) A depressão do bulbo úmido $\bar{e} > 3$ graus.

ENTÃO : 1) Existe evidência (0.6) de que as condições locais em 24 horas são favoráveis.

REGRA 9

SE : 1) As condições de pressão nas últimas 24 horas não são favoráveis, &

2) A depressão do bulbo úmido $\bar{e} < 3$ graus.

ENTÃO : 1) Existe evidência (0.4) de que as condições locais em 24 horas são favoráveis.

REGRA 10

SE : 1) As condições de pressão nas últimas 24 horas não são favoráveis, &

2) A depressão do bulbo úmido $\bar{e} > 3$ graus

ENTÃO : 1) Existe evidência (0.7) de que as condições locais em 24 horas não são favoráveis.

REGRA 11

SE : 1) As condições de pressão nas últimas 3 horas são favoráveis, &

2) A depressão do bulbo úmido $\bar{e} < 3$ graus

ENTÃO : 1) Existe forte evidência (0.8) de que as condições locais para menos de 24 horas são favoráveis.

REGRA 12

- SE : 1) As condições de pressão nas últimas 3 horas são favoráveis,
&
2) A depressão do bulbo úmido é > 3 graus.
- ENTÃO : 1) Existe evidência (0.4) de que as condições locais para menos de 24 horas são favoráveis.

REGRA 13

- SE : 1) As condições de pressão nas últimas 3 horas não são favoráveis, &
2) A depressão do bulbo úmido é < 3 graus.
- ENTÃO : 1) Existe evidência (0.4) de que as condições locais para menos de 24 horas são favoráveis.

REGRA 14

- SE : 1) As condições de pressão nas últimas 3 horas não são favoráveis, &
2) A depressão do bulbo úmido é > 3 graus.
- ENTÃO : 1) Existe evidência (0.7) de que as condições locais para menos de 24 horas não são favoráveis.

REGRA 15

- SE : 1) Há um cavado em altos níveis se aproximando, &
2) O cavado está se amplificando
- ENTÃO : 1) Existe forte evidência (0.8) de que as condições em altos níveis são favoráveis.

REGRA 16

- SE : 1) Há uma nuvem convectiva próxima à região, &
2) A nuvem convectiva está desenvolvendo-se na parte da manhã
- ENTÃO : 1) Existe evidência (0.5) de que as condições locais para menos de 24 horas são favoráveis.

REGRA 17

- SE : 1) Há um jato em altos níveis
- ENTÃO : 1) Existe evidência (0.7) de que as condições em altos níveis são favoráveis.

REGRA 18

SE : 1) Há um vórtice ciclônico em altos níveis se aproximando
ENTÃO : 1) Existe forte evidência (0.8) de que as condições em altos níveis são favoráveis.

REGRA 19

SE : 1) Há uma zona frontal em São Paulo, &
2) A zona está ativa
ENTÃO : 1) Existe forte evidência (0.9) de que as condições de grande escala em menos de 24 horas são favoráveis.

REGRA 20

SE : 1) Há uma zona frontal no Paraná, &
2) A zona está ativa, &
3) As condições em altos níveis são favoráveis.
ENTÃO : 1) Existe evidência (0.7) de que as condições de grande escala em menos de 24 horas são favoráveis.

REGRA 21

SE : 1) As condições locais para menos de 24 horas são favoráveis, &
2) Estamos no verão
ENTÃO : 1) Existe evidência (0.7) de que haverá precipitação em menos de 24 horas em São José dos Campos.

REGRA 22

SE : 1) (1.1 0 mês atual $\bar{e} < 3$, ou
1.2 0 mês atual $\bar{e} > 10$)
ENTÃO : 1) Com certeza (1.0) estamos no verão.

REGRA 23

SE : 1) As condições locais para menos de 24 horas são favoráveis, &
2) Não estamos no verão
ENTÃO : 1) Existe evidência (0.4) de que haverá precipitação em menos de 24 horas em São José dos Campos.

REGRA 24

SE : 1) As condições locais em 24 horas são favoráveis, &
2) Estamos no verão
ENTÃO : 1) Existe evidência (0.6) de que haverá precipitação em 24 horas em São José dos Campos.

REGRA 25

SE : 1) As condições locais em 24 horas são favoráveis, &
2) Não estamos no verão
ENTÃO : 1) Existe evidência (0.5) de que haverá precipitação em 24 horas em São José dos Campos.

REGRA 26

SE : 1) As condições de grande escala em menos de 24 horas são favoráveis, &
2) Estamos no verão
ENTÃO : 1) Existe forte evidência (0.9) de que haverá precipitação em menos de 24 horas em São José dos Campos

REGRA 27

SE : 1) As condições de grande escala em menos de 24 horas são favoráveis, &
2) Não estamos no verão
ENTÃO : 1) Existe forte evidência (0.8) de que haverá precipitação em menos de 24 horas em São José dos Campos.

REGRA 28

SE : 1) As condições de grande escala em 24 horas são favoráveis, &
2) Estamos no verão
ENTÃO : 1) Existe forte evidência (0.8) de que haverá precipitação em 24 horas em São José dos Campos.

REGRA 29

SE : 1) As condições de grande escala em 24 horas são favoráveis, &
2) Não estamos no verão
ENTÃO : 1) Existe evidência (0.7) de que haverá precipitação em 24 horas em São José dos Campos

REGRA 30

SE : 1) A hora atual $\bar{e} \geq 10$, &
A hora atual $\bar{e} \leq 16$

ENTÃO : 1) Com certeza (1.0) a pressão deveria estar diminuindo, &
2) Com certeza (1.0) a pressão não deveria estar aumentando.

REGRA 31

SE : 1) A hora atual $\bar{e} \geq 0$, &
2) A hora atual $\bar{e} \leq 4$

ENTÃO : 1) Com certeza (1.0) a pressão deveria estar diminuindo, &
2) Com certeza (1.0) a pressão não deveria estar aumentando.

REGRA 32

SE : 1) A hora atual $\bar{e} \geq 4$, &
A hora atual $\bar{e} \leq 10$

ENTÃO : 1) Com certeza (1.0) a pressão deveria estar aumentando, &
2) Com certeza (1.0) a pressão não deveria estar diminuindo.

REGRA 33

SE : 1) A hora atual $\bar{e} \geq 16$, &
2) A hora atual $\bar{e} \leq 24$

ENTÃO : 1) Com certeza (1.0) a pressão deveria estar aumentando, &
2) Com certeza (1.0) a pressão não deveria estar diminuindo.

REGRA 34

SE : 1) O valor na carta isalobárica da região onde se localiza São José dos Campos $\bar{e}$ menor que -6 MB, &
2) O valor na carta isalobárica da região onde se localiza São José dos Campos $\bar{e} \geq -10$ MB

ENTÃO : 1) Existe forte evidência (0.8) de que as condições de pressão nas últimas 24 horas são favoráveis.

REGRA 35

SE : 1) O valor na carta isalobárica da região onde se localiza São José dos Campos $\bar{e}$ menor que -2 MB, &
2) O valor na carta isalobárica da região onde se localiza São José dos Campos $\bar{e} \geq -6$ MB.

ENTÃO : 1) Existe evidência (0.5) de que as condições de pressão nas últimas 24 horas são favoráveis.

REGRA 36

SE : 1) O valor na carta isalobárica da região onde se localiza São José dos Campos é menor que 0, &
2) O valor na carta isalobárica da região onde se localiza São José dos Campos é ≥ -2 MB

ENTÃO : 1) Existe fraca evidência (0.2) de que as condições de pressão nas últimas 24 horas são favoráveis.

REGRA 37

SE : 1) A pressão está diminuindo, &
2) A pressão deveria estar diminuindo, &
3) A tendência da pressão é < -3 MB

ENTÃO : 1) Existe evidência (0.6) de que as condições de pressão nas últimas 3 horas são favoráveis.

REGRA 38

SE : 1) A pressão está aumentando, &
2) A pressão deveria estar aumentando, &
3) A tendência da pressão é > -3 MB

ENTÃO : 1) Existe evidência (0.6) de que as condições de pressão nas últimas 3 horas não são favoráveis.

REGRA 39

SE : 1) A pressão está estacionária, &
2) A pressão deveria estar diminuindo

ENTÃO : 1) Existe evidência (0.6) de que as condições de pressão nas últimas 3 horas não são favoráveis.

REGRA 40

SE : 1) A pressão está estacionária, &
2) A pressão deveria estar aumentando

ENTÃO : 1) Existe evidência (0.6) de que as condições de pressão nas últimas 3 horas são favoráveis.

REGRA 41

SE : 1) O mês atual é ≥ 3 , &
2) O mês atual é ≤ 10

ENTÃO : 1) Com certeza (1.0) não estamos no verão.

REGRA 42

SE : 1) As condições locais em 24 horas não são favoráveis, &
2) Estamos no verão

ENTÃO : 1) Existe evidência (0.5) de que não haverá precipitação em 24 horas em São José dos Campos.

REGRA 43

SE : 1) As condições locais em 24 horas não são favoráveis, &
2) Não estamos no verão

ENTÃO : 1) É definitivo (1.0) que não haverá precipitação em 24 horas em São José dos Campos.

REGRA 44

SE : 1) As condições de grande escala em 24 horas não são favoráveis,
&
2) Estamos no verão

ENTÃO : 1) Existe evidência (0.5) de que não haverá precipitação em 24 horas em São José dos Campos.

REGRA 45

SE : 1) As condições de grande escala em 24 horas não são favoráveis, &
2) Não estamos no verão

ENTÃO : 1) Existe evidência (0.7) de que não haverá precipitação em 24 horas em São José dos Campos.

REGRA 46

SE : 1) As condições locais para menos de 24 horas não são favoráveis, &
2) Estamos no verão

ENTÃO : 1) Existe forte evidência (0.8) de que não haverá precipitação em menos de 24 horas em São José dos Campos.

REGRA 47

SE : 1) As condições locais para menos de 24 horas não são favoráveis, &
2) Não estamos no verão

ENTÃO : 1) É definitivo (1.0) que não haverá precipitação em menos de 24 horas em São José dos Campos.

REGRA 48

SE : 1) As condições de grande escala em menos de 24 horas não são favoráveis, &
2) Estamos no verão

ENTÃO : 1) Existe evidência (0.7) de que não haverá precipitação em menos de 24 horas em São José dos Campos.

REGRA 49

SE : 1) As condições de grande escala em menos de 24 horas não são favoráveis, &
2) Não estamos no verão

ENTÃO : 1) Existe forte evidência (0.9) de que não haverá precipitação em menos de 24 horas em São José dos Campos.

APÊNDICE E

LISTAGEM DE PROGRAMAS

E.1 - CRIABANCO

```

100  BEGIN
200      FILE =EGRASDISK(KIND=DISK,      MAXRECSIZE=150,
300          BLOCKSIZE=900,
400          APEASIZE=12,
500          AREAS=1),
600
700      OBJETIVOS(KIND=DISK,      MAXRECSIZE=14,
800          BLOCKSIZE=420,
900          APEASIZE=30,
1000         AREAS=1),
1100
1200      PROPATRIBUTOS(KIND=DISK,      MAXRECSIZE=70,
1300          BLOCKSIZE=840,
1400          AREASIZE=12,
1500          AREAS=1),
1600
1700      TEXTODISK(KIND=DISK,      MAXRECSIZE=90,
1800          BLOCKSIZE=900,
1900          APEASIZE=10,
2000         AREAS=1),
2100
2200      CLASSEDISK(KIND=DISK,      MAXRECSIZE=20,
2300          BLOCKSIZE=600,
2400          APEASIZE=30,
2500          AREAS=1),
2600
2700      ATRIBUTOSPEF(KIND=DISK,      MAXRECSIZE=100,
2800          BLOCKSIZE=1800,
2900          APEASIZE=10,
3000         AREAS=1),
3100
3200      ESQUELETODISK(KIND=DISK,      MAXRECSIZE=30,
3300          BLOCKSIZE=300,
3400          APEASIZE=10,
3500         AREAS=1),
3600
3700      TIPOCONTEXTODISK(KIND=DISK,      MAXRECSIZE=40,
3800          BLOCKSIZE=120,
3900          UNITS=WCRES,
4000          AREASIZE=30,
4100          AREAS=1),
4200
4300      AIRDISK(KIND=DISK,      MAXRECSIZE=10,
4400          BLOCKSIZE=720,
4500          APEASIZE=72,
4600          AREAS=1),
4700
4800      PUTRASPALAVRAS(KIND=DISK,      MAXRECSIZE=7,
4900          BLOCKSIZE=210,
5000          APEASIZE=30,
5100          AREAS=1),
5200
5300      VALLRCONHECTIDISK(KIND=DISK,      MAXRECSIZE=30,
          BLOCKSIZE=300,
          APEASIZE=10),

```

```

540:      OBJETOETIPOS(KIND=DISK,      MAXRECSIZE=4,
550:                      BLOCKSIZE=120,
560:                      APEASIZE=30,
570:                      AREAS=1),
580:      VIDEO(KIND=REMOTE,UNITS=CHARACTERS,MYUSE=10))
590:
600:      ARRAY REGRA(0:149),
610:             REFERENCIA(0:179),
620:             LISTADETIPOS(0:31),
630:             CLASS(0:19),
640:             OBJETIVO(1:14),
650:             PROPRIEDADE(0:69),
660:             NOMEEDOMINIO(0:9),
670:             ESQUELETO(0:29),
680:             PALAVRA(0:4),
690:             DIRETORIO(0:9),
700:             CONHECO(0:29),
710:             CHAVEVALOR(0:0),
720:             TEXTO(0:89),
730:             REGISTRO(0:39),
740:             REGANTERIOR(0:39);
750:
760:      INTEGER ASSOC,
770:             PROXREG,
780:             TIPOATRIBUTO;
790:
800:      STRING DLMINIO,
810:             NOMEAUX;
820:
830:      DEFINE
840:          MUDA(ARQUIVO,ENTREASPAS)=
850:              ARQUIVO.FLEXIBLE:=TRUE;
860:              REPLACE POINTER(NOMEEDOMINIO) BY (DOMINIO CAT ENTREASPAS CAT ".");
870:              REPLACE ARQUIVO.TITLE BY POINTER(NOMEEDOMINIO);
880:              IF ARQUIVO.PRESENT THEN CLOSE(ARQUIVO,PURGE) #;
890:
900:          LIMPA(GRAVA(CLASSE) =
910:              FILL DIRETORIO[*] WITH 10(0);
920:              DIRETORIO(0).[47:05]:=CLASSE;
930:              DIRETORIO(0).[42:10]:=PROXREG;
940:              DIRETORIO(0).[32:10]:=PROXREG;
950:              DIRETORIO(0).[22:10]:=ASSOC;
960:              DIRETORIO(0).[02:03]:=TIPOATRIBUTO;
970:              REPLACE POINTER(DIRETORIO(3)) BY NOMEAUX CAT "/";
980:              WRITE(DIRECT[PROXREG],10,DIRETORIO);
990:              PROXREG:=**+1 #;
1000:
1010:      COMMENT:
1020:          * PROGRAMA PRINCIPAL *
1030:
1040:          ESTE PROGRAMA CRIA OS ARQUIVOS QUE FORMARAO
1050:          O BANCO DE CONHECIMENTO DE UM SISTEMA ESPECIALISTA.
1060:          CASO O BANCO REFERENTE A ESTE SISTEMA JA EXISTA,
1070:          OS ARQUIVOS REFERENTES A ELE SAO LIMPOS ;
1080:
1090:
1100:      WRITE(VIDEO(STOP),<"** ENTRE COM O DOMINIO DE APLICACAO : ">);
1110:      READ(VIDEO,<A3>,DOMINIO);
1120:      DDMI TO:=HEAD( TAIL(DOMINIO, " "), NOT " ") CAT "/";
1130:
1140:      MUDA(PEGRASDISK,"REGRASDISK");
1150:      MUDA(TEXTODISK,"TEXTODISK");

```

```
1160: MUDA(ATRIBUTOSREF,"ATRIBUTOSREF");
1170: MUDA(ESQUELETONDISK,"ESQUELETONDISK");
1180: MUDA(DIRDISK,"DIRDISK");
1190: MUDA(CLASSEDISK,"CLASSEDISK");
1200: MUDA(OBJETIVOS,"OBJETIVOS");
1210: MUDA(PROPATRIBUTOS,"PROPATRIBUTOS");
1220: MUDA(OUTRASPALAVRAS,"OUTRASPALAVRAS");
1230: MUDA(TIPOSCONTEXTU,"TIPOSCONTEXTU");
1240: MUDA(VALORCONHECIDO,"VALORCONHECIDO");
1250: MUDA(OBJETISETIPOS,"OBJETISETIPOS");
1260:
1270: WRITE(REGRASDISK(0),150,REGRA);
1280: WRITE(TEXTDISK(0),90,TEXTO);
1290: WRITE(ATRIBUTOSREF(0),150,REFERENCIA);
1300: WRITE(ESQUELETONDISK(0),30,ESQUELETO);
1310: WRITE(OUTRASPALAVRAS(0),7,PALAVRA);
1320: WRITE(OBJETISETIPOS(0),4,LISTADETIPOS);
1330: WRITE(DIRDISK(0),10,DIRETORIO);
1340: WRITE(OBJETIVOS(0),14,OBJETIVO);
1350: WRITE(CLASSEDISK(0),20,CLASS);
1360: WRITE(VALORCONHECIDO(0),30,CONHECO);
1370: WRITE(PROPATRIBUTOS(0),70,PROPRIEDADE);
1380: WRITE(TIPOSCONTEXTU(0),40,REGISTPO);
1390:
1400:
1410: PRQXREG:=1;
1420: NOMEAUX:="IGUAL";
1430: LIMPAFGRAVA(3);
1440: NOMEAUX:="MOSTRA/R";
1450: LIMPAFGRAVA(3);
1460: NOMEAUX:="N/D";
1470: LIMPAFGRAVA(6);
1480: NOMEAUX:="PODE SER";
1490: LIMPAFGRAVA(3);
1500: NOMEAUX:="DEFINITIVO";
1510: LIMPAFGRAVA(3);
1520: NOMEAUX:="CONHECIDO";
1530: LIMPAFGRAVA(3);
1540: NOMEAUX:="CONFERMADO";
1550: LIMPAFGRAVA(3);
1560: NOMEAUX:="MOSTRA PREMISSA";
1570: LIMPAFGRAVA(3);
1580: NOMEAUX:="MAIOR";
1590: LIMPAFGRAVA(3);
1600: NOMEAUX:="MENOR";
1610: LIMPAFGRAVA(3);
1620: NOMEAUX:="MAIOR-IGUAL";
1630: LIMPAFGRAVA(3);
1640: NOMEAUX:="MENOR-IGUAL";
1650: LIMPAFGRAVA(3);
1660: NOMEAUX:="DIFERENTE";
1670: LIMPAFGRAVA(3);
1680: PRQXREG:=22;
1690: TIPOATRIBUTO:=1;
1700: ASSOC:=1;
1710: NOMEAUX:="HORA ATUAL";
1720: LIMPAFGRAVA(1);
1730: NOMEAUX:="DIA ATUAL";
1740: LIMPAFGRAVA(1);
```

```
1750: NOMEAUX:="MES ATUAL";
1760: LIMPAEGRVA(1);
1770:
1780: LOCK(PEGRASDISK);
1790: LOCK(TEXTODISK);
1800: LOCK(ATRIBUTOSREF);
1810: LOCK(FSQUELETOISK);
1820: LOCK(VALORCONHECTO);
1830: LOCK(DIRDISK);
1840: LOCK(OBJETIVOS);
1850: LOCK(CLASSEDISK);
1860: LOCK(PROPATRIBUTOS);
1870: LOCK(OUTRASPALAVRAS);
1880: LOCK(TIPLSCONTEXT);
1890: LOCK(OBJETOETIPES);
1900:
1910: FND.
```

E.2 - INICIALIZA

```
100 BEGIN
120 FILE VINFO(KIND=PEMOTIE,MYUSE=10,UNITS=CHARACTERS),
140     DTPDISK(KIND=DISK,FILETYPE=7),
160     TIPOSCONTEXTG(KIND=DISK,FILETYPE=7),
180     OBJETIPOS(KIND=DISK,FILETYPE=7),
200     PROPATRIBUTOS(KIND=DISK,FILETYPE=7),
220     CLASSDISK(KIND=DISK,FILETYPE=7),
240     ATRIBUTOSREF(KIND=DISK,FILETYPE=7),
260     OUTRASPALAVRAS(KIND=DISK,FILETYPE=7),
280     ESQUELETODISK(KIND=DISK,FILETYPE=7),
300     REGRASDISK(KIND=DISK,FILETYPE=7),
320     OBJETIVOS(KIND=DISK,FILETYPE=7),
340     TEXTODISK(KIND=DISK,FILETYPE=7),
360     VALORCONHECIDO(KIND=DISK,FILETYPE=7);
380
400 ARRAY REGRA[0:140],
420     REFERENCIA[0:179],
440     LISTADETIPOS[0:3],
460     CLASS[0:19],
480     OBJETIVO[1:14],
500     PROPRIEDADE[0:69],
520     NOMEEDOMINIO[0:9],
540     ESQUELETO[0:29],
560     PALAVRA[0:4],
580     TABELA[0:9],
600     DIRETORIO[0:19],
620     CONHEC[0:29],
640     CHAVEVALOR[0:0],
660     TEXTO[0:69],
680     REGISTRO[0:39],
700     REGANTERIOR[0:39];
720
740 INTEGER L,
760     PROXPEC,
780     ULTIMOTIPO,
800     ASSOC,
820     CLASSE,
840     CHAVEPEC,
860     SINDINIC,
880     ANTONINC,
900     TIPOCATIEUDO,
920     MODIFIC/,
940     SUBMULTIPLU;
960
980 REAL VALORSUE;
1000 STRING DOMINIO,
1020     NOMEAUX;
1040 BOOLEAN LSCREVA;
1060
1080 DEFINE
1100     CHAVELISTA=DIRETORIO[1].[09:10]#,
1120     LISTAL=LISTADETIPOS[0].[46:4]#,
1140     NULO=LISTADETIPOS[0].[47:1]#,
1160     TNULO=LISTADETIPOS[2].[47:1]#,
1180     CONTEXTO(X)=LISTADETIPOS[X DIV 5].[39-(X-(X DIV 5)*4 -1)*10:10]#,
1200     NROTIPL(X)=LISTADETIPOS[X DIV 5 +2].[39-(X-(X DIV 5)*4 -1)*10:10]#,
1220     GRAVANDICION=ARJO(CONDE) =
```

```

5700 BEGIN
5800 IF CLASSE = 2
5900 THEN BEGIN
6000 CHAVEREG:=OBJETOETIPOS*LASTRECORD + 1;
6100 ARRUMA LISTADETIPOS ZERANDO LTOTAL E NULO....
6200 LTOTAL:=0;
6300 NULO:=0;
6400 WRITE(OBJETOETIPOS[CHAVEREG],2,LISTADETIPOS);
6500 END
6600 ELSE CHAVEREG:=0;
6700 FILL DIRETORIO(*) WITH 10(" ");
6800 DIRETORIO[0].[47:05]:=CLASSE;
6900 DIRETORIO[0].[42:10]:=IF SINONIMO NEQ 0 THEN SINONIMO
7000 ELSE ONDE;
7100 DIRETORIO[0].[32:10]:=IF ANTONIMO NEQ 0 THEN ANTONIMO
7200 ELSE ONDE;
7300 DIRETORIO[0].[22:10]:=ASSEC;
7400 DIRETORIO[0].[02:03]:=TIPOATRIBUTO;
7500 DIRETORIO[0].[12:10]:=SURLTIPLD;
7600 DIRETORIO[1].[09:10]:=CHAVEREG;
7700 DIRETORIO[2]:=VALORSUB;
7800 REPLACE POINTER(DIRETORIO[3]) BY NOVEAUX CAT "/" ;
7900 WRITE(DIRDISK[ONDE],10,DIRETORIO);
8000 SINONIMO:=ANTONIMO:=ASSOC:=TIPOATRIBUTO:=0;
8100 CHAVEREG:=SURLTIPLD:=VALORSUB:=0;
8200 END#;
8300 MUDA(ARQUIVO,ENTREASPAS)=
8400 REPLACE POINTER(NOMEDEDOMINIO) BY (DOMINIO CAT ENTREASPAS CAT ".");
8500 REPLACE ARQUIVO.TITLE BY POINTER(NOMEDEDOMINIO);
8600 OPEN(ARQUIVO)#;
8700 PROCEDURE ERRO(NFPR);
8800 INTEGER NFPR;
8900 BEGIN
9000 COMMENT:
9100 * INICIO DE ERRO *
9200
9300 IMPRIME A MENSAGEM DE ERRO REFERENTE 'A NFPR ;
9400
9500 ESCREVA:=FALSE;
9600 CASE NFPR OF
9700 BEGIN
9800 1:WRITE(VIDEO,"** NAO SE ACEITA PALAVRA VAZIA">);
9900 2:WRITE(VIDEO,"** ESTA PALAVRA JA CONSTA EM DIRDISK">);
1000 3:WRITE(VIDEO,"** MODIFICACAO NAO EFETUADA-REGISTRO NAO ENCONTRADO">);
1010 4:WRITE(VIDEO,"** ESTA REGISTRO NAO PODE SER MODIFICADO">);
1020 5:WRITE(VIDEO,"** ESTE IDENTIFICADOR JA CONSTA EM DIRDISK COM OUTRA "
1030 "CLASSE">);
1040 6:WRITE(VIDEO,"** ESTA CONTEXTO JA ESTA ASSOCIADO A ESTE OBJETO EM "
1050 "OUTRO REGISTRO">);
1060 7:WRITE(VIDEO,"** O CONTEXTO NULO NAO PODE SER ASSOCIADO A ESTE "
1070 "OBJETO">);
1080 8:WRITE(VIDEO,"** ESTE CONTEXTO NAO PODE SER ASSOCIADO A ESTE OBJETO ",
1090 "/>"POIS ESTE ESTA ASSOCIADO AO CONTEXTO NULO">);
1100 END;
1110 END ERRO;
1120 PROCEDURE LIMPAERVA(X);
1130 INTEGER X;
1140 BEGIN
1150

```



```

11600
11700 COMMENT:
11800
11900      * INICIO DE LIMPAEGRVA *
12000
12100      GRAVA UM REGISTRO NO DICIONARIO ;
12200
12300      CLASSE:=X;
12400      SINONIMO:=ANTONIMO:=SUBMULTIPLO:=VALORSUB:=0;
12500      GRAVANDICIONARIO(PROXREG);
12600      PROXREG:=**+1;
12700 END LIMPAEGRVA;
12800 PROCEDURE TRATAIDENT;
12900 BEGIN
13000     BOOLEAN ACABOU;
13100     STRING PALAVRA,CARTAO,CADEIA;
13200
13300 COMMENT:
13400      * INICIO DE TRATAIDENT *
13500
13600      ADQUIRE OS IDENTIFICADORES QUE REPRESENTAM AS
13700      PALAVRAS-CHAVE DO DOMINIO ;
13800
13900      CADEIA:=EMPTY;
14000      ACABOU:=FALSE;
14100      WRITE(VIDEOSTOP,"<"> ">");
14200      READ(VIDEO,<A70>,CARTAO);
14300      CARTAO:=TAIL(CARTAO," ");
14400      IF CARTAO = EMPTY THEN NOMEAUX:=EMPTY
14500      ELSE
14600         IF TAKE(CARTAO,1) = "/" THEN BEGIN
14700             CAPITAL:=DPOP(CARTAO,1);
14800             WHILE NOT ACABOU DO BEGIN
14900                 PALAVRA:=HEAD(TAIL(CARTAO," "),NOT " ");
15000                 CARTAO:=TAIL(TAIL(CARTAO," "),NOT " ");
15100                 IF PALAVRA = EMPTY THEN BEGIN
15200                     WRITE(VIDEOSTOP,"<"> ">");
15300                     READ(VIDEO,<A70>,CARTAO);
15400                     END
15500                 ELSE IF DECP(PALAVRA,LENGTH(PALAVRA) - 1) = "/"
15600                     THEN BEGIN
15700                     ACABOU:=TRUE;
15800                     PALAVRA:=TAKE(PALAVRA,LENGTH(PALAVRA) - 1);
15900                     IF PALAVRA = EMPTY
16000                     THEN NOMEAUX:=TAKE(CADEIA,LENGTH(CADEIA) - 1)
16100                     ELSE NOMEAUX:=CADEIA CAT PALAVRA;
16200                     END
16300                 ELSE CADEIA:=CADEIA CAT PALAVRA CAT " ";
16400             END;
16500         END
16600     ELSE NOMEAUX:=HEAD(TAIL(CARTAO," "),NOT " ");
16700 END TRATAIDENT;
16800 BOOLEAN PROCEDURE PESQUISADIRETORIO;
16900 BEGIN
17000     BOOLEAN ACHEI;
17100
17200 COMMENT:
17300      * INICIO DE PESQUISADIRETORIO *
17400

```

```

17500 PESQUISA O DICCIONARIO USANDO COMO CHAVE O NOME DA
17600 PALAVRA. A VARIÁVEL "L" CONTEM O NUMERO DO REGISTRO ;
17700
17800 ACHEI:=FALSE;
17900 L:=1;
18000 DO BEGIN
18100   READ(DIRDISK[L],10,DIRETORIO);
18200   IF NOMEAUX = HEAD( STRING( POINTER(DIRETORIO[3]),42), NOT "/" )
18300   THEN ACHEI:=TRUE
18400   ELSE L:=L+1;
18500 END
18600 UNTIL ACHEI OR (L=DIRDISK*LASTRECORD + 1);
18700 PESQUISADIRETORIO:=ACHEI;
18800 END PESQUISADIRETORIO;
18900 PROCEDURE DICCIONARIO;
19000 BEGIN
19100   INTEGER
19200     AUX;
19300   BOOLEAN
19400     MACACABOU;
19500   STRING
19600     CODIGO;
19700
19800   DEFINE
19900     PEDNOVOREGISTRO=
20000     DO BEGIN
20100       WRITE(VIDEO,<"** NOVO REGISTRO"/>," C S A A S C T V.">);
20200       READ(VIDEO,<713,F7.2>>CLASSE,SYNONIMO,ANTONIMO,ASSOC,
20300         SUBMULTIPLO,CHAVEFREQ,TIPOATRIBUTO,VALORSUB);
20400       IF CLASSE NEO 0 THEN TRATAIDENT;
20500     END
20600     UNTIL LENGTH(NOMEAUX) <= 42;#;
20700
20800   COMMENT:
20900     * INICIO DE DICCIONARIO *
21000
21100   ADQUIRE,EXCLUI E MODIFICA PALAVRAS-CHAVE ;
21200
21300   WRITE(VIDEO,<"/>,"** ATUALIZACAO DE DIRDISK **">);
21400   MACACABOU:=TRUE;
21500   WHILE MACACABOU
21600   DO BEGIN
21700     WRITE(VIDEO(ISTOP),<"** ENTRE COM O MODO DA OPERACAO (A M L F) :>);
21800     READ(VIDEO,<A>>CODIGO);
21900     IF CODIGO="A"
22000     THEN BEGIN
22100       WRITE(VIDEO,<"** INSERCAO">);
22200       CLASSE:=1;
22300       WHILE CLASSE NEO 0
22400       DO BEGIN
22500         PEDNOVOREGISTRO;
22600         IF CLASSE NEO 0 THEN
22700         IF NOMEAUX = EMPTY
22800         THEN ERRO(1)
22900         ELSE IF PESQUISADIRETORIO
23000         THEN ERRO(2)
23100         ELSE BEGIN
23200           GRAVANO DICCIONARIO(PPOXREG);
23300           PROXREG:=#+1;

```

```

2340:         END;
2350:     END
2360: ELSE IF CODIGO="M"
2370: THEN BEGIN
2380:     WRITE(VIDEO,"** ENTRE COM O NUMERO DO REGISTRO ",
2390:         "QUE DESEJA MODIFICAR">);
2400:     READ(VIDEO,"MODIFICA");
2410:     IF MODIFICA<1 OR MODIFICA GEQ PROXREG
2420:     THEN ERRO(3)
2430:     ELSE IF MODIFICA < 25
2440:     THEN ERRO(4)
2450:     ELSE BEGIN
2460:         PEDENOVOREGISTRO;
2470:         IF CLASSE NEO 0 THEN
2480:             IF HEAD(NOMEAUX,NOT " ") = EMPTY
2490:             THEN ERRO(1)
2500:             ELSE IF PESQUISADIRETORIO AND MODIFICA NEO L
2510:             THEN ERRO(2)
2520:             ELSE GRAVANDICIONARIO(MODIFICA);
2530:         END;
2540:         MODIFICA:=0;
2550:     END
2560: ELSE IF CODIGO="L"
2570: THEN BEGIN
2580:     WRITE(VIDEO,"** PALAVRAS CONECTADAS:">);
2590:     " REG CLS SIN ANT ASS SUB CHV TAT VALOR ">);
2600:     AUX:=1;
2610:     WHILE AUX < PROXREG
2620:     DO BEGIN
2630:         READ(DIRETORIO[AUX,10,DIRETORIO]);
2640:         CLASSE:=DIRETORIO[0].L47:05;
2650:         SINONIMO:=DIRETORIO[0].L42:10;
2660:         ANTONIMO:=DIRETORIO[0].L32:10;
2670:         ASSOC:=DIRETORIO[0].L22:10;
2680:         SUMULTIPLC:=DIRETORIO[0].L12:10;
2690:         TIPOATRIBUTO:=DIRETORIO[0].L02:03;
2700:         CHAVEREG:=DIRETORIO[1].L09:10;
2710:         VALORSUB:=DIRETORIO[2];
2720:         NOMEAUX:=STRING(POINTER(DIRETORIO[3]).42);
2730:         IF CLASSE NEO 0 THEN
2740:             WRITE(VIDEO,<15,"":>715*FC.1,"NOME = ",A42>);
2750:             AUX,CLASSE,SINONIMO,ANTONIMO,ASSOC,
2760:             SUMULTIPLC,CHAVEREG,TIPOATRIBUTO,VALORSUB
2770:             ,NOMEAUX);
2780:             AUX:=++1;
2790:         END
2800:     END
2810: ELSE IF CODIGO="F"
2820: THEN NADACAROU:=FALSE;
2830: END;
2840: END DICCIONARIO;
2850: PROCEDURE CRIATIPSCONTEXTOS;
2860: BEGIN
2870:     ARRAY
2880:     ASSOCI[1:4];
2890:     STRING
2900:     PRIMEIPOND,
2910:     NOADICIONAL;
2920:

```

```

29300      TRADUZ,
29400      DADO,
29500      CALEIA,
29600      PROPOSICAO;
29700  INTEGER
29800      T,
29900      REG,
30000      MODG,
30100      NUMERO,
30200      CONJA,
30300      TAMANHO,
30400      SALVAASSOC,
30500      LISTATUAL;
30600  POINTER
30700      P;
30800  BOOLEAN
30900      ACHEI,
31000      NAOLETUDDO,
31100      CNTXTNULO,
31200      MODIFICANDO,
31300      ACABOU,
31400      CONTINUA;
31500  DEFINE
31600      OBJTOANT = REGANTERIOR[0].[47:10] #,
31700      OBJTO = REGISTRO[0].[47:10] #,
31800      ASSUMIR = REGISTRO[0].[37:11] #,
31900      NUMASSOCIADOS = REGISTRO[0].[36:3] #,
32000      ASSOCIADO(X) = REGISTRO[0].[37-(X-1)*8:8] #,
32100      ATRIBUTOFPINCIPAL(X) = REGISTRO[1].[47-(X-1)*10:10] #,
32200      LENG(X) = REGISTRO[2].[37-(X-1)*10:10] #,
32300      TIPLCNTXT=REGISTRO[2].[47:10]#,
32400      TIPOANT = REGANTERIOR[2].[47:10] #;
32500
32600  PROCEDURE ATUALIZAUNIDADE(PATRIBUTO);
32700  VALUE PATRIBUTO;
32800  INTEGER PATRIBUTO;
32900  BEGIN
33000      BOOLEAN ERROIDENTIFICADO;
33100      INTEGER PUNIDADE;
33200
33300  COMMENT:
33400      * INICIO DE ATUALIZAUNIDADE *
33500
33600      TOMA A UNIDADE-PADRAO REFERENTE AO ATRIBUTO, CASO
33700      ELA EXISTA ;
33800
33900      WRITE(VIDEO,<"** ESTE ATRIBUTO ESTA ASSOCIADO A ALGUMA "
34000          "UNIDADE ?">);
34100      TRATAIDENT;
34200      IF NOMEAUX = "SIM" OR NOMEAUX = "S" THEN DO BEGIN
34300          ERROIDENTIFICADO:=FALSE;
34400          WRITE(VIDEO,<"** QUAL A UNIDADE PRINCIPAL ASSOCIADA "
34500              "A ESTE ATRIBUTO?">);
34600          TRATAIDENT;
34700          IF NOMEAUX = EMPTY THEN ERROIDENTIFICADO:=TRUE
34800          ELSE BEGIN
34900              IF PESQUISADIRETORIO THEN PUNIDADE:=L
35000              ELSE BEGIN
35100                  ASSOC:=PATRIBUTO;

```

```

35200          PUNIDADE:=PROXREG;
35300          LIMPAEGRAVA(5);
35400          END;
35500          READ(DIRDISK(PATRIBUTO),10,DIRETORIO);
35600          DIRETORIO(10).L22:10:=PUNIDADE;
35700          WRITE(DIRDISK(PATRIBUTO),10,DIRETORIO);
35800          END;
35900      END
36000      UNTIL NOT ERROIDENTIFICADO;
36100      END ATUALIZAUNIDADE;
36200
36300  PROCEDURE LEDADO(X);
36400  INTEGER X;
36500  BEGIN
36600
36700      COMMENT:
36800          * INICIO DE LEDADO *
36900
37000      LE CADIAS REFERENTES 'A PROPRIEDADE "X" DE UM
37100      ATRIBUTO ;
37200
37300      CADEIA:=EMPTY;
37400      ACABOU:=FALSE;
37500      WHILE NOT ACABOU DO BEGIN
37600          WRITE(VIDEO(ISTOP),<"> ">);
37700          READ(VIDEO,<A70>,DADO);
37800          IF TAIL(DADO," ") = EMPTY THEN DADO:="" ELSE
37900              WHILE DPOP(DADO,LENGTH(DADO) - 1) = " " DO
38000                  DADO:=TAKE(DADO,LENGTH(DADO) - 1);
38100          IF X = 1 THEN BEGIN
38200              CADEIA:=DADO;
38300              ACABOU:=TRUE;
38400          END
38500          ELSE BEGIN
38600              CADEIA:=CADEIA CAT HEAD(DADO,NOT " ");
38700              IF TAIL(DADO,NOT " ") NEQ EMPTY
38800              THEN ACABOU:=TRUE;
38900          END;
39000      END;
39100      IF TAMANHO + LENGTH(CADEIA) > 222
39200      THEN CADEIA:=TAKE(CADEIA,222 - TAMANHO);
39300      REPLACE P/P BY CADEIA;
39400      LENG(X):=LENGTH(CADEIA);
39500      TAMANHO:=TAMANHO + LENG(X);
39600      END LEDADO;
39700
39800  PROCEDURE GRAVAREGISTRO;
39900  BEGIN
40000      ARRAY NUM(0:0);
40100      STRING NUMER;
40200      DEFINE PROCUPAEMOBJETOETIPOS(TIPO,CHAVE)=
40300          BEGIN
40400              READ(OBJETOETIPOS[CHAVE],2,LISTADETIPOS);
40500              ACHEI:=FALSE;
40600              CONTA:=1;
40700              WHILE NOT ACHEI AND CONTA<=LTOTAL
40800              DO IF CONTEXTD(CONTA)=TIPO
40900                  THEN ACHEI:=TRUE
41000                  ELSE CONTA:=++1;

```

```

41100      END#
41200      ATUALIZASSOC =
41300      COMMENT: O CAMPO ASSOC PARA ATRIBUTOS INDICA:
41400      ASSOC = 0 - ATRIBUTO F' DO TIPO SIM/NAO NA MENSAGEM
41500      ASSOC = 1 - ATRIBUTO REQUER UM VALOR (NUMERICO) &
41600      UNIDADE (1 SERA SUBSTITUIDO PELO ENDEREÇO
41700      DA UNIDADE PRINCIPAL, QUANDO ESSE FOR CN-
41800      NHECIDO)
41900      ASSOC = A - ATRIBUTO REQUER <VALOR> (SEM UNIDADE)
42000      (A = ENDEREÇO DO ATRIBUTO);
42100
42200      DO BEGIN
42300      WRITE(VIDEO,<"COMO ESSE ATRIBUTO PODE SER CLASSIFICADO?",
42400      //,"0) ATRIBUTO SIM/NAO (VERDADEIRO OU FALSO)",//,
42500      "1) ATRIBUTO TIPO UNICO",//,
42600      "2) ATRIBUTO TIPO MULTIPLO",//>);
42700
42800      ACHEI:=TRUE;
42900      READ(VIDEO,<A6>,<NUM>);
43000      NUMBER:=TAIL(HEAD(STRING(NUM,6),NOT " "),NOT " ");
43100      IF TAIL(NUMBER,"0123456789") = EMPTY
43200      THEN IF DECIMAL(NUMBER) > 2 THEN ACHEI:=FALSE
43300      ELSE TIPOATRIBUTO:=DECIMAL(NUMBER);
43400      ASSOC:=0;
43500      IF ACHEI AND DECIMAL(NUMBER) NEO 0 THEN BEGIN
43600      WRITE(VIDEO,<"** COMO OS VALORES PARA ESTE ATRIBUTO",
43700      " PODER SER CLASSIFICADOS ?",
43800      //,"0) TEXTO",//,
43900      "1) NUMERICO",//,
44000      "2) NAO NUMERICO">);
44100      READ(VIDEO,<A6>,<NUM>);
44200      NUMBER:=HEAD(TAIL(STRING(NUM,6)," "),NOT " ");
44300      IF TAIL(NUMBER,"0123456789") = EMPTY
44400      THEN IF DECIMAL(NUMBER) > 2 THEN ACHEI:=FALSE
44500      ELSE ASSOC:=DECIMAL(NUMBER)
44600      ELSE ACHEI:=FALSE;
44700      END;
44800      END
44900      UNTIL ACHEI;#;
45000
45100      COMMENT:
45200      * INICIO DE GRAVA REGISTRO *
45300
45400      GRAVA UM NOVO REGISTRO EM TIPOSCONTEXTO ;
45500
45600      CNTXTNULO:=FALSE;
45700      FSCREVA:=TRUE;
45800      FILL REGISTRO(*) WITH 40(0);
45900      WRITE(VIDEO,<//,"** TIPO DE CONTEXTO NO. ">J3>,<REG>);
46000      P:=POINTER(REGISTRO[3]);
46100      WRITE(VIDEO,<"OBJETO : ">);
46200      TRAJAIDENT;
46300      IF NOMEAUX = EMPTY
46400      THEN ESCREVA:=FALSE
46500      ELSE IF PESQUISADIRETORIO
46600      THEN IF DIRETORIO[0].[47:05] NEO 2
46700      THEN ERRO(5)
46800      ELSE OBJETO:=L
46900      ELSE BEGIN

```

```

4700:         OBJETO:=PROXREG;
4710:         LIMPAEGRAVA(2);
4720:     END;
4730: IF FSCREVA
4740: THEN BEGIN
4750:     LISTATUAL:=CHAVELISTA;
4760:     WRITE(VIDEO,"TIPO DE CONTEXTO : ">);
4770:     TRATAIDENT;
4780:     IF NOMFAUX NEQ EMPTY
4790:     THEN IF PESQUISADIRETORIO
4800:     THEN IF DIRETORIO[0].[47:05] NEQ 7
4810:     THEN ERRO(5)
4820:     ELSE BEGIN
4830:         TIPOCNTXT:=L;
4840:         PROCURAIMOBJETOETIPDS(L,LISTATUAL);
4850:         IF NOT (MODIFICANDO AND
4860:         OBJETO=OBJETOANT AND TIPOCNTXT=TIPOANT)
4870:         THEN IF ACHEI THEN ERRO(6)
4880:         ELSE IF NULO=1 THEN ERRO(8) ELSE
4890:         ELSE IF NULO = 1 THEN ERRO(6);
4900:     END
4910:     ELSE IF NULO=1 AND (NOT MODIFICANDO OR
4920:     NOT (OBJETO=OBJETOANT AND TIPOCNTXT=TIPOANT))
4930:     THEN ERRO(8)
4940:     ELSE BEGIN
4950:         TIPOCNTXT:=PROXREG;
4960:         LIMPAEGRAVA(7);
4970:     END
4980:     ELSE BEGIN
4990:         READ(OBJETOETIPDS[CHAVELISTA],2,LISTADETIPDS);
5000:         IF (LTOTAL>=1 AND NOT MODIFICANDO) OR
5010:         (MODIFICANDO AND (LTOTAL>1 OR OBJETO NEQ OBJETOANT))
5020:         THEN ERRO(7);
5030:         TIPOCNTXT:=0;
5040:         CNTXINULO:=TRUE;
5050:     END;
5060:     END;
5070: IF FSCREVA
5080: THEN BEGIN
5090:     TAMANH0:=0;
5100:     WRITE(VIDEO,"TRADUZ : ">);
5110:     LEADO(1);
5120:     WRITE(VIDEO[STOP],"ASSUMIR : ">);
5130:     READ(VIDEO,/,NUMERO);
5140:     ASSUMIR:=NUMERO;
5150:     WRITE(VIDEO,"PRIMEIRO DO (TERMINE STRINGS COM ')') : ">);
5160:     LEADO(2);
5170:     WRITE(VIDEO,"NOADICIONAL (TERMINE STRINGS COM ')') : ">);
5180:     LEADO(3);
5190:     DO BEGIN
5200:         WRITE(VIDEO[STOP],"NUMASSOCIADOS (MAX = 4) : ">);
5210:         READ(VIDEO,/,NUMERO);
5220:     END
5230:     UNTIL NUMERO<=4;
5240:     NUMASSOCIADOS:=NUMERO;
5250:     IF NUMERO>0
5260:     THEN BEGIN
5270:         WRITE(VIDEO[STOP],"ASSOCIADOS : ">);
5280:         READ(VIDEO,/,FOR I:=1 STEP 1 UNTIL NUMERO DO ASSOCI(I));

```

```

52900      END;
53000      FOR I:=1 STEP 1 UNTIL NUMERO DO ASSOCIADO(I):=ASSOCI(I);
53100      WRITE(VIDEU,"ATRIBUTOS PRINCIPAIS : ");
53200      I:=1;
53300      NADEUTUDO:=TRUE;
53400      WHILE NADEUTUDO AND I LEQ 4
53500      DO BEGIN
53600          TRATAIDENT;
53700          IF NOMEAUX NEO EMPTY
53800          THEN IF PESQUISA DIRETORIO
53900              THEN IF DIRETORIO[0].[47:05]=1
54000                  THEN BEGIN
54100                      ATRIBUTO PRINCIPAL(I):=L;
54200                      I:=I+1;
54300                  END
54400                  ELSE BEGIN
54500                      NADEUTUDO:=FALSE;
54600                      ERRO(5);
54700                  END
54800              ELSE BEGIN
54900                  ATRIBUTO PRINCIPAL(I):=PROXREG;
55000                  I:=I+1;
55100                  ATUALIZASSOC;
55200                  SALVAASSOC:=ASSOC;
55300                  LIMPAEGRAPA(1);
55400                  IF SALVAASSOC = 1 THEN ATUALIZA UNIDADE(PROXREG - 1);
55500              END
55600          ELSE NADEUTUDO:=FALSE;
55700      END;
55800      END;
55900      IF ESCRIVA
56000      THEN BEGIN
56100          WRITE(TIPOSCONTEXT[REG],40,REGISTRO);
56200          IF MODIFICANDO
56300          THEN BEGIN
56400              %LIMPA OBJETO ANTERIOR DE CONTEXTO ANTERIOR
56500              READ(DIRDISK(OBJETOANT),10,DIRETORIO);
56600              PROCUPAE MOJETO TIPOS(TIPOANT,CHAVELISTA);
56700              IF NULO=1 THEN NULO:=0;
56800              ELSE BEGIN
56900                  CONTEXTO(CONTA):=CONTEXTO(LTOTAL);
57000                  NRO TIPO(CONTA):=NRO TIPO(LTOTAL);
57100              END;
57200              LTOTAL:=LTOTAL-1;
57300              WRITE(OBJETO TIPOS[CHAVELISTA],4,LISTADE TIPOS);
57400          END;
57500          %COLoca TIPOCNXT EM CHAVELISTA
57600          READ(OBJETO TIPOS[LISTATUAL],4,LISTADE TIPOS);
57700          LTOTAL:=LTOTAL+1;
57800          IF CNXTNULO
57900          THEN BEGIN
58000              NULO:=1;
58100              TNULO:=REG;
58200          END
58300          ELSE BEGIN
58400              CONTEXTO(LTOTAL):=TIPOCNXT;
58500              NRO TIPO(LTOTAL):=REG;
58600          END;
58700          WRITE(OBJETO TIPOS[LISTATUAL],4,LISTADE TIPOS);

```



```
58800      END;
58900      END GRAVAREGISTRO;
59000
59100      COMMENT:
59200          * INICIO DE CRIATIPOSCONTEXTO *
59300
59400      OS TIPOS DE CONTEXTO SAO INCLUIDOS, EXCLUIDOS OU
59500      MODIFICADOS, SENDO CRIADAS PALAVRAS-CHAVE NO
59600      DICCIONARIO SEMPRE QUE NECESSARIO ;
59700
59800
59900      WRITE(VIDEO,<"/>,"** ATUALIZACAO DE TIPOSCONTEXTO">);
60000      CONTINUA:=TRUE;
60100      TIPOANT:=0;
60200      OBJETOANT:=0;
60300      WHILE CONTINUA
60400      DO BEGIN
60500          MODIFICANDO:=FALSE;
60600          WRITE(VIDEO[STOP],<"/>,"** MODO DE OPERACAO (M A L F): ">);
60700          READ(VIDEO,1,DADO);
60800
60900          IF DADO="M" THEN MODO:=2
61000          ELSE IF DADO="A" THEN MODO:=3
61100          ELSE IF DADO="L" THEN MODO:=4
61200          ELSE IF DADO="F" THEN MODO:=5
61300          ELSE MODO:=6;
61400
61500      CASE MODO OF BEGIN
61600      2: WRITE(VIDEO[STOP],<"/>,"** QUAL REGISTRO MODIFICAR? ">);
61700          READ(VIDEO,/,REG);
61800          IF REG<1 OR REG>ULTIMOTIPO
61900          THEN WRITE(VIDEO,<"/>,"** MODIFICACAO NAO EFETUADA ",
62000                  "(REGISTRO NAO ENCONTRADO)">);
62100          ELSE BEGIN
62200              MODIFICANDO:=TRUE;
62300              READ(TIPOSCONTEXTO[REG],40,REGANTE[REG]);
62400              GRAVAREGISTRO;
62500          END;
62600
62700      3: REG:=ULTIMOTIPO;
62800          DO BEGIN
62900              REG:=REG+1;
63000              GRAVAREGISTRO;
63100              IF ESCREVA
63200              THEN ULTIMOTIPO:=**+1
63300              END;
63400          UNTIL NOT ESCREVA;
63500
63600      4: FOR REG:=1 STEP 1 UNTIL ULTIMOTIPO DO BEGIN
63700          READ(TIPOSCONTEXTO[REG],40,REGISTRO);
63800          P:=POINTER(REGISTRO[3]);
63900          WRITE(VIDEO,<"/>,"TIPO DE CONTEXTO NO. ",J3,<"/>,REG);
64000          TRADUZ:=STRING(P,LENG(1));
64100          WRITE(VIDEO,<"/>,"TIPO : ",J4,<"/>," TRADUZ : ",A+,<"/>,TIPOCNTXT,
64200                  LENGTH(TRADUZ),TRADUZ);
64300          WRITE(VIDEO,<"/>,"OBJETO = ",J4,<"/>," (ASSUMIP=",I1,<"/>," ATRIBUTOS",
64400                  " PRINCIPAIS : ",A15,<"/>,"OBJETO,ASSUMIP,FOR I:=1 STEP 1
64500                  UNTIL 4 DO ATRIBUTOPRINCIPAL(I));
64600          PRIMEIRONO:=STRING(P + LENG(1),LENG(2));
```

```

64700 NOADICIONAL:=STRING(P +(LENG(1) +LENG(2)), LENG(3));
64800 WRITE(VIDEO,"ASSOCIADOS : ",*I4>>NUMASSOCIADOS,
64900 FOR I:=1 STEP 1 UNTIL NUMASSOCIADOS DO ASSOCIADO(I));
65000 WRITE(VIDEO,"PRIMEIRO :");
65100 WRITE(VIDEO,"PRIMEIRO");
65200 WRITE(VIDEO,"NOADICIONAL :");
65300 WRITE(VIDEO,"NOADICIONAL");
65400 END;
65500 WRITE(VIDEO,"72( "-" )");
65600
65700 5: CONTINUA:=FALSE;
65800
65900 6:
66000
66100 END CASE;
66200 END;
66300 END;
66400
66500
66600 COMMENT:
66700 * PROGRAMA PRINCIPAL *
66800
66900 ESTE PROGRAMA ADQUIRE AS PALAVRAS-CHAVE DE UM DADO
67000 SISTEMA ESPECIALISTA. NESTA FASE, OS ARQUIVOS
67100 DICCIONARIO E TIPOSCONTEXTO PODEM SER MODIFICADOS
67200 PELO ESPECIALISTA ;
67300
67400
67500 WRITE(VIDEO[STOP],"** ENTRE COM O DOMINIO DE APLICACAO : ">);
67600 READ(VIDEO,"A3">>DOMINTO);
67700 DOMINIO:=HEAD( TAIL(DOMINIO, " "), NOT " ") CAT "/";
67800
67900 MUDA(REFGRASDISK,"REGRASDISK");
68000 MUDA(TEXTODISK,"TEXTODISK");
68100 MUDA(ATRIBUTOSPEF,"ATRIBUTOSPEF");
68200 MUDA(ESQUELETODISK,"ESQUELETODISK");
68300 MUDA(DIFDISK,"DIFDISK");
68400 MUDA(CLASESDISK,"CLASESDISK");
68500 MUDA(OBJETIVOS,"OBJETIVOS");
68600 MUDA(PROPATRIBUTOS,"PROPATRIBUTOS");
68700 MUDA(OUTRASPALAVRAS,"OUTRASPALAVRAS");
68800 MUDA(TIPOSCONTEXTO,"TIPOSCONTEXTO");
68900 MUDA(VALORCONHECIDO,"VALORCONHECIDO");
69000 MUDA(OBJETOETIPOS,"OBJETOETIPOS");
69100
69200 READ(DIFDISK[0],10,DIRETORIO);
69300 READ(OBJETOETIPOS[0],4,LISTADETIPOS);
69400 READ(TIPOSCONTEXTOL[0],40,REGISTRO);
69500
69600 PROXPG:=DIFDISK.LASTRECORD + 1;
69700 ULTIMOTIPO:=TIPOSCONTEXTOL.LASTRECORD;
69800
69900 COMMENT:
70000 VERIFICA SE AQUISIÇÃO JA FOI USADO ATRAVES DA MARCA
70100 EM OBJETIVOS. NESTE CASO, INICIALIZA NAO PODERA SER
70200 UTILIZADO ;
70300 READ(OBJETIVOS[0],14,OBJETIVO);
70400 IF OBJETIVO[1] NEQ 999
70500 THEN BEGIN

```

```
70601          DICONAPIO;
70701          CHATIPCECONTEXTU;
70801          END
70901      ELSE WRITE(VIDEO,<///,"ESTE PROGRAMA NAO PODE SER UTILIZADO "
71001                  "PARA ESTE DOMINIO"///," JA QUE AQUISICAO "
71101                  "FOI USADO">);
71201      END.
```

E.3 - AQUISIÇÃO

```

100 BEGIN
200 FILE VIDEO(KIND=REMOTE,MYUSE=10);
300 ARRAY
400     VFTORTIPOS[0:10],           % TIPOS DE CONTEXTO DE UMA CLAUSULA
500     DESATIVADA[0:149],         % FILHA DE REGRAS DESATIVADAS
600     SEMANTONIMO[0:49],         % FUNCOES QUE NAO TEM ANTONIMO CONECTADO
700     NMEDOMINIO[0:9],          % DOMINIO DE APLICACAO
800     INTEGER ARRAY LISTA[1:100];
900     STRING
1000     LINHA,
1100     CARTAO,
1200     PALAVRA,
1300     DOMINIO,
1400     CLAUSULA,
1500     LISTADEPALAVRAS;
1600     TRUTHSET DIGITOS("0123456789");
1700     BOOLEAN
1800         MENU,                  % INICIALIZACAO POR MENU
1900         ENTAO,                 % ANALISANDO A PARTE DE ACAN DA REGRA
2000         PREMISSE,             % ANALISANDO A PREMISSE DA REGRA
2100         FIMSESSAO,            %
2200         COMBARRA,             % INDICA SE O IDENTIFICADOR ESTA ENTRE BARRAS
2300         ALTERANDO,           % REGRA SENDO MODIFICADA
2400         ENTRODECU,            % INDICA INICIO DETETADO DE CLAUSULA OU
2500         ENTRODECU,            % INDICA INICIO DETETADO DE CLAUSULA OU
2600         REGRAESPECIAL,        %
2700         EPROIDENTIFICADO;     %
2800     INTEGER
2900         L,                     % USADO COMO ENDEPEC DE REGISTROS
3000         AVAIL,                % ESPACO DISPONIVEL EM REGRA
3100         TOTAL,                % NUMERO DE ELEMENTOS EM "LISTA"
3200         DELTH,                 % CLASSE DE DELIMITADOR
3300         CODIGO,                %
3400         NCLAUSULA,            %
3500         PERGUNTAS,             % NUMERO DE PERGUNTAS DE ANTONIMOS NUMA SESSAO
3600         INICIODECU,            % ENDEPECO DO "HEAD" DE UMA CLAUSULA OU
3700         NCLUBCLAUSULA,        %
3800         NUMERODAREGRA;         %
3900     REAL
4000     CFATOR;
4100 ----- DICCIONARIO DE PALAVRAS CONHECIDAS -----
4200
4300 FILE DISK(KIND=DISK,FILETYPE=7);
4400 COMMENT:-----
4500         MAXRECSIZE ..... 10
4600         BLOCKSIZE ..... 720
4700         AREASIZE ..... 72;
4800
4900 ARRAY DIRETORIO[0:9];
5000 DEFINE
5100     CLASSE = DIRETORIO[0].[47:05] #,
5200     SINONIMO = DIRETORIO[0].[42:10] #,
5300     ANTONIMO = DIRETORIO[0].[32:10] #,
5400     ASSOC = DIRETORIO[0].[22:10] #,
5500     SIMULTIFLO = DIRETORIO[0].[12:10] #,
5600     TIPDTRIBUTO = DIRETORIO[0].[02:03] #,

```

```

5700          CHAVEPEG      = DIRETORIO[1].[00:10] #,
5800          VALDRSUR      = DIRETORIO[2] #,
5900          PNOME         = POINTER(DIRETORIO[3]) #,
6000          NOME          = HEAD(STRING(PNOME,42),NCT "/" ) #;
6100
6200  ----- IMPORTANTE -----
6300
6400          COMMENT: O DICCIONARIO DEVERA CONTER INICIALMENTE (PELO MENOS) :
6500                  FUNCOES DE PREDICADO          1 = IGUAL
6600                  -----
6700                  E TODOS OS MODIFICADORES;
6800          DEFINE IGUAL     = 1 #,
6900                  MOSTRAR = 2 #;
7000
7100  ----- OBJETIVOS DE CONSULTA -----
7200
7300          FILE OBJETIVOS(KIND=DISK,FILETYPE=7);
7400          COMMENT:-----
7500                  MAXRECSIZE..... 14
7600                  BLOCKSIZE..... 420
7700                  AREASIZE..... 30;
7800
7900          BBOOLEAN ACARDU;
8000          ARRAY OBJETIVO(1:14);
8100
8200  ----- TEXTOS NAS REGRAS -----
8300
8400          FILE TEXTODISK(KIND=DISK,FILETYPE=7);
8500          COMMENT:-----
8600                  MAXRECSIZE ..... 90
8700                  BLOCKSIZE ..... 900
8800                  AREASIZE ..... 10;
8900
9000          BBOOLEAN TEXTO;
9100          INTEGER ETXTOS;
9200          STRING TEXTUAL;
9300
9400  ----- NOME DOS ELEMENTOS QUE CONSTITUEM UMA CLAUSULA -----
9500
9600          STRING ARRAY ELEMENTO(1:7);
9700          DEFINE
9800              ATRIBUTO      = 1 #,
9900              OBJETO        = 2 #,
10000             FUNCAO        = 3 #,
10100             VALOR         = 4 #,
10200             UNIDADE        = 5 #,
10300             NOMEATRIBUTO    = ELEMENTO[1] #,
10400             NOMEOBJETO      = ELEMENTO[2] #,
10500             NOMEFUNCAO      = ELEMENTO[3] #,
10600             NOMEVALOR       = ELEMENTO[4] #,
10700             NOMEUNIDADE     = ELEMENTO[5] #,
10800             NOMEMODIFICADOR = ELEMENTO[6] #,
10900             NOMECONTEXTUS   = ELEMENTO[7] #;
11000
11100  ----- ESTRUTURA INTERNA DE UMA REGRA -----
11200
11300          FILE REGPASCTSK(KIND=DISK,FILETYPE=7);
11400          COMMENT:-----
11500                  MAXRECSIZE ..... 150

```

```

11600                                BLOCKSIZE ..... 900
11700                                AREASIZE ..... 128
11800
11900      ARRAY REGRA[0:149];
12000      DEFINE
12100          ELEMENTOS(X) = REGRA[X].[44:07] #
12200          EESPECIAL    = RFGRA[0].[33:01] #
12300          EMOSTRAREGRA = REGRA[0].[32:01] #
12400          EPREMISSA    = RFGRA[0].[31:07] #
12500          ENTAD        = RFGRA[0].[24:07] #
12600          EULT         = RFGRA[0].[10:07] #
12700          MARCA(X)     = REGRA[X].[47:02] #
12800          OU1(X)        = RFGRA[X].[45:01] #
12900          PROXC(X)      = RFGRA[X].[05:06] #
13000          PROXL(X)     = RFGRA[X].[13:07] #
13100          EATRIBUTO    = REGRA[I].[45:10] #
13200          ELBJETO       = RFGRA[I].[35:10] #
13300          EFUNCAO       = RFGRA[I].[25:10] #
13400          EUNIDADEF     = REGRA[I].[15:10] #
13500          EIIPGVALOR    = RFGRA[I+1].[05:02] #
13600          ESQUELETO     = RFGRA[I+1].[37:10] #
13700          EMODFUNCAO   = RFGRA[I+1].[27:10] #
13800          EREFPERGUNTA  = REGRA[I+1].[03:01] #
13900          EVALOP        = RFGRA[I+2] #
14000          EFATOP        = RFGRA[I+3] #
14100
14200      ----- CODIFICACAO DE CLAUSULA SIM
14300
14400      ARRAY CLAUSULACOD[0:2];
14500      DEFINE
14600          CATRIEUTL    = CLAUSULACOD[0].[45:10] #
14700          CLBJETO       = CLAUSULACOD[0].[35:10] #
14800          CFUNCAO       = CLAUSULACOD[0].[25:10] #
14900          CUNIDADEF     = CLAUSULACOD[0].[15:10] #
15000          CREPERGUNTA  = CLAUSULACOD[1].[03:01] #
15100          CIIPGVALOR    = CLAUSULACOD[1].[05:02] #
15200          CMODFUNCAO    = CLAUSULACOD[1].[27:10] #
15300          CESQUELETO    = CLAUSULACOD[1].[37:10] #
15400          CVALOP        = CLAUSULACOD[2] #
15500
15600      ----- DECODIFICACAO DE CLAUSULA
15700
15800      FILE ESQUELETOLISK(KIND=DISK,FILETYPE=7);
15900      COMMENT:-----
16000                          MAXPECSIZE ..... 30
16100                          BLOCKSIZE ..... 300
16200                          AREASTZE ..... 108
16300
16400      ARRAY ESQUELETL[0:29];
16500      STRING SKELETON;
16600
16700      ----- OUTRAS PALAVRAS CONHECIDAS
16800
16900      FILE OUTRASPALAVRAS(KIND=DISK,FILETYPE=7);
17000      COMMENT:-----
17100                          MAXRECSI7E ..... 7
17200                          BLUCKSIZE ..... 210
17300                          AREASIZE ..... 308
17400

```

```

17500 ***** REFERENCIAS A ATRIBUTOS *****
17600
17700 FILE ATRIBUTOSREF(KIND=DISK,FILETYPE=7);
17800 COMMENT:-----
17900 MAXRECSIZE ..... 180
18000 BLOCKSIZE ..... 1800
18100 AREASIZE ..... 10;
18200
18300 ARRAY REFERENCIAS[0:179];
18400 DEFINE
18500 LLINK(X) = REFERENCIAS[X].[47:07] #,
18600 TIPORULE(X) = REFERENCIAS[X].[40:01] #,
18700 NUMERODEGRA(X) = REFERENCIAS[X].[39:14] #,
18800 NEGACAO(X) = REFERENCIAS[X].[15:06] #,
18900 RLINK(X) = REFERENCIAS[X].[07:06] #;
19000
19100 ***** TIPOS DE CONTEXTO *****
19200
19300 FILE TIPOSCONTEXTO(KIND=DISK,FILETYPE=7);
19400 COMMENT:-----
19500 MAXPECSIZE ..... 40
19600 BLOCKSIZE ..... 120
19700 AREASIZE ..... 30;
19800
19900 ARRAY REGISTRO[0:39];
20000 DEFINE TOBJETO = REGISTRO[0].[47:10] #,
20100 TLENG(X) = REGISTRO[21].[37-(X-1)*10:10] #;
20200
20300 ***** LISTA DE TIPOS *****
20400
20500 FILE OBJETCETIPOS(KIND=DISK,FILETYPE=7);
20600 COMMENT:-----
20700 MAXRECSIZE ..... 4
20800 BLOCKSIZE ..... 120
20900 AREASIZE ..... 30;
21000
21100 ARRAY LISTADETIPOS[0:31];
21200 DEFINE CONTEXTO(X) = LISTADETIPOS[X DIV 5].[39-(X-(X DIV 5)*4 -1)
21300 *10:10] #,
21400 NROTIPO(X) = LISTADETIPOS[X DIV 5 +2].[39-
21500 (X-(X DIV 5)*4 -1)*10:10] #,
21600 NULL = LISTADETIPOS[0].[47:11] #,
21700 TNULL = LISTADETIPOS[21].[47:8] #,
21800 LTOTAL = LISTADETIPOS[01].[46:4] #;
21900
22000 ***** PROPIEDADES DOS ATRIBUTOS *****
22100
22200 FILE PROPATRIBUTOS(KIND=DISK,FILETYPE=7);
22300 COMMENT:-----
22400 MAXRECSIZE ..... 70
22500 BLOCKSIZE ..... 840
22600 AREASIZE ..... 12;
22700
22800 ARRAY PROPIEDADE[0:69];
22900
23000 ***** CLASSES DE PALAVRAS *****
23100
23200 FILE CLASSFOTSK(KIND=DISK,FILETYPE=7);
23300 COMMENT:-----

```

```

23400          MAXRECSIZE ..... 20
23500          BLOCKSIZE ..... 600
23600          AREASIZE ..... 300
23700
23800          ARRAY SCLASS[0:19];
23900
24000  *----- AUXILIAPES -----*
24100          OFFINE SIMNAD      = 1 #,
24200          NUMERICA          = 2 #,
24300          OPDEFMCDKRETA     = 3 #,
24400          VALORCEKTEZA      = 4 #,
24500          NAODONUMERO        = 5 #,
24600          LIMPATELA         = 12 #,
24700          BUZINA            = 47 #,
24800          LOOKUP            = CDEFIM:=IF PALAVRA = "&" THEN 0 ELSE
24900                                IF PALAVRA = "UU" THEN 1 ELSE
25000                                IF PALAVRA = "UU1" THEN 2 ELSE
25100                                IF PALAVRA = EMPTY THEN 3 ELSE 4 #,
25200          LERCARTAO          = READ(VIDEO,<A72>>,CAPTAO) #,
25300          EXISTEVALOR        = ( CVALOR NEO C OR CTIPOVALOR = 1 ) #,
25400          TIPOFEGKA(I)       = VETORTIPOS[0].[47-(I-1)*A:8] #,
25500          MUDA(ARGUIVO,ENTREASPAS)=
25600          REPLACE POINTER(NOMEDOMINIO)
25700          BY (DOMINIO CAT ENTREASPAS CAT ".");
25800          REPLACE ARQUIVO.TITLE BY POINTER(NOMEDOMINIO) ;
25900          OPEN(ARGUIVO) #;
26000
26100  *-----*
26200
26300  PROCEDURE SUBSTITUI(EMTEXTO,ULTIMOX,PORY);
26400  STRING EMTEXTO,ULTIMOX,PORY;
26500  BEGIN
26600
26700      COMMENT :
26800          * INICIO DE SUBSTITUI *
26900
27000      <SUBSTITUI TODAS AS INSTANCIAS DE "ULTIMOX" POR "PORY" NO TEXTO
27100      "EMTEXTO" >;
27200
27300      STRING S,P;
27400      S:=EMPTY;
27500      P:=HEAD(EMTEXTO,NOT " ");
27600      WHILE EMTEXTO NEQ EMPTY DO BEGIN
27700          S:=P CAT " " CAT S;
27800          EMTEXTO:=TAIL(TAIL(EMTEXTO,NOT " ")," ");
27900          P:=HEAD(EMTEXTO,NOT " ");
28000          END;
28100      P:=HEAD(S,NOT " ");
28200      WHILE (P NEQ ULTIMOX) AND (S NEQ EMPTY) DO BEGIN
28300          EMTEXTO:=P CAT " " CAT EMTEXTO;
28400          S:=TAIL(TAIL(S,NOT " ")," ");
28500          P:=HEAD(S,NOT " ");
28600          END;
28700      IF P = ULTIMOX THEN BEGIN
28800          S:=TAIL(PORY CAT TAIL(S,NOT " ")," ");
28900          WHILE S NEQ EMPTY DO BEGIN
29000              EMTEXTO:=HEAD(S,NOT " ") CAT " " CAT EMTEXTO;
29100              S:=TAIL(TAIL(S,NOT " ")," ");
29200          END;

```



```

29300      END;
29400      END;
29500
29600      -----
29700
29800      BOOLEAN PROCEDURE NUMERO(X);
29900      VALUE X;   STRING X;
30000      BEGIN
30100      COMMENT :
30200          * INICIO DE NUMERO *
30300
30400      DEVOLVE "TRUE" SE O STRING X FOR UM NUMERO ;
30500
30600      NUMERO:=FALSE;
30700      IF X NEQ EMPTY THEN BEGIN
30800          IF TAKE(X,1) = "-" THEN X:=DROP(X,1);
30900          X:=TAIL(X,DIGITOS);
31000          IF LENGTH(X) = 0 THEN NUMERO:=TRUE
31100          ELSE BEGIN
31200              IF TAKE(X,1) = "." THEN X:=TAIL(DROP(X,1),DIGITOS);
31300              IF LENGTH(X) = 0 THEN NUMERO:=TRUE;
31400          END;
31500      END;
31600      END NUMERO;
31700
31800      -----
31900
32000      PROCEDURE ATIVAREGRA;
32100      BEGIN
32200          DEFINE DISPO = DESATIVADA[0] #;
32300
32400      COMMENT :
32500          * INICIO DE ATIVAREGRA *
32600
32700      NUMERODAREGRA CONTERA A PROXIMA REGRA A SER ATIVADA ,QUE SERA O
32800      A REGRA QUE ESTA NO TOPO DA PILHA DE REGRAS DESATIVADAS OU O
32900      PROXIMO REGISTRO VAZIO DE REGRASDISK ;
33000
33100      READ(REGRASDISK[0],150,DESATIVADA);
33200      IF DISPO > 0 AND DISPO < 150 THEN BEGIN
33300          NUMERODAREGRA:=DESATIVADA[DISPO];
33400          DISPO:=DISPO+1;
33500      END
33600      ELSE NUMERODAREGRA:=REGRASDISK.LASTRECORD+1;
33700      WRITE(REGRASDISK[0],150,DESATIVADA);
33800      END;
33900
34000      -----
34100
34200      PROCEDURE IMPRIMELINHA;
34300      BEGIN
34400          INTEGER C;
34500
34600      COMMENT :
34700          * INICIO DE IMPRIMELINHA *
34800
34900      IMPRIME O STRING "LINHA" EM LINHAS DE 60 CARACTERES ;
35000
35100      C:=MIN(72,LENGTH(LINHA));

```

```

35200 WRITE(VIDEO,<A*>,C,TAKE(LINHA,C));
35300 LINHA:=DEOP(LINHA,C);
35400 WHILE LINHA NEQ EMPTY DO BEGIN
35500   C:=MIN(60,LENGTH(LINHA));
35600   WRITE(VIDEO,<X12,A*>,C,TAKE(LINHA,C));
35700   LINHA:=EROP(LINHA,C);
35800 END;
35900 END IMPRIMELINHA;

-----
36300 PROCEDURE ERPO(X);
36400 VALUE X; INTEGER X;
36500 BEGIN
36600 COMMENT :
36700 * INICIO DE ERRO *
36800
36900 IMPRIME A MENSAGEM DE ERRO CORRESPONDENTE A "X" , LIMPA A VARIÁVEL
37000 "CARTAO" E FAZ ERROIDENTIFICADO = TRUE ;
37100
37200 INTEGER I;
37300 STRING ARRAY NOMES[1:6];
37400 NOMES[1]:=" ATRIBUTO" ;
37500 NOMES[2]:=" DEJEITO" ;
37600 NOMES[3]:="A FUNCAO" ;
37700 NOMES[4]:=" VALOR" ;
37800 NOMES[5]:="A UNIDADE" ;
37900 NOMES[6]:=" MODIFICADOR DE FUNCAO" ;
38000
38100 CASE X OF BEGIN
38200 01:
38300   LINHA:="IDENTIFICADO MAIS DE UM" CAT NOMES[CLASSE] CAT
38400   " . CLAUSULA INVALIDA";
38500
38600 04:
38700   LINHA:="NÃO POSSO ACEITAR NOVAS FUNCOES. (SEMANTICA ?)";
38800
38900 05:
39000   LINHA:="IDENTIFICADO VALOR PARA UM ATRIBUTO TIPO SIM/NÃO";
39100
39200 06:
39300   LINHA:="UNIDADE INVALIDA PARA O ATRIBUTO " CAT NOMEATRIBUTO;
39400
39500 07:
39600   LINHA:="UNIDADE INCORRETA. CLAUSULA INVALIDA";
39700
39800 09:
39900   LINHA:="PALAVRA CAT " NÃO É UMA PALAVRA CONHECIDA";
40000
40100 10:
40200   LINHA:="FIM DE TEXTO (')') NÃO ENCONTRADO";
40300
40400 11:
40500   LINHA:=" 'OU' PODE OCORRER APENAS NA PREMISA";
40600
40700 12:
40800   LINHA:="ESSA PALAVRA JA É CONHECIDA COMO UM" CAT NOMES[CLASSE];
40900
41000 13:
41100   LINHA:="TEXTOS NÃO SÃO PERMITIDOS NA PREMISA";
41200
41300 14:
41400   LINHA:="CLAUSULA MUITO GRANDE (MAXIMO = 1170 CARACTERES)";
41500
41600 15:
41700   LINHA:="TEXTO MUITO GRANDE (MAXIMO = 540 CARACTERES)";
41800
41900 16:
42000   LINHA:="INICIALIZAÇÃO ERRADA. ESPERADO: INCLUIR/EXCLUIR/" CAT
42100   "ALTERAR/FIM";

```

```

4110: 17: LINHA:="'&' E 'OU' SAO PALAVRAS RESERVADAS";
4120:
4130: 18: LINHA:="PEGRA " CAT STRING(NUMERODAREGRA,*) CAT " NAO EXISTE";
4140:
4150: 19: IF CODIGO = 1 THEN LINHA:="A INCLUSAO " ELSE
4160: IF CODIGO = 2 THEN LINHA:="A EXCLUSAO " ELSE
4170: LINHA:="A ALTERACAO ";
4180: LINHA:=LINHA CAT "DA CLAUSULA " CAT STRING(NCLAUSULA,*)
4190: CAT " SUBCLAUSULA " CAT STRING(NSUBCLAUSULA,*) CAT
4200: " NA REGRA " CAT STRING(NUMERODAREGRA,*) CAT
4210: " NAO E' PERMITIDA";
4220:
4230: 20: IF CODIGO = 1 THEN LINHA:="A INCLUSAO " ELSE
4240: IF CODIGO = 2 THEN LINHA:="A EXCLUSAO " ELSE
4250: LINHA:="A ALTERACAO ";
4260: LINHA:=LINHA CAT "DA CLAUSULA " CAT STRING(NCLAUSULA,*)
4270: IF PREMISA THEN LINHA:=LINHA CAT " NA PREMISA" ELSE
4280: LINHA:=LINHA CAT " NA CONCLUSAO";
4290: LINHA:=LINHA CAT " DA REGRA " CAT STRING(NUMERODAREGRA,*) CAT
4300: " NAO E' PERMITIDA";
4310:
4320: 21: IF ALTERANDO THEN
4330: LINHA:="A INCLUSAO NAO SERA REALIZADA. NAO EXISTE ESPACO "
4340: CAT "NA REGRA " CAT STRING(NUMERODAREGRA,*) CAT
4350: " SUFICIENTE PARA A NOVA CLAUSULA"
4360: ELSE LINHA:="O ESPACO DE ARMAZENAMENTO FOI EXCEDIDO. " CAT
4370: "PEGRA NAO ADQUIRIDA";
4380:
4390: 22: LINHA:="INICIALIZACAO ERRADA. ESPERADO: CLAUSULA";
4400:
4410: 23: LINHA:="INICIALIZACAO ERRADA. ESPERADO: NUMERO DA CLAUSULA";
4420:
4430: 24: LINHA:="INICIALIZACAO ERRADA. ESPERADO: NUMERO DA SUBCLAUSULA";
4440:
4450: 25: LINHA:="INICIALIZACAO ERRADA. ESPERADO: PREMISA/CONCLUSAO";
4460:
4470: 26: LINHA:="INICIALIZACAO ERRADA. ESPERADO: NUMERO DA REGRA";
4480:
4490: 27: LINHA:="INICIALIZACAO ERRADA. ESPERADO: REGRA NOVA/EXCLUSAO/"
4500: CAT "ALTERACAO/MAISPE/"
4510: CAT "REFERENCIAS/PEGRA ESPECIAL/MENU/FIM";
4520:
4530: 28: LINHA:="LISTA MUITO GRANDE (MAXIMO = 72 CAPACIDADES)";
4540:
4550: 29: LINHA:="LISTA INVALIDA. JA UTILIZADA PARA IDENTIFICACAO " CAT
4560: "DE OUTRA PALAVRA";
4570:
4580: 30: LINHA:="EXCLUSAO IMPOSSIVEL. ";
4590: IF PREMISA THEN LINHA:=LINHA CAT "A PREMISA "
4600: ELSE LINHA:=LINHA CAT "A CONCLUSAO ";
4610: LINHA:=LINHA CAT "NAO PODE SER VAZIA";
4620:
4630: 40:
4640: LINHA:="POR FAVOR, RESPONDA SIM OU NAO";
4650:
4660: 41: LINHA:="SUA RESPOSTA DEVE SER NUMERICA";
4670:
4680: 42: LINHA:="NAO ENTENDI SUA RESPOSTA. POR FAVOR, RESPONDA NOVAMENTE";
4690:
4700: 43:

```

```

4700: LINHA:="SUA RESPOSTA DEVE SER UM VALOR DO INTERVALO [1,10]";
4710: 44:
4720: LINHA:="ESPERADO: ";
4730: FOR I:=1 STEP 1 UNTIL TOTAL DO
4740: LINHA:=LINHA CAT STRING(LISTAC(I),*) CAT " ";
4750: 45:
4760: LINHA:="PELO MENOS UM TIPO DE CONTEXTO DEVE SER DADO";
4770: 46:
4780: LINHA:="O IDENTIFICADOR NAO PODE SER VAZIO";
4790: 47:
4800: LINHA:="REGRA ESPECIAL SO' ACEITA A FUNCAO 'IGUAL' OU "
4810: CAT "UM SEU SINONIMO";
4820: 48:
4830: LINHA:="EM REGRA ESPECIAL NAO E' ACEITA A MANIPULACAO "
4840: CAT "DE SUPCLAUSULAS";
4850: 49:
4860: LINHA:="SUA RESPOSTA DEVE ESTAR ENTRE 1 E 4";
4870: 50:
4880: LINHA:="SUA RESPOSTA DEVE ESTAR ENTRE 1 E 7";
4890: END CASE;
4900: WRITE(VIDEO,<A1>>RUZINA);
4910: IMPRIMLINHA;
4920: ERROIDENTIFICADO:=TRUE;
4930: CATAO:=EMPTY;
4940: END ERRO;

```

```

4060:
4070:
4080: PROCEDURE PEGAPALAVRA;
4090: BEGIN
4100:   BOOLEAN ACABOU;
4110:   STRING CADEIA;
4120:   DEFINE TRATAMENTO =
4130:     BEGIN
4140:       ACABOU:=FALSE;
4150:       CADEIA:=ORLP(PALAVRA,1);
4160:       WHILE NOT ACABOU DO BEGIN
4170:         PALAVRA:=HEAD(TAIL(CARTAO," "),"NOT " " ");
4180:         CAPTAC:=TAIL(TAIL(CARTAO," "),"NOT " " ");
4190:         IF PALAVRA = EMPTY THEN BEGIN
4200:           WRITE(VIDEO[STOP],<">">);
4210:           READ(VIDEO,<A60>,<CARTAO>);
4220:           END
4230:         ELSE IF DROP(PALAVRA,LENGTH(PALAVRA) - 1) = "/" THEN BEGIN
4240:           PALAVRA:=TAKE(PALAVRA,LENGTH(PALAVRA) - 1);
4250:           IF PALAVRA NEG EMPTY THEN
4260:             CADEIA:=CADEIA CAT " " CAT PALAVRA;
4270:           ACABOU:=TRUE;
4280:           END
4290:         ELSE CADEIA:=CADEIA CAT " " CAT PALAVRA;
4300:       END;
4310:       PALAVRA:=TAIL(CADEIA," ");
4320:       COMBARRA:=TRUE;
4330:     END #;
4340:
4350: COMMENT :
4360:   * INICIO DE PEGAPALAVRA *
4370:
4380: TOMA A PROXIMA PALAVRA DO CARTAO *CASO O CARTAO JA TENHA ACABADO,

```

```

5290: E' LIDO UM NOVO CARTAO E TOMA-SE A PRIMEIRA PALAVRA, SE O CARTAO
5300: NOVO ESTIVER VAZIO A PALAVRA E' DEVOLVIDA COMO SENDO VAZIA ;
5310:
5320: COMBARRA:=FALSE;
5330: CARTAO:=TAIL(CARTAO," ");
5340: IF CARTAO = EMPTY THEN BEGIN
5350:   WRITE(VIDEO[STOP], "<" ">");
5360:   READ(VIDEO, <A66>, CARTAO);
5370:   CARTAO:=TAIL(CARTAO," ");
5380:   IF CARTAO = EMPTY THEN PALAVRA:=EMPTY;
5390:   END;
5400: PALAVRA:=HEAD(CARTAO, NOT " ");
5410: CARTAO:=TAIL(CARTAO, NOT " ");
5420: IF PALAVRA NEQ EMPTY THEN
5430:   IF TAKE(PALAVRA,1) = "/" THEN TPATAIDENT;
5440: IF LENGTH(PALAVRA) > 42 THEN BEGIN
5450:   WRITE(VIDEO, <A1>, BUZINA);
5460:   PALAVRA:=TAKE(PALAVRA,42);
5470:   WRITE(VIDEO, "<" PALAVRA COM MAIS DE 42 CARACTERES. "/.
5480:   "VOU CONSIDERAR <" A*, ">">, LENGTH(PALAVRA), PALAVRA);
5490:   END;
5500: END PGAPALAVRA;
5510:
5520: -----
5530:
5540: PROCEDURE LERESPSTA(X);
5550: VALUE X; INTEGER X;
5560: BEGIN
5570:
5580: COMMENT :
5590:   * INICIO DE LERESPSTA *
5600:
5610: LE A RESPOSTA DO ESPECIALISTA E VERIFICA SE SE ENCAIXA AO PADRAO
5620: DA PERGUNTA FEITA PELO SISTEMA ;
5630:
5640:
5650: ERROIDENTIFICADO:=FALSE;
5660: PGAPALAVRA;
5670: CASE X OF BEGIN
5680:   SIMNAO:
5690:     IF PALAVRA NEQ "NAO" AND PALAVRA NEQ "SIM" AND
5700:       PALAVRA NEQ "N" AND PALAVRA NEQ "S" THEN ERRO(40);
5710:   NUMERICA:
5720:     IF NOT NUMERO(PALAVRA) THEN ERRO(41);
5730:   LPODEMOCORRETA:
5740:     BEGIN
5750:       STRING P;
5760:       INTEGER I,N;
5770:       BOOLEAN ERROD;
5780:       ARRAY ELEMENTO[1:5];
5790:       ERROD:=FALSE;
5800:       FILL ELEMENTO[*] WITH 5(0);
5810:       N:=IF EXISTEVALOR THEN
5820:         IF CUNIDADE NEQ 0 THEN 5
5830:         ELSE 4
5840:         ELSE 3;
5850:       IF LENGTH(PALAVRA) > N THEN ERRO(42)
5860:       ELSE
5870:         IF TAIL(PALAVRA,DIGITOS) NEQ EMPTY THEN ERRO(42)

```

```

58800 ELSE BEGIN
58900 F:=PALAVRA;
59000 WHILE P NEQ EMPTY DO BEGIN
59100 ELEMENTO[DECIMAL(TAKE(P,1))]:=++1;
59200 P:=DROP(P,1);
59300 END;
59400 FOR I:=1 STEP 1 UNTIL N DO
59500 IF ELEMENTO[I] NEQ 1 THEN LPRADO:=TRUE;
59600 IF LPRADO THEN ERRO(42);
59700 END;
59800 END;
59900 VALORCFATEZA:
60000 IF NUMERO(PALAVRA) THEN BEGIN
60100 CFATOR:=DECIMAL(PALAVRA);
60200 IF CFATOR < 1 OR CFATOR > 10 THEN ERRO(43);
60300 ELSE CFATOR:=CFATOR/10;
60400 END;
60500 ELSE LPRU(43);
60600 NAODUNUMERO:
60700 BEGIN
60800 ROLCAN ACHET;
60900 INTEGER I;
61000 IF PALAVRA NEQ EMPTY AND
61100 PALAVRA NEQ "NAO" AND PALAVRA NEQ "N" THEN BEGIN
61200 IF NUMERO(PALAVRA) THEN BEGIN
61300 ACHET:=FALSE;
61400 FOR I:=1 STEP 1 WHILE (I <= TOTAL) AND (NOT ACHET) DO
61500 IF LISTA[I] = DECIMAL(PALAVRA) THEN ACHET:=TRUE;
61600 IF NOT ACHET THEN ERRO(44);
61700 END;
61800 ELSE ERRO(44);
61900 END;
62000 END;
62100 END CASE;
62200 IF ERROIDENTIFICADO THEN LERESPOTA(X);
62300 END LERESPOTA;
62400
62500 -----
62600
62700 PROCEDURE LIMPAEGRVA(X);
62800 VALUE X; INTEGER X;
62900 BEGIN
63000 INTEGER ULTIMO;
63100 ARRAY DIRECTORY(0:9);
63200
63300 COMMENT :
63400 * INICIO DE LIMPAEGRVA *
63500
63600 GRAVA UM REGISTRO EM DISK ;
63700
63800 ULTIMO:=DISK.LASTRECORD+1;
63900 FILL DIRECTORY[*] WITH 10(0);
64000 DIRECTORY[0].[47:05]:=X;
64100 DIRECTORY[0].[42:10]:=ULTIMO;
64200 DIRECTORY[0].[32:10]:=ULTIMO;
64300 DIRECTORY[0].[22:10]:=ASSOC;
64400 REPLACE POINTER(DIRECTORY[3]) BY (PALAVRA CAT "/");
64500 WRITE(DISK[ULTIMO],10,DIRECTORY);
64600 END;

```

```

6470
6480
6490
6500
6510
6520
6530
6540
6550
6560
6570
6580
6590
6600
6610
6620
6630
6640
6650
6660
6670
6680
6690
6700
6710
6720
6730
6740
6750
6760
6770
6780
6790
6800
6810
6820
6830
6840
6850
6860
6870
6880
6890
6900
6910
6920
6930
6940
6950
6960
6970
6980
6990
7000
7010
7020
7030
7040
7050

```

```

PROCEDURE PESQUISADIRETORIO;
BEGIN
    BOOLEAN ACHEI;

    COMMENT :
        * INICIO DE PESQUISADIRETORIO *

    PESQUISA O DICCIONARIO USANDO COMO CHAVE O NOME DA PALAVRA *A
    VARIÁVEL "L" CONTEM O NRO DO REGISTRO ;

    ACHEI:=FALSE;
    L:=1;
    DO BEGIN
        READ(DIRDISK[L],10,DIRETORIO);
        IF PALAVRA = NOME THEN ACHEI:=TRUE
        ELSE L:=L+1;
    END
    UNTIL (ACHEI) OR (L = DIRDISK*LASTRECORD + 1);
    PESQUISADIRETORIO:=ACHEI;
END PESQUISADIRETORIO;

```

```

INTEGER PROCEDURE OUTRAPALAVRA(X);
STRING X;
BEGIN
    INTEGER I;
    ARRAY P[0:6];
    BOOLEAN ACH I;

    COMMENT :
        * INICIO DE OUTRAPALAVRA *

    PROCURA A PALAVRA "X" NO ARQUIVO OUTRASPALAVRAS .CASO NAO ESTEJA LA
    A PALAVRA E' INSERIDA NO ARQUIVO .O PROCEDIMENTO RETORNA O NUMERO
    CORRESPONDENTE A "X" ;

    I:=0;
    ACHEI:=FALSE;
    DO BEGIN
        READ(OUTRASPALAVRAS[I],7,P);
        IF X = HE/D(STRING(POINTER(P),42),NOT "") THEN ACHEI:=TRUE
        ELSE I:=I+1;
    END
    UNTIL (ACHEI) OR (I = OUTRASPALAVRAS*LASTRECORD + 1);
    IF NOT ACHEI THEN BEGIN
        FILL P[*] WITH 7(" ");
        REPLACE POINTER(P) BY X CAT "";
        WRITE(OUTRASPALAVRAS[I],7,P);
    END;
    OUTRAPALAVRA:=I;
END;

```

```

PROCEDURE DECODIFICACLAUSULA(X);

```

```

7060:  VALUE X;  INTEGER X;
7070:  BEGIN
7080:      STRING CARTAO,PALAVRA;
7090:      DEFINE PEGAPALAVRA =
7100:          CARTAO:=TAIL(CARTAO," ");
7110:          IF CARTAO NEQ EMPTY THEN BEGIN
7120:              PALAVRA:=HEAD(CARTAO,NOT " ");
7130:              CARTAO:=TAIL(CARTAO,NOT " ");
7140:              END
7150:          ELSE PALAVRA:="" #;
7160:
7170:
7180:  COMMENT :
7190:      * INICIO DE DECODIFICACLAUSULA *
7200:
7210:      TOMA O ESQUELETO E SUBSTITUI CADA SIMBOLO PELA PALAVRA
7220:      CORRESPONDENTE. SE FOR UM DOS PARAMETROS PRINCIPAIS, ESTE E' TIRADO
7230:      DO ARRAY ELEMENTO, SENAO DO ARQUITVO OUTRASPALAVRAS ;
7240:
7250:      READ(ESQUELETO,DISK(X),30,ESQUELETO);
7260:      PART,0:=STRING(POINTER(ESQUELETO),180);
7270:
7280:      PEGAPALAVRA;
7290:      WHILE PALAVRA NEQ "" DO BEGIN
7300:          IF TAKE(PALAVRA,1) = "s" THEN BEGIN
7310:              INTEGER N;
7320:              DEFINE UNUMERO = N > 6 #;
7330:              N:=DECIMAL(DROP(PALAVRA,1));
7340:              IF UNUMERO THEN PALAVRA:=STRING(CVALOR,N);
7350:              ELSE PALAVRA:=ELEMENTO[N];
7360:              END
7370:
7380:          ELSE BEGIN
7390:              ARRAY P[0:6];
7400:              IF PALAVRA=EMPTY THEN WRITE(VIDEO,"FISSO");
7410:              READ(OUTRASPALAVRAS,DECIMAL(PALAVRA),7,P);
7420:              PALAVRA:=HEAD(STRING(POINTER(P),30),NOT " ");
7430:              END;
7440:              LINHA:=LINHA CAT PALAVRA CAT " ";
7450:              PEGAPALAVRA;
7460:              END;
7470:
7480:  END DECODIFICACLAUSULA;
7490:
7500:  -----
7510:
7520:  PROCEDURE ARMAZENATEXTO;
7530:  BEGIN
7540:  COMMENT :
7550:      CONCATENA AS PALAVRAS DO TEXTO E ARMAZENA EM TEXTODISK NO PRIMEIRO
7560:      REGISTRO DISPONIVEL OU NO PRIMEIRO DEPOIS DO ULTIMO REGISTRO DE
7570:      TEXTODISK ;
7580:      ROLLEA TEXTODARMAZENADO;
7590:      STRING TEXT;
7600:      ARRAY ESPACO[0:89];
7610:      DEFINE AVAIL = ESPACO[0] #;
7620:      PEGAPALAVRA =
7630:          TEXTUAL:=TAIL(TEXTUAL," ");
7640:          IF TEXTUAL NEQ EMPTY

```



```
7650:         THEN BEGIN
7660:             PALAVRA:=HEAD(TTEXTUAL ,NOT " ");
7670:             TEXTUAL:=TAIL(TTEXTUAL ,NOT " ");
7680:             END #;
7690:
7700: -----
7710:
7720: PROCEDURE FIMDETETO;
7730: BEGIN
7740:
7750:     COMMENT:
7760:         * INICIO DE FIMDETETO *
7770:
7780:     RECUPERAR INFORMACAO SOBRE ESPACO DISPONIVEL, RESULTANTE DE
7790:     TEXTOS DESATIVADOS;
7800:
7810:     TEXTO:=TRUE;
7820:     READ(T:XTODISK(0),90,ESPACO);
7830:     IF AVAIL NEG 0 THEN BEGIN
7840:         ETEXT0:=ESPACO(AVAIL);
7850:         AVAIL:=AVAIL-1;
7860:         WRITE(TXTODISK(0),90,ESPACO);
7870:         END
7880:     ELSE ETEXT0:=TEXTODISK.LASTRECORD+1;
7890:     FILL ESPACO(+) WITH 90(0);
7900:     TEXT:=TAIL(TEXT," ");
7910:     REPLACE POINTER(ESPACO) BY TEXT;
7920:     WRITE(TXTODISK(ETEXT0),90,ESPACO);
7930:     TEXTOARMAZENADO:=TRUE;
7940:     END FIMDETETO;
7950:
7960:     COMMENT:
7970:         * INICIO DE ARMAZENATEXTO *
7980:
7990:     CONCATENA AS PALAVRAS DO TEXTO E ARMAZENA EM TEXTODISK
8000:     NO PRIMEIRO REGISTRO DISPONIVEL OU NO PROXIMO AO
8010:     ULTIMO REGISTRO DE TEXTODISK ;
8020:
8030:     TEXT:=EMPTY;
8040:     TEXTUAL:=TEXTUAL CAT " ";
8050:     WHILE NOT TEXTOARMAZENADO DO BEGIN
8060:         PEGAPALAVRA;
8070:         IF PALAVRA = EMPTY THEN ERRO(10)
8080:         ELSE
8090:             IF PALAVRA = ";" THEN BEGIN
8100:                 TEXT:=TEXT CAT " " CAT " ";
8110:                 FIMDETETO;
8120:                 END
8130:             ELSE
8140:                 IF LENGTH(TEXT)+LENGTH(PALAVRA) < 540 THEN
8150:                     TEXT:=TEXT CAT " " CAT PALAVRA
8160:                 ELSE ERRO(15);
8170:             END;
8180:         END ARMAZENATEXTO;
8190:
8200: -----
8210:
8220: PROCEDURE RECONHECECLAUSULA;
8230: BEGIN
```

```

8240:      INTEGER CONTADOR;
8250:
8260:  -----
8270:
8280:  PROCEDURE IDENTIFICAPALAVRAS;
8290:  BEGIN
8300:      RFAL VAL;
8310:      BOOLEAN ACHEIVALOR;
8320:      INTEGER UNIDADECOMPATIVEL;
8330:      DEFINE SINON = DIRECTORY[0].[42:10] #;
8340:      ANTON = DIRECTORY[0].[32:10] #;
8350:      TRATA(CLAUSULA) =
8360:  BEGIN
8370:      BOOLEAN ACABOU;
8380:      STRING CADEIA;
8390:
8400:      CADEIA:=EMPTY;
8410:      CLAUSULA:=TAIL(DROP(CLAUSULA,1)," ");
8420:      ACABOU:=FALSE;
8430:      WHILE NOT ACABOU DO BEGIN
8440:          PALAVRA:=HEAD(TAIL(CLAUSULA," "),NOT " ");
8450:          CLAUSULA:=TAIL(TAIL(CLAUSULA," "),NOT " ");
8460:          IF PALAVRA = EMPTY THEN ACABOU:=TRUE;
8470:          ELSE IF DROP(PALAVRA,LENGTH(PALAVRA) - 1) = "/" THEN REGTH
8480:              ACABOU:=TRUE;
8490:              PALAVRA:=TAKE(PALAVRA,LENGTH(PALAVRA) - 1);
8500:              CADEIA:=CADEIA CAT PALAVRA;
8510:              END
8520:          ELSE CADEIA:=CADEIA CAT PALAVRA CAT " ";
8530:      END;
8540:      IF CADEIA = EMPTY THEN PALAVRA:=EMPTY;
8550:      ELSE IF DROP(CADEIA,LENGTH(CADEIA) - 1) = " "
8560:          THEN PALAVRA:=DROP(CADEIA,LENGTH(CADEIA) - 1)
8570:          ELSE PALAVRA:=CADEIA;
8580:      END #;
8590:
8600:  ARRAY DIRECTORY[0:9];
8610:  -----
8620:  PROCEDURE LISTAPALAVRASCLASSE(X);
8630:  VALUE X;  INTEGER X;
8640:  BEGIN
8650:      STRING ARRAY N[1:100];
8660:      DEFINE UNIDADE = 5 #;
8670:      PROCENALISTA = BEGIN
8680:          TOTAL:=TOTAL+1;
8690:          LISTA[TOTAL]:=L;
8700:          N[TOTAL]:=NPME;
8710:          END #;
8720:
8730:  COMMENT :
8740:      * INICIO DE LISTAPALAVRASCLASSE *
8750:
8760:  LISTA TODAS AS PALAVRAS DE DIRDISK DA CLASSE "X" ;
8770:
8780:  L:=1;
8790:  TOTAL:=0;
8800:  DO BEGIN
8810:      READ(DIRDISK[L],10,DIRETORIO);
8820:      IF CLASSE = X THEN

```

```

8830:         IF X = UNIDADE THEN
8840:             IF ASSOC = CATRIBUTO THEN POENALISTA
8850:             ELSE
8860:                 ELSE POENALISTA;
8870:             L:=L+1;
8880:         END
8890:     UNTIL L > DIRDISK.LASTRECORD;
8900:     IF TOTAL NEQ 0 THEN
8910:         WRITE(VIDEO, << "PALAVRAS CONHECIDAS:" // "*" (J3, " = ", A+, // ">>, TOTAL,
8920:         FOR X:=1 STEP 1 UNTIL TOTAL DO ( LISTA[X], LENGTH(N[X]), N[X] );
8930:     END LISTAPALAVRASCLASSE;
8940:
8950: -----
8960: PROCEDURE PROCURASINONIMODE(X);
8970: VALUE X;    INTEGER X;
8980: BEGIN
8990:
9000: COMMENT :
9010:         * INICIO DE PROCURASINONIMODE *
9020:
9030:         MANDA LISTAR AS PALAVRAS DA CLASSE "X" E PERGUNTA AO USUARIO SE
9040:         A NOVA PALAVRA E' SINONIMO DE UMA DESTAS ;
9050:
9060:         LISTAPALAVRASCLASSE(X);
9070:         IF TOTAL NEQ 0 THEN BEGIN
9080:             LINHA:=PALAVRA CAT " E' SINONIMO DE ALGUMA PALAVRA CONHECIDA?";
9090:             IMPRIMELINHA;
9100:             LEESPONSTA(NAOQUUNUMERO);
9110:             END
9120:         ELSE PALAVRA:=EMPTY;
9130:     END PROCURASINONIMODE;
9140:
9150: -----
9160:
9170: PROCEDURE VERIFICAANTONIMODEFUNCAO;
9180: BEGIN
9190:     INTEGER X;
9200:     ENDEFUNCO;
9210:     STRING P;
9220:     BOOLEAN PERGUNTEI;
9230:     NOVAFUNCAO;
9240:
9250: COMMENT:
9260:         * INICIO DE VERIFICAANTONIMODEFUNCAO *
9270:
9280:         ENCONTRADA UMA CONSTRUCAO DO TIPO "MODIFICADOR-FUNCAO":
9290:         (1) VERIFICA SE EXISTE ANTONIMO CONHECIDO
9300:         (2) CASO EXISTA, SUBSTITUI A FUNCAO PELO ANTONIMO E
9310:             ELIMINA O MODIFICADOR
9320:         (3) CASO NAO EXISTA, PERGUNTA AO USUARIO SE EXISTE SINONIMO PARA
9330:             AQUELA CONSTRUCAO, E EM CASO AFIRMATIVO, EXECUTA (2) E
9340:             ATUALIZA O DIRETORIO. EM CASO NEGATIVO, ATUALIZA O
9350:             VETOR SEMANTONIMO QUE CONTEM AS FUNCOES QUE NAO TEM ANTONIMO;
9360:
9370:         NOVAFUNCAO:=FALSE;
9380:         L:=CFUNCAO;
9390:         READ(DIRDISK[L], 10, DIRECTORY);
9400:
9410:

```

```

94200 IF ANTON NEO L THEN BEGIN
94300 READ(DIRDISK(ANTON),10,DIRETORIO);
94400 CFUNCAO:=SINONIMO;
94500 READ(DIRDISK(CFUNCAO),10,DIRETORIO);
94600 NOMEFUNCAO:=NOME;
94700 CMODFUNCAO:=0;
94800 END
94900
95000 ELSE BEGIN
95100 PERGUNTEI:=FALSE;
95200 FOR X:=1 STEP 1 WHILE (X <= PERGUNTAS) AND (NOT PERGUNTEI) DO
95300 IF (CFUNCAO = SEMANTONIMO(X)) THEN PERGUNTEI:=TRUE;
95400 IF NOT PERGUNTEI THEN BEGIN
95500 LINHA:="EXISTE SINONIMO PARA '" CAT NOME MODIFICADOR CAT "' "
95600 CAT NOMEFUNCAO CAT "'? (S/N)";
95700 IMPRIMELINHA;
95800 PEGAPALAVRA;
95900 IF PALAVRA NEO "NAO" AND PALAVRA NEO "N" AND
96000 PALAVRA NEO EMPTY THEN BEGIN
96100
96200 IF PALAVRA = "SIM" OR PALAVRA = "S" THEN BEGIN
96300 WRITE(VIDEO(STOP),<"ENTE COM O SINONIMO: ">);
96400 PEGAPALAVRA;
96500 PALAVRA:=* CAT "*";
96600 END;
96700
96800 COMMENT: APAGAR MODIFICADOR & ATUALIZAR ESQUELETO;
96900
97000 CMODFUNCAO:=0;
97100 SKELETON:=HEAD(SKELETON,NOT "S6") CAT
97200 DROP(TAIL(SKELETON,NOT "S6"),2);
97300 ENDEFECO:=L;
97400 IF PESQUISADIRETORIO THEN
97500 IF CLASSE NEO FUNCAO THEN ERRO(12)
97600 ELSE BEGIN
97700 CFUNCAO:=L;
97800 NOMEFUNCAO:=NOME;
97900 ANONIMO:=SINON;
98000 ANON:=SINONIMO;
98100 WRITE(DIRDISK(L),10,DIRETORIO);
98200 WRITE(DIRDISK(ENDEFECO),10,DIRECTORY);
98300 END
98400
98500 ELSE BEGIN
98600
98700 COMMENT:
98800 NESSE CASO, A PALAVRA NAO E CONHECIDA DO SISTEMA.
98900 DEVE-SE VERIFICAR SE ELA POSSUI UM SINONIMO CONHECIDO.
99000 (NOTA: QUE CASO NAO EXISTA SINONIMO, A CLAUSULA IRA
99100 CONTER A CONSTRUCAO <MODIFICADOR><FUNCAO>, E UM
99200 CUIDADO ESPECIAL DEVERA SER TOMADO PARA SUA AVALIACAO);
99300
99400 STRING P;
99500 P:=PALAVRA;
99600 PROCURASINONIMO(X);
99700 IF PALAVRA NEO "NAO" AND PALAVRA NEO "N" AND
99800 PALAVRA NEO EMPTY THEN
99900 READ(DIRDISK(DECIMAL(PALAVRA)),10,DIRECTORY);
10000 ELSE NOVAFUNCAO:=TRUE;

```

```

10010-
10020- COMMENT: ATUALIZACAO DO DIFETURIO;
10030-
10040- CLASSE:=X;
10050- IF NOVAFUNCAO THEN BEGIN
10060-     SINONIMO:=DIRDISK*LASTRECORD + 1;
10070-     READ(DIRDISK[ENDRECORD],10,DIRECTORY);
10080-     ANTON:=SINONIMO;
10090-     WRITE(DIRDISK[ENDRECORD],10,DIRECTORY);
10100-     END
10110- ELSE BEGIN
10120-     IF ANTON = DECIMAL(PALAVRA) THEN BEGIN
10130-         SINONIMO:=DIRDISK*LASTRECORD + 1;
10140-         READ(DIRDISK[DECIMAL(PALAVRA)],10,DIRECTORY);
10150-         ANTON:=ENDRECORD;
10160-         WRITE(DIRDISK[DECIMAL(PALAVRA)],10,DIRECTORY);
10170-         END
10180-     ELSE SINONIMO:=ANTON;
10190-     END;
10200-     ANTONIMO:=SINON;
10210-     REPLACE PNAME BY P;
10220-     CFUNCAO:=DIRDISK*LASTRECORD + 1;
10230-     NOMEFUNCAO:=NOME;
10240-     WRITE(DIRDISK[DIRDISK*LASTRECORD + 1],10,DIRECTORY);
10250-     END;
10260- END
10270- ELSE BEGIN
10280-     PERGUNTAS:=**+1;
10290-     SEMANTONIMO[PERGUNTAS]:=CFUNCAO;
10300-     END;
10310- END;
10320- END;
10330- END VERIFICAANTONINODEFUNCAO;
10340-
-----
10350-
10360-
10370- PROCEDURE ATUALIZAREGISTRODECLAUSULA;
10380- BEGIN
10390-     DEFINE
10400-         PERGUNTARPELOVALOR =
10410-             BEGIN
10420-                 WRITE(VIDEOP,"NAO FOI IDENTIFICADO VALOR PARA ",A**);
10430-                 LENGTH(NOMEATRIBUTO)*NOMEATRIBUTO);
10440-                 WRITE(VIDEOP[STOP],"ENTRE COM O VALOR: ");
10450-                 PEGAPALAVRA;
10460-                 IF NUMERO(PALAVRA) THEN BEGIN
10470-                     CVALOR:=DECIMAL(PALAVRA);
10480-                     CTIPOVALOR:=1;
10490-                     END
10500-                 ELSE
10510-                     IF PESQUISADIRETORIO THEN
10520-                         IF CLASSE NEQ VALOR THEN ERRO(12)
10530-                     ELSE BEGIN
10540-                         CVALOR:=L;
10550-                         NOMEVALOR:=NOME;
10560-                         CTIPOVALOR:=0;
10570-                         END
10580-                     ELSE INSEREPALAVRACLASSE(VALOR);
10590-                     END #;

```

```

10600
10610
10620
10630
10640
10650
10660
10670
10680
10690
10700
10710
10720
10730
10740
10750
10760
10770
10780
10790
10800
10810
10820
10830
10840
10850
10860
10870
10880
10890
10900
10910
10920
10930
10940
10950
10960
10970
10980
10990
11000
11010
11020
11030
11040
11050
11060
11070
11080
11090
11100
11110
11120
11130
11140
11150
11160
11170
11180

```

```

PROCEDURE INSEPEPALAVRACLASSE(X);
VALUE X; INTEGER X;
BEGIN
  BOOLEAN ACHEI;
  INTEGER CODIC, CHAVE, ULTIMO;
  STRING P;
  DEFINE
    ATUALIZAATRIBUTO =
    BEGIN
      BOOLEAN ACHEI;
      ACHEI:=FALSE;
      DO BEGIN
        WRITE(VIDEO, "<" COMO ESTE ATRIBUTO PODE SER CLASSIFICADO?"
          ,//, " 0) ATRIBUTO SIM/NAD" ,//,
            " 1) ATRIBUTO DO TIPO UNICO" ,//,
            " 2) ATRIBUTO DO TIPO MULTIPLA">);
        LERESPONSTA(NUMERICA);
        TIPOATRIBUTO:=DECIMAL(PALAVRA);
        IF TIPOATRIBUTO GEQ 0 AND TIPOATRIBUTO LEQ 2
          THEN ACHEI:=TRUE;
        END
      UNTIL ACHEI;
      ACHEI:=FALSE;
      IF TIPOATRIBUTO NEQ 0 THEN
        DO BEGIN
          WRITE(VIDEO, "<" COMO OS VALORES RELACIONADOS A ESTE"
            " ATRIBUTO DEVEM SER CLASSIFICADOS?" ,//,
              " 0) TIPO TEXTO" ,//,
              " 1) TIPO NUMERICO" ,//,
              " 2) TIPO NAO-NUMERICO(NOMES)">);
          LERESPONSTA(NUMERICA);
          ASSOC:=DECIMAL(PALAVRA);
          IF ASSOC GEQ 0 AND ASSOC LEQ 2
            THEN ACHEI:=TRUE;
          END
        UNTIL ACHEI;
      END #;
    END;

PROCEDURE ATUALIZAUNIDADE(PATRIBUTO);
INTEGER PATRIBUTO;
BEGIN
  STRING F;
  INTEGER PUNIDADE;

  COMMENT:
    * INICIO DE ATUALIZAUNIDADE *

  TOMA A UNIDADE-PADRAO REFERENTE AO ATRIBUTO, SE FLE
  EXISTIR ;

  WRITE(VIDEO, "<"** ESTE ATRIBUTO ESTA ASSOCIADO A ALGUMA "
    "UNIDADE ?">);
  LERESPONSTA(SIMNAD);
  IF PALAVRA = "SIM" OR PALAVRA = "S" THEN DO BEGIN
    ERROIDENTIFICADO:=FALSE;

```

```

11190:      WRITE(VIDEO, <"** QUAL A UNIDADE PRINCIPAL ASSOCIADA "
11200:      "A ESTA ATRIBUTO?"> />);
11210:      PUNIDADE:=DIRDISK.LASTRECORD + 1;
11220:      PEGAPALAVRA;
11230:      P:=PALAVRA;
11240:      IF P = EMPTY THEN ERROIDENTIFICADO:=TRUE
11250:      ELSE BEGIN
11260:        IF PESQUISADIRETORIO THEN PUNIDADE:=L
11270:        ELSE BEGIN
11280:          PROCURASINONIMO(PUNIDADE);
11290:          IF NUMERO<PALAVRA>
11300:          THEN READ(DIRDISK(DECIMAL(PALAVRA),10,DIRETORIO))
11310:          ELSE BEGIN
11320:            FILL DIRETORIO[*] WITH 10(0);
11330:            CLASSE:=PUNIDADE;
11340:            SINONIMO:=PUNIDADE;
11350:            ASSOC:=PATRIBUTO;
11360:            END;
11370:            REPLACE PNAME BY (P CAT "/");
11380:            ANTONIMO:=PUNIDADE;
11390:            WRITE(DIRDISK(PUNIDADE,10,DIRETORIO));
11400:            END;
11410:            READ(DIRDISK(PATRIBUTO,10,DIRETORIO));
11420:            ASSOC:=PUNIDADE;
11430:            WRITE(DIRDISK(PATRIBUTO,10,DIRETORIO));
11440:            END;
11450:          END
11460:          UNTIL NOT ERROIDENTIFICADO;
11470:        END;
11480:
11490:  PROCEDURE ATUALIZATIPOSCONTEXTOS;
11500:  BEGIN
11510:    ARRAY
11520:      ASSOCI[1:4], NOME[0:8];
11530:    STRING
11540:      PRIMEIROM,
11550:      NOADICIONAL,
11560:      TRADUZ,
11570:      DADO,
11580:      CADEIA,
11590:      NRLEPOSICAO;
11600:    INTEGER
11610:      T,
11620:      REG,
11630:      MODU,
11640:      NUMERO,
11650:      CONTA,
11660:      ULTIMO,
11670:      TAMANHO,
11680:      LISTATUAL;
11690:    BOOLEAN
11700:      ACHEI,
11710:      ESCREVA,
11720:      MAGLEUTUDO,
11730:      CNTXTNULO,
11740:      ACABOU,
11750:      CONTINUA;
11760:    POINTER
11770:      P;

```

```

117800  DEFINE
117900      OBJETO = REGISTRO[0].[47:10] #
118000      ASSUMIR = REGISTRO[0].[37:1] #
118100      NUMASSOCIADOS = REGISTRO[0].[36:3] #
118200      ASSOCIADO(X) = REGISTRO[0].[33-(X-1)*8:8] #
118300      ATRIBUTOPRINCIPAL(X) = REGISTRO[1].[47-(X-1)*10:10] #
118400      LENG(X) = REGISTRO[2].[37-(X-1)*10:10] #
118500      TIPOCNTXT=REGISTRO[2].[47:10]#
118600
118700
118800      PROCURAEMOBJETOETIPOS(TIPO,CHAVE)=
118900          BEGIN
119000              READ(OBJETOETIPOS[CHAVE],4,LISTAETIPOS);
119100              ACHEI:=FALSE;
119200              CONTA:=1;
119300              WHILE NOT ACHEI AND CONTA<=LTOTAL
119400                  DO IF CONTEXTO(CONTA)=TIPO
119500                      THEN ACHEI:=TRUE
119600                      ELSE CONTA:=*+1;
119700              END#;
119800
119900      PROCEDURE LEADADO(X);
120000      INTEGER X;
120100      BEGIN
120200          CADEIA:=EMPTY;
120300          ACABOU:=FALSE;
120400          WHILE NOT ACABOU DO BEGIN
120500              WRITE(VIDEO[STOP],<">">);
120600              READ(VIDEO,<A66>,DADO);
120700              IF TAIL(DADO," ") = EMPTY THEN DADO:=" " ELSE
120800                  WHILE DROP(DADO,LENGTH(DADO) - 1) = " " DO
120900                      DADO:=TAKE(DADO,LENGTH(DADO) - 1);
121000              IF X = 1 THEN BEGIN
121100                  CADEIA:=DADO;
121200                  ACABOU:=TRUE;
121300                  END
121400              ELSE BEGIN
121500                  CADEIA:=CADEIA CAT HEAD(DADO,NOT " ");
121600                  IF TAIL(DADO,NOT " ") NEQ EMPTY
121700                      THEN ACABOU:=TRUE;
121800                  END;
121900              END;
122000          IF TAMANHO + LENGTH(CADEIA) > 222
122100              THEN CADEIA:=TAKE(CADEIA,222 - TAMANHO);
122200          REPLACE PIP BY CADEIA;
122300          LENG(X):=LENGTH(CADEIA);
122400          TAMANHO:=TAMANHO + LENG(X);
122500      END;
122600
122700
122800  COMMENT :
122900      * INICIO DE ATUALIZATIPOSCONTEXTO *
123000
123100      ATUALIZA OS ARQUIVOS TIPOSCONTEXTO E OBJETOETIPOS. QUANDO E'
123200      CITADO UM ATRIBUTO NAO EXISTENTE, ATUALIZA TAMBEM O DICINARIO ;
123300
123400      REG:=TIPOSCONTEXTO*LASTRECORD + 1;
123500      OBJETO:=CODIC;
123600      CNTXTNULO:=FALSE;

```



```

123700  ESCRIVA:=TRUE;
123800  WRITE(VIDEO,<"/,"** TIPO DE CONTEXTO NO. ">J3>"REG));
123900  P:=POINTER(REGISTRO[31]);
124000  LISTATUAL:=CHAVEREG;
124100  WRITE(VIDEO,<"TIPO DE CONTEXTO : ">);
124200  PEGAPALAVRA;
124300  IF PALAVRA NEQ EMPTY
124400  THEN IF PESQUISADIRETORIO
124500    THEN IF DIRETORIO[0].[47:05] NEQ 7
124600      THEN ERRO(5)
124700      ELSE BEGIN
124800        PROCURAEMOBJETMETIPOS(L,LISTATUAL);
124900        IF ACHEI THEN ERRO(6)
125000        ELSE IF NULO = 1 THEN ERRO(8);
125100        TIPOCNTXT:=L;
125200        END
125300        ELSE IF NULO = 1
125400          THEN ERRO(8)
125500          ELSE BEGIN
125600            TIPOCNTXT:=DIRDISK.LASTRECORD + 1;
125700            LIMPAEGRAVA(7);
125800            END
125900        ELSE BEGIN
126000          READ(OBJETOETIPOS[CHAVEREG],4,LISTADETIPOS);
126100          IF LTOTAL >= 1 THEN ERRO(7);
126200          TIPOCNTXT:=0;
126300          CNTXINULO:=TRUE;
126400          END;
126500  IF ESCRIVA
126600  THEN BEGIN
126700    TAMANHUI:=0;
126800    WRITE(VIDEO,<"TRADUZ : ">);
126900    LEUADO(1);
127000    DO BEGIN
127100      WRITE(VIDEO[STOP],<"ASSUMIR : ">);
127200      READ(VIDEO,<A66>"CAPTAD);
127300      LERESPSTA(NUMERICA);
127400      END
127500    UNTIL ASSUMIR = 0 OR ASSUMIR = 1;
127600    ASSUMIR:=DECIMAL(PALAVRA);
127700    WRITE(VIDEO,<"PRIMEIRO DO (TERMINE STRINGS COM ' ') : ">);
127800    LEUADO(2);
127900    WRITE(VIDEO,<"NOADICIONAL (TERMINE STRINGS COM ' ') : ">);
128000    LEUADO(3);
128100    DO BEGIN
128200      WRITE(VIDEO[STOP],<"NUMASSOCIADOS (MAX = 4) : ">);
128300      READ(VIDEO,<A40>"CAPTAD);
128400      LERESPSTA(NUMERICA);
128500      END
128600    UNTIL DECIMAL(PALAVRA)<=4;
128700    NUMASSOCIADOS:=NUMBER:=DECIMAL(PALAVRA);
128800    IF NUMBER>0
128900    THEN BEGIN
129000      WRITE(VIDEO[STOP],<"ASSOCIADOS : ">);
129100      READ(VIDEO,<"/>FOR I:=1 STEP 1 UNTIL NUMBER DO ASSOCI(I));
129200      END;
129300    FOR I:=1 STEP 1 UNTIL NUMBER DO ASSOCIADO(I):=ASSOCI(I);
129400    WRITE(VIDEO,<"ATRIBUTOS PRINCIPAIS : ">);
129500    I:=1;

```

```

129600 NAOLEUTUDO:=TRUE;
129700 WHILE NAOLEUTUDO AND I LEQ 4
129800 DO BEGIN
129900     PEGAPALAVRA;
130000     IF PALAVRA NEQ EMPTY
130100     THEN IF PESQUISA DIRETORIO
130200         THEN IF DIRETORIO[I].[47:05]=1
130300             THEN BEGIN
130400                 ATRIBUTO PRINCIPAL(I):=L;
130500                 I:=I+1;
130600             END
130700             ELSE BEGIN
130800                 NAOLEUTUDO:=FALSE;
130900                 ERRO(5);
131000             END
131100         ELSE BEGIN
131200             INTEGER ULTIMO;
131300             STRING PAL;
131400             ULTIMO:=DIRDISK*LASTRECORD + 1;
131500             ATRIBUTO PRINCIPAL(I):=ULTIMO;
131600             I:=I+1;
131700             PAL:=PALAVRA;
131800             PROCURASINONIMO(ATRIBUTO);
131900             IF NUMERO(PALAVRA) THEN BEGIN
132000                 READ(DIRDISK[DECIMAL(PALAVRA)],10,DIRETORIO);
132100                 IF ANTONIMO = DECIMAL(PALAVRA)
132200                 THEN ANTONIMO:=ULTIMO;
132300                 REPLACE PHONE BY (PAL CAT "/");
132400                 WRITE(DIRDISK[ULTIMO],10,DIRETORIO);
132500             END
132600             ELSE BEGIN
132700                 FILL DIRETORIO[*] WITH 10(0);
132800                 ATUALIZAATRIBUTO;
132900                 CLASSE:=1;
133000                 REPLACE PHONE BY (PAL CAT "/");
133100                 SINONIMO:=ULTIMO;
133200                 ANTONIMO:=ULTIMO;
133300                 WRITE(DIRDISK[ULTIMO],10,DIRETORIO);
133400                 IF ASSOC = 1 THEN ATUALIZAUNIDADE(ULTIMO);
133500             END;
133600         END
133700     ELSE NAOLEUTUDO:=FALSE;
133800 END;
133900 END;
134000 IF ESCREVA
134100 THEN BEGIN
134200     WRITE(TIPOSCONTEXT[REG],40,REGISTRO);
134300     XCOLCA TIPOCNTXT FM CHAVEREG
134400     READ(OBJETO TIPOS[LISTATUAL],4,LISTA DE TIPOS);
134500     LTOTAL:=LTOTAL+1;
134600     IF CNTXTNULO
134700     THEN BEGIN
134800         NULO:=1;
134900         TNULO:=REG;
135000     END
135100     ELSE BEGIN
135200         CONTEXTO[LTOTAL]:=TIPOCNTXT;
135300         NOTIPO[LTOTAL]:=REG;
135400     END;

```

```

13550:         WRITE(OBJETOETIPOS(LISTATUAL),4,LISTADETIPOS);
13560:     END;
13570: END ATUALIZATIPOSCONTEXTOS;
13580:
13590: COMMENT:
13600: * INICIO DE INSEREPALAVRACLASSE *
13610:
13620: INSERE UMA NOVA PALAVRA NO DICCIONARIO E ATUALIZA
13630: TODOS OS CAMPOS CORRESPONDENTES 'A SUA CLASSF.
13640: PARA OBJETO, O ARQUIVO TIPOSCONTEXTOS E' ATUALIZADO
13650: TAMBEM. O PARAMETRO DA CLAUSULA CORRESPONDENTE 'A
13660: CLASSE DA PALAVRA TAMBEM E' ATUALIZADO ;
13670:
13680: COMMENT: SALVAR A PALAVRA A SER INSERIDA NO DICCIONARIO;
13690: P:=PALAVRA;
13700:
13710: COMMENT: VERIFICAR SE E' SINONIMO DE ALGUMA JA CONHECIDA;
13720:
13730: PROCURASINONIMO(X);
13740: IF NUMERO(PALAVRA) THEN BEGIN
13750:     READ(DIRDISK(DECIMAL(PALAVRA)),10,DIRETORIO);
13760:     IF ANTONIMO = DECIMAL(PALAVRA)
13770:     THEN ANTONIMO:=DIRDISK.LASTRECORD + 1;
13780:     ELEMENTO(X):=P;
13790:     REPLACE PNUME BY (P CAT "/");
13800:     WRITE(DIRDISK(DIRDISK.LASTRECORD+1),10,DIRETORIO);
13810:     CASE X OF BEGIN
13820:         1: CATRIBUTO:=DIRDISK.LASTRECORD;
13830:         2: OBJETO:=DIRDISK.LASTRECORD;
13840:         3: CFUNCAO:=DIRDISK.LASTRECORD;
13850:         4: CVALOR:=DIRDISK.LASTRECORD;
13860:         5: CUNIDADE:=DIRDISK.LASTRECORD;
13870:         6: CMODFUNCAO:=DIRDISK.LASTRECORD;
13880:     END;
13890: END
13900: ELSE
13910:
13920: COMMENT: NOVAS FUNCOES NAO PODEM SER ACEITAS (SEMANTICA ???);
13930: IF X = FUNCAO THEN ERRO(4)
13940: ELSE BEGIN
13950:
13960:     COMMENT: ATUALIZACAO DE CAMPOS PARA GRAVACAO NO DICCIONARIO;
13970:
13980:     FILL DIRETORIO(*) WITH 10(0);
13990:     CDIC:=DIRDISK.LASTRECORD+1;
14000:     CLASSE:=X;
14010:     VALDPSUE:=0;
14020:     REPLACE PNUME BY (P CAT "/");
14030:     SINONIMO:=CDIC;
14040:     ANTONIMO:=CDIC;
14050:
14060:     COMMENT: ATUALIZACAO DE CAMPOS ESPECIFICOS DE CADA CLASSE;
14070:     CASE CLASSE OF BEGIN
14080:
14090:         ATRIBUTO:
14100:             CATRIBUTO:=CDIC;
14110:             ATUALIZAATRIBUTO;
14120:             NOMEATRIBUTO:=NOME;
14130:             WRITE(DIRDISK(CDIC),10,DIRETORIO);

```

```

141400 IF ASSOC = 1 THEN ATUALIZAUNIDADE(CD1C);
141500
141600 OBJETO:
141700   COBJETO:=CD1C;
141800   NOMEDEBJETO:=NOME;
141900   NULO:=0; LTOTAL:=0;
142000   CHAVFREG:=OBJETOETIPDS.LASTRECORD +1;
142100   WRITE(OBJETOETIPOS[CHAVFREG],4,LISTADETIPOS);
142200   WRITE(D1RDISK[CD1C],10,DIRETORIO);
142300   ATUALIZATIPOSCONTEXT0;
142400
142500 UNIDADE:
142600   COMMENT: DETERMINAR O ENDEREÇO DA UNIDADE PRINCIPAL
142700           ASSOCIADA AO ATRIBUTO PARA ATUALIZAR "SUBMULTIPLO"
142800           E "VALORSUB";
142900
143000 BEGIN
143100   ARRAY DIRECTORY[0:9];
143200   DEFINE
143300     CLASS   = DIRECTORY[0].[47:15] #,
143400     SUBMULT = DIRECTORY[0].[12:10] #,
143500     ASSOCT  = DIRECTORY[0].[22:10] #,
143600     NAME    = HEAD(STRING(POINTER(DIRECTORY[3]),42),NOT "/")#;
143700   ASSOCT:=CATRIBUTO;
143800   L:=1;
143900   ACHEI:=FALSE;
144000 DO BEGIN
144100     READ(D1RDISK[L],10,DIRECTORY);
144200     IF CLASS = UNIDADE AND SUBMULT = L AND
144300        ASSOCT = CATRIBUTO THEN ACHEI:=TRUE;
144400     ELSE L:=L+1;
144500   END
144600 UNTIL ACHEI OR L > D1RDISK.LASTRECORD;
144700
144800 COMMENT: VALOR RELATIVO DESSA NOVA UNIDADE;
144900 IF ACHEI THEN BEGIN
145000   VALORSUB:=0;
145100   DO BEGIN
145200     WRITE(VIDEO,"QUANTO VALE A PELACAO: ",
145300           A*,"/",A*," ?">LENGTH(P)*P*LENGTH(NAME)*NAME);
145400     PEGAPALAVRA;
145500     IF NUMERO(PALAVRA) THEN VALORSUB:=1/DECTMAL(PALAVRA)
145600     ELSE BEGIN
145700       STRING NUMER,DENOM;
145800       NUMER:=HEAD(PALAVRA,NOT "/");
145900       PALAVRA:=TAIL(PALAVRA,NOT "/");
146000       IF TAKE(PALAVRA,1) = "/" THEN
146100         DENOM:=HEAD(DROP(PALAVRA,1),NOT " ");
146200       IF NUMERO(NUMER) AND NUMERO(DENOM) THEN
146300         VALORSUB:=DECTMAL(DENOM)/DECTMAL(NUMER);
146400       END;
146500     END
146600   UNTIL VALORSUB NEQ 0;
146700   SUBMULTIPLO:=L;
146800   CUNIDADE:=CD1C;
146900   END
147000 ELSE BEGIN
147100   VALORSUB:=1;
147200   SUBMULTIPLO:=CD1C;

```

```

14730:          CUNIDADE:=SUBMULTIPLO;
14740:          END;
14750:          NOMEUNIDADE:=NOME;
14760:          READ(DIRDISK[CATRIBUTO],10,DIRECTORY);
14770:          ASSOCT:=SUBMULTIPLO;
14780:          WRITE(DIRDISK[CATRIBUTO],10,DIRECTORY);
14790:          WRITE(DIRDISK[CDIC],10,DIRETORIO);
14800:          END;
14810:          VALOR:
14820:          ASSOC:=CATRIBUTO;
14830:          CTIPOVALOR:=0;
14840:          CVALOR:=CDIC;
14850:          NOMEVALOR:=NOME;
14860:          WRITE(DIRDISK[CDIC],10,DIRETORIO);
14870:          END;
14880:
14890:          END;
14900:      END INSEREPALAVFACLASSEX;
14910:
14920:      COMMENT:
14930:          * INICIO DE ATUALIZAPEGISTRODECLAUSULA *
14940:
14950:          VERIFICA SE FORAM IDENTIFICADOS ATRIBUTO, OBJETO,
14960:          FUNCAO E VALOR, ATUALIZANDO "CLAUSULACND".
14970:          (CVALOR E' IDENTIFICADO EM IDENTIFICA-PALAVRAS);
14980:
14990:          IF CATRIBUTO = 0 AND CUNIDADE NEQ 0
15000:              THEN BEGIN
15010:
15020:                  COMMENT: NESSE CASO, NAO FOI IDENTIFICADO ATRIBUTO
15030:                  MAS FOI IDENTIFICADA UNIDADE. VERIFICAR SE
15040:                  O ATRIBUTO E' O ASSOCIADO A ESSA UNIDADE;
15050:
15060:                  REAL SALVASSOC;
15070:                  READ(DIRDISK[CUNIDADE],10,DIRETORIO);
15080:                  SALVASSOC:=ASSOC;
15090:                  READ(DIRDISK[ASSOC],10,DIRETORIO);
15100:                  WRITE(VIDEO,"O ATRIBUTO E' "A*" ?">,"LENGTH(NOME),
15110:                  NOME);
15120:                  PEGAPALAVRA;
15130:                  IF PALAVRA = "SIM" OR PALAVRA = "S" THEN BEGIN
15140:                      CATRIBUTO:=SALVASSOC;
15150:                      NOMEATRIBUTO:=NOME;
15160:                  END
15170:                  END;
15180:          IF CATRIBUTO = 0
15190:              THEN BEGIN
15200:
15210:                  COMMENT: NAO FOI IDENTIFICADO ATRIBUTO;
15220:
15230:                  WRITE(VIDEO,"A1>>BUZYNA");
15240:                  WRITE(VIDEO [STOP],<"QUAL O ATRIBUTO? : ">);
15250:                  PEGAPALAVRA;
15260:                  IF PALAVRA=EMPTY THEN ERPO(46) ELSE
15270:                  IF PESQUISADIRETORIO THEN
15280:                      IF CLASSE NEQ ATRIBUTO THEN ERPO(12)
15290:                      ELSE BEGIN
15300:                          NOMEATRIBUTO:=NOME;
15310:                          CATRIBUTO:=L;

```

```

153200      END
153300      ELSE INSEREPALAVRACLASSE(ATRIBUTO);
153400      END;
153500
153600      IF OBJETO = 0 AND NOT ERROIDENTIFICADO
153700      THEN BEGIN
153800          WRITE(VIDEO, <A1>, RUZINA);
153900          WRITE(VIDEO, STOP, <"QUAL O OBJETO? : ">);
154000          PEGAPALAVRA;
154100          IF PALAVRA=EMPTY THEN ERRO(46) ELSE
154200          IF PESQUISADIRETORIO THEN
154300              IF CLASSE NEQ OBJETO THEN ERRO(12)
154400              ELSE BEGIN
154500                  COBJETO:=L;
154600                  NOMEOBJETO:=NOME;
154700                  END
154800              ELSE INSEREPALAVRACLASSE(OBJETO);
154900          END;
155000
155100      COMMENT: NO CASO DE TER SIDO IDENTIFICADO <VALOR> A FUNCAO
155200      SERA CONSIDERADA COMO SENDO 'IGUAL';
155300
155400      IF CFUNCAO = 0 AND NOT ERROIDENTIFICADO THEN BEGIN
155500
155600          IF EXISTEVALOR THEN BEGIN
155700              CFUNCAO:=IGUAL;
155800              NOMEFUNCAO:="IGUAL";
155900              END
156000          ELSE IF NOT REGRAESPECIAL THEN BEGIN
156100              WRITE(VIDEO, STOP, <"QUAL E' A FUNCAO? : ">);
156200              PEGAPALAVRA;
156300              IF PALAVRA=EMPTY THEN ERRO(46) ELSE
156400              IF PESQUISADIRETORIO THEN
156500                  IF CLASSE NEQ FUNCAO THEN ERRO(12)
156600                  ELSE BEGIN
156700                      CFUNCAO:=L;
156800                      NOMEFUNCAO:=NOME;
156900                      END
157000                  ELSE INSEREPALAVRACLASSE(FUNCAO);
157100              END;
157200          IF CMDFUNCAO NEQ 0 THEN VERIFICAANTONIMODEFUNCAO;
157300          END;
157400
157500      COMMENT: VERIFICAR SE <ATRIBUTO>, <VALOR> E <UNIDADE>
157600      SAO COMPATIVETS;
157700
157800      READ(DIPDISK[CFUNCAO], 10, DIRETORIO);
157900      IF NOT ERROIDENTIFICADO
158000      THEN IF SINONIMO = MOSTRAR
158100          THEN IF PREMISA THEN ERRO(11)
158200          ELSE BEGIN
158300              COMMENT : JA TEM ATRIBUTO E VALOR, E COLOCA O
158400              LINK PARA O TEXTO NO CAMPO DO VALOR ;
158500              ARMAZENATEXTO;
158600              CVALOR:=ETEXTO;
158700              END
158800          ELSE BEGIN
158900              READ(DIPDISK[ATRIBUTO], 10, DIRETORIO);
159000              IF ASSOC = 0 THEN

```

```

159100 IF EXISTEVALOR THEN ERRO(5)
159200 ELSE NOMEVALOR:=EMPTY
159300 ELSE
159400 IF ASSOC = 2 OR ASSOC = 1 THEN
159500 IF CUNIDADE NEQ 0 THEN ERRO(6)
159600 ELSE
159700 IF NOT EXISTEVALOR THEN PERGUNTARPELOVALOR
159800 ELSE
159900 ELSE IF CUNIDADE NEQ ASSOC THEN BEGIN
160000 REAL AUX;
160100 AUX:=ASSOC;
160200 RE,DC(DIRDISK(CUNIDADE),10,DIRETORIO);
160300 IF AUX NEQ SUBMULTIPLO THEN
160400
160500 COMMENT: ATRIBUTO PODE REQUERER <VALOR><UNIDADE> OU APENAS
160600 <VALOR>;
160700
160800 IF CUNIDADE NEQ 0 THEN ERRO(7)
160900 ELSE BEGIN
161000 WRITE(VIDEO,"DESEJA ENTRAR COM A UNIDADE?");
161100 PEGAPALAVRA;
161200 IF PALAVRA NEQ "NAO" AND PALAVRA NEQ "N" THEN REGTN
161300 IF PALAVRA = "SIM" OR PALAVRA = "S" THEN BEGIN
161400 WRITE(VIDEO,"ENTRE COM A UNIDADE: ");
161500 PEGAPALAVRA;
161600 END;
161700 IF PALAVRA NEQ EMPTY THEN
161800 IF PESQUISADIRETORIO THEN
161900 IF CLASSE NEQ UNIDADE THEN ERRO(12)
162000 ELSE BEGIN
162100 CUNIDADE:=L;
162200 NOMEUNIDADE:=NOME;
162300 END
162400 ELSE INSEREPAVRACLASSE(UNIDADE);
162500 END;
162600 END;
162700 IF NOT ERROIDENTIFICADO THEN
162800 IF NOT EXISTEVALOR THEN PERGUNTARPELOVALOR;
162900 END;
163000 END;
163100 END ATUALIZAPEGISTRODECLAUSULA;
163200
163300
163400 COMMENT:
163500 * INICIO DE IDENTIFICAPALAVRAS *
163600
163700 TOMA CADA PALAVRA E VERIFICA SE ELA E' PARAMETRO
163800 DA CLAUSULA. SE FOR, SETA A VARIÁVEL CORRESPONDENTE
163900 SENDO LESPPEZA A PALAVRA. CASO HAJA MODIFICADOR
164000 DE FUNCAO, VERIFICA-SE SE HA' UM ANTONIMO PARA A
164100 FUNCAO. VERIFICA-SE SE TODOS OS PARAMETROS FORAM
164200 ENCONTRADOS E, SE NAO FORAM, ESTES SAO PEDIDOS AO
164300 ESPECIALISTA POR "ATUALIZAPEGISTRODECLAUSULA".
164400 O ARQUIVO DE ESQUELETOS E' ATUALIZADO PERCORRENDO
164500 -SE NOVAMENTE A CLAUSULA. OS PARAMETROS SAO
164600 CODIFICADOS COM SIMBOLOS ESPECIAIS E AS DEMAIS
164700 PALAVRAS PELO NUMERO DE SEU REGISTRO NO ARQUIVO
164800 OUTRASPALAVRAS ;
164900

```

```

16500: ACHIEVALOR:=FALSE;
16510: TEXTUAL:=SKELETON:=TAIL(CLAUSULA," ");
16520: TEXTO:=FALSE;
16530: DO BEGIN
16540:   PALAVRA:=EMPTY;
16550:   CLAUSULA:=TAIL(CLAUSULA," ");
16560:   IF CLAUSULA NEQ EMPTY THEN
16570:     IF TAKE(CLAUSULA,1) = "/" THEN TRATA(CLAUSULA)
16580:     ELSE BEGIN
16590:       PALAVRA:=HEAD(CLAUSULA,NOT " ");
16600:       CLAUSULA:=TAIL(CLAUSULA,NOT " ");
16610:       END;
16620:   IF PALAVRA = EMPTY THEN CLAUSULA:=EMPTY
16630:   ELSE BEGIN
16640:     IF NUMERO(PALAVRA) THEN BEGIN
16650:       VAL:=DECIMAL(PALAVRA);
16660:       IF ACHIEVALOR THEN SUBSTITUI(SKELETON,PALAVRA,EMPTY)
16670:       ELSE SUBSTITUI(SKELETON,PALAVRA,"&7");
16680:       ACHIEVALOR:=TRUE;
16690:       END
16700:     ELSE
16710:       IF PESQUISADIRETORIO THEN
16720:         CASE CLASSE OF BEGIN
16730:           1: IF CATRIBUTO = 0
16740:             THEN BEGIN
16750:               CATRIBUTO:=L;
16760:               NOMEATRIBUTO:=NOME;
16770:               END;
16780:           2: IF OBJETO = 0 THEN BEGIN
16790:             OBJETO:=L;
16800:             NOMEOBJETO:=NOME;
16810:             END;
16820:           3: IF CFUNCAO = 0 THEN
16830:             IF NOT REGRAESPECIAL OR SINGNIND = 1
16840:             THEN BEGIN
16850:               CFUNCAO:=L;
16860:               NOMEFUNCAO:=NOME;
16870:               END
16880:             ELSE ERRO(47);
16890:           4: IF CVALOR = 0
16900:             THEN BEGIN
16910:               CTIPOVALOR:=0;
16920:               CVALOR:=L;
16930:               NOMEVALOR:=NOME;
16940:               END;
16950:           5: IF CUNIDADE NEQ 0 THEN
16960:             IF SUBMULTIPLO NEQ UNIDADECOMPATIVEL THEN
16970:               ERRO(7)
16980:             ELSE
16990:               IF ACHIEVALOR THEN BEGIN
17000:                 VAL:=VAL*VALORSUB;
17010:                 READ(DIRDISK(CUNIDADE),10,DIRETORIO);
17020:                 CVALOR:=CVALOR*VALORSUP+VAL;
17030:                 CUNIDADE:=UNIDADECOMPATIVEL;
17040:                 READ(DIRDISK(CUNIDADE),10,DIRETORIO);
17050:                 SUBSTITUI(SKELETON,NOMEUNIDADE,NOME);
17060:                 SUBSTITUI(SKELETON,PALAVRA,EMPTY);
17070:                 SUBSTITUI(SKELETON,"E",EMPTY);
17080:                 NOMEUNIDADE:=NOME;

```



```

17090:                                     END
17100:                                     ELSE
17110:                                     ELSE BEGIN
17120:                                         UNDADECOMPATIVEL:=SUBMULTIPLO;
17130:                                         CUNIDADE:=L;
17140:                                         NOMEUNIDADE:=NOME;
17150:                                         IF ACHEIVALOR THEN CVALOR:=VAL;
17160:                                         END;
17170:                                     6: IF CMODFUNCAO NEQ 0 THEN CMODFUNCAO:=0
17180:                                     ELSE BEGIN
17190:                                         CMODFUNCAO:=L;
17200:                                         NOMEMODIFICADOR:=NOME;
17210:                                         END;
17220:                                     ELSE:
17230:                                     END;
17240:                                     END;
17250:                                     END
17260: UNTIL TAIL(CLAUSULA," ") = EMPTY;
17270: IF ACHEIVALOR THEN BEGIN
17280:     CTIPOVALOR:=1;
17290:     IF CUNIDADE = 0 THEN CVALOR:=VAL;
17300:     END;
17310:
17320: COMMENT: ATUALIZAR O REGISTRO DE CONTEXTO;
17330:
17340: IF NOT ERROIDENTIFICADO THEN BEGIN
17350:     IF (CMODFUNCAO NEQ 0) AND (CFUNCAO NEQ 0) AND NOT REGRAESPECIAL
17360:     THEN VERIFICAANTONIMODEFUNCAO;
17370:     ATUALIZAREGISTRODECLAUSULA;
17380:
17390: COMMENT: ATUALIZACAO DO ARQUIVO DE ESQUELETOS;
17400:
17410: IF NOT ERROIDENTIFICADO AND NOT TEXTO THEN BEGIN
17420:     STRING CARTAO,PALAVRA;
17430:     ARRAY DESATIVADO[0:29];
17440:     DEFINE PEGAPALAVRA =
17450:     BEGIN
17460:         CARTAO:=TAIL(CARTAO," ");
17470:         IF CARTAO NEQ EMPTY THEN
17480:             IF TAKE(CARTAO,1) = "/" THEN TRATA(CARTAO)
17490:             ELSE BEGIN
17500:                 PALAVRA:=HEAD(CARTAO,NOT " ");
17510:                 CARTAO:=TAIL(CARTAO,NOT " ");
17520:                 END
17530:             ELSE PALAVRA:="?";
17540:             END #;
17550:         AVAIL = DESATIVADO[0] #;
17560:         CARTAO:=SKELETON;
17570:         SKELETON:=EMPTY;
17580:         PEGAPALAVRA;
17590:         WHILE PALAVRA NEQ "?" DO BEGIN
17600:             STRING S;
17610:             IF PALAVRA = NOMEATRIBUTO THEN S:="$1" ELSE
17620:             IF PALAVRA = NOMEOBJETO THEN S:="$2" ELSE
17630:             IF PALAVRA = NOMEFUNCAO THEN S:="$3" ELSE
17640:             IF PALAVRA = NOMEVALOR THEN S:="$4" ELSE
17650:             IF PALAVRA = NOMEUNIDADE THEN S:="$5" ELSE
17660:             IF PALAVRA = NOMEMODIFICADOR THEN S:="$6" ELSE
17670:             IF TAKE(PALAVRA,1) = "s" THEN S:=PALAVRA ELSE

```

```

17680:         S:=STRING(OUTRAPALAVRA(PALAVRA),*);
17690:         SKELETON:=SKELETON CAT " " CAT S;
17700:         PEGAPALAVRA;
17710:         END;
17720:         FILL ESQUELETO[*] WITH 30(" ");
17730:         REPLACE POINTER(ESQUELETO) BY SKELETON;
17740:
17750:         COMMENT: VERIFICAR SE O ESQUELETO JA EXISTE;
17760:
17770:         BEGIN
17780:             BOOLEAN EXISTE;
17790:             INTEGER I;
17800:             EXISTE:=FALSE;
17810:             I:=1;
17820:             WHILE (NOT EXISTE) AND (I <= ESQUELETO.DISK.LASTRECORD)
17830:             DO BEGIN
17840:                 READ(ESQUELETO.DISK[I],30,DESATIVADO);
17850:                 IF POINTER(DESATIVADO) = POINTER(ESQUELETO) FOR 180
17860:                     THEN EXISTE:=TRUE;
17870:                 ELSE I:=I+1;
17880:             END;
17890:
17900:             IF EXISTE THEN CESQUELETO:=I
17910:             ELSE BEGIN
17920:                 READ(ESQUELETO.DISK[0],30,DESATIVADO);
17930:                 IF AVAIL > 0 THEN BEGIN
17940:                     CESQUELETO:=DESATIVADO(AVAIL);
17950:                     AVAIL:=AVAIL-1;
17960:                     WRITE(ESQUELETO.DISK[0],30,DESATIVADO);
17970:                     END
17980:                 ELSE CESQUELETO:=ESQUELETO.DISK.LASTRECORD+1;
17990:                 WRITE(ESQUELETO.DISK[CESQUELETO],30,CESQUELETO);
18000:                 END;
18010:             END;
18020:         END;
18030:     END;
18040: END IDENTIFICAPALAVRAS;
18050:
18060:
18070: COMMENT:
18080:     * INICIO DE RECONHECERCLAUSULA *
18090:
18100: IDENTIFICA AS PALAVRAS DA CLAUSULA. SE A CLAUSULA
18110: FOR DE CONCLUSAO E' PERGUNTADO O FATOR DE CERTEZA
18120: E CASO A CLAUSULA NAO SEJA UM TEXTO E' VERIFICADO
18130: QUATS SAO OS CONTEXTOS QUE SE APLICAM A ELA ;
18140:
18150:
18160: FOR CONTADOR:=1 STEP 1 UNTIL 7 DO ELEMENTO[CONTADOR]:=EMPTY;
18170: FILL CLAUSULACOD[*] WITH 3(0);
18180: ERROIDENTIFICADO:=FALSE;
18190: IDENTIFICAPALAVRAS;
18200: CFATOR:=1.;
18210: IF ENTAD AND NOT ERROIDENTIFICADO AND NOT REGRAESPECIAL THEN BEGIN
18220:     WRITE(VIDEO,"<NUMA ESCALA DE 1 A 10, QUAL A CERTEZA ">
18230:         "QUE DEPOSITA NESSA CONCLUSAO?">);

```

```

18240:      LEPESPOSTA(VALORCERTEZA);
18250:      END;
18260:      IF NOT ERROIDENTIFICADO THEN BEGIN
18270:        STRING NAMEFUNCAO,NAMEVALOR;
18280:        IF CMODFUNCAO = 0 THEN NAMEFUNCAO:=NOMEFUNCAO
18290:        ELSE NAMEFUNCAO:=NOMEMODIFICADOR CAT " " CAT NOMEFUNCAO;
18300:        IF CTIPOVALOR = 0 THEN NAMEVALOR:=NOMEVALOR
18310:        ELSE NAMEVALOR:=STRING(CVALOR,*);
18320:        WRITE(VIDEO,<A1>BUZINA);
18330:        WRITE(VIDEO,<///,"ELEMENTOS IDENTIFICADOS NESTA CLAUSULA:">
18340:          /*"ATRIBUTO ..... ",A*
18350:          /*"OBJETO ..... ",A*
18360:          /*"FUNCAO ..... ",A*
18370:          /*"VALOR ..... ",A*
18380:          /*"UNIDADE ..... ",A*///>
18390:          LENGTH(NOMEATRIBUTO), NO"ATRIBUTO, LENGTH(NOMEOBJETO),
18400:          NO"OBJETO, LENGTH(NAMEFUNCAO), NAMEFUNCAO,
18410:          LENGTH(NAMEVALOR), NAMEVALOR, LENGTH(NOMEUNIDADE),
18420:          NO"UNIDADE);
18430:        WRITE(VIDEO[STOP],<"OK? (S/N)">);
18440:        LERESPUESTA(SIMNAD);
18450:        IF PALAVRA = "NAD" OR PALAVRA = "N" THEN
18460:          ERROIDENTIFICADO:=TRUE;
18470:        END;
18480:      IF ENTAD AND NOT ERROIDENTIFICADO
18490:      THEN BEGIN
18500:        STRING AUX1,AUX2;
18510:        POINTER PONT;
18520:        INTEGER I,J,AUX,NUM,CONTADOR;
18530:        BOOLEAN SAT,ACHOU,PRIMEIRAVEZ;
18540:        ARRAY DIRECTORY[0:9];
18550:        DEFINE
18560:          NAME =HEAD(STRING(POINTER(DIRECTORY[I]),42),NOT "/" ) #;
18570:        IF NOT REGRAESPECIAL THEN BEGIN
18580:          %
18590:          %   PERGUNTAR SE PERGUNTA DE NOVO
18600:          %
18610:          WRITE(VIDEO,<"ESTA CONCLUSAO E' DO TIPO PERGUNTAR "
18620:            "DE NOVO ?">);
18630:          LERESPUESTA(SIMNAD);
18640:          IF PALAVRA = "SIM" OR PALAVRA = "S" THEN CRPERGUNTA:=1;
18650:        END;
18660:        %
18670:        %   PERGUNTAR PELOS TIPOS DE CONTEXTO
18680:        %
18690:        READ(DIRDISK[OBJETO],10,DIRETORIO);
18700:        READ(DIRDISK[SINONIMO],10,DIRETORIO);
18710:        READ(OBJETOETIPOS[CHAVEFREG],4,LISTADETIPOS);
18720:        VETORTIPOS[0]:=0;
18730:        IF NULO = 1
18740:        THEN BEGIN
18750:          NOMECONTEXTOS:=" ";
18760:          TIPOPEGRA(1):=TNULO;
18770:        END
18780:        ELSE BEGIN
18790:          BOOLEAN ARRAY PERTENCE[1:LTOTAL];
18800:          WRITE(VIDEO,<"ESTA REGRA APLICA"SF A : (MAX=6)">);
18810:          FOR I:=1 STEP 1 UNTIL LTOTAL
18820:            DO BEGIN

```

```

18830:         READ(DIRDISK(CONTEXT0(I)),10,DIRTORY);
18840:         WRITE(VIDEO,15," : ",A42>>I,NAME);
18850:         END;
18860:     SAI:=FALSE;
18870:     AUX:=0;
18880:     PRIMEIRAVEZ:=TRUE;
18890:     DO BEGIN
18900:         ERROIDENTIFICADO:=FALSE;
18910:         PEGAPALAVRA;
18920:         IF PALAVRA = EMPTY THEN
18930:             IF PRIMEIRAVEZ THEN ERRO(45)
18940:             ELSE SAI:=TRUE;
18950:         ELSE
18960:             IF NOT NUMERO(PALAVRA) THEN ERRO(41)
18970:             ELSE BEGIN
18980:                 NUM:=DECIMAL(PALAVRA);
18990:                 IF NUM > 0 AND NUM LEU LTOTAL
19000:                 THEN IF NOT PERTENCE(NUM)
19010:                     THEN BEGIN
19020:                         PERTENCE(NUM):=TRUE;
19030:                         AUX:=AUX+1;
19040:                         PRIMEIRAVEZ:=FALSE;
19050:                     END
19060:                     ELSE
19070:                         ELSE ERRO(44)
19080:                         END;
19090:                 IF AUX = 6 THEN SAI:=TRUE;
19100:                 END
19110:                 UNTIL SAI;
19120:                 AUX:=0;
19130:                 FOR I:=1 STEP 1 UNTIL LTOTAL
19140:                 DO IF PERTENCE[I]
19150:                     THEN BEGIN
19160:                         AUX:=AUX+1;
19170:                         TIPOREGRA(AUX):=NROTIPO(I);
19180:                         READ(DIRDISK(CONTEXT0(I)),10,DIRTORY);
19190:                         NOMECONTEXTOS:=NOMECONTEXTOS CAT "*" CAT NAME;
19200:                     END;
19210:                 END;
19220:             END;
19230:         END;
19240:     END RECONHECECLAUSULA;
19250:
19260: -----
19270:
19280:     PROCEDURE DECODIFICAREGRA;
19290:     BEGIN
19300:         INTEGER I;
19310:
19320: -----
19330:
19340:     PROCEDURE DECODIFICARCLAUSULA;
19350:     BEGIN
19360:
19370:     COMMENT:
19380:         * INICIO DE DECODIFICARCLAUSULA *
19390:
19400:         A CLAUSULA E' NUMERADA, SEUS ELEMENTOS PRINCIPAIS
19410:         SAO LIDOS E COLOCADOS NO ARRAY ELEMENTO, E A CLAUSULA

```

```

19420: E' RECDIFICADA NO PROCEDIMENTO DECDNICACLAUSULA F
19430: IMPRESSA ;
19440:
19450: LINHA:=" ";
19460: IF NCLAUSULA = 1 THEN
19470:   IF PREMISSA AND NSUBCLAUSULA = 1 THEN
19480:     IF REGRAESPECIAL THEN WRITE(VIDEO,"TRANSFIRA !")
19490:     ELSE LINHA:="SE : "
19500:   ELSE IF ENTAD THEN
19510:     IF REGRAESPECIAL THEN WRITE(VIDEO,"SE :")
19520:     ELSE LINHA:="ENTAD : ";
19530:
19540: COMMENT: NUMERACAO DAS CLAUSULAS;
19550:
19560: IF DENTRODEU THEN
19570:   IF NSUBCLAUSULA = 1 THEN LINHA:=LINHA CAT STRING(NCLAUSULA,**) CAT
19580:   " (" CAT STRING(NCLAUSULA,**) CAT "." CAT
19590:   STRING(NSUBCLAUSULA,**) CAT " "
19600: ELSE LINHA:=LINHA CAT " " CAT STRING(NCLAUSULA,**) CAT "." CAT
19610:   STRING(NSUBCLAUSULA,**) CAT " "
19620: ELSE LINHA:=LINHA CAT STRING(NCLAUSULA,**) CAT ") ";
19630:
19640: COMMENT: RECUPERA OS ELEMENTOS DA CLAUSULA E DECDNIFICA;
19650:
19660: READ(DIRDISK[EFUNCAO],10,DIRETORIO);
19670: IF C/NLIMO = MOSTRAR THEN BEGIN
19680:   ARRAY CLAUSE[0:169];
19690:   READ(TEXT[DISK[EVALOR],90,CLAUSE]);
19700:   LINHA:=LINHA CAT
19710:   HEAD(STRING(PRINTER(CLAUSE),540),NOT ";");
19720: END
19730: ELSE BEGIN
19740:   READ(DIRDISK[EATRIBUTO],10,DIRETORIO);
19750:   NOMEATRIBUTO:=NOME;
19760:   READ(DIRDISK[FOBJETO],10,DIRETORIO);
19770:   NOMEOBJETO:=NOME;
19780:   READ(DIRDISK[EFUNCAO],10,DIRETORIO);
19790:   NOMEFUNCAO:=NOME;
19800:   IF EMODFUNCAO NEQ 0 THEN BEGIN
19810:     READ(DIRDISK[EMODFUNCAO],10,DIRETORIO);
19820:     NOMEEMODIFICADOR:=NOME;
19830:     END;
19840:   IF ETIPOVALOR = 0 THEN BEGIN
19850:     READ(DIRDISK[EVALOR],10,DIRETORIO);
19860:     NOMEVALOR:=NOME;
19870:     END
19880:   ELSE CVALOR:=EVALOR;
19890:   IF EUNIDADE NEQ 0 THEN BEGIN
19900:     READ(DIRDISK[EUNIDADE],10,DIRETORIO);
19910:     NOMEUNIDADE:=NOME;
19920:     END;
19930:   CMODFUNCAO:=EMODFUNCAO;
19940:   CTIPOVALOR:=ETIPOVALOR;
19950:   CUNIDADE:=EUNIDADE;
19960:   IF NOT REGRAESPECIAL THEN
19970:   IF ENTAD THEN
19980:     IF EFATOR = 1. THEN LINHA:=LINHA CAT "E' DEFINITIVO (1.0) QUE "
19990:     ELSE
20000:     IF EFATOR GEQ .8 THEN

```

```

20010: LINHA:=LINHA CAT "EXISTE FORTE EVIDENCIA (" CAT
20020: STRING(EFATOR,*) CAT ") QUE "
20030: ELSE
20040: IF EFATOR GEQ .4 THEN
20050: LINHA:=LINHA CAT "EXISTE EVIDENCIA (" CAT
20060: STRING(EFATOR,*) CAT ") QUE "
20070: ELSE LINHA:=LINHA CAT "EXISTE FRACA EVIDENCIA (" CAT
20080: STRING(EFATOR,*) CAT ") QUE ";
20090:
20100: DECODIFICACLAUSULA(EESQUELETO);
20110: END;
20120:
20130: COMMENT: DELIMITADORES FINAIS DE CLAUSULA;
20140:
20150: IF DENTRODEOU THEN
20160: IF MARCA(PROXC(I)) = 3 THEN
20170: IF MARCA(PROXL(PROXC(I))) = 1 THEN
20180: LINHA:=LINHA CAT ")"
20190: ELSE LINHA:=LINHA CAT ")" &
20200: ELSE IF DENTRODEOU1
20210: THEN LINHA:=LINHA CAT " OU1"
20220: ELSE LINHA:=LINHA CAT " OU"
20230: ELSE
20240: IF MARCA(PROXC(I)) NEQ 1 AND MARCA(PROXC(I)) NEQ 2
20250: THEN LINHA:=LINHA CAT " &";
20260:
20270: COMMENT: CLAUSULA PRONTA PARA IMPRESSAO;
20280:
20290: TMP:=IMELINHA;
20300: END DECODIFICACLAUSULA;
20310:
20320:
20330: COMMENT:
20340: * INICIO DE DECODIFICAREGRA *
20350:
20360: DECODIFICA CADA CLAUSULA DA PREMISSE E DA CONCLUSAO
20370: DA REGRA CORRENTE. CLAUSULA DO TIPO "OU1" E'
20380: DECODIFICADA COMO CLAUSULA "OU", SOMENTE MUDANDO O
20390: CONECTIVO NO FINAL ;
20400:
20410:
20420: ENTAD:=FALSE;
20430: DENTRODEOU:=FALSE;
20440: DENTRODEOU1:=FALSE;
20450: PREMISSE:=TRUE;
20460: NCLAUSULA:=1;
20470: NSUBCLAUSULA:=1;
20480: WRITE(VIDEO, <A1>, BUZINA);
20490: WRITE(VIDEO, <72("=")>, "/, "REGRA ", J4, A11>, NUNKEODAREGRA,
20500: IF ALTERANDO THEN " MODIFICADA" ELSE " ");
20510: I:=PROXC(EPREMISSE);
20520: WHILE MARCA(I) NEQ 1 DO BEGIN
20530: IF MARCA(I) = 3 THEN BEGIN
20540: DENTRODEOU:=TRUE;
20550: IF OU1(I)=1 THEN DENTRODEOU1:=TRUE;
20560: I:=PROXC(I);
20570: WHILE MARCA(I) NEQ 3 DO BEGIN
20580: DECODIFICACLAUSULA;
20590: I:=PROXC(I);

```

```

20500      NSUBCLAUSULA:=NSUBCLAUSULA+1;
20610      END;
20620      NSUBCLAUSULA:=1;
20630      I:=PROXL(I);
20640      END
20650      ELSE BEGIN
20660          DENTRODEQU:=FALSE;
20670          DENTRODEQU1:=FALSE;
20680          DECODIFICARACLAUSULA;
20690          I:=PROXC(I);
20700      END;
20710      NCLAUSULA:=NCLAUSULA+1;
20720      END;
20730      PREMISSA:=FALSE;
20740      DENTRODEQU:=FALSE;
20750      DENTRODEQU1:=FALSE;
20760      ENTAD:=TRUE;
20770      NCLAUSULA:=1;
20780      NSUBCLAUSULA:=0;
20790      I:=PROXC(ENTAD);
20800      WHILE MARCA(I) NEQ 2 DO BEGIN
20810          DECODIFICARACLAUSULA;
20820          I:=PROXC(I);
20830          NCLAUSULA:=NCLAUSULA+1;
20840      END;
20850      WRITE(VIDEO,<72("=")>);
20860      END DECODIFICAREGRA;

-----

20900
20910      PROCEDURE ARMARENAREGRA;
20920      BEGIN
20930
-----

20940
20950
20960      BODCFAN PROCEDURE REGRACOMPATIVEL;
20970      BEGIN
20980          INTEGER
20990          LOCALIZACAO;
21000
-----

21010
21020
21030      PROCEDURE ATUALIZAREFERENCIAS;
21040      BEGIN
21050          INTEGER I;
21060          ARRAY LISTADEATRIBUTOS[0:75];
21070          OFFILE
21080              DISP          = LISTADEATRIBUTOS[0] #,
21090              NUMATRIB(X) = LISTADEATRIBUTOS[X].[22:12] #;
21100
-----

21110
21120
21130      PROCEDURE TRATAREFERENCIAS;
21140      BEGIN
21150          INTEGER
21160              X,
21170              K,
21180              AUX;

```

```

21190:      AUXCHAVE;
21200:      ENDATRIBUTO;
21210:      STRING AUXNOME;
21220:      BOOLEAN DUPLICATA;
21230:      DEFINE AVAIL = REFERENCIAS[0].[15:8] #;
21240:
21250:  PROCEDURE ATUALIZAPROPRATRIBUTOS;
21260:  BEGIN
21270:    BOOLEAN ERROR;
21280:    INTEGER I,PD;
21290:    STRING LISTAPALAVRAS,LISTA,RESPOSTA,SVALOR,ESPERADO,PERGUNTA,TRADUCAO;
21300:
21310:    DEFINE
21320:      LERRESPOSTA =
21330:        READ(VID=0,<A30>,RESPOSTA);
21340:        RESPOSTA:=HEAD(TAIL(RESPOSTA," ")," "); NOT " " #;
21350:
21360:      PEGAVALOR =
21370:        BEGIN
21380:          STRING PALAVRA;
21390:          DEFINE PEGAPALAVRA =
21400:            LISTA:=TAIL(LISTA," ");
21410:            IF LISTA NEQ EMPTY THEN
21420:              IF TAKE(LISTA,1) = "/" THEN BEGIN
21430:                PALAVRA:=HEAD(DROP(LISTA,1),NOT "/");
21440:                LISTA:=DROP(LISTA,LENGTH(PALAVRA) + 2);
21450:                END
21460:              ELSE BEGIN
21470:                PALAVRA:=HEAD(LISTA,NOT " ");
21480:                LISTA:=TAIL(LISTA,NOT " ");
21490:                END
21500:              ELSE PALAVRA:=EMPTY #;
21510:            PEGAPALAVRA;
21520:            SVALOR:=PALAVRA;
21530:            IF SVALOR NEQ EMPTY THEN
21540:              IF TAKE(SVALOR,1) = "(" THEN
21550:                WHILE DROP(SVALOR,LENGTH(SVALOR)-1) NEQ ")" DO BEGIN
21560:                  PEGAPALAVRA;
21570:                  SVALOR:=SVALOR CAT " " CAT PALAVRA;
21580:                  END
21590:                ELSE
21600:                  IF SVALOR = "UMDOS" THEN BEGIN
21610:                    PEGAPALAVRA;
21620:                    SVALOR:=SVALOR CAT " " CAT PALAVRA;
21630:                    END;
21640:                  END #;
21650:
21660:      LESTRING =
21670:        READ(VID=0,<A66>,CARTAO);
21680:        LISTAPALAVRAS:=EMPTY;
21690:        DO BEGIN
21700:          PEGAPALAVRA;
21710:          IF TAIL(PALAVRA,NOT " ") NEQ EMPTY THEN
21720:            PALAVRA:="/" CAT PALAVRA CAT "/";
21730:            LISTAPALAVRAS:=LISTAPALAVRAS CAT PALAVRA CAT " ";
21740:            END
21750:          UNTIL PALAVRA = EMPTY #;
21760:
21770:      PEGACADEIA =

```



```

21780: BEGIN
21790:   PALAVRA:=HEAD(TAIL (CADEIA," "), NOT " ");
21800:   CADEIA:=TAIL(TAIL (CADEIA," "), NOT " ");
21810: END;
21820:
21830: TRATANUMERO =
21840: BEGIN
21850:   STRING LINE,CADEIA;
21860:   CADETA:=TAKE(SVALOR,1);
21870:   PEGACADEIA;
21880:   IF PALAVRA = "CONSTANTES"
21890:   THEN DO BEGIN
21900:     PEGACADEIA;
21910:     IF NOT NUMERO(PALAVRA)
21920:     THEN IF PALAVRA = "CDM"
21930:     THEN BEGIN
21940:       PEGACADEIA;
21950:       IF PALAVRA NEO ")" THEN ERROR:=TRUE;
21960:     END
21970:   END
21980:   UNTIL ERROR OR PALAVRA = ")"
21990: ELSE BEGIN
22000:   I:=1;
22010:   WHILE I < 4 AND NOT ERROR
22020:   DO BEGIN
22030:     CASE I OF
22040:     BEGIN
22050:       1:IF NUMERO(PALAVRA) THEN LINE:=PALAVRA ELSE ERROR:=TRUE;
22060:       2:IF DECIMAL(LINE) > DECIMAL(PALAVRA) THEN ERROR:=TRUE;
22070:       3:IF PALAVRA="CDM" THEN PEGACADEIA;
22080:       IF PALAVRA NEO ")" THEN ERROR:=TRUE;
22090:     END;
22100:     I:=I+1;
22110:     PEGACADEIA;
22120:   END;
22130: END;
22140: END;
22150:
22160: TPATAVALOR =
22170: BEGIN
22180:   PALAVRA:=SVALOR;
22190:   IF NOT P.SQUISADIRECTORIO THEN LIMPAEGRVA(VLOR);
22200:   ELSE IF CLASSE NEO VALOR THEN ERROR:=TRUE
22210: END;
22220:
22230: TRATAUNDOS =
22240: BEGIN
22250:   STRING NOMECLASSE,LISTAVALORES,AUX;
22260:   BOOLEAN SAI;
22270:   I:=1;
22280:   SAI:=FALSE;
22290:   NOMECLASSE:=HEAD(TAIL(DROP(SVALOR,5)," "),NOT " ");
22300:   WHILE NOT SAI AND I <= CLASSFISK.LASTRECORD
22310:   DO BEGIN
22320:     READ(CLASSEDISK[I],20,SCLASS);
22330:     IF HEAD(TAIL (STRING (SCLASS,120), " "), NOT " ") = NOMECLASSE
22340:     THEN SAI:=TRUE;
22350:     I:=I+1;
22360:   END;

```

```

22370: IF NOT SAI
22380: THEN BEGIN
22390:   WRITE(VIDEO[STOP],<"** ENTRE COM OS VALORES DA CLASSE:"//,">">);
22400:   LISTAVALORES:=EMPTY;
22410:   READ(VIDEO,<AA6>,>CAPTAD);
22420:   DO BEGIN
22430:     PEGAPALAVRA;
22440:     LISTAVALORES:=LISTAVALORES CAT PALAVRA CAT "**";
22450:     TRATAVALOR
22460:   END
22470:   UNTIL ERROR OR PALAVRA = EMPTY;
22480:   AUX:=NONECLASSE CAT "**" CAT LISTAVALORES;
22490:   IF LENGTH(AUX) > 120 THEN ERROR:=TRUE;
22500:   IF NOT ERROR
22510:   THEN BEGIN
22520:     FILL SCLASS[*] WITH 20(" ");
22530:     REPLACE POINTER(SCLASS) BY AUX;
22540:     WRITE(CLASSEDISK[1],20,SCLASS);
22550:   END
22560: END;
22570: END#;
22580:
22590: COMMENT:
22600:   * INICIO DE ATUALIZAPROPATRIBUTOS *
22610:
22620:   ATUALIZA AS PROPRIEDADES DE UM ATRIBUTO.
22630:   PRIMEIRAMENTE O ESPECIALISTA ENTRA COM A LISTA DE
22640:   VALORES ESPERADOS PARA AQUELE ATRIBUTO OBEDECENDO A
22650:   SINTAXE. O VALOR PODE SER UM OU MAIS NOMES, UM
22660:   INTERVALO NUMÉRICO COM OU SEM UNIDADE, UMA LISTA
22670:   DE NÚMEROS, OU PODE PERTENCER A UMA CLASSE.
22680:   CASO UMA DADA CLASSE NÃO EXISTA AINDA, O SISTEMA
22690:   ADQUIRE-A INTERATIVAMENTE COM O ESPECIALISTA.
22700:   DEPOIS O SISTEMA PERGUNTA SE A VARIÁVEL DEVE SER
22710:   REDUZIDA OU NÃO, QUAL A PERGUNTA A SER USADA
22720:   QUANDO O VALOR TIVER QUE SER PERGUNTADO.
22730:   EM SEGUIDA, PERGUNTA PELA TRADUÇÃO DAQUELE ATRIBUTO,
22740:   IE, O NOME COM O QUAL AQUELE ATRIBUTO VAI SER
22750:   REFERENCIADO;
22760:
22770:   WRITE(VIDEO,<"/,"** PROPRIEDADES DE "A*//>,LENGTH(AUXNOME),
22780:     AUXNOME);
22790:   DO BEGIN
22800:     IF A<SQC = 0
22810:     THEN LISTAPALAVRAS:="SIM NÃO S N"
22820:     ELSE DO BEGIN
22830:       WRITE(VIDEO,<"** ATUALIZACAO DE PROPATRIBUTOS"//," SINTAXE PARA "
22840:         "PREENCHIMENTO DO CAMPO 'ESPERADO'(UNIDADE=?)"//,
22850:         "/x10,"ELEMENTO := #<ELEMENTO># //x19,
22860:         "j:= #<NOME># / UMOS #<CLASSE># //x19,"j:= (CONSTANTES "
22870:         "<#<CONSTANTE># [COM] //x19,"j:= (<CONSTANTE> <CONSTANTE>"
22880:         "<# [COM]>)">);
22890:       WRITE(VIDEO,<"** ENTRE COM OS VALORES ESPERADOS P/ O ATRIBUTO",
22900:         " (MAX = 60 CAPAC)">);
22910:       WRITE(VIDEO[STOP],<">">);
22920:       ERROR:=FALSE;
22930:       LESETIG;
22940:       LISTA:=LISTAPALAVRAS;
22950:     DO BEGIN

```

```

22960:      ERROR:=FALSE;
22970:      PEGAVALOR;
22980:      IF LENGTH(SVALOR)>1
22990:      THEN IF TAKE(SVALOR,1) ="("
23000:           THEN TRATANUMERO
23010:           ELSE IF SVALOR = "UMDOS"
23020:                THEN TRATAUMDOS
23030:                ELSE TRATAVALOR
23040:      ELSE TRATAVALOR
23050:      END
23060:      UNTIL SVALOR = EMPTY
23070:      END
23080:      UNTIL NOT ERROR;
23090:      IF LENGTH(LISTAPALAVRAS)>60
23100:      THEN LISTAPALAVRAS:=TAKE(LISTAPALAVRAS,60);
23110:      ESPERADL:=LISTAPALAVRAS;
23120:      DO BEGIN
23130:          ERROR:=FALSE;
23140:          WRITE(VIDEO," O VALOR DO ATRIBUTO DEVA SER PERGUNTADO(P) ",
23150:                "OU DEDUZIDO(D) ?">);
23160:          WRITE(VIDEO[STOP],"<">);
23170:          LEFSPOSTA;
23180:          IF RESPOSTA = "P" THEN PD:=1 ELSE PD:=0;
23190:          WRITE(VIDEO,"ENTRE COM A PERGUNTA A SER FEITA AO USUARIO ",
23200:                "(MAX = 90 CARAC)">);
23210:          WRITE(VIDEO[STOP],"<">);
23220:          LESTRING;
23230:          IF LENGTH(LISTAPALAVRAS) > 90
23240:          THEN LISTAPALAVRAS:=TAKE(LISTAPALAVRAS,90);
23250:          PERGUNTA:=LISTAPALAVRAS;
23260:          END
23270:          UNTIL NOT ERROR;
23280:          WRITE(VIDEO,"** QUAL A TRADUCAO DESTA ATRIBUTO?(MAX=48 CARAC)">);
23290:          WRITE(VIDEO[STOP],"<">);
23300:          LESTRING;
23310:          IF LENGTH(LISTAPALAVRAS) > 48
23320:          THEN LISTAPALAVRAS:=TAKE(LISTAPALAVRAS,48);
23330:          TRADUCAO:=LISTAPALAVRAS;
23340:          FILL PROPRIEDADE[*] WITH 70(" ");
23350:          REPLACE POINTF(PROPRIEDADE) BY ESPERADO ;
23360:          REPLACE POINTF(PROPRIEDADE[30]) BY PERGUNTA ;
23370:          REPLACE POINTF(PROPRIEDADE[62]) BY TRADUCAO ;
23380:          PROPRIEDADE[61]:=PD;
23390:          WRITE(VIDEO,"** PROPRIEDADES DO ATRIBUTO ",A," :"/,"ESPERADO =",
23400:                10A6,2(/,X10,10A6),"/,"PERGUNTA =",10A6,2(/,X10,10A6),"/,X10,
23410:                A6,/"PD = ",11,X,"TRADUCAO =",8A6>>LENGTH(AUXNOME),
23420:                AUXNOME,PROPRIEDADE[*]);
23430:          WRITE(VIDEO[STOP],"/," TUDO OK ?(S/N)">);
23440:          LEFSPOSTA;
23450:          IF RESPOSTA = "N" THEN ERROR:=TRUE
23460:          ELSE WRITE(PROFATRIBUTOS[PROPATRIBUTOS+1],70,PROPRIEDADE);
23470:          END
23480:          UNTIL NOT ERROR ;
23490:      END ATUALIZAPROPATRIBUTO ;
23500:
23510:      COMMENT;
23520:          * INICIO DE TRATAREFERENCIA *
23530:
23540:      O ATRIBUTO E' LIDO EM D1RDISK. SE CHAVEREG = 0

```

```

235500 SIGNIFICA QUE E' A PRIMEIRA VEZ QUE AQUELE
235600 ATRIBUTO E' REFERENCIADO. O ARQUIVO D1RDISK E'
235700 ATUALIZADO E AS PROPRIEDADES DO ATRIBUTO SAO TOMADAS
235800 NO PROCEDIMENTO ATUALIZAFROPATRIBUTOS E O REGISTRO
235900 REFERENTE AO ATRIBUTO E' INICIALIZADO EM ATRIBUTOSREF.
236000 O REGISTRO DE ATRIBUTOSREF E' LIDO (CHAVEREG=CONCLUSAO,
236100 CHAVEREG=1-PPREMISSA). SE O ATRIBUTO JA' ESTIVER NO
236200 REGISTRO ELE NAO E' COLOCADO. CASO CONTRARIO, ELE
236300 E' INSERIDO E OS PONTEIROS SAO ATUALIZADOS.
236400
236500 L:=EATRIBUTO;
236600 READ(D1RDISK(L),10,DIRETORIO);
236700 READ(D1RDISK(SINONIMO),10,DIRETORIO);
236800 AUXCHAVE:=CHAVEREG;
236900 IF CHAVEREG = 0 THEN BEGIN
237000
237100 COMMENT:
237200 ATUALIZAR O CAMPO CHAVE DE ACESSO A ATRIBUTOSREF
237300 LOCAL DA REFERENCIA * LISTA * ENDERECO
237400 -----
237500 1. PREMISA * LOCKHEAD * CHAVEFG=1
237600 2. CONCLUSAO * UPDATEDBY * CHAVEFG
237700
237800 CHAVEREG:=ATRIBUTOSREF.LASTRECOPD+2;
237900 WRITE(D1RDISK(SINONIMO),10,DIRETORIO);
238000 AUXCHAVE:=CHAVEREG;
238100 AUXNOME:=NOME;
238200 %ATUALIZACAO DE PROPATRIBUTOS
238300 ATUALIZAFROPATRIBUTOS;
238400
238500 COMMENT: INICIALIZACAO DO ARQUIVO DE REFERENCIAS;
238600
238700 FOR X:=1 STEP 1 UNTIL 179 DO PLINK(X):=X+1;
238800 RLINK(0):=0;
238900 LLINK(0):=0;
239000 NUMEROREGRA(0):=10000;
239100 AVAIL:=1;
239200 WRITE(ATRIBUTOSREF(AUXCHAVE-1),180,REFERENCIAS);
239300 WRITE(ATRIBUTOSREF(AUXCHAVE),180,REFERENCIAS);
239400 END;
239500 ENDATRIBUTO:=IF LOCALIZACAO = 1 THEN AUXCHAVE-1
239600 ELSE AUXCHAVE;
239700
239800 COMMENT:
239900 VERIFICAR SE O ATRIBUTO APARECE MAIS DE UMA VEZ NUM MESMO
240000 LOCAL (PREMISSA OU CONCLUSAO) DA REGRA. CASO APARECA, ESSA
240100 REGRA JA FAZ PARTE DA LISTA DE REFERENCIAS;
240200
240300 DUPLICATA:=FALSE;
240400 FOR K:=0 STEP 1 WHILE (K < DISP) AND (NOT DUPLICATA) DO
240500 IF NUMATRIB(K) = ENDATRIBUTO THEN DUPLICATA:=TRUE;
240600
240700 COMMENT: ATUALIZAR A LISTA DE REFERENCIAS;
240800
240900 IF NOT DUPLICATA THEN BEGIN
241000 NUMATRIB(K):=ENDATRIBUTO;
241100 DISP:=DISP+1;
241200 READ(ATRIBUTOSREF(ENDATRIBUTO),180,REFERENCIAS);
241300 X:=RLINK(0);

```

```

24140: WHILE NUMERODAREGRA(X) < NUMERODAREGRA DO X:=RLINK(X);
24150: IF AVAIL NEQ 180 THEN BEGIN
24160:   AUX:=AVAIL;
24170:   AVAIL:=RLINK(AVAIL);
24180:   NUMERODAREGRA(AUX):=NUMERODAREGRA;
24190:   NEGACAO(AUX):=EMODFUNCAO;
24200:   RLINK(AUX):=X;
24210:   LLINK(AUX):=LLINK(X);
24220:   RLINK(LLINK(X)):=AUX;
24230:   LLINK(X):=AUX;
24240:   WRITE(ATRIBUTOSREF(ENDATRIBUTO),180,REFERENCIAS);
24250:   END;
24260:   END;
24270: END TRATAREFERENCIAS;
24280:
24290: COMMENT:
24300:   * INICIO DE ATUALIZAREFERENCIAS *
24310:
24320:   PERCORRE A PREMISSE E DEPOIS A CONCLUSAO, TOMANDO O
24330:   ATRIBUTO DE CADA CLAUSULA E COLOCANDO O NUMERO DA
24340:   REGRA NO REGISTRO DO ATRIBUTO. CASO O ATRIBUTO
24350:   AINDA NAO ESTEJA NO ATRIBUTOSREF, ELE E'
24360:   INICIALIZADO E O ARQUIVO PROPATRIBUTOS E' ATUALIZADO ;
24370:
24380:   READ(REGRASDISK(NUMERODAREGRA),150,REGRA);
24390:   DTSP:=1;
24400:   I:=PROXC(EPPREMISSE);
24410:   LOCALIZACAO:=1;
24420:   WHILE MARCA(I) NEQ 1 DO
24430:     IF MARCA(I) = 3 THEN BEGIN
24440:       I:=PROXC(I);
24450:       WHILE MARCA(I) NEQ 3 DO BEGIN
24460:         TRATAREFERENCIAS;
24470:         I:=PROXC(I);
24480:       END;
24490:       I:=PROXL(I);
24500:     END
24510:   ELSE BEGIN
24520:     TRATAREFERENCIAS;
24530:     I:=PROXC(I);
24540:   END;
24550:   I:=PROXC(EENTAO);
24560:   LOCALIZACAO:=2;
24570:   WHILE MARCA(I) NEQ 2 DO BEGIN
24580:     TRATAREFERENCIAS;
24590:     I:=PROXC(I);
24600:   END;
24610: END ATUALIZAREFERENCIAS;
24620:
24630: COMMENT:
24640:   * INICIO DE REGRACOMPATIVEL *
24650:
24660:   ATUALIZA AS REFERENCIAS A ATRIBUTOS E (UM DIA)
24670:   VERIFICARA A EXISTENCIA DE CONFLITOS ;
24680:
24690:   ATUALIZAREFERENCIAS;
24700:   REGRACOMPATIVEL:=TRUE;
24710:
24720: COMMENT: -----

```

```

247300          INSERT VERIFICACAO DE CONFLITOS
247400          -----;
247500
247600  END REGRACOMPATIVEL;
247700
247800  COMMENT:
247900          * INICIO DE ARMAZENAREGRA *
248000
248100  A REGRA E' ARMAZENADA. DEPOIS CHAMA-SE REGRACOMPATIVEL
248200  ONDE O ARQUIVO ATRIBUTOSREF E' ATUALIZADO E ONDE
248300  VERIFICA-SE A EXISTENCIA OU NAO DE INCOMPATIBILIDADES.
248400  SE O ESPECIALISTA DESEJAR VER A REGRA, ELA E'
248500  DECODIFICADA ;
248600
248700  WRITE(REGRASDISK(NUMERO DA REGRA),150,REGRA);
248800  IF REGRACOMPATIVEL THEN BEGIN
248900      WRITE(VIDEO,<A1>,BUZINA);
249000      WRITE(VIDEO,<"REGRA " & J4 & " ADQUIRIDA. DESEJA VER A REGRA?">);
249100      NUMERO DA REGRA);
249200      LERESPONSA(SIMNAO);
249300      IF PALAVRA = "SIM" OR PALAVRA = "S" THEN DECODIFICAREGRA;
249400      END
249500  ELSE BEGIN
249600      WRITE(VIDEO,<A1>,BUZINA);
249700      WRITE(VIDEO,<"REGRA " & J4 & " APRESENTA INCOMPATIBILIDADES",
249800      " COM O ACERVO DE REGRAS"> & NUMERO DA REGRA);
249900      WRITE(VIDEO[STOP],<"DESEJA VER A REGRA?">);
250000      LERESPONSA(SIMNAO);
250100      IF PALAVRA = "SIM" OR PALAVRA = "S" THEN DECODIFICAREGRA;
250200      END;
250300  END ARMAZENAREGRA;
250400
250500  -----
250600
250700  PROCEDURE LISTAREFERENCIA;
250800  BEGIN
250900      INTEGER ENDATRIBUTO;
251000      DEFINE LISTAREFERENCIAS =
251100          IF ENDATRIBUTO GEQ 1 AND
251200          ENDATRIBUTO LEQ ATRIBUTOSREF*LASTRECORD THEN BEGIN
251300              INTEGER I;
251400              READ(ATRIBUTOSREF[ENDATRIBUTO],180,REFERENCIA);
251500              I:=PLINK(0);
251600              LINHA:=EMPTY;
251700              WHILE I NEQ 0 DO BEGIN
251800                  LINHA:=LINHA CAT "REGRA " CAT
251900                  STRING(NUMERO DA REGRA(I),*) CAT " ";
252000                  I:=RLINK(I);
252100              END;
252200              IMPRIMELINHA;
252300          END;
252400      WRITE(VIDEO,<72("=")>) #;
252500
252600  COMMENT:
252700          * INICIO DE LISTAREFERENCIA *
252800
252900  VERIFICA SE O ATRIBUTO ESTA OK E LISTA OS NUMEROS
253000  DAS REGRAS QUE REFERENCIAM ESTE ATRIBUTO, PRIMEIRO
253100  NA PREMISSE E DEPOIS NA CONCLUSAO ;

```

```

253200
253300 WRITE(VIDEO,<2A1>,"LIMPATELA,PUZINA");
253400 WRITE(VIDEO[STOP],<"REFERENCIAS AO ATRIBUTO: ">);
253500 DO BEGIN
253600   PEGAPALAVRA;
253700   IF PALAVRA NEW EMPTY THEN
253800     IF PESQUISADIRETORIO THEN
253900       IF CLASSE NEW ATRIBUTO THEN ERRO(12)
254000       ELSE BEGIN
254100         WRITE(VIDEO,<72("<=">)/,"ATRIBUTO "<A>," NA PREMISA DE:">);
254200         LENGTH(PALAVRA),PALAVRA);
254300         READ(DIPDISK(SINONIMO),10,DIRETORIO);
254400         ENDATRIBUTO:=CHAVEFEG-1;
254500         LISTAREFERENCIAS;
254600         WRITE(VIDEO,<"ATRIBUTO "<A>," NA CONCLUSAO DE:">);
254700         LENGTH(PALAVRA),PALAVRA);
254800         ENDATRIBUTO:=*+1;
254900         LISTAREFERENCIAS;
255000         END
255100       ELSE ERRO(9);
255200     END
255300   UNTIL PALAVRA = EMPTY;
255400   END LISTAREFERENCIAS;
255500
-----
255600
255700
255800 PROCEDURE REGRAATIVA;
255900 BEGIN
256000
256100   COMMENT:
256200     * INICIO DE REGRAATIVA *
256300
256400     ESSE PROCEDIMENTO VERIFICA SE O NUMERO DA REGRA EM QUESTAO
256500     APARECE NA FILHA DE REGRAS DESATIVADAS, CASO EM QUE A REGRA
256600     NAO E ATIVA;
256700
256800   INTEGER X;
256900   BOOLEAN ATIVA;
257000   DEFINE DISPO = DESATIVADA[0] #;
257100
257200   ATIVA:=TRUE;
257300   FOR X:=DISPO STEP -1 WHILE (X > 0) AND (ATIVA) DO
257400     IF DESATIVADA[X] = NUMERODAREGRA THEN ATIVA:=FALSE;
257500   REGRAATIVA:=ATIVA;
257600   END REGRAATIVA;
257700
-----
257800
257900
258000 PROCEDURE MOSTRAREGRA;
258100 BEGIN
258200
258300   COMMENT:
258400     * INICIO DE MOSTRAREGRA *
258500
258600     SE A REGRA ESTIVER ATIVA ELA E' LIDA E DECODIFICADA ;
258700
258800   READ(REGRASDISK[0],150,DESATIVADA);
258900   IF NUMERODAREGRA > 0 AND
259000     NUMERODAREGRA LEQ REGRASDISK.LASTRECORD AND REGRAATIVA

```



```

25910 THEN BEGIN
25920 READ(REFRASDISK(NUMERODAREGRA),150,REGRA);
25930 IF EFESPECIAL = 1 THEN REGRAESPECIAL:=TRUE;
25940 DECODIFICAREGRA
25950 END
25960 ELSE ERRO(18);
25970 END MOSTRAREGRA;

-----
26000
26010 PROCEDURE EXCLUIREGRA;
26020 BEGIN
26030
26040 ARRAY REGRACORRENTE(0:149);
26050 BOOLEAN TEXTO;
26060 INTEGER I;
26070 DEFINE
26080 MOSTRAFEGRAEXCLUIDA =
26090 BEGIN
26100 WRITE(VIDEO,<A1>BUZINA);
26110 WRITE(VIDEO,</>,"REGRA ",JA," EXCLUIDA. ",
26120 "DESEJA VER A REGRA?">NUMERODAREGRA);
26130 LERESPONSA(SIMNAO);
26140 IF PALAVRA = "SIM" OR PALAVRA = "S" THEN DECODIFICAREGRA;
26150 END #;
26160
-----
26170
26180
26190 PROCEDURE DEVOLVEAREA;
26200 BEGIN
26210
26220 COMMENT:
26230 * INICIO DE DEVOLVEAREA *
26240
26250 SE A VARIÁVEL "TEXTO" FOR VERDADEIRA, A ÁREA A SER
26260 DEVOLVIDA SERÁ A DO ARQUIVO TEXTODISK, SENÃO A DO
26270 ARQUIVO REGRASDISK. EM AMBOS OS CASOS O REGISTRO F'
26280 LIDO É O NÚMERO DO REGISTRO VAI PARA A PILHA DE
26290 AVATL ;
26300
26310 IF TEXTO THEN BEGIN
26320 ARRAY TEXT(0:89);
26330 DEFINE DISPO = TEXT(0) #;
26340 READ(TEXTODISK(0),90,TEXT);
26350 DISPO:=DISPO+1;
26360 IF DISPO NEQ 90 THEN BEGIN
26370 TEXT[DISPO]:=L;
26380 WRITE(TEXTODISK(0),90,TEXT);
26390 TEXTO:=FALSE;
26400 END;
26410 END
26420 ELSE BEGIN
26430 ARRAY REGRALIMPA(0:149);
26440 DEFINE DISPO = DESATIVADA(0) #;
26450 FILL REGRALIMPA[*] WITH 150(0);
26460 DISPO:=DISPO+1;
26470 IF DISPO NEQ 150 THEN BEGIN
26480 DESATIVADA[DISPO]:=NUMERODAREGRA;
26490 WRITE(REFRASDISK(NUMERODAREGRA),150,REGRALIMPA);

```



```

26500:      WRITE(PEGRASDISK(0),150,DESATIVADA);
26510:      END;
26520:    END;
26530:  END DEVOLVEAREA;
26540:
26550:  -----
26560:
26570:  PROCEDURE ATUALIZAREFERENCIAS;
26580:  BEGIN
26590:    INTEGER I;
26600:    LOCALIZACAO;
26610:
26620:  -----
26630:
26640:  PROCEDURE TRATAREFERENCIA;
26650:  BEGIN
26660:    INTEGER X=ENDATRIBUTO;
26670:
26680:  COMMENT:
26690:    * INICIO DE TRATAREFERENCIA *
26700:
26710:    LE O ATRIBUTO DA CLAUSULA EM DIRDISK E TOMA O
26720:    PONTEIRO PARA ATRIBUTOSREF. A SEGUIR LE-SE EM
26730:    ATRIBUTOSREF O REGISTRO APONTADO SE A CLAUSULA FOR
26740:    DA CONCLUSAO, SENAO LE-SE O REGISTRO ANTERIOR.
26750:    PROCURA-SE NO REGISTRO O NUMERO DA REGRA. QUANDO
26760:    ENCONTRA, ATUALIZA OS PONTEIROS E COLOCA AQUELA
26770:    PALAVRA NO AVAIL. SE A FUNCAO FOR "MOSTRAR" A
26780:    AREA DO TEXTO E DEVOLVIDA;
26790:
26800:  COMMENT: ATUALIZAR ATRIBUTOSREF;
26810:
26820:  L:=FATRIBUTO;
26830:  READ(DIRDISK(L),10,DIRETORIO);
26840:  READ(DIPDISK(SINONIMO),10,DIRETORIO);
26850:  ENDATRIBUTO:=IF LOCALIZACAO = 1 THEN CHAVEREQ=1 ELSE CHAVFPEG;
26860:  READ(ATRIBUTOSREF(ENDATRIBUTO),180,REFERENCIAS);
26870:  Y:=RLINK(X);
26880:  WHILE X NEQ 0 DO
26890:    IF NUMERDAREGRA(X) = NUMERDAREGRA THEN BEGIN
26900:      OFFINE AVAIL = REFERENCIAS(0).L15:81;
26910:      RLINK(LLINK(X)):=FLINK(X);
26920:      LLINK(PLINK(X)):=LLINK(X);
26930:      RLINK(X):=AVAIL;
26940:      AVAIL:=X;
26950:      X:=RLINK(LLINK(X));
26960:    END
26970:    ELSE IF NUMERDAREGRA(X) < NUMERDAREGRA THEN X:=RLINK(X)
26980:    ELSE X:=0;
26990:  WRITE(ATRIBUTOSREF(ENDATRIBUTO),180,REFERENCIAS);
27000:  READ(DIRDISK(FUNCAO),10,DIRETORIO);
27010:  IF SINONIMO = MOSTRAR AND NOT ALTERANDO THEN BEGIN
27020:    L:=EVALOR;
27030:    TEXTO:=TRUE;
27040:    DEVOLVEAREA;
27050:  END;
27060:
27070:  END TRATAREFERENCIA;
27080:

```

```
27680:      ARRAY ORGANIZADA[0:149];
27690:
27700:      COMMENT:
27710:          * INICIO DE REORGANIZAREGRA *
27720:
27730:      ESTE PROCEDIMIENTO REORGANIZA AS CLAUSULAS DENTRO
27740:      DA REGRA CORRENTE ;
27750:
27760:      I:=PROXC(EPREMISSA);
27770:      EP_FMI-SA:=1;
27780:      PROXC(EPREMISSA):=2;
27790:      AVAIL:=1;
27800:      ORGANIZADA[AVAIL]:=REGRA[1];
27810:      AVAIL:=**+1;
27820:      WHILE MARCA(I) NEQ 1 DO
27830:          IF MARCA(I) = 3 THEN BEGIN
27840:              AUX:=I;
27850:              INICIODEEU:=AVAIL;
27860:              I:=PROXC(I);
27870:              PROXC(AUX):=AVAIL+1;
27880:              AVAIL:=**+1;
27890:              WHILE MARCA(I) NEQ 3 DO BEGIN
27900:                  AUX:=I;
27910:                  I:=PROXC(I);
27920:                  PROXC(AUX):=IF MARCA(I) = 3 THEN INICIODEEU ELSE AVAIL+3;
27930:                  ORGANIZADA[AVAIL]:=REGRA[AUX];
27940:                  ORGANIZADA[AVAIL+1]:=REGRA[AUX+1];
27950:                  ORGANIZADA[AVAIL+2]:=REGRA[AUX+2];
27960:                  AVAIL:=**+3;
27970:                  END;
27980:              AUX:=I;
27990:              I:=PROXC(I);
28000:              PROXC(AUX):=IF MARCA(I) = 1 THEN EPREMISSA ELSE AVAIL;
28010:              ORGANIZADA[INICIODEEU]:=REGRA[AUX];
28020:              END
28030:          ELSE BEGIN
28040:              AUX:=I;
28050:              I:=PROXC(I);
28060:              PROXC(AUX):=IF MARCA(I) = 1 THEN EPREMISSA ELSE AVAIL+3;
28070:              ORGANIZADA[AVAIL]:=REGRA[AUX];
28080:              ORGANIZADA[AVAIL+1]:=REGRA[AUX+1];
28090:              ORGANIZADA[AVAIL+2]:=REGRA[AUX+2];
28100:              AVAIL:=**+3;
28110:              END;
28120:      IF FENIAO;
28130:      EENTAD:=AVAIL;
28140:      AUX:=I;
28150:      I:=PROXC(I);
28160:      PROXC(AUX):=AVAIL+1;
28170:      ORGANIZADA[AVAIL]:=REGRA[AUX];
28180:      AVAIL:=**+1;
28190:      WHILE MARCA(I) NEQ 2 DO BEGIN
28200:          AUX:=I;
28210:          I:=PROXC(I);
28220:          PROXC(AUX):=IF MARCA(I) = 2 THEN EENTAD ELSE AVAIL+5;
28230:          ORGANIZADA[AVAIL]:=REGRA[AUX];
28240:          ORGANIZADA[AVAIL+1]:=REGRA[AUX+1];
28250:          ORGANIZADA[AVAIL+2]:=REGRA[AUX+2];
28260:          ORGANIZADA[AVAIL+3]:=REGRA[AUX+3];
```

```

27090: COMMENT:
27100:      * INICIO DE ATUALIZAREFERENCIAS *
27110:
27120:      LE A REGRA, PERCORRE CADA CLAUSULA DA PREMISSE
27130:      E DEPOIS CADA UMA DAS CONCLUSOES ATUALIZANDO AS
27140:      REFERENCIAS ;
27150:
27160:      READ(REGRASDISK(NUMERODAREGRA),150,REGRA);
27170:      I:=PROXC(EPREMISSE);
27180:      LOCALIZACAO:=1;
27190:      WHILE MARCA(I) NEQ 1 DO
27200:      IF MARCA(I) = 3 THEN BEGIN
27210:          I:=PROXC(I);
27220:          WHILE MARCA(I) NEQ 3 DO BEGIN
27230:              TRATAREFERENCIA;
27240:              I:=PROXC(I);
27250:          END;
27260:          I:=PROXL(I);
27270:      END
27280:      ELSE BEGIN
27290:          TRATAREFERENCIA;
27300:          I:=PROXC(I);
27310:      END;
27320:      I:=PROXC(EENTAO);
27330:      LOCALIZACAO:=2;
27340:      WHILE MARCA(I) NEQ 2 DO BEGIN
27350:          TRATAREFERENCIA;
27360:          I:=PROXC(I);
27370:      END;
27380:  END ATUALIZAREFERENCIAS;
27390:
27400: COMMENT:
27410:      * INICIO DE EXCLUIREGRA *
27420:
27430:      VERIFICA SE A REGRA A SER EXCLUIDA ESTA ATIVA.
27440:      DAI PARA TODOS OS ATRIBUTOS REFERENCIADOS NA REGRA
27450:      E' DETIPADA A MENCAO DA REGRA EM ATRIBUTOSPEF. A
27460:      REGRA E' MOSTRADA AO ESPECIALISTA, DEPOIS E' LIMPA
27470:      E O SEU NUMERO E' COLOCADO NA PILHA DE DESATIVADAS;
27480:
27490:      FOR I:=0 STEP 1 UNTIL 149 DO REGRACORRENTE[I]:=REGRA[I];
27500:      IF REGRAATIVA THEN
27510:          IF NUMERODAREGRA > REGRASDISK.LASTRECORD THEN ERRO(18)
27520:      ELSE BEGIN
27530:          ATUALIZAREFERENCIAS;
27540:          IF NOT ALTERANDO THEN MOSTPAREGRAEXCLUIDA;
27550:          DEVOLVEAREA;
27560:      END
27570:      ELSE ERRO(18);
27580:      FOR I:=0 STEP 1 UNTIL 149 DO REGRA[I]:=REGRACORRENTE[I];
27590:  END EXCLUIREGRA;
27600:
27610: -----
27620:
27630: PROCEDURE REDORGANIZAREGRA;
27640: BEGIN
27650:     INTEGER I;
27660:     AUX;
27670:     INICIODEBOL;

```

```

276800  ARRAY ORGANIZADA[0:149]
276900
277000  COMMENT:
277100      * INICIO DE REORGANIZAREGRA *
277200
277300  ESTE PROCEDIMENTO REORGANIZA AS CLAUSULAS DENTRO
277400  DA REGRA CORRENTE ;
277500
277600  I:=PROXC(EPREMISSA);
277700  EPREMISSA:=1;
277800  PROXC(EPREMISSA):=2;
277900  AVAIL:=1;
278000  ORGANIZADA[AVAIL]:=REGRA[1];
278100  AVAIL:=**+1;
278200  WHILE MARCA(I) NEO 1 DO
278300      IF MARCA(I) = 3 THEN BEGIN
278400          AUX:=I;
278500          INICIODEOU:=AVAIL;
278600          I:=PROXC(I);
278700          PROXC(AUX):=AVAIL+1;
278800          AVAIL:=**+1;
278900          WHILE MARCA(I) NEO 3 DO BEGIN
279000              AUX:=I;
279100              I:=PROXC(I);
279200              PROXC(AUX):=IF MARCA(I) = 3 THEN INICIODEOU ELSE AVAIL+3;
279300              ORGANIZADA[AVAIL]:=REGRA[AUX];
279400              ORGANIZADA[AVAIL+1]:=REGRA[AUX+1];
279500              ORGANIZADA[AVAIL+2]:=REGRA[AUX+2];
279600              AVAIL:=**+3;
279700          END;
279800          AUX:=I;
279900          I:=PROXC(I);
280000          PROXC(AUX):=IF MARCA(I) = 1 THEN EPREMISSA ELSE AVAIL;
280100          ORGANIZADA[INICIODEOU]:=REGRA[AUX];
280200          END
280300      ELSE BEGIN
280400          AUX:=I;
280500          I:=PROXC(I);
280600          PROXC(AUX):=IF MARCA(I) = 1 THEN EPREMISSA ELSE AVAIL+3;
280700          ORGANIZADA[AVAIL]:=REGRA[AUX];
280800          ORGANIZADA[AVAIL+1]:=REGRA[AUX+1];
280900          ORGANIZADA[AVAIL+2]:=REGRA[AUX+2];
281000          AVAIL:=**+3;
281100      END;
281200  I:=FENTAO;
281300  FENTAO:=AVAIL;
281400  AUX:=I;
281500  I:=PROXC(I);
281600  PROXC(AUX):=AVAIL+1;
281700  ORGANIZADA[AVAIL]:=REGRA[AUX];
281800  AVAIL:=**+1;
281900  WHILE MARCA(I) NEO 2 DO BEGIN
282000      AUX:=I;
282100      I:=PROXC(I);
282200      PROXC(AUX):=IF MARCA(I) = 2 THEN EENTAO ELSE AVAIL+5;
282300      ORGANIZADA[AVAIL]:=REGRA[AUX];

```

```

28240      ORGANIZADA(AVAIL+1):=REGRA(AUX+1);
28250      ORGANIZADA(AVAIL+2):=REGRA(AUX+2);
28260      ORGANIZADA(AVAIL+3):=REGRA(AUX+3);
28270      ORGANIZADA(AVAIL+4):=REGRA(AUX+4);
28280      AVAIL:=AVAIL+5;
28290      END;
28300      EULT:=AVAIL;
28310      FOR I:=1 STEP 1 UNTIL 149 DO REGRA(I):=ORGANIZADA(I);
28320      WRITE(VIDEO,<"/,"REGRA "J4," REORGANIZADA. DESEJA VER A REGRA?">;
28330      NUMERODAREGRA);
28340      LERESPONSA(SIMNAO);
28350      IF PALAVRA = "SIM" OR PALAVRA = "S" THEN DECODIFICAREGRA;
28360      WRITE(VIDEO,<72("=")>);
28370      END REORGANIZAREGRA;
28380
-----
28390
28400
28410      PROCEDURE ALTERAREGRA;
28420      BEGIN
28430          BOOLEAN SEMSUB;
28440          LABEL ALTERE,
28450          ACABOU;
28460          INTEGER I,
28470          AUX,
28480          LOCAL,
28490          CLAUSE,
28500          ANTERIOR,
28510          INICIODEQU;
28520          DEFINE INCLUIR           = 1 #,
28530          EXCLUIR           = 2 #,
28540          ALTERAR           = 3 #,
28550          FIM           = 4 #,
28560          NAPREMISSA           = 1 #,
28570          NACONCLUSAO           = 2 #,
28580          ENDREGRA           = REGRADISK*LASTRECORD #,
28590
28600          EXISTEERRO(X)           = BEGIN
28610              ERRO(X);
28620              GO ACABOU;
28630              END #,
28640
28650          MOSTRAMENU           =
28660
28670      BEGIN
28680          WRITE(VIDEO,<"QUE OPERACAO DEVE SER FEITA ?"/>;
28690          " 1) INCLUSAO DE CLAUSULA"/>;
28700          " 2) EXCLUSAO DE CLAUSULA"/>;
28710          " 3) ALTERACAO DE CLAUSULA">);
28720          IF NOT PEGRAESPECIAL THEN WRITE(VIDEO,<
28730          " 4) INCLUSAO DE SUBCLAUSULA"/>;
28740          " 5) EXCLUSAO DE SUBCLAUSULA"/>;
28750          " 6) ALTERACAO DE SUBCLAUSULA"/>;
28760          " 7) FIM">);
28770          ELSE WRITE(VIDEO,<" 4) FIM">);
28780          PGAPALAVRA;
28790          IF NOT NUMERO(PALAVRA) THEN CODIGO:=4 ELSE
28800          IF DECIMAL(PALAVRA) > 7 THEN EXISTEERRO(16);
28810          ELSE IF REGRAESPECIAL AND DECIMAL(PALAVRA) > 4
28820          THEN EXISTEERRO(49)

```

```

288300 ELSE BEGIN
288400 CODIGO:=DECIMAL(PALAVRA)
288500 IF NOT REGRAESPECIAL OR CODIGO < 4 THEN
288600 CASE CODIGO OF BEGIN
288700 1:2:3: WRITE(VIDEO, <"A OPERACAO SERA FEITA NA :"/>
288800 " 1: PREMISSA"/>
288900 " 2: CONCLUSAO"/>);
289000 PEGAPALAVRA;
289100 IF NOT NUMERO(PALAVRA) THEN EXISTEERRO(25) ELSE
289200 IF DECIMAL(PALAVRA) = 1 THEN PREMISSA:=TRUE ELSE
289300 IF DECIMAL(PALAVRA) = 2 THEN ENTAO:=TRUE
289400 ELSE EXISTEERRO(25);
289500 IF NOT ERROIDENTIFICADO AND CODIGO NEQ 1 THEN BEGIN
289600 WRITE(VIDEO, <"QUAL O NUMERO DA CLAUSULA ?"/>);
289700 PEGAPALAVRA;
289800 IF NOT NUMERO(PALAVRA) THEN EXISTEERRO(23)
289900 ELSE NCLAUSULA:=DECIMAL(PALAVRA);
290000 END;
290100 4:5:6: CODIGO:=CODIGO + 3;
290200 NSUBCLAUSULA:=1;
290300 PREMISSA:=TRUE;
290400 WRITE(VIDEO, <"ENTRE COM O NUMERO DA CLAUSULA"/>);
290500 PEGAPALAVRA;
290600 IF NOT NUMERO(PALAVRA) THEN EXISTEERRO(23)
290700 ELSE NCLAUSULA:=DECIMAL(PALAVRA);
290800 IF NOT ERROIDENTIFICADO AND CODIGO = 3 OR CODIGO = 2 THEN BEGIN
290900 WRITE(VIDEO, <"ENTRE COM O NUMERO DA SUBCLAUSULA"/>);
291000 PEGAPALAVRA;
291100 IF NOT NUMERO(PALAVRA) THEN EXISTEERRO(25)
291200 ELSE NSUBCLAUSULA:=DECIMAL(PALAVRA);
291300 END;
291400 7: CODIGO:=4;
291500 END;
291600 END;
291700 END #;

291800
291900 CODIGODAOOPERACAO =
292000 BEGIN
292100 COMMENT ;
292200 SINTAXE PARA ALTERACAO :
292300 EXCLUIR/ALTERAR CLAUSULA * PREMISSA/CONCLUSAO
292400 " " " " * SUBCLAUSULA *
292500 INCLUIR CLAUSULA PREMISSA/CONCLUSAO
292600 INCLUIR CLAUSULA * SUBCLAUSULA
292700 FIM
292800
292900 VERIFICA SE A SINTAXE DO COMANDO QUE O ESPECIALISTA
293000 ENTROU ESTA CORRETA, DEVOLVE O CODIGO E SETA AS
293100 VARIÁVEIS QUE INDICAM SE A ALTERACAO E NA PREMISSA
293200 OU CONCLUSAO, E SE ESTA DENTRO DE CLAUSULA "OU"
293300 OU "E" ;
293400 WRITE(VIDEO[STOP], <"OPERACAO: "/>);
293500 SENSUB:=FALSE;
293600 PEGAPALAVRA;
293700 IF PALAVRA = "INCLUIR" THEN CODIGO:=1 ELSE
293800 IF PALAVRA = "EXCLUIR" THEN CODIGO:=2 ELSE
293900 IF PALAVRA = "ALTERAR" THEN CODIGO:=3 ELSE
294000 IF PALAVRA = "FIM" THEN CODIGO:=4 ELSE
294100 EXISTEERRO(16);

```

```

294200 PEGAPALAVRA;
294300 IF PALAVRA NEQ "CLAUSULA" THEN EXISTEERRO(22);
294400 PEGAPALAVRA;
294500 IF NOT NUMERO(PALAVRA)
294600 THEN IF CODIGO=1 THEN SEMSUB:=TRUE
294700 ELSE EXISTEERRO(23)
294800 ELSE BEGIN
294900     NCLAUSULA:=DECIMAL(PALAVRA);
295000     PEGAPALAVRA;
295100     END;
295200 IF PALAVRA = "SUBCLAUSULA"
295300 THEN IF REGRAESPECIAL THEN EXISTEERRO(48)
295400 ELSE IF SEMSUB THEN EXISTEERRO(25)
295500 ELSE IF CODIGO NEQ 1
295600 THEN BEGIN
295700     PEGAPALAVRA;
295800     IF NUMERO(PALAVRA) THEN NSUBCLAUSULA:=DECIMAL(PALAVRA)
295900     ELSE EXISTEERRO(24);
296000     PREMISSA:=TRUE;
296100     END
296200     ELSE BEGIN PREMISSA:=TRUE; NSUBCLAUSULA:=1; END
296300 ELSE IF PALAVRA = "PREMISSA" THEN PREMISSA:=TRUE
296400 ELSE IF PALAVRA = "CONCLUSAO" THEN ENTAO:=TRUE
296500 ELSE EXISTEERRO(25);
296600 END #;
296700
296800 CLAUSULATIPOE
296900 IF PROXC(EPREMISSA) NEQ INICIODEOU THEN BEGIN
297000     BOOLEAN ACHEI;
297100     ACHEI:=FALSE;
297200     I:=EPREMISSA;
297300     DO
297400         IF MARCA(I) = 3 THEN
297500             IF PROXL(I) = INICIODEOU THEN BEGIN
297600                 ACHEI:=TRUE;
297700                 PROXL(I):=PROXC(INICIODEOU);
297800                 PROXC(PROXC(INICIODEOU)):=PROXL(INICIODEOU);
297900                 END
298000                 ELSE I:=PROXL(I)
298100             ELSE
298200                 IF PROXC(I) = INICIODEOU THEN BEGIN
298300                     ACHEI:=TRUE;
298400                     PROXC(I):=PROXC(INICIODEOU);
298500                     PROXC(PROXC(INICIODEOU)):=PROXL(INICIODEOU);
298600                     END
298700                     ELSE I:=PROXC(I)
298800                 UNTIL ACHEI;
298900             END
299000         ELSE BEGIN
299100             PROXC(EPREMISSA):=PROXC(INICIODEOU);
299200             PROXC(PROXC(INICIODEOU)):=PROXL(INICIODEOU);
299300             END #;
299400
299500 -----
299600
299700 PROCEDURE RECONHECERCLAUSULA;
299800 BEGIN
299900     BOOLEAN RECONHECE;
300000

```



```

300100 COMMENT:
300200 * INICIO DE RECONHECERCLAUSULA *
300300
300400 CONCATENA-SE AS PALAVRAS ATÉ CHEGAR NO "PONTO E
300500 VIRGULA" E RECONHECE-SE A CLAUSULA. ATÉ SE SE QUIZER
300600 INCLUIR UMA NOVA CLAUSULA E NAO HOUVER ESPACO PARA
300700 ELA, A REGRA É REORGANIZADA
300800
300900 CLAUSULA:=EMPTY;
301000 RECONHECE:=FALSE;
301100 FILL CLAUSULACOD(*) WITH 3(0);
301200 WRITE(VIDEO,<A1>BUZINA);
301300 WRITE(VIDEO,<"ENTRE COM A NOVA CLAUSULA (TERMINE COM ' ')">);
301400 DO BEGIN
301500   PEGAPALAVRA;
301600   LOOKUP;
301700   CASE CODELIM OF BEGIN
301800     % .. DELIMITADORES "&" E "OU"
301900     0112: EXISTEERRO(17);
302000     3: EXISTEERRO(17);
302100
302200     4: IF PALAVRA = "]" THEN BEGIN
302300       % .. DELIMITADOR "]"
302400       RECONHECECLAUSULA;
302500       IF ERROIDENTIFICADO THEN GO ACABOU;
302600       IF CODIGO = INCLUIR AND EULT > 148 THEN BEGIN
302700         REORGANIZAREGRA;
302800         IF EULT > 148 THEN EXISTEERRO(21);
302900       END;
303000       AVAIL:=EULT;
303100       RECONHECE:=TRUE;
303200     END
303300     ELSE BEGIN
303400       % .. NAO É UM DELIMITADOR
303500       IF COMBARRA THEN PALAVRA:="/" CAT PALAVRA CAT "/";
303600       CLAUSULA:=CLAUSULA CAT PALAVRA CAT " ";
303700       IF LENGTH(CLAUSULA) > 540 THEN EXISTEERRO(14);
303800     END
303900   END CASE;
304000 END
304100 UNTIL RECONHECE;
304200 END RECONHECERCLAUSULA;
304300
304400 COMMENT:
304500 * INICIO DE ALTERAREGRA *
304600
304700 LE A REGRA. LE-SE O COMANDO DO ESPECIALISTA E
304800 ENQUANTO NAO APARECER O "FIM" SAO FEITAS A INCLUSAO
304900 EXCLUSAO E MODIFICACAO DE CLAUSULAS. AO FINAL DE
305000 CADA COMANDO, A REGRA É DECODIFICADA. DEPOIS DE
305100 FEITAS TODAS AS ALTERACOES A REGRA ANTIGA É
305200 EXCLUIDA DO BANCO, E A NOVA É ARMAZENADA NA
305300 PRIMEIRA DISPONIVEL;
305400
305500 ALTERANDO:=TRUE;
305600 IF EESPECIAL = 1 THEN REGRAESPECIAL:=TRUE;
305700 READ(PEGRASDISK(NUMERODAREGRA),150,PEGRA);
305800 WRITE(VIDEO,<2A1>BUZINA,BUZINA);
305900 UNTIL ALTERANDO DO BEGIN

```



```

306000 NCCLAUSULA:=0;
306100 AVAIL:=EULT;
306200 NSUBCLAUSULA:=0;
306300 ENTAD:=FALSE;
306400 PREMISA:=FALSE;
306500 IF NUMERODAREGRA > ENDREGRA OR NOT REGRAATIVA
306600 THEN EXISTEERRO(18);
306700 ELSE BEGIN
306800     IF MENU THEN MOSTRAMENU ELSE CODIGODADOERACAO;
306900     I:=IF PREMISA THEN EPREMISA ELSE EENTAD;
307000     CLAUSE:=IF NSUBCLAUSULA = 0 THEN
307100         IF CODIGO = ALTERAR THEN NCLAUSULA
307200         ELSE NCLAUSULA-1
307300     ELSE NCLAUSULA;
307400     LOCAL:=IF PREMISA THEN 1 ELSE 2;
307500     DENTRODEOU:=IF NSUBCLAUSULA = 0 THEN FALSE ELSE TRUE;
307600     CASE CODIGO OF BEGIN
307700
307800         INCLUIR:
307900
308000     COMMENT:
308100         A INCLUSAO DE UMA CLAUSULA "E" NA PREMISA OU CONCLUSAO
308200         SAO A MESMA COISA. A CLAUSULA E' RECONHECIDA, COLOCADA
308300         NA REGRA E O DESCRITOR E' ALTERADO. PARA SE INCLUIR UMA
308400         SUBCLAUSULA VERIFICA-SE SE JA' EXISTE UM DESCRITOR "OU".
308500         A NOVA CLAUSULA SERA' INCLUIDA ANTES DAS OUTRAS, IE, O
308600         DESCRITOR APONTARA' PARA ELA ;
308700
308800         CASE LOCAL OF BEGIN
308900
309000             NAPREMISA:
309100
309200                 IF DENTRODEOU THEN
309300                     IF ELEMENTOS(I) >= CLAUSE THEN BEGIN
309400                         THRU CLAUSE DO BEGIN
309500                             ANTERIOR:=I;
309600                             I:=IF MARCA(I) = 3 THEN PROXL(I)
309700                                 ELSE PROXC(I);
309800                         END;
309900                         RECONHECERACLUSULA;
310000                         IF MARCA(I) = 3 THEN INICIODEOU:=I
310100                         ELSE BEGIN
310200                             % CRIA NOVO DESCRITOR "OU"
310300                             INICIODEOU:=AVAIL;
310400                             AVAIL:=AVAIL + 1;
310500                             EULT:=AVAIL;
310600                             MARCA(INICIODEOU):=3;
310700                             ELEMENTOS(INICIODEOU):=1;
310800                             PROXC(ANTERIOR):=INICIODEOU;
310900                             PROXL(INICIODEOU):=PROXC(I);
311000                             PROXC(INICIODEOU):=I;
311100                             PROXC(I):=INICIODEOU;
311200                         END;
311300                         REGRA(AVAIL):=CLAUSULACOL[0];
311400                         REGRA(AVAIL+1):=CLAUSULACOL[1];
311500                         REGRA(AVAIL+2):=CVALOR;
311600                         PROXC(AVAIL):=PROXC(INICIODEOU);
311700                         PROXC(INICIODEOU):=AVAIL;
311800                         EULT:=AVAIL+3;

```

```

31190      ELEMENTOS(INICIODEOU):=
31200      ELEMENTOS(INICIODEOU)+1;
31210      END
31220      ELSE EXISTEERRO(19)
31230      ELSE BEGIN
31240          RECONHECERCLAUSULA;
31250          REGRA(AVAIL):=CLAUSULACOD(0);
31260          REGRA(AVAIL+1):=CLAUSULACOD(1);
31270          REGRA(AVAIL+2):=CVALOR;
31280          PROXC(AVAIL):=PROXC(EPREMISSA);
31290          PROXC(EPREMISSA):=AVAIL;
31300          EULT:=AVAIL+3;
31310          ELEMENTOS(EPREMISSA):=ELEMENTOS(EPREMISSA)+1;
31320      END;
31330
31340      MACONCLUSAO:
31350
31360      BEGIN
31370          RECONHECERCLAUSULA;
31380          REGRA(AVAIL):=CLAUSULACOD(0);
31390          REGRA(AVAIL+1):=CLAUSULACOD(1);
31400          REGRA(AVAIL+2):=CVALOR;
31410          REGRA(AVAIL+3):=CFATOR;
31420          REGRA(AVAIL+4):=VETORTIPOS(0);
31430          PROXC(AVAIL):=PROXC(EENTAO);
31440          PROXC(EENTAO):=AVAIL;
31450          ELEMENTOS(EENTAO):=ELEMENTOS(EENTAO)+1;
31460          EULT:=AVAIL+5;
31470      END
31480      END INCLUIR;
31490
31500      EXCLUIR:
31510
31520      COMMENT:
31530      PARA EXCLUIR UMA CLAUSULA "E" DA PREMISSE OU CONCLUSAO
31540      VERIFICA-SE SE A CLAUSULA E' UNICA. SE FOR DA' ERRO,
31550      SENAO OS PONTEIROS E O DESCRITOR SAO ATUALIZADOS. NO
31560      CASO DE SUBCLAUSULA, ELA E' EXCLUIDA, E SE SO' SOBRA
31570      UMA NO DESCRITOR "OU" ESTA ULTIMA E' TRANSFORMADA
31580      EM CLAUSULA "E" E O DESCRITOR "OU" DESAPARECE ;
31590
31600      CASE LOCAL OF BEGIN
31610
31620      NAPREMISSE:
31630
31640      IF DENTRODEOU THEN
31650          IF ELEMENTOS(I) GEO CLAUSE THEN BEGIN
31660              THRU CLAUSE DO I:=IF MARCA(I) = 3 THEN PROXL(I)
31670                  ELSE PROXC(I);
31680
31690              IF MARCA(I) = 3 THEN
31700                  IF ELEMENTOS(I) = 1 AND ELEMENTOS(EPREMISSA) = 1
31710                  THEN EXISTEERRO(30) ELSE
31720                      IF ELEMENTOS(I) GEO NSUBCLAUSULA THEN BEGIN
31730                          INICIODEOU:=I;
31740                          THRU NSUBCLAUSULA-1 DO I:=PROXC(I);
31750                          AUX:=PROXC(PROXC(I));
31760                          PROXC(I):=AUX;
31770                          ELEMENTOS(INICIODEOU):=

```

```

317800 IF ELEMENTOS(INICIODEOU) = 1 THEN CLAUSULATIPOE
317900 ELSE IF ELEMENTOS(INICIODEOU) = 0
318000 THEN BEGIN
318100 THRU NCLAUSULA = 1 DO
318200 I:=IF MARCA(I)=3 THEN PROXL(INICIODEOU)
318300 ELSE PROXC(INICIODEOU);
318400 IF MARCA(I)=3
318500 THEN PROXL(I):=PROXL(INICIODEOU)
318600 ELSE PROXC(I):=PROXL(INICIODEOU);
318700 END;
318800 END
318900 ELSE EXISTEERRO(19)
319000 ELSE EXISTEERRO(19)
319100 END
319200 ELSE EXISTEERRO(19)
319300 ELSE
319400 IF ELEMENTOS(I) GTR CLAUSE THEN
319500 IF ELEMENTOS(I) GTR 1 THEN BEGIN
319600 INTEGER ADDRESS;
319700 THRU CLAUSE DO I:=IF MARCA(I) = 3 THEN PROXL(I)
319800 ELSE PROXC(I);
319900 AUX:=IF MARCA(I) = 3 THEN PROXL(I) ELSE PROXC(I);
320000 ADDRESS:=IF MARCA(AUX) = 3 THEN PROXL(AUX)
320100 ELSE PROXC(AUX);
320200 IF MARCA(I) = 3 THEN PROXL(I):=ADDRESS
320300 ELSE PROXC(I):=ADDRESS;
320400 ELEMENTOS(EPREMISSA):=ELEMENTOS(EPREMISSA)-1;
320500 END
320600 ELSE EXISTEERRO(30)
320700 ELSE EXISTEERRO(20);
320800
320900 NACONCLUSAO:
321000
321100 IF ELEMENTOS(I) GTR CLAUSE THEN
321200 IF ELEMENTOS(I) GTR 1 THEN BEGIN
321300 THRU CLAUSE DO I:=PROXC(I);
321400 AUX:=PROXC(PROXC(I));
321500 PROXC(I):=AUX;
321600 ELEMENTOS(EENTAO):=ELEMENTOS(EENTAO)-1;
321700 END
321800 ELSE EXISTEERRO(30)
321900 ELSE EXISTEERRO(20);
322000 END EXCLUIR;
322100
322200 ALTERAR:
322300
322400 CASE LOCAL OF BEGIN
322500
322600 COMMENT:
322700 A NOVA CLAUSULA E' RECONHECIDA E ESCRITA EM CIMA
322800 DA ANTIGA ;
322900
323000 NAPREMISSA:
323100
323200 IF DENTRODEOU THEN
323300 IF ELEMENTOS(I) GEO CLAUSE THEN BEGIN
323400 THRU CLAUSE DO I:=IF MARCA(I) = 3 THEN PROXL(I)
323500 ELSE PROXC(I);
323600 IF MARCA(I) = 3 THEN

```

```

323700      IF ELEMENTOS(I) GEQ NSUBCLAUSULA THEN BEGIN
323800          THRU NSUBCLAUSULA DO I:=PROXC(I);
323900          AUX:=PROXC(I);
324000          RECONHECERACLAUSULA;
324100          REGRA(I):=CLAUSULACOD(0);
324200          REGRA(I+1):=CLAUSULACOD(1);
324300          REGRA(I+2):=CVALOR;
324400          PROXC(I):=AUX;
324500          END
324600      ELSE EXISTEERRO(19)
324700      ELSE EXISTEERRO(19);
324800      END
324900      ELSE EXISTEERRO(19)
325000  ELSE
325100      IF ELEMENTOS(I) GEQ CLAUSE THEN BEGIN
325200          THRU CLAUSE DO I:=IF MARCA(I) = 3 THEN PROXL(I)
325300                                  ELSE PROXC(I);
325400          AUX:=PROXC(I);
325500          RECONHECERACLAUSULA;
325600          REGRA(I):=CLAUSULACOD(0);
325700          REGRA(I+1):=CLAUSULACOD(1);
325800          REGRA(I+2):=CVALOR;
325900          PROXC(I):=AUX;
326000          END
326100      ELSE EXISTEERRO(20);
326200
326300  MACONCLUSAO:
326400
326500      IF ELEMENTOS(I) GEQ CLAUSE THEN BEGIN
326600          THRU CLAUSE DO I:=PROXC(I);
326700          AUX:=PROXC(I);
326800          RECONHECERACLAUSULA;
326900          REGRA(I):=CLAUSULACOD(0);
327000          REGRA(I+1):=CLAUSULACOD(1);
327100          REGRA(I+2):=CVALOR;
327200          REGRA(I+3):=CFATOR;
327300          REGRA(I+4):=VETORTIPOS(0);
327400          PROXC(I):=AUX;
327500          END
327600      ELSE EXISTEERRO(20);
327700      END ALTERAR;
327800
327900      FIM: ALTERANDO:=FALSE;
328000
328100      END CASE CODIGO;
328200      END;
328300  DEBODIFICAREGRA;
328400  ACABOU:
328500      IF ALTERANDO THEN BEGIN
328600          WRITE(VIDEO(STOP), <"MAIS ALGUMA ALTERACAO? (S/N)">);
328700          LERESPOSTA(SIMNAO);
328800          IF PALAVRA = "NAO" OR PALAVRA = "N" THEN ALTERANDO:=FALSE;
328900          END;
329000      IF NOT ALTERANDO THEN BEGIN
329100          EXCLUIREGRA;
329200          IF NOT RFGRASPECIAL THEN BEGIN
329300              WRITE(VIDEO, <"ESTA REGRA DEVE SER MOSTRADA NO ENCADEAMENTO ">
329400                  <"DE 'PORQUE' ?">);
329500              LERESPOSTA(SIMNAO);

```

```

329600 IF PALAVRA = "NAO" OR PALAVRA = "N" THEN EMOSTRAREGRA:=1;
329700 END;
329800 ALTERANDO:=FALSE;
329900 ATIVAREGRA;
330000 ARMAZENAREGRA;
330100 END;
330200 END;
330300 END ALTERAREGRA;
330400
330500 -----
330600
330700 PROCEDURE ADQUIREREGRA;
330800 BEGIN
330900     INTEGER I;
331000     AVAIL;
331100     BOOLEAN FIM;
331200     FIMDEPREMISSA;
331300
331400 -----
331500
331600 PROCEDURE RECONHECECLAUSULAE;
331700 BEGIN
331800
331900     COMMENT:
332000         * INICIO DE RECONHECECLAUSULAE *
332100
332200     RECONHECE A CLAUSULA E ARMAZENA-A SE A CLAUSULA ANTERIOR
332300     FOR "OU" OU "OU1" FAZ DENTRODEOU=FALSE E ATUALIZA OS
332400     LINKS E ELEMENTOS DE DENTRODEOU E EPREMISSA SE FOR
332500     CLAUSULA DA PREMISSA ENTAO ATUALIZA OS ELEMENTOS DE
332600     EPREMISSA E O LINK DA CLAUSULA. SE A CLAUSULA FOR DA
332700     CONCLUSAO ARMAZENA E ATUALIZA ELEMENTOS(EENTAO) E
332800     COLUCA FATOR E TIPODECONTEXTO NAS 2 ULTIMAS PALAVRAS ;
332900
333000 RECONHECECLAUSULA;
333100 IF NOT ERROIDENTIFICADO THEN BEGIN
333200     AVAIL:=AVAIL+1;
333300     REGRA(AVAIL):=CLAUSULACOD[0];
333400     AVAIL:=AVAIL+1;
333500     REGRA(AVAIL):=CLAUSULACOD[1];
333600     AVAIL:=AVAIL+1;
333700     REGRA(AVAIL):=CVALOR;
333800     IF PREMISSA THEN
333900         IF DENTRODEOU THEN BEGIN
334000             DENTRODEOU:=FALSE;
334100             DENTRODEOU1:=FALSE;
334200             NSUBCLAUSULA:=1;
334300             PROXC(AVAIL-2):=INICIODEOU;
334400             PROXL(INICIODEOU):=IF FIMDEPREMISSA THEN EPREMISSA
334500                 ELSE AVAIL+1;
334600             ELEMENTOS(INICIODEOU):=ELEMENTOS(INICIODEOU)+1;
334700             ELEMENTOS(EPREMISSA):=ELEMENTOS(EPREMISSA)+1;
334800             END
334900         ELSE BEGIN
335000             PROXC(AVAIL-2):=IF FIMDEPREMISSA THEN EPREMISSA
335100                 ELSE AVAIL+1;
335200             ELEMENTOS(EPREMISSA):=ELEMENTOS(EPREMISSA)+1;
335300             END
335400         ELSE BEGIN

```

```
335500      PROXC(AVAIL-2):=AVAIL+3;
335600      AVAIL:=AVAIL+1;
335700      REGRA(AVAIL):=CFATOR;
335800      AVAIL:=AVAIL+1;
335900      REGRA(AVAIL):=VETORTIPOS[0];
336000      ELEMENTOS(EENTAO):=ELEMENTOS(EENTAO)+1;
336100      END;
336200      END;
336300  END RECONHECECLAUSULAE;
```

```
336400
336500  *-----*
336600
336700  PROCEDURE RECONHECECLAUSULAOU(TIPODEOU);
336800  INTEGR TIPODEOU;
336900  BEGIN
337000
337100  COMMENT:
337200      * INICIO DE RECONHECECLAUSULAOU *
337300
337400  RECONHECE A CLAUSULA. SE E' A PRIMEIRA CLAUSULA "OU" OU
337500  "OU1", INICIALIZA O DESCRITOR. A CLAUSULA E' ARMazenADA
337600  E AVAIL E' ATUALIZADO ;
337700
337800  RECONHECECLAUSULA;
337900  IF NOT ERROIDENTIFICADO THEN BEGIN
338000      AVAIL:=AVAIL+1;
338100      IF NOT DENTRODEOU THEN BEGIN
338200          DENTRODEOU:=TRUE;
338300          IF TIPODEOU = 2 THEN BEGIN
338400              DENTRODEOU:=TRUE;
338500              OU(AVAIL):=1;
338600          END;
338700          INICIODEOU:=AVAIL;
338800          ELEMENTOS(INICIODEOU):=0;
338900          MARCA(AVAIL):=3;
339000          PROXC(AVAIL):=AVAIL+1;
339100          AVAIL:=AVAIL+1;
339200      END;
339300      REGRA(AVAIL):=CLAUSULACOD[0];
339400      AVAIL:=AVAIL+1;
339500      REGRA(AVAIL):=CLAUSULACOD[1];
339600      AVAIL:=AVAIL+1;
339700      REGRA(AVAIL):=CVALOR;
339800      PROXC(AVAIL-2):=AVAIL+1;
339900      ELEMENTOS(INICIODEOU):=ELEMENTOS(INICIODEOU)+1;
340000      END;
340100  END RECONHECECLAUSULAOU;
```

```
340200
340300  *-----*
340400
340500  PROCEDURE MONTACLAUSULA;
340600  BEGIN
340700
340800  COMMENT:
340900
341000      * INICIO DE MONTACLAUSULA *
341100
341200  CONCATENA AS PALAVRAS DA CLAUSULA ;
341300
```

```

341400      IF COMBARRA THEN PALAVRA:="/" CAT PALAVRA CAT "/"
341500      CLAUSULA:=CLAUSULA CAT PALAVRA CAT " "
341600      IF LENGTH(CLAUSULA) > 540 THEN BEGIN
341700          ERRO(14);
341800          FIM:=TRUE;
341900          END;
342000      END MONTACLAUSULA;
342100
342200      COMMENT:
342300          * INICIO DE ADQUIREREGRA *
342400
342500      PRIMEIRAMENTE DESCOBRE-SE QUAL O NUMERO DA REGRA A SER
342600      ADQUIRIDA (A PRIMEIRA DA PILHA DE DISPONIVEIS, OU A DO
342700      ULTIMO REGISTRO + 1). DEPOIS PERCORRE TODAS AS CLAUSULAS
342800      DA PREMISA E DA CONCLUSAO ANALISANDO-AS. AO FINAL DECODIFICA
342900      A REGRA E SUBMETE-A AO USUARIO. SE APROVADA A REGRA E'
343000      ARMAZENADA. SENAO, CASO O QUEIRA MODIFICA-LA, ELA E' ARMAZENADA
343100      E DEPOIS E' MODIFICADA ;
343200
343300
343400      ATIVAREGRA;
343500      FIL REGRA[*] WITH 150(0);
343600      WRITE(VIDEO,<2A1>,"LIMPITE LA,BUZINA");
343700      WRITE(VIDEO,<"REGRA ",J4>,"NUMERODAREGRA");
343800      IF MENU THEN BEGIN
343900          WRITE(VIDEO,<"ESTA REGRA E' DO TIPO ESPECIAL ?">);
344000          LFRESPSTA(SIMNAO);
344100          IF PALAVRA = "SIM" OR PALAVRA = "S" THEN REGRAESPECIAL:=TRUE;
344200          END;
344300      IF REGRAESPECIAL THEN EESPECIAL:=1;
344400      CLAUSULA:=EMPTY;
344500      EPREMISA:=1;
344600      PREMISA:=TRUE;
344700      FIMDEPREMISA:=FALSE;
344800      ENTAO:=FALSE;
344900      FIM:=FALSE;
345000      AVAIL:=1;
345100      PROXC(EPREMISA):=2;
345200      MARCA(EPREMISA):=1;
345300      ELEMENTOS(EPREMISA):=0;
345400      NCLAUSULA:=1;
345500      NSURCLAUSULA:=1;
345600      WRITE(VIDEO,<A1>,"BUZINA");
345700      WRITE(VIDEO,<"ENTRE COM A CLAUSULA ",J2>,"NCLAUSULA");
345800      LEPCARTAO;
345900      DO BEGIN
346000
346100      COMMENT:
346200      AS PALAVRAS DA CLAUSULA SAO ANALISADAS. CASO NAO SEJA UM
346300      DELIMITADOR ELA E' CONCATENADA 'AS OUTRAS. QUANDO A ULTIMA
346400      CLAUSULA DE PREMISA ENTROU (TERMINADA COM "."), O SISTEMA
346500      PASSA A PEDIR PELAS CLAUSULAS DA CONCLUSAO, SENDO QUE
346600      DELIMITADOR DO TIPO "OU" NAO E' ACEITO ;
346700
346800      PEGAPALAVRA;
346900      LOOKUP;
347000      CASE CDELIM OF BEGIN
347100          1
347200          2 ... "&" = CLAUSULA A SER RECONHECIDA

```

```
347300      01 BEGIN
347400          RECONHECECLAUSULAE;
347500          IF NOT ERROIDENTIFICADO THEN NCLAUSULA:=**+1;
347600          WRITE(VIDEO,<2A1>,"LIMPATELA,BUZINA");
347700          IF PREMISSA THEN
347800              WRITE(VIDEO,<"ENTRE COM A CLAUSULA ","J2>,"NCLAUSULA);
347900          ELSE WRITE(VIDEO,<"ENTRE COM A CONCLUSAO ","J2>,"NCLAUSULA);
348000          LERCARTAO;
348100      END;
348200      *
348300      * ... "OU" = CLAUSULA TIPO OU A SER RECONHECIDA
348400      112: IF ENTAD THEN ERRO(11)
348500          ELSE IF REGRAESPECIAL THEN ERRO(48)
348600              ELSE BEGIN
348700                  RECONHECECLAUSULADU(COELIM);
348800                  IF NOT ERROIDENTIFICADO THEN NSUBCLAUSULA:=**+1;
348900                  WRITE(VIDEO,<2A1>,"LIMPATELA,BUZINA");
349000                  WRITE(VIDEO,<"ENTRE COM A CLAUSULA ","J2," (SUBCLAUSULA "
349100                      J2,"")>,"NCLAUSULA,NSUBCLAUSULA);
349200                  LERCARTAO;
349300              END;
349400              *
349500              * ... FIM DE PREMISSA OU DE REGRA
349600      31 BEGIN
349700          IF PREMISSA THEN FIMDEPREMISSA:=TRUE;
349800          RECONHECECLAUSULAE;
349900          IF ERROIDENTIFICADO THEN BEGIN
350000              IF PREMISSA THEN
350100                  WRITE(VIDEO,<"/>,"ENTRE COM A CLAUSULA NOVAMENTE">);
350200              ELSE WRITE(VIDEO,<"/>,"ENTRE COM A CONCLUSAO NOVAMENTE">);
350300              END
350400          ELSE
350500              IF PREMISSA THEN BEGIN
350600                  PREMISSA:=FALSE;
350700                  AVAIL:=AVAIL+1;
350800                  EENTAO:=AVAIL;
350900                  PROXC(AVAIL):=AVAIL+1;
351000                  MARCA(AVAIL):=2;
351100                  ELEMENTOS(AVAIL):=0;
351200                  ENTAD:=TRUE;
351300                  NCLAUSULA:=1;
351400                  WRITE(VIDEO,<2A1>,"LIMPATELA,BUZINA");
351500                  WRITE(VIDEO,<"ENTRE COM A CONCLUSAO ","J2>,"NCLAUSULA);
351600                  LERCARTAO;
351700              END
351800              ELSE BEGIN
351900                  PROXC(AVAIL-4):=EENTAO;
352000                  EULT:=AVAIL+1;
352100                  FIM:=TRUE;
352200              END;
352300          END;
352400      *
352500      * ... PALAVRA NAO E DELIMITADOR
352600      41 MONTACLAUSULA;
352700      END;
352800  END
352900  UNTIL FIM;
353000  IF NOT ERROIDENTIFICADO THEN BEGIN
353100      IF NOT REGRAESPECIAL THEN BEGIN
```



```
353200 WRITE(VIDEO,<"ESTA REGRA DEVE SER MOSTRADA NO ENCADEAMENTO ">
353300 "DE 'PORQUE' ?">);
353400 LERESPOSTA(SIMNAO);
353500 IF PALAVRA = "NAO" OR PALAVRA = "N" THEN EMOSTRAREGRA:=1;
353600 END;
353700 DECODIFICAREGRA;
353800 ARMAZENAREGRA;
353900 WRITE(VIDEO,<A1>,>BUZINA);
354000 WRITE(VIDEO[STOP],<"OK? (S/N)">);
354100 LERESPOSTA(SIMNAO);
354200 IF PALAVRA = "NAO" OR PALAVRA = "N"
354300 THEN BEGIN
354400 WRITE(VIDEO,<A1>,>BUZINA);
354500 WRITE(VIDEO[STOP],<"DESEJA ALTERAR A REGRA (S/N) ?">);
354600 LERESPOSTA(SIMNAO);
354700 IF PALAVRA = "SIM" OR PALAVRA = "S"
354800 THEN ALTERAREGRA
354900 ELSE EXCLUIREGRA;
355000 END;
355100 END;
355200 END 'DOIREREGRA;
355300
355400 -----
355500 INTEGER PROCEDURE INICIALIZA;
355600 BEGIN
355700 INTEGER CODIGO;
355800 OFFLINE ACHANUMERODAREGRA =
355900 BEGIN
356000 PEGAPALAVRA;
356100 IF PALAVRA = "REGRA" THEN BEGIN
356200 PEGAPALAVRA;
356300 IF NUMFRO(PALAVRA) THEN BEGIN
356400 NUMERODAREGRA:=DECIMAL(PALAVRA);
356500 IF NUMERODAREGRA < 1 THEN ERRO(26);
356600 END
356700 ELSE ERRO(26);
356800 END
356900 ELSE ERRO(26);
357000 END #;
357100
357200 COMMENT:
357300 * INICIO DE INICIALIZA *
357400
357500 VERIFICA QUAL A OPCAO DO ESPECIALISTA E DEVOLVE O CODIGO
357600 PARA O PROGRAMA PRINCIPAL ;
357700
357800 REGRAESPECIAL:=FALSE;
357900 NCLAUSULA:=0;
358000 NSUBCLAUSULA:=0;
358100 NUMERODAREGRA:=0;
358200
358300 ENTAO:=FALSE;
358400 PREMISSA:=FALSE;
358500 ALTERANDO:=FALSE;
358600
358700 IF NOT MENU THEN BEGIN
358800 WRITE(VIDEO,<2A1>,>BUZINA,BUZINA);
358900 WRITE(VIDEO[STOP],<"CODIGO DE INICIALIZACAO: ">);
359000 PEGAPALAVRA;
```

```
359100 IF PALAVRA = "MENU" THEN MENU:=TRUE
359200 ELSE IF PALAVRA = "REGRAESPECIAL" THEN REGRAESPECIAL:=TRUE
359300 CODIGO:=IF PALAVRA = "REGRANOVA" OR PALAVRA = "REGRAESPECIAL"
359400 THEN 1 ELSE
359500 IF PALAVRA = "EXCLUA" THEN 2 ELSE
359600 IF PALAVRA = "ALTERE" THEN 3 ELSE
359700 IF PALAVRA = "MOSTRE" THEN 4 ELSE
359800 IF PALAVRA = "REFERENCIAS" THEN 5 ELSE
359900 IF PALAVRA = "FIM" THEN 6 ELSE
360000 IF PALAVRA = "MENU" THEN 7 ELSE 0
360100 IF CODIGO > 1 AND CODIGO < 5 THEN ACHANUMERODAREGRA
360200 END
360300 IF MENU THEN BEGIN
360400 WRITE(VIDEO,"QUE OPERACAO DEVE SER FEITA :","/
360500 " 1) AQUISICAO DE REGRA"/
360600 " 2) EXCLUSAO DE REGRA"/
360700 " 3) ALTERACAO DE REGRA"/
360800 " 4) LISTAGEM DE REGRA"/
360900 " 5) LISTAGEM DE REFERENCIAS"/
361000 " 6) FIM"/
361100 " 7) DESATIVACAO DE MENU">))
361200 PEGAPALAVRA
361300 IF NOT NUMERO(PALAVRA) THEN CODIGO:=0
361400 ELSE BEGIN
361500 CODIGO:=DECIMAL(PALAVRA)
361600 IF CODIGO = 7 THEN MENU:=FALSE ELSE
361700 IF CODIGO<1 OR CODIGO>7 THEN CODIGO:=0
361800 ELSE IF CODIGO > 1 AND CODIGO < 5 THEN BEGIN
361900 WRITE(VIDEO[STOP],"NUMERO DA REGRA : ">))
362000 PEGAPALAVRA
362100 IF NUMERO(PALAVRA) THEN BEGIN
362200 NUMERODAREGRA:=DECIMAL(PALAVRA)
362300 IF NUMERODAREGRA < 1 THEN ERRO(26)
362400 END
362500 ELSE ERRO(26)
362600 END
362700 END
362800 END
362900
363000 INICIALIZA:=CODIGO
363100 IF ERROIDENTIFICADO THEN INICIALIZA:=0
363200 END INICIALIZA
363300
363400 *-----*
363500
363600 COMMENT:
363700
363800 * PROGRAMA PRINCIPAL *
363900
364000 ESTE PROGRAMA ADQUIRE, ALTERA E EXCLUI AS REGRAS QUE FORMAM
364100 UM BANCO DE CONHECIMENTO, MANIPULANDO COM ISSO TODOS OS
364200 ARQUIVOS TODOS OS ARQUIVOS ESTATICOS DO SISTEMA. O ACESSO E
364300 FEITO ATRAVES DE UMA LINGUAGEM DE COMANDOS OU ATRAVES DE
364400 "MENUS". A AQUISICAO DAS REGRAS E FEITA A LINGUAGEM PSEUDO-
364500 NATURAL COM AJUDA DE UM DICCIONARIO DE PALAVRAS"CHAVE"
364600
364700
364800 WRITE(VIDEO,"** ENTRE COM O DOMINIO DE APLICACAO">))
364900 WRITE(VIDEO[STOP],">")>))
```

```

36500 READ(VIDEO,<A30>,DOMINIO);
36510 DOMINIO:=HEAD( TAIL(DOMINIO, " "), NOT " ") CAT "/";
36520
36530 MUDA(REGRASDISK,"REGRASDISK");
36540 MUDA(TEXTODISK,"TEXTODISK");
36550 MUDA(ATRIBUTOSREF,"ATRIBUTOSREF");
36560 MUDA(ESQUELETODISK,"ESQUELETODISK");
36570 MUDA(DIRDISK,"DIRDISK");
36580 MUDA(OUTRASPALAVRAS,"OUTRASPALAVRAS");
36590 MUDA(TIPOSCONTEXTOS,"TIPOSCONTEXTOS");
36600 MUDA(OBJETOETIPOS,"OBJETOETIPOS");
36610 MUDA(PROPATRIBUTOS,"PROPATRIBUTOS");
36620 MUDA(CLASSEDISK,"CLASSEDISK");
36630 MUDA(OBJETIVOS,"OBJETIVOS");
36640
36650 MENU:=TRUE;
36660 FIMSESSAO:=FALSE;
36670 PERGUNTAS:=0;
36680 WRITE(VIDEO,<2A1>,"LIMPATELA,BUZINA");
36690 DO BEGIN
36700     * MANIPULACAO DAS REGRAS
36710     ERROIDENTIFICADO:=FALSE;
36720     CASE INICIALIZA OF BEGIN
36730         0: IF NOT ERROIDENTIFICADO THEN
36740             IF MENU THEN ERRO(50) ELSE ERRO(27);
36750         1: ADQUIREREGRA;
36760         2: EXCLUIREGRA;
36770         3: ALTERAREGRA;
36780         4: MOSTRAREGRA;
36790         5: LISTAREFERENCIA;
36800         6: FIMSESSAO:=TRUE;
36810         7:
36820     END;
36830 END;
36840 UNTIL FIMSESSAO;
36850 BEGIN
36860     INTEGER NRO;BOOLEAN SAI;
36870     DEFINE COLOCAOBJETIVO(I) =
36880         IF NOT PESQUISADIRETORIO OR CLASSE NEO I
36890         THEN BEGIN
36900             WRITE(VIDEO,<"OBJETIVO REJEITADO">);
36910             SAI:=TRUE;
36920         END
36930         ELSE OBJETIVO[I]:=L;
36940
36950     * AQUISICAO DOS OBJETIVOS POSSIVEIS PARA O DOMINIO
36960
36970     WRITE(VIDEO,<"** QUAIS OS OBJETIVOS POSSIVEIS NUMA CONSULTA?">,
36980         "(MAX = 12)">);
36990     OBJETIVO[1]:=999;
37000     WRITE(OBJETIVOS[0],14,OBJETIVO);
37010     READ(OBJETIVOS[0],14,OBJETIVO);
37020     NRO:=OBJETIVOS.LASTRECORD + 1;
37030     WHILE NRO<=12
37040     DO BEGIN
37050         SAI:=FALSE;
37060         WRITE(VIDEO,<"OBJETIVO ",15,NRO>);
37070         WRITE(VIDEO[STOP],<"** ATRIBUTO 1">);
37080         READ(VIDEO,<A48>,"CARTAO");

```

```

370900 PEGAPALAVRA;
371000 IF PALAVRA = EMPTY THEN NRO:=13
371100 ELSE BEGIN
371200   COLOCAOBJETIVO(1);
371300   IF NOT SAI THEN BEGIN
371400     WRITE(VIDEO[STOP],<"** OBJETO !">);
371500     READ(VIDEO,<A50>,CARTAO);
371600     PEGAPALAVRA;
371700     COLOCAOBJETIVO(2);
371800     IF NOT SAI THEN BEGIN
371900       WRITE(VIDEO,<"** COLOQUIALMENTE !">);
372000       READ(VIDEO,<A66>,CARTAO);
372100       REPLACE POINTER(OBJETIVO(3)) BY CARTAO;
372200       WRITE(VIDEO,<"** OBJETO !",I5," ATRIBUTO !",I5,"
372300         ** ",A66,"//","OK?(S/N)">,OBJETIVO(1),
372400         OBJETIVO(2),CARTAO);
372500       CARTAO:=EMPTY;
372600       PEGAPALAVRA;
372700       IF PALAVRA="SIM" OR PALAVRA="S" OR PALAVRA=EMPTY
372800       THEN BEGIN
372900         WRITE(OBJETIVOS(NRO),14,OBJETIVO);
373000         NRO:=NRO+1;
373100         END
373200       END
373300     END
373400   END
373500 END
373600 END
373700 END.

```

E.4 - CONSULTA

```

100 BEGIN
200   FILE VIDEO(KIND=REMOTE,UNITS=CHARACTERS);
300   FILE TERM(KIND=REMOTE,UNITS = CHARACTERS,MYUSE=10);
400   FILE SAIDA(KIND=PRINTER);
500
600
700 -----ARQUIVO TIPOSCONTEXTOS-----
800
900   FILE TIPOSCONTEXTOS(KIND=DISK,FILETYPE=7);
1000  ARRAY TIPOCNTXT[0:39];
1100  BOOLEAN TRACE;
1200  STRING TRADUZ;
1300  PRIMEIRONO;
1400  NOADICIONAL;
1500  POINTER PP;
1600  DEFINE OBJETO= TIPOCNTXT[0].[47:10] #,
1700  ASSUMIR          = TIPOCNTXT[0].[37:1] #,
1800  NUMASSOCIADOS    = TIPOCNTXT[0].[36:3] #,
1900  ASSOCIADO(X)     = TIPOCNTXT[0].[33-(X-1)*8:8] #,
2000  ATRIBUTOPRINCIPAL(X) = TIPOCNTXT[1].[47-(X-1)*10:10] #,
2100  LENG(X)          = TIPOCNTXT[2].[37-(X-1)*10:10] #,
2200
2300  LETIPOSCONTEXTOS(X) =
2400  X-----
2500  READ(TIPOSCONTEXTOS[X],40,TIPOCNTXT);
2600  PPI=POINTER(TIPOCNTXT[3]);
2700  TRADUZ:=STRING(PP,LENG(1));
2800  PRIMEIRONO:=STRING(PP,LENG(1),LENG(2));
2900  NOADICIONAL:=STRING(PP+(LENG(1)+LENG(2)),LENG(3));
3000
3100  BUSQUEATRIBUTOSPRINCIPAIS =
3200  X-----
3300  BEGIN
3400    INTEGER I;
3500    FOR I=1 STEP 1 UNTIL 6 DO
3600      IF ATRIBUTOPRINCIPAL(I) = 0 THEN
3700        %BUSCAVALOR(ATRIBUTOPRINCIPAL(I),P,0,0,REGRAPI);
3800      END #;
3900
4000 -----ARVORE DE CONTEXTOS-----
4100
4200  ARRAY ARVORECONTEXTOS[0:127];
4300  INTEGER AVAIL;
4400  DEFINE FILHO(X)      = ARVORECONTEXTOS[X].[47:7] #,
4500  PROXIMO(X)          = ARVORECONTEXTOS[X].[40:7] #,
4600  IRMAD(X)            = ARVORECONTEXTOS[X].[33:7] #,
4700  TIPO(X)             = ARVORECONTEXTOS[X].[26:7] #,
4800  NUMCONTEXTOS(X)     = ARVORECONTEXTOS[X].[19:3] #,
4900  PAI(X)              = ARVORECONTEXTOS[X].[16:7] #,
5000  OBJECT(X)          = ARVORECONTEXTOS[X].[9:10] #,
5100
5200 -----ARVORE DE RACIOCINIO -----
5300
5400  ARRAY REGRACONSULTA[0:13,0:1999];
5500  INTEGER REGRA;
5600  DEFINE REGRA(I)      = REGRACONSULTA[0,I].[47:10] #,

```

```
5700      RPAI(I)          = REGRACONSULTA[0,I].[37:10] #,
5800      RIRMAO(I)        = REGRACONSULTA[0,I].[27:10] #,
5900      RFILHO(I)         = REGRACONSULTA[0,I].[17:10] #,
6000      RCONCLUSAO(I)    = REGRACONSULTA[0,I].[7:4]  #,
6100      RSUCCESSO(I)    = REGRACONSULTA[0,I].[1:1]   #,
6200      RTIPO(I)          = REGRACONSULTA[0,I].[0:1]   #,
6300      RPROXIMO(I)       = REGRACONSULTA[1,I].[47:10] #,
6400      RPROXAPLICACAO(I) = REGRACONSULTA[1,I].[37:10] #,
6500      RCRIACONTEXTO(I)  = REGRACONSULTA[1,I].[27:10] #,
6600      RATRIBUTO(I)      = REGRACONSULTA[1,I].[17:10] #,
6700      RCONTEXTO(I)      = REGRACONSULTA[1,I].[07:07] #,
6800      RESPECIAL(I)      = REGRACONSULTA[1,I].[00:01] #,
6900      RCL(I)            = REGRACONSULTA[2,I].[47:05] #,
7000      RSUBCL(I)        = REGRACONSULTA[2,I].[42:05] #,
7100      RCERTEZA(I)       = REGRACONSULTA[2,I].[37:07] #,
7200      RNEGACAO(I)       = REGRACONSULTA[2,I].[30:01] #,
7300      RVALOR(I)         = REGRACONSULTA[3,I] #,
7400
7500      X-----6----- ANSWER -----
7600
7700      ARRAY ANSWER[1:200,0:11]
7800      INTEGER ULTIMALINHA,PERGUNTAMUDADA
7900
8000      X-----6----- PROPRIEDADES DE ATRIBUTOS -----
8100
8200      FILE PROPATRIBUTOS(KIND=DISK,FILETYPE=7)
8300      ARRAY PROPRIEDADE[0:69]
8400      DEFINE DEDUZIRVALOR = PROPRIEDADE[61] = 0 #,
8500      ESPERADO           = STRING(POINTER(PROPRIEDADE),180) #,
8600      PERGUNTA           = STRING(POINTER(PROPRIEDADE[30]),186) #,
8700      TRADUCAO           = STRING(POINTER(PROPRIEDADE[62]),48) #,
8800
8900      X-----6----- ESQUELETOS -----
9000
9100      FILE ESQUELETODISK(KIND=DISK,FILETYPE=7)
9200      COMMENT:-----
9300      MAXRECSIZE ..... 30
9400      BLOCKSIZE ..... 300
9500      AREASIZE ..... 10
9600
9700      ARRAY ESQUELETO[0:29]
9800      STRING SKELETON
9900
10000     X-----6----- OUTRAS PALAVRAS CONHECIDAS -----
10100
10200     FILE OUTRASPALAVRAS(KIND=DISK,FILETYPE=7)
10300     COMMENT:-----
10400     MAXRECSIZE ..... 7
10500     BLOCKSIZE ..... 210
10600     AREASIZE ..... 30
10700
10800     X-----6----- CLASSES DE VALORES -----
10900
11000     FILE CLASSEDISK(KIND=DISK,FILETYPE=7)
11100     ARRAY CLASS[0:19]
11200     DEFINE NOMECLASSE = HEAD(STRING(POINTER(CLASS),120),NOT " ") #,
11300     ELEMENTOS = TAIL(STRING(POINTER(CLASS),120),NOT " ") #,
11400
11500     X-----6----- PALAVRAS CONHECIDAS -----
```

```

11600
11700 FILE DIRDISK(KIND=DISK, FILETYPE=7)
11800 INTEGER L
11900 ARRAY DIRETORIO[0:9]
12000 DEFINE TIPOATRIBUTO = DIRETORIO[0].[12:13] #,
12100 SUBMULTIPL = DIRETORIO[0].[12:10] #,
12200 ASSOC = DIRETORIO[0].[22:10] #,
12300 SINONIMO = DIRETORIO[0].[42:10] #,
12400 CLASSE = DIRETORIO[0].[47:5] #,
12500 CHAVEREG = DIRETORIO[1].[9:10] #,
12600 VALORSUB = DIRETORIO[2] #,
12700 DIRNOME = HEAD(STRING(POINTER(DIRETORIO[3]),42),NOT "/")#)
12800
12900 *----- REFERENCIAS A ATRIBUTOS -----*
13000
13100 FILE ATRIBUTOSREF(KIND=DISK, FILETYPE=7)
13200 DEFINE TYPEREGRA(X) = REFERENCIAS[X].[40:01] #,
13300 NUMREGRA(X) = REFERENCIAS[X].[39:14] #,
13400 NEGACAO(X) = REFERENCIAS[X].[15:6] #,
13500 RLINK(X) = REFERENCIAS[X].[7:8] #
13600
13700 *----- LISTA DE TIPOS -----*
13800
13900 FILE OBJETOETIPOS(KIND=DISK, FILETYPE=7)
14000 COMMENT:-----*
14100 MAXRECSIZE ..... 4
14200 BLOCSSIZE ..... 120
14300 AREASIZE ..... 30
14400
14500 ARRAY LISTADETIPOS[0:3]
14600 DEFINE CONTEXTO(X) = LISTADETIPOS[X DIV 5].[39-(X-(X DIV 5)*4 -1)
14700 *10:10] #,
14800 NROTIPO(X) = LISTADETIPOS[X DIV 5 +2].[39-
14900 (X-(X DIV 5)*4 -1)*10:10] #,
15000 NULO = LISTADETIPOS[0].[47:1] #,
15100 ?NULO = LISTADETIPOS[2].[47:8] #,
15200 LTOTAL = LISTADETIPOS[0].[46:4] #
15300
15400 *----- ARQUIVO DE REGRAS -----*
15500
15600 FILE REGRASDISK(KIND=DISK, FILETYPE=7)
15700 DEFINE MARCA(X) = REGRA[X].[47:2] #,
15800 GUI(X) = REGRA[X].[45:1] #,
15900 PROXC(X) = REGRA[X].[5:6] #,
16000 PROXL(X) = REGRA[X].[13:7] #,
16100 EESPECIAL = REGRA[0].[33:01] #,
16200 EMOSTRAREGRA = REGRA[0].[32:01] #,
16300 EPREMISSA = REGRA[0].[31:7] #,
16400 ENTAD = REGRA[0].[24:7] #,
16500 EATRIBUTO = REGRA[1].[45:10] #,
16600 EOBJETO = REGRA[1].[35:10] #,
16700 EFUNCAO = REGRA[1].[25:10] #,
16800 EUNIDADE = REGRA[1].[15:10] #,
16900 ESQUELETO = REGRA[1+1].[37:10] #,
17000 EMODFUNCAO = REGRA[1+1].[27:10] #,
17100 ETIPOVALOR = REGRA[1+1].[05:02] #,
17200 EREPERGUNTA = REGRA[1+1].[03:01] #,
17300 EVALOR = REGRA[1+2] #,
17400 EFATOR = REGRA[1+3] #

```

```

17500 ETIPOREGRA(X)= REGRA[I+4].[47-(X-1)*8 18] #
17600 REGRAESPECIAL= (FESPECIAL= 1) #
17700
17800 *----- ARQUIVO DE VALORES CONHECIDOS -----*
17900
18000 FILE VALORCONHECIDO(KIND=DISK,FILETYPE=7)
18100 ARRAY CONHECO[1:30]
18200 DEFINE KCONTEXTO(X) = CONHECO[X].[47:7] #
18300 KMB(X) = CONHECO[X].[39:7] #
18400 KMD(X) = CONHECO[X].[32:7] #
18500 KPD(X) = CONHECO[X].[25:1] #
18600 KREGRA(X) = CONHECO[X].[24:10] #
18700 KNUMPERGUNTA(X) = CONHECO[X].[14:6] #
18800 KUNI(X) = CONHECO[X].[8:1] #
18900 KDES(X) = CONHECO[X].[7:1] #
19000 KTIPO(X) = CONHECO[X].[6:2] #
19100 KVALOR(X) = CONHECO[X+1] #
19200
19300 *----- OBJETIVOS DE CONSULTA -----*
19400
19500 FILE OBJETIVOS(KIND=DISK,FILETYPE=7)
19600 COMMENT-----
19700 MAXRECSIZE..... 14
19800 BLOCKSIZE..... 420
19900 AREASIZE..... 30
20000
20100 ARRAY OBJETIVO[1:14]
20200
20300 *----- TEXTOS NAS REGRAS -----*
20400
20500 FILE TEXTODISK(KIND=DISK,FILETYPE=7)
20600 COMMENT-----
20700 MAXRECSIZE..... 90
20800 BLOCKSIZE..... 900
20900 AREASIZE..... 10
21000
21100 BOOLEAN TEXTOS
21200 INTEGER ETEXTOS
21300 STRING TEXTUAL
21400 ARRAY CLAUSE[0:89]
21500
21600 *----- AUXILIARES -----*
21700
21800 INTEGER P, TIPOCERTO, CONCLUSO, NUMPERGUNTA, PVETOR, ATRIBUTOOBJETIVO,
21900 TIPOATRIB, ZERO, AUX, COUNTER
22000 STRING RESPOSTA, LINHA, PALAVRA, LINE
22100 BOOLEAN ACABOU, CONTEXTOCRIADO, FALSO, MUDANDO, DENTRONDESE
22200 REAL VAL, CFMI, VALH, VALD, VALM
22300 ARRAY LASTCNTXT[0:49], TABELA[1:65, 0:11], DESATIVADA[0:149],
22400 NOME DOMINIO[0:9]
22500 BOOLEAN ARRAY NAO TEM CONTEXTO[0:99], REGACESSADO[0:99]
22600 LABEL FIMDETUDO, REINICIA
22700 INTEGER ARRAY VETORCONCLUSOES[1:15]
22800 STRING DUMINIO
22900 DEFINE MUDA(ARQUIVO, ENTREASPAS)=
23000 *-----
23100 REPLACE POINTER(NOME DOMINIO)
23200 BY (DOMINIO CAT ENTREASPAS CAT ".");
23300 REPLACE ARQUIVO.TITLE BY POINTER(NOME DOMINIO) ;

```



```

23400      OPEN(ARQUIVO) #
23500      SIMNAD = 0 #
23600      LERESPSTA(X) =
23700      PEGA(RESPOSTA)
23800      CASE X OF BEGIN
23900      SIMNAD WHILE (RESPOSTA NEQ "SIM" AND RESPOSTA NEQ "S")
24000      AND (RESPOSTA NEQ "NAO" AND RESPOSTA NEQ "N")
24100      DO PEGA(RESPOSTA)
24200      END CASE #
24300      REGISTROJAEXISTE = REGACESSADO[CHAVEREG/2] #
24400      OBJETO(X) = LASTCNTXT[X],[47:10] #
24500      NUMERO(X) = LASTCNTXT[X],[37:10] #
24600      NAONUMERICO = 0 #
24700      NUMERICO = 1 #
24800      VALORMULTIPLO = 2 #
24900      VALORUNICO = 1 #
25000      RAIZ = 0 #
25100      LIMPATELA = 12 #
25200      BUZINA = 47 #
25300      ERRO(X) = WRITE(VIDEO,<U>,X) #
25400
25500      DIASEMANA = T.[05:06] #
25600      SEGUNDOS = T.[11:06] #
25700      MINUTOS = T.[17:06] #
25800      HORA = T.[23:06] #
25900      DIA = T.[29:06] #
26000      MES = T.[35:06] #
26100      ANO = T.[47:12] #
26200
26300      ESCREVFEMREGRA =
26400      WRITE(SAIDA,<X5,"REGRA PAI IRMAO FILHO CONCL SUCE TIPO PROX"
26500      " PAPL CRIA ATRI CNTXT RCL RSUBCL">))
26600      FOR I:=0 STEP 1 UNTIL PREGRA
26700      DO WRITE(SAIDA,<I2," = ",I5,I4,I6,I6,I6,I6,I5,I5,I5,I5,I6,
26800      I5,I5>,I,RRREGRA(I),RPAI(I),RIRMAO(I),RFILHO(I),RCONCLUSAO(I),
26900      RSUCFSSC(I),RTIPO(I),RPROXIMO(I),RPROXAPLICACAO(I),
27000      RCRFACONTEXT(I),RATRIBUTO(I),RCONTEXT(I),RCL(I),RSUBCL(I))) #
27100      X-----
27200
27300      DEFINE
27400      PEGAPALAVRADOCARTAO =
27500      X-----
27600
27700      CARTAO:=TAIL(CARTAO," ")
27800      IF CARTAO NEQ EMPTY THEN BEGIN
27900      PALAVRA:=HEAD(CARTAO,NOT " ")
28000      CARTAO:=TAIL(CARTAO,NOT " ")
28100      END
28200      ELSE PALAVRA!="?" #
28300
28400      MOSTREMENSAGEM(X) =
28500      X-----
28600      BEGIN
28700      STRING M,L
28800      M:=X
28900      WRITE(VIDEO,</>)
29000      WHILE LENGTH(M) > 60 DO
29100      IF TAIL(M," ") = EMPTY THEN M:=EMPTY
29200      ELSE BEGIN

```

```

2930      L:=TAKE(M,60);
2940      WHILE DROP(L,LENGTH(L) - 1) NEO " "
2950      DO L:=TAKE(L,LENGTH(L) - 1);
2960      M:=DROP(M,LENGTH(L));
2970      WRITE(VIDEO,<A>,LENGTH(L),L);
2980      IF TRACE THEN WRITE(SAIDA,<A>,LENGTH(L),L);
2990      END;
3000      IF LENGTH(M) > 0 THEN WRITE(VIDEO,<A>,LENGTH(M),M);
3010      WRITE(VIDEO,</>);
3020      IF TRACE AND LENGTH(M) > 0 THEN WRITE(SAIDA,<A>,LENGTH(M),M);
3030      END #;
3040
3050      DECODIFICAESQUELETO =
3060      *-----*
3070      BEGIN
3080
3090      COMMENT:
3100      TOMA O ESQUELETO E SUBSTITUI CADA SIMBOLO PELA PALAVRA
3110      CORRESPONDENTE. SE FOR UM DOS PARAMETROS PRINCIPAIS "ESTE F"
3120      "TIRADO DO ARRAY ELEMENTO" SENAO DO ARQUIVO OUTRASPALAVRAS ;
3130
3140
3150      READ(ESQUELETO,DISK(EESQUELETO),30,ESQUELETO);
3160      CARTAO:=STRING(POINTER(ESQUELETO),160);
3170
3180      PEGAPALAVRADOCARTAO;
3190      WHILE PALAVRA NEO "?" DO BEGIN
3200          IF TAKE(PALAVRA,1) = "s" THEN BEGIN
3210              INTEGER N;
3220              DEFINE UMNUMERO = N > 7 #;
3230              N:=DECIMAL(DROP(PALAVRA,1));
3240              IF NOT UMNUMERO THEN BEGIN
3250                  CASE N OF BEGIN
3260                      1 : CL:=EATRIBUTO;
3270                      2 : CL:=EOBJETO;
3280                      3 : CL:=EFUNCAO;
3290                      4 : CL:=EVALOR;
3300                      5 : CL:=EUNIDADE;
3310                      6 : CL:=EMODFUNCAO;
3320                      7 : PALAVRA:=STRING(EVALOR,*);
3330                      END;
3340                  IF N NEO 7 THEN BEGIN
3350                      READ(DIRDISK[CL],10,DIRETORIO);
3360                      PALAVRA:=DIRNOME;
3370                      END;
3380                  END;
3390              END
3400
3410              ELSE BEGIN
3420                  ARRAY P[0:6];
3430                  READ(OUTRASPALAVRAS[DECIMAL(PALAVRA)-1,7,P]);
3440                  PALAVRA:=HEAD(STRING(POINTER(P),42),NOT " ");
3450                  END;
3460                  LINE:=LINE CAT PALAVRA CAT " ";
3470                  PEGAPALAVRADOCARTAO;
3480                  END;
3490              END #;
3500
3510      PREENCHECOMANDOS =

```

```

35200      2---2-----
35300
35400      BEGIN
35500          INTEGER I;
35600          FOR I:=1 STEP 1 UNTIL 65 DO FILL TABELA[I,*] WITH 12("      ");
35700          FILL TABELA[01,*] WITH "COMANDOS";
35800          FILL TABELA[02,*] WITH "-----";
35900          FILL TABELA[03,*] WITH "      MOSTRA ESTA LISTA DE COMANDOS";
36000          FILL TABELA[05,*] WITH "PERGUNTA";
36100          FILL TABELA[06,*] WITH "-----";
36200          FILL TABELA[07,*] WITH "      REPETE A PERGUNTA FEITA AO USUARIO";
36300          FILL TABELA[09,*] WITH "REGRA <LISTA>";
36400          FILL TABELA[10,*] WITH "-----";
36500          FILL TABELA[11,*] WITH "      MOSTRA AS REGRAS DA LISTA OU A REGRA COR
36600 RENTE CASO LISTA SEJA OMITIDA";
36700          FILL TABELA[13,*] WITH "PORQUE";
36800          FILL TABELA[14,*] WITH "-----";
36900          FILL TABELA[15,*] WITH "      PERCORRE A ARVORE DE RACIOCINIO EM ORDEM
37000 INVERSA A CADA CHAMADA";
37100          FILL TABELA[16,*] WITH "      CONSECUTIVA DE 'PORQUE' SOBRE UM NIVEL NA
37200 ARVORE";
37300          FILL TABELA[18,*] WITH "QUANDO E USADO <ATRIBUTO> ?";
37400          FILL TABELA[19,*] WITH "-----";
37500          FILL TABELA[20,*] WITH "      MOSTRA TODAS AS REGRAS QUE POSSUEM O ATR
37600 IBUTO NA PREMISSE";
37700          FILL TABELA[22,*] WITH "COMO CONCLUIR <ATRIBUTO> ?";
37800          FILL TABELA[23,*] WITH "-----";
37900          FILL TABELA[24,*] WITH "      MOSTRA TODAS AS REGRAS QUE POSSUEM O ATR
38000 IBUTO NA CONCLUSAO";
38100          FILL TABELA[26,*] WITH "RESPOSTAS";
38200          FILL TABELA[27,*] WITH "-----";
38300          FILL TABELA[28,*] WITH "      MOSTRA OS VALORES POSSIVEIS PARA O ATRIB
38400 IBUTO CORRENTE";
38500          FILL TABELA[30,*] WITH "ARVORE";
38600          FILL TABELA[31,*] WITH "-----";
38700          FILL TABELA[32,*] WITH "      MOSTRA TODA A ARVORE DE CONTEXTOS";
38800          FILL TABELA[34,*] WITH "ANCESTRAIS DE CONTEXTO";
38900          FILL TABELA[35,*] WITH "-----";
39000          FILL TABELA[36,*] WITH "      MOSTRA OS NOS ACIMA DE <CONTEXTO> NA ARV
39100 ORE DE CONTEXTOS";
39200          FILL TABELA[38,*] WITH "DESCENDENTES DE <CONTEXTO>";
39300          FILL TABELA[39,*] WITH "-----";
39400          FILL TABELA[40,*] WITH "      MOSTRA OS NOS ABAIXO DE <CONTEXTO> NA AR
39500 VORE DE CONTEXTOS";
39600          FILL TABELA[42,*] WITH "VALORES DE <CONTEXTO> PARA <LISTA>";
39700          FILL TABELA[43,*] WITH "-----";
39800          FILL TABELA[44,*] WITH "      MOSTRA OS VALORES DE CADA ATRIBUTO DA <L
39900 ISTA> OU DO ATRIBUTO";
40000          FILL TABELA[45,*] WITH "      CORRENTE CASO SE DIGITE SOMENTE 'VALORES
40100 DE <CONTEXTO>'";
40200          FILL TABELA[47,*] WITH "COMO CONCLUIU <ATRIBUTO> DE <CONTEXTO> = <V
40300 ALOR> ?";
40400          FILL TABELA[48,*] WITH 10("-----");
40500          FILL TABELA[49,*] WITH "      EXIBE O TRECHO DA ARVORE DE RACIOCINIO Q
40600 UE VALIDOU O <ATRIBUTO>";
40700          FILL TABELA[50,*] WITH "      MOSTRANDO AS REGRAS OU A PERGUNTA UTILIZ
40800 ADA PARA CONCLUI-LO";
40900          FILL TABELA[52,*] WITH "MUDAR <ATRIBUTO> DE <CONTEXTO>";
41000          FILL TABELA[53,*] WITH "-----";

```

```

41100      FILL TABELA[54,*] WITH "      LIMPA AS ESTRUTURAS DINAMICAS, TOMA O NO
41200      VO VALOR E REINICIA A";
41300      FILL TABELA[55,*] WITH "      CONSULTA DESDE O INICIO";
41400      FILL TABELA[57,*] WITH "SE <ATRIBUTO1> DE <CONTEXTO> = <VALOR> ENTA
41500      O <ATRIBUTO2> ?";
41600      REPLACE POINTER(TABELA[58,*]) BY REPEAT("-",58);
41700      FILL TABELA[59,*] WITH "      MONTA UMA CADEIA DE RACIOCINIO SIMBOLICO
41800      : AS CONDICoes A SEREM";
41900      FILL TABELA[60,*] WITH "      AVALIADAS POR 'BUSCAVALOR' SERAO CONSIDE
42000      RADAS VERDADEIRAS";
42100      FILL TABELA[62,*] WITH "TERMINE";
42200      FILL TABELA[63,*] WITH "-----";
42300      FILL TABELA[64,*] WITH "      TERMINA A CONSULTA";
42400      END #;
42500
42600      EFLINHA =
42700      X-----
42800
42900      BEGIN
43000          ULTIMALINHA:=* + 1;
43100          IF NOT MUDANDO THEN BEGIN
43200              WRITE(VIDEO[STOP],<" > " >);
43300              READ(TERM,<A68>,LINHA);
43400              REPLACE POINTER(ANSWER[ULTIMALINHA,*])
43500              BY LINHA;
43600          END
43700          ELSE LINHA:=STRING(POINTER(
43800              ANSWER[ULTIMALINHA,*]),68);
43900          IF TRACE THEN WRITE(SAIDA,<" > " > <A68>,LINHA);
44000          END #;
44100
44200      MOSTREPERGUNTA(X) =
44300      X-----
44400
44500      BEGIN
44600          NUMPERGUNTA:=*+1;
44700          X:= STRING(NUMPERGUNTA,*) CAT " } " CAT X;
44800          IF NOT MUDANDO THEN MOSTREMENSAGEM(X);
44900          END #;
45000
45100      X-----
45200
45300      PROCEDURE IMPRIMELINHA;
45400      BEGIN
45500
45600      COMMENT :
45700          ** IMPRIMELINHA **
45800          -----
45900      IMPRIME O STRING "LINE" EM LINHAS DE 60 CARACTERES &
46000
46100      FILL VIDEO(KIND=REMOTE,UNITS=CHAPACTERS);
46200      STRING L;
46300      L:=TAKE(LINE,MIN(72,LENGTH(LINE)));
46400      IF LENGTH(L) = 72 THEN
46500          WHILE DROP(L,LENGTH(L) - 1) NEO " "
46600              DO L:=TAKE(L,LENGTH(L) - 1);
46700      WRITE(VIDEO,<A>,LENGTH(L),L);
46800      LINE:=DROP(LINE,LENGTH(L));
46900      WHILE LINE NEO EMPTY DO BEGIN

```

```
4700      LI:=TAKE(LINE,MIN(60,LENGTH(LINE)))
4710      IF LENGTH(L) = 60 THEN
4720          WHILE DROP(L,LENGTH(L) - 1) NEQ " "
4730              DO LI:=TAKE(L,LENGTH(L) - 1)
4740              WRITE(VIDEO,<X12,A*>LENGTH(L),L)
4750              LINE:=DROP(LINE,LENGTH(L))
4760          END
4770      END IMPRIMELINHA
4780
4790      *-----*
4800
4810      BOOLEAN PROCEDURE REGRAATIVA(NUMERODAREGRA);
4820      INTEGER NUMERODAREGRA;
4830      BEGIN
4840
4850          INTEGER X;
4860          BOOLEAN ATIVA;
4870          DEFINE DISPO = DESATIVADA[0] #;
4880
4890          COMMENT:
4900              ** REGRAATIVA **
4910              -----
4920              ESSE PROCEDIMENTO VERIFICA SE O NUMERO DA REGRA EM QUESTAO
4930              APARECE NA PILHA DE REGRAS DESATIVADAS, CASO EM QUE A REGRA
4940              NAO E ATIVA;
4950
4960              ATIVA:=TRUE;
4970              FOR X:=DISPO STEP -1 WHILE (X > 0) AND (ATIVA) DO
4980                  IF DESATIVADA[X] = NUMERODAREGRA THEN ATIVA:=FALSE;
4990              REGRAATIVA:=ATIVA;
5000              END REGRAATIVA;
5010
5020      *-----*
5030
5040      PROCEDURE DECODIFICAREGRA(NUMERODAREGRA,WHY,NCL,NSUBCL,NCONCLUSAO);
5050      VALUE NUMERODAREGRA,WHY,NCL,NSUBCL,NCONCLUSAO;
5060      INTEGER NUMERODAREGRA,WHY,NCL,NSUBCL,NCONCLUSAO;
5070
5080      BEGIN
5090          ARRAY REGRA[0:149];
5100          INTEGER I,CL,NCLAUSULA,NSUBCLAUSULA,NUMCLAUSULA,NUMSUBCLAUSULA;
5110          BOOLEAN PREMISA,ENTAO,DENTRODEOU,DENTRODEOU1,NUMFRADO;
5120          STRING PALAVRA,CARTAO;
5130
5140          DEFINE PORQUE = WHY = 1 #;
5150          *-----*
5160
5170      *-----*
5180
5190      PROCEDURE DECODIFICARCLAUSULA;
5200
5210      COMMENT:
5220          ** DECODIFICARCLAUSULA **
5230          PREPARA A CLAUSULA PARA IMPRESSAO COMANDADA TANTO POR
5240          "MOSTRAR" QUANTO POR "PORQUE", E A IMPRIME ;
5250
5260      BEGIN
5270          IF PORQUE AND PREMISA THEN BEGIN
5280              NUMCLAUSULA:=NCLAUSULA - NCL + 1;
```

```

52901      NUMSUBCLAUSULA:=NSUBCLAUSULA + NSUBCL + 1;
53001      END
53101  ELSE BEGIN
53201      NUMCLAUSULA:=NCLAUSULA;
53301      NUMSUBCLAUSULA:=NSUBCLAUSULA;
53401      END;
53501  IF NOT PORQUE THEN LINE:=""      *)
53601  IF NUMCLAUSULA = 1 OR (ENTAO AND PORQUE) THEN
53701      IF PREMISSA THEN
53801          IF PORQUE THEN BEGIN
53901              NUMERADO:=TRUE;
54001              IF NCL > 1 OR NSUBCL > 1
54101              THEN WRITE(VIDEO,"ASSIM SENDO SE 1")
54201              ELSE WRITE(VIDEO,"SE 1");
54301              END
54401              ELSE IF NUMSUBCLAUSULA < 2 THEN LINE:="SE      1 "
54501              ELSE IF PORQUE THEN WRITE(VIDEO,"ENTAO 1")
54601              ELSE LINE:="ENTAO 1 "
54701
54801  COMMENT: NUMERACAO DAS CLAUSULAS;
54901
55001  IF NUMERADO THEN
55101  IF DENTRODEU THEN
55201  IF NUMSUBCLAUSULA = 1 THEN LINE:=LINE CAT STRING(NUMCLAUSULA,*) CAT
55301      " ) (" CAT STRING(NUMCLAUSULA,*) CAT "." CAT
55401      STRING(NUMSUBCLAUSULA,*) CAT " "
55501  ELSE LINE:=LINE CAT "      " CAT STRING(NUMCLAUSULA,*) CAT "." CAT
55601      STRING(NUMSUBCLAUSULA,*) CAT " "
55701  ELSE LINE:=LINE CAT STRING(NUMCLAUSULA,*) CAT " ) "
55801
55901  COMMENT: RECUPERA OS ELEMENTOS DA CLAUSULA E DECODIFICA;
56001
56101  READ(DIRDISK(EFUNCAO),10,DIRETORIO);
56201  IF SINOIMO = 2 THEN BEGIN
56301      READ(TEXTODISK(EVALOR),90,CLAUSE);
56401      LINE:=LINE CAT
56501          HEAD(STRING(POINTER(CLAUSE),540),NOT " ");
56601      END
56701  ELSE BEGIN
56801      IF ENTAO THEN
56901          IF EFATOR = 1. THEN LINE:=LINE CAT "E' DEFINITIVO (1.0) QUE "
57001          ELSE
57101              IF EFATOR GEQ .8 THEN
57201                  LINE:=LINE CAT "EXISTE FORTE EVIDENCIA (" CAT
57301                  STRING(EFATOR,*) CAT " ) QUE "
57401              ELSE
57501                  IF EFATOR GEQ .4 THEN
57601                      LINE:=LINE CAT "EXISTE EVIDENCIA (" CAT
57701                      STRING(EFATOR,*) CAT " ) QUE "
57801                  ELSE LINE:=LINE CAT "EXISTE FRACA EVIDENCIA (" CAT
57901                  STRING(EFATOR,*) CAT " ) QUE "
58001
58101      DECODIFICAESQUELETO;
58201
58301      END;
58401
58501  COMMENT: DELIMITADORES FINAIS DE CLAUSULA;
58601
58701

```

```

5880 IF NUMERADO THEN
5890 IF DENTRODEU THEN
5900 IF MARCA(PROXC(I)) = 3 THEN
5910 IF MARCA(PROXL(PROXC(I))) = 1 THEN
5920 LINE:=LINE CAT ")"
5930 ELSE LINE:=LINE CAT ") &"
5940 ELSE IF DENTRODEU1
5950 THEN LINE:=LINE CAT " OU1"
5960 ELSE LINE:=LINE CAT " OU"
5970 ELSE
5980 IF MARCA(PROXC(I)) NEQ 1 AND MARCA(PROXC(I)) NEQ 2
5990 THEN LINE:=LINE CAT " &"
6000
6010 IF ENTÃO AND PORQUE THEN LINE:=LINE CAT " (REGRA " CAT
6020 STRING(NUMERODAREGRA,*) CAT ")")
6030 COMMENT: CLAUSULA PRONTA PARA IMPRESSAO
6040
6050 IMPRIMELINHA
6060 END DECODIFICARCLAUSULA
6070
6080 COMMENT:
6090 ** INICIO DE DECODIFICAREGRA **
6100 DECODIFICA A REGRA DADA CLAUSULA A CLAUSULA
6110
6120 READ(REGRASDISK(NUMERODAREGRA),150,REGRA)
6130 ENTÃO:=FALSE
6140 DENTRODEU:=FALSE
6150 DENTRODEU1:=FALSE
6160 PREMISA:=TRUE
6170 NCLAUSULA:=1
6180 NSUBCLAUSULA:=1
6190 IF NOT PORQUE THEN WRITE(VIDEO,<A1>BUZINA)
6200 NUMERADO:=TRUE
6210 IF NOT PORQUE THEN
6220 WRITE(VIDEO,<72("=")>,"REGRA "J4>NUMERODAREGRA)
6230 ELSE IF NCL > 1 OR NSUBCL > 1 THEN BEGIN
6240 WRITE(VIDEO,<"JA SEI QUE !">)
6250 NUMERADO:=FALSE
6260 END
6270 I:=PROXC(EPREMISA)
6280 WHILE MARCA(I) NEQ 1 DO BEGIN
6290 IF MARCA(I) = 3 THEN BEGIN
6300 DENTRODEU:=TRUE
6310 IF OU1(I)=1 THEN DENTRODEU1:=TRUE
6320 I:=PROXC(I)
6330 WHILE MARCA(I) NEQ 3 DO BEGIN
6340 DECODIFICARCLAUSULA
6350 I:=PROXC(I)
6360 NSUBCLAUSULA:=NSUBCLAUSULA+1
6370 END
6380 NSUBCLAUSULA:=1
6390 I:=PROXL(I)
6400 END
6410 ELSE BEGIN
6420 DENTRODEU:=FALSE
6430 DENTRODEU1:=FALSE
6440 DECODIFICARCLAUSULA
6450 I:=PROXC(I)
6460 END

```

```

6470:      NCLAUSULA:=NCLAUSULA+1;
6480:      END;
6490:      PREMISSA:=FALSE;
6500:      DENTRODEQUI:=FALSE;
6510:      DENTRODEQUI:=FALSE;
6520:      ENTAD:=TRUE;
6530:      NCLAUSULA:=1;
6540:      NSURCLAUSULA:=0;
6550:      I:=PROXC(ENTAD);
6560:      IF PORQUE THEN NUMERADO:=FALSE;
6570:      WHILE MARCA(I) NEQ 2 DO BEGIN
6580:        IF NOT PORQUE OR NCLAUSULA = NCONCLUSAO THEN
6590:          DECODIFICARCLAUSULA;
6600:          I:=PROXC(I);
6610:          NCLAUSULA:=NCLAUSULA+1;
6620:          END;
6630:      IF NOT PORQUE THEN WRITE(VIDEO,<72("=")>);
6640:      END DECODIFICAREGRA;
6650:
6660:  *-----*
6670:
6680:  PROCEDURE VERIFICASEEXISTE(ATRIBUTO,CNTXT,TEMVALOR,DESCONHECIDO,CONT);
6690:  INTEGER ATRIBUTO,CNTXT,CONT;
6700:  BOOLEAN TEMVALOR,DESCONHECIDO;
6710:  BEGIN
6720:    BOOLEAN SAI;
6730:
6740:    COMMENT:
6750:      ** VERIFICASEEXISTE **
6760:      VERIFICA SE UM DADO ATRIBUTO E CONTEXTO JA' TEM
6770:      VALOR NO BANCO DE FATOS. CASO EXISTA, A VARIÁVEL
6780:      "CONT" RETORNA A POSICAO DO PRIMEIRO VALOR
6790:      ASSOCIADO A ELES ;
6800:
6810:      TEMVALOR:=DESCONHECIDO:=SAI:=FALSE;
6820:      CONT:=1;
6830:      READ(DIRDISK(ATRIBUTO),10,DIRETORIO);
6840:      IF REGISTROJAEXISTE THEN BEGIN
6850:        READ(VALORCONHECIDO[CHAVEREG/2],30,CONHECIDO);
6860:        WHILE NOT SAI DO IF CONT > 30 THEN SAI:=TRUE
6870:          ELSE IF KCONTEXTO(CONT) = 0 THEN SAI:=TRUE
6880:            ELSE IF KCONTEXTO(CONT) = CNTXT THEN BEGIN
6890:              SAI:=TRUE;
6900:              TEMVALOR:=TRUE;
6910:              IF KDES(CONT) = 1 THEN DESCONHECIDO:=TRUE;
6920:              END
6930:              ELSE CONT:=CONT + 2
6940:            END;
6950:      END ;
6960:
6970:  *-----*
6980:
6990:  PROCEDURE ATUALIZAVALORESCONSULTA(A,C,V,T,U,F,PD,R,NC,NEG);
7000:  INTEGER A,C,T,U,PD,R,NC,NEG;
7010:  REAL V,F;
7020:  COMMENT:
7030:    A - NUMERO DO ATRIBUTO
7040:    C - PONTEIRO PARA ARVORE DE CONTEXTOS
7050:    V - VALOR

```



```

70600 T - TIPO DO VALOR
70700 U - UNIDADE (PRESENTE/AUSENTE)
70800 F - FATOR DE CERTEZA
70900 PD - PERGUNTADO (=1) OU DEDUZIDO (=0)
71000 R - NUMERO DA REGRA OU PERGUNTA USADA PARA DEDUZIR
71100 NC - NUMERO DA CLAUSULA
71200 NEG - VALOR POSITIVO QDO NEG = 0 ;
71300
71400 BEGIN
71500   BOOLEAN VALORPOSITIVO,TAI;
71600   REAL FANTERIOR;
71700   LABEL FIM;
71800   INTEGER I,J,FINTEIROS;
71900
72000 COMMENT:
72100   ** INICIO DE ATUALIZAVALORESCONSULTA **
72200   -----
72300   COEBCA UM NOVO VALOR PARA O ATRIBUTO E CONTEXTO
72400   DADOS E ATUALIZA O CF ACUMULADO DAQUELE VALOR ;
72500
72600
72700   & ATUALIZA ARQUIVO VALORCONHECIDO
72800   & -----
72900   & ESTE ARQUIVO SO E CHAMADO SE O VALOR E CONHECIDO,SE O
73000   & VALOR E DESCONHECIDO "KDES" E ATUALIZADO EM BUSCAVALOR.
73100
73200   IF TRACE THEN WRITE(SAIDA,</,"* ATUALIZAVALORESCONSULTA *",/
73300   " ATRIBUTO =",I3," CONTEXTO =",I3," VALOR =",
73400   F7.2," TIPOVALOR =",I2," UNIDADE =",I2,/
73500   " FATOR =",F7.2," PD =",I2," REGRA =",I3,
73600   " NCLAUSULA =",I3," NEGACAO =",I3>>A,C,V,T,
73700   U,F,PD,R,NC,NEG);
73800
73900   VALORPOSITIVO:=NEG=0;
74000   READ(DIRDISK(A),10,DIRETORIO);
74100   IF REGISTROJAEXISTE
74200   THEN READ(VALORCONHECIDO[CHAVEREG/2],30,CONHECO)
74300   ELSE FILL CONHECO[+] WITH 30(0) ;
74400   I:=1;
74500   TAI:=FALSE;
74600   WHILE KCONTEXTO(I) NEQ 0 AND NOT TAI
74700   DO IF KCONTEXTO(I) = C AND KVALOR(I) = V AND KTIPO(I) = T
74800   THEN TAI:=TRUE
74900   ELSE I:=I+2;
75000   IF NOT TAI THEN BEGIN
75100     IF A = ATRIBUTOOBJETIVO AND C = 1 THEN BEGIN
75200       PVETOR:=* + 1;
75300       VETORCONCLUSOES[PVETOR]:=I;
75400     END;
75500     KCONTEXTO(I):=C;
75600     KVALOR(I):=V;
75700     KTIPO(I):=T;
75800     KUNI(I):=U;
75900     KMB(I):=0;
76000     KMD(I):=0;
76100     KPD(I):=PD;
76200     KDES(I):=0;
76300     IF PD = 0
76400     THEN KREGRA(I):=R

```

```
7650 ELSE KNUMPERGUNTA(I):=R;
7660 END;
7670 % NAO MEXER NAS PROXIMAS LINHAS SENAO DA! PAU
7680 % POR IDIOSSINCRASIA DO BURROUGHS.
7690 FANTERIOR:=IF VALORPOSITIVO THEN KMB(I)/100 ELSE KMD(I)/100;
7700 FANTERIOR:=(F+FANTERIOR + F*FANTERIOR)*100;
7710 FINTEIRO:=FANTERIOR;
7720 IF VALORPOSITIVO THEN KMB(I):=FINTEIRO ELSE KMD(I):=FINTEIRO;
7730 REGACESSADO[CHAVEG/2]:=TRUE;
7740 WRITE(VALORCONHECIDO[CHAVEG/2],30,CONHECO);
7750 IF TRACE THEN WRITE(SAIDA,<" AO FINAL KMB =">I3,>" KMD =">I3,>
7760 " PONT =">I2,>,KMB(I),KMD(I),I);
7770 END PROCEDURE ATUALIZAVALORECONSULTA;
```

```
7780
7790 -----
7800
7810 PROCEDURE PEGA(X);
7820 STRING X;
7830 BEGIN
7840
7850 COMMENT:
7860 ** PEGA **
7870 ----
7880 PEGA A PRIMEIRA PALAVRA DA LINHA. CASO ESTA ESTEJA
7890 VAZIA, LE-SE UMA NOVA ;
7900
7910 LINHA:=TAIL(LINHA," ");
7920 IF LINHA = EMPTY THEN BEGIN
7930 LELINHA;
7940 PEGA(X);
7950 END
7960 ELSE BEGIN
7970 X:=HEAD(LINHA,NOT " ");
7980 LINHA:=TAIL(LINHA,NOT " ");
7990 END;
8000 END PEGA;
```

```
8010
8020 -----
8030
8040 BOOLEAN PROCEDURE NUMERO;
8050 BEGIN
8060 STRING X;
8070 TRUTHSET DIGITOS("0123456789");
8080
8090 COMMENT:
8100 ** NUMERO **
8110 ----
8120 VERIFICA SE A CADEIA DE CARACTERES GLOBAL "PALAVRA"
8130 E' UM NUMERO. SE FOR DEVOLVE O NRO NA VARIÁVEL "VAL" ;
8140
8150 X:=PALAVRA;
8160 NUMERO:=TRUE;
8170 IF TAKE(X,1) = "-" THEN X:=DROP(X,1);
8180 X:=TAIL(X,DIGITOS);
8190 IF LENGTH(X) = 0 THEN VAL:=DECIMAL(PALAVRA)
8200 ELSE BEGIN
8210 IF TAKE(X,1) = "." THEN X:=TAIL(DROP(X,1),DIGITOS);
8220 IF LENGTH(X) = 0 THEN VAL:=DECIMAL(PALAVRA)
8230 ELSE NUMERO:=FALSE;
```

```
82400      END;
82500      END PROCEDURE NUMERO;
82600
82700      -----
82800
82900      PROCEDURE BUSCAVALOR(X1,X2,X3,X4,X5,X6,X7,X8,X9);
83000      VALUE X1,X2,X7,X8,X9;
83100      INTEGER X1,X2,X3,X4,X5,X7,X8,X9;
83200      BOOLEAN X6;
83300      FORWARD;
83400
83500      -----
83600
83700      PROCEDURE CRIACONTEXTO(TIPOCERTO,REGRAPI);
83800      INTEGER TIPOCERTO,REGRAPI;
83900      BEGIN
84000
84100          COMMENT :
84200          ESSE PROCEDIMENTO CRIA UM NOVO NO' (Q) NA ARVORE DE CONTEXTOS,
84300          COMO DESCENDENTE DO NO' P (GLOBAL AO PROCEDIMENTO). O TIPO DE Q
84400          E' 'TIPOCERTO', OS TIPOS DE CONTEXTO POSSIVEIS PARA O PAI DE Q
84500          SAO DADOS POR 'ASSOCIADO(Q)'. CASO 'TIPO(P)' NAO ESTEJA EM
84600          'ASSOCIADO(Q)', SERA PRECISO USAR EXPANSAO IMPLICITA;
84700
84800          ARRAY ASSOC[116];
84900          INTEGER I,NUMASSOC;
85000
85100      -----
85200
85300      PROCEDURE EXPANDEARVORE(XP,XT);
85400      INTEGER XP,XT;
85500      BEGIN
85600          INTEGER I;
85700
85800          COMMENT:
85900          ** EXPANDEARVORE **
86000          -----
86100          CRIA UM NOVO CONTEXTO NA ARVORE DE CONTEXTOS E INFERE
86200          OS VALORES DE SEUS ATRIBUTOS PRINCIPAIS ;
86300
86400          IF TRACE
86500          THEN WRITE(SAIDA,</,"* EXPANDEARVORE *",/, " XP =",I3," XT =",I3>,
86600          XP,XT);
86700          I:=0;
86800          WHILE (I LEQ 49) AND (OBJETOC(I) NEQ 0)
86900          AND (OBJETOC(I) NEQ OBJETO) DO I:=I+1;
87000          IF OBJETOC(I) = OBJETO THEN NUMEROC(I):=NUMEROC(I)+1
87100          ELSE
87200              IF OBJETOC(I) = 0 THEN BEGIN
87300                  OBJETOC(I):=OBJETO;
87400                  NUMEROC(I):=1;
87500              END
87600              ELSE ERRO("NUMERO DE CONTEXTOS MUITO GRANDE");
87700          LINHA:="----- " CAT TRADUZ CAT "-" CAT STRING(NUMEROC(I),*)
87800          CAT " -----";
87900          IF NOT MUDANDO THEN MOSTREMENSAGEM(LINHA);
88000          IF FILHO(XP) = 0 THEN FILHO(XP):=AVAIL
88100          ELSE BEGIN
88200              INTEGER Q;
```

```

8830      Q:=FILHO(XP);
8840      WHILE PROXIMO(Q) NEQ 0 DO Q:=PROXIMO(Q);
8850      PROXIMO(Q):=AVAIL;
8860      END;
8870      TIPO(AVAIL):=XT;
8880      NUMCONTEXTO(AVAIL):=NUMERO(I);
8890      OBJECT(AVAIL):=OBJETO;
8900      PAI(AVAIL):=XP;
8910      PROXIMO(AVAIL):=0;
8920      AVAIL:=AVAIL+1;
8930      BDSQUEATRIBUTOSPRINCIPAIS;
8940      END;
8950
-----
8960      X-----
8970
8980      PROCEDURE EXPANSADIMPLICITA(P, TIPOFILHO);
8990      INTEGER P, TIPOFILHO;
9000      BEGIN
9010
9020          INTEGER I, FIRST, LAST, NCALL;
9030          BOOLEAN ERRO, EXPANSADFEITA;
9040          ARRAY DESCENDENCIA[1:50];
9050          DEFINE
9060              DTIPO(X) = DESCENDENCIA[X].[47:10] #,
9070              DFILHO(X) = DESCENDENCIA[X].[37:10] #,
9080              SALVAP(X) = DESCENDENCIA[X].[27:10] #,
9090              CLOSED(X) = DESCENDENCIA[X].[00:10] #;
9100
9110      X-----
9120
9130      PROCEDURE EXPANDE(NPAI, D);
9140      INTEGER NPAI, D;
9150      BEGIN
9160          STRING TRADUZ, PRIMEIRONO, NOADICIONAL;
9170          LABEL FIMEXPANDE;
9180          DEFINE RECUPERACAO =
9190              X-----
9200      IF EXPANSADFEITA THEN
9210      IF NPAI NEQ SALVAP(NCALL) THEN GO FIMEXPANDE
9220      ELSE BEGIN
9230          IF PROXIMO(FILHO(SALVAP(NCALL))) NEQ 0 THEN BEGIN
9240              AVAIL:=PROXIMO(FILHO(SALVAP(NCALL)));
9250              PROXIMO(FILHO(SALVAP(NCALL))):=0;
9260              ERRO:=FALSE;
9270              END;
9280              NCALL:=NCALL-1;
9290              GO FIMEXPANDE;
9300              FND
9310      ELSE GO FIMEXPANDE #;
9320
9330      COMMENT:
9340          ** INICIO DE EXPANDE **
9350          -----
9360      CRIA OS CONTEXTOS QUE FALTAM NA ARVORE DE CONTEXTOS
9370      PARA LIGAR O NO' RAIZ AO NO' P ;
9380
9390      IF TRACE
9400      THEN WRITE(SAIDA, <"/, "** EXPANDE **", /, " NPAI =", I3, " D =", I3, >,
9410      NPAI, D);

```

```

9420: EXPANSAOFEITA:=FALSE;
9430: LERTIPOSCONTEXTOD(TIPO(D));
9440: IF ASSUMIR = 0 THEN BEGIN
9450:   MOSTREPERGUNTA(PRIMEIROD);
9460:   LERESPOSTA(SIMNAO);
9470:   IF (RESPOSTA = "NAO") OR (RESPOSTA = "N")
9480:   THEN BEGIN
9490:     ERRO:=TRUE;
9500:     NAOTEMCONTEXTOD(TIPO(D)):=TRUE;
9510:   END;
9520: END;
9530:
9540: IF ERRO THEN RECUPERACAO;
9550: EXPANDEARVORE(NPAI,DTIPO(D));
9560: NCALL:=NCALL+1;
9570: SALVAP(NCALL):=NPAI;
9580:
9590: IF DFILHO(D) NEQ 0 THEN EXPANDE(AVAIL-1,DFILHO(D));
9600: IF ERRO THEN RECUPERACAO;
9610:
9620: EXPANSAOFEITA:=TRUE;
9630: CONTEXTOCRIADO:=TRUE;
9640: MOSTREPERGUNTA(NOADICIONAL);
9650: LERESPOSTA(SIMNAO);
9660: WHILE (RESPOSTA = "SIM") OR (RESPOSTA = "S") DO BEGIN
9670:   EXPANDEARVORE(NPAI,DTIPO(D));
9680:   IF DFILHO(D) NEQ 0 THEN EXPANDE(AVAIL-1,DFILHO(D));
9690:   IF ERRO THEN RECUPERACAO;
9700:   MOSTREPERGUNTA(NOADICIONAL);
9710:   LERESPOSTA(SIMNAO);
9720: END;
9730: NCALL:=NCALL-1;
9740: FIMEXPANDE;
9750: END PROCEDURE EXPANDE;
9760:
9770: COMMENT:
9780:   ** INICIO DE EXPANSAOIMPLICITA **
9790:   -----
9800:   INICIA A CRIACAO DE CONTEXTOS MOTIVADA PELA
9810:   TENTATIVA DE CRIACAO DE UM CONTEXTO CUJO PAI NAO
9820:   ESTA NA ARVORE ;
9830:
9840:   IF TRACE
9850:   THEN WRITE(SAIDA,<,"* EXPANSAOIMPLICITA *",/,," P =",I3,
9860:   " TIPOFILHO =",I3>>,P,TIPOFILHO);
9870:   FIRST:=1;
9880:   LAST:=1;
9890:   DTIPO(FIRST):=TIPOFILHO;
9900:   DFILHO(FIRST):=0;
9910:   CLOSED(FIRST):=0;
9920:
9930:   WHILE FIRST LEQ LAST DO BEGIN
9940:     IF CLOSED(FIRST) = 0 THEN BEGIN
9950:       LERTIPOSCONTEXTOD(TIPO(FIRST));
9960:       I:=1;
9970:       WHILE I LEQ NUMASSOCIADOS DO BEGIN
9980:         LAST:=LAST+1;
9990:         DTIPO(LAST):=ASSOCIADO(I);
10000:         DFILHO(LAST):=FIRST;

```

```

100100      CLOSED(LAST):=IE ASSOCIADO(I) = TIPO(P) THEN 1 ELSE 0;
100200      I:=I+1;
100300      END;
100400      END;
100500      FIRST:=FIRST+1;
100600      END;
100700
100800      *      APLICAR ALGORITMO DE EXPANSAO IMPLICITA PARA TODO CAMINHO
100900      *      -----
101000
101100      FOR I:=LAST STEP -1 UNTIL 1 DO
101200          IF CLOSED(I) = 1 THEN BEGIN
101300              NCALL:=0;
101400              ERRO:=FALSE;
101500              FIRST:=AVAIL;
101600              IF NOT NAOTEMCONTEXTO(DTIPO(DFILHO(I))) THEN
101700                  EXPANDE(P,DFILHO(I));
101800              IF NOT EXPANSAOFEITA THEN AVAIL:=FIRST;
101900              END;
102000      END PROCEDURE EXPANSADIMPLICITA;
102100
102200      *-----
102300
102400      PROCEDURE EXPANSADIRETA(P,TIPOFILHO);
102500      INTEGER P,TIPOFILHO;
102600      BEGIN
102700          LABEL FIMEXPANSAO;
102800
102900
103000      COMMENT:
103100          ** INICIO DE EXPANSAO DIRETA **
103200          -----
103300          ESSE PROCEDIMENTO TENTA ACRESCENTAR 'A ARVORE DE CONTEXTOS,
103400          UM NO DE TIPO 'TIPOFILHO' COMO FILHO DO NO 'P'.
103500          E IMPORTANTE NOTAR QUE A CADA EXPANSAO:
103600          * SAO BUSCADOS OS ATRIBUTOS PRINCIPAIS DO NOVO NO'
103700          * AVAIL E' O NUMERO DO PRIMEIRO NO' DISPONIVEL;
103800
103900      IF TRACE
104000      THEN WRITE(SAIDA, <"/>,"* EXPANSADIRETA ">"/>" P =",I3,
104100      " TIPOFILHO =",I3>P,TIPOFILHO);
104200
104300      LERTIPUSCONTEXTO(TIPOFILHO);
104400
104500      IF ASSUMIR = 0 THEN BEGIN
104600          MOSTREPERGUNTA(PRIMEIRONO);
104700          LERESPSTA(SIMNAO);
104800          IF (RESPOSTA = "NAO") OR (RESPOSTA = "N")
104900          THEN BEGIN
105000              NAOTEMCONTEXTO(TIPOFILHO):=TRUE;
105100              GO FIMEXPANSAO;
105200          END;
105300      END;
105400
105500      CONTEXTOCRIADO:=TRUE;
105600      EXPANDEARVORE(P,TIPOFILHO);
105700
105800      MOSTREPERGUNTA(NOADICIONAL);
105900      LERESPSTA(SIMNAO);

```

```

106000 WHILE (RESPOSTA = "SIM") OR (RESPOSTA = "S") DO BEGIN
106100   EXPANDEARVORE(P, TIPOFILHO);
106200   MOSTREPERGUNTA(NOADICIONAL);
106300   RESPOSTA(SIMNAO);
106400   END;
106500
106600 FIMEXPANSAO;
106700 END PROCEDURE EXPANSAODIRETA;
106800
106900 COMMENT:
107000   ** INICIO DE CRIACONTEXT0 **
107100   -----
107200   TENTA CRIAR UM NO' NA ARVORE DE CONTEXTOS ;
107300
107400   WRITE(SAIDA, <"NAO TEM CONTEXTO - INICIO DE CRIACONTEXT0"/>,
107500   2013, //, 20(X2, L1)) >> FOR I:=0 STEP 1 UNTIL 19 DO I, FOR
107600   I:=0 STEP 1 UNTIL 19 DO NAO TEM CONTEXTO(I));
107700   IF TRACE
107800   THEN WRITE(SAIDA, <"/>** CRIACONTEXT0 **"/>, TIPOCERTO = "I3",
107900   " REGRAPAI = "I3", TIPOCERTO, REGRAPAI);
108000
108100   IF NOT NAO TEM CONTEXTO(TIPOCERTO) THEN BEGIN
108200     LERTIPOS CONTEXTO(TIPOCERTO);
108300
108400     COMMENT:
108500     VERIFICAR SE A EXPANSAO PODE SER FEITA DIPETAMENTE, ISTO E',
108600     SE 'TIPO(P)' PERTENCE A 'ASSOCIADO(Q)';
108700
108800     CONTEXTOCRIADO:=FALSE;
108900     NUMASSOC:=NUMASSOCIADOS;
109000     FOR I:=1 STEP 1 UNTIL NUMASSOC DO ASSOC[I]:=ASSOCIADO(I);
109100
109200     IF AVAIL = 1 THEN BEGIN
109300       & EXPANSAO DO NO RAIZ
109400       EXPANDEARVORE(P, TIPOCERTO);
109500       CONTEXTOCRIADO:=TRUE;
109600       END
109700     ELSE BEGIN
109800       FOR I:=1 STEP 1 UNTIL NUMASSOC DO
109900         IF ASSOC[I] = TIPO(P) THEN EXPANSAODIRETA(P, TIPOCERTO);
110000
110100       IF NOT CONTEXTOCRIADO THEN
110200         EXPANSADIMPLICITA(TIPO(P), TIPOCERTO);
110300       END;
110400
110500       IF NOT CONTEXTOCRIADO THEN
110600         ERRO("IMPOSSIVEL EXPANSAO DA ARVORE DE CONTEXTOS");
110700       END;
110800
110900       WRITE(SAIDA, <"NAO TEM CONTEXTO - FIM DE CRIACONTEXT0"/>,
111000       2013, //, 20(X2, L1)) >> FOR I:=0 STEP 1 UNTIL 19 DO I, FOR
111100       I:=0 STEP 1 UNTIL 19 DO NAO TEM CONTEXTO(I));
111200       END PROCEDURE CRIACONTEXT0;
111300
111400   -----
111500
111600   BOOLEAN PROCEDURE PESQUISADIRETORIO;
111700   BEGIN
111800     BOOLEAN ACHEI;

```

```

11190
11200 COMMENT:
11210      ** PESQUISADIRETORIO **
11220      -----
11230      VERIFICA SE A VARIÁVEL GLOBAL "PALAVRA" É UMA
11240      PALAVRA-CHAVE DO SISTEMA: SE FOR, DEVOLVE O NRO
11250      DO SEU REGISTRO NA VARIÁVEL "L" ;
11260
11270      ACHEI:=FALSE;
11280      LI:=1;
11290      DO BEGIN
11300          READ(DIRDISK(L),10,DIRETORIO);
11310          IF PALAVRA = DIRNOME THEN ACHEI:=TRUE
11320          ELSE LI:=L+1;
11330      END
11340      UNTIL (ACHEI) OR (L = DIRDISK.LASTRECORD+1);
11350      LI:=SINONIMO;
11360      P<SQUISADIRETORIO:=ACHEI;
11370      END PESQUISADIRETORIO;
11380
11390      -----
11400
11410      PROCEDURE RESPONSTACONSISTENTE(ATR,CNTXT,DENOV,SP,SC,VALORSABIDO,
11420      FRASE,REGRAATUAL,PR,NCL,NSUBCL,NCONCLUSAO,FRAGE);
11430      VALUE ATR,CNTXT,NCL,NSUBCL,NCONCLUSAO;
11440      INTEGER ATR,CNTXT,REGRAATUAL,PR,NCL,NSUBCL,NCONCLUSAO,SP,SC,FRAGE;
11450      BOOLEAN VALORSABIDO,DENOV;
11460      STRING FRASE;
11470      BEGIN
11480
11490          COMMENT :
11500          ATR = ATRIBUTO CORRENTE
11510          CNTXT = PONTEIRO PARA A ARVORE DE CONTEXTOS
11520
11530          SINTAXE DOS VALORES ACEITAVEIS :
11540          -----
11550          <VALOR ACEITAVEL> ::= <VALOR> <FATOR>
11560          <VALOR> ::= SIM/NAO/<NOME>/<NUMERO>/<NUMERO><UNIDADE>
11570          <FATOR> ::= (<NUMERO>)/EMPTY;
11580
11590      FILE VIDEO(KIND=PEMOT,UNITS=CHARACTERS);
11600      INTEGER TIPOVALOR,I,NOATUAL,UNIDADENUM,TIPOAT,NEGADO;
11610      BOOLEAN ACHEI,ACABOU,DENTRODEPORQUE,ACHOU,SAI,FATORERRADO;
11620      STRING LISTA,VALOR,VALORLIDO,UNIDADELIDA,COMANDO,COMANDOS,
11630      PRIMEIRAPALAVRA,NOMEATRIBUTO,CADEIA;
11640      REAL VALORNUM,FATOR;
11650
11660      DEFINE PEGAVALOR =
11670      -----
11680
11690      BEGIN
11700          DEFINE PEGAPALAVRA =
11710          BEGIN
11720              LISTA:=TAIL(LISTA," ");
11730              IF LISTA NEQ EMPTY THEN
11740                  IF TAKE(LISTA,1) NEQ "/" THEN BEGIN
11750                      VALOR:=HEAD(LISTA,NOT " ");
11760                      LISTA:=TAIL(LISTA,NOT " ");
11770                      END

```



```

11780 ELSE BEGIN
11790   STRING CADEIA;
11800   CADEIA:=HEAD(DROP(LISTA,1),NOT "/");
11810   LISTA:=TAIL(TAIL(DROP(LISTA,1),NOT "/"),"/");
11820   VALOR:=EMPTY;
11830   DO BEGIN
11840     PALAVRA:=HEAD(TAIL(CADEIA," "),NOT " ");
11850     CADEIA:=TAIL(TAIL(CADEIA," "),NOT " ");
11860     IF PALAVRA NEQ EMPTY
11870     THEN VALOR:=VALOR CAT " " CAT PALAVRA
11880     ELSE VALOR:=DROP(VALOR,1);
11890   END
11900   UNTIL PALAVRA = EMPTY;
11910 END;
11920 END #;
11930 PEGAPALAVRA;
11940 IF VALOR NEQ EMPTY THEN
11950   IF TAKE(VALOR,1) = "(" THEN BEGIN
11960     VALOR:=VALOR CAT " " CAT HEAD(LISTA,NOT ")") CAT ")";
11970     LISTA:=TAIL(TAIL(LISTA,NOT ")"),");");
11980   END
11990   ELSE IF VALOR = "UMDOS" THEN BEGIN
12000     PEGAPALAVRA;
12010     VALOR:="UMDOS" CAT VALOR;
12020   END;
12030 END #;
12040
12050 -----
12060
12070
12080 PEGAPALAVRA =
12090 -----
12100 BEGIN
12110   PALAVRA:=HEAD(TAIL(LINHA," "),NOT " ");
12120   LINHA:=TAIL(TAIL(LINHA," "),NOT " ");
12130   IF PALAVRA NEQ EMPTY
12140   THEN IF NOT NUMERO THEN BEGIN
12150     SAI:=FALSE;
12160     CADEIA:=PALAVRA;
12170     WHILE NOT SAI DO BEGIN
12180       PALAVRA:=HEAD(TAIL(LINHA," "),NOT " ");
12190       IF PALAVRA = EMPTY THEN PALAVRA:="(";
12200       IF TAKE(PALAVRA,1) = "(" THEN BEGIN
12210         SAI:=TRUE;
12220         PALAVRA:=CADEIA;
12230       END
12240       ELSE BEGIN
12250         LINHA:=TAIL(TAIL(LINHA," "),NOT " ");
12260         CADEIA:=CADEIA CAT " " CAT PALAVRA;
12270       END;
12280     END;
12290   END;
12300 END #;
12310
12320 MOSTRAVALORESPOSSIVEIS =
12330 -----
12340 BEGIN
12350   INTEGER X;
12360   X:=0;

```

```

123700 IF DENOVO THEN LISTA:="SIM NAO S N" ELSE LISTA:=ESPERADO;
123800 WRITE(VIDEO,</>,"** VALORES ESPERADOS PARA O ATRIBUTO ",
123900 A**" </>LENGTH(NOMEATRIBUTO)NOMEATRIBUTO))
124000 DO BEGIN
124100 PEGAVALOR;
124200 XI:=XI+1;
124300 WRITE(VIDEO,<I3,". ">A*>X,LENGTH(VALOR),VALOR) END
124400 UNTIL TAIL(LISTA," ") = EMPTY;
124500 ACABOU:=FALSE;
124600 WRITE(VIDEO,</>);
124700 END #;
124800
124900 DEFINE VERIFICAVALORACEITAVEL =
125000 -----
125100
125200 BEGIN
125300 STRING PALAVRA;
125400 ACHEI:=FALSE;
125500 IF TIPOVALOR = NAONUMERICO THEN
125600 DO BEGIN
125700 PEGAVALOR;
125800 IF VALOR NEQ EMPTY THEN
125900 IF HEAD(TAIL(VALOR," "),NOT " ") NEQ "UMDOS"
126000 THEN IF VALOR = VALORLIDO THEN ACHEI:=TRUE ELSE
126100 ELSE % TRATAR CLASSE
126200 END
126300 UNTIL ACHEI OR TAIL(LISTA," ") = EMPTY
126400 ELSE DO BEGIN
126500 PEGAVALOR;
126600 IF VALOR NEQ EMPTY THEN
126700 IF TAKE(VALOR,1) = "(" THEN BEGIN
126800 VALOR:=HEAD(DROP(VALOR,1),NOT " ");
126900 IF HEAD(TAIL(VALOR," "),NOT " ") = "CONSTANTES" THEN
127000 WRITE(VIDEO,"AINDA NAO ESTOU TRATANDO CONST")
127100 ELSE BEGIN
127200 % NAO ESTA TRATANDO UNIDADE
127300 PALAVRA:=HEAD(HEAD(TAIL(VALOR," "),NOT " "),NOT " ");
127400 VALOR:=TAIL(TAIL(TAIL(VALOR," "),NOT " "),NOT " ");
127500 IF DECIMAL(VALORLIDO) GEQ DECIMAL(PALAVRA) THEN
127600 BEGIN
127700 PALAVRA:=HEAD(TAIL(VALOR," "),NOT " ");
127800 IF DECIMAL(VALORLIDO) LEQ DECIMAL(PALAVRA)
127900 THEN ACHEI:=TRUE;
128000 END;
128100 END;
128200 END;
128300 END
128400 UNTIL ACHEI OR TAIL(LISTA," ") = EMPTY;
128500 END #;
128600
128700 -----
128800
128900 BOOLEAN PROCEDURE PEGAFATOR;
129000 BEGIN
129100
129200 COMMENT:
129300 ** PEGAFATOR **
129400 -----
129500 TOMA O FATOR DE CERTEZA NA LINHA: CASO O FATOR FOR

```

```

129600      ONTID05, ELE SERA' FEITO = 1 ;
129700
129800      PEGAFATOR:=FALSE;
129900      LINHA:=TAIL(LINHA," ");
130000      PEGAPALAVRA;
130100      IF PALAVRA = EMPTY THEN BEGIN
130200          PEGAFATOR:=TRUE;
130300          FATOR:=1;
130400          END
130500      ELSE
130600          IF TAKE(PALAVRA,1) = "(" THEN BEGIN
130700      LINHA:=PALAVRA CAT " " CAT LINHA;
130800      PALAVRA:=HEAD(HEAD(TAIL(DROP(LINHA,1)," "),NOT " "),NOT " ");
130900          IF NUMERO THEN
131000              IF VAL < 0 OR VAL > 1 THEN FATORERRADO:=TRUE
131100              ELSE BEGIN
131200          PEGAFATOR:=TRUE;
131300          FATOR:=VAL;
131400          END
131500      END;
131600      END PROCEDURE PEGAFATOR;
131700
131800      -----
131900
132000      PROCEDURE EXPLANACAO;
132100      BEGIN
132200          ARRAY REFERENCIAS[0:179];
132300          INTEGER CONTEXTO,NREGRA,ATRIBUTO,ATRIBUTO1,ATRIBUTO2,ENDATRIBUTO;
132400          BOOLEAN DEUCERTO;
132500          REAL VALORN;
132600      -----
132700      DEFINE PEGAUMAPALAVRA =
132800      -----
132900          BEGIN
133000              PALAVRA:=HEAD(TAIL(LINHA," "),NOT " ");
133100              LINHA:=TAIL(TAIL(LINHA," "),NOT " ");
133200          END #;
133300
133400      PEGAPALAVRALINHA =
133500      -----
133600          BEGIN
133700              IF TAIL(LINHA," ") = EMPTY THEN LELINHA;
133800              PEGAUMAPALAVRA;
133900          END #;
134000
134100      PEGATEPMO(LIMITE) =
134200      -----
134300          BEGIN
134400              CADEIA:=EMPTY;
134500              SAI:=FALSE;
134600              DO BEGIN
134700                  PEGAPALAVRALINHA;
134800                  IF PALAVRA = EMPTY OR PALAVRA = LIMITE THEN BEGIN
134900                      SAI:=TRUE;
135000                      LINHA:=PALAVRA CAT " " CAT LINHA;
135100                      END
135200                  ELSE CADEIA:=CADEIA CAT " " CAT PALAVRA
135300                  END
135400              UNTIL SAI;

```

```

13550:         IF CADEIA NEQ EMPTY THEN PALAVRA:=DROP(CADEIA,1);
13560:     END #;
13570:
13580: ERRO1(TEXT0) =
13590: $-----
13600:     BEGIN
13610:         STRING X;
13620:         IF NOT MUDANDO THEN BEGIN
13630:             X:="** ERRO , " CAT TEXT0 ;
13640:             ERRO(X);
13650:         END;
13660:     END #;
13670:
13680: TIRAINTERROGACAO =
13690: $-----
13700:     PRIMEIRAPALAVRA:=HEAD(TAIL(LINHA," "),NOT " ");
13710:     IF PRIMEIRAPALAVRA = "?" THEN PEGAPALAVRALINHA #;
13720:
13730:     TRATAVALOR =
13740:     $-----
13750:     IF PALAVRA = EMPTY THEN DEUCERTO:=FALSE;
13760:     VALORN:=0;
13770:     IF DEUCERTO THEN
13780:         IF NUMERO THEN BEGIN
13790:             TIPOVALOR:=NUMERICO;
13800:             VALORN:=VAL;
13810:         END
13820:     ELSE BEGIN
13830:         TIPOVALOR:=NAONUMERICO;
13840:         IF TIPOAT NEQ 0 THEN
13850:             IF PESQUISADIRETORIO THEN VALORN:=L
13860:             ELSE DEUCERTO:=FALSE;
13870:         END#;
13880:
13890: LISTAREFERENCIAS =
13900: $-----
13910:     IF ENDATRIBUTO GEQ 1 AND
13920:     ENDATRIBUTO LEQ ATRIBUTOSREF*LASTRECORD THEN BEGIN
13930:         INTEGER I;
13940:         WRITE(VIDEO,<///  

13950:         READ(ATRIBUTOSREF[ENDATRIBUTO],180,REFERENCIA);
13960:         I:=RLINK(0);
13970:         LINE:=EMPTY;
13980:         WHILE I NEQ 0 DO BEGIN
13990:             LINE:=LINE CAT "REGRA " CAT
14000:             STRING(NUMREGRA(I),*) CAT " ";
14010:             I:=RLINK(I);
14020:         END;
14030:         IMPRIMELINHA;
14040:     END;
14050:     WRITE(VIDEO,<72("=")>) #;
14060:
14070: LISTACOMANDOS =
14080: $-----
14090:     BEGIN
14100:         INTEGER I,J;
14110:         WRITE(VIDEO,<72("="),///  

14120:         <72("=")>);
14130:         FOR I:=1 STEP 1 UNTIL 65 DO

```

```

141400 WRITE(VIDEO,'<12A6> FOR J1=0 STEP 1 UNTIL 11 DO
141500 TABELA[I,J]);
141600 WRITE(VIDEO,'</72("=")>');
141700 END$;
141800 -----
141900
142000 PROCEDURE PORQUE;
142100 BEGIN
142200   BOOLEAN SAI;
142300   ARRAY REGRA[0:149];
142400   BOOLEAN PROCEDURE PODEMOSTRARAREGRA;
142500   BEGIN
142600     READ(REGRADISK[RREGRA(NOATUAL)],150,REGRA);
142700     IF FMOSTRAREGRA = 0 THEN BEGIN
142800       SAI:=TRUE;
142900       PODEMOSTRARAREGRA:=TRUE;
143000     END
143100     ELSE PODEMOSTRARAREGRA:=FALSE;
143200   END;
143300
143400   COMMENT:
143500     ** INICIO DE PORQUE **
143600     -----
143700     EXPLICA O PORQUE DA ULTIMA PERGUNTA FEITA AO USUARIO,
143800     DECODIFICANDO A CLAUSULA DE ONDE SE ORIGINOU A PERGUNTA,
143900     E MOSTRANDO-A. SE ESTE COMANDO FOR CHAMADO NOVAMENTE, O
144000     SISTEMA DECODIFICA A REGRA ACIMA DESTA EM REGRACONSULTA ;
144100
144200   WRITE(VIDEO,'<//>');
144300   IF NOT DENTRODEPORQUE THEN BEGIN
144400     DENTRODEPORQUE:=TRUE;
144500     NOATUAL:=PR;
144600   END
144700   ELSE NOATUAL:=RPAI(NOATUAL);
144800   SAI:=FALSE;
144900   WHILE NOT SAI DO BEGIN
145000     IF PODEMOSTRARAREGRA THEN
145100       IF RTIPO(NOATUAL) = 0 THEN BEGIN
145200         SAI:=TRUE;
145300         READ(DIRDISK[RATRIBUTO(NOATUAL)],10,DIRETORIO);
145400         LINE:="ESTOU TENTANDO DEDUZIR O ATRIBUTO " CAT DIRNOME
145500           CAT " NA REGRA " CAT STRING(RREGRA(NOATUAL));
145600         IMPRIMELINHA;
145700         DECODIFICAREGRA(RREGRA(NOATUAL),1,RCL(NOATUAL),RSUBCL(NOATUAL),
145800           NCONCLUSAO);
145900         WRITE(VIDEO,'<//>');
146000       END
146100     ELSE: % CONTEXTO
146200     IF RATRIBUTO(RPAI(NOATUAL)) = 0 THEN BEGIN
146300       DENTRODEPORQUE:=FALSE;
146400       WRITE(VIDEO,'<//>');
146500       LINE:="FIM DE ENCADEAMENTO DE 'PORQUE'. A PROXIMA CHAMADA DO COMAND
146600 O REINICIARA O ENCADEAMENTO .";
146700       MOSTREMENSAGEM(LINE);
146800     END
146900     ELSE IF NOT SAI THEN NOATUAL:=RPAI(NOATUAL);
147000   END;
147100 END;
147200

```

```

147300  *-----
147400
147500 PROCEDURE COMOCONCLUIU(ATRIBUTO,CONTEXTO,VALOR,TIPOVALOR);
147600 INTEGER ATRIBUTO,CONTEXTO,TIPOVALOR;
147700 REAL VALOR;
147800 BEGIN
147900   REAL CFA,KMBA,KMDA,CFREGRA;
148000   INTEGER RULE,I;
148100   BOOLEAN TAI,SIMNAO;
148200   STRING CADEIA,FRASE,CVALOR,CCONTEXTO,CATRIBUTO;
148300
148400   COMMENT:
148500     **  COMOCONCLUIU  **
148600     -----
148700     PEGA EM VALORCONHECIDO A PRIMEIRA REGRA QUE CONCLUIU
148800     TAL VALOR PARA ESTES ATRIBUTO E CONTEXTO. A SEGUIR
148900     PERCORRE O ARQUIVO REGRACONSULTA ATRAVES DO LINK
149000     "PROXIMO", IMPRIMINDO QUAIS REGRAS CONCLUIRAM O
149100     VALOR E COM QUAL CF ;
149200
149300   * PERCORRER VALORCONHECIDO ATE ACHAR (CONTEXTO,VALOR,TIPOVALOR)
149400   *   ARMZENADOS NA POSICAO I *
149500   *
149600   READ(DIRDISK(ATRIBUTO),10,DIRETORIO);
149700   IF NOT REGISTROJAEXISTE
149800   THEN MOSTREMENSAGEM("ESTE ATRIBUTO AINDA NAO TEM NENHUM VALOR")
149900   ELSE BEGIN
150000     READ(VALORCONHECIDO[CHAVEREG/2],30,CONHECO);
150100     I:=1;
150200     TAI:=FALSE;
150300     WHILE NOT TAI AND KCONTEXTO(I) NEQ 0 AND I < 30 DO
150400       IF KCONTEXTO(I) = CONTEXTO AND KVALOR(I) = VALOR AND KTIPO(I) =
150500       TIPOVALOR THEN TAI:=TRUE ELSE I:=I+2;
150600     IF NOT TAI THEN MOSTREMENSAGEM(
150700     "ESTE VALOR NAO FOI ASSOCIADO A ESTE ATRIBUTO E CONTEXTO")
150800     ELSE BEGIN
150900       *
151000       *   MONTAR AS CADEIAS CATRIBUTO, CVALOR E CCONTEXTO PARA A
151100       *   IMPRESSAO.
151200       *
151300       SIMNAO:=FALSE;
151400       CATATRIBUTO:=DIRNOME;
151500       IF TIPOATRIBUTO = 0 THEN SIMNAO:=TRUE
151600       ELSE CASE ASSOC OF
151700         BEGIN
151800           * TRATAMENSAGEM
151900           0: READ(TEXTODISK(VALOR),90,CLAUZE);
152000             CVALOR:="" CAT HEAD(STRING(POINTER(CLAUSE),540),NOT " ");
152100             CAT " ";
152200           * TRATA NUMERO SEM UNIDADE
152300           1: CVALOR:=STRING(VALOR,*);
152400           * TRATA CADEIA
152500           2: READ(DIRDISK(VALOR),10,DIRETORIO);
152600             CVALOR:=DIRNOME;
152700           * TRATA NUMERO COM UNIDADE
152800           ELSE;
152900             READ(DIRDISK(ASSOC),10,DIRETORIO);
153000             CVALOR:=STRING(VALOR,*) CAT " " CAT DIRNOME;
153100         END;

```

```
153200 READ(DIRDISK(OBJECT(CONTEXTO)),10,DIRETORIO);
153300 CCONTEXTO:=DIRNOME CAT " " CAT STRING(NUMCONTEXTO(CONTEXTO),**);
153400 X
153500 X
153600 X IMPRIME AS REGRAS QUE CONCLUIPAM (ATRIBUTO,CONTEXTO,VALOR)
153700 X
153800 IF SIMNAO
153900 THEN CADEIA:="PARA CONCLUIR A VALIDADE DE " CAT CATRIBUTO
154000 CAT " DE " CAT CCONTEXTO CAT " UTILIZOU-SE ";
154100 ELSE CADEIA:="PARA CONCLUIR QUE " CAT CATRIBUTO CAT " DE "
154200 CAT CCONTEXTO CAT " E " CAT CVALOR CAT " UTILIZOU-SE ";
154300 MOSTREMENSAGEM(CADEIA);
154400 CFA:=KMBA:=KMDA:=0;
154500 IF KPD(1) = 1
154600 THEN BEGIN
154700 CFA:=KMB(1) - KMD(1);
154800 CADFIA:="RESPOSTA 'A PERGUNTA DE NUMERO " CAT
154900 STRING(KNUMPERGUNTA(1),**) CAT " (FC = " CAT
155000 STRING(CFA/100,**) CAT ")";
155100 MOSTREMENSAGEM(CADEIA);
155200 END
155300 ELSE BEGIN
155400 RULE:=KREGRA(1);
155500 WHILE RULE NEO 0 DO BEGIN
155600 IF RSUCESSO(RULE) = 0 AND VALOR = RVALOR(RULE)
155700 THEN BEGIN
155800 CFREGRA:=RCERTEZA(RULE);
155900 IF RNEGACAO(RULE) = 0
156000 THEN KMBA:=CFREGRA + KMBA - CFREGRA*KMBA
156100 ELSE BEGIN
156200 KMDA:=CFREGRA + KMDA - CFREGRA*KMDA;
156300 CFREGRA:= - CFREGRA;
156400 END;
156500 CFA:=KMBA - KMDA;
156600 CADEIA:="REGRA " CAT STRING(RREGRA(RULE),**) CAT " ( " CAT
156700 STRING(CFREGRA/100,**) CAT " ) " CAT "RESULTANDO NUM FC "
156800 CAT "ACUMULADO DE " CAT STRING(CFA/100,**)
156900 MOSTREMENSAGEM(CADEIA);
157000 END;
157100 RULE:=RPROXIMO(RULE);
157200 END;
157300 END;
157400 END;
157500 END;
157600 END;
157700 X-----X
157800 PROCEDURE MUDAR(ATRIBUTO,CONTEXTO);
157900 INTEGER ATRIBUTO,CONTEXTO;
158000 BEGIN
158100 INTEGER ATR,CONT;
158200 BOOLEAN TEMVALOR,DESCONECIDO;
158300
158400 COMMENT:
158500 ** MUDAR **
158600 C-----
158700 PROCURA QUAL PERGUNTA ATUALIZOU TAL ATRIBUTO E
158800 CONTEXTO E ARMAZENA EM "PERGUNTAMUDADA". OS
158900 ARQUIVOS DINAMICOS SAO LIMPOS E O PROCESSAMENTO
159000 REINICIADO NO PROGRAMA PRINCIPAL. AO INVES DE
```

```

159100 FAZER AS PERGUNTAS QUE ANTECEDEM "PERGUNTAMUDADA"
159200 O SISTEMA TOMA OS VALORES DO ARQUIVO DE RESPOSTAS.
159300 AO CHEGAR EM PERGUNTAMUDADA, O SISTEMA VOLTA A
159400 INQUIRIR O USUARIO (MUDANCA FEITA NA ROTINA
159500 RESPOSTA CONSISTENTE) ;
159600
159700      % ACHA O REGISTRO REFERENTE A ATRIBUTO E CONTEXTO
159800      % E VE SE O VALOR FOI PERGUNTADO
159900      VERIFICASEEXISTE(ATRIBUTO,CONTEXTO,TEMVALOR,
160000      DESCONHECIDO,CONT);
160100      IF NOT TEMVALOR OR DESCONHECIDO THEN
160200      ERRO1("AINDA NAO FORAM ATUALIZADOS ATR E CNTXT")
160300      ELSE IF KPD(CONT) = 0      % DEDUZIDO
160400      THEN ERRO1("NAO IMPLEMENTADO NESTA VERSAO")
160500      ELSE BEGIN
160600          % ACHA O NUMERO DA PERGUNTA E LIMPA
160700          % ARQUIVOS DINAMICOS REINICIANDO O
160800          % PROCESSAMENTO
160900          MUDANDO:=TRUE;
161000          ULTIMALINHA:=0;
161100          PERGUNTAMUDADA:=KNUMPERGUNTA(CONT);
161200          GO REINICIA;
161300      END;
161400  END;
161500
161600 -----
161700
161800 PROCEDURE MOSTRAVALOR(QUALATRIB,CONTEXTO);
161900 INTEGER QUALATRIB,CONTEXTO;
162000 BEGIN
162100
162200     ARRAY ARRANJO[1:15,0:1];
162300     INTEGER I,J,CONTADOR,CF,QUALASSOC,QUALTIPO;
162400     BOOLEAN DESCONHECIDO;
162500     STRING LINHA;
162600
162700 COMMENT:
162800     ** MOSTRAVALOR **
162900     -----
163000     IMPRIME TODOS OS VALORES PARA TAIS ATRIBUTO E
163100     CONTEXTO, ORDENADOS POR CF CRESCENTE ;
163200
163300     % MONTA PARTE COMUM DA LINHA
163400
163500     READ(DIRDISK(OBJECT(CONTEXTO)),10,DIRETORIO);
163600     LINHA:=" PARA O CONTEXTO " CAT DIRNOME;
163700     IF NUMCONTEXTO(CONTEXTO) NEQ 0 THEN
163800         LINHA:=LINHA CAT "-" CAT STRING(NUMCONTEXTO(CONTEXTO),*);
163900     READ(DIRDISK(QUALATRIB),10,DIRETORIO);
164000     LINHA:= "VALORES DO ATRIBUTO " CAT DIRNOME CAT LINHA;
164100     QUALASSOC:=ASSOC;
164200     QUALTIPO:=TIPOATRIBUTO;
164300
164400     % ODO ACHA O PRIMEIRO VALOR IMPRIME A PARTE COMUM
164500
164600     I:=1;
164700     CONTADOR:=0;
164800     DESCONHECIDO:=FALSE;
164900     IF REGISTROJAEXISTE

```



```

16500 THEN READ(VALORCONHECIDO[CHAVEREG/2],30,CONHECO)
16510 ELSE FILL CONHECO[*] WITH 30(0)
16520 FOR J:=1 STEP 1 UNTIL 15 DO ARRANJO[J,1]:=-1000 ;
16530 WHILE KCONTEXTO(I) NEQ 0 AND I<30 DO BEGIN
16540   IF KCONTEXTO(I) = CONTEXTO THEN
16550     IF KDES(I) = 1 THEN BEGIN
16560       LINHA:="** O SISTEMA E O USUARIO DESCONHECEM OS "
16570       CAT LINHA;
16580       MOSTREMENSAGEM(LINHA);
16590       DESCONHECIDO:=TRUE;
16600       I:=30;
16610     END
16620   ELSE BEGIN
16630     % ORDENA OS VALORES POR CF DECRESCENTE
16640     CF:=KMB(I) - KMD(I);
16650     CONTADOR:=1;
16660     WHILE CONTADOR < 15 AND ARRANJO[CONTADOR,1] GFO CF
16670       DO CONTADOR:=* + 1;
16680     FOR J:=14 STEP -1 UNTIL CONTADOR DO BEGIN
16690       ARRANJO[J + 1,0]:=ARRANJO[J,0];
16700       ARRANJO[J + 1,1]:=ARRANJO[J,1];
16710     END;
16720     ARRANJO[CONTADOR,0]:=I;
16730     ARRANJO[CONTADOR,1]:=CF;
16740   END;
16750   Y:=I+2;
16760   FND;
16770 IF CONTADOR = 0 THEN BEGIN
16780   LINHA:="** AINDA NAO EXISTE " CAT LINHA CAT " ARMAZENADOS ";
16790   MOSTREMENSAGEM(LINHA);
16800   FND
16810 ELSE IF NOT DESCONHECIDO THEN BEGIN
16820   % IMPRIME A PARTE COMUM E TRATA CADA VALOR
16830
16840   LINHA:="** " CAT LINHA;
16850   MOSTREMENSAGEM(LINHA);
16860   CONTADOR:=1;
16870   WHILE ARRANJO[CONTADOR,0] NEQ 0 AND CONTADOR <= 15 DO BEGIN
16880     % VERIFICA O TIPO DO VALOR E O PREPARA PARA IMPRESSAO
16890
16900
16910     I:=ARRANJO[CONTADOR,0];
16920     CASE QUALASSOC OF BEGIN
16930       0: IF QUALTIPO = 0 THEN
16940         IF ARRANJO[CONTADOR,1] < 0 THEN BEGIN
16950           ARRANJO[CONTADOR,1]:=-ARRANJO[CONTADOR,1];
16960           LINE:="FALSO";
16970         END
16980         ELSE LINE:="VERDADEIRO"
16990       ELSE READ(TEXTODISK[KVALOR(I)],<A90>,LINE);
17000       1: LINE:=STRING(KVALOR(I),*);
17010       2: READ(DIRDISK[KVALOR(I)],10,DIRETORIO);
17020       LINE:=DIRNOME;
17030     ELSE: READ(DIRDISK[QUALASSOC],10,DIRETORIO);
17040     LINE:=STRING(KVALOR(I),*) CAT " " CAT DIRNOME;
17050   END;
17060
17070   LINE:=STRING(CONTADOR,*) CAT ". " CAT LINE CAT "( " CAT
17080   STRING(ARRANJO[CONTADOR,1]/10,*) CAT " )";

```

```

170900      IF CONTADOR < 10 THEN LINHA:= " " CAT LINHA;
171000      IMPRIMELINHA;
171100      CONTADOR:=* + 1;
171200      END;
171300      END;
171400      END;
171500      *-----
171600
171700      PROCEDURE TRATACONTEXTOS;
171800      BEGIN
171900          STRING IDENT,NUMERICO;
172000          INTEGER X,NCONTEXT;
172100          BOOLEAN ENCONTREI;
172200
172300      COMMENT:
172400          ** TRATACONTEXTOS **
172500          -----
172600          VERIFICA SE O STRING "PALAVRA" E' UM CONTEXTO, E
172700          SENDO, SE ELE ESTA' PRESENTE NA ARVORE DE CONTEXTOS.
172800          CASO ESTEJA, A VARIAVEL GLOBAL "CONTEXTO", RETORNA
172900          SEU REGISTRO NA ARVORE ;
173000
173100
173200      CONTEXTO:=0;
173300      NUMERICO:=TAIL(TAIL(PALAVRA,NOT "-"),"-");
173400      PALAVRA:=HEAD(PALAVRA,NOT "-");
173500      IF PESQUISADIRETORIO AND CLASSE = 2 THEN BEGIN
173600          IF NUMERICO = EMPTY THEN PALAVRA:="1" ELSE PALAVRA:=NUMERICO;
173700          IF NUMERO THEN BEGIN
173800              NCONTEXT:=DECIMAL(PALAVRA);
173900              * PROCURA O CONTEXTO NA ARVORE DE CONTEXTOS
174000              ENCONTREI:=FALSE;
174100              XI:=0;
174200              WHILE NOT ENCONTREI AND X < AVAIL DO BEGIN
174300                  XI=X + 1;
174400                  IF OBJECT(X) = L AND NUMCONTEXTO(X) = NCONTEXT
174500                      THEN ENCONTREI:=TRUE;
174600                  END;
174700                  IF ENCONTREI THEN CONTEXTO:=X;
174800              FND
174900          END;
175000      END;
175100      *-----
175200
175300      PROCEDURE DESCENDENCIA(X);
175400      VALUE X;
175500      INTEGER X;
175600      BEGIN
175700          STRING LINHA;
175800          INTEGER I;
175900
176000      COMMENT:
176100          ** DESCENDENCIA **
176200          -----
176300          IMPRIME TODOS OS CONTEYTOS ABAIXO DE "X" NA ARVORE
176400          DE CONTEXTOS ;
176500
176600          LINHA:= " (PAI = " CAT DIRNOME CAT "-" CAT STRING(NUMCONTEXTO(X),*)
176700          CAT ")" ;

```

```

176800     FOR I:=1 STEP 1 UNTIL (AVAIL = 1) DO
176900         IF PAI(I) = X THEN BEGIN
177000             READ(DIRDISK(OBJECT(I)),10,DIRETORIO);
177100             LINE:=DIRNOME CAT "-" CAT STRING(NUMCONTEXTO(I),*) CAT LINHA;
177200             IMPRIMELINHA;
177300             DESCENDENCIA(I);
177400             END;
177500     END;
177600 *-----*
177700     PROCEDURE ARVORE;
177800     BEGIN
177900         COMMENT:
178000             ** ARVORE **
178100             -----
178200             IMPRIME TODA A ARVORE DE CONTEXTOS ;
178300             READ(DIRDISK(OBJECT(1)),10,DIRETORIO);
178400             WRITE(VIDEO,</>);
178500             LINE:=DIRNOME CAT "-1" CAT " (NO' RAIZ)";
178600             IMPRIMELINHA;
178700             DESCENDENCIA(1);
178800             END;
178900 *-----*
179000     PROCEDURE DESCENDENTES(X);
179100     INTEGER X;
179200     BEGIN
179300         COMMENT:
179400             ** DESCENDENTES **
179500             -----
179600             FAZ IMPRIMIR TODOS OS CONTEXTOS ABAIXO DE "X" NA
179700             ARVORE DE CONTEXTOS ;
179800             READ(DIRDISK(OBJECT(X)),10,DIRETORIO);
179900             LINE:="DESCENDENTES DE " CAT DIRNOME CAT "-" CAT
180000             STRING(NUMCONTEXTO(X),*);
180100             IMPRIMELINHA;
180200             WRITE(VIDEO,</>);
180300             DESCENDENCIA(X);
180400             END;
180500 *-----*
180600     PROCEDURE ANCESTRAIS(X);
180700     INTEGER X;
180800     BEGIN
180900         INTEGER I;
181000         COMMENT:
181100             ** ANCESTRAIS **
181200             -----
181300             IMPRIME TODOS OS CONTEXTOS ACIMA DE "X" NA ARVORE DE
181400             CONTEXTOS ;
181500             READ(DIRDISK(OBJECT(X)),10,DIRETORIO);
181600             LINE:="ANCESTRAIS DE " CAT DIRNOME CAT "-" CAT
181700             STRING(NUMCONTEXTO(X),*);
181800             IMPRIMELINHA;
181900             WRITE(VIDEO,</>);
182000             ANCESTRAIS(PAI(X));
182100             END;
182200
182300

```

```

18240: IMPRIMELINHA;
18250: WRITE(VIDEO,</>);
18260: I:=PAI(X);
18270: WHILE I NEO 0 DO BEGIN
18280:   READ(DIRDISKIOBJECT(I),10,DIRETORIO);
18290:   LINE:=DIRNOME CAT "-" CAT STRING(NUMCONTEXTO(I),*);
18300:   IMPRIMELINHA;
18310:   I:=PAI(I);
18320: END;
18330: END;
18340: -----
18350: PROCEDURE RESPOSTAS;
18360: BEGIN
18370:
18380:   COMMENT:
18390:     ** RESPOSTAS **
18400:     -----
18410:     ENCONTRA E ESCRIBE OS VALORES POSSIVEIS PARA O ATRIBUTO ;
18420:
18430:     READ(DIRDISK(ATR),10,DIRETORIO);
18440:     NOMFATRIBUTO:=DIRNOME;
18450:     READ(PROPATRIBUTOS(CHAVEREG/2),70,PROPRIETADE);
18460:     MOSTRAVALORESPOSSIVEIS;
18470: END;
18480: -----
18490:
18500: PROCEDURE SE(ATRIBUTO1,ATRIBUTO2,CONTEXTO,VALOR,
18510:   TIPOVALOR);
18520: INTEGER ATRIBUTO1,ATRIBUTO2,CONTEXTO,TIPOVALOR;
18530: REAL VALOR;
18540: BEGIN
18550:   ARRAY LIMPO[1:30],SALVAARVORECONTEXTOS[0:127],
18560:   SALVAREGRACONSULTA[0:3,0:999],SALVACONCLUSOES[0:99],
18570:   SALVAVALORCONHECIDO[1:30,0:99],SALVALASTCNTXT[0:99],
18580:   SALVAANSWER[1:200,0:11];
18590:   BOOLEAN ARRAY SALVANAOTEMCONTEXTO[0:99],
18600:   SALVAPEGACESSADO[0:99];
18610:   INTEGER I,J,SALVAP,SALVAPREGRA,SALVAAVAIL,SALVACOUNTER,
18620:   SALVAULTIMALINHA,SALVANUMPERGUNTA;
18630:
18640:   COMMENT:
18650:     ** INICIO DE SE **
18660:     -----
18670:     SALVA OS ARQUIVOS DINAMICOS E OS PONTEIROS DO
18680:     SISTEMA. LIMPA OS ARQUIVOS COM EXCECAO DA ARVORE
18690:     DE CONTEXTOS. ATUALIZA-SE O VALOR DE ATRIBUTO1 EM
18700:     VALORCONHECIDO E BUSCA-SE O VALOR DE ATRIBUTO2.
18710:     OS VALORES DE ATRIBUTO2 SAO IMPRESSOS E OS ARQUIVOS
18720:     E PONTEIROS RESTAURADOS ;
18730:
18740:     DENTRONOSE:=TRUE;
18750:     % SALVA OS ARQUIVOS DINAMICOS E LIMPA-OS.
18760:     FILL LIMPO[+] WITH 30(0);
18770:     FOR I:=0 STEP 1 UNTIL 127 DO
18780:       SALVAARVORECONTEXTOS[I]:=ARVORECONTEXTOS[I];
18790:     FOR I:=0 STEP 1 UNTIL 999 DO
18800:       FOR J:=0,1,2,3 DO BEGIN
18810:         SALVAREGRACONSULTA[J,I]:=REGRACONSULTA[J,I];
18820:         REGRACONSULTA[J,I]:=0;

```

```

18830:     END;
18840:   FOR I:=0 STEP 1 UNTIL 99 DO BEGIN
18850:     IF REGACESSADO[I] THEN BEGIN
18860:       READ(VALORCONHECIDO[I],30,CONHECO);
18870:       FOR J:=1 STEP 1 UNTIL 30 DO
18880:         SALVAVALORCONHECIDO[J,I]:=CONHECO[J];
18890:       WRITE(VALORCONHECIDO[I],30,LIMPO);
18900:       SALVAREGACESSADO[I]:=REGACESSADO[I];
18910:       REGACESSADO[I]:=FALSE;
18920:     END;
18930:     SALVANAOTEMCONTEXTO[I]:=NAOTEMCONTEXTO[I];
18940:     SALVALASTCNTXT[I]:=LASTCNTXT[I];
18950:     SALVACONCLUSDES[I]:=VETORCONCLUSDES[I];
18960:     VETORCONCLUSDES[I]:=0;
18970:   END;
18980:   FOR I:=1 STEP 1 UNTIL 200 DO
18990:     FOR J:=0 STEP 1 UNTIL 11 DO BEGIN
19000:       SALVAANSWER[I,J]:=ANSWER[I,J];
19010:       ANSWER[I,J]:=0;
19020:     END;
19030:   % SALVA VARIÁVEIS GLOBAIS
19040:   SALVAP:=P;
19050:   SALVAPREGRA:=PREGRA;
19060:   SALVANUMPERGUNTA:=NUMPERGUNTA;
19070:   SALVAAVAIL:=AVAIL;
19080:   SALVACOUNTER:=COUNTER;
19090:   SALVAULTIMALINHA:=ULTIMALINHA;
19100:   PREGRA:=NUMPERGUNTA:=COUNTER:=ULTIMALINHA:=0;
19110:   AVAIL:=1;
19120:   % ATUALIZA HORA, DIA E MES ATUAIS
19130:   ATUALI7AVALORESCONSULTA(22,1,VALH,1,0,1,0,0,0,0);
19140:   ATUALI7AVALORESCONSULTA(23,1,VALD,1,0,1,0,0,0,0);
19150:   ATUALI7AVALORESCONSULTA(24,1,VALM,1,0,1,0,0,0,0);
19160:   % ATUALIZA O VALOR DE ATRIBUTO1
19170:   ATUALI7AVALORESCONSULTA(ATRIBUTO1,CONTEXTO,VALOR,
19180:     TIPOVALOR,0,1,1,0,0,0);
19190:   % INFERE O VALOR DE ATRIBUTO2
19200:   BUSCAVALOR(ATRIBUTO2,CONTEXTO,0,ZERO,ZERO,FALSO,0,0,0);
19210:   % TOMA OS VALORES DE ATRIBUTO2 E IMPRIME-OS
19220:   MOSTRAVALOR(ATRIBUTO2,CONTEXTO);
19230:   % DESATIVA "SE"
19240:   DENTRODESE:=FALSE;
19250:   % RESTAURA OS ARQUIVOS DINAMICOS E PONTEIROS
19260:   FOR I:=0 STEP 1 UNTIL 127 DO
19270:     ARVORECONTEXTOS[I]:=SALVAARVORECONTEXTOS[I];
19280:   FOR I:=0 STEP 1 UNTIL 999 DO
19290:     FOR J:=0,1,2,3 DO
19300:       REGRACONSULTA[J,I]:=SALVAREGRACONSULTA[J,I];
19310:   FOR I:=0 STEP 1 UNTIL 99 DO BEGIN
19320:     IF SALVAREGACESSADO[I] THEN BEGIN
19330:       FOR J:=1 STEP 1 UNTIL 30 DO
19340:         CONHECO[J]:=SALVAVALORCONHECIDO[J,I];
19350:       WRITE(VALORCONHECIDO[I],30,CONHECO);
19360:     END;
19370:     REGACESSADO[I]:=SALVAREGACESSADO[I];
19380:     NAOTEMCONTEXTO[I]:=SALVANAOTEMCONTEXTO[I];
19390:     LASTCNTXT[I]:=SALVALASTCNTXT[I];
19400:     VETORCONCLUSDES[I]:=SALVACONCLUSDES[I];
19410:   END;

```

```

19420: FOR I:=1 STEP 1 UNTIL 200 DO
19430:   FOR J:=0 STEP 1 UNTIL 11 DO
19440:     ANSWER[I,J]:=SALVAANSWER[I,J]
19450:   % RESTAURA PONTEIROS
19460:   P:=SALVAP;
19470:   PREGRA:=SALVAPREGRA;
19480:   NUMPERGUNTA:=SALVANUMPERGUNTA;
19490:   AVATL:=SALVAAVAIL;
19500:   COUNTER:=SALVACOUNTER;
19510:   ULTIMALINHA:=SALVAULTIMALINHA;
19520: END;
19530:
19540:
19550: *-----*
19560:
19570: COMMENT:
19580:   ** INICIO DE EXPLANACAO **
19590:   -----
19600:   ESTE PROCEDIMENTO E' ACESSADO POR UMA LINGUAGEM
19610:   DE COMANDOS, PARA ACOMPANHAMENTO DA CONSULTA
19620:   PELO USUARIO ;
19630:
19640:
19650:   CONTEXTO:=0;
19660:   PEGAUMAPALAVRA;
19670:   IF I NEG 4 THEN DENTRODEPORQUE:=FALSE;
19680:   CASE I OF BEGIN
19690: 1: % COMANDOS
19700:   IF NOT MUDANDO THEN LISTACOMANDOS;
19710: 2: % PERGUNTA
19720:   IF NOT MUDANDO THEN MOSTREMENSAGEM(FPASE);
19730: 3: % REGRA
19740:   BEGIN
19750:     PEGAUMAPALAVRA;
19760:     IF PALAVRA = EMPTY THEN DECODIFICAREGRA(REGRAATUAL,0,0,0,0);
19770:     ELSE WHILE PALAVRA NEG EMPTY DO
19780:       IF NUMERO THEN BEGIN
19790:         NREGRA:=DECIMAL(PALAVRA);
19800:         IF NREGRA LEQ REGRASDISK*LASTRECORD AND REGRAATIVA(NREGRA)
19810:         THEN BEGIN
19820:           IF NOT MUDANDO THEN
19830:             DECODIFICAREGRA(NREGRA,0,0,0,0);
19840:           PEGAUMAPALAVRA;
19850:           END
19860:         ELSE BEGIN
19870:           ERRO1("ESTA REGRA NAO ESTA ATIVA");
19880:           PALAVRA:=EMPTY;
19890:           END;
19900:         END
19910:       ELSE BEGIN
19920:         ERRO1("ESPERADO = NUMERO DA REGRA");
19930:         PALAVRA:=EMPTY;
19940:         END
19950:       END;
19960: 4: % PORQUE
19970:   IF NOT MUDANDO THEN PORQUE;
19980: 5: % QUANDO
19990:   BEGIN
20000:     PEGAPALAVRALINHA;

```

```

200100 IF PALAVRA = "E" THEN BEGIN
200200   PEGAPALAVRALINHA;
200300   IF PALAVRA = "USADO" THEN BEGIN
200400     PEGATERMO("?");
200500     IF PESQUISADIRETORIO AND CLASSE = 1 THEN BEGIN
200600       ENDATRIBUTO:=CHAVEREG = 1;
200700       IF NOT MUDANDO THEN LISTAREFERENCIAS;
200800       TIRAINTERROGACAO;
200900       END
201000     ELSE ERRO1("ESPERADO:ATRIBUTO")
201100     END
201200   ELSE ERRO1("ESPERADO:USADO")
201300   END
201400 ELSE ERRO1("ESPERADO:E");
201500 END;
61 * COMO
201600 BEGIN
201700   PEGAPALAVRALINHA;
201800   IF PALAVRA = "CONCLUIR" THEN BEGIN
201900     PEGATERMO("?");
202000     IF PESQUISADIRETORIO AND CLASSE = 1 THEN BEGIN
202100       ENDATRIBUTO:=CHAVEREG;
202200       IF NOT MUDANDO THEN LISTAREFERENCIAS;
202300       TIRAINTERROGACAO;
202400       END
202500     ELSE ERRO1("ESPERADO:ATRIBUTO")
202600     END
202700   ELSE
202800     IF PALAVRA = "CONCLUIU" THEN BEGIN
202900       PEGATERMO(".DE");
203000       IF PESQUISADIRETORIO AND CLASSE = 1 THEN BEGIN
203100         ATRIBUTO:=L;
203200         TIPOAT:=TIPOATRIBUTO;
203300         PEGAPALAVRALINHA;
203400         IF PALAVRA = ".DE" THEN BEGIN
203500           PEGATERMO("=");
203600           TRATACONTEXTOS;
203700           IF CONTEXTO NEO 0 THEN BEGIN
203800             PEGAPALAVRALINHA;
203900             IF PALAVRA = "=" THEN BEGIN
204000               PEGATERMO("?");
204100               DEUCERTO:=TRUE;
204200               TRATAVALOR;
204300               IF NOT MUDANDO THEN
204400                 IF DEUCERTO THEN COMOCONCLUIU(ATRIBUTO,CONTEXTO,
204500                 VALOR,TIPOVALOR)
204600               ELSE ERRO1("PROBLEMA NO VALOR");
204700               TIRAINTERROGACAO;
204800               END
204900             ELSE ERRO1("ESPERADO==")
205000             END
205100           ELSE ERRO1("PROBLEMA NO CONTEXTO")
205200           END
205300         ELSE ERRO1("ESPERADO=.DE")
205400         END
205500       ELSE ERRO1("PROBLEMA NO ATRIBUTO")
205600       END
205700     ELSE ERRO1("ESPERADO=CONCLUIU OU CONCLUIR")
205800   END;
205900 END;

```

```
206000 7: % RESPOSTAS
206100 IF NOT MUDANDO THEN RESPOSTAS;
206200 8: % ARVORE
206300 IF NOT MUDANDO THEN ARVORE;
206400 9: % ANCESTRAIS
206500 BEGIN
206600 PEGAPALAVRALINHA;
206700 IF PALAVRA = ".DE" THEN BEGIN
206800 PEGATERMO("?");
206900 TRATACONTEXTO;
207000 IF NOT MUDANDO THEN
207100 IF CONTEXTO NEQ 0 THEN ANCESTRAIS(CONTEXTO)
207200 ELSE ERRO1("PROBLEMA NO CONTEXTO")
207300 END
207400 ELSE ERRO1("ESPERADO=.DE")
207500 END;
207600 10: % DESCENDENTES
207700 BEGIN
207800 PEGAPALAVRALINHA;
207900 IF PALAVRA = ".DE" THEN BEGIN
208000 PEGATERMO("?");
208100 TRATACONTEXTO;
208200 IF NOT MUDANDO THEN
208300 IF CONTEXTO NEQ 0 THEN DESCENDENTES(CONTEXTO)
208400 ELSE ERRO1("PROBLEMA NO CONTEXTO")
208500 END
208600 ELSE ERRO1("ESPERADO=.DE")
208700 END;
208800 11: % VALORES
208900 BEGIN
209000 PEGAPALAVRALINHA;
209100 IF PALAVRA = ".DE" THEN BEGIN
209200 PEGATERMO(".PARA");
209300 TRATACONTEXTO;
209400 IF CONTEXTO NEQ 0 THEN BEGIN
209500 PEGAUMAPALAVRA;
209600 IF PALAVRA = EMPTY THEN MOSTRAVALOR(ATR,CONTEXTO)
209700 ELSE
209800 IF PALAVRA = ".PARA" THEN BEGIN
209900 PEGATERMO(",");
210000 WHILE PALAVRA NEQ EMPTY
210100 DO IF PESQUISADIRETORIO AND CLASSE = 1
210200 THEN BEGIN
210300 IF NOT MUDANDO THEN
210400 MOSTRAVALOR(L,CONTEXTO);
210500 PEGAUMAPALAVRA;
210600 IF PALAVRA = "," THEN PEGATERMO(",");
210700 END
210800 ELSE BEGIN
210900 ERRO1("PROBLEMA EM ATRIBUTO");
211000 PALAVRA:=EMPTY;
211100 END;
211200 END
211300 ELSE ERRO1("ESPERADO = .PARA")
211400 END
211500 ELSE ERRO1("PROBLEMA EM CONTEXTO")
211600 END
211700 ELSE ERRO1("ESPERADO=.DE")
211800 END;
```



```
21190 12: % MUDAR
21200 BEGIN
21210 PEGATERMO("DE");
21220 IF PESQUISADIRETORIO AND CLASSE = 1 THEN BEGIN
21230 ATRIBUTO1=L;
21240 PEGAPALAVRALINHA;
21250 IF PALAVRA = "DE" THEN BEGIN
21260 PEGATERMO(".");
21270 TRATACONTEXTO;
21280 IF NOT MUDANDO THEN
21290 IF CONTEXTO NEQ 0 THEN
21300 IF DENTRODESE THEN
21310 ERRO1("ESTA DENTRO DE SE");
21320 ELSE MUDAR(ATRIBUTO,CONTEXTO);
21330 ELSE ERRO1("PROBLEMA NO CONTEXTO");
21340 END
21350 ELSE ERRO1("ESPERADO=DE");
21360 END
21370 ELSE ERRO1("PROBLEMA NO ATRIBUTO")
21380 END;
21390 13: % SE
21400 BEGIN
21410 PEGATERMO("DE");
21420 IF PESQUISADIRETORIO AND CLASSE = 1 THEN BEGIN
21430 ATRIBUTO1=L;
21440 PEGAPALAVRALINHA;
21450 IF PALAVRA = "DE" THEN BEGIN
21460 PEGATERMO("");
21470 TRATACONTEXTO;
21480 IF CONTEXTO NEQ 0 THEN BEGIN
21490 PEGAPALAVRALINHA;
21500 IF PALAVRA = "" THEN BEGIN
21510 PEGATERMO("ENTAO");
21520 TRATAVALOR;
21530 IF DEUCEPTO THEN BEGIN
21540 PEGAPALAVRALINHA;
21550 IF PALAVRA = "ENTAO" THEN BEGIN
21560 PEGATERMO("?");
21570 IF PESQUISADIRETORIO AND CLASSE = 1 THEN BEGIN
21580 ATRIBUTO2=L;
21590 IF NOT MUDANDO THEN
21600 IF DENTRODESE THEN
21610 ERRO1("DENTRO DE SE");
21620 ELSE
21630 SE(ATRIBUTO1,ATRIBUTO2,CONTEXTO,VALOR,TIPOVALOR);
21640 TIRAINTERROGACAO;
21650 END
21660 ELSE ERRO1("PROBLEMA EM ATRIBUTO")
21670 END
21680 ELSE ERRO1("ESPERADO=ENTAO")
21690 END
21700 ELSE ERRO1("PROBLEMA NO VALOR")
21710 END
21720 ELSE ERRO1("ESPERADO=")
21730 END
21740 ELSE ERRO1("PROBLEMA NO CONTEXTO")
21750 END
21760 ELSE ERRO1("ESPERADO=DE")
21770 END
```

```

217800 ELSE ERRO1("PROBLEMA NO ATRIBUTO")
217900 END;
218000 14: % TERMINE
218100 GO FIMDETUDO;
218200 END CASE;
218300 END;
218400
218500 COMMENT:
218600 ** INICIO DE RESPONSA CONSISTENTE **
218700 -----
218800 VERIFICA SE A RESPOSTA DO USUARIO E' VALIDA. CASO
218900 A RESPOSTA SEJA UMA CHAMADA AO "EXPLANACAO" O
219000 COMANDO E' TRATADO E O PROCEDIMENTO VOLTA A ESPERAR
219100 A RESPOSTA A PERGUNTA FORMULADA ;
219200
219300
219400 COMMENT
219500 IF TRACE
219600 THEN WRITE(SAIDA,</,"* RESPONSA CONSISTENTE *",,, ATR ="",13,
219700 " CNTXT ="",13," CONHECO = ",L1>,ATR,CNTXT,CONHECO);
219800 ACABOU:=FALSE;
219900 VALORSABIDO:=FALSE;
220000 WHILE NOT ACABOU DO BEGIN
220100 DENTRODEPORQUE:=FALSE;
220200 DO BEGIN
220300 COMANDOS:="COMANDOS*PERGUNTA*REGR*PORQUE*QUANDO*" CAT
220400 "COMO*RESPOSTAS*ARVORE*ANCESTRAIS*" CAT
220500 "DESCENDENTES*VALOPES*MUDAR*SF*TERMINE";
220600 PRIMEIRAPALAVRA:=EMPTY;
220700 IF NOT VALORSABIDO THEN
220800 WHILE PRIMEIRAPALAVRA = EMPTY DO BEGIN
220900 LELINHA;
221000 PRIMEIRAPALAVRA:=HEAD(TAIL(LINHA," "),NOT " ");
221100 END
221200 ELSE BEGIN
221300 LELINHA;
221400 PRIMEIRAPALAVRA:=HEAD(TAIL(LINHA," "),NOT " ");
221500 END;
221600 I:=1;
221700 ACHOU:=FALSE;
221800 WHILE NOT ACHOU AND I < 15 DO BEGIN
221900 COMANDO:=HEAD(COMANDOS,NOT "");
222000 COMANDOS:=TAIL(TAIL(COMANDOS,NOT ""),"");
222100 IF COMANDO = PRIMEIRAPALAVRA THEN ACHOU:=TRUE
222200 ELSE I:=I+1;
222300 END;
222400 IF ACHOU THEN EXPLANACAO;
222500 END
222600 UNTIL NOT ACHOU;
222700 VALORLIDO:=EMPTY;
222800
222900 COMMENT RECERE E TRATA O VALOR;
223000
223100 FATOREPRADO:=FALSE;
223200 READ(DIRDISK(ATR),10,DIRETORIO);
223300 NOMEATRIBUTO:=DIRNOME;
223400 TIPOATR:=TIPOATRIBUTO;
223500 IF DENOVO THEN LYSTA:="SIM NAO S N"
223600 ELSE BEGIN

```

```

22370L READ(PPROPATRIBUTOS[CHAVEPEG/2],70,PPROPRIENADE);
22380L LISTA:=ESPERADO;
22390L END;
22400L PEGAPALAVRA;
22410L VALORLIDO:=PALAVRA;
22420L IF PALAVRA = EMPTY THEN ACABOU:=TRUE;
22430L IF NOT ACABOU THEN
22440L IF NUMERO THEN BEGIN
22450L DEFINE UNIDADE = 5 #;
22460L TIPOVALOR:=NUMERICO;
22470L VALORNUM:=VAL;
22480L UNIDADELIDA:=EMPTY;
22490L UNIDADENUM:=0;
22500L IF NOT PEGAFATOR THEN
22510L IF PESQUISADIRETORIO THEN
22520L IF CLASSE = UNIDADE THEN BEGIN
22530L UNIDADENUM:=L;
22540L UNIDADELIDA:=PALAVRA;
22550L PEGAFATOR;
22560L END
22570L ELSE FATORERRADO:=TRUE
22580L ELSE FATORERRADO:=TRUE;
22590L IF FATORERRADO AND NOT MUDANDO
22600L THEN WRITE(VIDEO,<A1>*** RESPOSTA INVALIDA"
22610L ,". FATOR DEVE ESTAR ENTRE 0. E 1.">,BUZINA);
22620L END
22630L ELSE BEGIN
22640L TIPOVALOR:=NAONUMERICO;
22650L IF TIPOAT NEG 0 THEN
22660L IF PESQUISADIRETORIO THEN VALORNUM:=L;
22670L UNIDADELIDA:=EMPTY;
22680L UNIDADENUM:=0;
22690L PEGAFATOR;
22700L END;
22710L
22720L IF VALORLIDO = ".*" THEN ACABOU:=TRUE ;
22730L IF NOT ACABOU AND NOT FATORERRADO THEN BEGIN
22740L VERIFICAVALORACEITAVEL;
22750L IF NOT ACHEI
22760L THEN IF NOT MUDANDO THEN BEGIN
22770L WRITE(VIDEO,<A1>/,*** RESPOSTA INVALIDA .">,BUZINA);
22780L MOSTRAVALORESPOSSIVEIS;
22790L END
22800L ELSE
22810L ELSE BEGIN
22820L VALORSABIDO:=TRUE;
22830L IF DENOVU THEN BEGIN
22840L ARRAY REGRA[0:149];
22850L ACABOU:=TRUE;
22860L IF VALORLIDO = "S" OR VALORLIDO = "SIM" THEN BEGIN
22870L INTEGER I;
22880L I:=SC;
22890L READ(REGRASDISK[SP],150,REGRA);
22900L VALORNUM:=EVALOR;
22910L UNIDADENUM:=EUNIDADE;
22920L TIPOVALOR:=ETIPOVALOR;
22930L NEGADO:=EMODFUNCAO;
22940L END;
22950L END

```

```

229600 ELSE IF TIPOAT = 0 THEN BEGIN
229700 ACABOU:=TRUE;
229800 VALORNUM:=0;
229900 IF VALORLIDO = "N" OR VALORLIDO = "NAO"
230000 THEN NEGADO:=-1 ELSE NEGADO:=0;
230100 END;
230200 ATUALIZAVALORESCONSULTA(ATR,CNTXT,VALORNUM,TIPOVALOR,
230300 UNIDADENUM,FATOR,1,FRASE,0,NEGADO);
230400 END;
230500 END;
230600 END;
230700 IF TRACE THEN WRITE(SAIDA," AD FINAL FATOR =",F7.2,FATOR);
230800 END PROCEDURE RESPONSTACONSISTENTE;
230900

```

```

231000 -----
231100
231200 PROCEDURE PERGUNTARPELOVALOR(A,C,S,SP,SC,K,P,PR,NCL,NSUBCL,
231300 NCONCLUSAO);
231400 VALUE A,C,S,SP,SC,NCL,NSUBCL,NCONCLUSAO;
231500 INTEGER A,C,SP,SC,R,PR,NCL,NSUBCL,NCONCLUSAO;
231600 BOOLEAN K,S;
231700 BEGIN
231800
231900 COMMENT :
232000 A = ENDEREÇO DO ATRIBUTO
232100 C = CONTEXTO (PONTEIRO PARA A ARVORE DE CONTEXTOS)
232200 S = QDO = 1 INDICA QUE ESTA SE PERGUNTANDO PELA 2A VEZ
232300 SP= NRO DA REGRA PERGUNTADA PELA SEGUNDA VEZ
232400 SC= NRO DA CONCLUSAO DA REGRA ACIMA
232500 K = USUARIO DESCONHECE RESPOSTA SE K FOR FALSO
232600 R = REGRA ATUAL
232700 PR = POSICAO EM REGRACONSULTA ;
232800
232900 BOOLEAN ACHOUOBJETO;
233000 STRING VALOR,CONTEXTO,FRASE,CADEIA;
233100 REAL NVALOR,FATOR;
233200 INTEGER UNID,NEGADO,I;
233300 DEFINE
233400 SUBSTITUI(X,Y) =
233500 X-----
233600 BEGIN
233700 STRING CABECA,CARTAO,PALAVRA;
233800 CARTAO:=FRASE;
233900 CABECA:=EMPTY;
234000 WHILE NOT ACHOUOBJETO AND CARTAO NEQ EMPTY DO BEGIN
234100 PALAVRA:=HEAD(TAIL(CARTAO," "),NOT " ");
234200 CARTAO:=TAIL(TAIL(CARTAO," "),NOT " ");
234300 IF PALAVRA = X THEN BEGIN
234400 FRASE:=CABECA CAT Y CAT CARTAO;
234500 ACHOUOBJETO:=TRUE;
234600 END
234700 ELSE CABECA:=CABECA CAT PALAVRA CAT " ";
234800 END;
234900 END #;
235000
235100 PROCEDURE DECODIFICAPERGUNTA;
235200 BEGIN
235300 ARRAY REGRA[0:149];
235400 STRING CARTAO,PALAVRA;

```

```

235500      INTEGER CL,I,CONTADOR;
235600
235700  COMMENT:
235800      ** DECODIFICAPERGUNTA **
235900      -----
236000  DECODIFICA A PERGUNTA A SER FEITA AO USUARIO ;
236100
236200      READ(REGRASK[SP],150,REGRA);
236300      CONTADOR:=0;
236400      I:=EENTAO;
236500      DO BEGIN
236600          CONTADOR:=* + 1;
236700          I:=PROXC(I);
236800      END
236900      UNTIL CONTADOR = SC;
237000      READ(DIPDISK[EFUNCAO],10,DIRETORIO);
237100      IF SINONIMO = 2 THEN BEGIN
237200          READ(TEXTUDISK[FVALOR],90,CLAUSE);
237300          FRASE:=HEAD(STRING(POINTER(CLAUSE),540),NOT "?");
237400      END
237500      ELSE BEGIN
237600          DECODIFICAESQUELETO;
237700          FRASE:=LINE;
237800      END;
237900      READ(DIPDISK[EOBJETO],10,DIRETORIO);
238000      CADEIA:=DIRNOME;
238100      SUBSTITUI(CADEIA,CONTEXTO);
238200      IF NOT ACHOUOBJETO THEN
238300          FRASE:=FRASE CAT "( " CAT CONTEXTO CAT " )";
238400      FRASE:=FRASE CAT " ?";
238500      FND;
238600
238700  COMMENT:
238800      ** INICIO DE PERGUNTARPELOVALOR **
238900      -----
239000  PERGUNTA PELO VALOR DOS ATRIBUTO E CONTEXTO DADOS
239100  E VERIFICA A CONSISTENCIA DA RESPOSTA. A RESPOSTA
239200  E TIRADA DE "ANSWER" CASO "MUDANDO" SEJA TRUE ;
239300
239400      ACHOUOBJETO:=FALSE;
239500      IF TRACE THEN
239600          WRITE(SAIDA,/<,"* PERGUNTARPELOVALOR *",/," ATRIBUTO =",
239700              I3," CONTEXTO =","I3," REGRA =","I3>>A,C,R);
239800      LERTIPUSCONTEXTO(TIPO(C));
239900      CONTEXTO:=TRADUZ CAT "-" CAT STRING(NUMCONTEXTO(C),*);
240000
240100      IF S THEN DECODIFICAPERGUNTA
240200      ELSE BEGIN
240300          READ(DIPDISK[A],10,DIRETORIO);
240400          READ(PROPATRIBUTOS[CHAVEREG/2],70,PROPRIEDADE);
240500          FRASE:=PERGUNTA;
240600          SUBSTITUI("","CONTEXTO);
240700      END;
240800      % DESATIVA "MUDANDO" QUANDO CHEGA-SE 'A PERGUNTA
240900      % QUE DEVE SER MUDADA
241000
241100      IF MUDANDO AND PERGUNTAMUDADA = NUMPERGUNTA + 1
241200      THEN MUDANDO:=FALSE;
241300

```

```

24140:      MOSTREPERGUNTA(FRASE);
24150:
24160:      RESPOSTACONSISTENTE(A,C,S,SP,SC,K,FRASE,R,PR,NCL,NSUBCL,
24170:        NCONCLUSAO,NUMPERGUNTA);
24180:      END PROCEDURE PERGUNTARPELOVALOR;
24190:
24200:  *-----
24210:
24220:      PROCEDURE MONITOR(X1,X2,X3,X4,X5);
24230:      VALUE X1,X2,X4;
24240:      INTEGER X1,X2,X3,X4,X5;
24250:      FORWARD;
24260:
24270:  *-----
24280:
24290:      PROCEDURE BUSCAVALOR(ATRIBUTO,CNTXT,REGRAPI,PREGRAPI,PIRMAO,
24300:        DEDUTIVEL,NCLPAI,NSUBCLPAI,NCONCLUSADPAI);
24310:      VALUE ATRIBUTO,CNTXT,NCLPAI,NSUBCLPAI,NCONCLUSADPAI;
24320:      INTEGER ATRIBUTO,CNTXT,REGRAPI,PREGRAPI,PIRMAO,NCLPAI,NSUBCLPAI,
24330:        NCONCLUSADPAI;
24340:      BOOLEAN DEDUTIVEL;
24350:      BEGIN
24360:
24370:        ARRAY REFERENCIAS[0:179],REGRA[0:149],PERGUNTA[1:50,1:2],
24380:          ESPECIAL[1:50,1:2];
24390:        BOOLEAN TEMVALOR DESCONHECIDO;
24400:        INTEGER CONCLUE,NUMCONCLUSAO,PREGRAANTEPIOR,CONT,U;
24410:        SALVAREGRAPI,ESPECIALS,SALVAPREGPA,TIPOVALOR,ULTIMA;
24420:        REAL VALOR;
24430:        DEFINE
24440:          APLICARMONITOR =
24450:            *-----
24460:          BEGIN
24470:            INTEGER X,I,AUXATR,AUXOBJ;
24480:            BOOLEAN SAI,ACHOU;
24490:            PREGRAANTERIOR:=0;
24500:            RPAI(0):=PREGRAPI;
24510:            XI:=RLINK(0);
24520:            IF X NEQ 0
24530:            THEN BEGIN
24540:              RIRMAO(PIRMAO):=PREGRA+1;
24550:              PIRMAO:=PREGRA+1;
24560:              DEDUTIVEL:=TRUE;
24570:            END;
24580:            WHILE X NEQ 0 DO BEGIN
24590:              * PROCURAR O NUMERO DA CONCLUSAO
24600:              * -----
24610:              READ(REGRASDISK[NUMREGRA(X)],150,REGRA);
24620:              I:=PROXC(EENTAO);
24630:              NUMCONCLUSAO:=1;
24640:              IF TRACE THEN
24650:                WRITE(SAIDA,"APLICARMONITOR REGRA =",I3,"ATRIBUTO =",I3," CONTFXTO =",
24660:                  I3," OBJ CNTXT =",I3,NUMREGRA(X),ATRIBUTO,CNTXT,OBJECT(CNTXT));
24670:              SAI:=FALSE;
24680:              ACHOU:=FALSE;
24690:              WHILE NOT ACHOU AND NOT SAI DO BEGIN
24700:                READ(DIRDISK[ATRIBUTO],10,DIRETORIO);
24710:                AUXATR:=SINDNIMO;
24720:                READ(DIRDISK[OBJETO],10,DIRETORIO);

```

```

247300 AUXOBJ:=SINONIMO;
247400 IF AUXATR = ATRIBUTO AND AUXOBJ = OBJECT(CNTXT)
247500 THEN ACHOUI:=TRUE
247600 ELSE IF PROXC(I) = EENTAO THEN SAI:=TRUE
247700 ELSE BEGIN
247800 I:=PROXC(I);
247900 NUMCONCLUSAO:=NUMCONCLUSAO+1;
248000 END;
248100 END;
248200
248300 COMMENT : O SISTEMA ESTA ADMITINDO REGRAS COM O MESMO ATRIBUTO NA
248400 PREMISSA E CONCLUSAO. AQUI SE UMA A REGRA QUE SERIA
248500 APLICADA A MONITOR POR A MESMA QUE CHAMOU BUSCAVALOR
248600 ELA SERA DESCARTADA ;
248700 IF EMODFUNCAO = 0 THEN CONCLUE:=EATRIBUTO
248800 ELSE CONCLUE:=EATRIBUTO;
248900 IF NUMREGRA(X) = REGRAPI THEN SAI:=TRUE;
249000 IF NOT SAI THEN
249100 IF REGRAESPECIAL THEN BEGIN
249200 ESPECIAIS:=* + 1;
249300 ESPECIAL[ESPECIAIS,1]:=NUMREGRA(X);
249400 ESPECIAL[ESPECIAIS,2]:=NUMCONCLUSAO;
249500 END
249600 ELSE BEGIN
249700 IF REPERGUNTA = 1 THEN BEGIN
249800 ULTIMA:=* + 1;
249900 PERGUNTA[ULTIMA,1]:=NUMREGRA(X);
250000 PERGUNTA[ULTIMA,2]:=1;
250100 END;
250200 MONITOR(NUMREGRA(X),NUMCONCLUSAO,REGRAPI,CONCLUE,
250300 PREGRAANTERIOR);
250400 END;
250500 X:=RLINK(X);
250600 END;
250700 FOR U:=1 STEP 1 UNTIL ESPECIAIS DO
250800 MONITOR(ESPECIAL[U,1],ESPECIAL[U,2],REGRAPI,CONCLUE,
250900 PREGRAANTERIOR);
251000 END ;
251100
251200
251300 ATUALIZAKDES =
251400 ;-----
251500 BEGIN
251600 READ(DIRDISK[ATRIBUTO],10,DIRETORIO);
251700 IF REGISTROJAEXISTE
251800 THEN READ(VALORCONHECIDO[CHAVEREG/2],30,CONHECO)
251900 ELSE FILL CONHECO[*] WITH 30(0);
252000 KCONTEXT(CNT):=CNTXT;
252100 KDES(CNT):=1;
252200 REGACESSADO[CHAVEREG/2]:=TRUE;
252300 WRITE(VALORCONHECIDO[CHAVEREG/2],30,CONHECO);
252400 END ;
252500
252600 PROCEDURE PERGUNTARDENOVO;
252700 BEGIN
252800 INTEGER I,J,K,L,CONTEXT0;
252900 ARRAY REGRA[C:149];
253000 BOOLEAN EXISTE;
253100 DEFINE

```

```

253200 VERIFICASEEXISTEDVALOR =
253300 EXISTE:=FALSE;
253400 READ(DIRDISK(ATRIBUTO),10,DIRETORIO);
253500 IF REGISTRUAEXISTE THEN BEGIN
253600   K:=1;
253700   READ(VALORCONHECIDO(CHAVFREG/2),3,CONHECIDO);
253800   WHILE KCONTEXTO(K) NEQ 0 AND NOT EXISTE DO
253900     IF KCONTEXTO(K) = CONTEXTO AND AVALOR(K) = EVALOR
254000       AND KTIPO(K) = ETIPOVALOR
254100         THEN EXISTE:=TRUE
254200           ELSE K:=K + 2;
254300   END #;
254400   END #;
254500   FOR LI:=1 STEP 1 UNTIL ULTIMA DO BEGIN
254600     READ(REGRASDISK(PERGUNTA[L,1]),150,REGRA);
254700     I:=PERGUNTA[L,2];
254800     FOR J:=1 STEP 1 UNTIL 6 DO
254900       IF ETIPOREGRA[J] NEQ 0 THEN
255000         FOR CONTEXTO:=1 STEP 1 UNTIL AVAIL - 1 DO
255100           IF TIPO(CONTEXTO) = ETIPOREGRA[J] THEN BEGIN
255200             VERIFICASEEXISTEDVALOR;
255300             IF NOT EXISTE THEN PERGUNTAPPELOVALOR(ATRIBUTO,
255400               CONTEXTO,TRUE,PERGUNTA[L,1],PERGUNTA[L,2],TEMVALOR,
255500               SALVAREGRAPI,SALVAPREGRA,NCLPAI,NSUBCLPAI,
255600               NCONCLUSAOPAI);
255700             END;
255800           END;
255900         END;
256000       IF TRACE
256100         THEN WRITE(SAIDA,<"/," BUSCAVALOR "<"/," ATRIBUTO ="<I3,
256200           " CNTXT="<I3," REGRAPAI ="<I3>>ATRIBUTO,CNTXT,REGRAPAI);
256300         TEMVALOR:=FALSE;
256400         DEDUTIVEL:=FALSE;
256500         READ(DIRDISK(ATRIBUTO),10,DIRETORIO);
256600         READ(PROPATRIBUTOS(CHAVFREG/2),70,PROPRIEDADE);
256700
256800 COMMENT :
256900   ** INICIO DE BUSCAVALOR **
257000   SE O ATRIBUTO FOR DO TIPO DEDUTIVEL ,APLICA-SE MONITOR A
257100   CADA UMA DAS REGRAS QUE CONCLUEM ESTE ATRIBUTO .SE NAO
257200   CONSEGUIR DEDUZIR NENHUM VALOR (E NAO SE ESTA NO TOPO DA
257300   ARVORE E O ATRIBUTO NAO E O OBJETIVO DA CONSULTA) ENTAO
257400   PERGUNTA-SE O VALOR .
257500   SE O ATRIBUTO FOR DO TIPO NAO DEDUTIVEL ,O VALOR E PERGUNTADO.
257600   NAO SE OBTENDO RESPOSTA TENTA-SE DEDUZIR O VALOR APLICANDO
257700   MONITOR A TODAS AS REGRAS QUE CONCLUEM O ATRIBUTO .
257800   SE NAO SE CONSEGUIU O VALOR DO ATRIBUTO NEM DEDUZINDO-O NEM
257900   PERGUNTANDO-SE ,ENTAO ESTE E ATUALIZADO EM VALORCONHECIDO
258000   COMO SENDO DESCONHECIDO ;
258100
258200   ULTIMA:=0;
258300   ESPECIAIS:=0;
258400   SALVAREGRAPI:=REGRAPAI;
258500   SALVAPREGRA:=PREGRAPI;
258600   IF DEDUZIRVALOR THEN BEGIN
258700     READ(ATRIBUTOSREF(CHAVFREG),180,REFERENCIAS);
258800     APLICARMONITOR;
258900     VERIFICASEEXISTE(ATRIBUTO,CNTXT,TEMVALOR,DESCONHECIDO,CONT);
259000

```



```

25910E      IF NOT TEMVALOR AND ((ATRIBUTO NEQ ATRIBUTOOBJETIVO)
25920E      OR (CNTXT NEQ 1)) THEN
25930E          PERGUNTARPELOVALOR(ATRIBUTO,CNTXT,FALSE,0,0,TEMVALOR,
25940E          SALVAREGRAPI,SALVAPREGRA,NCLPAI,NSUBCLPAI,NCONCLUSAGPAI);
25950E      PERGUNTARDENOVO;
25960E      VERIFICASEEXISTE(ATRIBUTO,CNTXT,TEMVALOR,DESCONHECIDO,CONT);
25970E      IF NOT TEMVALOR THEN ATUALIZAKDES ;
25980E      END
25990E
26000E
26010E      ELSE BEGIN
26020E          PERGUNTARPELOVALOR(ATRIBUTO,CNTXT,FALSE,0,0,TEMVALOR,
26030E          SALVAREGRAPI,SALVAPREGRA,NCLPAI,NSUBCLPAI,NCONCLUSAGPAI);
26040E          IF NOT TEMVALOR THEN BEGIN
26050E              READ(ATRIBUTOSREF,CHAVEREG,180,REFERENCIAS);
26060E              APLICARMUNITOR;
26070E          END;
26080E          PERGUNTARDENOVO;
26090E          VERIFICASEEXISTE(ATRIBUTO,CNTXT,TEMVALOR,DESCONHECIDO,CONT);
26100E          IF NOT TEMVALOR THEN ATUALIZAKDES ;
26110E      END;
26120E
26130E      IF TRACE THEN WRITE(SAIDA,<"/>"* FIM DE BUSCAVALOR COM ATRIBUTO",I3,
26140E      " *>,>ATRIBUTO);
26150E      WRITE(SAIDA,<"* BUSCAVALOR - ATRIBUTO ",I3,>,>ATRIBUTO);
26160E      ESCHFEEMREGPA;
26170E      END PROCEDURE BUSCAVALOR;
26180E
26190E  *-----*
26200E
26210E      PROCEDURE AVALIA(NREGRA,NUMCONCLUSAO,CNTXT,REGRAPI,ACAO,TUDDOK,
26220E      NOATUAL);
26230E      VALUE NREGRA,NUMCONCLUSAO,CNTXT,ACAO,NOATUAL;
26240E      INTEGER NREGRA,CNTXT,REGRAPI,ACAO,NUMCONCLUSAO,NOATUAL;
26250E      BOOLEAN TUDDOK;
26260E      BEGIN
26270E
26280E          COMMENT :
26290E          NREGRA      - NUMERO DA REGRA A SER AVALIADA
26300E          CNTXT      - PONTEIRO PARA A ARVORE DE CONTEXTOS
26310E          REGRAPI    - PONTEIRO PARA O ARQUIVO 'REGRACONSULTA'
26320E          Acao       - INDICA O NUMERO DA CONCLUSAO DE 'REGRA' E SE
26330E          ELA APAPECE NEGADA (ACAO < 0)--ATRIBUTO CONCLUIDO
26340E          NUMCONCLUSAO - NUMERO DA CONCLUSAO (IMPORTANTE SE A REGRA
26350E          TIVER MAIS DE UMA CONCLUSAO)
26360E          NOATUAL    - NO DA REGRA ATUAL EM REGPACONSULTA;
26370E
26380E          BOOLEAN CLAUSULAFALSA,PRIMEIRAVEZ,PELOMENOSUM,SAIU,CONTINUA,
26390E          TEMVALOR,DESCONHECIDO;
26400E          REAL CERTEZA,CFMAX,TALLY,MBATUAL,MDATUAL,MB,MD;
26410E          INTEGER I,X,PREGRAPI,PREGRAANTERIOR,PIRMAC,NCL,NSUBCL,
26420E          CONTEXTOATUAL,CONT;
26430E      INTEGER ATRIBU;
26440E      ARRAY CLAUSULA(0:4),REGRA(0:149);
26450E      LABEL FIMAVALIA;
26460E      DEFINE ATRIBUTO      = CLAUSULA(0).[45:10] #,
26470E          OBJETO          = CLAUSULA(0).[35:10] #,
26480E          FUNCAO          = CLAUSULA(0).[25:10] #,
26490E          UNIDADE         = CLAUSULA(0).[15:10] #,

```

```

26500:      TIPOVALOR = CLAUSULA[1].[5:2] #,
26510:      MODFUNCAO = CLAUSULA[1].[27:10] #,
26520:      ATRIB(X) = CLAUSULA[1].[2:10] #,
26530:      VALOR = CLAUSULA[2] #,
26540:      FATOR = CLAUSULA[3] #,
26550:
26560:      ACHACONTEXTODATUAL =
26570:      X-----
26580:
26590:      COMMENT :
26600:      ACHA O CONTEXTO RELATIVO AQUELA CLAUSULA, PROCURANDO NA ARVORE
26610:      O CONTEXTO QUE CONTENHA O OBJETO DA CLAUSULA, A PARTIR DO CONTEXTO
26620:      MAIS BAIXO DA REGRA ;
26630:
26640:      BEGIN
26650:      INTEGER NO;
26660:      NO:=CNTXT;
26670:      READ(DIRDISK(OBJETO),10,DIRETLPI);
26680:      WHILE OBJECT(NO) NEQ SINONIMO DO NO:=PAI(NO);
26690:      CONTEXTODATUAL:=NO;
26700:      END #;
26710:
26720:      X-----
26730:      PROCEDURE AVALIAR;
26740:      BEGIN
26750:
26760:      DEFINE NORMALIZAVALOR =
26770:      X-----
26780:      IF TIPOVALOR = NUMERICO THEN
26790:      IF UNIDADE NEQ 0 THEN BEGIN
26800:      READ(DIRDISK(UNIDADE),10,DIRETORIO);
26810:      IF SUBMULTIPLO NEQ 0 THEN
26820:      VALOR:=VALOR * VALORSUB
26830:      END # ;
26840:
26850:      DEFINE TIPOCLAUSULA = CLAUSULA[1].[5:1] #,
26860:      CLAUSULAESPECIAL = TIPOCLAUSULA = 1 #;
26870:      X-----
26880:
26890:      INTEGER X,FUNC;
26900:      REAL RESULT,CFVALOR,CFTOTAL,CFIGUAL,CFDIF,CFMAIOR,CFMENOR;
26910:      BOOLEAN AVALIACAO,MODIFICADOR;
26920:      ARRAY LISTA[0:10];
26930:
26940:      X-----
26950:
26960:      BOOLEAN PROCEDURE CONHECIDO;
26970:      BEGIN
26980:
26990:      BOOLEAN ACABOU;
27000:      INTEGER PONT;
27010:
27020:      COMMENT:
27030:      ** CONHECIDO **
27040:      -----
27050:      E' AVALIADA COM VERDADEIRA SE O CF DO VALOR NA BASE
27060:      PARA UM DADO ATRIBUTO E CONTEXTO ? MAIOR QUE .2. O
27070:      FATOR E' DEVOLVIDO EM "RESULTA" ;
27080:

```

```

27090: CFMAIOR:=0;
27100: ACABOU:=FALSE;
27110: FOR PONT:=1 STEP 2 WHILE NOT ACABOU AND PONT < 30
27120: DO IF KCONTEXTO(PONT) = 0 THEN ACABOU:=TRUE
27130: ELSE IF KCONTEXTO(PONT) = CONTEXTOATUAL
27140: THEN IF KDES(PONT) = 1 THEN ACABOU:=TRUE
27150: ELSE BEGIN
27160: CFVALOR:=ARS(KMB(PONT) - KMD(PONT))/100;
27170: IF CFVALOR > CFMAIOR THEN CFMAIOR:=CFVALOR
27180: END;
27190: CONHECIDO:=IF CFMAIOR GEQ .2
27200: THEN IF MODIFICADOR THEN FALSE ELSE TRUE
27210: ELSE IF MODIFICADOR THEN TRUE ELSE FALSE;
27220: RESULT:=CFMAIOR;
27230: END;

```

```

27240:
27250:
27260:
27270: PROCEDURE TRATASIMNAD;
27280: BEGIN
27290:
27300:   BOOLEAN ACABOU;
27310:   INTEGER PONT;
27320:
27330: COMMENT:
27340:   ** TRATASIMNAD **
27350:   -----
27360:   RETORNA O CF ASSOCIADO A UM ATRIBUTO SIM/NAO E UM
27370:   CONTEXTO NA VARIÁVEL "CFTOTAL";
27380:
27390:   ACABOU:=FALSE;
27400:   FOR PONT:=1 STEP 2 WHILE NOT ACABOU AND PONT < 30
27410:   DO IF KCONTEXTO(PONT) = 0 THEN ACABOU:=TRUE
27420:   ELSE IF KCONTEXTO(PONT) = CONTEXTOATUAL
27430:   THEN IF KDES(PONT) = 1 THEN ACABOU:=TRUE
27440:   ELSE BEGIN
27450:     ACABOU:=TRUE;
27460:     CFTOTAL:=(KMB(PONT) - KMD(PONT))/100;
27470:     IF MODIFICADOR THEN CFTOTAL:=CFTOTAL;
27480:   END;
27490:
27500: END;

```

```

27510:
27520:
27530: PROCEDURE MAIORMENOR;
27540: BEGIN
27550:
27560:   BOOLEAN ACABOU;
27570:   INTEGER PONT;
27580:
27590: COMMENT:
27600:   ** MAIORMENOR **
27610:   -----
27620:   RETORNA EM CFIGUAL, CFMAIOR E CFMENOR RESPECTIVAMENTE,
27630:   OS CF ACUMULADOS DOS VALORES NA BASE DE FATOS, IGUAIS,
27640:   MAIORES E MENORES A UM DADO VALOR PARA DETERMINADOS
27650:   ATRIBUTO E CONTEXTO;
27660:
27670:   CFMAIOR:=CFMENOR:=CFIGUAL:=0;

```

```

27680: IF TRACE THEN WRITE(SAIDA, <"** MAIORMENOR VALOR =">F7.2, " TIPO =">
27690: I3>VALOR, TIPOVALOR))
27700: ACABOU:=FALSE;
27710: FOR PONT:=1 STEP 2 WHILE NOT ACABOU AND PONT < 30
27720: DO IF KCONTEXTO(PONT) = 0 THEN ACABOU:=TRUE
27730: ELSE IF KCONTEXTO(PONT) = CONTEXTOATUAL
27740: THEN IF KDES(PONT) = 1 THEN ACABOU:=TRUE
27750: ELSE BEGIN
27760: CFVALOR:=(KMB(PONT) - KMD(PONT))/100;
27770: WRITE(SAIDA, <"KVALOR =">F7.2, " KTIPO =">I3>KVALOR(PONT), KTIPO(PONT));
27780: IF KVALOR(PONT) = VALOR AND KTIPO(PONT) = TIPOVALOR
27790: THEN CFIGUAL:=CFIGUAL+CFVALOR - CFIGUAL*CFVALOR
27800: ELSE IF KVALOR(PONT) > VALOR
27810: THEN CFMAIOR:=CFMAIOR+CFVALOR - CFMAIOR*CFVALOR
27820: ELSE CFMENOR:=CFMENOR+CFVALOR - CFMENOR*CFVALOR
27830: END;
27840: WRITE(SAIDA, <"MENOR =">F7.2, "MAIOR =">F7.2, " IGUAL =">F7.2>>CFMENOR,
27850: CFMAIOR, CFIGUAL);
27860: END;
27870:
27880: -----
27890:
27900: PROCEDURE IGUALDIFERENTE;
27910: BEGIN
27920:
27930: BOOLEAN ACABOU;
27940: INTEGER PONT;
27950:
27960: COMMENT:
27970: ** IGUALDIFERENTE **
27980: -----
27990:
28000: RETORNA EM CFIGUAL E CFDIF RESPECTIVAMENTE, OS CF
28010: ACUMULADOS DOS VALORES NA BASE DE FATOS, IGUAIS E
28020: DIFERENTES A UM DADO VALOR PARA DETERMINADOS
28030: ATRIBUTO E CONTEXTO ;
28040:
28050: CFIGUAL:=CFDIF:=0;
28060: ACABOU:=FALSE;
28070: IF TRACE THEN WRITE(SAIDA, <"** IGUALDIFERENTE VALOR =">F7.2,
28080: " TIPO =">I3>VALOR, TIPOVALOR);
28090: FOR PONT:=1 STEP 2 WHILE NOT ACABOU AND PONT < 30
28100: DO IF KCONTEXTO(PONT) = 0 THEN ACABOU:=TRUE
28110: ELSE IF KCONTEXTO(PONT) = CONTEXTOATUAL
28120: THEN IF KDES(PONT) = 1 THEN ACABOU:=TRUE
28130: ELSE BEGIN
28140: CFVALOR:=(KMB(PONT) - KMD(PONT))/100;
28150: IF KVALOR(PONT) = VALOR AND KTIPO(PONT) = TIPOVALOR
28160: THEN CFIGUAL:=CFIGUAL+CFVALOR - CFIGUAL*CFVALOR
28170: ELSE CFDIF:=CFDIF+CFVALOR - CFDIF*CFVALOR;
28180: END;
28190:
28200: END;
28210:
28220: -----
28230:
28240: BOOLEAN PROCEDURE DESIGUALDADE;
28250: BEGIN
28260: COMMENT:
28270: ** DESIGUALDADE **

```

```

28270:      -----
28280:      E' VALIADA COMO VERDADEIRA SE OS VALORES ARMazenADOS
28290:      NA BASE DE FATOS PARA DETERMINADOS ATRIBUTO E CONTEXTO
28300:      SAO DIFERENTES DO VALOR DADO, SEGUNDO AS INDICACOES DE
28310:      FUNC (MAIOR,MENOR,MAIOR=IGUAL,MENOR=IGUAL,DIFERENTE).
28320:      O CF RESULTANTE E' DEVOLVIDO EM "RESULT"
28330:
28340:      IF TRACE THEN WRITE(SAIDA,"** DESIGUALDADE FUNCAD =",13,
28350:      " MOD = ",L1," TIPOATRIBUTO =",13>>FUNC,MODIFICADOR,
28360:      TIPOATRIBUTO);
28370:      CASE FUNC OF
28380:  BEGIN
28390:
28400:      9: % MAIOR
28410:      MAIORMENOR;
28420:      IF TIPOATRIBUTO = 1
28430:      THEN BEGIN
28440:          CFTOTAL:=CFMAIOR - CFMENOR - CFIGUAL;
28450:          IF MODIFICADOR THEN CFTOTAL:= - CFTOTAL;
28460:          END
28470:      ELSE IF NOT MODIFICADOR THEN CFTOTAL:=CFMAIOR
28480:      ELSE CFTOTAL:=CFMENOR + CFIGUAL;
28490:
28500:      10: % MENOR
28510:      MAIORMENOR;
28520:      IF TIPOATRIBUTO = 1
28530:      THEN BEGIN
28540:          CFTOTAL:=CFMENOR - CFMAIOR - CFIGUAL;
28550:          IF MODIFICADOR THEN CFTOTAL:= - CFTOTAL;
28560:          END
28570:      ELSE IF NOT MODIFICADOR THEN CFTOTAL:=CFMENOR
28580:      ELSE CFTOTAL:=CFMAIOR + CFIGUAL;
28590:
28600:      11: % MAIOR=IGUAL
28610:      MAIORMENOR;
28620:      IF TIPOATRIBUTO = 1
28630:      THEN BEGIN
28640:          CFTOTAL:=CFMAIOR + CFIGUAL - CFMENOR;
28650:          IF MODIFICADOR THEN CFTOTAL:= - CFTOTAL;
28660:          END
28670:      ELSE IF NOT MODIFICADOR THEN CFTOTAL:=CFMAIOR + CFIGUAL
28680:      ELSE CFTOTAL:=CFMENOR;
28690:      12: % MENOR=IGUAL
28700:      MAIORMENOR;
28710:      IF TIPOATRIBUTO = 1
28720:      THEN BEGIN
28730:          CFTOTAL:=CFMENOR + CFIGUAL - CFMAIOR;
28740:          IF MODIFICADOR THEN CFTOTAL:= - CFTOTAL;
28750:          END
28760:      ELSE IF NOT MODIFICADOR
28770:      THEN CFTOTAL:=CFMENOR + CFIGUAL
28780:      ELSE CFTOTAL:=CFMAIOR;
28790:
28800:      13: % DIFERENTE
28810:      IF TIPOATRIBUTO = 0
28820:      THEN BEGIN
28830:          TRATASIMNAD;
28840:          CFTOTAL:= - CFTOTAL;
28850:          END

```

```

288600      ELSE BEGIN
288700          IGUALDIFERENTE;
288800          IF TIPOATRIBUTO = 1
288900              THEN BEGIN
289000                  CFTOTAL:=CFDIF - CFIGUAL;
289100                  IF MODIFICADOR THEN CFTOTAL:= - CFTOTAL;
289200                  END
289300              ELSE IF NOT MODIFICADOR THEN CFTOTAL:=CFDIF
289400                  ELSE CFTOTAL:=CFIGUAL;
289500              END;
289600      END;
289700      DESIGUALDADE:=CFTOTAL GEQ .2;
289800      RESULT:=CFTOTAL;
289900      IF TRACE THEN WRITE(SAIDA, <"/>,"* FIM DE DESIGUALDADE COM RESULT =">
290000          F7.2>, RESULT);
290100      END;
290200
290300      *-----*
290400
290500      BOOLEAN PROCEDURE IGUAL;
290600      BEGIN
290700
290800      COMMENT:
290900          **   IGUAL   **
291000          ----
291100          E' AVALIADA COMO VERDADEIRA, SE O VALOR ARMAZENADO
291200          NA BASE DE FATOS PARA DETERMINADOS ATRIBUTO E
291300          CONTEXTO E' IGUAL AO VALOR DADO. O CF RESULTANTE
291400          E' RETORNADO EM "RESULT" .
291500
291600          IF TIPOATRIBUTO = 0 THEN TRATASIMNAD
291700              ELSE BEGIN
291800                  IGUALDIFERENTE;
291900                  IF TIPOATRIBUTO = 1
292000                      THEN IF NOT MODIFICADOR
292100                          THEN CFTOTAL:=CFIGUAL - CFDIF
292200                              ELSE CFTOTAL:=CFDIF - CFIGUAL
292300                      ELSE IF NOT MODIFICADOR
292400                          THEN CFTOTAL:=CFIGUAL
292500                              ELSE CFTOTAL:=CFDIF
292600                      END;
292700                  CASE FUNC OF
292800                      BEGIN
292900                          1: IGUAL:= CFTOTAL GEQ 0.2 ;
293000                          4: IGUAL:= CFTOTAL GEQ -0.2 ;
293100                          5:7: IGUAL:=CFTOTAL = 1 ;
293200                      END;
293300                      RESULT:=CFTOTAL;
293400                      IF TRACE THEN WRITE(SAIDA, <"/>,"* FIM DE IGUAL COM RESULT =">
293500                          F7.2>, RESULT);
293600                      END PROCEDURE IGUAL;
293700
293800      *-----*
293900
294000
294100      % *** INICIO DE AVALIAR ***
294200      % -----
294300
294400      COMMENT :

```

```

29450: AVALIAR VERIFICA A VALIDADE DE UMA DADA CLAUSULA. SE A CLAUSULA
29460: FOR SIMPLES BUSCA-SE O VALOR DO ATRIBUTO NO ARQUIVO VALORCONHECIDO
29470: CASO ESTE NAO SEJA ENCONTRADO ELF E BUSCADO USANDO-SE BUSCAVALOR
29480: DEPOIS TOMA-SE O VALOR NO VALORCONHECIDO .
29490: PROCURA-SE O VALOR DO ATRIBUTO PARA O CONTEXTO INSTANCIADO NA
29500: CLAUSULA E CASO O FATOR DE DESCRENCA NAO SEJA 1 NORMALIZA-SE O
29510: VALOR E AVALIA-SE ATRAVES DA FUNCAO SE OS VALORES DO NAO CASAM
29520: COM OS DA CLAUSULA ;
29530:
29540: READ(DIRDISK(ATRIBUTO),10,DIRETORIO);
29550: ATRIBU:=SINONIMO;
29560: ACHACONTEXTUAL;
29570: IF TRACE THEN WRITE(SAIDA, </,"* AVALIAR *",/," CLAUSULA =",I3,
29580: " SUBC =",I3," REGRA =",I3," CONTEXTO =",I3," NCL, NSUBCL, NREGRA,
29590: CONTEXTUAL);
29600: RESULT:=1;
29610: IF REGRAESPECIAL THEN BEGIN
29620:   AVALIACAO:=TRUE;
29630:   ATUALIZAVALORESCONSULTA(ATRIBU,CONTEXTUAL,VALOR,
29640:     TIPOVALOR,UNIDADE,1,0,NOATUAL,NUMCONCLUSAO,MODFUNCAO);
29650:   END
29660:
29670: ELSE BEGIN
29680:   BOOLEAN DP;
29690:   DP:=FALSE;
29700:   VERIFICASEEXISTE(ATRIBU,CONTEXTUAL,TEMVALOR,DESCONHECIDO,
29710:     CONT);
29720:   *
29730:   * CASO NAO SE ENCONTRE VALOR EM "VALORCONHECIDO" PARA O ATRIBUTO
29740:   * "BUSCAVALOR" NAO SERA CHAMADO SE A FUNCAO FOR "CONFIRMADO"(FUNC=7)
29750:   *
29760:   IF NOT TEMVALOR AND NOT DESCONHECIDO
29770:   THEN IF FUNC NEQ 7 THEN BEGIN
29780:     BUSCAVALOR(ATRIBU,CONTEXTUAL,NREGRA,PREGRAPAI,
29790:       PIRMAO,DP,NCL,NSUBCL,NUMCONCLUSAO);
29800:     IF PRIMEIRAVEZ AND DP THEN
29810:     BEGIN
29820:       PRIMEIRAVEZ:=FALSE;
29830:       RFILHO(PREGRAPAI):=PREGRAPAI+1;
29840:     END;
29850:     VERIFICASEEXISTE(ATRIBU,CONTEXTUAL,TEMVALOR,DESCONHECIDO,
29860:       CONT);
29870:   END;
29880:   IF NOT TEMVALOR OR DESCONHECIDO THEN AVALIACAO:=FALSE
29890:   ELSE BEGIN
29900:     NORMALIZAVALOR;
29910:     READ(DIRDISK(FUNCAO),10,DIRETORIO);
29920:     FUNC:=SINONIMO;
29930:     READ(DIRDISK(ATRIBU),10,DIRETORIO);
29940:     READ(VALORCONHECIDO[CHAVEREG/2],30,CONHECO);
29950:     IF MODFUNCAO NEQ 0 THEN MODIFICADOR:=TRUE;
29960:     CASE FUNC OF BEGIN
29970:       1:4:5:7: AVALIACAO:=IGUAL;
29980:       6: AVALIACAO:=CONHECIDO;
29990:       8: * MOSTRAPREMISSA;
30000:     9:10:11:12:13: AVALIACAO:=DESIGUALDADE;
30010:     END;
30020:   END;
30030: END;

```

```

30040: TALLY:=RESULT;
30050: CLAUSULAFALSA:=NOT AVALIACAO;
30060: IF TRACE THEN WRITE(SAIDA, <"/>,"* FIM DE AVALIAR CLAUSULA ",I3,
30070: " SUBCLAUSULA ",I3," REGRA ",I3,<"/>," AVALIACAO =",L1," RESULT =",
30080: F7.2>>NCL,NSUBCL,NREGRA,AVALIACAO,RESULT);
30090: END PROCEDURE AVALIAR;
30100:
30110:
30120: * *** INICIO DE AVALIA ***
30130: * -----
30140:
30150: COMMENT:
30160: O PROCEDIMENTO AVALIA VERIFICA A VALIDADE DE UMA DADA REGRA .
30170: A CERTEZA NUMA REGRA INICIALMENTE E 1 .PARA CADA CLAUSULA ,SE FOR
30180: AVALIADA COMO VERDADEIRA ,A CERTEZA E TOMADA COMO SENDO O MINIMO
30190: ENTRE A CERTEZA ANTERIOR E A DA CLAUSULA (TALLY). NO CASO DE
30200: DISJUNCOES DE CLAUSULAS ,A CERTEZA E O MAXIMO (TALLY) DAS CERTEZAS DA
30210: CLAUSULAS DISJUNTIVAS. A DISJUNCAO E FALSA SE TODAS AS CLAUSULAS
30220: FOREM FALSAS .A CONJUNCAO E FALSA SE PELO MENOS UMA E FALSA;
30230:
30240: IF TRACE
30250: THEN WRITE(SAIDA, <"/>,"* AVALIA ",I3," NREGRA =",I3,
30260: " NUMCONCLUSAO =",I3," CNTXT =",I3," REGRAPAI =",I3,
30270: <"/>," ACAO =",I3," TUDOOK = ",L1>>NREGRA,NUMCONCLUSAO,
30280: CNTXT,REGRAPAI,ACAO,TUDOOK);
30290: READ(REGRASDISK[NREGRA],150,REGRA);
30300:
30310: CONTINUA:=FALSE;
30320: CONCLUD:=ABS(ACAO);
30330: TUDOOK:=FALSE;
30340: CLAUSULAFALSA:=FALSE;
30350: CERTEZA:=1.;
30360: PRIMEIRAVEZ:=TRUE;
30370: PREGRAPAI:=PREGRA;
30380: PTIRMAO:=0;
30390:
30400: NCL:=0;
30410: PCL(NDATUAL):=0;
30420: IF REGRAESPECIAL THEN BEGIN
30430: I:=1;
30440: X:=PROXC(EENTAO);
30450: WHILE I NEQ NUMCONCLUSAO
30460: DO BEGIN
30470: X:=PROXC(X);
30480: I:=I+1;
30490: END;
30500: FOR I:=0 STEP 1 UNTIL 3
30510: DO CLAUSULA[I]:=REGRA[X + I];
30520: ACHACONTEXTODATUAL;
30530: VERIFICASEEXISTE(ATRIBU,CONTEXTODATUAL,TEMVALOR,
30540: DESCONHECIDO,CONT);
30550: IF TEMVALOR THEN BEGIN
30560: INTEGER FAT;
30570: READ(DIRDISK[ATRIBUTO],10,DIRETORIO);
30580: READ(VALORCONHECIDO[CHAVERFG/2],30,CONHECIDO);
30590: FAT:=KMB(CONT) - KMD(CONT);
30600: IF EMODFUNCAO NEQ 0 THEN FAT:= -FAT;
30610: CONTINUA:=IF FAT = 1 THEN TRUE ELSE FALSE;
30620: END;

```



```

306300      END;
306400      X:=PROXC(EPREMISSA);
306500      IF NOT REGRAESPECIAL OR CONTINUA THEN
306600        WHILE MARCA(X) NEQ 1 DO
306700
306800      * AVALIACAO DE DISJUNCAO DE CLAUSULAS :
306900
307000      IF MARCA(X) = 3 THEN BEGIN
307100        CFMAX:=0;
307200        PELOMENOSUM:=FALSE;
307300        NCL:=NCL+1;
307400        RCL(NDATUAL):=NCL;
307500        NSUBCL:=0;
307600        RSUBCL(NDATUAL):=NSUBCL;
307700        IF DU1(X) = 1 THEN BEGIN
307800          INTEGER DESCRIPTOR;
307900          DESCRITOR:=X;
308000          X:=PROXC(X);
308100          SAIU:=FALSE;
308200          WHILE MARCA(X) NEQ 3 AND NOT SAIU DO BEGIN
308300            FOR I:=0 STEP 1 UNTIL 2 DO CLAUSULA[I]:=REGRA[X+I];
308400            NSUBCL:=NSUBCL+1;
308500            RSUBCL(NDATUAL):=NSUBCL;
308600            AVALIAR;
308700            IF NOT CLAUSULAFALSA THEN BEGIN
308800              CFMAX:=MAX(CFMAX,TALLY);
308900              PELOMENOSUM:=TRUE;
309000              X:=DESCRIPTOR;
309100              SAIU:=TRUE;
309200            END
309300            ELSE X:=PROXC(X);
309400          END
309500        END
309600      ELSE BEGIN
309700        X:=PROXC(X);
309800        WHILE MARCA(X) NEQ 3 DO BEGIN
309900          FOR I:=0 STEP 1 UNTIL 2 DO CLAUSULA[I]:=REGRA[X+I];
310000          NSUBCL:=NSUBCL + 1;
310100          RSUBCL(NDATUAL):=NSUBCL;
310200          AVALIAR;
310300          IF NOT CLAUSULAFALSA THEN BEGIN
310400            CFMAX:=MAX(CFMAX,TALLY);
310500            PELOMENOSUM:=TRUE;
310600          END;
310700          X:=PROXC(X);
310800        END;
310900      END;
311000      IF NOT PELOMENOSUM THEN GO FIMAVALIA;
311100      CERTEZA:=MIN(CERTEZA,CFMAX);
311200      X:=PROXL(X);
311300      END
311400
311500      * AVALIACAO DE CONJUNCAO DE CLAUSULAS :
311600
311700      ELSE BEGIN
311800        FOR I:=0 STEP 1 UNTIL 2 DO CLAUSULA[I]:=REGRA[X+I];
311900        NCL:=NCL + 1;
312000        RCL(NDATUAL):=NCL;
312100        AVALIAR;

```

```

31220:         IF CLAUSULAFALSA THEN GO FIMAVALIA;
31230:         CERTEZA:=MIN(CERTEZA,TALLY);
31240:         X:=PROXC(X);
31250:     END;
31260:
31270: COMMENT:
31280: SE A CERTEZA FOR MAIOR QUE 0.2 A CERTEZA FINAL SERA CERTEZA*FATOR
31290: DA CONCLUSAO E OS VALORES SAO ATUALIZADOS (CASO CONTRARIO A REGRA
31300: E AVALIADA COMO FALSA (TUDDOK=F) )
31310:
31320: IF NOT REGRAESPECIAL THEN BEGIN
31330:     IF CERTEZA < 0.2 THEN GO FIMAVALIA;
31340:     I:=1;
31350:     X:=PROXC(EENTAO);
31360:     WHILE I NEQ NUMCONCLUSAO
31370:     DO BEGIN
31380:         X:=PROXC(X);
31390:         I:=I+1;
31400:     END;
31410:     FOR I:=0 STEP 1 UNTIL 3
31420:     DO CLAUSULA[I]:=REGRA[X + I];
31430:     CERTEZA:=CERTEZA * FATOR;
31440:     RCERTEZA(NDATUAL):=CERTEZA ;
31450:     ACHACONTEXTODATUAL;
31460:
31470:     READ(DIRDISK(ATRIBUTO),10,DIRETORIO);
31480:     ATRIBU:=SINONIMO;
31490:     ATUALIZAVALORESCONSULTA(ATRIBU,CONTEXTODATUAL,VALOR,TIPOVALOR,
31500:         UNIDADE,CERTEZA,0,NDATUAL,NUMCONCLUSAO,MODFUNCAO);
31510:
31520:     TUDDOK:=TRUE;
31530: END;
31540:
31550: FIMAVALIA:
31560: IF TRACE THEN WRITE(SAIDA,</, "& FIM DE AVALIA DA REGRA",I3," &"),
31570:     NREGRA);
31580: END PROCEDURE AVALIA;
31590:
-----
31600:
31610:
31620: PROCEDURE MONITOR(NUMEROREGRA,NUMCONCLUSAO,REGRAPAI,CONCLUSAO,
31630:     PREGRAANTERIOR);
31640: VALUE NUMEROREGRA,NUMCONCLUSAO,CONCLUSAO;
31650: INTEGER NUMEROREGRA,NUMCONCLUSAO,REGRAPAI,CONCLUSAO,PREGRAANTERIOR;
31660: BEGIN
31670:
31680:     INTEGER I,J,K,X,N,CNTXT,PAPLANTERIOR,SALVO;
31690:     INTEGER ARRAY CONTEXTODATUAL[1:20];
31700:     BOOLEAN TUDDOK;
31710:     ARRAY REGPA[0:149];
31720:     DEFINE COLOCAREGRA =
31730:     PREGPA:=**+1;
31740:     RREGPA(PREGPA):=NUMEROREGRA;
31750:     RPAI(PREGPA):=RPAI(PAPLANTERIOR);
31760:     RCONCLUSAO(PREGPA):=NUMCONCLUSAO;
31770:     RVALOR(PREGPA):=EVALOR;
31780:     IF MODFUNCAO = 0 THEN RNEGACAO(PREGPA):=0
31790:     ELSE RNEGACAO(PREGPA):=1;
31800:     RTIPO(PREGPA):=0;

```

```

31810:  RATRIBUTO(PREGRA):=ABS(CONCLUSAO);
31820:  RCONTEXTID(PREGRA):=CNTXT;
31830:  RPRDXAPLICACAO(PAPLANTERIOR):=PREGRA;
31840:  PAPLANTERIOR:=PREGRA;
31850:  SALVO:=PREGRA; #
31860:
31870:  COMMENT :
31880:  'ETIPOPREGRA' INDICA O TIPO DE CONTEXTO QUE DEVE EXISTIR NA ARVORE
31890:  PARA QUE 'REGRA' POSSA SER APLICADA. NO CASO DE EXISTIR MAIS DE
31900:  UM, A REGRA SERA USADA MAIS DE UMA VEZ. CASO NAO EXISTA CONTEXTO
31910:  A ARVORE SERA EXPANDIDA;
31920:
31930:  -----
31940:
31950:  PROCEDURE CONTEXTODEAPLICACAO(TIPOCERTO);
31960:  INTEGER TIPOCERTO;
31970:  BEGIN
31980:
31990:      COMMENT :
32000:          ** CONTEXTODEAPLICACAO **
32010:          -----
32020:          ESSE PROCEDIMENTO VERIFICA SE EXISTEM NA ARVORE DE CONTEXTOS
32030:          CONTEXTOS DE TIPOS IGUAIS A 'TIPOCERTO'. OS CONTEXTOS ENCON-
32040:          TRADOS SERAO ARMAZENADOS EM 'CONTEXTOATUAL[1..N]';
32050:
32060:      INTEGER P;
32070:
32080:      IF TRACE
32090:      THEN WRITE(SAIDA, <"/,"* VERIFICACONTEXTOLAPLICACAO *"/,"
32100:      " TIPOCERTO =",I3>,"TIPOCERTO);
32110:      N:=0;
32120:      FOR P:=1 STEP 1 UNTIL AVAIL-1 DO
32130:          IF TIPO(P) = TIPOCERTO THEN BEGIN
32140:              N:=N+1;
32150:              CONTEXTOATUAL[N]:=P END;
32160:      END PROCEDURE VERIFICACONTEXTODEAPLICACAO;
32170:
32180:  * INICIO DE MONITOR
32190:  * -----
32200:
32210:  COMMENT :
32220:  ACHAM-SE TODOS OS TIPOS DE CONTEXTO APLICAVEIS NAQUELA CONCLUSAO.
32230:  PARA CADA DESTES TIPOS VERIFICA-SE SE HA NA ARVORE ALGUM CONTEXTO
32240:  IGUAL AQUELE. SE NAO HOUVER, CRIA-SE O CONTEXTO. PARA CADA UM DOS
32250:  CONTEXTOS PERTINENTES E FEITA UMA AVALIACAO DA REGRA ;
32260:
32270:
32280:  PAPLANTERIOR:=0;
32290:  RPAI(PAPLANTERIOR):=RPAI(PREGRAANTERIOR);
32300:  RPROXIMO(PREGRAANTERIOR):=PREGRA+1;
32310:  PREGRAANTERIOR:=PREGRA+1;
32320:  IF TRACE THEN
32330:      WRITE(SAIDA, <"/,"* MONITOR *"/," NUMEROREGRA =",I3,
32340:      " NUMCONCLUSAO =",I3," REGRAPAI =",I3, <"/," CONCLUSAO =",
32350:      I3,I1>,"NUMEROREGRA,NUMCONCLUSAO,REGRAPAI,CONCLUSAO);
32360:  READ(REGRASDISK(NUMEROREGRA),150*REGRA);
32370:  * ACHA A LISTA DE TIPOS DA CONCLUSAO CORRENTE
32380:  J:=1;
32390:  II:=PROXC(EENTAO);

```

```

32400: WHILE J NEQ NUMCONCLUSAO
32410: DO BEGIN
32420:   I:=PROXC(I);
32430:   J:=J+1;
32440: END;
32450: FOR X:=1 STEP 1 UNTIL 6 DO
32460: IF ETIPOREGRA(X) NEQ 0 THEN BEGIN
32470:
32480:   CONTEXTUDEAPLICACAO(ETIPOREGRA(X));
32490:   IF N NEQ 0 THEN
32500:     FOR K:=1 STEP 1 UNTIL N DO BEGIN
32510:       CNTXT:=CONTEXTOATUAL(K);
32520:       COLOCAREGRA;
32530:       TUDOOK:=FALSE;
32540:       AVALIA(NUMEROREGRA,NUMCONCLUSAO,CNTXT,REGRAPI,CONCLUSAO,
32550:         TUDOOK,SALVO);
32560:       IF NOT TUDOOK THEN RSUCFSSO(SALVO):=1;
32570:     END
32580:   ELSE BEGIN
32590:     CRIACONTEXTO(ETIPOREGRA(X),REGRAPI);
32600:     IF CONTEXTOCRIADO THEN BEGIN
32610:       CONTEXTUDEAPLICACAO(ETIPOREGRA(X));
32620:       FOR K:=1 STEP 1 UNTIL N DO BEGIN
32630:         CNTXT:=CONTEXTOATUAL(K);
32640:         COLOCAREGRA;
32650:         TUDOOK:=FALSE;
32660:         AVALIA(NUMEROREGRA,NUMCONCLUSAO,CNTXT,REGRAPI,
32670:           CONCLUSAO,TUDOOK,SALVO);
32680:         IF NOT TUDOOK THEN RSUCCESSO(SALVO):=1;
32690:       END;
32700:     END;
32710:   END;
32720: END;
32730: IF TRACE THEN WRITE(SAIDA,<"/>
32740:   NUMEROREGRA);
32750: WRITE(SAIDA,<"* MONITOR - REGRA ">I3>,NUMEROREGRA);
32760: ESCREVEEMREGRA;
32770: END PROCEDURE MONITOR;
32780:
32790: -----
32800: BEGIN
32810:   X CABECALHO
32820:   X -----
32830:   STRING MESES,SEMANA;
32840:
32850:   MESES:="JANFEV MARABR MAI JUN JUL AGO SET OUT NOV DEZ";
32860:   SEMANA:="DOMINGO,SEGUNDA-FEIRA,TERCA-FEIRA,QUARTA-FEIRA," CAT
32870: "QUINTA-FEIRA,SEXTA-FEIRA,SABADO";
32880:   T:=TIME(7);
32890:   THRU DIA SEMANA DO SEMANA:=DROP(TAIL(SEMANA,NOT ",",1));
32900:   THRU MES:=1 DO MESES:=DROP(MESES,3);
32910:   WRITE(VIDEO,<2A1>,LIMPATELA,BUZINA);
32920:   WRITE(VIDEO,<"/>X25,"MODULO DE CONSULTA">///,X25,
32930:     "< SESSAO ABERTA >">///,X13,A15,I5,"/">A3,"/">I4,
32940:     I5,"!">J2,"!">J2,///,72(">"/>
32950:     HEAD(SEMANA,NOT ",",DIA,TAKE(MESES,3),ANO,HORA,
32960:     MINUTOS,SEGUNDOS);
32970: END;
32980:

```

```

32990: COMMENT:
33000:          ** PROGRAMA PRINCIPAL **
33010:          -----
33020:          ESTE PROGRAMA E' A MAQUINA DE INFERENCIA DO SISTEMA.
33030:          O ENCADEAMENTO DE REGRAS E' FEITO DE MODO RETRATIVO
33040:          ATRAVES DE DOIS PROCEDIMENTOS BASICOS: BUSCAVALOR E
33050:          MONITOR. NESTE PROGRAMA TAMBEM E' ACESSADO O MODULO
33060:          DE EXPLANACAO DO SISTEMA ATRAVES DO USO DA LINGUAGEM
33070:          DE COMANDOS, AO INVEZ DE SE RESPONDER A PERGUNTA
33080:          FORMULADA PELO SISTEMA ;
33090:
33100:          WRITE(VIDEO,"** ENTRE COM O DOMINIO DE APLICACAO: ",/);
33110:          WRITE(VIDEO,[STOP]," > ");
33120:          READ(TERM,"A30",DOMINIO);
33130:          DOMINIO:=HEAD( TAIL(DOMINIO, " "), NOT " ") CAT "/";
33140:
33150:          MUDA(TEGRASDISK,"TEGRASDISK");
33160:          MUDA(TEXTODISK,"TEXTODISK");
33170:          MUDA(ATRIBUTOSREF,"ATRIBUTOSREF");
33180:          MUDA(ESQUELETODISK,"ESQUELETODISK");
33190:          MUDA(DIRDISK,"DIRDISK");
33200:          MUDA(OUTRASPALAVRAS,"OUTRASPALAVRAS");
33210:          MUDA(TIPOSCONTEXTO,"TIPOSCONTEXTO");
33220:          MUDA(OBJETOETIPOS,"OBJETOETIPOS");
33230:          MUDA(PROPATRIBUTOS,"PROPATRIBUTOS");
33240:          MUDA(OBJETIVOS,"OBJETIVOS");
33250:          MUDA(CLASEDISK,"CLASEDISK");
33260:          MUDA(VALORCONHECIDO,"VALORCONHECIDO");
33270:
33280:          READ(TEXTODISK[0],90,CLAUSE);
33290:          WRITE(TEXTODISK[0],90,CLAUSE);
33300:          READ(OBJETOETIPOS[0],4,LISTADETIPOS);
33310:          WRITE(OBJETOETIPOS[0],4,LISTADETIPOS);
33320:          READ(CLASEDISK[0],20,CLASS);
33330:          WRITE(CLASEDISK[0],20,CLASS);
33340:
33350:          IF VALORCONHECIDO.RESIDENT
33360:             THEN REMOVEFILE(POINTER(NOMEDOMINIO));
33370:          CLOSE(VALORCONHECIDO);
33380:          VALORCONHECIDO.NEWFILE:=TRUE;
33390:          VALORCONHECIDO.MYUSE:=2;
33400:          VALORCONHECIDO.OPEN:=TRUE;
33410:          WRITE(VALORCONHECIDO[0],30,CONHECO);
33420:          LOCK(VALORCONHECIDO);
33430:          VALORCONHECIDO.MYUSE:=3;
33440:          VALORCONHECIDO.UPDATEFILE:=FALSE;
33450:          WRITE(VALORCONHECIDO[0],30,CONHECO);
33460:          READ(VALORCONHECIDO[0],30,CONHECO);
33470:
33480:          BEGIN                                % OBJETIVO DA CONSULTA
33490:                                             % -----
33500:          PEAL VALOR;
33510:          INTEGER I;
33520:          BOOLEAN ACHEI;
33530:
33540:          WRITE(VIDEO,"/","** OBJETIVOS POSSIVEIS : ",/);
33550:          DO BEGIN
33560:             I:=1;
33570:             WHILE NOT READ(OBJETIVOS[I],14,OBJETIVO) DO BEGIN

```

```
33580: WRITE(VIDEO,<I3>," : ",A66>>I,POINTER(OBJETIVO[31]));
33590: I:=I+1;
33600: END;
33610: WRITE(VIDEO,<///,"** QUAL O OBJETIVO DA CONSULTA :"/>);
33620: PEGA(PALAVRA);
33630: ACHEI:=FALSE;
33640: IF NUMERO
33650: THEN BEGIN
33660: VAL:=DECIMAL(PALAVRA);
33670: IF VAL>=0 AND VAL<=OBJETIVOS.LASTRECORD
33680: THEN BEGIN
33690: READ(OBJETIVOS[VAL],14,OBJETIVO);
33700: ACHEI:=TRUE;
33710: END
33720: ELSE
33730: END
33740: ELSE ERRO("SUA RESPOSTA DEVE SER NUMERICA");
33750: END
33760: UNTIL ACHEI;
33770: END;
33780:
33790: TRACE:=FALSE;
33800: WRITE(VIDEO,<///,"** DESEJA ACOMPANHAR A EXECUCAO ?(S/N)"/>);
33810: WRITE(VIDEO[STOP],<" >"/>);
33820: READ(TERM,<A20>,LINHA);
33830: IF HEAD(TAIL(LINHA," ")) NOT " " THEN TRACE:=TRUE;
33840: WRITE(VIDEO,<2A1>,LIMPAATELA,BUZINA);
33850:
33860: READ(REGRASDISK[0],150,DESATIVADA);
33870: PREFECHECOMANDOS;
33880: * TOMADA DA DATA E HORA ATUAL
33890: T:=TIME(7);
33900: VALH:=HORA;
33910: VALD:=DIA;
33920: VALM:=MES;
33930:
33940: REINICIA :
33950:
33960: * LIMPA ARRAYS E ARQUIVOS DINAMICOS
33970: BEGIN
33980: INTEGER I,J;
33990: FILL ARVORECONTEXTOS[*] WITH 128(0);
34000: FOR I:=0 STEP 1 UNTIL 99 DO
34010: FOR J:=0,1,2,3 DO REGRACONSULTA[J,I]:=0;
34020: FILL CONHECO[*] WITH 30(0);
34030: FOR I:=0 STEP 1 UNTIL 99 DO
34040: IF REGACESSADO[I] THEN WRITE(VALORCONHECIDO[I],30,CONHECO);
34050: FOR I:=0 STEP 1 UNTIL 99 DO NADTEMCONTEXTID[I]:=FALSE;
34060: FOR I:=0 STEP 1 UNTIL 99 DO REGACESSADO[I]:=FALSE;
34070: FILL VETORCONCLUSOES[*] WITH 15(0);
34080: FILL LASTCNTXT[*] WITH 50(0);
34090: END;
34100: * ATUALIZA HORA ATUAL
34110: ATUALIZAVALORESCONSULTA(22,1,VALH,1,0,1,0,0,0,0,0);
34120: * ATUALIZA DIA ATUAL
34130: ATUALIZAVALORESCONSULTA(23,1,VALD,1,0,1,0,0,0,0,0);
34140: * ATUALIZA MES ATUAL
34150: ATUALIZAVALORESCONSULTA(24,1,VALM,1,0,1,0,0,0,0,0);
34160: PREGRA:=0;
```

```

34170: P:=0;
34180: PVETOR:=0;
34190: NUMPERGUNTA:=0;
34200: AVAIL:=1;
34210: COUNTER:=0;
34220: CRIACONTEXT(1,0);
34230: PI:=TIPO(1);
34240: READ(DIRDISK(OBJETIVO(1)),10,DIRETORIO);
34250: UCTIMALINHA:=0;
34260: ATRIBUTOOBJETIVO:=SINONIMO;
34270: ZERO:=0;
34280: FALSO:=FALSE;
34290:
34300: BUSCAVALOR(OBJETIVO(1),1,0,ZERO,ZERO,FALSO,0,0,0);
34310: READ(DIRDISK(ATRIBUTOOBJETIVO),10,DIRETORIO);
34320: READ(VALORCONHECIDO[CHAVEPG/2],30,CONHECO);
34330: WRITE(VIDEO,<2A1>LIMPA TELA,BUZINA);
34340: WRITE(VIDEO,<///>X25,"CONCLUSOES :","/");
34350: IF PVETOR = 0 THEN BEGIN
34360:   LINHA:="INFELIZMENTE NAO FOI POSSIVEL DEDUZIR " CAT
34370:   STRING(POINTER(OBJETIVO(3)),72);
34380:   MOSTREMENSAGEM(LINHA);
34390:   END
34400: ELSE BEGIN
34410:   TIPOATRIB:=TIPOATRIBUTO;
34420:   IF PVETOR = 1 THEN LINHA:="VALOR" ELSE LINHA:="VALORES";
34430:   LINHA:=LINHA CAT " DE " CAT STRING(POINTER(OBJETIVO(3)),72)
34440:   CAT " = ";
34450:   MOSTREMENSAGEM(LINHA);
34460:   WRITE(VIDEO,<///>);
34470:   FOR AUX:=1 STEP 1 UNTIL PVETOR DO BEGIN
34480:     COUNTER:=VETORCONCLUSOES[AUX];
34490:     CFMI:=(KMB(COUNTER) - KMD(COUNTER))/100;
34500:     CASE TIPOATRIB OF BEGIN
34510: 0: % TIPO SIM/NAO
34520:   IF CFMI > 0 THEN LINHA:="VERDADEIRO" ELSE LINHA:="FALSO";
34530:   CFMI:=ABS(CFMI);
34540: 1:2: % TIPOS UNICO E MULTIPLO
34550:   CASE ASSOC OF BEGIN
34560:   0: % MENSAGEM
34570:     READ(TEXTODISK[KVALOR(COUNTER)],<A540>,LINHA);
34580:     LINHA:=HEAD(LINHA,"NOT ");
34590:   1: % NUMERICO SEM UNIDADE
34600:     LINHA:=STRING(KVALOR(COUNTER),*);
34610:   2: % NAO-NUMERICO
34620:     READ(DIRDISK[KVALOR(COUNTER)],10,DIRETORIO);
34630:     LINHA:=DIRNOME;
34640:   ELSE: % NUMERICO COM UNIDADE
34650:     READ(DIRDISK[ASSOC],10,DIRETORIO);
34660:     LINHA:=STRING(KVALOR(COUNTER),*) CAT " " CAT DIRNOME;
34670:   END;
34680: END;
34690:
34700: LINHA:=STRING(AUX,*) CAT " ) " CAT LINHA CAT " (FATOR = "
34710:   CAT STRING(CFMI,*) CAT " )";
34720:   MOSTREMENSAGEM(LINHA);
34730:   END;
34740: END;
34750: FIMDETUDO ;
34760: LOCK(VALORCONHECIDO,CRUNCH);
34770: END.

```