

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

MARCOS PAULO MACHADO

**DINBAYES - COMPONENTES DINÂMICOS EM SISTEMAS ESPECIALISTAS
PROBABILÍSTICOS**

CRICIÚMA, JUNHO DE 2006.

MARCOS PAULO MACHADO

**DINBAYES - COMPONENTES DINÂMICOS EM SISTEMAS ESPECIALISTAS
PROBABILÍSTICOS**

Trabalho de Conclusão de Curso para a Obtenção do
Grau de Bacharel em Ciência da Computação da
Universidade do Extremo Sul Catarinense.

Orientadora: Profa. M.Sc. Priscyla Waleska T. A.
Simões

CRICIÚMA, JUNHO DE 2006.

À minha família, por estar sempre presente, desde o ingresso na universidade, até o seu término, dando todo o apoio que necessitei, e à minha namorada Lara, que compreendeu os momentos em que estive ausente e me incentivou nos momentos em que mais precisei.

AGRADECIMENTOS

Aos meus pais José Paulo e Marilene, a quem sempre desejo ser um motivo de orgulho.

A meu irmão, Felipe, pessoa a quem pretendo servir de inspiração ao longo de sua vida.

À minha namorada Lara, que sempre esteve ao meu lado nos momentos bons e ruins deste longo caminho acadêmico.

Aos meus amigos pelos momentos de apoio e descontração no decorrer do curso.

Aos professores do curso de Ciência da Computação da Unesc, que direta ou indiretamente me auxiliaram na realização deste trabalho.

À minha orientadora Priscyla, que confiou em mim e me forneceu todo o auxílio necessário para a realização e conclusão deste trabalho.

E a todos que de uma forma ou de outra contribuíram para que este sonho se tornasse realidade.

Muito obrigado!

*"Nunca ande pelo caminho traçado,
pois ele conduz somente até onde os
outros foram."*

Alexandre Graham Bell

RESUMO

Esta pesquisa tem como objetivo a realização de estudos sobre a utilização de componentes dinâmicos em Sistemas Especialistas Probabilísticos (SEP), para que possa servir de auxílio, tanto para o engenheiro do conhecimento como para o especialista no domínio de aplicação em suas tomadas de decisão. Na área médica, que apresenta um grau de incerteza elevado no processo de raciocínio clínico, esse tipo de sistema é uma boa alternativa para auxiliar o médico no processo de tomada de decisão. A maioria dos SEP desenvolvidos utiliza base de conhecimento estática, ou seja, uma vez definida esta não poderá ser alterada. O resultado obtido com a realização dessa pesquisa foi um SEP Dinâmico, que integra o ambiente de programação C++ Builder com a API da *shell* Netica, e permite ao usuário diversas funcionalidades, sendo essas divididas em duas ações. Na primeira, o usuário pode realizar a criação de uma nova base de conhecimento, incluindo nós, estados e valores das probabilidades condicionais, alterar estados, valores de probabilidades condicionais dos nós, bem como excluí-los. Na segunda, pode-se carregar uma base de conhecimento já existente, e realizar as mesmas ações disponíveis no processo de criação de uma nova rede. Considerando que a maioria dos SEP desenvolvidos na área médica caem em desuso devido a necessidade de atualização da base de conhecimento, essa pesquisa contribui para uma utilização continuada desses sistemas.

Palavras-chaves: Inteligência Artificial, Redes Bayesianas, Sistemas Especialistas Probabilísticos, Base de Conhecimento Dinâmica.

ABSTRACT

This research has its aim the realization of studies about the utilization of the dynamic components in Probabilistic Expert Systems (PES), so that it can serve as support, as well as for the knowledge engineer as for the specialist in the application domain in the decisions he takes. In the medical area, which brings up a high level degree in the clinical reasoning process, this kind of system is a good alternative to help the doctor in his making decision process. Most of the PES developed make use of static base knowledge, it means, once defined it cannot be changed. The result obtained with this research realization was a dynamic PES, which integrates the environment programming C++ Builder with the *API* of *shell* Netica, and allows the user lots of performing, divided in two actions. In the first one, the user can work on the creation of a new knowledge bases, including nodes, states and values of the conditional probabilities, to change state, values of the conditional probabilities of the nodes, as well as to eliminate them. In the second one, it can already carry a knowledge base on hand, and work the same actions available in the creation process of a new net. Considering that most of the PES developed in the medical area are useless due to the necessity of the knowledge base updating, this research contributes to the continual use of these systems.

Key Words: Artificial Intelligence, Bayesians Nets, Probabilistic Expert System, Dynamic Knowledge Base.

LISTA DE ILUSTRAÇÕES

Figura 1. Áreas de Aplicação de Sistemas Especialistas.....	26
Figura 2. Arquitetura de um Sistema Especialista Probabilístico	27
Figura 3. Interface Gráfica do Netica.....	41
Figura 4. Interface Gráfica do Hugin	44
Figura 5. Interface Gráfica do MSBN.....	46
Figura 6. Interface Gráfica do DXPRESS.....	47
Figura 7. Interface Gráfica do SPIRIT	49
Figura 8. Listagem das avaliações de sintomas do pacientes no SEDACaP.....	52
Figura 9. Tela inicial do Biowoman.....	54
Figura 10. Tela inicial do SEDIN.....	56
Figura 11. Interface do DXPlain	57
Figura 12. Tela inicial do BioWoman.....	70
Figura 13. Inserção de uma nova evidência no BioWoman.....	71
Figura 14. Ações do Sistema.....	77
Figura 15. Realização de Inferências	78
Figura 16. Tela de Construção	80
Figura 17. Comando de criação do novo nó.....	81
Figura 18. Criação dos Edits Dinamicamente	81
Figura 19. Comando para criação dos edits dinamicamente	82
Figura 20. Comando para contagem das colunas.....	82
Figura 21. Comando para atribuição dos nomes dos estados no grid	83
Figura 22. Inserção dos nomes no Grid.....	83
Figura 23. Comando para atribuição dos nomes dos estados do nó.....	84
Figura 24. Inserção das probabilidades condicionais dos estados do nó.....	85
Figura 25. Comando para atribuição das probabilidades condicionais dos estados.....	85
Figura 26. Criação do <i>link</i> entre os nós.....	86
Figura 27. Comando para inserção de links entre os nós	86
Figura 28. Apagando um nó da rede	87
Figura 29. Comando para apagar um nó da rede.....	87
Figura 30. Comando para salvar a rede.....	88
Figura 31. Base de conhecimento do BioWoman	92
Figura 32. Tela de consultas do sistema DIAGLER	107
Figura 33. Prognostico para pessoa do sexo feminino no PROBUCAL.....	109
Figura 34. Tela de suspeitas e análise recordatória do SACI.....	110
Figura 35. Interface do SISPAN.....	112

LISTA DE TABELAS

Tabela 1. Probabilidades Condicionais para uma rede bayesiana	23
Tabela 2. Informações das <i>shells</i>	49

LISTA DE SIGLAS

<i>API</i>	<i>Application Programming Interface</i>
BC	Base de Conhecimento
BOL	Bayes <i>On-Line</i>
CTAI	Centro de Tecnologia em Automação e Informática
DLL	<i>Dinamic Link Library</i>
IA	Inteligência Artificial
PC	Probabilidade Condicional
PUC	Pontifícia Universidade Católica
RB	Rede Bayesiana
RN	Rede Neural
SE	Sistema Especialista
SENAI	Serviço Nacional de Aprendizagem Industrial
SEP	Sistema Especialista Probabilístico
SEPD	Sistema Especialista Probabilístico Dinâmico
UFSC	Universidade Federal de Santa Catarina
UNAMA	Universidade da Amazônia
UNESC	Universidade do Extremo Sul Catarinense

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVO GERAL.....	15
1.2	OBJETIVOS ESPECÍFICOS	15
1.3	JUSTIFICATIVA.....	16
1.4	ESTRUTURA DO TRABALHO.....	17
2	RACIOCÍNIO SOB INCERTEZA	19
2.1	MODELO PROBABILISTA	20
2.2	TEORIA DA PROBABILIDADE BAYESIANA	21
3	SISTEMAS ESPECIALISTAS PROBABILÍSTICOS	24
3.1	HISTÓRICO DOS SISTEMAS ESPECIALISTAS PROBABILÍSTICOS.....	24
3.2	ESTRUTURA DE UM SISTEMA ESPECIALISTA PROBABILÍSTICO	26
3.2.1	Base de Conhecimento	27
3.2.2	Máquina de Inferência	28
3.2.3	Memória de Trabalho	28
3.2.4	Sistema de Justificação	29
3.2.5	Subsistema de Aquisição do Conhecimento.....	29
3.2.6	Interface Com o Usuário.....	30
4	INCLUSÃO DE INFORMAÇÃO EM BASES DE CONHECIMENTO	31
4.1	AQUISIÇÃO DE CONHECIMENTO.....	32
4.1.1	Métodos Não-Informatizados.....	34
4.1.2	Métodos Informatizados	35
4.1.2.1	Semi-Intermediário.....	35
4.1.2.2	Semi-direto	36
4.1.2.3	Direto Supervisionado	36
5	SHELL PARA SISTEMAS ESPECIALISTAS PROBABILÍSTICOS	38
5.1	SHELL NETICA.....	39

5.2	HUGIN EXPERT	41
5.3	BNG.....	44
5.4	MICROSOFT BELIEF NETWORK TOOLS.....	45
5.5	DXPRESS	46
5.6	SPIRIT.....	48
6	ESTADO DA ARTE.....	51
6.1	SISTEMA ESPECIALISTA ON-LINE DE AUXÍLIO AO DIAGNÓSTICO DE CÂNCER DE PRÓSTATA.....	51
6.2	ABS-BAYES: SISTEMA ESPECIALISTA PROBABILÍSTICO PARA APOIO AO DIAGNÓSTICO DE ABSENTEÍSMO.....	53
6.3	BIOWOMAN - BASE DE CONHECIMENTO DINÂMICA PARA SISTEMAS ESPECIALISTAS PROBABILÍSTICOS	53
6.4	SISTEMA ESPECIALISTA PROBABILÍSTICO PARA APOIO AO DIAGNÓSTICO DE POTENCIAL ECONÔMICO – SEPE.....	54
6.5	SISTEMA ESPECIALISTA PROBABILÍSTICO: BASE DE CONHECIMENTO DINÂMICA.....	55
6.6	DXPLAIN	56
7	PROJETO DE INTERFACE COM O USUÁRIO.....	58
7.1	MODELOS DE PROJETOS DE INTERFACES	58
7.2	PADRÕES DE INTERFACE	60
7.3	ERGOLIST.....	62
7.4	USABILIDADE	64
7.4.1	Definições de Usabilidade	65
8	DINBAYES - COMPONENTES DINÂMICOS EM SISTEMAS ESPECIALISTAS PROBABILÍSTICOS	67
8.1	LEVANTAMENTO BIBLIOGRÁFICO	67
8.2	ESTUDO DO BIOWOMAN.....	68
8.3	ESTUDO DAS FUNÇÕES DA API DA SHELL NETICA	71
8.4	PROJETO DA INTERFACE COM O USUÁRIO	72
8.4.1	Organização da Interface	73

8.4.1.1	Presteza.....	74
8.4.1.2	Agrupamento por Localização	74
8.4.1.3	Feedback.....	74
8.4.1.4	Legibilidade.....	74
8.4.1.5	Ações Mínimas.....	75
8.4.1.6	Ações Explícitas do Usuário	75
8.4.1.7	Qualidade das Mensagens	75
8.4.1.8	Consistência.....	76
8.4.2	Ações do Sistema	76
8.4.2.1	Criação de Uma Nova Rede	78
8.4.2.2	Carregar Uma Rede Existente	79
8.4.3	Exemplo de Utilização do Sistema	80
8.5	DEFINIÇÃO DA BASE DE CONHECIMENTO	89
8.5.1	Base de Conhecimento do Biowoman.....	89
	CONCLUSÃO.....	93
	REFERÊNCIAS	95
	BIBLIOGRAFIA RECOMENDADA.....	98
	APÊNDICE	99

1 INTRODUÇÃO

Sistemas Especialistas Probabilísticos (SEP), são implementados para atuarem nas mais diversas áreas, desde a resolução de problemas, como a evasão escolar, a situações de maior complexidade como o diagnóstico de doenças. Problemas desse tipo são de natureza incerta, ou seja, o conhecimento necessário para sua resolução não é conhecido. Uma série de maneiras para tratamento da incerteza, também chamados de formalismos, foram desenvolvidos e continuam sendo um tópico de muita pesquisa em IA, o Raciocínio sob Incerteza.

Em SEP, as informações referentes a probabilidades e como estas são tratadas estão armazenadas na base de conhecimento que pode ser estática ou dinâmica, sendo a última a mais apropriada para o desenvolvimento dessa categoria de sistemas, já que novas informações podem ser inseridas, e informações antigas ou desnecessárias podem ser removidas sem afetar o restante do sistema. Esse é um ponto positivo dos Sistemas Especialistas Probabilísticos Dinâmicos (SEPD), pois é possível avaliar a influência de uma informação em relação as outras já existentes na base de conhecimento.

A presente pesquisa consiste no desenvolvimento de um protótipo de sistema especialista probabilístico com base de conhecimento dinâmica e interface que apresenta a utilização de componentes visuais dinâmicos, criada a partir da integração da DLL da *shell* Netica e o ambiente de programação Borland C++ Builder.

O protótipo do sistema foi desenvolvido a fim de auxiliar profissionais em suas tomadas de decisão, em domínios de conhecimento com a presença de incerteza por aleatoriedade, dando a eles a possibilidade de criar uma nova rede, inserindo os nós, nomes de estados e valores de probabilidades condicionais, ou carregar uma rede já

existente permitindo que o usuário altere nomes e valores das probabilidades condicionais, insira e/ou remova nós conforme necessário.

1.1 OBJETIVO GERAL

Desenvolver um protótipo de sistema especialista probabilístico, com componentes dinâmicos na Base de Conhecimento e interface, por meio da integração da API da *shell* Netica com o ambiente de desenvolvimento C++ Builder.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos dessa pesquisa são:

- a) compreender a estrutura e funcionamento de sistemas especialistas probabilísticos;
- b) permitir a inclusão, exclusão e alteração de evidências em Sistemas Especialistas Probabilísticos (SEP) via API da *shell* Netica;
- c) realizar a integração da *shell* Netica com o ambiente de desenvolvimento Borland C++ Builder;
- d) disponibilizar documentação a respeito das funções da API da *shell* Netica para inclusão, exclusão e alteração de informações em uma Base de Conhecimento, bem como da função de carregar uma rede já existente;
- e) aplicar o desenvolvimento do SEP dinâmico em uma problemática da área da saúde;

- f) oferecer uma interface dinâmica com o usuário que utilize componentes visuais interagindo com a Base de Conhecimento dinâmica.

1.3 JUSTIFICATIVA

A cada ano que passa, há uma rápida e crescente evolução da tecnologia e do número de informações transmitidas em diversas áreas, e com a Inteligência Artificial não é diferente, pois cada vez mais se torna necessário a realização de estudos em novas áreas de conhecimento para que se possa cumprir as exigências da demanda tecnológica.

Além da realização de estudos, é preciso ter uma visão empreendedora, pois investir em tecnologias para serem utilizadas a curto prazo não é vantajoso. Na área de IA, com os sistemas especialistas probabilísticos é possível exemplificar e compreender essa afirmação.

Os SEP apresentam em sua estrutura uma base de conhecimento para realizarem inferências e assim, fornecerem suas probabilidades, e conseqüentemente diagnóstico do sistema. Geralmente, nos SEP a base de conhecimento é estática, ou seja, uma vez construída não poderá mais ser modificada. Porém, pode-se adotar a área da saúde como exemplo, onde em um curto espaço de tempo surgem novas doenças, e informações novas necessitam ser levadas em consideração no momento de um diagnóstico. Caso a BC seja estática, essas informações não podem ser aproveitadas pois ela não possibilita a sua inclusão.

Justifica-se este trabalho, devido a dificuldade em encontrar SEP onde seja possível realizar a manipulação de informações da BC, ou seja, incluir uma evidência,

alterar probabilidades condicionais e remover valores da base de conhecimento, por meio de uma interface de fácil compreensão e utilização.

A relevância para realização deste estudo está na possibilidade de criação de um protótipo de SEPD, com o intuito de auxiliar usuários na manipulação da BC, diminuindo assim a necessidade de manutenções constantes no SEP por parte do Engenheiro do Conhecimento.

1.4 ESTRUTURA DO TRABALHO

O presente trabalho é composto de sete capítulos, onde são abordados tópicos relacionados ao desenvolvimento desta pesquisa, destacando-se Sistemas Especialistas Probabilísticos (SEP), Inteligência Artificial (IA), Raciocínio Sob Incerteza, Inclusão de informações em Bases de Conhecimento, entre outros. Por fim, são realizadas as considerações finais e sugestões para trabalhos futuros.

No capítulo 2, Raciocínio Sob Incerteza, comenta-se as formas de representação do conhecimento em SEP, ou seja, a forma como será tratada a incerteza, destacando-se o modelo probabilístico, utilizado nessa pesquisa.

No capítulo 3, sobre Sistemas Especialistas Probabilísticos, é descrita sua definição, histórico, bem como sua estrutura e algumas áreas onde podem ser aplicados.

Em seguida, o capítulo 4, Inclusão de Informação em Bases de Conhecimento, apresenta algumas técnicas de aquisição de conhecimento, conceituando os métodos informatizados e não-informatizados.

Já no capítulo 5, são descritas algumas *shells* para Sistemas Especialistas Probabilísticos, suas características e funções disponíveis.

No Estado da Arte, capítulo 6, encontram-se alguns exemplos de sistemas especialistas probabilísticos desenvolvidos.

No capítulo 7, que aborda o projeto de interface com o usuário, são apresentadas algumas informações sobre o desenvolvimento da interface com o usuário, seus modelos e formas de padronização.

Por fim no capítulo 8, Componentes Dinâmicos em Sistemas Especialistas Probabilísticos, são apresentadas as etapas metodológicas de desenvolvimento do protótipo do sistema, que são finalizadas com a implementação desse e elaboração da documentação dos componentes dinâmicos disponíveis pela *API* da *shell* Netica.

As considerações finais abordam a conclusão dessa pesquisa e algumas sugestões para trabalhos futuros.

2 RACIOCÍNIO SOB INCERTEZA

Com o intuito de realizar a modelagem do conhecimento especialista em um sistema especialista probabilístico, existem diversos formalismos que podem ser adotados, ou seja, a forma como será tratada a incerteza. O formalismo escolhido deve ser bem analisado, pois afetará todo o desenvolvimento do sistema posteriormente.

Não existe qualquer tipo de método que auxilie o engenheiro do conhecimento na escolha do formalismo. O que pode auxiliá-lo são apenas critérios subjetivos de avaliação. A representação e manipulação do conhecimento são tarefas complexas que vem sendo estudadas ao longo dos anos para que possam ser dominadas completamente, e assim, serem melhor utilizadas (BARONE, 2003).

A imperfeição da informação em sistemas baseados em conhecimento é conhecida como *incerteza*. Esse termo, no entanto, é muito restritivo. O que se convencionou chamar de tratamento da incerteza pode na verdade, endereçar outras imperfeições, como imprecisão, possibilidade, probabilidade, entre outras. Eis alguns exemplos descritos por Barreto (2001), para demonstrar esses termos:

- a) informação perfeita: A nota do aluno no trabalho final foi 9,03.
- b) informação imprecisa: A nota do aluno no trabalho final foi entre 9 e 10.
- c) informação incerta: Eu acho que a nota do aluno no trabalho final foi 9,03 (mas não tenho certeza).
- d) informação probabilista: É provável que a nota do aluno no trabalho final tenha sido 9.
- e) informação possibilista: É possível que a nota do aluno no trabalho final tenha sido 9.

As informações que obtém-se podem variar de perfeita, quando se descobre exatamente o que se quer saber, a completamente imperfeita, ou seja, ausência total de informação, ou pela ocorrência de informações conflitantes. Mesmo lidando com esses tipos de informações é possível tomar decisões razoáveis para solucionar nossos problemas. Isso também ocorre, ou deveria ocorrer nos sistemas baseados em conhecimento (BARRETO, 2001).

Para cada tipo de informação descrito, existe um modelo formal para tratá-la. A informação probabilística pode ser tratada pela teoria de probabilidades ou pela teoria da evidência (também conhecida como Dempster-Shafer). Já a informação imprecisa e/ou vaga pode ser tratada pela teoria dos conjuntos nebulosos, pela teoria dos conjuntos de aproximação ou pela manipulação de classes de referência. A informação incerta pode ser tratada tanto pela teoria de probabilidades, possibilidades ou evidência (BARRETO, 2001).

A seguir será descrito o modelo de raciocínio probabilístico.

2.1 MODELO PROBABILISTA

É talvez o mais antigo que trata com mecanismos de incerteza. É aquele que, apoiando-se em informações probabilísticas sobre fatos de um domínio chega a uma conclusão a respeito de um novo fato, conclusão esta associada a uma probabilidade (SAVARIS, 2002).

A teoria da probabilidade teve início com os jogos de azar, como roleta e cartas. Representa a incerteza por aleatoriedade, isto é, não se pode prever com toda certeza o que acontecerá num novo caso, mesmo diante do conhecimento de casos anteriores. A probabilidade de um evento ocorrer assume valor de 0 a 1, sendo estas

probabilidades caracterizadas por funções de distribuição de probabilidade. Nestes sistemas é considerada também a probabilidade de que ocorra um evento B (“conseqüência”) condicionada a ocorrência de um outro evento A (“causa”) (NASSAR, 2005).

Nos SEP os valores de probabilidade refletem a crença do especialista sobre o que ele espera que ocorra em situações similares àquelas que tem experienciado e aprendido, isto é, o especialista está tentando extrapolar com base em sua experiência e aprendizado no domínio de aplicação (NASSAR, 2005).

Uma importante meta de muitos sistemas especialistas probabilísticos é coletar evidências no decorrer do sistema e modificar o comportamento deste, com base nessas evidências. Com o objetivo de moldar este comportamento, é preciso uma teoria estatística de evidências, sendo a teoria bayesiana uma delas (RICH; KNIGHT, 1993).

2.2 TEORIA DA PROBABILIDADE BAYESIANA

O teorema de Bayes ou regra de Bayes é a base de todos os sistemas modernos de IA para inferência probabilística (LUGER, 2002). Sua fórmula é escrita assim:

$$P(H_i | E) = \frac{P(E | H_i) * P(H_i)}{\sum P(E | H_n) * P(H_n)}$$

Onde:

$P(H_i | E)$ = a probabilidade da hipótese H_i ser verdadeira, dada a evidência E ,

$P(E | H_i)$ = a probabilidade de observar a evidência E , dado que a hipótese H_i é verdadeira,

$P(H_i)$ = a probabilidade a priori de H_i ser verdadeira na ausência de evidências específicas.

O raciocínio bayesiano é baseado na teoria das probabilidades, sendo usado em várias áreas, incluindo reconhecimento de padrões e classificação (LUGER, 2002).

A regra de Bayes exige três termos, uma probabilidade condicional e duas probabilidades incondicionais. Ela é muito útil na prática, pois é possível fazer boas estimativas de probabilidades desses três números e assim encontrar o quarto. No caso da construção de um SEP para a área médica, esse tipo de regra se torna útil para a escolha de um diagnóstico por meio dos sintomas (RUSSEL; NORVIG, 2004).

As estatísticas bayesianas são base para um sistema de raciocínio incerto. Porém, segundo Rick e Knight (1993), quando o número de probabilidades conjuntas é muito grande, o teorema de Bayes torna-se impraticável por vários motivos:

- a) as pessoas são estimadores de probabilidades muito fracos;
- b) o espaço necessário para armazenar todas as probabilidades cresce proporcionalmente;
- c) o tempo exigido para computar todas as probabilidades é muito grande.

Devido a essas desvantagens, faz-se necessário um estudo do problema que se deseja resolver com o teorema de Bayes a fim de saber se é a solução mais viável.

A seguir, na Tabela 1, temos um exemplo de probabilidades condicionais para uma rede bayesiana:

Tabela 1. Probabilidades Condicionais para uma rede bayesiana

Atributo	Probabilidade
$P(\acute{U}mido \mid Irrigador, Chuva)$	0,95
$P(\acute{U}mido \mid Irrigador, \neg Chuva)$	0,9
$P(\acute{U}mido \mid \neg Irrigador, Chuva)$	0,8
$P(\acute{U}mido \mid \neg Irrigador, \neg Chuva)$	0,1
$P(Irrigador \mid Estação Chuvosa)$	0,0
$P(Irrigador \mid \neg Estação Chuvosa)$	1,0
$P(Chuva \mid Estação Chuvosa)$	0,9
$P(Chuva \mid \neg Estação Chuvosa)$	0,1
$P(Estação Chuvosa)$	0,5

Fonte: RICH, E; KNIGHT, K (1993)

A probabilidade prévia de estação chuvosa é 0,5. Então, se estiver na estação chuvosa, a probabilidade de chuva em uma noite qualquer é 0,9; se não, a probabilidade é de apenas 0,1.

Agora que já é possível compreender um pouco sobre os formalismos existentes para tratamento de incerteza, podem ser conhecidos os SE, onde o formalismo escolhido é utilizado.

3 SISTEMAS ESPECIALISTAS PROBABILÍSTICOS

Sistemas especialistas probabilísticos são programas de inferência sobre uma base de conhecimento específicos, coletados de um especialista na área de aplicação (RODRIGUES, 2002).

Estes sistemas têm em sua base de conhecimento fatos e regras que representam o conhecimento do especialista no domínio de aplicação. Aos fatos e regras é associada a incerteza presente no domínio e são explicitadas as crenças por meio de valores de probabilidade (RAMOS; AZEVEDO; LIMA apud TESSARI, 2003).

O raciocínio realizado pelo sistema deve considerar estas probabilidades para a partir dos dados de entrada (*input*) associar um vetor de probabilidades ao conjunto de hipóteses diagnósticas (*output*). A hipótese com maior probabilidade de ocorrência pode ser considerada a conclusão do sistema, notando-se que a esta conclusão está associado o grau de certeza da resposta do sistema (NASSAR, 2005).

De forma a desenvolver esses tipos de sistemas, são considerados os seguintes domínios de conhecimento: inteligência artificial, formas de tratamento da incerteza, sistemas especialistas e redes bayesianas.

3.1 HISTÓRICO DOS SISTEMAS ESPECIALISTAS PROBABILÍSTICOS

Em 1963, após vários anos de desenvolvimento, Allen Newel e Herbert A. Simon apresentaram o General Problem Solver (GPS), cujo objetivo era simular o pensamento humano e solucionar problemas em geral. Contudo, a comunidade interessada em IA percebeu que era necessário restringir o domínio de dados, para que

os sistemas fossem úteis. Assim, nasceram os sistemas especialistas (BARRETO, 2001).

No início dos estudos sobre sistema especialistas, o primeiro sistema a ser considerado especialista foi o DENDRAL em 1965. Ele podia prever as estruturas de compostos químicos desconhecidos baseado em análises de rotinas, porém sua utilização ficou destinada apenas ao meio acadêmico. Em 1976, desenvolveu-se o SE mais conhecido, o MYCIN. Este deve ser destacado por ser o primeiro a utilizar fatores de certeza, utilizando regras derivadas do domínio médico para raciocinar (deduzir) a partir de uma lista de sintomas de alguma doença em particular, mas assim como o DENDRAL também não saiu do meio acadêmico (BARRETO, 2001).

No final da década de 70, Peter Hart e Richard Duda, criaram o PROSPECTOR. Foram introduzidas muitas idéias do MYCIN, incorporando uma interface chamada *Lifer* para permitir a comunicação em linguagem técnica de geologia. Finalmente, com o desenvolvimento do XCON em 1982, os sistemas especialistas ganharam interesse das fábricas (FERNANDES, 2003).

A partir dos anos 80, quando os SE ganharam interesse das fábricas e deixaram de ser basicamente aplicativos de estudos acadêmicos, foi necessário ampliar mais sua área de abrangência. Apenas a utilização de fatores de certeza já não era o bastante e com isso iniciaram-se estudos sobre os fatores de incerteza e de probabilidade para a resolução de problemas, caracterizando assim os Sistemas Especialistas Probabilísticos (BARRETO, 2001).

Geralmente, os SEP assim como os SE possuem uma base de conhecimento estática¹. Com o grande crescimento do fluxo de informações existentes e o surgimento de novas características nas mais diversas áreas, faz-se necessário a utilização de bases

¹ Base de conhecimentos que não poderá ser ampliada ou atualizada.

de conhecimento dinâmicas, para que assim como nos sistemas convencionais, possa haver inserção, exclusão e alteração dos dados existentes, afim de tornar o sistema o mais atualizado e preciso possível. Surgem então os Sistemas Especialistas Probabilísticos Dinâmicos (SEPD), tipo de SEP que será abordado nessa pesquisa (RUSSEL; NORVIG, 2004).

Segundo Rabuske (1995), a principal razão do interesse em interação entre IA e bases de dados é o aumento da produtividade e funcionalidade dos sistemas.

Na Figura 1 pode-se observar algumas áreas de aplicação dos sistemas especialistas.

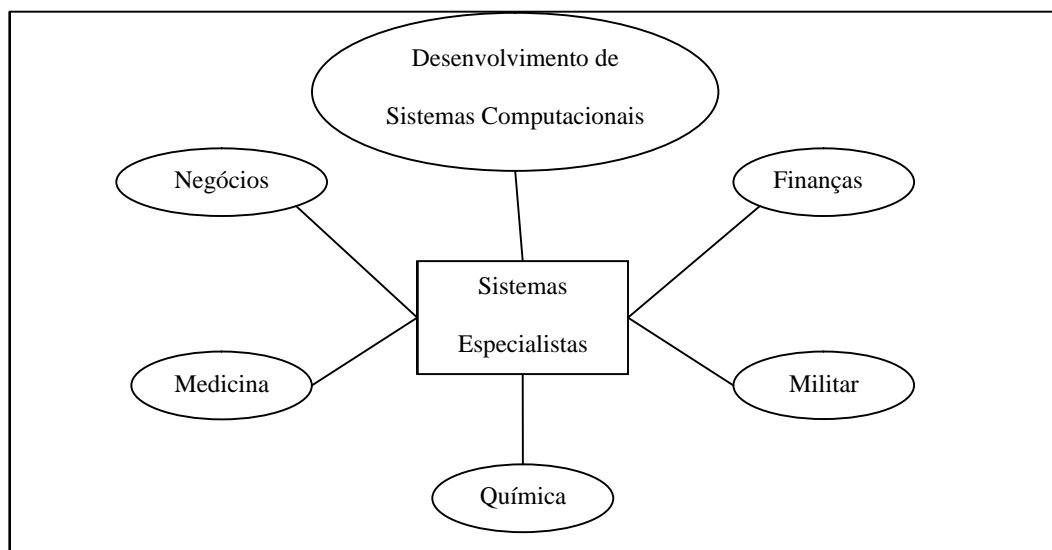


Figura 1. Áreas de Aplicação de Sistemas Especialistas
Fonte: NASSAR, S. (2005)

3.2 ESTRUTURA DE UM SISTEMA ESPECIALISTA PROBABILÍSTICO

Quanto a estrutura de um SEP pode ser feita uma analogia ao esqueleto humano, onde cada componente tem sua função específica e de suma importância para o bom funcionamento do sistema.

Os sistemas especialistas probabilísticos são formados pelos seguintes componentes:

- a) base de conhecimento;
- b) máquina de inferência;
- c) memória de trabalho;
- d) subsistema de explicação;
- e) subsistema de aquisição do conhecimento;
- f) interface com o usuário.

Na Figura 2, tem-se a arquitetura de um SEP.

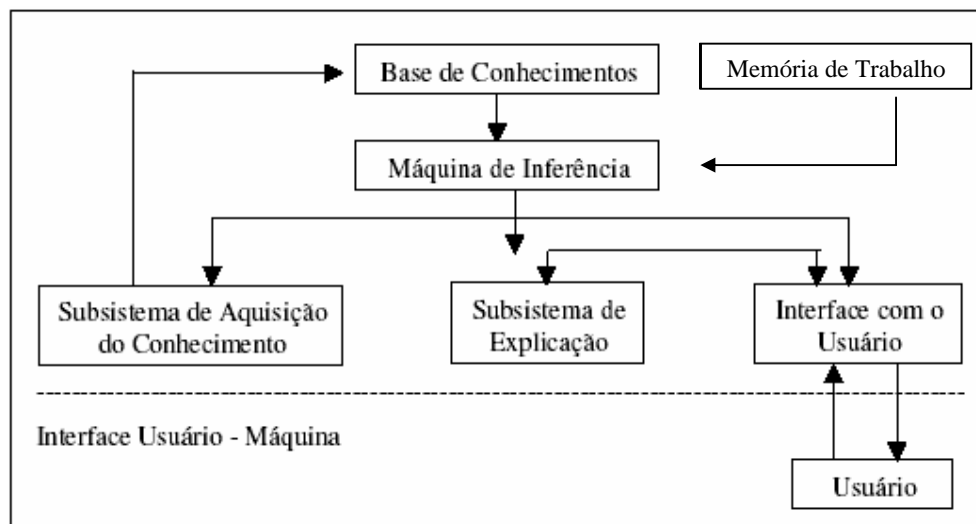


Figura 2. Arquitetura de um Sistema Especialista Probabilístico
 Fonte: Adaptado de RABUSKE, I. (1995)

3.2.1 Base de Conhecimento

A base de conhecimento é o local onde está localizado o conhecimento do especialista. Essa base pode ser construída de duas formas, estática ou dinâmica. A base de dados estática, uma vez construída não poderá mais ser alterada, o que não é vantajoso pelo fato de novos conhecimentos adquiridos pelo usuário não poderem ser

inseridos. Já a base dinâmica difere-se do outro modelo justamente nesse detalhe. Por ser dinâmica, novos conhecimentos do especialista podem ser acrescentados posteriormente, o que caracteriza o aprendizado do SEP. Os dados são acrescentados sem necessidade de modificações no sistema, já que a base de conhecimento é separada da máquina de inferência (BITTENCOURT, 2001).

3.2.2 Máquina de Inferência

Máquina de inferência é a parte do sistema especialista probabilístico responsável pela comparação entre os dados da memória de trabalho com as probabilidades da base de conhecimento, ou seja, realiza o raciocínio do SEP. Após a comparação, mostra novas informações ou gera uma conclusão.

A máquina de inferência envolve a manipulação das incertezas, como lógica bayesiana. Na maioria dos casos pode-se adotar softwares para a construção da máquina de inferência. Nos casos onde isso não é possível faz-se necessário sua elaboração (RABUSKE, 1995).

3.2.3 Memória de Trabalho

A memória de trabalho é o conhecimento do especialista em um domínio de conhecimento, abrangendo todas as etapas do mesmo na resolução de um problema e a maneira com que este é abordado. O conhecimento do especialista do domínio é a síntese do que foi adquirido com a experiência e pesquisas (BITTENCOURT, 2001).

3.2.4 Sistema de Justificação

É o mecanismo responsável pela explicação das ações do sistema especialista na resolução de determinado problema, sendo um requisito obrigatório nos sistemas especialistas probabilísticos. Assim como o usuário responde as perguntas propostas no sistema, o sistema também deve responder perguntas do usuário, caso este solicite. Assim, caso sejam solicitadas informações adicionais sobre algum item da pesquisa o usuário pode perguntar “Por Quê?” e o sistema responderá o motivo. Se o usuário não concordar com o resultado mostrado pelo sistema ele pode perguntar “Como?” e o raciocínio utilizado será mostrado passo a passo (BARRETO, 2001).

3.2.5 Subsistema de Aquisição do Conhecimento

Alimenta a base de conhecimento e possibilita a inclusão de novos conhecimentos e alteração ou eliminação de conhecimentos antigos (MANARIN, 2004).

Aparece, geralmente, munido de recursos para trabalhar o conhecimento que auxiliam na difícil tarefa de extraí-lo e aproveitá-lo adequadamente. Em muitos sistemas é a única forma de aprendizado, sendo sem dúvida, a parte mais crítica do sistema especialista (MANARIN, 2004).

A tarefa de ampliação e atualização da base de conhecimento é feita pelo engenheiro do conhecimento, por meio de estudos de casos, pesquisas e entrevistas com o especialista, para esclarecer o seu conhecimento na área e assim traduzí-lo em probabilidades (RABUSKE, 2001).

3.2.6 Interface Com o Usuário

A interface com o usuário é um ponto que deve ser bem planejado e elaborado nos SEP, sendo de fácil utilização e compreensão por parte do usuário, desde os mais leigos até os mais experientes (NASSAR, 2005).

Com a conclusão desse capítulo é possível entender um pouco sobre os SEP e sua estrutura. No capítulo a seguir, serão abordadas algumas formas de inclusão de informação em bases de conhecimento.

4 INCLUSÃO DE INFORMAÇÃO EM BASES DE CONHECIMENTO

Com o intuito de ser uma ferramenta eficaz e podendo ser utilizado por longos períodos de tempo, o sistema especialista probabilístico precisa interagir com o usuário. Dois recursos são importantes para garantir essa interação: adquirir conhecimento novo e modificar o existente; e também ser capaz de explicar seu raciocínio (RICH; KNIGHT, 1993).

Primeiramente, ele deve ser capaz de receber novos conhecimentos e modificar os existentes, se necessário. Nos SEPD esse recurso está presente e deve funcionar da melhor maneira possível, fazendo-se necessário disponibilizar ao usuário a maior quantidade de opções, para que se molde a base de conhecimento à sua maneira e consiga deixá-la mais precisa e atualizada possível. Essa é uma das formas de modelar o conhecimento do especialista humano para a máquina (RICH; KNIGHT, 1993).

Outro recurso necessário é a explicação do raciocínio seguido para a resolução do problema. Geralmente, os usuários não concordam totalmente com respostas e explicações que lhes são dadas e nos SEP isso não é diferente. O exemplo do médico consegue explicar claramente essa questão, pois dentro de pouco tempo os profissionais da saúde utilizarão com grande frequência os SEP para auxiliá-los na resolução de problemas. Por se tratar de uma área de risco eminente o profissional pode, eventualmente, pesquisar o raciocínio seguido pelo sistema (RICH; KNIGHT, 1993).

No próximo item será abordado o processo de aquisição do conhecimento e suas formas de aquisição, seja informatizada ou não.

4.1 AQUISIÇÃO DE CONHECIMENTO

A parte mais sensível e complexa no desenvolvimento da base de conhecimento de um SEP é a aquisição de conhecimento, por tratar da interação entre o especialista no domínio de aplicação e o engenheiro do conhecimento. Esta não pode limitar-se a adição de novos elementos de conhecimento à base de conhecimento, é necessário integrar o novo conhecimento ao já disponível, por meio da definição de relações entre os elementos que constituem o novo conhecimento e os elementos já armazenados na base (BITTENCOURT, 2001; NASSAR, 2005).

No processo de aquisição de conhecimento é importante realizar tratamento de incoerências.

Outro ponto importante na aquisição de conhecimento é o tratamento de incoerências. Dependendo da forma como o novo conhecimento é adquirido, pode haver erros de aquisição. Técnicas foram desenvolvidas para evitar esses erros, como, por exemplo, a especificação de regras de aquisição em que o tipo de conhecimento esperado é definido (BITTENCOURT, 2001, p. 257).

Já que os SEP envolvem grande esforço do especialista e geralmente são muito custosos, devido ao tempo gasto em estudos e testes na sua elaboração e dependendo de sua proporção, o capital disponibilizado para pesquisas, foram desenvolvidas uma série de diretrizes para determinar se um problema é capaz de ser solucionado com a sua utilização, conforme afirma Luger (2002):

- a) a necessidade de uma solução justifica o custo e o esforço de construir um sistema especialista;
- b) não se dispõe de perícia humana em todas as situações onde seja necessário;
- c) o problema pode ser resolvido usando raciocínio simbólico;

- d) O domínio do problema é bem estruturado e não requer raciocínio de senso;
- e) o problema não pode ser resolvido usando métodos de computação tradicionais;
- f) existem especialistas cooperativos e articulados;

Um SEP para diagnóstico de uma doença específica, se encaixa perfeitamente nessas diretrizes.

A construção da base de conhecimento de um sistema especialista envolve profissionais de pelo menos três áreas de conhecimento, o engenheiro do conhecimento que será o responsável pela escolha de software e hardware necessários para a criação do sistema. O especialista no domínio de aplicação que tem a função de fornecer as informações e passos necessários na resolução do problema, para que o engenheiro do conhecimento possa implementá-las. Por fim aparece a figura do usuário final, que determinará as suas necessidades e informará se está satisfeito ou não com o projeto (LUGER, 2002).

Faz-se necessário entender alguns pontos da base de conhecimento antes de iniciar o processo de aquisição do conhecimento:

A base de conhecimento de um sistema especialista probabilístico é chamada de Rede Bayesiana, nela o conhecimento é representado esquematicamente na forma de um grafo acíclico direcionado. Os nós representam dois tipos de variáveis: as variáveis de entrada (sinais ou evidências) e a variável de saída (conjunto de hipóteses diagnósticas). A força da ligação entre as variáveis é expressa no valor das probabilidades condicionais, que representam uma relação causal para a regra "Se A então B". A arquitetura da rede bayesiana é chamada de parte qualitativa da base de conhecimento. E os valores das probabilidades constituem a parte quantitativa da base de conhecimento (NASSAR, 2005, p 37).

Agora que já se tem idéia do funcionamento do processo de aquisição do conhecimento serão mostradas as formas de aplicá-lo. Segundo Rabuske (1995) os

métodos de aquisição do conhecimento são divididos em dois grupos: informatizados e não-informatizados.

4.1.1 Métodos Não-Informatizados

Estes métodos têm, de modo geral, sua aplicação voltada para o engenheiro do conhecimento, que fica de intermediário entre o sistema e o especialista no domínio.

Os métodos não informatizados de aquisição do conhecimento subdividem-se, basicamente em quatro classes:

- a) **entrevista**: é a forma mais tradicional de adquirir conhecimento, envolvendo perguntas, respostas e discussões entre o engenheiro do conhecimento e o especialista, que serão o ponto de partida para a construção da base de conhecimento;
- b) **brainwriting**: consiste em uma espécie de reunião em torno de uma mesa, onde um grupo de pessoas está disposto a emitir idéias sobre determinado assunto. No centro da mesa existem folhas com o tema a ser discutido e espaço para que cada pessoa escreva suas idéias a respeito. Cada participante pegará uma folha, realizará uma leitura acrescentando seu raciocínio, em seguida devolvendo a folha ao centro da mesa. O processo continua até que todos tenham visto as folhas;
- c) **análise de protocolos**: é um método em que o especialista é observado durante a realização de seu trabalho, para que o engenheiro do conhecimento possa coletar informações específicas sobre sua atuação, para isso, o especialista efetua seu raciocínio em voz alta para que possa ser acompanhado e registrado

d) **observação direta:** trata-se de uma generalização dos métodos dados, consistindo na observação do especialista atuando em seu ambiente de trabalho e fazendo registros necessários de diversas maneiras. A informação obtida é trabalhada manual ou “computacionalmente”, gerando lentamente a base de conhecimento. Esse método é eficaz quanto a quantidade de informação, porém necessita de muito tempo para ser realizado.

Os métodos não-informatizados funcionam perfeitamente, porém a crescente utilização do computador e a necessidade de maior velocidade nas mais variadas ações, fizeram com que a participação do engenheiro do conhecimento fosse sendo reduzida, dando lugar ao computador. A seguir são mostradas algumas técnicas informatizadas para a realização deste processo.

4.1.2 Métodos Informatizados

Dentre os métodos informatizados para a aquisição de conhecimento, tem-se: semi-intermediário, semi-direto e direto supervisionado. A seguir serão mostrados esses métodos que realizam a aquisição do conhecimento com o auxílio de ferramentas computacionais.

4.1.2.1 Semi-Intermediário

Neste método, o engenheiro do conhecimento é auxiliado por ferramentas computacionais para a aquisição do conhecimento de forma a auxiliá-lo neste processo.

Tais ferramentas permitem ao engenheiro do conhecimento executar os procedimentos necessários de forma mais eficiente e/ou efetiva (PEREIRA, 2004 *apud* TURBAN).

4.1.2.2 Semi-direto

No método semi-direto, parte do trabalho do engenheiro do conhecimento é realizada de forma automática por meio de ferramentas utilizadas pelo(s) especialista(s), também conhecidas como *shells*, sendo que estas ferramentas interagem com o especialista para aquisição de conhecimento para a base (PEREIRA, 2004 *apud* TURBAN).

Tais ferramentas requerem treinamento dos especialistas não somente para a sua utilização, mas também no processo de aquisição de conhecimento.

4.1.2.3 Direto Supervisionado

No método direto supervisionado, a aquisição do conhecimento é realizada de forma automatizada por meio de ferramentas que interagem com o especialista, sem a necessidade de participação do engenheiro do conhecimento para ajudar na codificação de conhecimento para a base, no entanto, tal método precisa ser supervisionado pelo engenheiro de conhecimento para a validação do conhecimento adquirido (PEREIRA, 2004 *apud* TURBAN).

Exemplos de sistemas que podem adquirir conhecimento de forma automática são os sistemas que por meio da análise de casos, como diagnósticos realizados pelos médicos de uma mesma área, e avaliações técnicas, conseguem

identificar padrões, para que futuros casos inseridos no sistema possam ser avaliados com a utilização desses padrões.

Após o engenheiro do conhecimento passar algum tempo com o especialista e obter uma visão geral do domínio, utilizando qualquer das técnicas descritas anteriormente, ele já está apto a iniciar o processo de construção do projeto do sistema, determinando a melhor maneira de representar o conhecimento e assim projetar a interface com o usuário (BITTENCOURT, 2001).

Esse protótipo passará por uma série de testes, por meio de problemas menores na área de domínio. O especialista e o engenheiro do conhecimento fornecem problemas para que ele solucione e corrigem erros existentes. Se tudo estiver de acordo, o protótipo pode ser aprimorado até se tornar o sistema final (BITTENCOURT, 2001).

Com a conclusão desse capítulo já é possível identificar alguns dos métodos de aquisição do conhecimento informatizados e não-informatizados. No próximo capítulo será feita uma introdução sobre as ferramentas que auxiliam na construção da base de conhecimento, também conhecidas como *shells*.

5 SHELL PARA SISTEMAS ESPECIALISTAS PROBABILÍSTICOS

A construção de Sistemas Especialistas Probabilísticos envolve investimentos em profissionais de diversas áreas, implicando em riscos maiores para a produção. Uma solução para este problema é o uso de ferramentas capazes de prototipar, avaliar e implementar o projeto de um sistema, as chamadas *shell*. Procura-se, assim, diminuir a necessidade de recursos, entre eles, tempo envolvido, e tenta-se conciliar a produtividade e a versatilidade de ferramentas qualificadas (SAVARIS, 2002).

Uma *shell* aplicada à implementação de bases de conhecimento pode fazer com que o desenvolvedor não se preocupe com a máquina de inferência, mas apenas com a distribuição de probabilidades que modelam o conhecimento desejado (SAVARIS, 2002).

Normalmente uma *shell* inclui:

- a) ferramentas para introduzir conhecimento na base de conhecimento;
- b) uma ou várias máquinas de inferência;
- c) mecanismos para tratar incertezas;
- d) mecanismos para seguir passo a passo o funcionamento do programa de modo a encontrar possíveis erros;
- e) uma comunicação homem-máquina.

A seguir serão mostrados alguns exemplos de *shells* para sistemas especialistas probabilísticos, dando ênfase a *shell* Netica, a qual foi utilizada na realização dessa pesquisa.

5.1 SHELL NETICA

A *shell* Netica é uma ferramenta que proporciona ao engenheiro do conhecimento a modelagem e manipulação da base de conhecimento, facilitando o desenvolvimento de sistemas especialistas probabilísticos (NETICA, 2005).

Desenvolvida pela empresa *Norsys Software Corporation* de Vancouver, Canadá, o Netica utiliza-se de redes bayesianas para encontrar o resultado do sistema. O usuário insere os valores de entrada e a partir deles é criado o resultado em porcentagem de todas as variáveis relacionadas no sistema, ou seja, determinando a probabilidade de algo acontecer de acordo com os valores iniciais. O usuário final de sistemas implementados nesta *shell*, tem uma apresentação gráfica permitindo que de acordo com as variáveis de entrada selecionadas, seja apontado o valor (relação) probabilístico da mesma acontecer, nos valores de saída (ROSA, 2002).

O programa Netica é composto do *Netica Application* e *Netica API*. O *Netica Application* é a interface gráfica que permite visualizar a base de conhecimento em forma de rede. O *Netica API* é a biblioteca do programa com funções escritas em C. Existe a possibilidade de integração entre o *Netica API* e alguns aplicativos como o Borland C++ Builder, utilizado para criar a interface do usuário, sendo possível inserir dados na base de conhecimento por meio dessas bibliotecas.

O funcionamento da *shell* acontece da seguinte maneira:

Dado um novo caso, que o usuário tem conhecimento limitado, Netica encontrará os valores ou probabilidades apropriadas para todas as variáveis desconhecidas. O caso pode ser, convenientemente, salvo em um arquivo e depois incluído dentro da rede (ou em uma rede diferente) para incrementar a consulta, ou para trazer uma nova informação sobre o caso. Netica pode usar diagramas de influência para encontrar as decisões ótimas, que maximizam os valores esperados das variáveis especificadas. Variáveis que não são de interesse por muito tempo podem ser removidas sem uma mudança geral entre os relacionamentos e entre as variáveis remanescentes (NASSAR, 2005, p 29).

Algumas vantagens da *shell* Netica, segundo Nassar (2005) são:

- a) gera uma apresentação gráfica de qualidade, que pode ser incorporada dentro de outros documentos;
- b) pode aprender relações probabilísticas por meio de dados;
- c) permite atualização fácil da rede de crença e dos diagramas de influência, incluindo: excluir, colar e duplicar nós da rede de crença e dos diagramas de influência;
- d) permite comentários;
- e) permite desfazer e refazer comando ilimitadamente;
- f) permite a entrada de relações probabilísticas por meio de equações, com uma extensa biblioteca de funções probabilísticas e funções matemáticas;
- g) permite integração com a linguagem C++.

A *shell* Netica está disponível para download gratuito no site <<http://www.norsys.com>> para as plataformas Windows 9x, 2000 e XP, Windows NT e Macintosh (NORSYS, 2006).

A seguir, na Figura 3 é mostrada a interface do Netica.

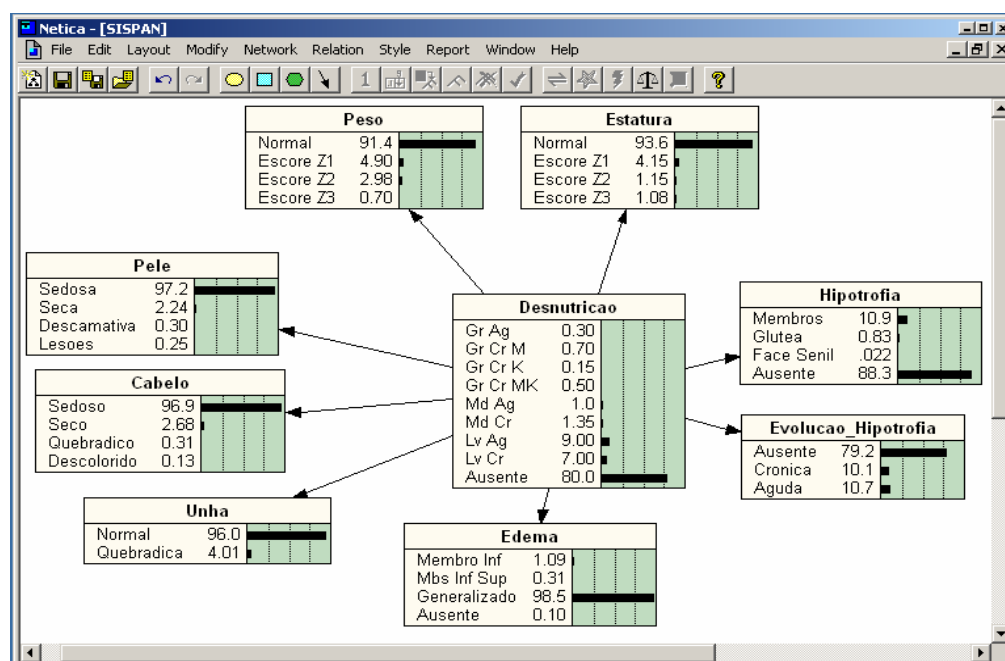


Figura 3. Interface Gráfica do Netica
Fonte: KOËHLER, C (1998)

5.2 HUGIN EXPERT

O *Hugin* possui como objetivo principal a construção de sistemas de apoio a decisão, baseando-se nas RB, e nos diagramas de influência (entendido como uma extensão das RB).

Referindo-se a estrutura da rede, o *Hugin* utiliza dois algoritmos para a aprendizagem: o PC e o NPC, que foi desenvolvido para solucionar alguns problemas do PC, e para estimar as probabilidades condicionais da RB a partir de bases de dados que possuam campos com valores ausentes, utiliza o método Maximização da

Expectativa (EM), método utilizado para estimar dados ausentes em base de dados (HUGIN, 2005).

A *shell* foi desenvolvida pela Hugin Expert A/S na Universidade de Aalborg, Dinamarca no ano de 1995, e divide-se em três partes: máquina de inferência, *Hugin* Interfaces de Programa de Aplicação (APIs), e *Hugin* interface gráfica do usuário.

A máquina de inferência (utilizado pelo *Hugin* interface de usuário gráfica) é um ambiente gráfico e pode-se utilizá-lo pelas APIs que são disponibilizadas como bibliotecas para C, C++, Java, e como um ActiveX. A parte principal da máquina de inferência é o compilador, que transforma as RB em árvores de junção, tornando possível a realização de inferências (ROSA, 2004 *apud* HUGIN).

O *Hugin* APIs permite que os programadores possam construir aplicações baseadas em RB, oferecendo recursos do motor de inferência para esse processo.

Quando o motor de inferência é empregado por uma das APIs, funciona como uma biblioteca utilizada usualmente no desenvolvimento de sistemas (ROSA, 2004 *apud* HUGIN).

O *Hugin* interface de usuário gráfica é utilizado tanto para criar, quanto para realizar alterações no modelo gerado da RB, e também permite realizar operações sobre evidências, exibição das distribuições de probabilidades resultantes, entre outros (ROSA, 2004 *apud* HUGIN).

A ferramenta disponibiliza um compilador que ao ser executado verifica se a rede possui erros, como, por exemplo, examina se a soma das probabilidades dos estados de um determinado nó deve estar totalizando 1. Após essa compilação, pode-se realizar inferências (ROSA, 2004 *apud* HUGIN).

O *Hugin* permite a exibição das probabilidades condicionais por meio de um recurso gráfico denominado janelas de monitoração, que podem ser colocadas próximas aos nós correspondentes, facilitando a visualização das probabilidades condicionais.

Além das RB, o *Hugin* manipula também os diagramas de influência, que são RB estendidas, e redes objeto-orientadas, que representam sub-redes (HUGIN, 2005).

A ferramenta disponibiliza um comando denominado gerador de casos, que pode simular casos e armazená-los em um arquivo de texto, baseando-se nas probabilidades condicionais. Por *default* o *Hugin* especifica criar 10.000 casos, assumindo 5 % de valores ausentes. É possível ao usuário a alteração dessas opções, bem como da especificação do método que deve ser utilizado para a geração dos casos (HUGIN, 2005).

Conforme *Hugin* (2005), a ferramenta pode ser utilizada na plataforma *Windows*, *Solaris*, *Linux* e *Mac*, e possui as seguintes versões:

- a) *Developer*: inclui o *Hugin* interface de usuário gráfica e o *Hugin* APIs. É direcionado para desenvolvedores que usam o motor de inferência como parte de aplicações específicas (em ambientes de desenvolvimento). Provê acesso a todas as funções do motor de inferência via *API*;
- b) *Lite*: possui as mesmas funcionalidades do *developer*, com exceção de manipular bases de dados com o número de registros limitado a 500. É a versão *free* do *developer* e pode ser obtida em <<http://www.Hugin.com>>;
- c) *Explorer*: possui as mesmas funcionalidades do *developer*, com o diferencial de basear-se no *Java*.

A seguir encontra-se a Figura 4 mostrando a interface da *shell*.

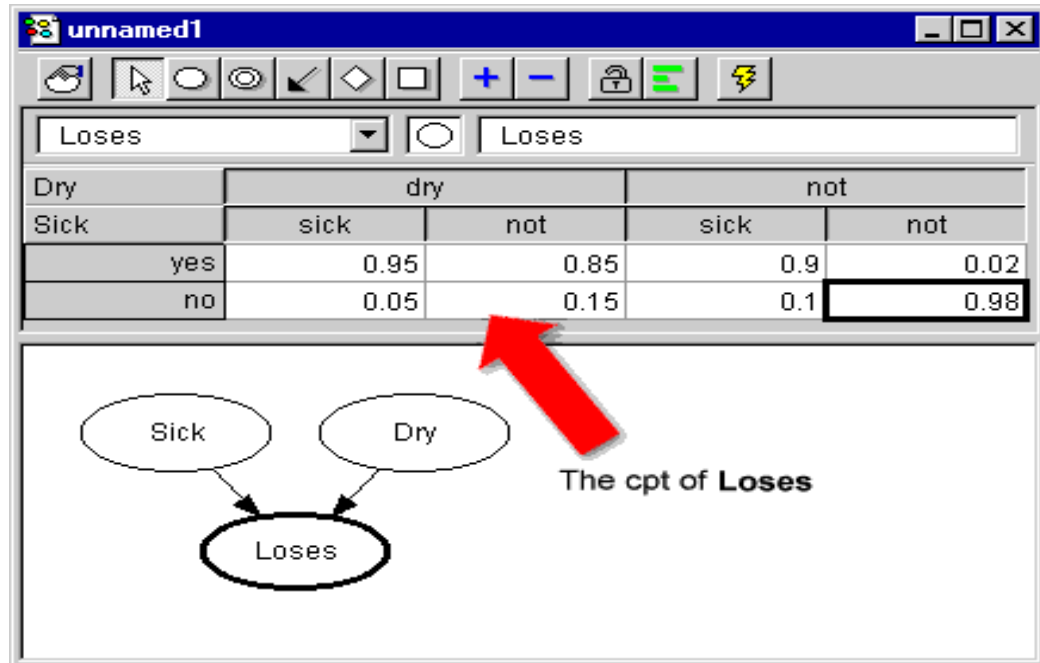


Figura 4. Interface Gráfica do Hugin
Fonte: HUGIN (2005)

5.3 BNG

O BNG é um sistema para construção de base de conhecimento com redes bayesianas. Uma classe de redes bayesianas é especificada com uma base de conhecimento de regras (BNG, 2005).

Dada uma base de conhecimento, um conjunto de informações sobre o contexto, algumas evidências e uma consulta, o BNG constrói uma rede bayesiana estruturalmente mínima para calcular a probabilidade a posteriori da consulta (BNG, 2005).

Foi desenvolvido no ano de 1995 pelo Departamento de Engenharia Elétrica e Ciência da Computação da Universidade de Wisconsin em Milwaukee, oferecendo duas vantagens sobre o uso das redes bayesianas tradicionais: representacional e computacional. A incorporação das variáveis quantificadas na base de conhecimento permite a representação da informação não permitidas em redes bayesianas tradicionais,

isso porque somente uma pequena parte de um modelo probabilístico pode ser relevante para o cálculo de uma particular probabilidade posterior, usando BNG pode resultar em economias computacionais significantes (NASSAR, 2005).

O algoritmo de construção da rede tem se mostrado seguro e completo, sendo a *shell* BNG escrita em CommonLisp e possuindo interface para a ferramenta IDEAL (BNG, 2005).

5.4 MICROSOFT BELIEF NETWORK TOOLS

Esta *shell* permite a criação e avaliação de redes de crença bayesianas. É composto pelo executável MSBN32 que utiliza uma versão proposta pelo *Bayes Net Interchange Format*, isto é, uma representação de um arquivo texto de redes de crença (MSBN, 2005).

A aplicação está escrita em Visual Basic 4.0 (32-bit), desenvolvida pela Microsoft e possui uma *Dynamic-Link Library (DLL)* para Windows, a MSBN32.DLL, disponibilizando funções para adicionar e remover nós, alterar suas probabilidades e salvar a rede. Este módulo é o componente ativo do conjunto, e suporta as seguintes operações:

- a) procura e armazena redes de crença em texto para criação e modificação da rede pela adição de nós e arcos da avaliação das probabilidades discretas das redes de crença;
- b) diagramas assimétricos de influência.
- c) é uma aplicação padrão com interface multi-documentos (MDI) para Windows. Ele tem uma barra de ferramentas e um help sensível ao contexto.

Esta ferramenta está disponível para as plataformas Windows 9x, XP e Windows NT (3.51 ou superior). Está disponível livre de encargos para usuários não comerciais em organizações de pesquisa e instituições educacionais. Uma cópia do aplicativo, mostrado na Figura 5, pode ser obtida enviando uma solicitação para: <<http://research.microsoft.com/adapt/MSBNx/>> (MSBN, 2005).

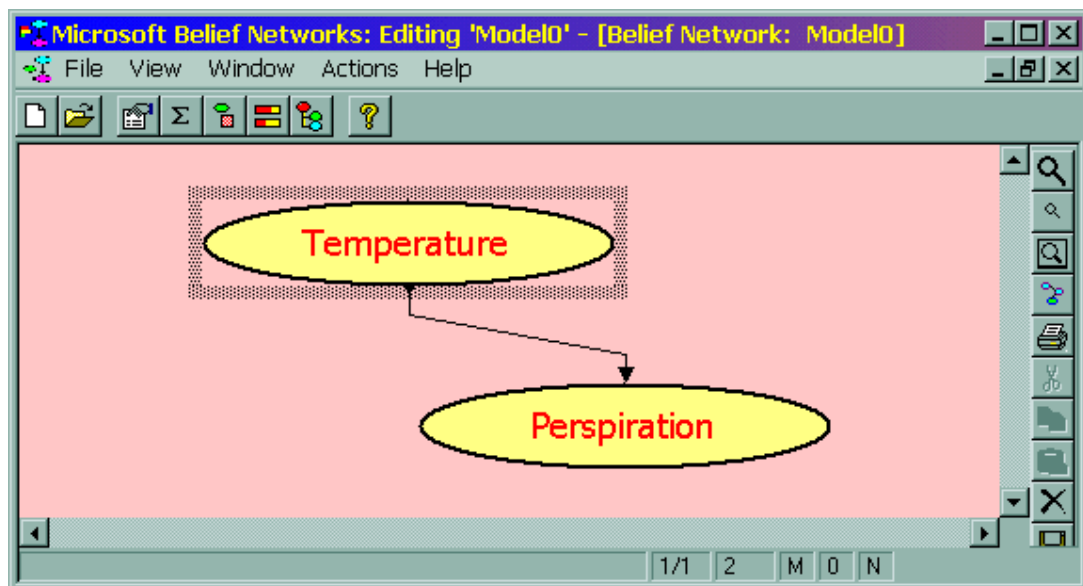


Figura 5. Interface Gráfica do MSBN
Fonte: MSBN (2006)

5.5 DXPRESS

DXPRESS trata-se de uma ferramenta para aquisição de conhecimento para desenvolvimento rápido de modelos probabilísticos para diagnósticos. Foi escrito em C++ pelo Rockwell Palo Alto Laboratory na Universidade de Stanford em 1994 (KIC, 2005).

Utiliza várias técnicas para acelerar a aquisição do conhecimento, e reduzir o tempo necessário no desenvolvimento de um sistema especialista probabilístico. Por exemplo, foi desenvolvido um protótipo de sistema de diagnóstico para uma turbina à jato usando energia elétrica e pneumática no DC-10. Este sistema diagnostica 60

diferentes falhas utilizando aproximadamente 50 testes ou observações diferentes (KIC, 2005).

Conforme Nassar (2005) algumas vantagens do DXPRESS são:

- a) avaliação rápida das distribuições de probabilidades;
- b) localiza o progresso durante as diferentes fases da aquisição do conhecimento;
- c) permite ao usuário dar entrada em notas e definições pertinentes para o processo de aquisição do conhecimento.

Possui API com bibliotecas para serem utilizadas em Visual C++, Visual Basic, e outras linguagens e servidores compatíveis com a plataforma Windows, como mostrado na Figura 6. Mais informações podem ser encontradas no site <<http://www.kic.com>> (KIC, 2005).

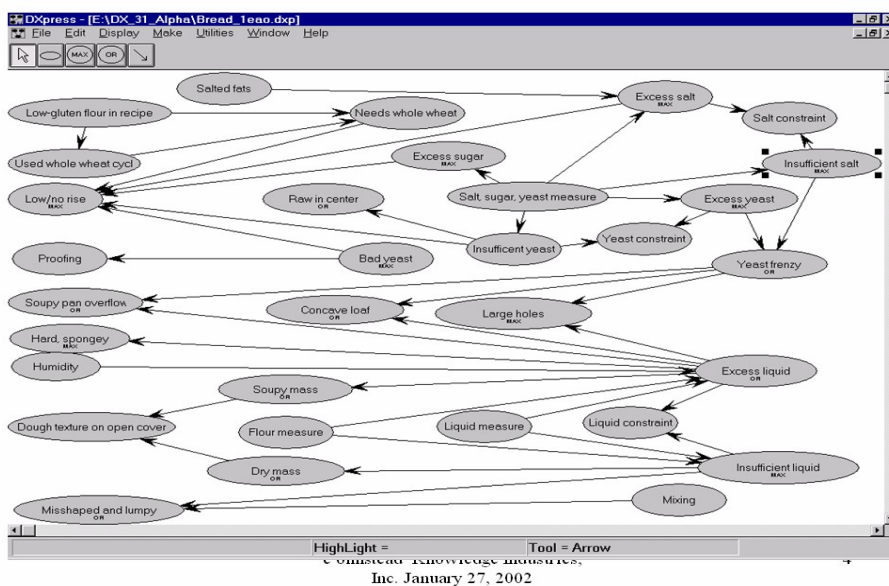


Figura 6. Interface Gráfica do DXPRESS
Fonte: KIC (2005)

5.6 SPIRIT

É uma *shell* para sistemas especialistas criada na Universidade de Hagen (FernUniversität Hagen), na Alemanha. Possui interface gráfica de desenvolvimento, e permite a criação de variáveis de diversos tipos, tais como: booleana, ordinal, cardinal e nominal. A parte qualitativa da rede de crença bayesiana é implementada a partir da inserção de regras se-então. Em seguida, valores de probabilidades são associados a estas regras e as variáveis, caracterizando a parte quantitativa da rede de crença bayesiana (SPIRIT, 2005).

Após a caracterização da rede, faz-se necessário inicializá-la, preparando-a para a compilação, ou seja, aprendizagem das regras. Feita a compilação podem ser efetuadas as inferências sobre a base, inserindo as evidências de um caso específico, chegando a um diagnóstico com um determinado valor de probabilidade (SPIRIT, 2005).

Como afirma Nassar (2005) as vantagens do SPIRIT são as seguintes:

- a) facilidade de acesso ao seu criador por meio do intercâmbio da UFSC com a *FernUniversität Hagen*;
- b) permite maior interação entre usuário e criador, por não ser um software comercial;
- c) trabalha em várias plataformas, inclusive Windows.

Versões de demonstração estão disponíveis em um endereço específico, mantido pela Universidade de Hagen: <<http://www.xspirit.de>>, bem como a *API* para ambiente Java, com diversas funções disponíveis para criação de uma rede, inserção, remoção e alteração de nós, inserção de valores de probabilidade e gravação. Pode ser

usada em Windows e Linux desde que a máquina do usuário possua o *Java Runtime Engine* instalado.

Na Figura 7 tem-se a interface gráfica do SPIRIT.

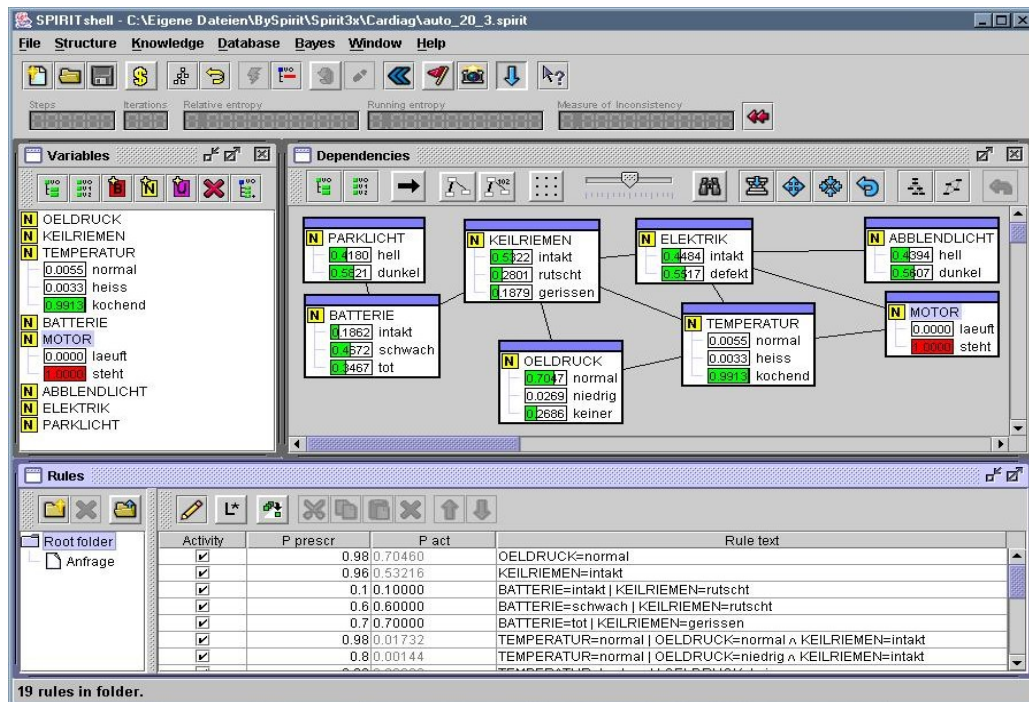


Figura 7. Interface Gráfica do SPIRIT
Fonte: XSPIRIT (2005)

Essas são algumas *shells* para realização de inserção de informações em bases de conhecimento. A seguir, na Tabela 2, tem-se algumas informações principais das *shells* mostradas anteriormente.

Tabela 2. Informações das *shells*

	Local	Fabricante	Plataforma	Site	Versão <i>Free</i>
Netica	Canadá	Norsys	Windows/Mac	www.norsys.com	Sim
Hugin	Dinamarca	Hugin Expert	Windows/Mac/ Solaris/Linux	www.hugin.com	Sim
BNG	Estados Unidos	Universidade de Wisconsin	CommonLisp	www.mcw.edu/midas/bng.html	N/D
MSBN	Estados Unidos	Microsoft	Windows	research.microsoft.com/adapt/ MSBNx/	Sim
DXPRESS	Inglaterra	Universidade de Stanford	Windows	www.kic.com	Sim
SPIRIT	Alemanha	Universidade de Hagen	Windows/ Linux	www.xspirit.de	Sim

No capítulo seguinte, serão mostrados alguns sistemas especialistas probabilísticos desenvolvidos, com suas características e funções principais.

6 ESTADO DA ARTE

Conforme descrito no histórico de sistemas especialistas probabilísticos, a partir dos anos 80, quando as fábricas demonstraram interesse na utilização e pesquisa nessa área de conhecimento, diversos sistemas foram desenvolvidos. No Brasil, podemos citar a UFSC e vários de seus acadêmicos, que produziram uma série de protótipos e avançam cada vez mais em suas pesquisas sobre sistemas especialistas probabilísticos, e também a UNESC, com alguns trabalhos realizados na área. A seguir, temos alguns exemplos de SEP desenvolvidos nos últimos anos em diversas áreas de aplicação.

6.1 SISTEMA ESPECIALISTA ON-LINE DE AUXÍLIO AO DIAGNÓSTICO DE CÂNCER DE PRÓSTATA

O presente trabalho desenvolvido na UFSC a título de Mestrado, aborda o desenvolvimento e implementação de um sistema especialista on-line dinâmico na área urológica para o auxílio ao diagnóstico de câncer de próstata (SEDACaP). O SEDACaP é uma ferramenta computacional capaz de gerenciar os dados clínicos de pacientes, e operando via Internet, auxiliar o médico na detecção da tendência da doença, principalmente em casos em que há suspeita de câncer de próstata. Foi realizado um amplo estudo sobre o câncer de próstata e outras doenças benignas da próstata (PEREIRA, 2004).

Nesse trabalho foram analisados os procedimentos clínicos para o diagnóstico do câncer de próstata para que o sistema refletisse o mais próximo possível estes procedimentos. A estrutura de banco de dados foi modelada para o

armazenamento e gerenciamento dos dados clínicos dos pacientes. O sistema especialista é um módulo do sistema e utiliza os dados clínicos armazenados para o processamento. A máquina de inferência está incorporada ao módulo especialista e ela foi desenvolvida especialmente para este sistema, para que o seu processamento fosse otimizado, aumentando conseqüentemente a performance (PEREIRA, 2004).

O sistema foi testado com casos clínicos reais para a avaliação dos resultados do módulo especialista em comparação com o resultado da última biópsia desses pacientes, e apresentou nos testes práticos a capacidade de gerenciar, de forma confiável, as sessões de usuários utilizando paralelamente o sistema, inclusive para o processamento especialista. O sistema especialista apresentou bons resultados, mostrando um grande potencial de auxílio médico para o diagnóstico (PEREIRA, 2004).

Com relação aos componentes dinâmicos, é possível realizar a inserção de novos médicos, pacientes, sintomas, doenças, entre outros. Foram utilizados no desenvolvimento do sistema a plataforma Java juntamente com o banco de dados Sybase PowerDesigner 9. Na Figura 8 tem-se uma das telas do sistema.

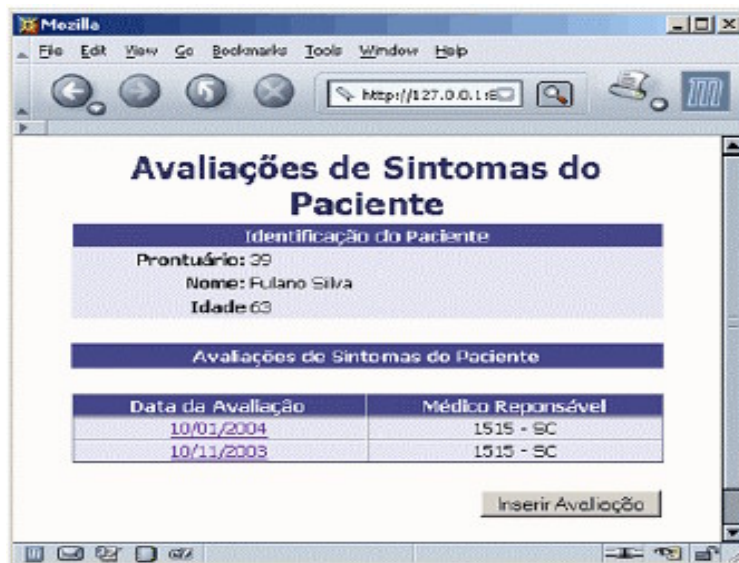


Figura 8. Listagem das avaliações de sintomas dos pacientes no SEDACaP
Fonte: PEREIRA, M (2004)

6.2 ABS-BAYES: SISTEMA ESPECIALISTA PROBABILÍSTICO PARA APOIO AO DIAGNÓSTICO DE ABSENTEÍSMO

O presente trabalho desenvolvido na UFSC a título de Mestrado, refere-se a um sistema especialista probabilístico dinâmico para apoio a área de recursos humanos de uma empresa, detectando os principais problemas de absenteísmo por doenças e suas principais características. Gera dinamicamente uma base de conhecimento, por estar ligado ao banco de dados de exames periódicos dos funcionários e ao banco de dados de atestados médicos apresentados por funcionários que faltam o trabalho. O usuário pode configurar as hipóteses diagnósticas ou deixar que o próprio sistema se encarregue de configurar com as doenças mais prevalentes na empresa (LOSI, 2003).

6.3 BIOWOMAN - BASE DE CONHECIMENTO DINÂMICA PARA SISTEMAS ESPECIALISTAS PROBABILÍSTICOS

O Biowoman foi resultado do trabalho de conclusão de curso da UNESC, sendo um Sistema Especialista Probabilístico Dinâmico, que auxilia no diagnóstico de leucorréia, ou seja, nas possíveis patologias encontradas no trato genital inferior feminino. Tem como característica principal, uma base de conhecimento dinâmica capaz de receber uma nova que possa auxiliar ao diagnóstico proposto, e ainda permitir a realização de estudos sobre a influência que essa nova evidência provoca nas demais já organizadas na base de conhecimento (RODRIGUES, 2002).

A aquisição do conhecimento se deu por entrevistas com a especialista na área ginecológica e pesquisas bibliográficas. Esse sistema foi desenvolvido utilizando-

se a API da *shell* Netica e a sua interface gráfica no *C++Builder*, como pode ser observado na Figura 9, a tela inicial do Biowoman.

Condição	Valor (%)
Flora Vaginal Normal	30,00
Candidíase	20,00
Vaginose Bacteriana	15,00
Trichomonas	5,00
Clamídia	1,50
Gonococo	2,00
Flora Mista	12,00
Vaginose Citolítica	10,00
Cocus	3,00
Vaginite Inflamatória Desecamativa	1,50

Figura 9. Tela inicial do Biowoman
Fonte: RODRIGUES, A (2002)

6.4 SISTEMA ESPECIALISTA PROBABILÍSTICO PARA APOIO AO DIAGNÓSTICO DE POTENCIAL ECONÔMICO – SEPE

Este trabalho foi desenvolvido na UFSC a título de mestrado e tem como objetivo apoiar o planejamento mercadológico a partir do conhecimento do potencial econômico dos municípios de Santa Catarina. Na sua elaboração, utilizam-se com dados incertos sobre o mercado, ou seja, tomam-se decisões sem ter as informações necessárias. Com o intuito de obter um diagnóstico mais eficiente, é necessário que os técnicos visualizem os dados relacionados ao potencial de uma forma gráfica, e que

sejam investigadas relações entre variáveis endógenas e exógenas² aos Correios, por meio de evidências observadas (PASINI, 2002).

O sistema SEPE explora o raciocínio probabilístico em sistemas especialistas, com uma base de conhecimento dinâmica, empregando ainda, técnicas de mineração de dados no processo de aquisição e atualização de conhecimento de uma base de dados. Foi desenvolvido para a plataforma Windows, no ambiente Visual Basic 5, utilizando as *DLL's* da *shell* Netica (PASINI, 2002).

6.5 SISTEMA ESPECIALISTA PROBABILÍSTICO: BASE DE CONHECIMENTO DINÂMICA

Nesta pesquisa desenvolvida a título de mestrado na UFSC, foi tomada a base de conhecimento do sistema SISPAN, e foi desenvolvido um sistema que permite ao usuário incluir e excluir nós, bem, como alterar valores de probabilidade da rede bayesiana. Dessa forma, pode ser avaliada a acurácia, ou ausência de erro de um Sistema Especialista Probabilístico, podendo ser utilizado como um ambiente de ensino por simulação. Entre os poucos sistemas com base de conhecimento dinâmica, está o sistema SEDIN – Sistema Especialista Dinâmico (STEIN, 2000).

O SEDIN foi avaliado por um especialista da área médica que o considerou adequado para a utilização tanto no diagnóstico de avaliação nutricional em crianças com até dois anos de idade, quanto para o ensino. Esse sistema mostrado na Figura 10, foi desenvolvido com o auxílio da *shell* Netica e o ambiente de programação C++ Builder, disponibiliza ao usuário as opções de inserção, alteração e exclusão de nós na base de conhecimento (STEIN, 2000).

² Variáveis endógenas são internas e exógenas são variáveis externas.

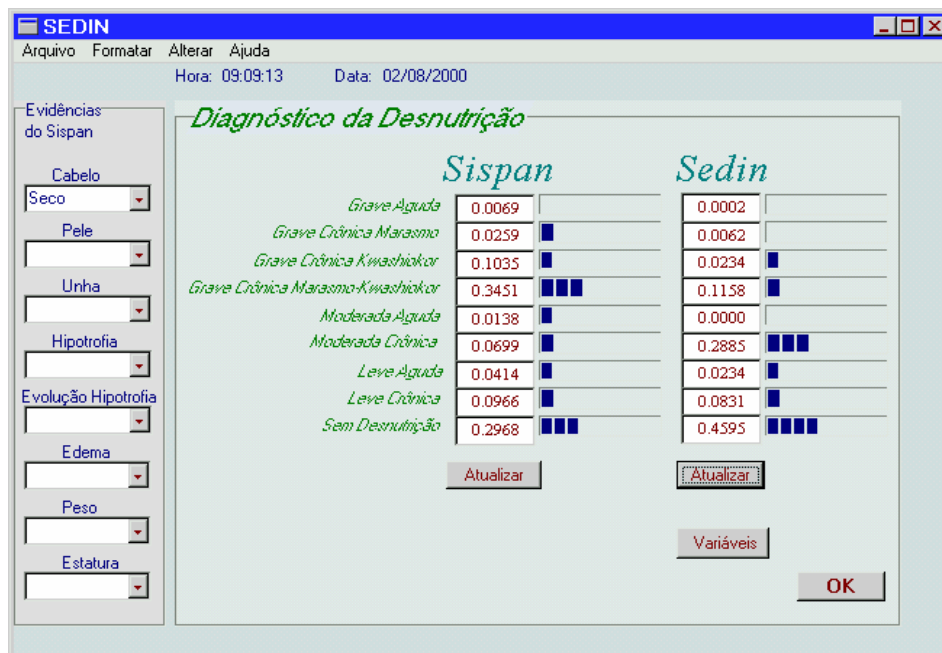


Figura 10. Tela inicial do SEDIN
Fonte: STEIN, C (2000)

6.6 DXPLAIN

É um sistema de apoio a decisão que utiliza um conjunto de dados clínicos (sinais, sintomas, dados laboratoriais) para produzir uma lista de diagnósticos associados as manifestações clínicas. Este software, produzido pelo grupo de pesquisa do Dr. Octo Barnett no Laboratório de Ciência da Computação do Hospital Geral de Massachusetts, possui uma grande base de dados sobre 5000 manifestações clínicas associadas com mais de 2000 diferentes doenças (DXPLAIN, 2005).

O sistema tem sido utilizado por milhares de médicos e estudantes de medicina, sendo que o banco de dados e o sistema estão sendo continuamente aperfeiçoados e atualizados, para que possa ser utilizado em vários hospitais e escolas de medicina para educação clínica, e como um auxílio educacional na solução de problemas clínicos. O DXplain tem as características de livro texto eletrônico, e um

manual de referência médico, oferecendo mais de 10 referências para cada doença, dando ênfase para revisões clínicas onde estas estão disponíveis (DXPLAIN, 2005).

A seguir encontra-se a Figura 11 com a interface do DXPlain.

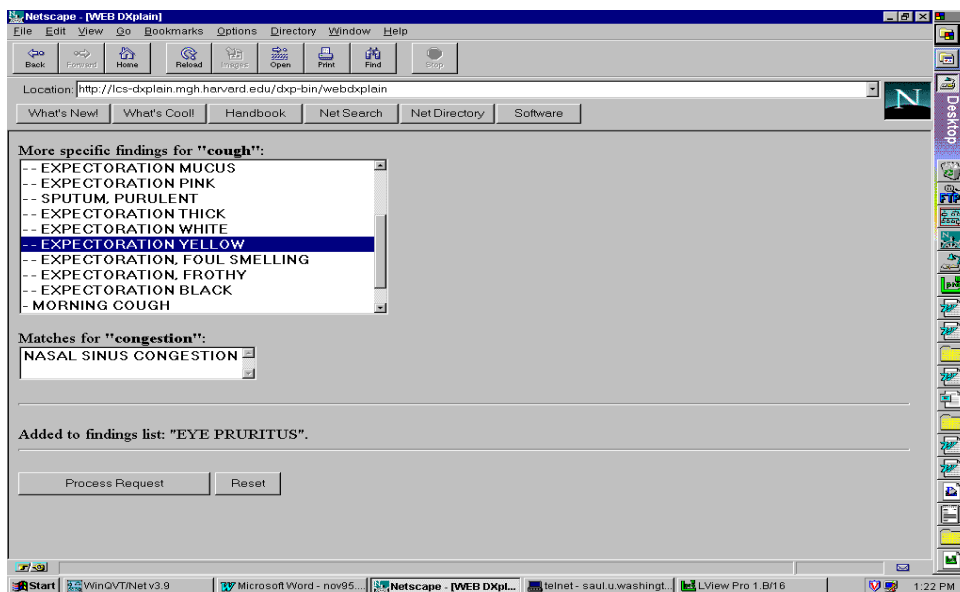


Figura 11. Interface do DXPlain
Fonte: DXPlain (2005)

Esses foram alguns exemplos de sistemas especialistas probabilísticos desenvolvidos no Brasil e exterior. Por ser uma área de estudos recente, observa-se que são poucos os sistemas especialistas probabilísticos dinâmicos. No Apêndice B encontram-se mais alguns exemplos de sistemas especialistas probabilísticos, porém com base de conhecimento estática. No capítulo a seguir, serão mostradas algumas características que devem existir no projeto de interface com o usuário em um SEP.

7 PROJETO DE INTERFACE COM O USUÁRIO

A engenharia de software é uma parte integrante da engenharia de sistemas e de hardware, que abrange um conjunto de três elementos: métodos, que proporcionam os detalhes de como construir o software; ferramentas, que proporcionam apoio automatizado ou semi-automatizado aos métodos; procedimentos, que são uma espécie de elo de ligação entre os métodos e as ferramentas para que haja uma boa construção do software (PRESSMAN, 1995).

O processo para criação da interface com o usuário é iniciado com a criação de diferentes modelos de função do sistema. As tarefas orientadas para o homem e o computador para permitir com que o sistema funcione são delineadas, questões de projeto que se aplicam aos projetos de interface são consideradas, são utilizadas ferramentas para construção do protótipo para posteriormente avaliá-lo quanto à qualidade (PRESSMAN, 1995).

No item a seguir serão descritos alguns modelos de projeto de interface com o usuário.

7.1 MODELOS DE PROJETOS DE INTERFACES

No momento do projeto de interface do sistema quatro modelos diferentes são considerados. O engenheiro de software cria um modelo de projeto; o engenheiro de software estabelece um modelo de usuário, o usuário final desenvolve uma imagem mental ou modelo do usuário e assim os programadores criam a imagem do sistema. O papel do projeto de interface é conciliar e apresentar uma representação consistente da interface, já que cada um desses modelos pode ser diferente (PRESSMAN, 1995).

A seguir serão descritos os quatro modelos de projeto de interface:

- a) **modelo de projeto:** incorpora representações de procedimentos, arquitetura e dos dados do software;
- b) **modelo de usuário:** descreve o perfil dos usuários finais do sistema, compreendendo idade, sexo, capacidades físicas, personalidade, etc. São classificados em principiantes, usuários instruídos e intermediários e por fim, usuários instruídos e freqüentes;
- c) **modelo do usuário:** trata-se da imaginação do usuário final quanto à imagem final do sistema. Quanto maior o conhecimento e experiência do usuário, melhor será a precisão da descrição;
- d) **imagem do sistema:** combina a aparência e a maneira como a interface é sentida com as informações de apoio, seja livros, manuais, vídeos. Para que o usuário sinta-se à vontade ao utilizar o sistema, o ideal é que a imagem do sistema coincida com o modelo do usuário.

Conforme o projeto da interface com o usuário vai se desenvolvendo, algumas questões devem ser observadas. Segundo Pressman (1995) as questões são as seguintes:

- a) tempo de resposta do sistema: caracteriza-se pelo tempo em que o usuário solicita uma ação até a resposta do software com a saída ou ação desejada. Caso o tempo de resposta seja muito longo poderá causar frustração e estresse ao usuário, porém um tempo curto também é prejudicial, fazendo com que o usuário se apresse e cometa erros;
- b) facilidades de ajuda: elas podem ser *integradas*, ou seja, projetadas no software desde o começo, possibilitando ao usuário a escolha de tópicos

relevantes às ações que está realizando. Outra facilidade é a *add-on*, que é adicionada ao software após a sua construção, sendo uma espécie de manual do usuário;

- c) manuseio de informações de erro: as mensagens devem informar de forma clara o que realmente aconteceu, fornecendo conselhos se possível, para que o usuário possa solucionar o problema.
- d) rotulação de comandos: caso o usuário não queira utilizar totalmente as facilidades de uma interface com janelas este pode fazer uso de comandos pré-definidos pelo programador. O único problema na utilização de comandos é que eles podem realizar ações diferentes dependendo do sistema, aumentando a possibilidade de erros.

No item a seguir serão mostradas algumas formas de se projetar uma interface padronizada, a fim de facilitar a sua utilização por parte do usuário.

7.2 PADRÕES DE INTERFACE

A padronização da interface dos sistemas computacionais vem sendo adotada cada vez mais pelos programadores, já que implementar uma nova janela toda vez que solicitado um novo sistema daria muito trabalho. Com essa padronização, o programador precisa apenas copiar a seqüência de códigos que necessita para cada parte do sistema, poupando tempo e facilitando a sua utilização por parte do usuário, já que os sistemas terão interfaces semelhantes (PRESSMAN, 1995).

Todos os sistemas interativos, exceto os especializados – os sistemas de tela pequena, displays coloridos e interfaces com o usuário, utilizam cores de diferentes

maneiras. Em alguns sistemas (como os processadores de textos) a cor é utilizada simplesmente para dar destaque; em outros (como os sistemas CAD), ela é utilizada para mostrar diferentes camadas em um projeto (SOMMERVILLE, 2003).

As cores podem aperfeiçoar as interfaces com o usuário, auxiliando na compreensão e a gerenciar a complexidade. Porém, como afirma Shneiderman (1998 apud SOMMERVILLE, 2003), algumas diretrizes devem ser levadas em consideração para um uso eficaz de cores em interfaces com o usuário, sendo elas:

- a) limite do número de cores utilizadas, isto é, não se deve utilizar mais do que quatro ou cinco cores em uma janela e não mais que sete em uma interface de sistema;
- b) utilizar a mudança de cores para mostrar uma modificação no status do sistema, ou seja, se houver uma mudança de cor na interface isso significa que algum evento importante ocorreu;
- c) utilizar código de cores de maneira cuidadosa e consistente, isto é, se uma parte do sistema exibe mensagens de erro com determinada cor, as outras partes devem fazer o mesmo;
- d) ser cuidadoso quanto a pares de cores, ou seja, por causa da fisiologia do olho humano as pessoas não podem focalizar o vermelho e o azul simultaneamente, podendo, caso utilizados, causar estresse visual.

Além da padronização, os sistemas devem conter mensagens de ajuda caso algo de errado aconteça no momento da utilização, devendo especificar detalhadamente o erro e quais opções o usuário terá para solucioná-lo.

A seguir, são mostradas algumas informações sobre o ErgoList, que consiste em um grupo de questões com a função de avaliar e melhorar a interface dos sistemas.

7.3 ERGOLIST

O ErgoList destina-se a apoiar a inspeção da interface e descobrir seus defeitos ergonômicos mais visíveis. Este projeto resulta da colaboração entre o SoftPólis, núcleo Softex-2000 de Florianópolis, e o LabiUtil, Laboratório de Utilizabilidade da UFSC/SENAI-SC/CTAI, podendo ser encontrado no *site* <<http://www.labiutil.inf.ufsc.br/ergolist>> (ERGOLIST, 2006).

A avaliação da interface com o usuário contempla 18 grupos de questões, sendo cada um deles especializado em um aspecto ou critério que determina a ergonomia de uma interface homem-máquina.

Dentre esses grupos de funções retiradas do Ergolist, alguns deles são mostrados a seguir:

- a) **presteza:** é possível dizer que um sistema possui presteza quando guia o usuário e lhe poupa, por exemplo, o aprendizado de uma série de comandos, assim, uma boa presteza facilita a navegação no aplicativo e diminui a ocorrência de erros;
- b) **agrupamento por Localização:** significa que o sistema possui uma boa organização dos objetos, como por exemplo, com a utilização de um menu ou colocando os itens em ordem alfabética;
- c) **feedback:** diz respeito as respostas do sistema as ações do usuário, que podem ir do simples pressionar de uma tecla até uma lista de comandos.
- d) **legibilidade:** diz respeito as características das informações apresentadas na tela que possam dificultar ou facilitar a leitura da mesma (brilho do caractere, contraste letra/fundo, tamanho da fonte, espaçamento entre

palavras, espaçamento entre linhas, espaçamento de parágrafos, comprimento da linha, entre outros);

- e) **ações mínimas:** está relacionada à carga de trabalho em relação ao número de ações necessárias à realização de uma tarefa. Então, quanto mais numerosas e complexas forem as ações necessárias para se chegar a uma meta, a carga de trabalho aumentará e, com ela, a probabilidade de ocorrência de erros;
- f) **ações explícitas do usuário:** se refere as relações entre o processamento pelo computador e as ações do usuário, sendo a relação explícita, isto é, o computador deve processar somente aquelas ações solicitadas pelo usuário e apenas quando solicitado a fazê-lo;
- g) **qualidade das mensagens:** favorece o aprendizado do sistema, indicando ao usuário a razão ou a natureza do erro cometido, o que ele fez de errado, o que ele deveria ter feito e o que ele deve fazer;
- h) **consistência:** é conveniente padronizar, tanto quanto possível, os objetos quanto a seu formato e a sua denominação, e padronizar a sintaxe dos procedimentos. A falta de consistência nos menus, por exemplo, pode aumentar consideravelmente o tempo de procura, sendo também, a falta de homogeneidade razão importante da recusa de utilização de alguns sistemas.

Conhecidas algumas das questões do ErgoList, no próximo item são dados alguns conceitos sobre usabilidade, uma parte da engenharia de software que vem sendo cada vez mais utilizada quando se trata da construção de sistemas.

7.4 USABILIDADE

Usabilidade é um termo usado para definir a facilidade com que as pessoas podem empregar uma ferramenta ou objeto a fim de realizar uma tarefa específica. A usabilidade pode também se referir ao estudo dos princípios por trás da eficiência percebida de um objeto (WIKIPEDIA, 2006).

Na interação homem-máquina e também na Ciência da Computação, usabilidade normalmente se refere à simplicidade e facilidade com que uma interface, um programa de computador ou um *website* podem ser utilizados. O termo também é utilizado em contexto de produtos como aparelhos eletrônicos, em áreas da comunicação e produtos de transferência de conhecimento, como manuais, documentos e ajudas on-line (WIKIPEDIA, 2006).

Em essência, o termo começou a ser usado no início da década de 80 pelas áreas de psicologia e ergonomia como substituto da expressão *user-friendly* (amigável, em português), uma vez que máquinas são ferramentas feitas para servir, não para serem amigas (WEBINSIDER, 2005).

Além disso, a expressão usabilidade passou a ser questionada como vaga e subjetiva. Para evitar o desgaste, muitos autores buscaram caracterizar usabilidade de várias maneiras, por meio de definições:

- a) orientadas às características ergonômicas do produto;
- b) orientadas ao usuário, mostrando a relevância de seu esforço mental ou de sua atitude frente ao produto;
- c) baseadas no desempenho do usuário, associadas à forma de interação, dando ênfase à facilidade de uso e grau de aceitação;
- d) orientadas ao contexto de uso.

7.4.1 Definições de Usabilidade

Conforme Elias (2006) a Internacional Standard Organization (ISO) dispõe de duas definições de usabilidade:

- a) a primeira descreve que usabilidade é a extensão na qual um produto pode ser usado por usuários específicos, para alcançar objetivos específicos com efetividade, eficiência e satisfação em um contexto de uso específico (ISO 9241-11).
- b) a segunda descreve a usabilidade como um conjunto de atributos de software relacionado ao esforço necessário para seu uso e para o julgamento individual de tal uso por determinado conjunto de usuários (ISO/IEC 9126).

A usabilidade está relacionada aos estudos de ergonomia e de interação homem-máquina e diretamente ligada ao diálogo na interface e, é a capacidade do software em permitir que o usuário alcance suas metas de interação com o sistema. Ser de fácil aprendizagem, permitir utilização eficiente e apresentar poucos erros são os aspectos fundamentais para a percepção da boa usabilidade por parte do usuário. Mas a usabilidade pode ainda estar relacionada com a facilidade de ser memorizada, e ao nível de satisfação do usuário (WIKIPEDIA, 2006).

Um sistema com boa efetividade permite que o usuário alcance os objetivos iniciais de interação, sendo observada em termos de finalização de uma tarefa, e também em termos de qualidade do resultado obtido (SANTOS, 2003).

Eficiência se refere à quantidade de esforço e recursos necessários para se chegar a um determinado objetivo. Os desvios que o usuário faz durante a interação, e a

quantidade de erros cometidos pode servir para avaliar o nível de eficiência do sistema (SANTOS, 2003).

A terceira medida de usabilidade, a satisfação, é a mais difícil de medir e quantificar, pois está relacionada com fatores subjetivos. De maneira geral, satisfação se refere ao nível de conforto que o usuário sente ao utilizar a interface, e qual a aceitação como maneira de alcançar seus objetivos (WIKIPEDIA, 2006).

Encerrado o capítulo sobre projeto de interface com o usuário, tem-se descritos os passos para a criação do DinBayes.

8 DINBAYES - COMPONENTES DINÂMICOS EM SISTEMAS ESPECIALISTAS PROBABILÍSTICOS

O trabalho proposto tem como objetivo a construção de uma base de conhecimento dinâmica via interface com o usuário. O fundamento metodológico que norteou o trabalho se baseou nas seguintes etapas:

- a) realizar levantamento bibliográfico;
- b) estudar o trabalho de conclusão de curso da UNESC intitulado Biowoman – Base de Conhecimento Dinâmica Para Sistemas Especialistas Probabilísticos (RODRIGUES, 2002);
- c) estudar as funções da API da *shell* Netica para gerenciamento dos nós, da rede e do ambiente;
- d) realizar o projeto da interface com o usuário e implementação;
- e) definir uma base de conhecimento da área da saúde para realização de testes do sistema;
- f) documentar as funções da API da *shell* Netica para inclusão, exclusão e alteração de informações na base de conhecimento, bem como da função de carregar uma rede já existente.

8.1 LEVANTAMENTO BIBLIOGRÁFICO

O levantamento bibliográfico foi o ponto de partida da elaboração do trabalho. Diversos temas foram estudados para que houvesse um maior embasamento teórico sobre a área de sistemas especialistas probabilísticos, para assim, realizar o projeto, bem como a implementação do SEPD.

Na área de IA foram estudados os assuntos sistemas especialistas probabilísticos, raciocínio sob incerteza, inclusão de informação em bases de conhecimento, aquisição de conhecimento e seus métodos e *shells* para SEP. Na área de engenharia de software, foi abordado o projeto de interface com o usuário, destacando-se as normas de usabilidade.

Nessa parte da elaboração do trabalho, foram encontradas algumas dificuldades com relação à compreensão do desenvolvimento de BC dinâmicas, já que a quantidade, e principalmente a qualidade da documentação sobre a utilização de componentes dinâmicos em SEP é muito pequena.

Finalizada a parte de levantamento bibliográfico, a próxima etapa compreende o estudo do trabalho de conclusão de curso da UNESC intitulado de BioWoman – Base de Conhecimento Dinâmica Para Sistemas Especialistas Probabilísticos, que foi realizado a fim de compreender a utilização de componentes dinâmicos da API da *shell* Netica, e a partir daí, explorar um pouco mais as funções dessa *shell* para a construção de um SEPD.

8.2 ESTUDO DO BIOWOMAN

O Biowoman é um trabalho de conclusão de curso da Universidade do Extremo Sul Catarinense (RODRIGUES, 2002), que compreendeu o desenvolvimento de um sistema especialista probabilístico dinâmico que auxilia no diagnóstico de leucorréia e as prováveis patologias encontradas no trato genital inferior feminino (Flora Vaginal Normal, Flora Mista, Vaginose Citolítica, Cocos, Vaginite Inflamatória Descamativa, Candidíase, Vaginose Bacteriana, Trichomonas, Gonococo e Clamídia)

Tem como característica principal, uma base de conhecimento dinâmica capaz de receber uma nova evidência que auxilie no diagnóstico e ainda possibilitar o estudo da influência que essa nova evidência provoca nas demais organizadas na base de conhecimento.

Os objetivos deste trabalho, que foram alcançados são os seguintes: permitir a inclusão de uma nova evidência em Sistemas Especialistas Probabilísticos (SEP), realizando a integração da *shell* Netica com o ambiente de desenvolvimento *Borland C++ Builder*, incluindo as funções para inclusão de novas informações na BC para que fossem aplicadas no auxílio ao diagnóstico da leucorréia. Por fim, disponibilizar documentação a respeito das funções da biblioteca do Netica para inclusão de novas informações na base de conhecimento (RODRIGUES, 2002).

Na construção do Biowoman foram utilizadas as funções de criação e gravação da rede, criação de um novo nó, e ainda, inserção dos estados e de suas probabilidades, juntamente com um *link* em relação ao nó diagnóstico. Quanto aos componentes dinâmicos utilizados na construção da interface, destaca-se a criação de ComboBox, Edit e Label de forma dinâmica (RODRIGUES, 2002).

A validação do BioWoman foi realizada pela ginecologista e obstetra Sandra Aparecida Manenti, e mais dois especialistas, professores do Departamento de Medicina, os quais testaram o funcionamento do sistema e verificaram que o desempenho se adequou ao objetivo proposto.

Ao iniciar a utilização do Biowoman, é apresentado ao usuário a tela inicial do sistema, Figura 12, onde são exibidas as evidências e as probabilidades iniciais de cada hipótese diagnóstica. Ao realizar uma consulta, automaticamente as probabilidades são atualizadas.

Menu Help Sobre

Diagnóstico:

Idade Cor Prurido Odor Odor c/ teste (KOH) Ardência Dispareunia Suspeita de DST

Flor. Ciclo Menstrual Imunossupressão Gruvidade Uso de Antibióticos pH Vaginal Exame a fresco

DIAGNÓSTICO

Flora Vaginal Normal	30,00	%
Candidíase	20,00	%
Vaginose Bacteriana	15,00	%
Trichomonas	5,00	%
Clamídie	1,50	%
Gonocococo	2,00	%
Flora Mista	12,00	%
Vaginose Clotítica	10,00	%
Cocus	3,00	%
Vaginite Inflamatória Descamativa	1,50	%

Limpar

Sair

Figura 12. Tela inicial do BioWoman
 Fonte: RODRIGUES, A (2002)

O sistema apresenta a possibilidade de adicionar uma nova evidência à base, inserindo seu nome, quantidade de estados e também os nomeando, bem como inserindo seus valores probabilísticos em relação a cada hipótese diagnóstica, como exibido na Figura 13.

Inseridas todas as probabilidades, a nova evidência aparece disponível para consulta pelo usuário, e então, pode-se salvar a rede e encerrar o programa.

Insira o nome da nova evidência: Obs: As letras devem ser minúsculas, sem acento e sem espaço em branco, exemplo: tipo_clima

Nº de estados: Obs: Forneça a quantidade de estados que essa nova evidência possui, exemplo: se frio e calor, então = 2.

Informe o(s) nome(s) do(s) estado(s):

Insira os valores probabilísticos referentes a cada hipótese diagnóstica:

	Frio	Calor
Flora Vaginal Normal	50	50
Candidíase	70	30
Vaginose Bacteriana	50	50
Trichomoníase	60	40
Clamídia	80	20
Gonococo	10	90
Flora Mista	50	50
Vaginose Citolítica	30	70
Coccos	40	60
Vaginite Inflamatória Descompensada	50	50

Obs: Os valores probabilísticos devem ser maior que 0. E, para cada diagnóstico (linha) esses valores devem fechar 100.
Exemplo: Para Flora Vaginal Normal, se Frio = 43, então Calor = 57.

Concluir

Diagnóstico

Salvar

Sair

Figura 13. Inserção de uma nova evidência no BioWoman
Fonte: RODRIGUES, A (2001)

Encerrada a etapa de estudos do Biowoman, seguiu-se no estudo das funções da API da *shell* Netica que serão descritas a seguir.

8.3 ESTUDO DAS FUNÇÕES DA API DA SHELL NETICA

O estudo das funções disponíveis pela API da *shell* Netica, para gerenciamento de nós, da rede e do ambiente foi uma das etapas mais importantes da construção do SEPD, pois serviu de base para a implementação. A partir dessa etapa foram decididas as funções que estariam disponíveis no sistema, intitulado então de DinBayes, para a posterior integração da base de conhecimento com o ambiente visual *C++ Builder*.

As funções da *shell* Netica podem ser separadas em grupos, sendo estes rede, nó, link e sistema.

No grupo rede, encontram-se as funções responsáveis pelo gerenciamento da rede, desde a criação e nomenclatura até a função para apagá-la totalmente, destacando-se:

- a) `NewNet_bn ()`, função responsável pela criação da rede;
- b) `DeleteNet_bn ()`, função para apagar a rede.

No grupo nó, tem-se as funções necessárias para gerenciamento dos nós da rede, como por exemplo a inserção de nós, alteração e remoção dos mesmos, destacando-se:

- a) `NewNode_bn ()`, responsável pela criação de um novo nó;
- b) `SetNodeStateNames ()`, é a função que atribui os nomes aos estados do nó.

Em seguida tem-se o grupo link, onde é possível criar uma relação de parentesco entre os nós criados na rede, ou desfazê-la, destacando-se:

- a) `AddLink_bn ()`, função para criação do link entre dois nós que possuem relação de parentesco;

Por último, a API da *shell* Netica possui o grupo sistema, responsável pelo gerenciamento do ambiente da rede, destacando-se as seguintes funções:

- a) `NewNeticaEnviron_bn ()`, função para criar o ambiente da rede;
- b) `InitNetica_bn ()`, função responsável pela inicialização do Netica.

8.4 PROJETO DA INTERFACE COM O USUÁRIO

O projeto de sistemas de computador abrange várias atividades, desde o projeto de hardware até o projeto de interface com o usuário, sendo esse fundamental para o sucesso de um sistema. Uma interface fácil de ser utilizada, na melhor das

hipóteses, resultará em alto nível de produtividade por parte do usuário, proporcionando maior rendimento e evitando o stress causado pelos longos períodos a frente do computador. Se as informações forem apresentadas de maneira confusa ou enganosa, os usuários poderão se confundir com o significado dessas informações. Eles podem iniciar uma seqüência de ações que venham a corromper os dados, ou mesmo causar falhas graves no sistema (SOMMERVILLE, 2003).

A seguir será descrito como foi realizada a organização da interface, considerando sua utilização em um SEP que prevê a utilização de componentes visuais dinâmicos. Devido a esse tipo de utilização dos componentes visuais, foi necessário contemplar alguns critérios de usabilidade, adaptabilidade e algumas questões do ErgoList para a construção da interface.

8.4.1 Organização da Interface

No momento em que se iniciou o projeto de interface com o usuário até a sua avaliação, optou-se pela utilização do ErgoList, pois esse destina-se a apoiar a inspeção da interface e descobrir seus defeitos ergonômicos mais visíveis (ERGOLIST, 2006).

A avaliação da interface com o usuário contempla 18 grupos de questões, sendo cada um deles especializado em um aspecto ou critério que determina a ergonomia de uma interface homem-máquina.

Dentre esses grupos de questões retiradas do Ergolist, as que se adequaram às ações disponíveis no DinBayes são: *presteza*, *agrupamento por localização*, *feedback*, *legibilidade*, *ações mínimas*, *ações explícitas*, *mensagens de erro* e *consistência*.

8.4.1.1 Presteza

No DinBayes o usuário não necessita digitar comandos para realizar as ações que necessita, isso por que os dados são inseridos em componentes visuais *edits* e acessados por meio de botões e menus.

8.4.1.2 Agrupamento por Localização

O DinBayes disponibiliza ao usuário duas opções de menu, o menu clássico no topo da tela, ou uma barra de ferramentas localizada no canto esquerdo, possuindo as funções mais importantes do sistema, com o objetivo de agilizar as ações por parte do usuário.

8.4.1.3 Feedback

O DinBayes traz ao usuário um *feedback* imediato, sendo que suas ações são respondidas em tempo real de execução, ou seja, assim que os dados são inseridos na rede, esses já estão disponíveis para serem utilizados em uma eventual consulta ou alteração.

8.4.1.4 Legibilidade

O DinBayes possui linguagem simples e clara, os ícones e botões são objetivos e seus rótulos descrevem exatamente que ação será executada.

8.4.1.5 Ações Mínimas

O DinBayes apresenta algumas das recomendações descritas no Ergolist para esse grupo, como por exemplo:

- a) no momento da inserção de dados, o sistema posiciona o cursor no primeiro campo;
- b) o usuário pode, de forma simples e rápida, percorrer os componentes visuais utilizados;
- c) a estrutura dos menus é concebida de modo a diminuir os passos necessários para a seleção.

8.4.1.6 Ações Explícitas do Usuário

O DinBayes se adequa a esse grupo, isso porque, é sempre o usuário quem comanda a navegação entre os campos de um formulário, e o sistema exige do usuário uma ação explícita de entrada/confirmação (ENTER) para completar um comando.

8.4.1.7 Qualidade das Mensagens

Com relação a possibilidade de erros implementada no DinBayes, suas mensagens são simples e direcionam diretamente ao erro ocorrido, facilitando ao máximo o entendimento e a correção do erro cometido por parte do usuário. Uma das mensagens de erro contida no sistema será mostrada caso o usuário tente mudar de um componente visual *edit* para outro deixando o anterior vazio, como no caso da inserção

de um novo nó, quando o usuário não poderá inserir a quantidade de estados sem antes inserir o nome do nó.

8.4.1.8 Consistência

O DinBayes apresenta menus, botões e textos de forma padronizada a fim de facilitar a sua utilização. Um controle contra erros do usuário foi utilizado, não permitindo que o usuário troque o campo de texto deixando-o vazio, ou também, digite uma letra em um campo numérico.

Esses foram alguns dos grupos de questões utilizados para a organização da interface do sistema. A seguir, serão descritas as ações disponíveis no sistema.

8.4.2 Ações do Sistema

O DinBayes possui diversas funções implementadas por meio da integração do ambiente de programação C++ Builder e a API da *shell* Netica, como mostrado na Figura 14.

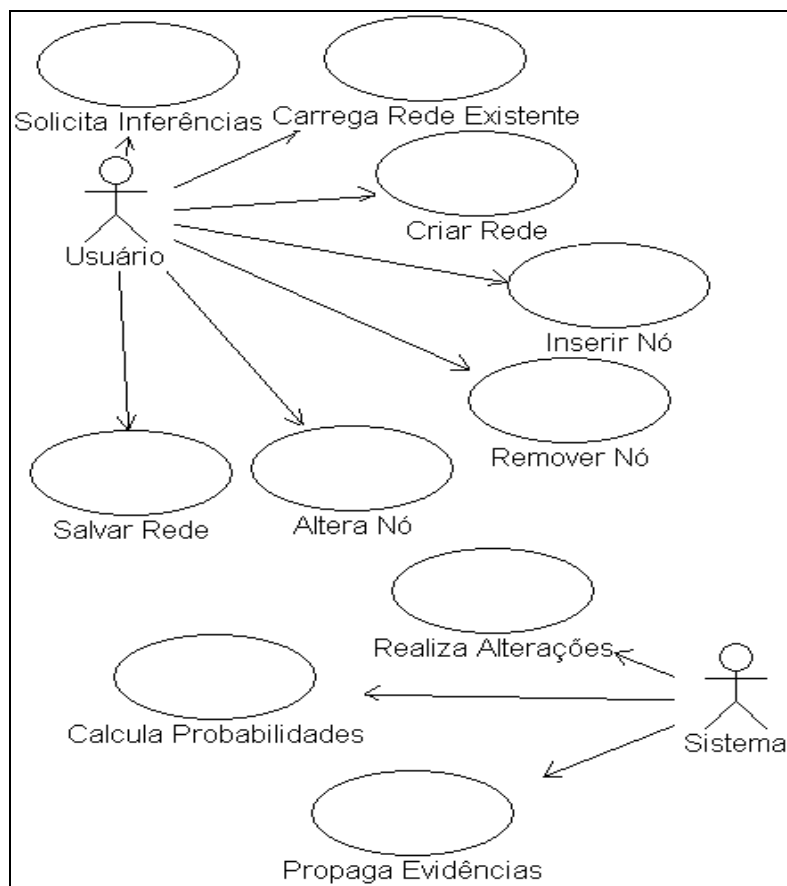


Figura 14. Ações do Sistema

Ao iniciar o DinBayes, o usuário tem dois caminhos a seguir, pode criar uma nova rede, inserindo os nós, nomes e valores da probabilidade condicional de cada um. A segunda opção é a de carregar uma rede já existente, podendo modificá-la como desejar.

Caso escolha a primeira opção, criar uma nova rede, o usuário estará criando a rede que armazenará os nós e valores das probabilidades inseridas por meio dos componentes dinâmicos da interface e da *API*.

Na segunda opção, carregar uma rede existente, o usuário pode carregar uma rede salva no disco. Os nós, seus estados e a tabela de probabilidades serão mostrados na tela, e o usuário tem a possibilidade de alterar as informações da BC utilizada.

8.4.2.1 Criação de Uma Nova Rede

Se o usuário optar pela criação de uma nova rede, automaticamente ele estará habilitando a função de construção da rede, sendo essa, o ponto de partida para a construção da base de conhecimento.

O primeiro nó da rede será escolhido por *default* como nó diagnóstico, e os demais serão seus filhos, que estarão ligados a ele por meio de um *link*. O nome do novo nó não deverá apresentar espaços e acentos, sendo essa uma restrição da *shell* Netica. Em seguida, o usuário deve escolher a quantidade de estados do nó a ser criado, e de acordo com o número escolhido, *edits* serão criados dinamicamente para que sejam inseridos os nomes dos estados. Ao confirmar os nomes, o usuário habilitará um *Grid* para a inserção dos valores das probabilidades condicionais, então o nó é inserido na base de conhecimento e inferências na interface, como mostrado na Figura 15.

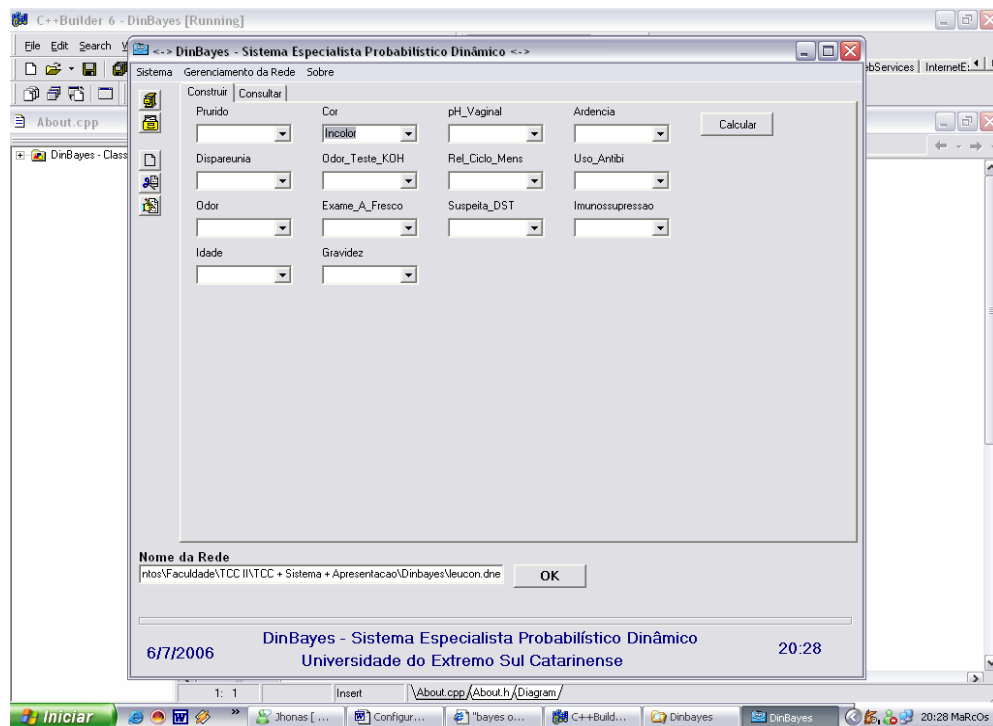


Figura 15. Realização de Inferências

Com a inserção do nó, duas novas opções são habilitadas: alteração e remoção de um nó da rede.

Escolhendo a opção de alterar um nó, será exibida uma tela solicitando o nome do nó que o usuário deseja alterar e quais características deseja modificar, sendo elas, o nome do nó e os valores das probabilidades.

Caso o usuário selecione a opção de apagar o nó, a tela exibida solicitará o nó a ser apagado, e após a confirmação, será apagado juntamente com seu *link* ao nó que apresenta relação de parentesco a ele.

Por fim, o usuário tem a possibilidade de salvar a rede criada em um arquivo, podendo esta ser carregada posteriormente.

8.4.2.2 Carregar Uma Rede Existente

A opção de carregar uma rede existente é uma das características principais do DinBayes, sendo um de seus diferenciais quando comparado a outros SEP desenvolvidos.

Caso o usuário opte por essa função, o sistema solicita a rede que deverá ser aberta. Ao selecionar o arquivo e confirmar a escolha, o usuário é levado a tela onde a rede será carregada, mostrando nesta os nós da rede juntamente com seus estados por meio de componentes visuais dinâmicos (*edit e combobox*).

A partir daí o usuário pode manipular a rede como desejar, já que o DinBayes permite, além de carregar a rede, inserir novos nós, alterá-los e conforme necessário, removê-los. Por fim, o usuário pode realizar inferências na base de conhecimento e salvar a rede com as alterações feitas.

Essas são as ações disponibilizadas ao usuário do sistema. No item a seguir, é descrito um exemplo de utilização do DinBayes.

8.4.3 Exemplo de Utilização do Sistema

Como foi descrito no item anterior, o usuário tem inicialmente as opções de carregar uma rede já existente ou criar uma nova rede. Caso o usuário opte por criar uma nova rede, a tela de construção será habilitada, permitindo a inserção de um novo nó na rede, como mostrado na Figura 16 e o protótipo da função na Figura 17.

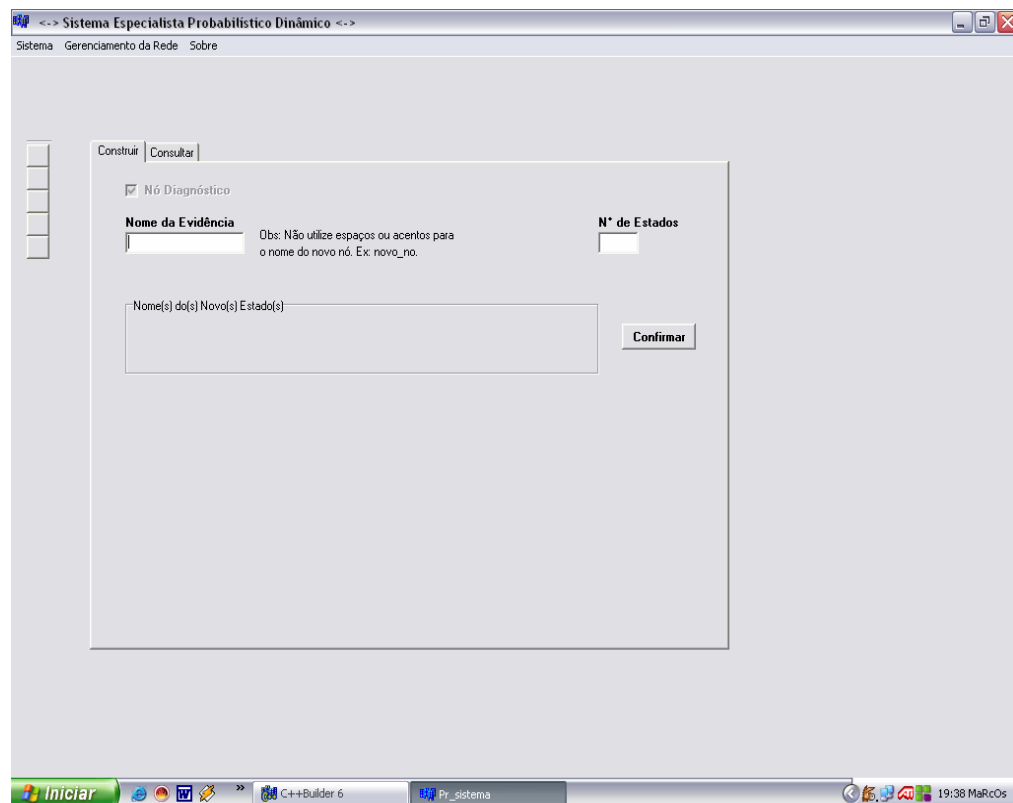


Figura 16. Tela de Construção

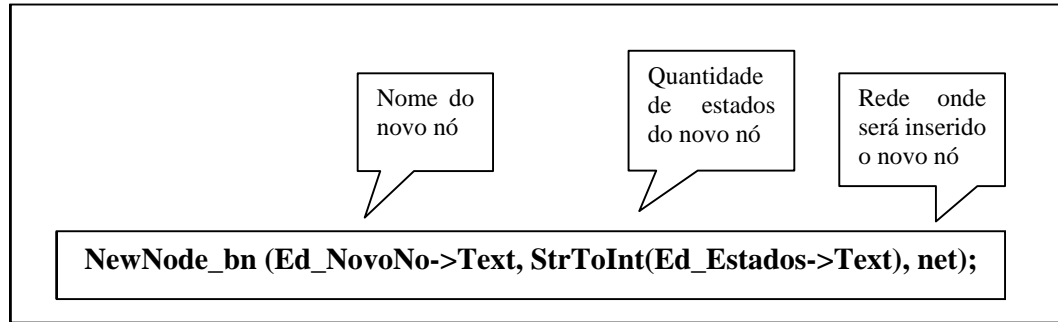


Figura 17. Comando de criação do novo nó

Nesse exemplo, o nó é criado pelo usuário durante a utilização do sistema, ficando a seu critério a escolha do nome e quantidade de estados que o novo nó possuirá.

Assim que o usuário escolher o nome e a quantidade de estados do novo nó, são criados os *edits* dinamicamente para que o usuário insira o nome de cada estado, de acordo com a quantidade escolhida, conforme mostrado na Figura 18 e implementado segundo a função da Figura 19.

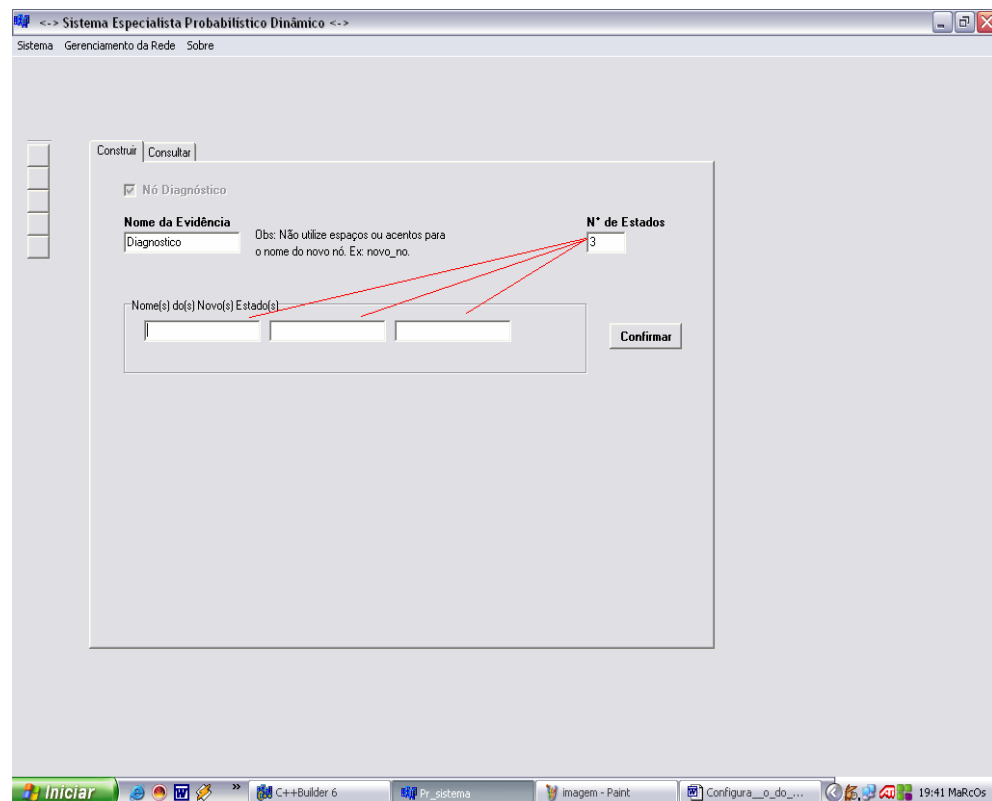


Figura 18. Criação dos Edits Dinamicamente

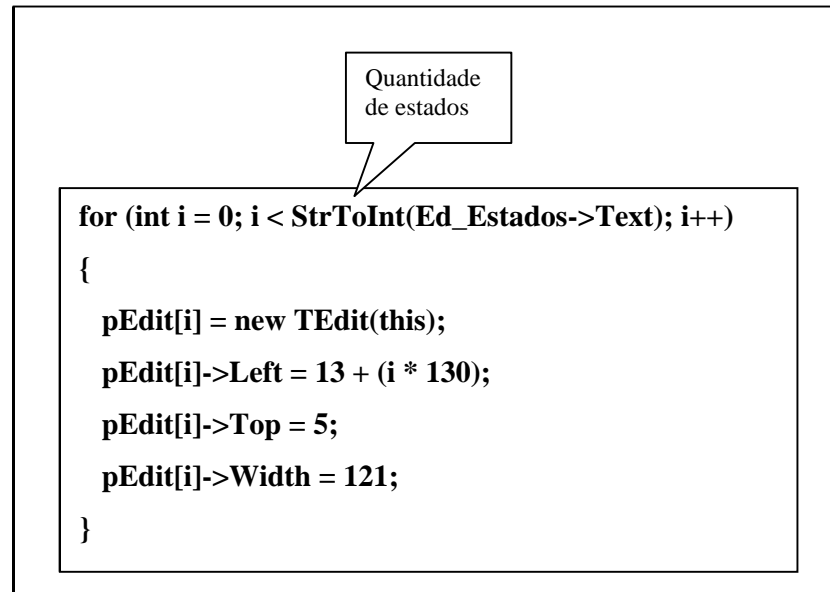


Figura 19. Comando para criação dos edits dinamicamente

Com essa função serão criados os *edits* dinamicamente, de acordo com o número de estados solicitado pelo usuário e ocorrerão os seus posicionamentos na tela.

A criação da tabela para valores das probabilidades ocorre no momento em que o usuário seleciona o botão de confirmação dos nomes dos estados do nó. A quantidade de colunas que farão parte do *Grid*, será composta de acordo com o número de estados do nó, como mostrado na Figura 20.

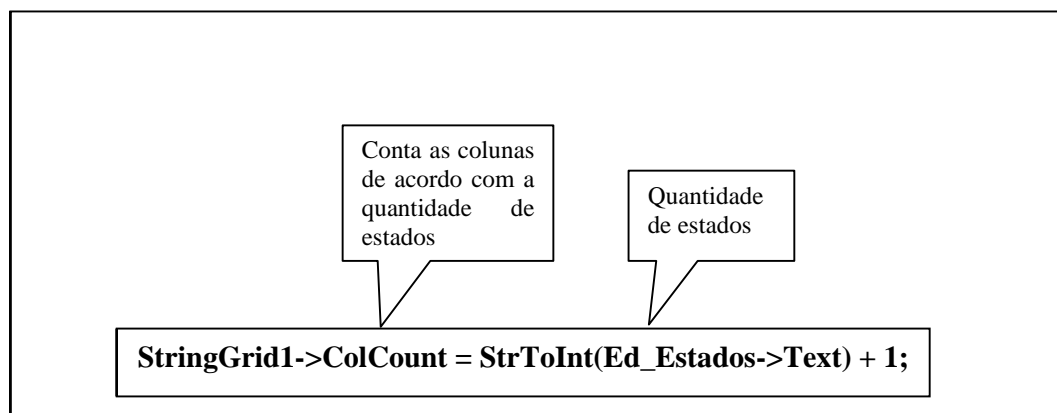


Figura 20. Comando para contagem das colunas

A atribuição do nome dos estados no Grid acontece usando a lógica usada na Figura 21:

```
for (int i = 0; i < StrToInt(Ed_Estados->Text); i++)
{
    StringGrid1->Cells[i+1][0] = pEdit[i]->Text;
}
```

Figura 21. Comando para atribuição dos nomes dos estados no grid

A seguir, na Figura 22, tem-se a tela ilustrando a inserção dos nomes no *Grid*.

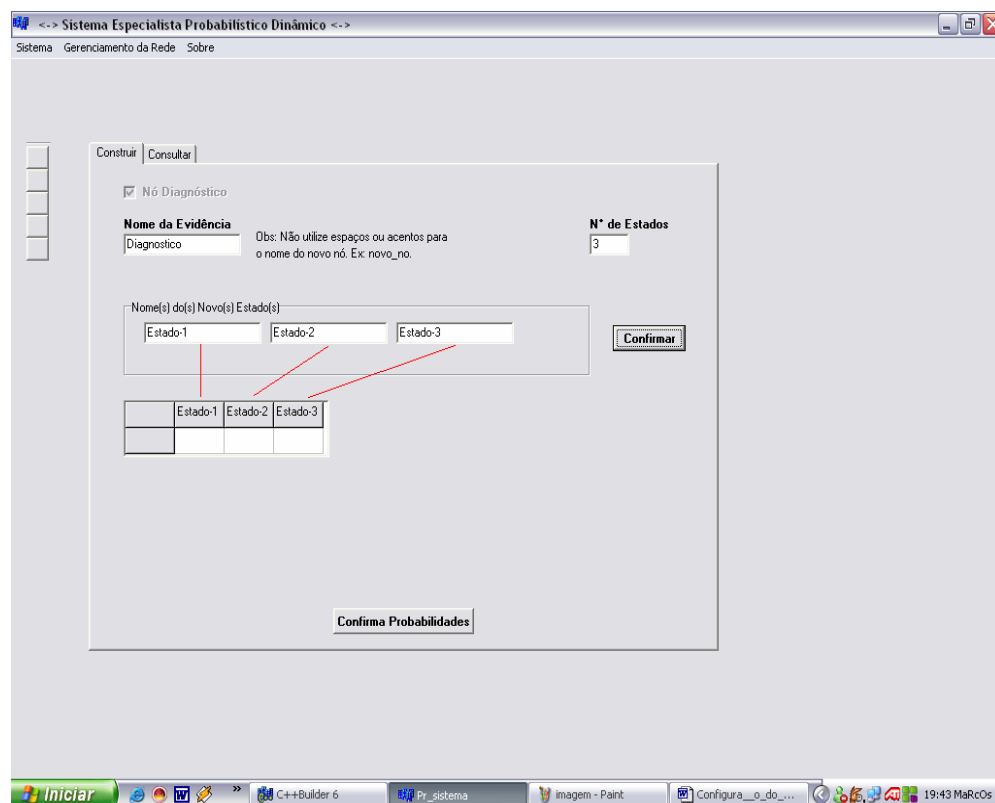


Figura 22. Inserção dos nomes no Grid

A atribuição dos nomes dos estados na rede é feita pela lógica a seguir, onde o primeiro parâmetro é o nome do nó e os posteriores os seus estados. No exemplo,

pEdit[0]->Text se refere ao nome digitado pelo usuário no primeiro *edit* criado, e em seguida o segundo nome, pEdit[1]->Text, conforme o código da Figura 23.

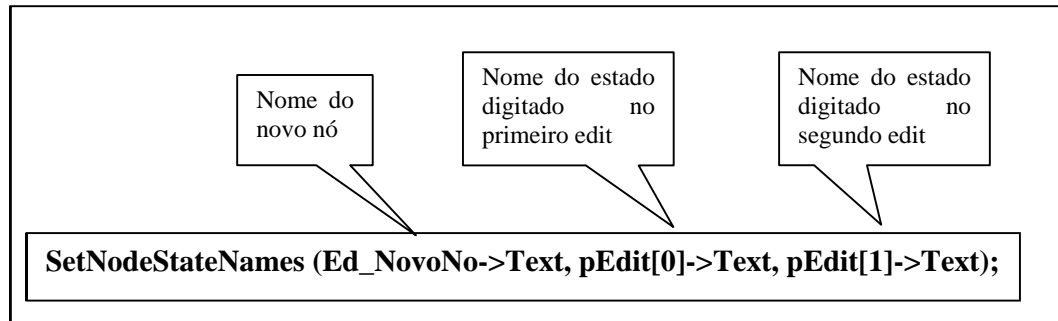


Figura 23. Comando para atribuição dos nomes dos estados do nó

Com o nó devidamente inserido na rede, o próximo passo é atribuir os valores das probabilidades condicionais. Caso o nó seja o diagnóstico, os valores inseridos serão das probabilidades *à priori*, caso contrário é preciso informar os valores das probabilidades condicionais dos estados do novo nó, de acordo com cada estado do seu pai (hipótese diagnóstica), totalizando 100%.

Na Figura 24 é exibida a tela de inserção das probabilidades de cada estado do nó, juntamente com seu protótipo na Figura 25.

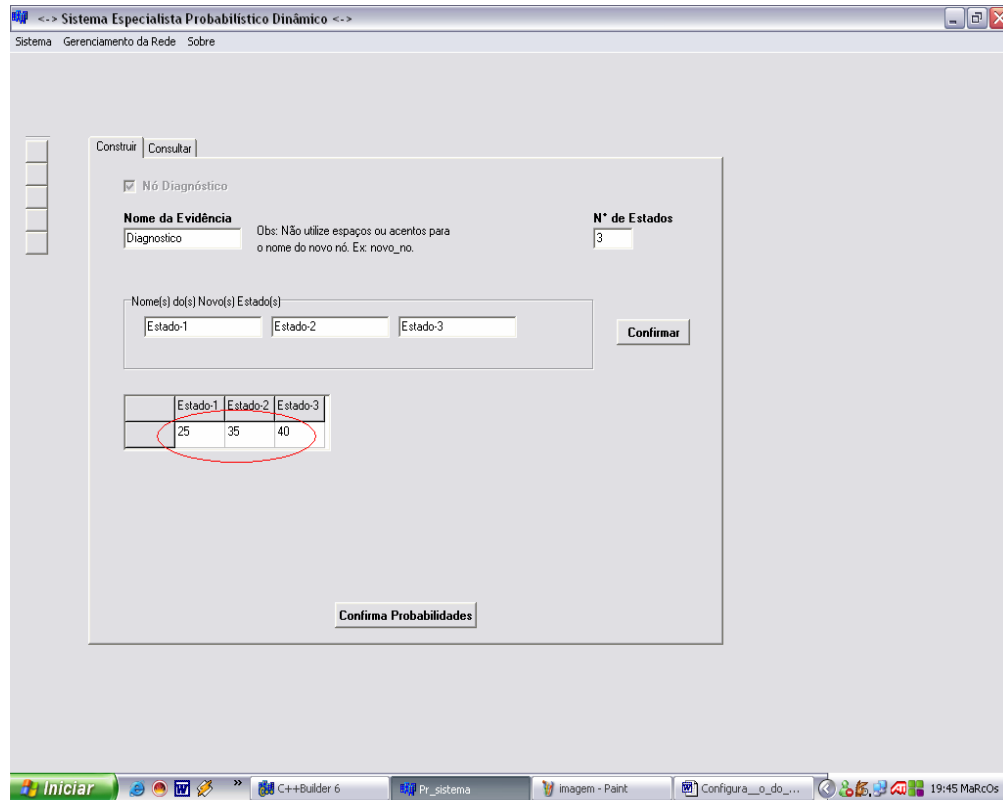


Figura 24. Inserção das probabilidades condicionais dos estados do nó

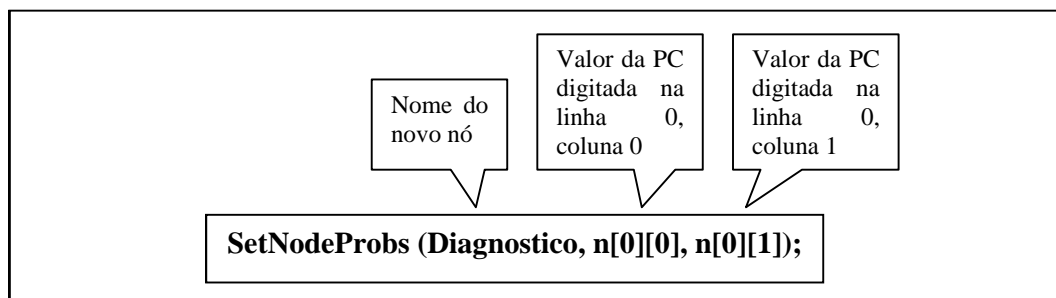


Figura 25. Comando para atribuição das probabilidades condicionais dos estados

No momento em que o usuário inicializar o sistema pela primeira vez, o primeiro nó inserido será automaticamente o nó diagnóstico. Os nós que forem inseridos após o nó diagnóstico devem ter uma relação com ele. Para que isso se torne possível, um comando para criação desse tipo de relação é utilizado, Figura 27, criando assim um *link* entre o nó diagnóstico (pai) e os demais nós da rede (filhos), como ilustrado na Figura 26.

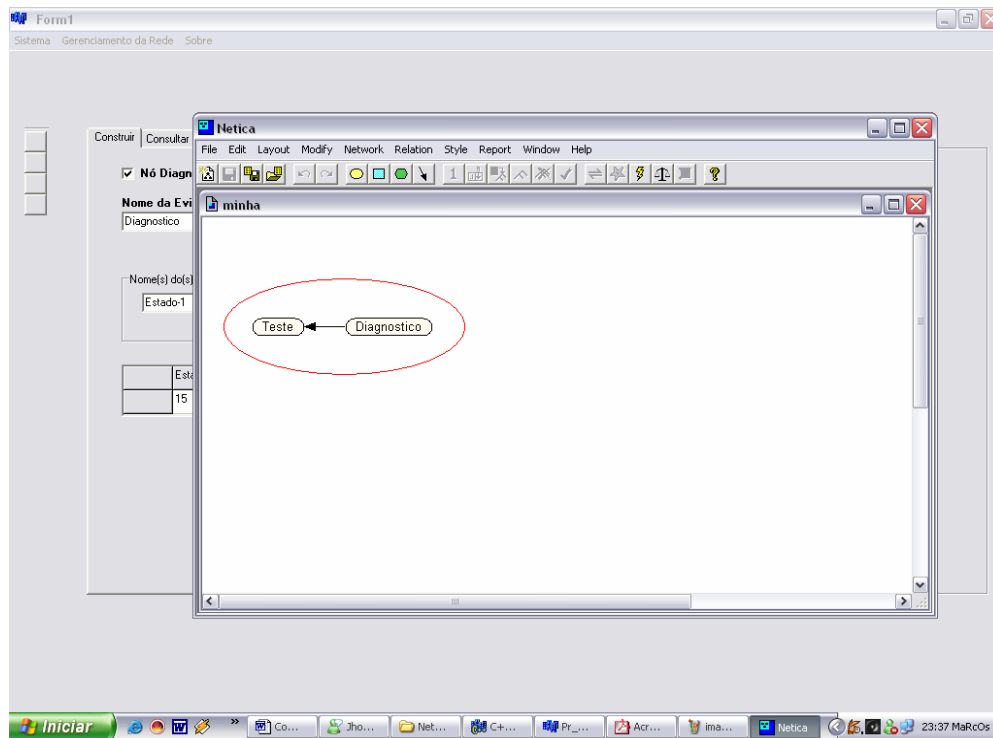


Figura 26. Criação do *link* entre os nós

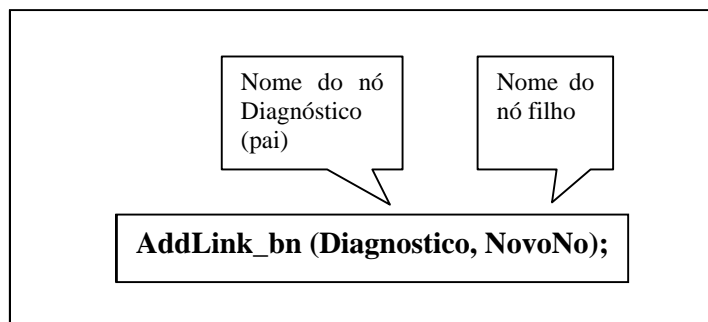


Figura 27. Comando para inserção de links entre os nós

Além de inserir nós na rede, o usuário também possui no sistema a função de apagar um nó inserido anteriormente.

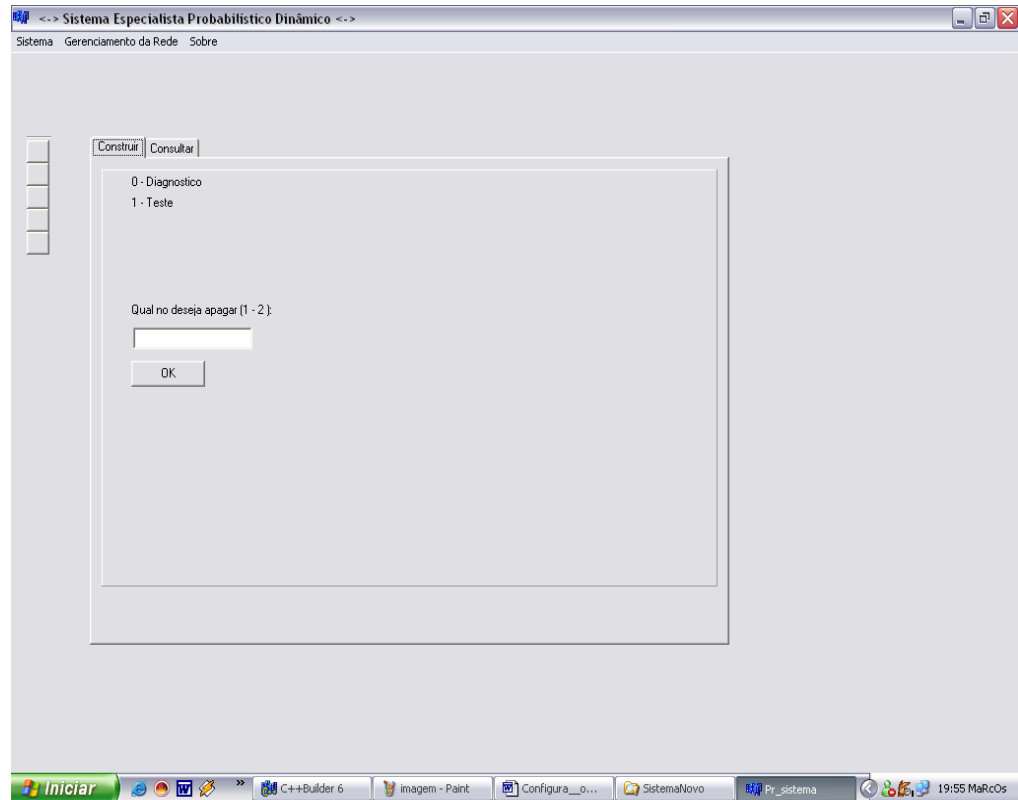


Figura 28. Apagando um nó da rede

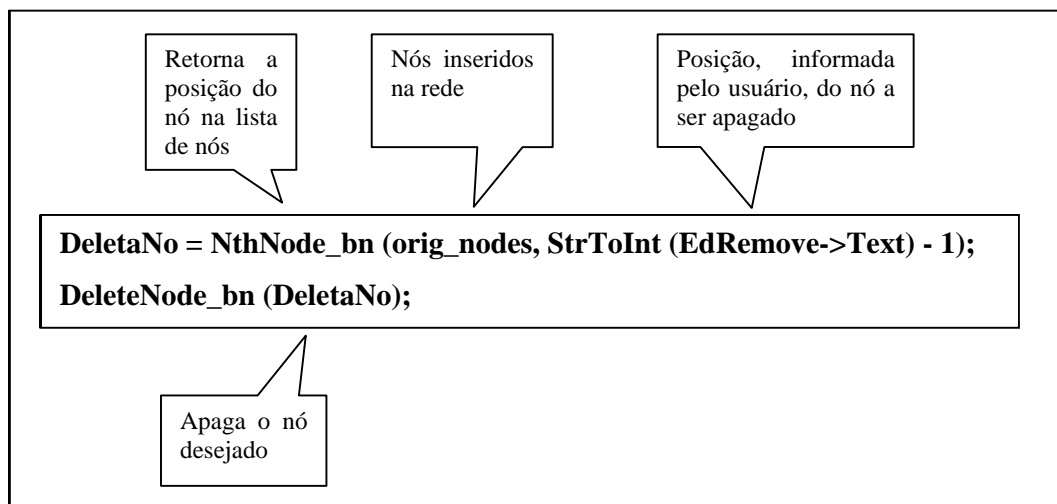


Figura 29. Comando para apagar um nó da rede

Ao selecionar a opção Apagar Nó da Rede, será exibida a tela para escolha do nó a ser apagado, (Figura 28). Essa tela apresenta um *edit* em branco, onde o usuário informa o número do nó a ser apagado. Na Figura 29, tem-se a função para apagar o nó selecionado

Ao finalizar a utilização do DinBayes, para salvar a nova base os seguintes comandos são necessários:

FileNamed_ns – retorna ao arquivo o nome dado a nova base.

WriteNet_bn – salva a rede em um arquivo.

FreeNet_bn – libera toda a memória usada pela rede e suas subestruturas.

CloseNetica_bn – fecha a rede.

Na Figura 30 tem-se os comandos necessários para salvar a rede.

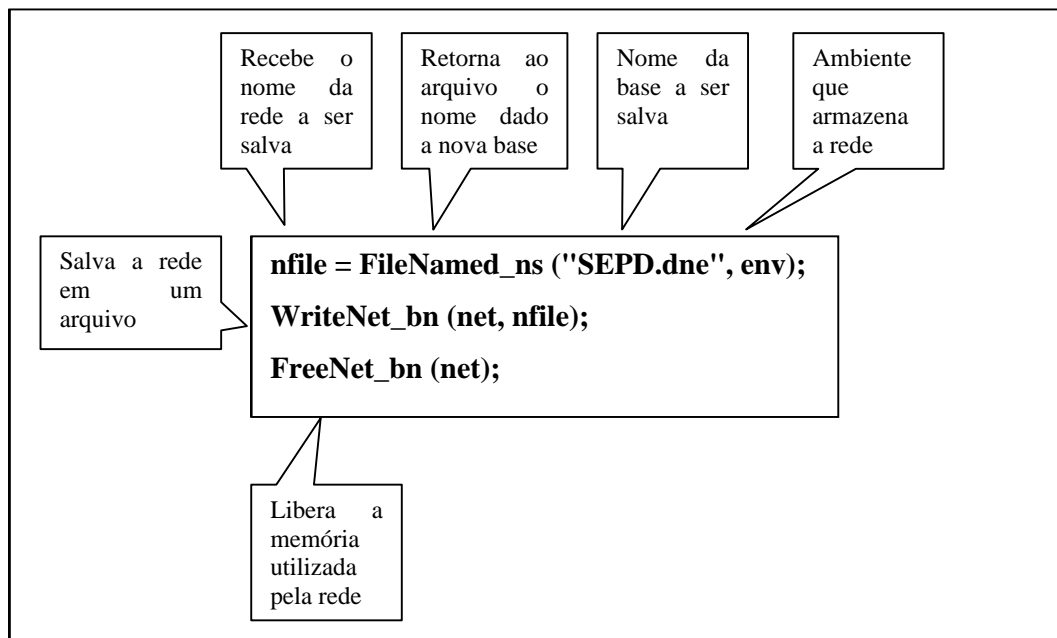


Figura 30. Comando para salvar a rede

De forma a realizar testes e utilizar a segunda parte do DinBayes, que diz respeito a utilizar e/ou alterar uma BC já existente, foi definida uma BC da área da saúde, descrita a seguir.

8.5 DEFINIÇÃO DA BASE DE CONHECIMENTO

O DinBayes, por ser dinâmico, tem o potencial de não se limitar a uma área de aplicação específica, já que uma de suas características principais é permitir que o usuário organize a base de conhecimento como desejar, inserindo ou alterando nomes e valores das probabilidades condicionais dos nós e seus respectivos estados.

Nessa etapa, optou-se pela utilização da base de conhecimento do Biowoman, que tem como objetivo auxiliar no diagnóstico da leucorréia, para a realização de testes da segunda parte do DinBayes. O motivo da escolha dessa base, se deu pelo fato de conter um bom número de informações que poderiam ser utilizadas nos testes, por ser uma base de conhecimento da área médica, onde é predominante a presença da incerteza, e por já ter sido utilizada no Biowoman que serviu de base para esse trabalho.

8.5.1 Base de Conhecimento do Biowoman

A modelagem da base de conhecimento do BioWoman possui 15 nós, sendo um para as hipóteses diagnósticas e os demais, para os sinais e sintomas, tendo como objetivo auxiliar no diagnóstico de algumas doenças relacionadas à leucorréia, encontradas no trato genital inferior da população feminina.

O nó com as hipóteses diagnósticas possui os seguintes estados:

- a) flora vaginal normal;
- b) flora mista;
- c) vaginose citolítica (aumento de lactobacilos);
- d) cocos;

- e) inflamação descamativa;
- f) candidíase;
- g) vaginose bacteriana (*Gardnerella*);
- h) trichomonas;
- i) gonococo;
- j) clamídia.

Os nós que representam os sinais e sintomas, ou seja, as evidências e seus respectivos estados estão representados na *shell* Netica (figura 31):

- a) idade, com 3 estados identificados como: do nascimento a menarca, ou seja, a primeira menstruação (excluindo essa), da menarca à menopausa, ou seja, última menstruação (excluindo essa) e a pós-menopausa;
- b) cor, que possui 9 estados (branco grumoso, incolor, amarelo purulento, amarelo, branco leitoso, amarelo esverdeado bolhoso, muco turvo, amarelo esverdeado e branco acinzentado) referentes aos possíveis tons de cores encontrados no corrimento;
- c) prurido, com 2 estados que indicam presença ou ausência do mesmo;
- d) odor, com 3 estados possíveis, que são: ausência do mesmo, presença de odor discreto ou presença de odor fétido;
- e) odor com teste de KOH³, com 2 estados que determinam se o teste deu positivo ou negativo;
- f) ardência, também com 2 estados indicando presença ou ausência desse sintoma;

³ Teste onde se utiliza uma gota de hidróxido de potássio em união com o líquido vaginal para perceber a presença de algum odor que caracterize uma doença.

- g) dispareunia (dor na relação sexual), com 3 estados, sendo um para ausência desse sintoma, um para dor no início da penetração e uma outra para dor na penetração profunda;
- h) se há suspeita ou não de Doença Sexualmente Transmissível (DST);
- i) relação com ciclo menstrual, com os estados referentes ao período pré-menstrual (fase lúteas), meio do ciclo (fase ovulatória) e pós-menstrual (fase folicular);
- j) Imunossupressão (diabete, AIDS, uso de corticóides e *stress*), apresentando influência ou não;
- k) gravidez, com 2 estados, sim ou não;
- l) utilização ou não de antibióticos;
- m) pH Vaginal, com 4 estados diferentes (menor que 3,8; entre 3,8 e 4,5; entre 4,5 e 6,8; entre 6,8 e 8,5) que determinam se o ambiente examinado está mais propício a ácido ou mais para básico;
- n) exame a fresco, com 9 estados possíveis: presença de células epiteliais e poucos lactobacilos; presença de lactobacilos em grande quantidade; presença de hifas, pseudo-hifas e esporos⁴; *clue-cells* (células-alvo); presença do trichomonas; diplococo intracelular; cocus; leucócitos; cocus e bacilos;

⁴ Hifas, pseudo-hifas e esporos são tipos de fungos que podem estar presentes no exame.

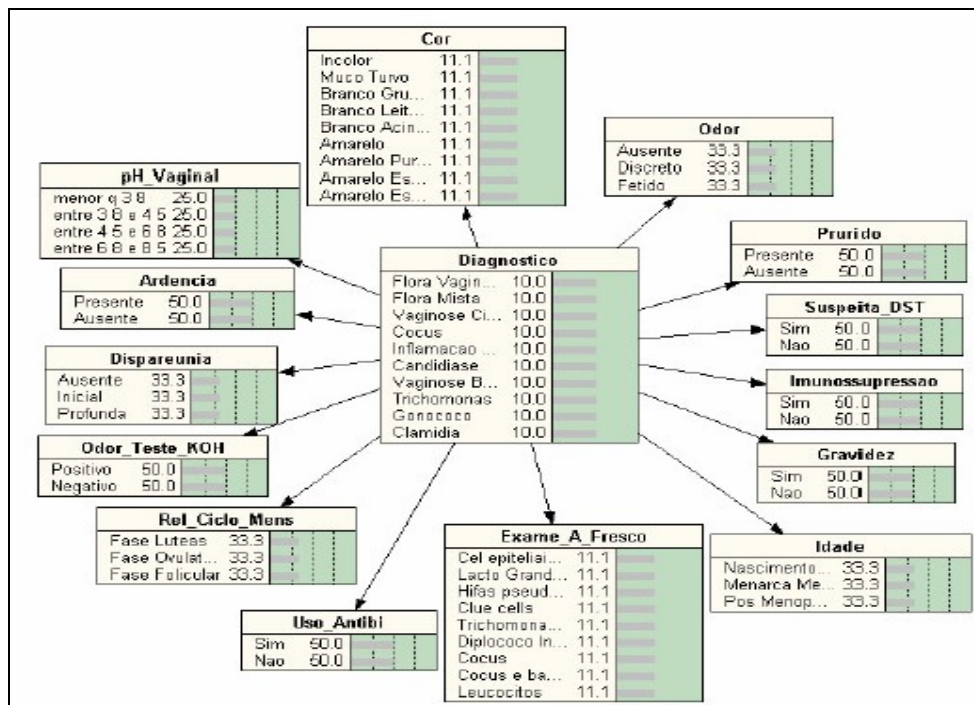


Figura 31. Base de conhecimento do *BioWoman*

Definida a base de conhecimento, conclui-se que o sistema está de acordo com o que se propôs realizar, proporcionando ao usuário as opções de criação de uma nova BC, ou de carregar e alterar uma base já existente. Os resultados obtidos por meio da utilização do DinBayes foram os mesmos do sistema BioWoman.

Finalizada a etapa de definição da base de conhecimento, foi elaborado um resumo de algumas das funções da API da *shell* Netica, as quais encontram-se no Apêndice A.

CONCLUSÃO

O DinBayes foi desenvolvido para auxiliar especialistas e engenheiros do conhecimento em suas tomadas de decisão. O protótipo do sistema não se destina a uma área de aplicação específica, pois o usuário tem a possibilidade de criar uma base de conhecimento, inserindo como desejar a quantidade de nós, estados e valores de probabilidades.

No momento da implementação da base de conhecimento foi utilizada a *shell* Netica, que permite a manipulação e testes dos dados armazenados via *API*. No desenvolvimento da interface utilizou-se o Borland C++ Builder, fazendo a integração deste com a base de conhecimento por meio da *DLL* disponível na *API* do Netica, resultando em uma interface que permite a manipulação dinâmica da BC.

O DinBayes apresenta as funções de criação de uma nova rede, inserção de nós e estados desses nós, juntamente com sua tabela de probabilidades. Permite ainda, alteração e remoção de nós da rede, bem como utilizar uma BC já existente. O DinBayes não possui restrições quanto a quantidade de nós que podem ser inseridos na rede. A única restrição está no fato da versão gratuita da *shell* Netica permitir a inserção de apenas 15 nós.

No desenvolvimento do DinBayes foram enfrentadas algumas dificuldades, principalmente na parte que o diferencia da maioria dos sistemas já implementados: o momento em que o usuário solicita carregar uma rede já existente, sendo essa uma parte pouco explorada em SEP, por isso existe uma pequena quantidade de material de referência para essa finalidade.

Outro ponto que demandou estudo, foi com relação a utilização de componentes dinâmicos na interface com o usuário, pois além do pouco material de

referência sobre o assunto, foi necessário estudar alternativas para a inserção desses componentes na interface, mantendo-a de fácil compreensão e utilização para o usuário, de forma a contemplar a usabilidade e adaptabilidade.

Com relação a trabalhos futuros, recomenda-se explorar a área de engenharia de software, a fim de criar sistemas especialistas probabilísticos mais robustos, que permitam a criação de grandes bases de conhecimento e ao mesmo tempo possuam interfaces de fácil utilização. A realização de testes junto aos usuários é uma boa maneira de analisar a usabilidade do sistema e saber o grau de dificuldade que esses encontram ao utilizá-lo.

Recomenda-se, por fim, explorar outras *API's* para construção de SEP, visto a contribuição que essa área de estudo traz para a pesquisa na área de Inteligência Artificial.

REFERÊNCIAS

- BARONE, Dante. **Sociedades Artificiais**: a nova fronteira da inteligência nas máquinas. Porto Alegre: Bookman, 2003. 332 p.
- BARRETO, Jorge Muniz. **Inteligência Artificial no Limiar do Século XXI**. 3.ed Florianópolis: Duplic, 2001. 392 p.
- BITTENCOURT, Guilherme. **Inteligência artificial**: Ferramentas e Teorias. 2.ed Florianópolis: UFSC, 2001. 362 p.
- BNG. **Bayesian Network Generator**. Disponível em:
<<http://www.mcw.edu/midas/bng.html>>. Acesso em nov. 2005.
- DIAVAL. **Sistema Especialista Bayesiano Para Ecocardiografia**. Disponível em:
<<http://www.ia.uned.es/~fjdiez/papers/puebla.html>>. Acesso em nov. 2005.
- DXPLAIN. Disponível em: <<http://www.lcs.mgh.harvard.edu/dxplain.asp>>. Acesso em nov. 2005.
- ELIAS, J. Disponível em: <<http://www.joaelias.com>>. Acesso em junho 2006.
- ERGOLIST. Disponível em: <<http://www.labiutil.inf.ufsc.br/ergolist/>>. Acesso em junho 2006.
- HUGIN. **Hugin Expert**. Disponível em: <<http://www.hugin.dk>>. Acesso em nov. 2005.
- KIC. **DXPress** Disponível em: <<http://www.kic.com>> Acesso: nov. 2005.
- KOEHLER, Cristiane. **Uma abordagem probabilística para sistemas especialistas**. 1998. 97f. Dissertação (Pós-Graduação em Ciência da Computação) – Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.
- LOSI, Nelso A. **ABS-Bayes: Sistema Especialista Probabilístico para Apoio ao Diagnóstico de Absenteísmo**. Disponível em:
<http://aspro02.npd.ufsc.br/arquivos/195000/199700/18_199797.htm?codBib=>> Acesso em: nov. 2005.
- LUGER, George F. **Inteligência Artificial**: Estruturas e Estratégias Para a Resolução de Problemas Complexos. 4. ed Porto Alegre: Bookman.

MANARIN, Daiane de Nez; **Aquisição De Conhecimento Em Sistemas Especialistas Probabilísticos Por Meio Da Descoberta Do Conhecimento Em Base De Dados Para Construção De Redes Bayesianas**. 2004. UNESC, Criciúma.

MICROSOFT BELIEF NETWORK TOOLS – **MSBN**. Disponível em: <<http://research.microsoft.com/adapt/MSBNx/>> Acesso em: nov. 2005.

NASSAR, Sílvia Modesto. “**Tratamento de Incerteza: Sistemas Especialistas Probabilísticos**”. Acessado em 01 de Set de 2005. Disponível em: <http://www.inf.ufsc.br/~silvia/disciplinas/sep/MaterialDidatico.pdf>

NETO, Eugênio R. **Sistema Especialista de Apoio à Gestão da Evasão Escolar – EBayes**. Dissertação de Mestrado, Programa de Pós-Graduação em Computação, Depto. de Informática e de Estatística, UFSC, 2002.
NORSYS. **Netica**. Disponível em: <<http://www.norsys.com>> Acesso em: nov. 2005.

PASINI, Hamilton. **Sistema Especialista Probabilístico para Apoio ao Diagnóstico de Potencial Econômico - SEPE**. Dissertação de Mestrado, Programa de Pós-Graduação em Computação, Depto. de Informática e de Estatística, UFSC, 2002.
Disponível em:
<http://aspro02.npd.ufsc.br/arquivos/180000/182800/18_182840.htm?codBib=>.
Acesso em: nov. 2005.

PEREIRA, Marcos A. **Sistema Especialista On-Line de Auxílio ao Diagnóstico de Câncer de Próstata**. Disponível em: <<http://150.162.90.250/teses/PEEL0928.pdf>>. Acesso em: nov. 2005.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995. 1056p.

RABUSKE, Renato Antônio. **Inteligência Artificial**. 1995. UFSC, Florianópolis.

RAMOS, Breno Y. K; AZEVEDO, Fábio R. A; LIMA, Maurício V. S. **Técnica de Tratamento de Incerteza - Uma abordagem Bayesiana para Sistema Especialista Probabilístico de Diagnóstico de LER – DIAGLER**. Disponível em: <<http://www.cci.unama.br/margalho/portaltcc/tcc2003/d2616.pdf>>. Acesso em: out. 2005.

RICH, Elaine; KNIGHT, Kevin. **Inteligência Artificial**. Tradução de Maria Cláudia Santos Ribeiro Ratto; revisão técnica Álvaro Antunes. São Paulo: Makron books, 1993. 722p.

RODRIGUES, Ana Carolina Braga; **Biowoman** – Base de Conhecimento Dinâmica para Sistemas Especialistas Probabilísticos. 2002. Trabalho de Conclusão do curso de Ciência da Computação Universidade do Extremo Sul Catarinense, Criciúma.

ROSA, Lilian Regina J. **Sistema Especialista Probabilístico Para Prognóstico de Doenças Bucais**. 2002. 75f. Trabalho de Conclusão do curso de Ciência da Computação – UNESC, Criciúma.

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência Artificial**. Rio de Janeiro: Elsevier, 2004. 1021 p.

SANTOS, Robson. **Alguns Conceitos Para Avaliar Usabilidade**. Disponível em: <<http://webinsider.uol.com.br/vernoticia.php/id/1771>>. Acesso em: maio. 2006.

SAVARIS, Silvana Valdemara A. M. **Sistema Especialista Para Primeiros Socorros Para Cães**. 2002. 156f. Dissertação (Mestrado no curso de Ciência da Computação) – UFSC, Florianópolis. Disponível em: <<http://www.inf.ufsc.br/~barreto/teses/savaris.pdf>>. Acesso em agosto 2005.

SIMÕES, Priscyla W.T.A. **Sistema de Apoio na Avaliação do Crescimento Infantil – SACL**. Dissertação de Mestrado, Programa de Pós-Graduação em Computação, Depto. de Informática e de Estatística, UFSC, 2001.

SOMMERVILLE, Ian. **Engenharia de software**/ Ian Sommerville. São Paulo: Addison-Wesley, 2003. 592p

SPIRIT. **Spirit Expert System**. Disponível em: <<http://www.xspirit.de>>. Acesso em nov. 2005.

STRATEGIST. **Strategist**. Disponível em: <<http://www.prevision.com/strategist.html>>. Acesso em: set. 2005.

STEIN, Carlos Efrain. **Sistema Especialista Probabilístico**: base de conhecimento dinâmica. 2000. 91f. Dissertação (Pós-Graduação em Ciência da Computação) – Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.

TOLEDO, Renata.V. **Sistema de Apoio ao Diagnóstico Diferencial de Cefaléia**. Dissertação de Mestrado, Programa de Pós-Graduação em Computação, Informática Aplicada, PUC-Paraná, 2000. Disponível em: <<http://www.inf.ufsc.br/~silvia/disciplinas/sep/MaterialDidatico.pdf>>. Acesso em: agosto 2005.

BIBLIOGRAFIA RECOMENDADA

CETENARESKI, João A. **Sistema de Monitoramento do Potencial de Risco de Infecção Hospitalar em UTI Neo-Natal Baseado em Agentes de Software.** Dissertação de Mestrado, Programa de Pós-Graduação em Tecnologia em Saúde, PUC-Paraná, 2005. Disponível em:
<http://www2.pucpr.br/multimedia/ppgts/teses/joao_andrei.pdf>. Acesso em: agosto 2005.

INTHURN, Cândida. **Qualidade & Teste de Software:** engenharia de software, qualidade de software, qualidade de produtos de software, teste de software, formalização do processo de teste, aplicação prática dos testes. Florianópolis: Visual Books, 2001. 108 p.

LINARES, Kathya S. C. **Sistema especialista nebuloso para diagnóstico médico.** 1997. 116f. Dissertação de mestrado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis.

LUNA, José Eduardo O. **Algoritmos EM para Aprendizagem de Redes Bayesianas a partir de Dados Incompletos.** Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Mato Grosso do Sul, 2004. Disponível em: <http://www.dct.ufms.br/mestrado/dissertacoes/jose_eduardo.pdf> Acesso em: Julho 2005.

NETO, Antonio Lopes de Sá; LIMA, Izaias Moreira; AFONSO, Mônica Barbosa. **Aplicação de Redes Bayesianas Para Extração de Conhecimento de Bases De Dados.** Trabalho de Conclusão de Curso de Ciência da Computação, Universidade de Amazônia, 2002. Disponível em:
<<http://www.cci.unama.br/margalho/portaltcc/tcc2002/RedesBayesianas.pdf>> Acesso em: junho 2005.

APÊNDICE

APÊNDICE A – GRUPOS DE FUNÇÕES DA API DA SHELL NETICA

As principais funções da API da *shell* Netica utilizadas para a criação de um Sistema Especialista Probabilístico Dinâmico podem ser separadas em quatro grupos, sendo estes:

- a) rede, onde ocorre o gerenciamento da rede;
- b) nó, onde temos as funções responsáveis pela inserção, alteração e remoção dos nós da rede;
- c) link, onde é criada a dependência entre os nós evidências e o nó diagnóstico;
- d) sistema, responsável pela criação do ambiente para que a rede possa ser criada, inicialização do Netica e seu encerramento.

No grupo rede, encontram-se as funções responsáveis pelo gerenciamento da rede, desde a criação e nomenclatura até a função para apagá-la totalmente.

Neste grupo podemos destacar as seguintes funções:

- a) **NewNet_bn ("Parâmetro1", Parametro2)**, onde NewNet é o nome da função de criação da rede, o primeiro argumento é o nome da rede, e o seguinte o ambiente que receberá a rede;
- b) **DeleteNet_bn (Parametro)**, onde DeleteNet é o nome da função para apagar a rede e liberar todos os recursos em uso (libera a memória) e todas as suas subestruturas (nós), do sistema, e o argumento aponta o nome da rede;

- c) **CompileNet_bn (net)** - compila a rede do Netica, ou seja, propaga as evidências de forma a obter todos os valores probabilísticos das hipóteses diagnósticas e disponibilizá-los;

```
Ex: nfile = FileNamed_ns ("SEPD.dne", env);  
    net = ReadNet_bn (nfile, FALSE);  
    CompileNet_bn (net);
```

- d) **WriteNet_bn** – Função responsável pela gravação da rede em um arquivo;

```
Ex: WriteNet_bn (net, nfile);
```

- e) **ReadNet_bn** – Essa função faz a leitura da rede a partir de um arquivo, retornando a nova leitura ou NULL se a leitura não for possível;

```
Ex: net = ReadNet_bn (nfile, FALSE);
```

- f) **FreeNet_bn** – Libera toda a memória usada pela rede;

```
Ex: FreeNet_bn (net);
```

- g) **GetNetNodes_bn** – Essa função mostra ao usuário os nós inseridos na rede;

No grupo nó, temos as funções necessárias para gerenciamento dos nós da rede, inserindo nomes e probabilidades e apagando os mesmos, se necessário.

Algumas funções que podem ser destacadas neste grupo são:

a) **NewNode_bn (Parametro1, Parametro2, Parametro3)**, sendo NewNode_bn a função para criação de um novo nó, Parametro1 é o nome dado ao novo nó, Parametro2 o número de estados do nó e Parametro3 o nome da rede onde este será inserido;

b) **SetNodeStateNames (Parametro1, Parametro2)**, onde SetNodeStateNames é a função para dar nome aos estados do nó, Parametro1 o nome do nó e Parametro2 é o nome dado ao estado do nó.

c) **GetNodeBeliefs_bn** – Essa função retorna a probabilidade do nó escolhido pelo usuário;

Ex: **belief = GetNodeBelief ("Parametro1", "Parametro2", Parametro3);**

d) **SetNodeProbs_bn** – Função responsável pela criação da tabela de probabilidades condicionais do nó, informada pelo usuário;

Ex: **SetNodeProbs (Parametro1, "Parametro2", n[0][0]);**

No exemplo, Parametro1 é o nome do nó, em seguida informa-se o estado e por fim as probabilidades desejadas, sendo n[0][0] os valores inseridos na linha 0 e coluna 0 de um Grid.

a) **SetNodeName_bn** – A função SetNodeName_bn nomeia o nó criado;

- b) **GetNodeName_bn** – Essa função retorna uma String com o nome do nó solicitado.

```
Ex: const char *nome = GetNodeName_bn (node[i]);  
pLabel->Caption = IntToStr(i) + " - " + nome;
```

No exemplo anterior, *nome receberá o nome do nó indicado pela posição node[i], onde i é o número do nó. Em seguida, o texto do label retornará a posição do nó e o seu nome.

- a) **DeleteNode_bn** – Essa função tem como objetivo apagar um nó da rede e liberar toda a memória utilizada por ele;

```
Ex: DeleteNode_bn (Parametro);
```

- b) **LengthNodeList_bn** – Com a utilização dessa função o usuário receberá o número de nós inseridos na lista dos nós da rede.

- c) **GetNodeParents_bn** – Essa função é utilizada para saber se o nó possui pais, ou seja, caso possua ele é um nó evidência, caso contrário ele é o nó diagnóstico.

- d) **GetNodeNumberStates_bn** – Essa função é responsável por retornar o número de estados do nó solicitado.

```
Ex: auxnumno = StrToInt(EdAlterar->Text);  
auxnumest = GetNodeNumberStates_bn (node[auxnumno]);
```

No exemplo anterior, auxnumest receberá o número de estados do nó informado pelo usuário, por meio do edit EdAlterar.

- e) **GetNodeStateName_bn** – Função responsável por retornar o nome do estado de determinado nó.

Em seguida temos o grupo link, onde ocorre a relação entre os nós criados na rede. Destaca-se nesse grupo as funções:

- a) **AddLink_bn (Parametro1, Parametro2)**, onde **AddLink_bn** é o nome da função para criação do link entre os nós, **Parametro1** e **Parametro2**;

Ex: **AddLink_bn (Diagnostico, NovoNo)**;

- b) **DeleteLink_bn (Diagnostico, NovoNo)**, onde **DeleteLink** é o nome da função para deletar o link entre os nós diagnóstico e **NovoNo**.

Por último temos o grupo sistema, responsável pelo gerenciamento do ambiente de rede, destacando-se as seguintes funções:

- a) **NewNeticaEnviron_bn (NULL)**, onde **NewNeticaEnviron_bn** é a função para criar o ambiente que receberá a rede e **NULL** é utilizado quando a versão do Netica for gratuita, caso contrário será substituído pela senha do usuário;

Ex: **env = NewNeticaEnviron_bn ("SENHA")**;

- b) **InitNetica_bn (&env, msg)**, sendo **InitNetica_bn** a função responsável pela inicialização do Netica, **env** é o ambiente de rede criado e **msg** informará uma mensagem caso a inicialização não seja possível.

c) **FileNamed_ns** – Retorna ao arquivo o nome dado a nova base;

Ex: **nfile = FileNamed_ns ("SEPD.dne", env);**

d) **GetNeticaVersion_bn** – Informa ao usuário a versão da *Shell* Netica utilizada multiplicada por 100, ou seja, se a versão utilizada for a 1.21 então o valor retornado será 121;

Ex: **GetNeticaVersion_bn (env, &version);**
printf ("Versão atual do Netica: %s\n", version);

e) **EnterFinding_bn** – No momento de uma pesquisa em que o usuário escolherá o estado de uma evidência, a função **EnterFinding_bn** realiza uma atualização da rede, renovando seus valores de acordo com o estado escolhido;

Ex: **EnterFinding ("NovoNo", "Estado1", net);**

f) **CloseNetica_bn** – A função **CloseNetica_bn** é responsável pelo fechamento da rede;

Ex: **res = CloseNetica_bn (env, mesg);**

**APÊNDICE B – EXEMPLOS DE SISTEMAS ESPECIALISTAS
PROBABILÍSTICOS COM BASE DE CONHECIMENTO ESTÁTICA**

Técnica de Tratamento de Incerteza - Uma Abordagem Bayesiana Para Sistema Especialista Probabilístico de Diagnóstico de LER – Diagler

O presente trabalho de conclusão de curso, desenvolvido na Universidade da Amazônia (UNAMA), denominado DIAGLER, tem como objetivo diagnosticar os tipos mais comuns de Lesões por Esforços Repetitivos (LER) nos membros superiores, e determinar o grau da lesão em que o paciente pode estar acometido, com base nos conhecimentos de redes bayesianas (RAMOS; AZEVEDO; LIMA, 2003).

O sistema foi motivado pela real necessidade de um conhecimento especialista na área médica que possibilitasse o preenchimento do campo “hipótese diagnóstica inicial” da notificação de acidente de trabalho no Sindicato de Processamento de Dados do Estado do Pará (SINDPD/PA) (RAMOS; AZEVEDO; LIMA, 2003).

O sistema especialista foi implementado com auxílio da *Shell* Netica, e pela necessidade de interação do usuário com o sistema, foi utilizado o ambiente de programação Delphi na implementação da interface, que permite a manipulação da base de conhecimento via *DLL*. Uma das telas do sistema pode ser vista na Figura 32 mostrada a seguir (RAMOS; AZEVEDO; LIMA, 2003).

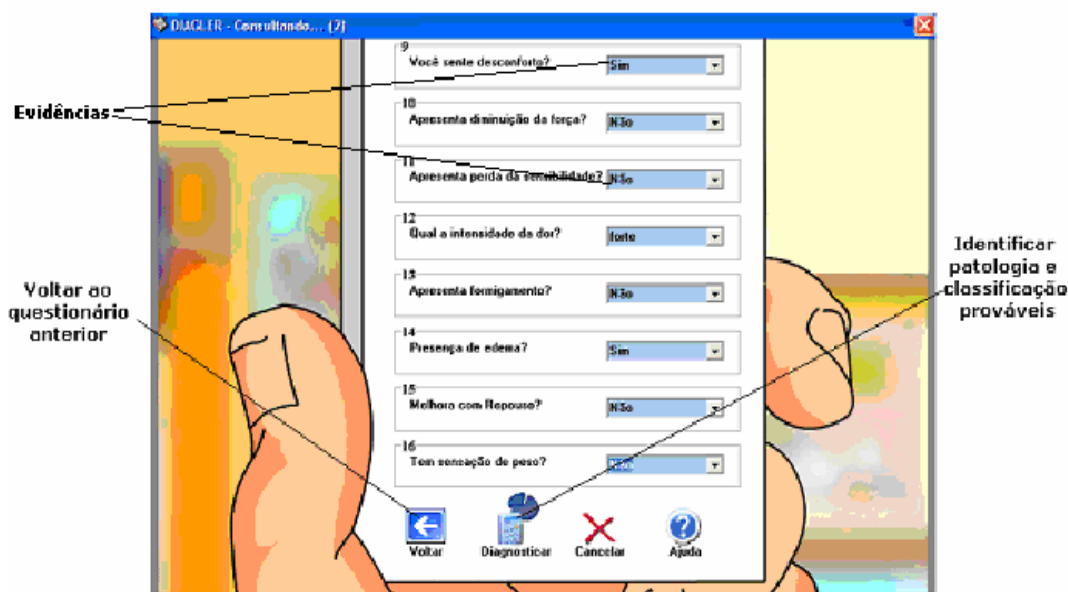


Figura 32. Tela de consultas do sistema DIAGLER
 Fonte: RAMOS, B; AZEVEDO, F; LIMA, M (2003)

Sistema de Avaliação da Evasão Escolar – E-Bayes

Este trabalho foi desenvolvido na UFSC, a título de especialização e utiliza uma rede bayesiana para apoiar o gestor de uma universidade a avaliar a evasão escolar de um curso universitário. Mostra o cenário da evasão discente, permitindo fazer a previsão da permanência de um aluno no curso, a partir de suas características sócio-demográficas, buscando na base de dados quais os alunos do curso que são similares às características do aluno em questão, explorando dessa forma a aplicação de sistemas especialistas em sistemas de informação (ROVARIS NETO, 2002),.

O autor conseguiu apresentar um modelo de rede bayesiana capaz de retratar a situação acadêmica atual do curso em estudo, bem como, os motivos que podem levar os alunos a evadirem ou permanecerem em seu curso. Afirma que o sistema criado não foi implementado para substituir o especialista humano, mas sim oferecer suporte ao mesmo, auxiliando-o no processo decisório da organização, assumindo, assim, uma

característica que é clara dos Sistemas de Apoio à Decisão (SAD⁵). Por fim, conclui que o *E-Bayes*, que foi desenvolvido utilizando a *shell* Netica e técnicas de *data mining*, vem demonstrando ser uma ferramenta capaz de quantificar e qualificar os dados transformando-os em informações e gerando conhecimento capaz de interferir nos SAD (ROVARIS NETO, 2002).

Sistema Especialista Probabilístico Para Prognóstico de Doenças Bucais – PROBUCAL

O presente trabalho de conclusão de curso da UNESC consiste no desenvolvimento de um sistema especialista probabilístico com aplicação na área odontológica, denominado Sistema Especialista Probabilístico para Prognóstico de Doenças Bucais – PROBUCAL.

O sistema objetiva realizar o prognóstico de algumas doenças bucais, como cárie, gengivite, periodontite e câncer, baseando-se em algumas evidências, sendo que no desenvolvimento deste, integra-se a base de conhecimento estática implementada com o auxílio da *shell* Netica e o ambiente de programação Delphi, como pode ser observado na Figura 33. O sistema foi desenvolvido devido a necessidade odontológica de identificação de doenças por meio do prognóstico, de forma a permitir a identificação de casos com tendência em apresentar algumas doenças bucais, e identificar as deficiências a serem supridas para preservação da saúde bucal (ROSA, 2002).

⁵SAD é um sistema de informação que apoia qualquer processo de tomada de decisão em áreas de planejamento estratégico, controle gerencial e controle operacional.

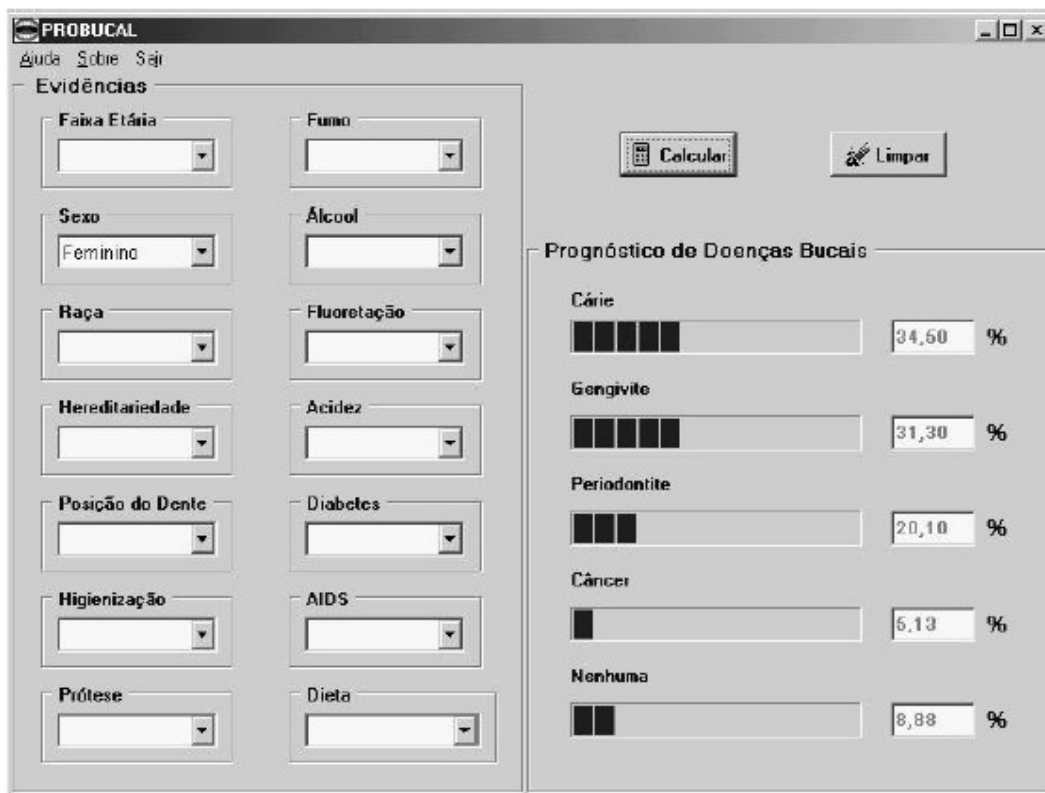


Figura 33. Prognóstico para pessoa do sexo feminino no PROBUCAL
Fonte: ROSA, L (2002)

Sistema de Avaliação do Crescimento Infantil – SACI

O presente trabalho desenvolvido a título de mestrado pela acadêmica da UFSC (SIMÕES, 2001), buscou o desenvolvimento de um Sistema Especialista Probabilístico para o diagnóstico de alguns distúrbios associados à falência do crescimento em crianças com até 2 anos. Contempla uma rede bayesiana composta de um nó para representar as hipóteses diagnósticas (Deficiência de Ferro, Má Absorção e Etiologia a Esclarecer), e outros três nós para representar as evidências necessárias para chegar às hipóteses diagnósticas (Diarréia, Taxa de Ferritina e D-Xilosemia) (SIMÕES, 2001).

O SACI permite que o diagnóstico seja realizado, disponibilizando os dados relacionados à evolução do crescimento da criança graficamente, e que sejam

investigadas as suspeitas de perda de peso e altura da criança, por meio de evidências a serem observadas. Na avaliação realizada junto a médicos pediatras, especialistas do domínio de conhecimento, verificou-se que o SACI é de fácil utilização, apresentando resultados satisfatórios às diversas consultas realizadas (SIMÕES, 2001).

Foi desenvolvido para a plataforma Windows, no ambiente Delphi 5, utilizando a *DLL* da *shell* Netica para esse ambiente de programação, como mostrado na Figura 34 (SIMÕES, 2001).

A imagem mostra a interface do sistema SACI Pro-Clípeo. No topo, há uma barra de título com o nome do sistema e ícones de janela. Abaixo, há uma barra de menu com 'Sistema' e 'Ajuda'. O formulário principal contém os seguintes elementos:

- Um menu suspenso para 'Sexo' com o valor 'Masculino' selecionado.
- Dois menus suspensos: 'Diagnóstico Preliminar - SISPAN' com o valor 'Ausência de Desnutrição' e 'Suspeita de perda de peso e/ou altura' com o valor 'Sim'.
- Uma seção intitulada 'Informe os 3 últimos medidos.' contendo três painéis de entrada de dados:
 - 1ª Medida:** Idade (1 mês, 0 dias), Peso (3.4 kg), Altura (52 cm).
 - 2ª Medida:** Idade (2 meses, 0 dias), Peso (3.8 kg), Altura (54 cm).
 - 3ª Medida:** Idade (5 meses, 0 dias), Peso (4.8 kg), Altura (64 cm).
- Um botão de seta vermelha no canto inferior direito.
- Na barra de status inferior, há o texto 'UFSC - Universidade Federal de Santa Catarina' e a data '28/1/2003'.

Figura 34. Tela de suspeitas e análise recorrente do SACI
Fonte: SIMÕES, P (2001)

Sistema de Apoio ao Diagnóstico Diferencial de Cefaléia

Trabalho desenvolvido a título de mestrado na PUC do Paraná, que oferece apoio a médicos generalistas, plantonistas ou residentes no diagnóstico diferencial de cefaléias (dores de cabeça). Este sistema utiliza o enfoque de rede bayesiana e sua base de conhecimento foi construída considerando os Critérios de Classificação da Sociedade Internacional de Cefaléias (IHS), levando em conta os sinais e sintomas dos pacientes,

nos valores estimados de probabilidades fornecidos pelos especialistas que participaram do projeto (TOLEDO, 2000).

A avaliação deste sistema foi realizada comparando as respostas fornecidas pelos especialistas e as respostas fornecidas pelo sistema, a partir de um conjunto de fichas médicas de pacientes com cefaléia selecionados aleatoriamente pelos especialistas do projeto. Os resultados experimentais indicam que o sistema foi capaz de fornecer os mesmos diagnósticos que os especialistas do projeto em 95% dos casos. Por outro lado, os mesmos casos clínicos foram avaliados por médicos generalistas e verificou-se que estes obtiveram um percentual de acerto de 53% (TOLEDO, 2000).

Portanto, o sistema desenvolvido utilizando a *shell* Netica apresenta um desempenho muito bom ao realizar diagnóstico diferencial de cefaléias (TOLEDO, 2000).

SISPAN - Sistema Especialista Probabilístico de Apoio a Avaliação Nutricional.

O SISPAN foi um trabalho desenvolvido a título de mestrado na UFSC, o qual realiza a avaliação nutricional de crianças com até dois anos de idade. Considera como entrada os dados relativos a peso, altura, idade, cabelo, pele, unha, hipotrofia e edema e assim, classifica o estado nutricional da criança em, sem desnutrição ou com desnutrição, podendo ser leve, moderada ou severa quanto à intensidade, em marasmo, kwashikor e marasmo-kwashikor quanto ao tipo (KOEHLER, 1998).

O SISPAN possui uma base de dados estática e se mostrou como uma ferramenta eficaz no ensino e diagnóstico médico, pois conduziu a médica especialista a repensar o seu raciocínio clínico e a considerar uma forma melhor para repassar o seu conhecimento aos alunos (KOEHLER, 1998).

Na programação da interface, que pode ser observada na Figura 35, foi utilizada a linguagem de programação C++ Builder 3.0, para o armazenamento da base de conhecimento foi utilizada a *shell* Netica 1.0, sendo que a base de conhecimento foi adquirida por meio de um grande número de entrevistas com o médico especialista na área de Nutrição (KOEHLER, 1998).

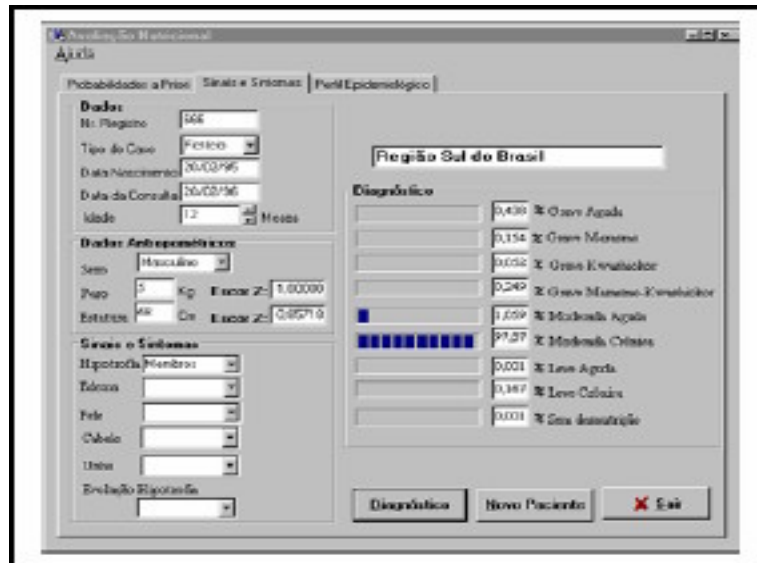


Figura 35. Interface do SISPAAN
Fonte: KOEHLER, C (1998)

DIAVAL

O DIAVAL é o resultado de uma tese de doutorado do Departamento de Informática e Automática - U.N.E.D em Madrid na Espanha, e tem 2 objetivos principais, contribuir para o estudo de redes bayesianas e aplicar redes bayesianas na construção de um sistema especialista para diagnóstico de ecocardiografia (DÍEZ, 1994).

Destina-se a ajudar médicos no diagnóstico de doenças cardíacas, calculando a probabilidade a posteriori, selecionando a mais provável como a doença mais relevante, gerando um relatório impresso, a partir, principalmente, de dados

ecocardiográficos. Tem como características a introdução dos dados, apresentação dos resultados do diagnóstico e explicação sobre o raciocínio (DÍEZ, 1994).