

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**PROTÓTIPO DE UM SISTEMA ESPECIALISTA UTILIZANDO
A FERRAMENTA EXPERT SINTA SHELL PARA AUXÍLIO NO
SETOR DE SUPORTE DE UMA *SOFTWARE HOUSE***

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO — BACHARELADO

ADRIANA BOMBASSARO ALEXANDRE

BLUMENAU, JUNHO/2000

2000/1-02

PROTÓTIPO DE UM SISTEMA ESPECIALISTA UTILIZANDO A FERRAMENTA EXPERT SINTA SHELL PARA AUXÍLIO NO SETOR DE SUPORTE DE UMA SOFTWARE HOUSE

ADRIANA BOMBASSARO ALEXANDRE

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Roberto Heinzle — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Roberto Heinzle

Prof. Marcel Hugo

Prof. Oscar Dalfovo

*Este trabalho de conclusão de curso
é dedicado à **Nivaldo**,
pelo amor carinho, apoio e incentivos
recebidos ao longo destes anos de graduação.*

*Àqueles que me incentivaram
MOACIR e ENEIDA,
meus irmãos e amigos*

AGRADECIMENTOS

Ao Professor e Orientador Roberto Heinzle pelo acompanhamento e incentivo na realização do trabalho.

A todos os amigos, professores e colegas do curso de Ciências da Computação pelo incentivo, ajuda, apoio e compreensão recebidos durante os anos de graduação.

SUMÁRIO

SUMÁRIO	V
LISTA DE FIGURAS	IX
LISTA DE TABELAS	X
RESUMO	XI
ABSTRACT	XII
1 INTRODUÇÃO	1
2 SISTEMAS ESPECIALISTAS	5
2.1 CONCEITOS.....	5
2.2 CARACTERÍSTICAS	6
2.3 COMPONENTES DE UM SISTEMA ESPECIALISTA.....	6
2.3.1 BASE DE CONHECIMENTOS	7
2.3.2 MECANISMO DE APRENDIZAGEM E AQUISIÇÃO DO CONHECIMENTO.....	8
2.3.3 MOTOR OU MÁQUINA DE INFERÊNCIA	8
2.3.4 SISTEMA DE CONSULTA.....	9
2.3.5 SISTEMA DE JUSTIFICAÇÃO	9
2.3.6 QUADRO NEGRO.....	10
2.4 REPRESENTAÇÃO DO CONHECIMENTO.....	10
2.5.1 ARQUITETURA DE UM SISTEMA ESPECIALISTA NO EXPERT SINTA SHELL.....	12
2.5.2 UTILIZANDO REGRAS DE PRODUÇÃO NO EXPERT SINTA SHELL	13
2.5.3 O MÉTODO DE EXTRAÇÃO DE CONHECIMENTO DO EXPERT SINTA SHELL	14
2.5.4 VARIÁVEIS UNIVALORADAS E VARIÁVEIS MULTIVALORADAS	15
2.5.5 FATORES DE CONFIANÇA	15
2.5.6 O CÁLCULO DE PROBABILIDADES NO EXPERT SINTA.....	16
2.5.7 GERENCIANDO BASES DE DADOS	17

2.6 FORMALIZAÇÃO DO CONHECIMENTO	17
2.7 EXPERT SINTA VISUAL COMPONENT LIBRARY (VCL).....	18
2.7.1 COMPONENTES DO VCL	19
2.7.2 RELAÇÃO ENTRE OS COMPONENTES	20
3 BANCO DE DADOS	23
3.1 CONCEITOS BÁSICOS	23
3.2 BANCO DE DADOS RELACIONAL	23
3.3 CARACTERÍSTICAS DE UM BANCO DE DADOS RELACIONAL	24
3.4 BANCO DE DADOS DISTRIBUÍDO	24
3.4.1 CARACTERÍSTICAS DE UM BANCO DE DADOS DISTRIBUÍDO	25
4 DATA MINING.....	28
4.1 DEFINIÇÃO.....	28
4.2 PROSPECÇÃO DO CONHECIMENTO	28
4.3 PROCESSO DE KDD	29
4.4 TAREFAS DO DATA MINING	31
4.4.1 CLASSIFICAÇÃO	31
4.4.2 ESTIMATIVA.....	32
4.4.3 AGRUPAMENTO POR AFINIDADE.....	33
4.4.4 PREVISÃO	33
4.4.5 SEGMENTAÇÃO	34
4.5 TÉCNICAS DE DATA MINING.....	34
4.5.1 ANÁLISE DE SELEÇÃO ESTATÍSTICA	37
4.5.2 MBR	37
4.5.3 ALGORITMOS GENÉTICOS	38
4.5.4 ANÁLISE DE GRUPOS.....	38

4.5.5 ANÁLISE DE VÍNCULOS	39
4.5.6 ÁRVORES DE DECISÃO E INDUÇÃO DE REGRAS.....	39
4.5.7 REDES NEURAI ARTIFICIAIS	39
5 SISTEMA DE GERENCIAMENTO A SUPORTE	41
5.1 ESTRUTURA DO SISTEMA DE GERENCIAMENTO A SUPORTE	41
5.2 MODELO ENTIDADE RELACIONAMENTO DO SISTEMA DE GERENCIAMENTO A SUPORTE.....	43
6 O PROTÓTIPO DESENVOLVIDO	45
6.1 INTRODUÇÃO	45
6.2 ESPECIFICAÇÃO	45
6.2.1 ATRIBUTOS E VALORES	46
6.2.2 CLÁUSULAS E PREDICADOS	47
6.2.3 REGRAS	47
6.2.4 FATOR DE CERTEZA	47
6.3 MODELAGEM ESSENCIAL.....	48
6.4 PLATAFORMA DE DESENVOLVIMENTO.....	49
6.5 PROCESSO DE KDD	49
6.5 OPERACIONALIDADE DO PROTÓTIPO.....	50
7 CONCLUSÕES E SUGESTÕES.....	58
7.1 CONCLUSÕES	58
7.2 LIMITAÇÕES.....	58
7.3 SUGESTÕES PARA TRABALHOS FUTUROS.....	59
APÊNDICE 1 – ESTRUTURA DAS TABELAS	60
APÊNDICE 2 – REGRAS DA BASE DE CONHECIMENTOS	62
APÊNDICE 3 – CÓDIGOS DA BASE DE CONHECIMENTOS.....	66
REFERÊNCIAS BIBLIOGRÁFICAS	68

LISTA DE FIGURAS

FIGURA 1 - COMPONENTES DE UM SISTEMA ESPECIALISTA.....	7
FIGURA 2 - ARQUITETURA DE UM SISTEMA ESPECIALISTA	12
FIGURA 3 - REGRA DE PRODUÇÃO.	18
FIGURA 4 – BARRA DE FERRAMENTAS NO DELPHI COM A BIBLIOTECA (EXPERT SINTA VCL).....	18
FIGURA 5 – RELACIONAMENTO ENTRE OS COMPONENTES DO EXPERT SINTA VCL.	22
FIGURA 6 - OS PASSOS DO PROCESSO DE KDD.	29
FIGURA 7 - MODELO RECEBE ENTRADAS E PRODUZ INFORMAÇÕES.	36
FIGURA 8 - TELA PARA ABERTURA DE CHAMADOS	42
FIGURA 9 - MODELO ENTIDADE RELACIONAMENTO	43
FIGURA 10 - DIAGRAMA DE CONTEXTO.....	48
FIGURA 11 - TELA DE ABERTURA	50
FIGURA 12 - TELA PARA GERAÇÃO DA BASE DE CONHECIMENTOS.....	51
FIGURA 13 - OPÇÃO PARA ABRIR A BASE DE CONHECIMENTOS DO SISTEMA ESPECIALISTA	51
FIGURA 14 - OPÇÃO PARA INICIAR O PROCESSO DE CONSULTA AO SISTEMA ESPECIALISTA.....	52
FIGURA 15 – TELA SOBRE DADOS DO CLIENTE.....	52
FIGURA 16 – TELA SOBRE SISTEMA QUE O CLIENTE ESTÁ ACESSANDO.	53
FIGURA 17 – TELA SOBRE MOTIVO DO CHAMADO DO CLIENTE.	53
FIGURA 18 – TELA SOBRE PROBLEMA DO CLIENTE	54
FIGURA 19 – TELA SOBRE DADOS ADICIONAIS DO PROBLEMA	54
FIGURA 20 – TELA DE COMPREENSÃO DOS RESULTADOS.....	55
FIGURA 21 – DEMONSTRAÇÃO DOS PASSOS DO SISTEMA ESPECIALISTA	56
FIGURA 22 – VARIÁVEIS DO SISTEMA E SEUS VALORES.....	56
FIGURA 23 – REGRAS DA BASE DE CONHECIMENTOS.....	57

LISTA DE TABELAS

TABELA 1 - COR_CLIENT.....	60
TABELA 2 - COR_SYSTEM.....	60
TABELA 3 - INF_CHAMDO_TECNO.....	60
TABELA 4 - INF_MOTIVO_CHAMDO	61
TABELA 5 - INF_OCORR_CHAMDO	61

RESUMO

A finalidade deste trabalho de conclusão de curso é desenvolver um protótipo de um sistema especialista que auxilie e facilite o trabalho dos profissionais nas chamadas de suporte dos clientes em uma empresa de desenvolvimento de software. As ferramentas utilizadas para a construção do sistema serão a *Shell “Expert SINTA”*, que utiliza um modelo de representação do conhecimento baseado em regras de produção e probabilidades, e a linguagem de programação *Delphi 3.0*.

ABSTRACT

The objective of this final paper at the end of the course is to develop the prototype of a dedicated system to help and to ease the work of the professionals when called to support and help the costumers of a software development firm.

The tools used to build the system will be the “Shell” “Expert Sinta” which uses a representation of know leage model based on production rules and probabilities. The programming language will be Delphi 3.0.

1 INTRODUÇÃO

Sistemas especialistas são programas de computador que procuram encontrar soluções para determinados problemas do mesmo modo que se espera que os especialistas humanos resolvam, se estiverem sob as mesmas condições ([LIA1999]). São sistemas com um conhecimento profundo sobre campos restritos do conhecimento. Os sistemas especialistas aplicam técnicas de Inteligência Artificial e conhecimento amplo em problemas específicos, para simular a atuação de especialistas humanos.

Para auxiliar na construção de sistemas especialistas foram criadas ferramentas de Inteligência Artificial orientadas para engenharia do conhecimento e construção de sistemas especialistas denominadas *shells*, que estão aptas a realizar muito do trabalho necessário para transpor um sistema especialista para o computador. Essas ferramentas permitem que o desenvolvedor do sistema preocupe-se somente com a representação do conhecimento do especialista, deixando para a *shell* a tarefa de interpretar o conhecimento representado e executá-lo em uma máquina, além de permitir explicações de como o computador chegou aquela(s) conclusão(ões). A principal função de uma *shell* é simplificar ao máximo o trabalho de implementação de um sistema especialista e permitir seu uso por qualquer pessoa sem conhecimentos de informática ([HEI1995]).

Os sistemas especialistas caracterizam-se pela manipulação de informações compostas por fatos a respeito do assunto e regras formais que descrevem relações. Estas informações compõem a chamada base de conhecimentos sobre a qual será feito o processamento. A fase de construção da base de conhecimentos é uma das mais complexas na implementação de um sistema especialista, pois o conhecimento de um especialista não encontra-se formalizado, precisando portanto de um trabalho prévio para tal ([FAY1996]).

Para a construção da base de conhecimentos do sistema será utilizado o processo de Descoberta do Conhecimento em Bases de Dados (KDD - *Knowledge Discovery in Databases*). O objetivo deste processo é reconhecer padrões nos dados, que serão posteriormente interpretados, a fim de obter informações que estão implícitas na base de dados, e não podem ser recuperadas através do uso de técnicas convencionais de consulta, como por exemplo a utilização de linguagens SQL. O processo de Descoberta do

Conhecimento em Bases de Dados é composto por várias etapas, que são: seleção, processamento, *data mining* e interpretação dos resultados ([AVI1998]). Na fase de seleção, devem ser escolhidos os dados relevantes, de acordo com critérios definidos, para a formação de uma base de dados onde a mineração de dados será realizada. No estágio de pré-processamento é realizada a limpeza dos dados, remoção de ruídos, ou seja, informações desnecessárias são descartadas. A mineração efetiva e a extração dos padrões de comportamento dos dados é realizada na fase de *Data Mining* ([GOU1998]). *Data Mining*, é o processo de descoberta de novas correlações, padrões e tendências entre as informações, através da análise dos dados armazenados. Seu objetivo é encontrar automaticamente padrões e regras com o propósito de transformar dados em informações úteis para a tomada de decisões e/ou avaliação dos resultados. No processo de *Data Mining*, várias etapas são realizadas e uma série de técnicas de diversas áreas de pesquisa, tais como reconhecimento de padrões, aprendizado de máquina e estatística são utilizadas ([AVI1998]).

A técnica de *Data Mining* que será utilizada no presente trabalho é a técnica de regras de produção. Num sistema baseado em regras de produção o conhecimento a ser processado é representado através do uso de regras com uma arquitetura previamente definida. Estas regras utilizam um par de condição-ação onde as condições são premissas e a ação é a conclusão. Este trabalho permitirá conhecer os elementos componentes destes sistemas, suas funções e características.

Propõe-se a especificação e implementação de um Sistema Especialista que auxiliará o suporte de informática mediante um chamado aberto, utilizando dados que já estão na base de dados do sistema de registro de chamados da empresa. No setor de atendimento são recebidas chamadas de suporte onde são registrados os problemas, o sistema/programa, o cliente que solicitou o suporte, qual a solução adotada para o problema, etc. As ferramentas utilizadas para a construção do sistema serão a *Shell "Expert SINTA"*, que utiliza um modelo de representação do conhecimento baseado em regras de produção e probabilidades, e a linguagem de programação *Delphi 3.0*.

O método de especificação utilizado no presente trabalho será expresso pela notação BNF (Backus-Naur Form).

1.1 OBJETIVOS DO TRABALHO

O trabalho proposto tem como objetivo a especificação e implementação de um protótipo de um sistema especialista que auxiliará e facilitará o trabalho dos profissionais nas chamadas de suporte dos clientes em uma empresa de desenvolvimento de software. Os objetivos específicos são:

- a) estudo sobre sistemas especialistas baseados em regras de produção e seus componentes;
- b) estudo da ferramenta *Expert SINTA Shell*;
- c) estudo das técnicas de *Data Mining* , em específico regras de produção;
- d) formalização do conhecimento através de regras de produção;
- e) implementação do protótipo;

1.2 ORGANIZAÇÃO DO TRABALHO

O trabalho está organizado em sete capítulos, descrevendo:

- Capítulo 1 - Introdução ao trabalho, com breve descrição do contexto deste, seus objetivos e sua organização.
- Capítulo 2 - Fundamentação teórica relativa a Sistemas Especialistas, abrangendo desde conceituação, ferramenta Expert Sinta Shell até formalização do conhecimento.
- Capítulo 3 - Fundamentação teórica sobre Banco de Dados, compreendendo conceituação, tipos, utilidades.
- Capítulo 4 – Fundamentação teórica sobre extração de conhecimento em bases de dados em específico Data Mining.
- Capítulo 5 - Consiste na descrição do funcionamento do Sistema de Suporte e Atendimento a Clientes;
- Capítulo 6 - Apresentação do protótipo desenvolvido;

- Capítulo 7 - Conclusões e sugestões para futuros trabalhos.

2 SISTEMAS ESPECIALISTAS

2.1 CONCEITOS

Um sistema especialista é um sistema computacional que resolve problemas de uma maneira bastante parecida com o especialista humano. São sistemas com um conhecimento específico e profundo sobre campos restritos do conhecimento ([RAB1995]). [LIA1999] descreve que: “sistemas especialistas são programas de computador que procuram atingir soluções de determinados problemas do mesmo modo que especialistas humanos se estiverem sob as mesmas condições”. Já para [GEN1986]: “sistemas especialistas são programas intensivamente baseados em conhecimento que resolvem problemas que geralmente requerem experiência humana. Eles executam muitas das funções secundárias que os especialistas executariam, como perguntar questões relevantes e explicar suas razões”. Como um especialista o sistema deve ser capaz de emitir decisões justificadas sobre um determinado assunto a partir de uma base de conhecimentos. Para tomar uma decisão o especialista busca em sua memória conhecimentos prévios, formula hipóteses, verifica os fatos que encontra e compara-os com as informações já conhecidas e então emite uma decisão. Neste processo o especialista realimenta a sua “base de conhecimentos” acerca do assunto. Os sistemas especialistas, portanto, caracterizam-se por armazenar um grande e profundo conhecimento sobre uma determinada área armazenando estas informações de modo organizado permitindo uma rápida e simplificada busca para respostas requeridas. Um sistema especialista deve além de inferir conclusões, ter a capacidade de aprender com novos conhecimentos e, desse modo, melhorar o seu desempenho e a qualidade de suas decisões, explicando como chegou a decisão final. Esta última é uma das principais características que distinguem os sistemas especialistas dos sistemas convencionais ([RIB1987]).

2.2 CARACTERÍSTICAS

Os sistemas especialistas caracterizam-se por um conhecimento profundo, no limite da perícia sobre uma área específica, organizado com o objetivo de simplificar a busca a respostas requeridas, por isso, o armazenamento da informação torna-se de suma importância.

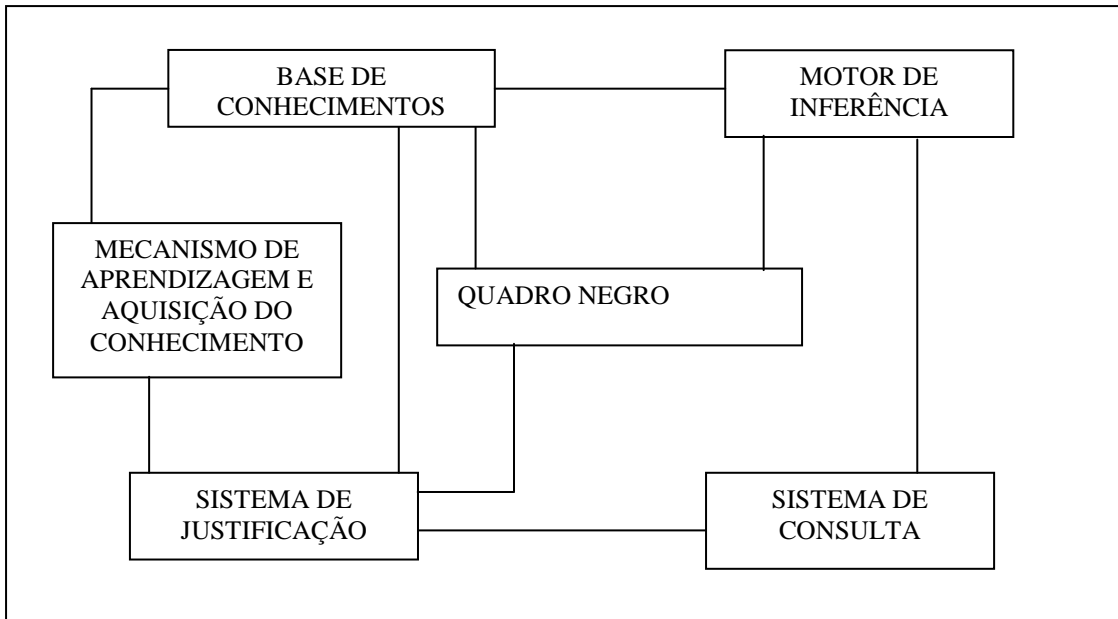
Outra característica dos sistemas especialistas é a utilização de técnicas de inferência para manipular informações visando uma solução. O mecanismo de inferência utiliza estratégias genéricas para adquirir conhecimento, processá-lo, tirar conclusões próprias e dar explicações acerca do processo de raciocínio. Essa abordagem baseada em conhecimento oferece a possibilidade de separar o conhecimento que descreve o domínio do problema do código de procedimentos que examina este conhecimento. Este mecanismo distingue os sistemas especialistas dos tradicionais ([HEI1995]).

Um sistema especialista pode chegar ou não a solução, e pode ainda chegar a uma conclusão distorcida. O sistema pode errar, porém o seu erro ou a não resposta advém de determinadas circunstâncias justificadas pelo próprio sistema ([RIB1987]).

2.3 COMPONENTES DE UM SISTEMA ESPECIALISTA

Segundo [RAB1995], “os componentes de um sistema especialista sofrem influências as mais variadas, desde a generalidade pretendida, os objetivos do mesmo, a representação do conhecimento e as ferramentas usadas na implementação”. O modelo geral do arquitetura de um sistema especialista apresentada por vários autores é mostrado na figura1. Especificamente, porém, a sua arquitetura depende da forma de representação do conhecimento e implementação adotadas.

Figura 1 - Componentes de um Sistema Especialista.



Fonte: [HEI1995]

2.3.1 BASE DE CONHECIMENTOS

A base de conhecimentos é o local onde se armazenam fatos e regras. Este conhecimento é passado ao sistema pelo especialista e armazenado de uma forma própria que permitirá ao sistema fazer posteriormente o processo de inferência. Um novo fato pode modificar todo o processo de inferência de acordo com as regras existentes sobre ele que estão sendo aplicadas e também sobre os novos fatos gerados pela avaliação dessas regras ([RIB1987]).

A qualidade do conhecimento armazenado é determinante no potencial do sistema especialista. A fase de construção da base de conhecimentos é uma das mais complexas na implementação do sistema especialista, pois o conhecimento de um especialista não se encontra formalizado, precisando de um trabalho prévio. A base de conhecimentos está interligada com quase todos os elementos do sistema, especialmente com a máquina de inferência, o mecanismo de aprendizagem e aquisição do conhecimento e o quadro negro ([HEI1995]).

2.3.2 MECANISMO DE APRENDIZAGEM E AQUISIÇÃO DO CONHECIMENTO

Segundo [RAB1995], “obter o conhecimento é a parte mais complexa da construção de um sistema especialista. A aquisição do conhecimento tende a caracterizar áreas de pesquisa específicas nas universidades e centros de pesquisas, geralmente ligadas a engenharia do conhecimento”. Conforme [GEN1986], “a aquisição de conhecimento é o processo de extração e formalização do conhecimento para uso em um sistema especialista”.

Para [LEV1988], “a fase de aquisição do conhecimento é a que apresenta maior dificuldade na construção de um sistema especialista. Esta dificuldade advém do fato de não existir uma linguagem comum de entendimento entre as partes envolvidas no projeto”. O especialista não tem suas idéias organizadas utilizando processos indutivos e dedutivos na obtenção das soluções. Desta forma cabe ao engenheiro de conhecimento tentar organizar esses elementos e obter as informações necessárias.

2.3.3 MOTOR OU MÁQUINA DE INFERÊNCIA

Para [RAB1995], a máquina de inferência não é normalmente um único módulo de programa. É , em geral, entendido como compreendendo o interpretador de regras e o escalonador das regras, quando o sistema especialista envolve regras de produção.

O processo de inferência está associado com a estrutura utilizada para o armazenamento do conhecimento na base de dados. De forma geral, pode-se afirmar que o processo envolve um encadeamento lógico que permite tirar conclusões a partir do conhecimento existente. Conforme [HEI1995], o motor de inferência é o responsável pela ação repetitiva de buscar, analisar e gerar novos conhecimentos.

[RIB1987] escreve que “o mecanismo de inferência depende de como se está representando o conhecimento. Nos sistemas de avaliação de regras, o mecanismo de inferência busca as regras na base de conhecimento e as avalia. Essa busca depende dos fatos e das hipóteses que existem e que se quer determinar a cada momento. Os objetivos a serem determinados pelo sistema de inferência devem ser relacionados com uma determinada ordem. A busca de regras é feita de maneira automática para que uma meta seja atingida.

Entretanto, existem casos em que a resposta pode ser obtida de maneira imediata e, nesses casos são estabelecidas estratégias de avaliação imediata, evitando todo o processo natural de busca e avaliação de regras. Outra estratégia usada consiste em o mecanismo de inferência proceder antes à busca das novas regras que foram causadas pela necessidade de se atender a uma meta, e avaliar essas regras a serem pesquisadas. Como os atributos são encontrados em diversas regras, o valor de uma cláusula já pode ter sido estabelecido. Esse valor, sozinho, permite determinar antecipadamente que a premissa da regra é falsa, e que não há razões para novas buscas. As estratégias de busca e avaliação de regras dependem do tipo de representação para o conhecimento e da arquitetura das próprias regras”.

2.3.4 SISTEMA DE CONSULTA

O usuário é geralmente, alguém que não participou da elaboração do sistema, sendo, portanto, natural que não conheça as estruturas do sistema e, que, provavelmente, não esteja familiarizado com as formas de representação do conhecimento adotadas. Para que os potenciais usuários possam acessar com proveito e sem maiores dificuldades o sistema especialista, é preciso muni-lo de recursos para consulta.

Conforme [HEI1995], “a maioria dos sistemas existentes usam técnicas simples de interação com o usuário, quase sempre utilizando perguntas já pré-formatadas e respostas tipo múltipla escolha. Outra técnica é a definição de uma sintética simples com um vocabulário restrito e limitado, própria para utilização do sistema”.

2.3.5 SISTEMA DE JUSTIFICAÇÃO

Para [HEIN1195], “o módulo de justificação é na verdade um recurso de questionamento fornecido ao usuário”. E portanto, o módulo de justificação é obrigatório nos sistemas especialistas, tendo, geralmente a capacidade de responder às seguintes perguntas:

Como chegou a esta conclusão?

Por que chegou a esta conclusão?

Por que não chegou a outra conclusão?

[RIB1987] descreve que “este módulo interage com o usuário esclarecendo-o de como o sistema chegou a determinada conclusão, ou por que está fazendo determinada pergunta. Utiliza diversos recursos e estruturas próprias para atender ao seu objetivo, mostrando que regra e que fatos foram usados da base de conhecimento, sempre que isso for solicitado por quem usa o sistema”.

2.3.6 QUADRO NEGRO

Segundo [RAB1995], “o quadro-negro é a área de trabalho do sistema especialista. Armazena informações, fatos e estruturas de suporte ao funcionamento do sistema, quando este efetua raciocínios. Embora todos os sistemas especialistas usem o quadro-negro, nem todos o têm como um componente explícito”.

2.4 REPRESENTAÇÃO DO CONHECIMENTO

Para [HEI1995], “para um sistema especialista resolver problemas é imprescindível que esteja associado a ele um razoável volume de conhecimentos relativos ao domínio do problema. Este conhecimento deve ser transformado em estruturas de dados organizadas de modo a permitir a sua utilização pelo computador”.

A representação do conhecimento é a formalização do conhecimento do sistema. Para que isto seja possível existem técnicas que permitem modelar o conhecimento de forma eficiente, sendo as principais descritas a seguir:

a) **lógica das preposições e dos predicados:** na lógica das preposições, será atribuído o valor lógico verdadeiro se as informações disponíveis permitirem tirar esta conclusão a respeito de uma preposição, caso contrário é atribuído o valor falso. Para se trabalhar com várias proposições utiliza-se operadores de conexão para assim obter as chamadas proposições compostas e aumentar a capacidade de expressão. Estes operadores são: AND, NOT, OR, IMPLIES, EQUIVALENT ([HEI1995]);

b) **regras de produção:** sua estrutura constitui-se basicamente de uma premissa, ou conjunto de premissas, e uma conclusão, ou conjunto de conclusões. As regras são armazenadas como uma coleção de declarações SE-ENTÃO (SE <premissas> ENTÃO

<conclusões). Onde a parte condicional consiste de uma expressão proposicional ou simplesmente um termo;

c) **redes semânticas**: foram inicialmente desenvolvidas para modelagem psicológica da memória humana, constituindo-se agora num método de representação padrão. São estruturas formadas por nós, conectados entre si através de arcos rotulados. Os nós representam objetos, conceitos, situações ou ações, e os arcos representam relações entre os nós ([LAP1993]);

d) **quadros ou frames**: esta forma de representação do conhecimento, organiza conhecimento de maneira a tornar evidente a compreensão de como a inferência pode ser feita ([LOP1997]). Um *frame* é constituído por um nome, uma coleção de atributos, chamados de escaninhos ou *slots*, e valores associados a eles.

2.5 A SHELL - EXPERT SINTA

O *Expert SINTA* é uma ferramenta computacional que utiliza técnicas de inteligência artificial para geração automática de sistemas especialistas. Esta ferramenta utiliza um modelo de representação do conhecimento baseado em regras de produção e probabilidades, tendo como objetivo principal simplificar o trabalho de implementação de sistemas especialistas através do uso de uma máquina de inferência compartilhada, do tratamento probabilístico das regras de produção e da utilização de explicações sensíveis ao contexto da base de conhecimento modelada.

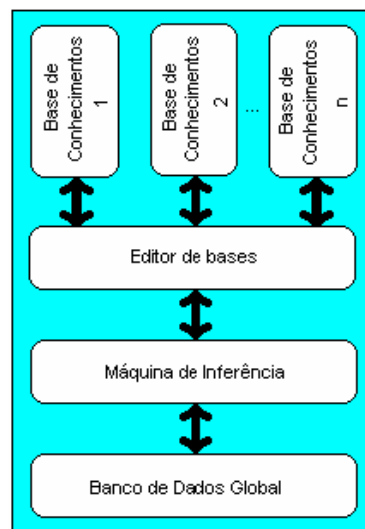
As principais características do *Expert SINTA*, conforme [LIA1999] são:

- a) utilização do encadeamento para trás (*backward chaining*);
- b) utilização de fatores de confiança;
- c) ferramentas de depuração;
- d) possibilidade de incluir ajudas on-line para cada base.

2.5.1 ARQUITETURA DE UM SISTEMA ESPECIALISTA NO EXPERT SINTA SHELL

Os sistemas especialistas que utilizam o *Expert SINTA* possuem a seguinte arquitetura, conforme mostra a figura 2:

Figura 2 - Arquitetura de um Sistema Especialista .



FONTE: [LIA1999]

- a) **base de conhecimentos** representa a informação (fatos e regras) que um especialista utiliza;
- b) **editor de bases** é o meio pelo qual a *shell* permite a implementação das bases desejadas;
- c) **máquina de inferência** o motor de inferência é o responsável pela ação repetitiva de buscar, analisar e gerar novos conhecimentos;
- d) **banco de dados global** são as evidências apontadas pelo usuário do Sistema Especialista durante uma consulta.

O objetivo do *Expert SINTA* é simplificar ao máximo as etapas de criação de um Sistema Especialista completo. Para tanto, já oferece uma máquina de inferência básica, fundamentada no encadeamento para trás (*backward chaining*) ([LIA1999]).

2.5.2 UTILIZANDO REGRAS DE PRODUÇÃO NO EXPERT SINTA SHELL

A representação do conhecimento por regras de produção é baseada nas propostas do matemático Emil Post (1943) que via nos sistemas de produção um modelo computacional geral de solução de problemas. A representação do conhecimento por regras de produção é a forma mais utilizada em sistemas especialistas. A justificativa é a naturalidade que representa para o homem pois, o par condição-ação para raciocinar e decidir, também é usado pela mente humana ([HEI1995]).

Para [LIA1999], as regras de produção possuem as seguintes vantagens:

- Modularidade: cada regra, por si mesma, pode ser considerada uma peça de conhecimento independente;
- Facilidade de edição: novas regras podem ser acrescentadas e antigas ser modificadas com relativa independência;
- Transparência do sistema: garante maior legibilidade da base de conhecimentos.

Por exemplo:

SE tem combustível no tanque	(Premissas)
E tem combustível no carburador	(Premissas)
ENTÃO o motor recebe combustível	(Conclusões da regra)

Para o projetista que cria bases utilizando o *Expert SINTA*, o seguinte critério para definições deve ser seguido:

A estrutura das premissas devem obedecer o seguinte modelo:

<CONECTIVO>	<ATRIBUTO>	<OPERADOR>	<VALOR>
-------------	------------	------------	---------

- Conectivo: **NÃO**, **E**, **OU**, sua função é unir a sentença ao conjunto de premissas que formam a seção de antecedentes de uma regra.

- Atributo: é uma variável capaz de assumir uma ou múltiplas instâncias no decorrer da consulta à base de conhecimento.
- Operador: é a ligação entre o atributo e o valor da premissa que define qual a comparação a ser realizada. São operadores tais como: =, >, <=, >=, <> entre outros.
- Valor: é um item de uma lista a qual foi previamente criada e relacionada a um atributo.

A estrutura da conclusão deve obedecer o seguinte modelo:

<ATRIBUTO>	=	<VALOR>	<GRAU DE CONFIANÇA>
------------	---	---------	---------------------

- Atributo: equivale ao atributo das premissas;
- “=” é um operador de **atribuição e não de igualdade**;
- Valor: equivale ao valor utilizado nas premissas;
- Grau de confiança: é um percentual indicando qual a confiabilidade da conclusão da regra. Varia de 0% à 100%.

2.5.3 O MÉTODO DE EXTRAÇÃO DE CONHECIMENTO DO EXPERT SINTA SHELL

O projetista deve incluir na definição da base quais os atributos que devem ser encontrados. A máquina de inferência encarrega-se de encontrar uma atribuição para o atributo desejado nas conclusões da regras. Para que a regra seja aprovada, suas premissas devem ser satisfeitas, obrigando à máquina a encontrar os atributos das premissas para que possam ser julgadas, acionando um encadeamento recursivo. Caso o atributo procurado não seja encontrado em nenhuma conclusão de regra, uma pergunta direta é feita ao usuário ([LIA1999]).

2.5.4 VARIÁVEIS UNIVALORADAS E VARIÁVEIS MULTIVALORADAS

Quando a máquina de inferência está atrás de encontrar instanciações para uma variável univalorada, ela irá procurar até encontrar um valor ou então esgotar todas as possibilidades da base de conhecimento. Se durante a busca de outra variável, uma variável univalorada receber um valor quando já possuía outro, esse valor antigo será descartado, vigorando o novo valor. Uma única variável pode receber vários valores em uma única consulta ao sistema ([LIA1999]).

A busca de valores para variáveis multivaloradas prossegue até que toda a base de conhecimento seja explorada. Os valores permanecem acumulados. Nesta hora é preciso ter cuidado com contradições presentes na base ([LIA1999]).

As variáveis numéricas são tratadas como univaloradas.

2.5.5 FATORES DE CONFIANÇA

Sabe-se que o conhecimento humano não é determinístico. Não há especialista que esteja sempre em condições de afirmar determinada conclusão com certeza absoluta. Graus de confiança são geralmente atribuídos às respostas, principalmente quando existe mais de uma, sendo este um dos pontos mais críticos na elaboração de uma representação computacional do saber humano.

Observam-se dificuldades para representar a confiabilidade das informações:

- a) especialistas não se sentem confortáveis em pensar em termos de probabilidades. Suas estimativas não precisam corresponder àquelas definidas matematicamente;
- b) tratamentos rigorosamente matemáticos de probabilidade utilizam informações nem sempre disponíveis ou simplificações que não são claramente justificáveis em aplicações práticas.

2.5.6 O CÁLCULO DE PROBABILIDADES NO EXPERT SINTA

A atribuição dos fatores de confiança na presente versão do Expert Sinta é a seguinte [LIA1999]:

- a) **quando deseja-se saber qual o valor final atribuído às variáveis na conclusão de uma regra:** deve-se seguir os seguintes passos: seja c_1 o grau de confiança atribuído ao resultado final da premissa de uma regra r . Na conclusão de r , deve-se ter expressões como $\text{var} = \text{value CNF } c_2$, onde var é uma variável, value é um termo qualquer que pode ser atribuído a uma variável, c_2 é um real pertencente entre zero e cem $[0,100]$ que representa o grau de confiança da atribuição. Mas, c_2 é apenas uma referência, pois o valor final é dependente do resultado da premissa. Assim sendo, será realizado a operação $\text{var} = \text{value CNF } c_1 * c_2$. Exemplo de uma aplicação para o cálculo de probabilidades: SE fumagina = sim ENTÃO suspeita de praga = mosca branca, grau de confiança (CNF) 70%. Assim sendo, supondo que o grau de confiança da igualdade fumagina = sim é 80%, teremos que à variável suspeita de praga será atribuído o valor mosca branca, com o respectivo grau de confiança $0.80 * 0.70 = 0.56 = 56\%$;
- b) **cálculo do grau de confiança com o operador E:** se possuí-se duas igualdades $\text{var}_1 = \text{value}_1$ e $\text{var}_2 = \text{value}_2$, com os respectivos graus de confiança c_1 e c_2 , têm-se que a sentença $\text{var}_1 = \text{value}_1$ E $\text{var}_2 = \text{value}_2$ retornará como valor de confiança $c_1 * c_2$. Exemplo de aplicação: SE estados das folhas = esfrelam facilmente E presença de manchas irregulares = sim...Se o grau de confiança da igualdade estados das folhas = esfrelam facilmente é 80% e o grau de confiança da igualdade presença de manchas irregulares = sim é 70%, temos que a conjunção das duas sentenças retornará um valor CNF de 56%, pois esse é o produto dos dois valores;
- c) **cálculo do grau de confiança com o operador OU:** se possuí-se duas igualdades $\text{var}_1 = \text{value}_1$ e $\text{var}_2 = \text{value}_2$, com os respectivos graus de confiança c_1 e c_2 , têm-se que a sentença $\text{var}_1 = \text{value}_1$ OU $\text{var}_2 = \text{value}_2$ retornará como valor de confiança $c_1 + c_2 - c_1 * c_2$. Exemplo de aplicação: SE besouros vermelhos = sim OU larvas marrons = sim ...Se o grau de confiança da igualdade besouros vermelhos = sim é 80% e o grau de confiança da igualdade larvas marrons = sim é

70%, temos que a disjunção das duas sentenças retornará um valor CNF de $0.70 + 0.80 - 0.70 * 0.80 = 1.50 - 0.56 = 0.94 = 94\%$;

- d) **quando uma variável recebe duas vezes o mesmo valor em pontos diferentes da consulta:** em momentos diferentes de uma consulta, uma mesma variável var pode receber o mesmo valor v, sendo que até à penúltima instanciación esta variável possuía grau de confiança c1, e a última atribuiu um CNF c2. Sendo assim, têm-se que o valor final de confiança para var = v será dado através da fórmula $ca + cn - ca * cn$, onde o ca representa o grau de confiança antes da última mudança e o cn representa o último grau de confiança atribuído. Exemplo de aplicação: a variável doença possuía valor mofo preto com grau de confiança 60%. Após a aplicação de outras regras chegou-se a uma outra atribuição doença = mofo preto, desta vez com CNF 50%;
- e) **O cálculo se dá de maneira semelhante à aplicação da regra OU:** doença terá como um dos valores mofo preto, com respectivo grau de confiança $0.60 + 0.50 - 0.60 * 0.50 = 1.10 - 0.30 = 0.80 = 80\%$. Obs.: O sistema admite 50% como valor mínimo de confiança para que uma igualdade seja considerada verdadeira, mas esse valor pode ser modificado. O intervalo de grau de confiança varia de 0 a 100.

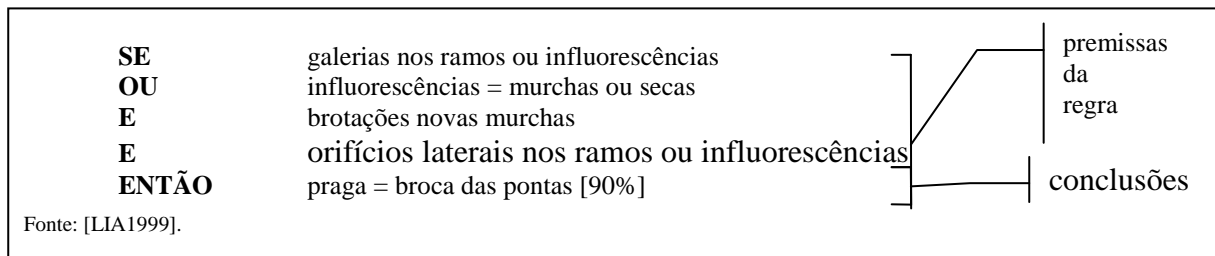
2.5.7 GERENCIANDO BASES DE DADOS

Como padrão, o *Expert SINTA* grava as bases de conhecimento geradas em arquivos com extensões *.BCM. Ao salvar a base pela primeira vez, uma caixa de diálogo aparecerá para que seja indicado o nome do arquivo na qual a base será gravada e qual pasta será encontrado este arquivo. Além disso, é possível também mudar a posição da barra de ferramentas, bastando apenas clicar sobre esta barra e arrastar o mouse para onde desejar movê-la ([LIA1999]).

2.6 FORMALIZAÇÃO DO CONHECIMENTO

Conforme [LIA1999], o *Expert SINTA* utiliza regras de produção para a representação do conhecimento. Essas regras são um conjunto de condições no estilo SE...ENTÃO..., com a possibilidade de inclusão de conectivos lógicos. Relacionando, deste modo, os atributos no escopo do conhecimento e o uso de probabilidades, como pode-se observar na figura 3.

Figura 3 - Regra de produção.

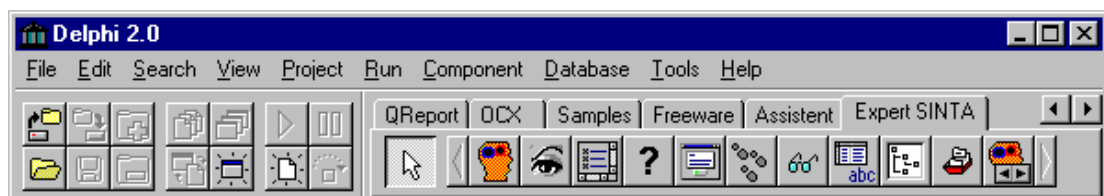


2.7 EXPERT SINTA VISUAL COMPONENT LIBRARY (VCL)

De acordo com [LIA1999], a *Expert SINTA VCL* é uma biblioteca de componentes para programação de Sistemas Especialistas. Na figura 4 temos a barra de ferramentas da *Expert SINTA VCL* para Delphi. De uma forma geral, esta biblioteca de componentes torna possível a criação de aplicações para bases de conhecimento geradas com o *Expert SINTA*. Esta biblioteca de componentes vem sendo utilizada na construção de Sistemas Especialistas. Os principais motivos da utilização do *Expert SINTA VCL* são:

- o *Expert SINTA (shell)* não provém toda a funcionalidade necessária a certos Sistemas Especialistas;
- não há meios de aproveitar os dados obtidos com o *shell* em outros programas;
- é inviável o acréscimo de vários recursos de interface e intercâmbio de dados na ferramenta em si.
- os Sistemas Especialistas devem ser compilados em uma dada linguagem de programação e utilizados de forma totalmente independente do *Expert SINTA*;
- é possível reaproveitar milhares de linhas de código já escritas na construção do *shell*.

Figura 4 – Barra de ferramentas no Delphi com a biblioteca (Expert SINTA VCL).



De uma forma geral, a *Expert SINTA VCL* torna possível a criação de *front-ends* para bases de conhecimento geradas com o *Expert SINTA*. Entre as tarefas desempenhadas por esta VCL, há:

- a) encapsulamento da máquina de inferência e a estrutura de dados que representa o conhecimento (regras de produção);
- b) fornecimento de mecanismos para entrada de dados do usuário;
- c) fornecimento mecanismos de depuração;
- d) permitir a personalização da aplicação final.

2.7.1 COMPONENTES DO VCL

Os componentes nativos da *Expert SINTA VCL* podem ser divididos em categorias, conforme descrito por [LIA1999]. A seguir, são relacionadas estas categorias dos componentes:



TExpertSystem - encapsula a máquina de inferência e a estrutura de dados que representa a base de conhecimento.



TruleView - exibe regras da base de conhecimento referenciada pelo componente *TExpertSystem*.



TExpertPrompt - menu para entrada de dados do usuário em resposta a uma determinada pergunta efetuada pelo sistema.



TlabelQuestion - a única opção de personalização de interface integrada no *shell* Expert SINTA é a possibilidade de mudança da mensagem que aparece em uma pergunta para cada variável.



TvaluesGrid - exibe as instâncias (valores) de uma dada variável por ordem decrescente de grau de confiança.



TwhyDialog - caixa de diálogo que exibe uma explicação para a necessidade de uma dada pergunta.



TdebugPanel – semelhante a *TRuleView*, exibe as regras da base de conhecimento de um sistema especialista em um painel, mas indica também qual premissa (ou conclusão) está sendo analisada pela máquina de inferência em determinado ponto de uma consulta.



TwatchPanel – de forma semelhante a opção *Watch* de um ambiente de programação, exibe as instâncias (valores atribuídos durante uma consulta) de todas variáveis através de dois painéis: o superior lista todas as variáveis; o inferior, as instâncias da variável selecionada no painel superior.



TconsultTree - este componente pode criar e exibir de forma hierárquica todos passos seguidos do começo ao fim de uma consulta.



TallVars - ao contrário de *TWatchPanel*, este componente são se atualiza automaticamente para cada nova instância criada pela máquina de inferência.



TexNavigator - um navegador que controla o fluxo da consulta em conjunto com as respostas entradas pelo usuário e outros componentes de interface acrescentados pelo desenvolvedor da aplicação.

2.7.2 RELAÇÃO ENTRE OS COMPONENTES

De acordo com [LIA1999], existe outra forma de classificar a VCL: componentes de atualização automática, os quais modificam-se automaticamente sempre que um fato relevante ocorre durante uma consulta e componentes passivos, que precisam da chamada de um método para exibir funcionalidade.

Basicamente, todos os componentes, à exceção de *TConsultTree* e *TAllVars*, são automáticos. Para que componentes automáticos procedam como tal, é preciso relacioná-los a um componente *TExpertSystem*. Para isso, existe a propriedade *ExpertSystem*. Através do *Object Inspector* de um ambiente visual como o Delphi, pode ser atribuído um sistema especialista para cada controle automático. Cada componente reage de acordo com a mudança

feita. Um exemplo é o *TruleVie*, que reage a mudanças do arquivo da base de conhecimento, mas nenhum componente nativo da *Expert SINTA VCL* é notificado sobre mudanças realizadas diretamente na estrutura de dados, como, por exemplo, uma alteração de nome de variável feita através do *shell*.

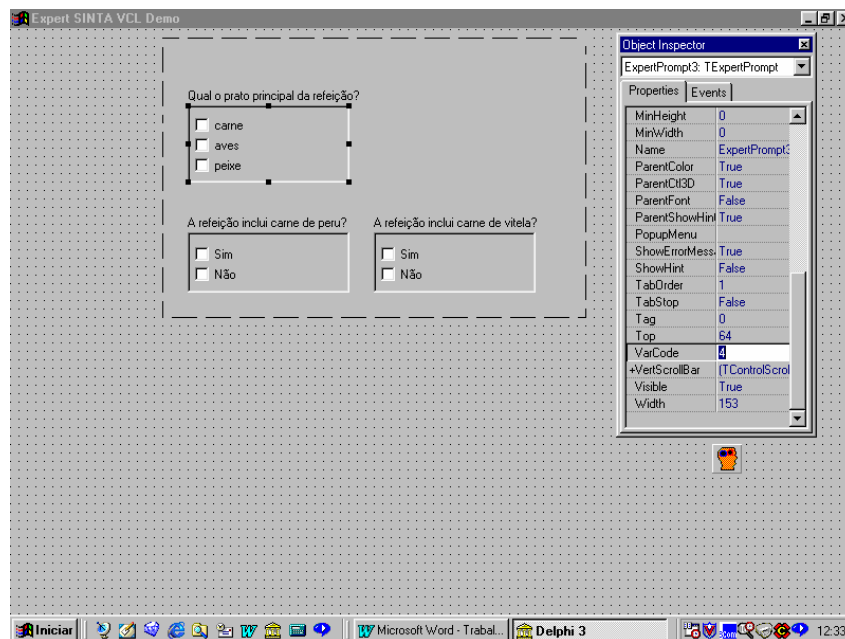
Outro parâmetro que aparece constantemente nos componentes da *Expert SINTA VCL* são os códigos de variáveis. Por exemplo, o componente *TExpertPrompt* monta automaticamente um menu de entrada de dados para que o usuário marque valores de uma dada variável. Logo, a variável é um parâmetro básico deste componente. É indicada a variável referenciada pelo componente através de seu código.

Cada variável criada através do *Expert SINTA* recebe um código interno que nunca muda (a não ser, óbvio, que esta seja apagada e inserida novamente - categoricamente nem sequer seria a mesma variável), conforme mostrado no apêndice 2. Assim, o uso de códigos é o modo mais estável de referência de variáveis.

Pode-se observar, na figura 5, o relacionamento de um dos três quadros que foi selecionado (*TExpertPrompt*) com o campo *VarCode* do *Object Inspector* do ambiente Delphi. Ao colocar o número do código da variável gerada pelo arquivo texto da ferramenta *Expert SINTA*, automaticamente são colocados os seus valores. Observa-se também o uso das perguntas realizadas através do *TlabelQuestion*, sendo que todos os componentes (*TExpertPrompt* e *TlabelQuestion*) devem estar relacionados com o componente *TExpertSystem*.

Para obter os códigos criados pelo *Expert SINTA*, deve-se abrir a base de conhecimento no *shell* e selecionar o menu Arquivo/Exportar/Códigos. A seguir, digita-se o nome do arquivo texto (.txt) onde sairão os resultados. Pode-se abrir posteriormente este arquivo em um editor de textos quando precisar saber o código de uma dada variável ou valor ([LIA1999]).

Figura 5 – Relacionamento entre os componentes do Expert SINTA VCL.



Fonte: adaptado de [LIA1999].

3 BANCO DE DADOS

3.1 CONCEITOS BÁSICOS

Para [CER1995], uma definição bastante difundida entre vários autores é: “um banco de dados é uma coleção de dados organizados e integrados, armazenados em forma de tabelas interligadas através de chaves primárias e estrangeiras, que constituem uma representação natural dos dados, sem imposição de restrições ou modificações, de forma a ser adequada a qualquer computador, podendo ser utilizada por todas as aplicações relevantes sem duplicação de dados, e sem a necessidade de serem definidos em programas, pois utiliza as definições existentes nas bases de dados, através do Dicionário de Dados ativo dinâmico”.

Segundo [DATE1991], banco de dados pode ser considerado uma sala de arquivos eletrônica, ou seja, um depósito de um conjunto de arquivos de dados computadorizados que oferece diversos recursos ao usuário, possibilitando a realização de diversas operações, como:

- a) adição de novos arquivos ao banco de dados;
- b) inserção de novos dados;
- c) recuperação de dados;
- d) atualização de dados;
- e) eliminação de dados;
- f) renovação de arquivos existentes.

3.2 BANCO DE DADOS RELACIONAL

A abordagem relacional dos dados é baseada na observação de que arquivos podem ser considerados como relações matemáticas. Conseqüentemente, a teoria elementar de relações pode ser usada para lidar com vários problemas práticos que surgem com os dados desses arquivos. Por isso, na maior parte da literatura relacional, tabelas são tratadas como relações.

As linhas dessas tabelas são usualmente conhecidas como tuplas, que possuem na literatura uma definição mais precisa do que linha ou registro.

O objetivo principal do sistema relacional é oferecer um banco de dados de fácil compreensão e bem definido, o que facilita sua manipulação. Seu princípio teórico é a identificação precisa do que pode ou não ser feito. Os operadores são de alto nível e de fácil aprendizado e entendimento. Quando se deseja efetuar mudanças nas definições dos bancos de dados, podem-se inserir novas colunas às tabelas, sem que se tenha que fazer qualquer alteração nos programas de aplicação.

3.3 CARACTERÍSTICAS DE UM BANCO DE DADOS RELACIONAL

As principais características da abordagem relacional são:

a) Estrutura Tabular : os dados são representados na forma de tabelas, nas quais temos linhas ou tuplas e atributos ou domínios;

b) Álgebra Relacional : a manipulação dessas tabelas é feita através de operadores que permitem qualquer tipo de acesso a uma tabela ou a um conjunto delas (seleção, junção, união etc.);

c) Dicionário de Dados : o dicionário de dados obrigatoriamente deve ser integrado, ativo e dinâmico, para dar consistência automática de campos, referência cruzada dos dados, suporte a análise estruturada ou orientada a objetos e ao Diagrama entidade/relacionamento (DER), integridade e segurança de dados.

3.4 BANCO DE DADOS DISTRIBUÍDO

Um sistema de banco de dados distribuído consiste numa coleção de sítios, conectados através de uma sistema de rede de comunicações, no qual cada sítio é um sistema de banco de dados em seu próprio direito, mas os sítios “concordaram” em cooperar, de forma que um

usuário em qualquer sítio pode ter acesso a qualquer dado da rede, exatamente como se o dado estivesse armazenado no próprio sítio do usuário.

Um sistema de banco de dados distribuído é, portanto, um sistema de um banco de dados virtual, cujos componentes estão armazenados fisicamente em um número distinto de bancos de dados “reais” num número distinto de sítios. É importante salientar que cada sítio possui seu próprio banco de dados, dentro de um ambiente descentralizado, Isto é, cada sítio mantém seu próprio banco de dados sobre o qual atua. O Banco de Dados Distribuído pode, portanto, ser classificado como um tipo de sociedade entre os bancos de dados e os nós locais.

3.4.1 CARACTERÍSTICAS DE UM BANCO DE DADOS DISTRIBUÍDO

Para que um sistema caracterize-se como distribuído deverá obedecer as seguintes regras:

a) Autonomia local: cada banco de dados físico particular pode ser administrado independentemente da rede e dos demais bancos de dados participantes do sistema distribuído. Todos os assuntos de segurança, integridade e representação de armazenagem permanecem sob o controle local.

b) Sem a dependência de um Centro de Processamento de Dados (CPD): a autonomia local significa que todos os sítios deverão ser tratados de forma idêntica; não deverá haver nenhuma dependência num CPD mestre tais como processamento de pesquisas ou gerenciamento de transações centralizadas.

c) Operação contínua: num sistema distribuído idealmente não deveria existir a necessidade de planejamento para o fechamento de sistema. No entanto existem algumas situações em que algum tipo de fechamento e conseqüente interrupção de serviços é inevitável num ambiente distribuído. O objetivo é manter a um mínimo tais interrupções.

d) Independência de local: os usuário não devem saber onde os dados se encontram armazenados, muito pelo contrário, devem se capazes de executar tarefas lógicas como se os dados estivessem todos em seu próprio CPD.

e) Independência de fragmentação: um sistema distribuído suporta fragmentação de dados se uma determinada relação puder ser dividida em pedaços ou fragmentos para fins de armazenagem física. A fragmentação é desejável por questões de desempenho. Os dados podem ser armazenados no local onde forem mais freqüentemente usados, de forma que a maioria das operações seja puramente local.

f) Independência de replicação: um sistema distribuído suporta dados replicados se uma determinada relação puder ser representada no nível físico por muitas cópias distintas armazenadas ou réplicas em vários locais distintos. A independência de replicação é desejável porque simplifica a programação do usuário e as atividades terminais. A independência de replicação permite que réplicas sejam criadas e destruídas dinamicamente e qualquer momento para atender a requisitos de mudanças, sem invalidar nenhum programa ou atividade do usuário.

g) Processamento de pesquisas distribuídas: a execução de pesquisas num sistema distribuído é, em si mesmo um processo distribuído, isto é, envolve tanto UCP e atividades como E/S locais em vários locais distintos em geral, junto com alguma comunicação de dados entre esses locais. Esse segundo componente é um fator importante de desempenho que tenderá a dominar todos os demais numa rede de comunicação de dados lenta e extensivamente utilizada. O principal objetivo do otimizador será minimizar esse fator. É importante que o otimizador faça um bom trabalho, desempenhando suas funções sob uma perspectiva global. O processo de otimização deve ser distribuído, isto é, deve envolver uma etapa de otimização geral para minimizar o tráfego, mais etapas de otimização local separadas em cada sítio.

h) Gerenciamento de Transações distribuídas: num sistema distribuído, uma simples transação pode envolver a execução de código em locais múltiplos e, portanto, envolver atualizações em locais múltiplos. Para se assegurar de que uma determinada transação permaneça atômica no ambiente distribuído, o sistema deverá assegurar que um grupo de locais envolvidos efetive ou desfaga em uníssono as atualizações.

i) Independência de máquina (hardware): as instalações de computadores envolvem computadores de diferentes fabricantes, existe uma necessidade premente de itegração dos dados de vários computadores em todos esses sistemas e de apresentar ao usuário a imagem

de um sistema único. Portanto é bastante desejável executar o mesmo SGBD em diferentes computadores, para que os sistemas diferentes possam participar como sócios em igualdade de condições num sistema distribuído.

j) Independência de sistema operacional: é obviamente desejável ter a capacidade de executar o SGBD em diferentes sistemas operacionais.

k) Independência de redes de comunicação: de forma ideal o sistema distribuído deve suportar redes locais, redes de longa distância, satélites, pacotes tipo Renpac e linhas telefônicas, com protocolos variados.

l) Independência de SGBD: é necessário que os SGBDs em cada diferente sítio suportem as mesmas interfaces, embora não seja necessário que todos os sítios possuam cópias do mesmo sistema. O sistema ideal distribuído deve fornecer independência de SGBD.

4 DATA MINING

4.1 DEFINIÇÃO

Avanços no armazenamento de informações, tais como, velocidade, capacidade de acesso e baixo custo, impulsionaram a geração de grandes quantidades de dados. Essas grandes bases de dados tornam quase impossível a tarefa de análise de seus dados de forma manual através de métodos tradicionais, ou seja, torna-se evidente a necessidade de automatização dos processos de análise de dados. Entretanto, a análise desses dados ainda é demorada, dispendiosa, pouco automatizada, e sujeita a erros, mal-entendidos e falta de acurácia. A automatização dos processos de análise de dados, com a utilização de softwares ligados diretamente à massa de informações, se tornou uma necessidade, já que o aproveitamento das informações já existentes, transformando-as em conhecimento, permite avanços sem paralelo na história do desenvolvimento dos bancos de dados ([FIG1998]).

Neste capítulo será apresentado o *Data Mining*, que é a área de pesquisa em Inteligência Artificial que busca encontrar padrões em bases de dados ([AVI998]). Serão descritas as etapas do Processo de *Knowledge Discovery in Databases* (KDD) e as tarefas que o *Data Mining* pode desempenhar.

4.2 PROSPECÇÃO DO CONHECIMENTO

Prospecção do conhecimento em bases de dados (*Knowledge Discovery in Databases – KDD*) é um processo que envolve a automação da identificação e do reconhecimento de padrões em um banco de dados. Sua principal característica é a extração não-trivial de informações a partir de uma base de dados de grande porte. Essas informações são necessariamente implícitas, previamente desconhecidas, e potencialmente úteis([DAL2000]).

Devido a essas características incomuns, todo o processo de KDD depende de uma nova geração de ferramentas e técnicas de análise de dados, e envolve diversas etapas. A

principal, que forma o núcleo do processo, e que muitas vezes se confunde com ele, chama-se *Data Mining*, ou Mineração de Dados, também conhecido como processamento de padrões de dados, arqueologia de dados, ou colheita de informação (*information harvesting*).

O KDD compreende todo o processo de descoberta de dados, enquanto o *Data Mining* refere-se a aplicação de algoritmos para extração de padrões de dados, sem os passos adicionais do KDD e da análise dos resultados ([AVI1998]).

4.3 PROCESSO DE KDD

O processo de KDD envolve numerosos passos com muitas decisões a serem tomadas sendo que cada etapa é fundamental para que os objetivos estabelecidos sejam alcançados. O processo de KDD (figura 6) começa com o entendimento do domínio da aplicação e dos objetivos finais a serem atingidos. A seguir é feito um agrupamento organizado de uma massa de dados, alvo da prospecção. A etapa da limpeza dos dados (*data cleaning*) vem a seguir, através de um pré-processamento dos dados, visando adequá-los aos algoritmos. Isso se faz através da integração de dados heterogêneos, eliminação de incompletude dos dados, repetição de registros, problemas de tipagem, etc. Essa etapa pode tomar até 80% do tempo necessário para todo o processo, devido às dificuldades de integração de bases de dados heterogêneas ([FAY1996]).

Figura 6 - Os passos do processo de KDD.



Fonte: [FIG1998]

Os dados pré-processados devem ainda passar por uma transformação que os armazena adequadamente, visando facilitar o uso das técnicas de *Data Mining*. Em algumas aplicações

de *Data Mining* mais específicas, ferramentas avançadas de representação de conhecimento podem descrever o conteúdo de um banco de dados por si só, usando esse mapeamento como uma meta-camada para os dados.

Prosseguindo no processo, chega-se à fase de *Data Mining* especificamente, que começa com a escolha dos algoritmos a serem aplicados. Essa escolha depende fundamentalmente do objetivo do processo de KDD: classificação, segmentação, agrupamento por afinidades, estimativas, etc. De modo geral, na fase de *Data Mining*, ferramentas especializadas procuram padrões nos dados. Essa busca pode ser efetuada automaticamente pelo sistema ou interativamente com um analista, responsável pela geração de hipóteses.

Diversas ferramentas distintas, como redes neurais, indução de árvores de decisão, sistemas baseados em regras e programas estatísticos, tanto isoladamente quanto em combinação, podem ser então aplicadas ao problema. Em geral, o processo de busca é interativo, de forma que os analistas revêem o resultado, formam um novo conjunto de questões para refinar a busca em um dado aspecto das descobertas, e realimentam o sistema com novos parâmetros.

Ao final do processo, o sistema de *Data Mining* gera um relatório das descobertas, que passa então a ser interpretado pelos analistas de mineração. Somente após a interpretação das informações obtidas encontra-se o conhecimento.

Uma diferença significativa entre *Data Mining* e outras ferramentas de análise está na maneira como exploram as interrelações entre os dados. As diversas ferramentas de análise disponíveis dispõem de um método baseado na verificação, isto é, o usuário constrói hipóteses sobre interrelações específicas e então verifica ou refuta, através do sistema. Esse modelo torna-se dependente da intuição e habilidade do analista em propor hipóteses interessantes, em manipular a complexidade do espaço de atributos, e em refinar a análise baseado nos resultados de consultas ao banco de dados potencialmente complexas. Já o processo de *Data Mining* fica responsável pela geração de hipóteses, garantindo mais rapidez, acurácia e completude aos resultados ([WAN1999]).

Estas etapas são interdependentes, pois os resultados de cada uma são a entrada da próxima etapa. Toda a abordagem é dirigida por resultados e cada estágio depende dos resultados do estágio anterior ([FIG1998]). Não existe uma ordem ou seqüência totalmente única para o andamento deste processo, porque isso depende das técnicas empregadas e dos dados sobre os quais o KDD está sendo aplicado ([AVI1998]). A qualquer momento, por exemplo, pode-se voltar o processo de KDD para uma etapa anterior, desde que a técnica e os dados empregados permitam.

4.4 TAREFAS DO DATA MINING

O *Data Mining* pode desempenhar uma série limitada de tarefas dependendo das circunstâncias. Cada classe de aplicação em *Data Mining* tem como base um conjunto de algoritmos que serão usados na extração de relações relevantes dentro de uma massa de dados ([FIG1998]):

- a) classificação;
- b) estimativa;
- c) agrupamento por afinidade;
- d) previsão;
- e) segmentação.

Cada uma destas propostas difere quanto à classe de problemas que o algoritmo será capaz de resolver.

4.4.1 CLASSIFICAÇÃO

Classificação é uma técnica que consiste na aplicação de um conjunto de exemplos pré-classificados para desenvolver um modelo capaz de classificar uma população maior de registros. Em geral, algoritmos de classificação incluem árvores de decisão ou redes neurais, e começam com um treinamento a partir de transações-exemplo. O algoritmo classificador usa

estes exemplos para determinar um conjunto de parâmetros, codificados em um modelo, que será mais tarde utilizado para a discriminação do restante dos dados ([FAY1996]).

Uma vez que o algoritmo classificador foi desenvolvido de forma eficiente, ele será usado de forma preditiva para classificar novos registros naquelas mesmas classes pré-definidas.

Alguns exemplos de classificação são:

- a) decidir se aplicações em empréstimos futuros serão ou não realizados;
- b) esclarecer pedidos de seguro fraudulentos;
- c) atribuir palavras-chave a artigos jornalísticos.

Um modelo de classificação apanha um novo registro e atribui ao mesmo uma classificação existente. Um modelo de previsão é semelhante a um modelo de classificação, exceto por não ser limitado a um conjunto de número de classes. Um modelo de agrupamento toma vários registros e retorna um número menor de grupos. Esses grupos podem então ser aplicados a novos registros, criando um modelo de classificação. Um modelo de séries temporais é como um modelo de classificação ou de previsão, exceto por incluir dados tomados com o decorrer do tempo ([BER1997]).

4.4.2 ESTIMATIVA

A estimativa lida com resultados contínuos, ao contrário da classificação que lida com resultados discretos. Fornecidos alguns dados, usa-se a estimativa para estipular um valor para alguma variável contínua desconhecida como receita, altura ou saldo de cartão de crédito.

Ao invés de um classificador binário determinar um risco “positivo” ou “negativo”, a técnica gera valores de “escore”, dentro de uma determinada margem. A abordagem de estimativa tem a grande vantagem de que os registros individuais podem ser agora ordenados por classificação, e as redes neurais são adequadas a esta tarefa ([FAY1996]).

Exemplos de estimativa incluem:

- a) estimar o número de filhos numa família;

- b) estimar a renda total de uma família;
- c) estimar o valor em tempo de vida de um cliente.

4.4.3 AGRUPAMENTO POR AFINIDADE

O algoritmo de agrupamento por afinidade identifica afinidades entre itens de um subconjunto de dados, sendo que as mesmas são expressas na forma de regras: “72% de todos os registros que contém os itens A, B, e C também contém D e E”. A porcentagem de ocorrência (72 no caso) representa o fator de confiança da regra, e costuma ser usado para eliminar tendências fracas, mantendo apenas as regras mais fortes. Dependências funcionais podem ser vistas como regras de associação com fator de confiança igual a 100%.

Trata-se de um algoritmo tipicamente endereçado à análise de mercado, onde o objetivo é encontrar tendências dentro de um grande número de registros de compras, por exemplo, expressas como transações. Essas tendências podem ajudar a entender e explorar padrões de compra naturais, e podem ser usadas para ajustar mostruários, modificar prateleiras ou propagandas, e introduzir atividades promocionais específicas ([FIG1998]).

4.4.4 PREVISÃO

Para [FAY1996], a previsão é o mesmo que classificação ou estimativa, exceto pelo fato de que os registros são classificados de acordo com alguma atitude futura prevista. Em um trabalho de previsão, o único modo de confirmar a precisão da classificação é esperar para ver. Essa tarefa é uma variante do problema de agrupamento por afinidades, onde as regras encontradas entre as relações podem ser usadas para identificar seqüências interessantes, que serão utilizadas para predizer acontecimentos subsequentes. Nesse caso, não apenas a coexistência de itens dentro de cada transação é importante, mas também a ordem em que aparecem, e o intervalo entre elas. Seqüências podem ser úteis para identificar padrões temporais, por exemplo entre compras em uma loja, ou utilização de cartões de crédito, ou ainda tratamentos médicos.

Exemplos de tarefas de previsão:

- a) previsão de quais clientes sairão nos próximos seis meses;
- b) previsão da quantia de dinheiro que um cliente utilizará caso seja oferecido a ele um certo limite de cartão de crédito.

4.4.5 SEGMENTAÇÃO

A segmentação é um processo de agrupamento de uma população heterogênea em vários subgrupos ou *clusters* mais homogêneos. O que a distingue da classificação é que a segmentação não depende de classes pré-determinadas.

A segmentação é realizada automaticamente por algoritmos que identificam características em comum e particionam o espaço n-dimensional definido pelos atributos. Os registros são agrupados de acordo com a semelhança e depende do usuário determinar qual o significado de cada segmento, caso exista algum. Muitas vezes a segmentação é uma das primeiras etapas dentro de um processo de *Data Mining*, já que identifica grupos de registros correlatos, que serão usados como ponto de partida para futuras explorações[FEI1999].

O exemplo clássico é o de segmentação demográfica, que serve de início para uma determinação das características de um grupo social, visando desde hábitos de compras até utilização de meios de transporte.

4.5 TÉCNICAS DE DATA MINING

As técnicas utilizadas atualmente em *Data Mining* são extensões de métodos analíticos. O que as diferencia, é que estas técnicas são mais voltadas a encontrar estratégias para os negócios. Isto acontece graças ao aumento de dispositivos e ao aumento de performance dos computadores ([DAL2000]).

Segundo [FIG1998], por causa da grande quantidade de dados, todo o processo de KDD ainda requer pré/pós-processamentos dos dados, necessários para assegurar o melhor aproveitamento da aplicação e a consistência dos resultados. Atividades de pré-processamento

incluem a seleção apropriada de subconjuntos de dados, por razões de desempenho, assim como complexas transformações de dados que servem de ponte para o chamado “*gap* representacional”, separação entre os dados e seu significado real. Pós-processamento envolve a subseleção de resultados volumosos e a aplicação de técnicas de visualização para auxiliar o entendimento. Essas atividades são críticas para contornar alguns problemas de implementação, tais como:

- a) alta suscetibilidade a dados “sujos”: as ferramentas de *Data Mining* via de regra não possuem uma estrutura dotada de semântica, orientada a aplicação, e como tal, tomam todos os dados factualmente. Torna-se necessário tomar precauções para assegurar que os dados analisados são “limpos”, o que pode significar uma exaustiva análise dos atributos que alimentam os algoritmos. Entretanto, um bom processo de “limpeza de dados” (*data cleaning*), certamente beneficia o processo de *Data Mining*.
- b) inabilidade para “explicar” resultados em termos humanos: mesmo em aplicações utilizando árvores de decisão e regras de indução, que são capazes de gerar informação sobre os atributos utilizados, o volume e formato da informação encontrada pode ser inútil sem um processamento adicional.
- c) “*gap*” representacional: a maior parte das fontes de dados das aplicações de *Data Mining* atuais está armazenada em grandes sistemas relacionais, e seus dados estão em geral normalizados, com os atributos espalhados em múltiplas tabelas. Além disso, a maioria das ferramentas é restrita em termos dos tipos de dados com as quais podem operar, tornando-se necessário categorizar variáveis ou remapeá-las.

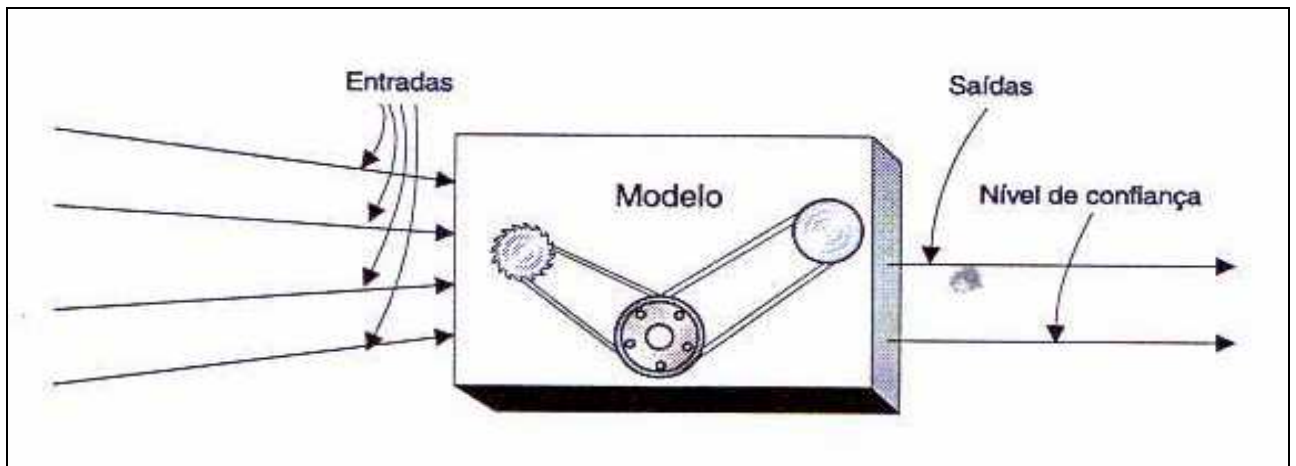
Conforme salienta [FIG1998], um modelo produz um ou mais valores a partir de um dado conjunto de entradas. A análise dos dados é, com frequência, o processo de construção de um modelo apropriado para os dados (figura 7). Um exemplo disso é uma regressão linear, onde é construída sobre um modelo em linha com a seguinte forma:

$$aX + bY + c = 0$$

Onde a , b , c são os parâmetros e X e Y são as variáveis. Para um dado valor de X , estima-se o valor de Y . Este tipo de modelo é um dos mais simples existentes.

O fato de um modelo existir não significa que proporcionará resultados precisos. Existem bons e maus modelos e, medir seus resultados é um passo crítico em seu uso e desenvolvimento ([FIG1998]).

Figura 7 - Modelo recebe entradas e produz informações.



Fonte: [BER1997]

Na criação dos modelos, a entrada é geralmente especificada claramente. Preparar os dados de sistemas para preencher o domínio de um modelo – chamado de depuração de dados ou *data scrubbing* – é mais desafiador do que a própria criação do modelo. Os dados que alimentarão o modelo podem afetar a escolha da técnica. Para problemas físicos, com muitas variáveis contínuas de entrada, as técnicas de regressão estatísticas normalmente funcionam muito bem. Quando as entradas têm muitas variáveis de categorias, as árvores de decisão funcionam melhor. Quando a relação entre as entradas e a saída de dados é difícil de ser estabelecida, as redes neurais são as melhores opções. Frequentemente a saída de dados de um modelo é especificada em primeiro lugar e geralmente é uma categoria ou uma variável contínua.

Segundo [BER1997], para criar um modelo para *Data Mining*, deve-se ter em mente o seguinte:

- a) um dos perigos no uso de modelos é o excesso ou a carência de dados;
- b) tanto o *Data Mining* direto quanto o indireto usam modelos, mas de maneira diversa;

- c) alguns modelos expõem sua finalidade melhor que outros;
- d) alguns modelos são mais fáceis de aplicar que outros.

Cada classe de aplicação em *data mining* tem como base um conjunto de algoritmos que serão usados na extração de relações relevantes dentro de uma massa de dados: análise de seqüências, clusterização, classificação, estimativas e regras de associação. Outras técnicas mais recentes incluem lógica difusa (*fuzzy logic*) e algoritmos genéticos. Cada uma destas propostas difere quanto à classe de problemas que o algoritmo será capaz de resolver.

4.5.1 ANÁLISE DE SELEÇÃO ESTATÍSTICA

A análise de seleção estatística é uma forma de agrupamento usada para encontrar grupos de itens que tendem a ocorrer em conjunto em uma seleção estatística. Como técnica de agrupamento, ela é útil quando se deseja saber quais itens ocorrem ao mesmo tempo ou em uma seqüência particular ([FIG1998]).

4.5.2 MBR

O MBR (*Memory-Based Reasoning* – raciocínio baseado em memória) é uma técnica de *data mining* dirigida que usa exemplos conhecidos como modelo para fazer previsões sobre exemplos desconhecidos. O MBR procura os vizinhos mais próximos nos exemplos conhecidos e combina seus valores para atribuir valores de classificação ou de previsão ([BER1997]).

Os elementos-chave no MBR são a função de distância usada para encontrar os vizinhos mais próximos e a função de combinação, que combina valores dos vizinhos mais próximos para fazer uma previsão. Uma vantagem do MBR é sua habilidade de aprender sobre novas classificações simplesmente introduzindo novos exemplos no banco de dados. Uma vez encontrada a função de distância e a função de combinação corretas tendem a permanecer muito estáveis, mesmo com a incorporação de novos exemplos para novas

categorias nos dados conhecidos. Aliás, esta é uma característica que diferencia o MBR da maior parte das outras técnicas de *data mining*.

4.5.3 ALGORITMOS GENÉTICOS

Para [AVI1998], um algoritmo genético é um procedimento de busca baseado em mecanismos de seleção natural e não no processo de raciocínio simulado. A idéia básica é usar um conjunto de descrições candidatas, chamada população, e gradualmente aumentar a qualidade dessa população, construindo novas descrições montadas a partir de partes das melhores descrições da população corrente. As novas descrições geradas formam a segunda geração e novamente as melhores descrições são recombinadas para formar a próxima geração, até que descrições de qualidade suficiente sejam geradas ou nenhuma evolução ocorra.

Eles são utilizados no *data mining* dirigido e são semelhantes à estatística, em que a forma do modelo precisa ser conhecida em profundidade. Os algoritmos genéticos usam os operadores seleção, cruzamento e mutação para desenvolver sucessivas gerações de soluções. Com a evolução do algoritmo, somente os mais previsíveis sobrevivem, até as funções convergirem em uma solução ideal ([BER1997]).

4.5.4 ANÁLISE DE GRUPOS

Esta técnica constitui-se na construção de modelos para encontrar dados semelhantes, e estas reuniões por semelhança são chamadas de grupos (*clusters*). É uma forma de *data mining* não-direcionado, onde a meta é encontrar similaridades não conhecidas anteriormente. Existem muitas técnicas para encontrar grupos, incluindo métodos geométricos, estatísticos e redes neurais ([HAR1998]).

4.5.5 ANÁLISE DE VÍNCULOS

A análise de vínculos segue as relações entre registros para desenvolver modelos baseados em padrões nas relações. Esse é um aplicativo de construção de teoria gráfica de *data mining*. Esta técnica não é muito compatível com a tecnologia de banco de dados relacionais e sua maior área de aplicação é a área policial, onde pistas são ligadas entre si para solucionar os crimes. As poucas ferramentas que existem enfocam mais a visualização de vínculos que a análise de padrões ([HAR1998]).

4.5.6 ÁRVORES DE DECISÃO E INDUÇÃO DE REGRAS

As árvores de decisão são usadas para o *data mining* dirigido, mais especificamente a classificação. Esta técnica divide os registros do conjunto de dados de treinamento em subconjuntos separados, cada um descrito por uma regra simples em um ou mais campos ([HAR1998]).

Uma vantagem nesta técnica é que o modelo é bem explicável, já que tem a forma de regras explícitas. Isto permite às pessoas avaliarem os resultados, identificando os atributos-chave do processo.

4.5.7 REDES NEURAIIS ARTIFICIAIS

Para [AVI1998], redes neurais, também conhecidas como modelos conexionistas, são redes interconectadas formadas por elementos computacionais muito simples. O conceito de redes neurais, assim como muitos outros conceitos computacionais, foi retirado de uma outra área da ciência, neste caso, o modelo cerebral. Redes neurais são construídas conectando a saída de um neurônio para a entrada de um ou mais neurônios, associando alguns nós não conectados como a entrada da rede e nós particulares como nós saídas da rede. O resultado obtido por redes neurais em relação a regras de produção e outros métodos simbólicos é

praticamente o mesmo, entretanto redes neurais trabalham melhor com informações que possuem ruído.

As redes neurais são modelos simples de interconexões neurais no cérebro, adaptados para o uso em computadores. Elas aprendem com um conjunto de dados de treinamento, generalizando modelos para classificação e previsão. Esta técnica pode também ser aplicada ao *data mining* não-dirigido (na forma de redes Kohonen e estruturas relacionadas) e às previsões em séries temporais ([HAR1998]).

5 SISTEMA DE GERENCIAMENTO A SUPORTE

O atendimento de sistemas tem como finalidade identificar as potenciais incorreções no funcionamento dos sistemas desenvolvidos pela empresa e, disponibilizar soluções na sua forma de utilização ou caso necessário na correção dos programas.

5.1 ESTRUTURA DO SISTEMA DE GERENCIAMENTO A SUPORTE

O sistema de gerenciamento à suporte foi desenvolvido para banco de dados Oracle, assim como todos os sistemas que a empresa desenvolve são para banco de dados Oracle e são sistemas específicos para cada cliente. Quando o cliente solicita suporte sobre algum problema em seu sistema, este suporte poderá ser:

- a) Suporte Operacional – atua na resolução de problemas com equipamentos (redes, terminais, micros, vídeos, teclados, mouses, impressoras);
- b) Suporte de Sistemas - atua na resolução de problemas apresentados durante a utilização dos sistemas desenvolvidos pela empresa.

A tela da figura 8 é a tela onde são registrados todas solicitações de atendimentos dos clientes tanto do suporte operacional como do suporte de sistemas. Quando o usuário do sistema entra em contato com a equipe de suporte ele deve informar qual é o cliente que está solicitando suporte, qual localidade, setor em que o usuário trabalha na empresa, qual o sistema e o programa que ele está utilizando e relatar o que está ocorrendo. Após relatado o problema o técnico irá informar a situação do chamado(pendente, fechado), qual a severidade do chamado , esta severidade é para determinar qual tempo para solução deste chamado, e o motivo do chamado que pode ser erro de programa, consulta, erro utilização do projeto e suporte operacional. O chamado só tem sua situação alterada para fechado com o aval do usuário que realizou a chamada de suporte.

Figura 8 - Tela para abertura de chamados

Oracle Developer Forms Runtime - [Atendimento de chamados]

Arquivo Editar Auxílio Opções Window

REGISTRAR ATENDIMENTO @

Chamado: Chamados Abertura: Fechamento:

Cliente:

Localidade: Setor:

Usuário: Fone:

Situação: Severidade:

Motivo:

Assunto: Versão Sistema:

Téc. Resp:

Ocorrências do Chamado

Data e Hora: 05/05/2000 08:00 Tipo: Técnico: 240

Enter a query: press F8 to execute, CTRL+q to cancel.
Record: 1/1 Enter-Query <DSC> <DBG>

Iniciar Oracle Developer For... 07:48

COR_CLIENT – Tabela onde são armazenados os dados cadastrais dos clientes da empresa.

COR_SISTEM - Tabela onde são armazenados os dados dos sistemas desenvolvidos pela empresa.

COR_USUAR – Tabela onde são armazenados os dados sobre os usuários .

INF_PRIOR – Tabela onde estão armazenados as prioridades de atendimento a suporte.

INF_CHAMDO_TECNO – Tabela onde são armazenados dados referentes aos chamados, nesta tabela são gravados a severidade, situação, motivo e o número dos mesmos.

INF_MOTIVO_CHAMDO - Nesta tabela são armazenados todos os possíveis motivos para um chamado. Os motivos para abertura de um chamado são:

Erro na Definição;

Erros de Programa;

Melhorias;

Consultas;

Erro Utilização projeto;

Suporte;

Corporativo;

INF_OCORR_CHAMDO – Tabela onde são armazenadas todas as ocorrências para um chamado. As ocorrências de uma chamado podem ser:

Problema (P);

Solução (S);

Comentário(C);

Aguardando Usuário (A);

Retorno (R);

INF_VERSAO_SISTEM - Tabela onde são armazenados o nome do sistema e a versão do mesmo.

A estrutura das tabelas são relacionadas no apêndice 1.

6 O PROTÓTIPO DESENVOLVIDO

6.1 INTRODUÇÃO

A proposta de construir um protótipo de um sistema especialista utilizando a ferramenta *Expert Sinta Shell* e técnicas de *data mining* para auxílio à tomada de decisão em suporte a sistemas, objetiva apoiar o especialista na realização de suas tarefas. O desenvolvimento de um protótipo deve necessariamente possuir uma especificação, onde define-se os requisitos da aplicação.

O passo inicial da fase de especificação do sistema é um levantamento de dados e informações para nortear o desenvolvimento do protótipo. Este levantamento é feito contatando-se o usuário, e elaborando-se uma descrição textual, por exemplo. A partir desta descrição, parte-se para a etapa de modelagem do protótipo. No capítulo 5 foi feita a descrição do sistema de gerenciamento a suporte, e com as informações coletadas pôde-se realizar a especificação através da notação BNF (Backus-Naur Form).

6.2 ESPECIFICAÇÃO

A forma de representação do conhecimento escolhida para o sistema especialista são regras de produção. A estrutura das regras é representada abaixo, expressa pela notação BNF(Backus-Naur Form):

$$\langle \text{regra} \rangle ::= \text{SE} \langle \text{condição} \rangle \text{ ENTÃO} \langle \text{cláusula} \rangle \langle \text{certeza} \rangle$$

$$\langle \text{condição} \rangle ::= \langle \text{cláusula} \rangle \mid \langle \text{cláusula} \rangle \text{ E} \langle \text{cláusula} \rangle$$

$$\langle \text{cláusula} \rangle ::= \langle \text{atributo} \rangle \langle \text{predicado} \rangle \langle \text{valor} \rangle$$

$$\langle \text{atributo} \rangle ::= \langle \text{cadeia} \rangle$$

$$\langle \text{predicado} \rangle ::= = \mid <$$

$\langle \text{valor} \rangle ::= \langle \text{cadeia} \rangle$

$\langle \text{cadeia} \rangle ::= \{ \text{letra minúscula} \}$

$\langle \text{letra minúscula} \rangle ::= a | b | c | d | \dots | v | x | z | |$

$\langle \text{certeza} \rangle ::= 1 | 2 | 3 | 4 | \dots | 98 | 99 | 100$

Como se observa na BNF, a regra é formada pelas partes SE e ENTÃO. Na parte SE, ou antecedente, da regra estão as premissas (caudas). Ela é formada por uma ou mais cláusulas. Já na parte ENTÃO, ou conseqüente (cabeça), está a conclusão da regra que é formada por uma cláusula única.

Uma cláusula é uma combinação de um atributo, um predicado e um valor. Os atributos e valores são cadeias de caracteres que o usuário cria livremente. Os predicados utilizados na regra são “=” (igual) e “<” (diferente). Para cada regra existe ainda um fator de certeza, este fator de certeza expressa o grau de certeza do conjunto de cláusulas que compõem a regra.

6.2.1 ATRIBUTOS E VALORES

Os atributos das regras são classificados em dois tipos: intermediário ou conclusivo. Um atributo conclusivo é aquele que representa uma solução para o problema, ou seja, existindo uma resposta para o atributo existirá uma resposta para o problema como um todo. Já os atributos intermediários são os que representam uma conclusão parcial que exigirá continuidade ao processo de busca da solução do problema.

Existe um domínio de respostas que cada atributo pode assumir. Estas possíveis respostas que um determinado atributo pode assumir são denominadas de valores. Veja alguns exemplos abaixo:

Atributo: estações do ano

Valores: {verão, inverno, outono, primavera}

Atributo: impressora não imprime

Valores: {falta papel, falta cartucho de impressão, impressora desligada}

A criação de atributos e valores para posterior utilização no sistema é livre. Cabe ao usuário do sistema fazer o cadastramento prévio dos mesmos para utilizá-los na elaboração das regras.

6.2.2 CLÁUSULAS E PREDICADOS

As cláusulas da regras são formadas pela ligação de um atributo com um valor através do uso de um predicado. O predicados disponíveis no *Expert Sinta Shell* são igualdade (“=”) e desigualdade (“<>”).

Atributo : Estações do Ano

Predicado =

Valor : verão

6.2.3 REGRAS

São elas que compõem o conhecimento do sistema especialista, sendo construídas através da conjunção de cláusulas. Cada regra é composta por uma cláusula obrigatória, na sua parte antecedente e uma cláusula, também obrigatória, na sua parte conseqüente. Alternativamente a cláusula pode possuir na sua parte antecedente qualquer número de cláusulas ligadas pelos conectivos lógicos “=” e “<>”.

6.2.4 FATOR DE CERTEZA

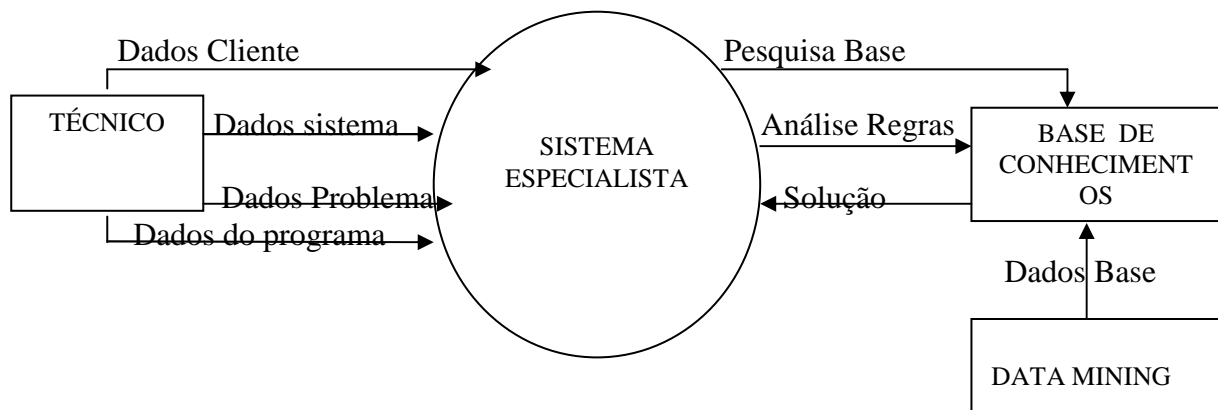
Existe um fator de certeza associado para cada regra na base de conhecimentos do sistema especialista. O fator de certeza é um número de 1 à 100 que demonstra o percentual de certeza do conhecimento descrito pela regra. Deve-se observar que o grau de certeza está relacionado à regra como um todo, ou seja, ao conjunto de cláusulas que compõem a regra.

6.3 MODELAGEM ESSENCIAL

Esta modelagem mostra, como o próprio nome diz, a essência do sistema a ser desenvolvido. É composta pelos modelos ambiental e comportamental. O modelo ambiental visa mostrar como o sistema interage com o ambiente externo e o modelo comportamental indica o que o sistema deve fazer para interagir com o ambiente externo. No presente trabalho somente foi desenvolvido o modelo ambiental.

O modelo ambiental é composto por um diagrama de contexto, que representa o fluxo de dados, e por uma lista de eventos, que representa as tarefas que devem ser executadas no sistema. A figura 10 mostra o diagrama de contexto.

Figura 10 - Diagrama de Contexto



Os eventos do protótipo são:

1. Técnico solicita consulta sobre sistema, cliente, programa, problema.
2. Processo de Data Mining gera dados para formação da base de conhecimentos para o sistema especialista.
3. Sistema Especialista pesquisa dados na base de conhecimentos.
4. Sistema Especialistas analisa dados da base de conhecimentos.
5. Base fornece a solução(conhecimento) para o sistema especialista

6.4 PLATAFORMA DE DESENVOLVIMENTO

O protótipo foi desenvolvido para a plataforma PC em ambiente Windows, utilizando-se um microcomputador pentium 100 Mhz, e 32 Mb de memória RAM, sendo que com esta plataforma há queda de performance. Para implementar o protótipo optou-se pelo ambiente de programação visual Borland Delphi, em sua versão 3.0, pois a biblioteca VCL *Expert Sinta Shell* na versão utilizada para o desenvolvimento do protótipo foi desenvolvida para ser utilizada com Delphi 3.0. O ambiente de programação Delphi possui algumas características merecedoras de destaque, como: "abordagem baseada em formulários e orientada a objetos, compilador extremamente rápido, suporte a banco de dados, integração com a programação em Windows e sua tecnologia de componentes" ([CAN1998]).

6.5 PROCESSO DE KDD

O processo de KDD para desenvolvimento do protótipo iniciou com o entendimento do domínio da aplicação e dos objetivos finais a serem atingidos. A seguir foi realizada a etapa da limpeza dos dados (*data cleaning*), através de um pré-processamento dos dados, visando adequá-los ao algoritmo a ser utilizado na prospecção do conhecimento, pois a base de dados do sistema de gerenciamento a suporte não foi desenvolvida visando-se a implementação de um sistema especialista por isso a mesma contém muita “sujeira”.

A implementação da etapa de limpeza dos dados baseou-se em pesquisas na base de informações, através de instruções em SQL (*Structured Query Language* - linguagem de consulta estruturada), que é uma linguagem de consulta e manipulação de bancos de dados. Estas pesquisas tem por finalidade identificar o número dos chamados que se relacionam com determinado problema.

Prosseguindo no processo, chegou-se à fase de *Data Mining* especificamente, que começou com a escolha do algoritmo a ser aplicado. O algoritmo utilizado para implementar a fase de *Data Mining* foi um algoritmo utilizando regras de produção.

Ao final do processo, o sistema de *Data Mining* gerou dados para a construção da base de conhecimentos do sistema especialista. Somente após a construção da base de

conhecimentos, realizada através da ferramenta Expert SINTA Shell, encontra-se o conhecimento.

6.5 OPERACIONALIDADE DO PROTÓTIPO

A seguir, será apresentado o funcionamento do protótipo. Nesta apresentação, serão mostradas as telas do protótipo, bem como as características de cada uma delas.

A figura 11 é a primeira tela, sendo a abertura do sistema.

Figura 11 - Tela de Abertura



Na opção de menu Arquivo -> Data Mining (figura 12) é iniciado o processo de mineração dos dados baseado na técnica de Regras de Produção, para gerar as regras da base de conhecimentos. Após a geração das regras na base de conhecimentos o usuário deverá utilizar a opção Arquivo -> Abrir Base (figura 13) para abrir a base de conhecimentos que foi gerada na opção Data Mining, o arquivo que contém a base está no diretório da aplicação com o nome TCC.BCM. Após a abertura da base o usuário poderá iniciar a consulta ao Sistema Especialista através da opção de menu Consulta -> Iniciar (figura 14)..

A partir deste momento, o sistema irá realizar uma seqüência de perguntas ao usuário, que deve responder clicando sobre uma das opções de cada quadro na tela (figura 15), no lado esquerdo da tela do protótipo são demonstradas todas as regras da base de

conhecimentos, pois o usuário poderá realizar a consulta passo à passo e o sistema vai mostrando quais as regras que ele está analisando em determinado momento da consulta.

O usuário responde as perguntas, clicando sobre cada uma delas, sendo que estas perguntas estão acima de cada quadro com mais de uma alternativa (figuras 15, 16, 17, 18 e 19).

Figura 12 - Tela para geração da Base de Conhecimentos



Figura 13 - Opção para abrir a base de Conhecimentos do Sistema Especialista



Figura 14 - Opção para iniciar o processo de consulta ao Sistema Especialista



Figura 15 – Tela sobre dados do cliente

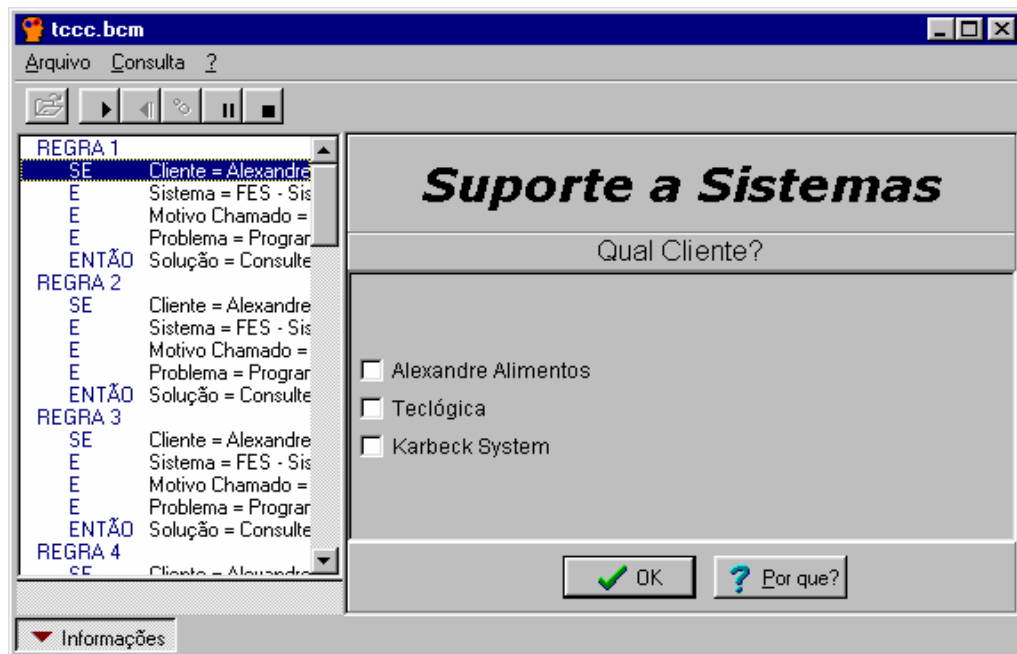


Figura 16 – Tela sobre sistema que o cliente está acessando.



Figura 17 – Tela sobre motivo do chamado do cliente.

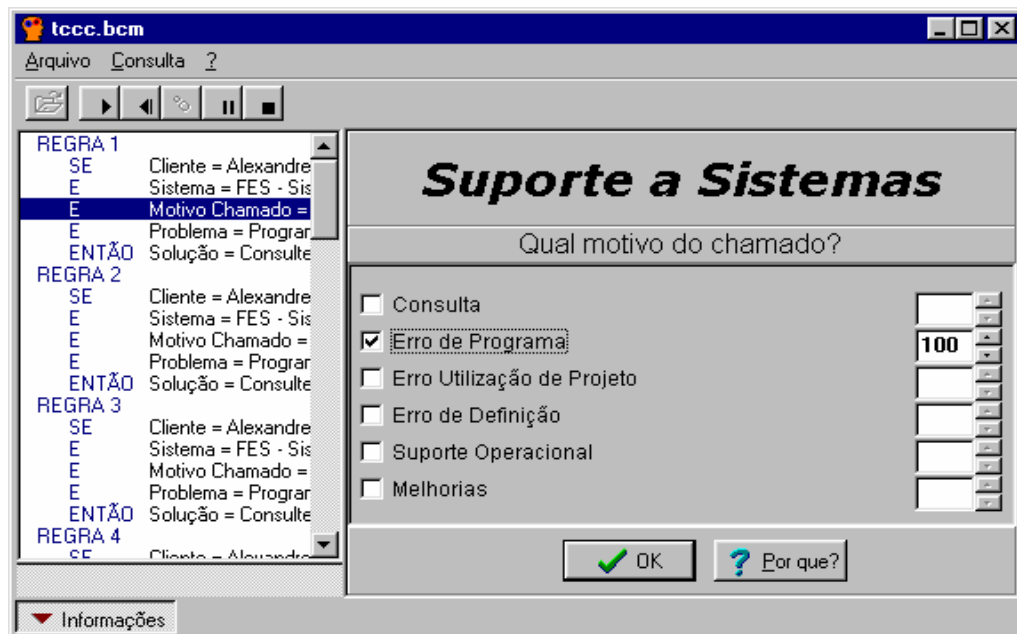


Figura 18 – Tela sobre problema do cliente

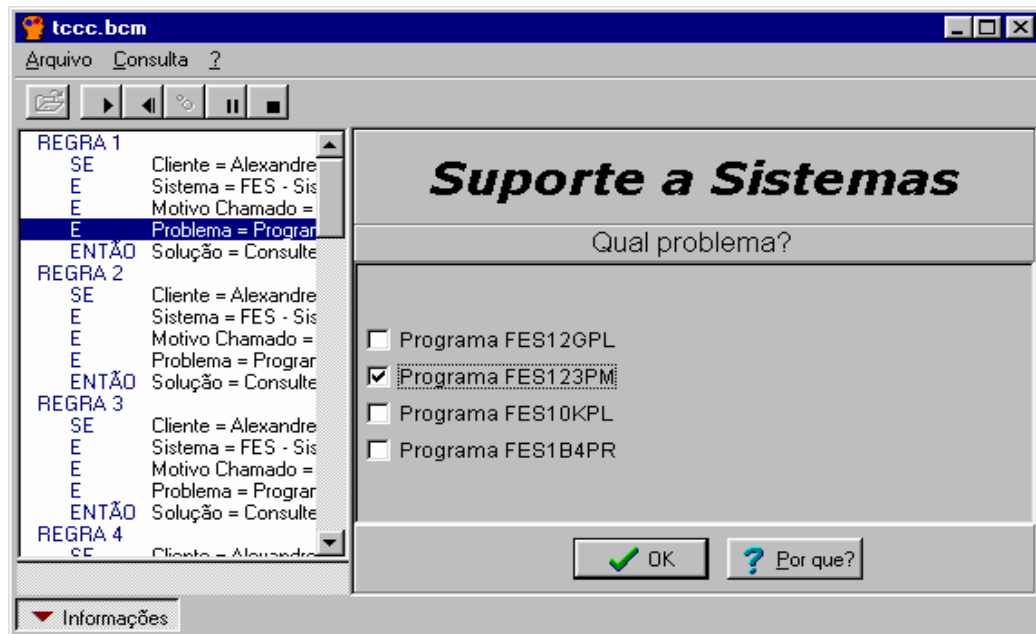
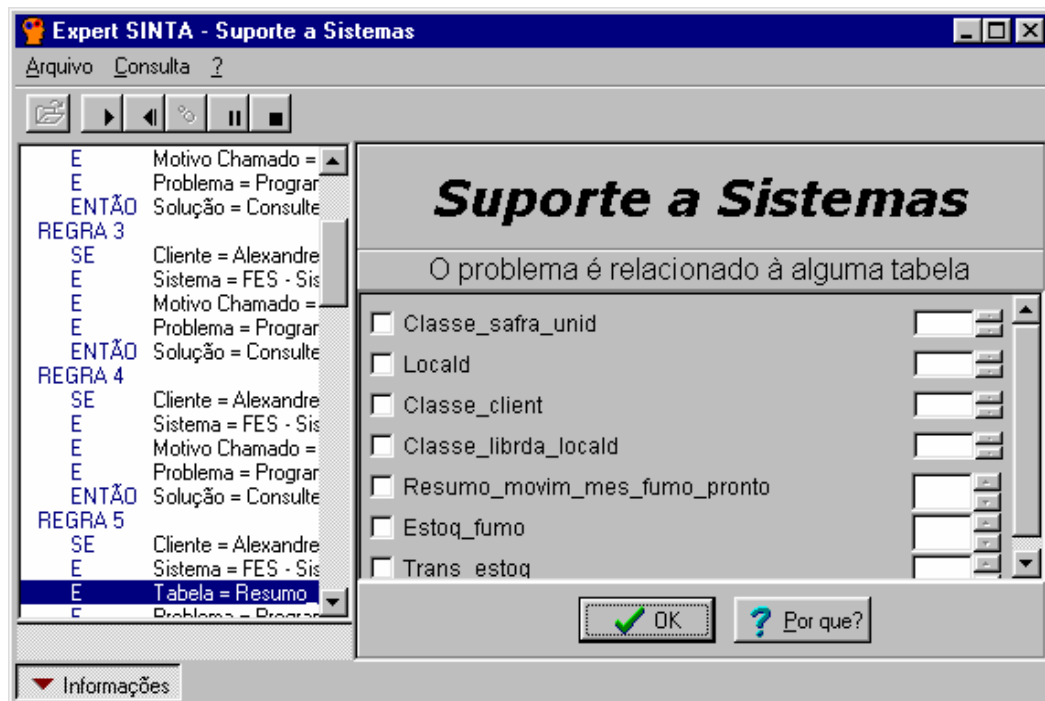


Figura 19 – Tela sobre dados adicionais do problema

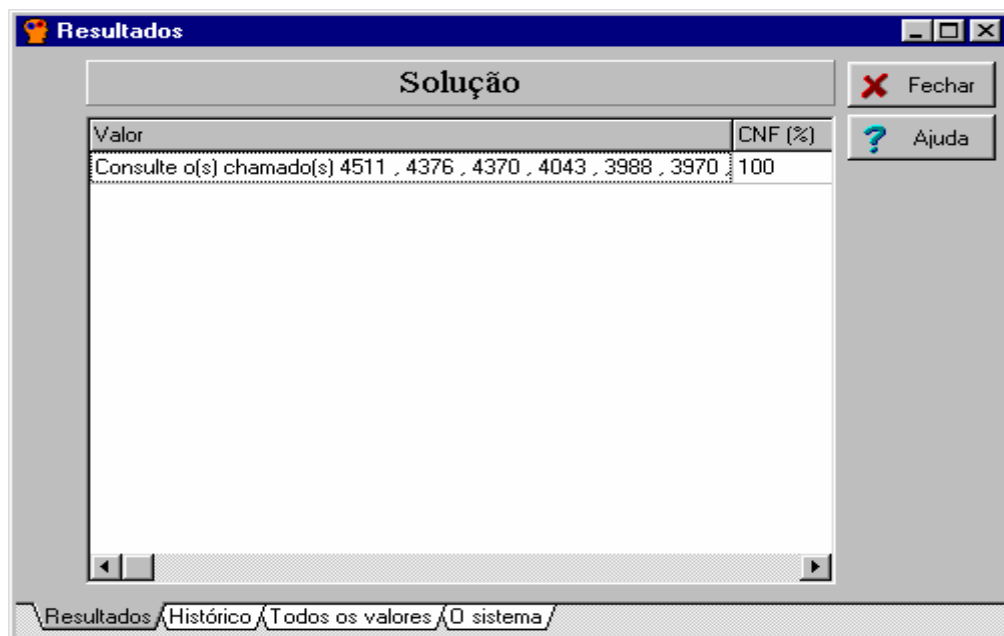


Sendo as perguntas realizadas de uma forma genérica, ou seja para todos os tipos de problemas descritos no sistema, algumas destas perguntas não necessitam ser respondidas, desde que não tenham relacionamento nenhum com a definição das soluções a serem apresentadas. Um exemplo a ser apresentado é a pergunta referente as tabelas que

supostamente possuem problema (figura 19). Neste exemplo caso o problema esteja relacionado a um programa e não a uma tabela do sistema, não é necessário responder a pergunta referente a tabela.

Na figura 20, é apresentada a solução para as perguntas que o usuário fez ao sistema especialista, sendo que estes valores foram extraídos da base de dados do sistema de gerenciamento o suporte e salvos da base de conhecimentos através do processo de KDD.

Figura 20 – Tela de compreensão dos resultados



Na figura 21 são demonstrados todos os passos que a máquina de inferência seguiu até encontrar uma solução para o questionamento do usuário.

Figura 21 – Demonstração dos passos do Sistema Especialista

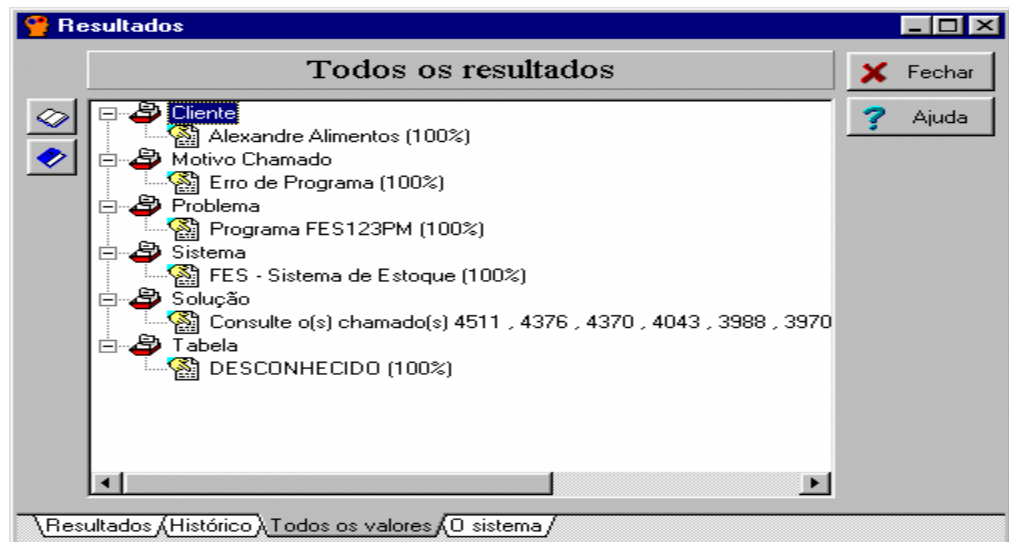
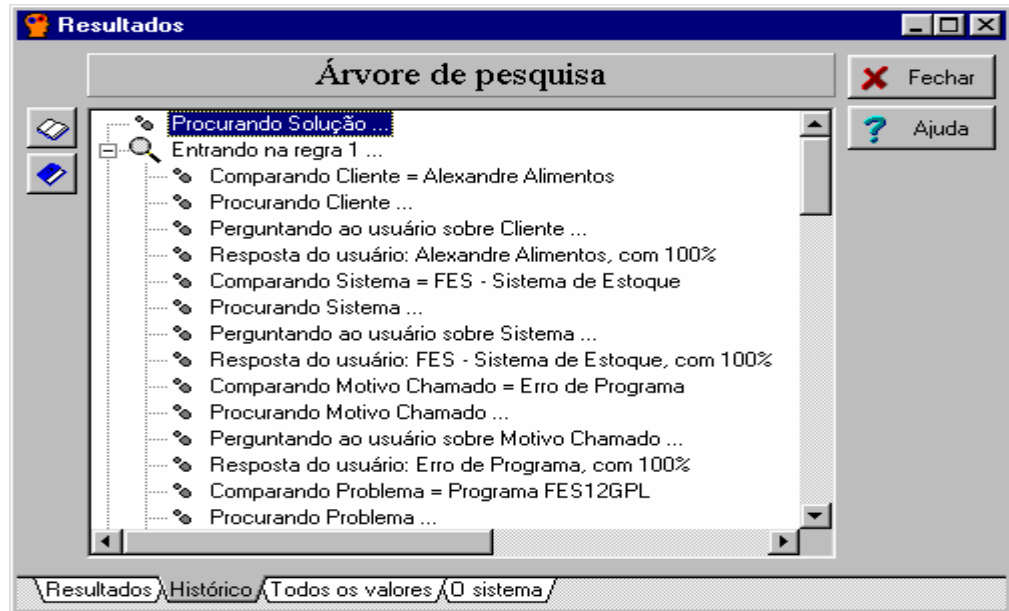
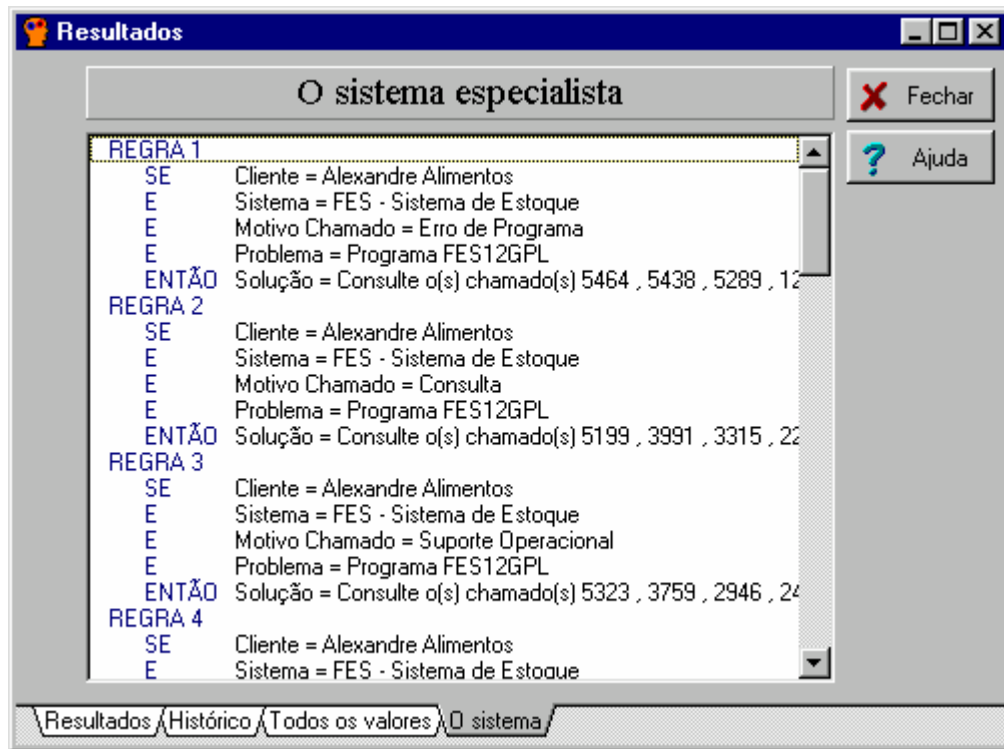


Figura 22 – Variáveis do Sistema e seus valores

Na figura 22 estão todas as variáveis que compõem a base de conhecimentos e os valores que elas assumiram durante determinada consulta.

Figura 23 – Regras da base de conhecimentos



Na figura 23 estão todas as regras que compõem a base de conhecimentos do sistema especialista. Após o fechamento desta tela, que deve ser realizado clicando sobre o botão Fechar, é finalizado a consulta, aparecendo uma mensagem ao usuário. Deste modo, pode ser iniciado novamente a consulta, voltando o sistema à figura 15. As respostas feitas anteriormente para cada pergunta permanecem marcadas, dando oportunidade de alterá-las caso desejar.

Para incluir novas variáveis na base de conhecimento, o usuário deverá abrir a base de conhecimentos na *Shell Expert SINTA* e utilizar o editor de bases para alterá-la, o usuário poderá também incluir regras fixas na base utilizando-se do mesmo recurso do editor de bases.

No apêndice 2 constam as quatorze regras que compõem a base de conhecimentos da aplicação, as quais podem ter seus valores alterados conforme são acrescentadas informações à base de dados.

7 CONCLUSÕES E SUGESTÕES

7.1 CONCLUSÕES

A proposta do presente trabalho foi de demonstrar a utilização da tecnologia dos sistemas especialistas, unindo a ferramenta *Expert Sinta Shell* a técnicas de *data mining*, aplicados na área de suporte a sistemas, objetivando auxiliar o técnico responsável por esta tarefa a solucionar o problema apresentado de forma rápida e eficiente.

O sistema especialista no domínio do conhecimento no qual foi construído demonstrou que os objetivos pretendidos foram alcançados. A soma de técnicas de *data mining* com a *Expert Sinta Shell* demonstrou a possibilidade de utilização de sistemas especialistas no suporte a sistemas, na forma de uma ferramenta de apoio à decisão, liberando o especialista para outras atividades correlacionadas.

O sistema especialista desenvolvido trouxe a *software house* uma maior agilidade na solução dos chamados, pois muitos dos problemas que o cliente enfrenta já foram em algum momento solucionados e estão armazenados na base do sistema de gerenciamento a suporte sem que o técnico possa consultá-los de uma maneira fácil e rápida.

7.2 LIMITAÇÕES

O protótipo construído possui algumas limitações, como geralmente acontece em trabalhos desenvolvidos em um espaço curto de tempo como o definido para este estudo. Pode-se citar as seguintes limitações:

- a) uso de poucas variáveis na definição da base de conhecimentos do sistema;
- b) não foi realizado estudo comparativo sobre as técnicas de *data mining* a fim de identificar qual se mostraria mais adequada ao projeto.

7.3 SUGESTÕES PARA TRABALHOS FUTUROS

Para as tecnologias apresentadas neste estudo, inúmeros caminhos poderão se abrir, mostrando um vasto número de aplicações possíveis. No caso específico deste trabalho, poderia ser feito um estudo mais aprofundado sobre todo o processo de suporte a sistemas e definido mais regras via *Data Mining*.

Poderia ser feito um estudo mais detalhado sobre as técnicas de *data mining*, aprimorando o3 processo aqui implementado.

APÊNDICE 1 – ESTRUTURA DAS TABELAS

Tabela 1 - COR_CLIENT

Nome	Null	Tipo
IDT_CLIENT	NOT NULL	NUMBER(5)
NOM_CLIENT	NOT NULL	VARCHAR2(30)

Tabela 2 - COR_SISTEM

Nome	Null	Tipo
IDT_SISTEM	NOT NULL	NUMBER(3)
SIG_SISTEM	NOT NULL	VARCHAR2(5)
DSC_SISTEM	NOT NULL	VARCHAR2(100)

Tabela 3 - INF_CHAMDO_TECNO

Nome	Null	Tipo
NUM_CHAMDO	NOT NULL	NUMBER(6)
IND_TIPO_CHAMDO	NOT NULL	VARCHAR2(1)
IND_SEVER_CHAMDO	NOT NULL	VARCHAR2(1)
IND_SITUAC_CHAMDO	NOT NULL	VARCHAR2(1)
NOM_USUAR	NOT NULL	VARCHAR2(30)
NUM_TEL_USUAR		VARCHAR2(20)
DAT_ABERT_CHAMDO	NOT NULL	DATE
DAT_FECHTO_CHAMDO		DATE
IDT_MOTIVO_CHAMDO	NOT NULL	NUMBER(3)
IDT_LOCALD_CHAMDO	NOT NULL	NUMBER(5)
IDT_SETOR_CLIENT	NOT NULL	NUMBER(5)
IDT_ASSUTO		NUMBER(5)
IDT_VERSAO_SISTEM		NUMBER(5)
IDT_TECNO		NUMBER(5)
IDT_AFS		NUMBER(10)

Tabela 4 - INF_MOTIVO_CHAMDO

Nome	Null	Tipo
IDT_MOTIVO_CHAMDO	NOT NULL	NUMBER(3)
DSC_ABREV_MOTIVO_CHAMDO	NOT NULL	VARCHAR2(20)

Tabela 5 - INF_OCORR_CHAMDO

Nome	Null	Tipo
IDT_OCORR_CHAMDO	NOT NULL	NUMBER(8)
IND_TIPO_OCORR_CHAMDO	NOT NULL	VARCHAR2(1)
DAT_HOR_OCORR_CHAMDO	NOT NULL	DATE
DSC_OCORR_CHAMDO	NOT NULL	VARCHAR2(2000)
NUM_CHAMDO	NOT NULL	NUMBER(6)
IDT_TECNO	NOT NULL	NUMBER(5)

APÊNDICE 2 – REGRAS DA BASE DE CONHECIMENTOS

Regra 1

SE Cliente = Alexandre Alimentos

E Sistema = FES - Sistema de Estoque

E Motivo Chamado = Erro de Programa

E Problema = Programa FES12GPL

ENTÃO Solução = Consulte o(s) chamado(s) 5464 , 5438 , 5289 , 1237 CNF 100%

Regra 2

SE Cliente = Alexandre Alimentos

E Sistema = FES - Sistema de Estoque

E Motivo Chamado = Consulta

E Problema = Programa FES12GPL

ENTÃO Solução = Consulte o(s) chamado(s) 5199 , 3991 , 3315 , 2270 , 1036 , 961
CNF 100%

Regra 3

SE Cliente = Alexandre Alimentos

E Sistema = FES - Sistema de Estoque

E Motivo Chamado = Suporte Operacional

E Problema = Programa FES12GPL

ENTÃO Solução = Consulte o(s) chamado(s) 5323 , 3759 , 2946 , 2482 CNF 100%

Regra 4

SE Cliente = Alexandre Alimentos

E Sistema = FES - Sistema de Estoque

E Motivo Chamado = Erro de Definição

E Problema = Programa FES12GPL

ENTÃO Solução = Consulte o(s) chamado(s) 2297 , 932 , CNF 100%

Regra 5

SE Cliente = Alexandre Alimentos

E Sistema = FES - Sistema de Estoque

E Tabela = Resumo_movim_mes_fumo_pronto

E Problema = Programa FES12GPL

ENTÃO Solução = Consulte o(s) chamado(s) 5438 ,3991,3759,2297,2270 CNF
100%

Regra 6

SE Cliente = Alexandre Alimentos

E Sistema = FES - Sistema de Estoque

E Motivo Chamado = Erro de Programa

E Tabela = Resumo_movim_mes_fumo_pronto

E Problema = Programa FES12GPL

ENTÃO Solução = Consulte o(s) chamado(s) 5438 CNF 100%

Regra 7

SE Cliente = Alexandre Alimentos

E Sistema = FES - Sistema de Estoque

E Motivo Chamado = Consulta

E Tabela = Resumo_movim_mes_fumo_pronto

E Problema = Programa FES12GPL

ENTÃO Solução = Consulte o(s) chamado(s) 3991 , 2270 CNF 100%

Regra 8

SE Cliente = Alexandre Alimentos

E Sistema = FES - Sistema de Estoque

E Motivo Chamado = Suporte Operacional

E Tabela = Resumo_movim_mes_fumo_pronto

E Problema = Programa FES12GPL

ENTÃO Solução = Consulte o(s) chamado(s) 3759 CNF 100%

Regra 9

SE Cliente = Alexandre Alimentos
E Sistema = FES - Sistema de Estoque
E Motivo Chamado = Erro de Definição
E Tabela = Resumo_movim_mes_fumo_pronto
E Problema = Programa FES12GPL
ENTÃO Solução = Consulte o(s) chamado(s) 2297 CNF 100%

Regra 10

SE Cliente = Alexandre Alimentos
E Sistema = FES - Sistema de Estoque
E Motivo Chamado = Erro de Programa
E Problema = Programa FES123PM
ENTÃO Solução = Consulte o(s) chamado(s) 4511 , 4376 , 4370 , 4043 , 3988 ,
3970 , 3931 , 3820 , 3386 , 1700 , 1195 CNF 100%

Regra 11

SE Cliente = Alexandre Alimentos
E Sistema = FES - Sistema de Estoque
E Motivo Chamado = Consulta
E Problema = Programa FES123PM
ENTÃO Solução = Consulte o(s) chamado(s) 5749 , 5299 , 5047 , 4882 , 4498 ,
3821 , 3525 , 3368 , 3361 , 3246 , 3240 , 3234 , 3222 , 3205 , 2969 , 2951 , 2755 ,
2160 , 2151 , 2142 , 2115 , 2086 , 1963 , 1957 , 1815 , 1785 , 1697 , 1660 , 1633 ,
1484 CNF 100%

Regra 12

SE Cliente = Alexandre Alimentos
E Sistema = FES - Sistema de Estoque
E Motivo Chamado = Suporte Operacional
E Problema = Programa FES123PM
ENTÃO Solução = Consulte o(s) chamado(s) 5707 , 5592 , 5587 , 5528 , 5362 ,
5303 , 4967 , 4892 , 4891 , 4890 , 4889 , 4858 , 4223 , 2971 , 2712 , 2069 , 2016 ,
1714 , 1662 , 1656 , 1649 , 1646 , 1644 , 1615 , 1455 CNF 100%

Regra 13

SE Cliente = Alexandre Alimentos

E Sistema = FES - Sistema de Estoque

E Motivo Chamado = Erro de Definição

E Problema = Programa FES123PM

ENTÃO Solução = Consulte o(s) chamado(s) 4381 , 4324 , 4020 , 1628 , 862 , 705 , 456 , 57 , 56 , 54 , 53 , 50 CNF 100%

Regra 14

SE Cliente = Alexandre Alimentos

E Sistema = FES - Sistema de Estoque

E Motivo Chamado = Erro Utilização de Projeto

E Problema = Programa FES123PM

ENTÃO Solução = Consulte o(s) chamado(s) 4964 , 4880 , 4301 , 4127 , 4041 , 4027 , 4018 , CNF 100%

APÊNDICE 3 – CÓDIGOS DA BASE DE CONHECIMENTOS

Variáveis - CÓDIGO, NOME

- 1, Problema
- 2, Motivo Chamado
- 3, Cliente
- 4, Sistema
- 5, Tabela
- 6, Solução

Valores - CÓDIGO, NOME, CÓDIGO DA VARIÁVEL, POSIÇÃO

- 1, Programa FES12GPL, 1, 1
- 2, Consulta, 2, 1
- 3, Erro de Programa, 2, 2
- 4, Erro Utilização de Projeto, 2, 3
- 5, Erro de Definição, 2, 4
- 6, Suporte Operacional, 2, 5
- 7, Melhorias, 2, 6
- 8, Alexandre Alimentos, 3, 1
- 9, Teclógica, 3, 2
- 10, Karbeck System, 3, 3
- 11, FES - Sistema de Estoque, 4, 1
- 12, FQC - Controle de Qualidade, 4, 2
- 13, FND - Notas Fiscais Diversas, 4, 3
- 14, Classe_safra_unid, 5, 1
- 15, Locald, 5, 2
- 16, Classe_client, 5, 3

17, Classe_librda_locald, 5, 4

18, Resumo_movim_mes_fumo_pronto, 5, 5

19, Estoq_fumo, 5, 6

20, Trans_estoq_fumo, 5, 7

Regras - POSIÇÃO, NOME, CÓDIGO

1, Regra 1, 1

REFERÊNCIAS BIBLIOGRÁFICAS

- [AVI1998] ÁVILA, Bráulio Coelho. Data Mining. In: **VI ESCOLA REGIONAL DE INFORMÁTICA DA SBC**. Curitiba : Champagnat, 1998. p. 87-106.
- [BER1997] BERRY, Michael J. A.; LINOFF, Gordon. **Data mining techniques**. USA : Wiley Computer Publishing, 1997.
- [CAN1998] CANTU, Marco. **Dominando o Delphi 3 - A Bíblia**. São Paulo : Makron Books, 1998.
- [DAT1991] DATE, C. J. . **Introdução a sistemas de banco de dados**. Rio de Janeiro : Campus, 1991.
- [DAL2000] DALFOVO, Oscar; AMORIM, Sammy Newton . **Quem tem informação é mais competitivo: o uso da informação pelos administradores e empreendedores que obtém vantagem competitiva**. Blumenau : Acadêmica, 2000.
- [FAY1996] FAYYAD, Usama M... [et all]. **Advances in knowledge discovery and data mining**. Mento Park : AAAI : MIT, 1996.
- [FEI1999] FEITEN, Wantoir. **Protótipo de um sistema especialista para análise de crédito de pessoas físicas**. Blumenau: FURB,1999. Trabalho de Conclusão de Curso – Ciências da Computação, Universidade Regional de Blumenau.
- [FIG1998] FIGUEIRA, Rafael Medeiros Andrade. **Miner: um software de inferência de dependências funcionais**. Rio de Janeiro, 1998. Trabalho de Conclusão de Curso – Instituto de Matemática, Universidade Federal do Rio de Janeiro.
- [HAR1988] HARMON, Paul; KING, David. **Sistemas Especialistas**. Rio de Janeiro : Editora Campus, 1988.
- [HEI1995] HEINZLE, Roberto. **Protótipo de uma ferramenta para criação de sistemas especialistas baseados em regras de produção**. Florianópolis : UFSC,

1995. Dissertação de Mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas.

[LEV1988] LEVINE, Robert I.; DRANG, Diane E.; EDELSON, Barry. **Inteligência artificial e sistemas especialistas**. São Paulo: McGraw-Hill, 1988.

[RAB1995] RABUSKE, Renato Antônio. **Inteligência artificial**. Florianópolis: Editora da UFSC, 1995.

[RIB1987] RIBEIRO, Horácio da Cunha e Souza. **Introdução aos sistemas especialistas**. Rio de Janeiro - LTC : Livros Técnicos e Científicos Editora, 1987.

[VIN1995] VINCENT, Oswald Cerícola. **Oracle banco de dados relacional e distribuído**. São Paulo – Makron Books, 1995.