

**UNIVERSIDADE CATÓLICA DE GOIÁS
DEPARTAMENTO DE COMPUTAÇÃO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**



**UM MÓDULO PARA FUSÃO DE DADOS UTILIZANDO REDES
BAYESIANAS.**

ALEXANDRE BELLEZI JOSÉ

NOVEMBRO
2004

**UNIVERSIDADE CATÓLICA DE GOIÁS
DEPARTAMENTO DE COMPUTAÇÃO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**UM MÓDULO PARA FUSÃO DE DADOS UTILIZANDO REDES
BAYESIANAS.**

Trabalho de Projeto Final de Curso II apresentado por Alexandre Bellezi José à Universidade Católica de Goiás, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação, aprovado em 17/12/2004, pela Banca Examinadora:
Professor Talles Marcelo Gonçalves de Andrade Barbosa, MsC. UCG – Orientador
Professor Iamar Zuza de Araújo, MsC. UCG
Professor José Olímpio Ferreira, Dr. UCG

GOIÂNIA
NOVEMBRO
2004

UM MÓDULO PARA FUSÃO DE DADOS UTILIZANDO REDES BAYESIANAS.

ALEXANDRE BELLEZI JOSÉ

Trabalho de Projeto Final de Curso II apresentado por Alexandre Bellezi José à Universidade Católica de Goiás, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Professor Talles Marcelo
Gonçalves de Andrade Barbosa,
MSc. – Orientador

Professor José Luiz de Freitas Júnior, Dr.
Coordenador de Projeto Final de Curso

A meus pais pelo amor e
compreensão.

“O SENHOR Deus modelou o
homem com o pó apanhado do
solo. Ele insuflou nas suas narinas
o hálito da vida, e o homem se
tornou um ser vivo.”

Gênesis 2:7

AGRADECIMENTOS

Ao Professor Talles Marcelo Gonçalves de Andrade Barbosa, orientador acadêmico, pelo apoio e confiança depositada. Aos professores Ivens Gervásio Sene Jr., José Olímpio Ferreira e Olegário Correa da Silva pela valorosa colaboração e incentivos. E aos meus colegas do Grupo de Estudos em Computação Aplicada na Medicina pelas críticas e sugestões.

RESUMO

O seguinte trabalho tem como objetivo a construção de um módulo capaz de executar a Fusão de Dados utilizando como técnica principal as Redes Bayesianas. Como aplicação do trabalho propõe-se a automatização do Interrogatório Sintomatológico, parte componente da anamnese, um procedimento médico.

Palavras Chave: Pervasive Computing, Wearable Computing, Fusão de Dados, Técnicas para Fusão de Dados, Inteligência Artificial, Interrogatório Sintomatológico, Redes Bayesianas.

UM MÓDULO PARA FUSÃO DE DADOS UTILIZANDO REDES BAYESIANAS.

SUMÁRIO

LISTA DE TABELAS	IX
LISTA DE ABREVIATURAS E SIGLAS	X
INTRODUÇÃO.....	11
CAPÍTULO 1	12
1.1 <i>Pervasive Computing</i> no Cuidado da Saúde	12
1.2 Sistema de Monitoramento da Saúde Humana.....	12
1.3 Definição de Fusão de Dados	13
1.4 O Interrogatório Sintomatológico.....	14
1.5 Sumário e Conclusões	15
CAPÍTULO 2	16
2.1 Modelando a Incerteza: Redes Bayesianas.....	16
2.2 Formalização das Redes Bayesianas	16
2.3 Etapas para Criação Manual de Redes Bayesianas	17
2.4 Propagação de Evidências	18
2.5 Ferramentas para Modelagem de Redes Bayesianas.....	19
2.5.1 Microsoft Belief Networks	19
2.6 O TradutorDSC	21
2.7 Suporte a Decisão.....	22
2.8 Coleta de Evidências: Algoritmos em Grafos	23
2.8.1 Busca em Profundidade	24
2.8.2 Heurística Proposta.....	24
2.9 Sumário e Conclusões	27
CAPÍTULO 3	29
3.1 Modelando o Interrogatório Sintomatológico	29
3.2 Implementando o Suporte a Decisão.....	31
3.2.1 A Classe TradutorDSC	31
3.2.2 A Classe Nodo	32
3.2.3 A Classe GAD	34
3.3 O Protótipo: BProfundidade.java.....	36
3.4 O Protótipo: BHeuristica.java.....	39
CAPÍTULO 4	43
4.1 Considerações Finais.....	43
4.2 Propostas para Trabalhos Futuros	43
REFERÊNCIAS BIBLIOGRÁFICAS	46
ANEXOS	48
Anexo A – Interrogatório Sintomatológico.....	48
Anexo B – TradutorDSC.java	56
Anexo C – Nodo.java	61
Anexo D – GAD.java	64
Anexo E – BProfundidade.java	66
Anexo F – Bheuristica.....	70
Anexo G – Principal.java	75

LISTA DE FIGURAS

Figura 1 – Exemplo de rede Bayesiana.	17
Figura 2 – Exemplo da Estrutura Gerada pela Ferramenta.	19
Figura 3 – Representação do nodo no arquivo dsc.	20
Figura 4 – Representação da tabela de probabilidade do nodo.	20
Figura 5 – Diagrama de Classes	22
Figura 6 – Busca em Profundidade	24
Figura 7 – Visão de Alto Nível do Domínio da Fusão de Dados	25
Figura 8 – Heurística para Busca no Grafo, Passo 1	26
Figura 9 – Heurística para Busca no Grafo, Passo 2	27
Figura 10 – Tipo de Problema Estudado e Abordagem Utilizada.....	28
Figura 11 – Parte do Interrogatório Sintomatológico.....	29
Figura 12 – Representação Gráfica da Figura 11.	30
Figura 13 – Propriedades do Nodo e Exibição das Evidências.	30
Figura 14 – Diagrama de Classes	32
Figura 15 – Estrutura de Dados da Classe Nodo.....	32
Figura 16 – Estrutura de Dados da Classe GAD.	35
Figura 17 – Primeira Pergunta ao Usuário.	36
Figura 18 – Lista de Evidências para o Usuário.....	36
Figura 19 – Nodo com Evidências Tipo “Sim ou Não”.	37
Figura 20 – Nodo com Evidências Diversas.	37
Figura 21 – Tela de Questionamento sobre o Final das Perguntas.....	38
Figura 22 – Tela de Evidências Coletadas pelo Protótipo.....	39
Figura 23 – Sugestão de Evidência para Avaliação pelo Usuário.....	40
Figura 24 – Representação do Grafo em Níveis Estruturados.....	41
Figura 25 – Representação do Grafo sem os Níveis Estruturados.	42
Figura 26 – Módulos Plugáveis.	44

LISTA DE TABELAS

Tabela 1 – Técnicas Comuns em Fusão de Dados.	23
---	----

LISTA DE ABREVIATURAS E SIGLAS

EARSeL	European Association of Remote Sensing Laboratories
JDL	Joint Directors of Laboratories
UML	Unified Modeling Language
WIMP	Windows-Icons-Menus-Pointers

UM MÓDULO PARA FUSÃO DE DADOS UTILIZANDO REDES BAYESIANAS.

INTRODUÇÃO

A evolução tecnológica relacionada aos dispositivos computacionais torna possível que vários desses sejam inseridos no espaço pessoal do usuário [1], possibilitando assim que a qualquer momento o usuário de tais dispositivos possa ter informações sobre o ambiente que o cerca, assim como sobre o próprio corpo [1,10,11,12,16]. Tais aparatos tecnológicos começam a ser tornar cada vez mais reais com o desenvolvimento da *Pervasive Computing* tecnologia que busca a computação onipresente [8] tornando possível pensar em sistemas vestíveis. Tais sistemas podem possuir um ou inúmeros dispositivos em sua arquitetura, e cada um a sua maneira gera uma determinada informação para o sistema. Assim o uso de técnicas para Fusão de Dados garante o co-relacionamento de tais informações vindas de inúmeras fontes [1,2,3,5], criando o suporte a decisão. Este trabalho, visa o uso de uma técnica específica da Fusão de Dados, as Redes Bayesianas, com o intuito de gerar o suporte a decisões baseadas no que se coleta através da realização do Interrogatório Sintomatológico que é uma parte componente da anamnese utilizada pelos médicos.

CAPÍTULO 1

1.1 *Pervasive Computing* no Cuidado da Saúde

Pervasive Computing é a integração de poder de processamento (microprocessadores) e sensoriamento (sensores) em qualquer tipo de objeto, não somente em objetos tradicionais como computadores, assistentes pessoais digitais (PDA), mas sim em objetos do dia a dia como brinquedos, casas, móveis, etc. Tornando assim mais real a computação onipresente [8].

O uso da *Pervasive Computing* no cuidado da saúde é uma de suas aplicações mais importantes, podendo dar suporte a estilos de vida diferentes, assim como sendo um auxiliar no controle de doenças, um exemplo de tal tipo de sistema responsável por ajudar pacientes diabéticos pode ser visto em [17]. Com o desenvolvimento de sensores é possível obter informações relacionadas à saúde do usuário mesmo fora dos hospitais, ou seja, a informação pode ser colhida durante o dia a dia do usuário. Durante a elaboração deste trabalho será considerado que a *Pervasive Computing* também pode ser implantada em objetos pessoais como roupas, assim sempre que for feita uma referência a sistema vestível deve-se entender um sistema que faz uso da definição da *Pervasive Computing* embutida em algo que o usuário utiliza durante seu dia a dia, como por exemplo um relógio de pulso, um PDA, colete com sensores, telefone celular, etc.

Um sistema vestível ou *Wearable System* tem como uma de suas principais características a capacidade de dar mobilidade ao seu usuário, assim como garantir que o usuário do sistema tenha em mãos as informações pertinentes ao que o sistema se propõe a monitorar. Assim, devido a grande proximidade entre usuário e sistema, surgiu a idéia de usar os sistemas vestíveis para monitorar sinais vitais de seus usuários [12,16], podendo então torná-los aliados no tratamento diário das mais diversas doenças.

1.2 Sistema de Monitoramento da Saúde Humana

Um Sistema de Monitoramento da Saúde Humana, deve ser capaz de coletar informações sobre a condição de saúde de seus usuários, para tal o sistema pode utilizar um Interrogatório Sintomatológico, que é um conjunto de questões referentes à saúde do indivíduo, com o intuito de isolar possíveis problemas.

O sistema deverá contar com um módulo de Fusão de Dados, definida na seção 1.3, para otimizar a ordem das perguntas e conseqüentemente o tempo em que o sistema ficará ativado aplicando o questionário ao usuário, gerando assim uma economia de energia para o dispositivo móvel e procurando diminuir o tempo de interação entre o usuário e o sistema. Para tal, o módulo de Fusão de Dados responsável pela otimização das perguntas, fará uso de uma das técnicas da Inteligência Artificial, as Redes Bayesianas, capazes de atuar em cenários em que raciocinar com incerteza é algo constante [4].

1.3 Definição de Fusão de Dados

De fato, a noção intuitiva de Fusão de Dados é algo relativamente simples: Unir informações de diversas fontes para aprimorar a compreensão do que se está monitorando, por exemplo. No entanto tal simplicidade leva muitos pesquisadores a adotarem terminologias próprias para se referir ao assunto. De acordo com L. Wald [2] vê-se a utilização dos seguintes termos: junção, combinação, sinergia, integração, etc. Portanto fez-se necessário à adoção pela comunidade científica de uma definição clara e um novo vocabulário para Fusão de Dados. Em 1991 o Departamento de Defesa dos Estados Unidos criou o primeiro vocabulário para tratar de Fusão de Dados. Como legado desse trabalho que foi pioneiro, surgiu uma estrutura funcional para modelar situações militares onde existam várias fontes, sensores por exemplo, gerando informações. Entretanto ficou evidente que os jargões militares e as características específicas para ambientes militares de sua estrutura funcional os deixariam restritos às aplicações militares. Mesmo com tais deficiências a definição e estrutura funcional do Departamento de Defesa dos Estados Unidos servem como base para as principais estruturas formais propostas atualmente.

Sendo assim, em 1998 um grupo de trabalho formado pela *European Association of Remote Sensing Laboratories* (EARSeL) e *French Society for Electricity and Electronics* (SEE, French affiliate of the IEEE) [2] definiram o termo Fusão de Dados.

Assim L. Wald [2] define Fusão de Dados como:

“Fusão de dados é uma estrutura formal na qual está expresso os meios e ferramentas para fusão dos dados originários de diferentes fontes. Seu objetivo é a obtenção de informação de maior qualidade; a definição exata de ‘maior qualidade’ irá depender da aplicação.”

É importante observar da definição que uma maior ênfase é dada na estrutura formal, *Framework*, da Fusão de Dados, os meios e ferramentas que serão utilizados deixa o pesquisador livre para escolher quais princípios utilizar em cada um.

A qualidade proposta pela definição é algo a ser definido pelo consumidor das informações provenientes do sistema de Fusão de Dados. Às vezes uma maior cobertura da área a ser estudada já pode ser considerado como um aumento da qualidade, por exemplo.

Em [14] são discutidos meios e técnicas para a criação de um sistema de Fusão de Dados que pode aprimorar as decisões clínicas, relativas a doenças cardíacas, também é apresentada uma definição para Fusão de Dados, onde é interessante observar a atribuição de níveis capazes de suportar a Fusão de Dados. Abaixo a definição é transcrita:

“Fusão da Informação se refere a combinação de dados originários de múltiplas fontes e são usados para aprimorar as tarefas de decisão – como classificação, estimação, e predição – e para prover um melhor entendimento do fenômeno sobre consideração. Fusão pode ocorrer no nível de aquisição dos dados, pré-processamento dos dados, representação do conhecimento ou dados, ou no nível de decisão.”

Nota-se ainda da definição anterior uma maior preocupação em ressaltar os aspectos “úteis ou práticos” da Fusão de Dados, ou seja, *“...aprimorar as tarefas de decisão... ...e para prover um melhor entendimento do fenômeno sobre consideração...”*.

1.4 O Interrogatório Sintomatológico

A técnica de anamnese, ou seja, levantamento do histórico dos antecedentes de uma doença, pode ser feita de maneira automatizada. *Celmo Celeno* em seu livro sobre Semiologia Médica [7] diz o seguinte:

“A anamnese, que era considerada reduto inatingível pela tecnologia, já foi por ela alcançada. Estão sendo criados, em número crescente, aparelhos capazes de manipular os dados do ‘interrogatório sintomatológico’, anotados até pelo próprio paciente, inclusive sem a presença ou participação do médico.”

O interrogatório é uma ferramenta médica que auxilia o médico a determinar um quadro sobre seu paciente e assim gerar hipóteses sobre o estado de saúde do paciente. O interrogatório é formado por questões que abrangem desde a condição física quanto psíquica da pessoa, sendo que neste trabalho, o interrogatório utilizado (Anexo A) se resume a condição física da pessoa.

1.5 Sumário e Conclusões

A definição de *Pervasive Computing* permite pensar em sistemas vestíveis, podendo então fazer uso de tais sistemas para a monitoração da saúde humana.

A implementação do Interrogatório Sintomatológico em Sistemas de Monitoração da Saúde Humana, permite que tais sistemas possam extrair informações sobre a condição de saúde de seus usuários.

No próximo capítulo segue-se uma descrição das metodologias e ferramentas adotadas para gerar a Rede Bayesiana a partir do Interrogatório Sintomatológico, assim como as abordagens computacionais usadas para melhorar e adequar a representação da rede.

CAPÍTULO 2

2.1 Modelando a Incerteza: Redes Bayesianas

A teoria de decisão Bayesiana é usada para gerar um modelo probabilístico, ela permite determinar probabilidades condicionais de uma evidencia *a priori*, já as probabilidades revisadas são chamadas probabilidades *a posteriori*. Redes Bayesianas podem ser criadas para lidar com problemas que possuem algum nível de incerteza, desde o estabelecimento de diagnósticos de doenças, seleção de tratamentos otimizado até na prescrição de tratamentos médicos [13].

Uma Rede Bayesiana pode ser representada através de uma estrutura gráfica associada a uma distribuição de probabilidade, ou seja, graficamente pode ser representada por um grafo acíclico e dirigido, onde os nodos do grafo estariam representando as variáveis do problema modelado e os arcos entre os nodos definem as relações de dependência probabilística entre as variáveis. Tais tipos de redes funcionam como módulos em sistemas de apoio a decisão, e são no caso de uma construção manual da rede, totalmente dependentes de conhecimentos prévios de especialistas humanos no problema modelado. Uma construção automatizada da rede precisaria apenas de uma base de dados e algoritmos capazes de criar uma rede através dos dados [13]. A desvantagem da rede construída manualmente é o tempo gasto em entrevistas com especialistas para coleta de informações, mas resulta em uma rede de maior qualidade. Já uma rede automatizada pode dispensar o especialista, mas precisa de uma base de dados para análise e normalmente gera uma rede de qualidade inferior a manual.

2.2 Formalização das Redes Bayesianas

Formalmente um o grafo que representa a rede Bayesiana é assim definido:

$$B = (\text{Pr}, G) \quad (1)$$

Onde em (1) temos:

$G = (V(G), A(G))$ definição do grafo G com nodos $V(G) = \{V_1, \dots, V_n\}, n \geq 1$ e arcos $A(G) \subseteq V(G) \times V(G)$.

Pr é a distribuição de probabilidade que atua sobre o grafo, onde para cada nodo V_i (variável) é atribuída um conjunto de distribuição de probabilidades condicionais, ou seja,

$\Pr(V_i | \mathbf{p}(V_i))$. Onde $\mathbf{p}(V_i)$ representa a combinação das probabilidades dos predecessores (pais) de V_i .

Na Figura 1, pode-se observar a estrutura geral de uma rede Bayesiana, vale ressaltar o conteúdo da variável Sobrevivência, na tabela de probabilidade atribuída ao nodo, nota-se a combinação de todas as probabilidades relacionadas aos acontecimentos dos eventos dos pais de V_i , ou seja $V_i = (\text{Sobrevivência ou } \neg\text{Sobrevivência})$ e os $\mathbf{p}(V_i) = (\text{câncer, DC ou } \neg\text{câncer, } \neg\text{DC})$ e, sendo que as mesmas não representam, para efeitos de simplificação, os valores de probabilidade (0 até 1), somente termos lógicos referentes aos eventos.

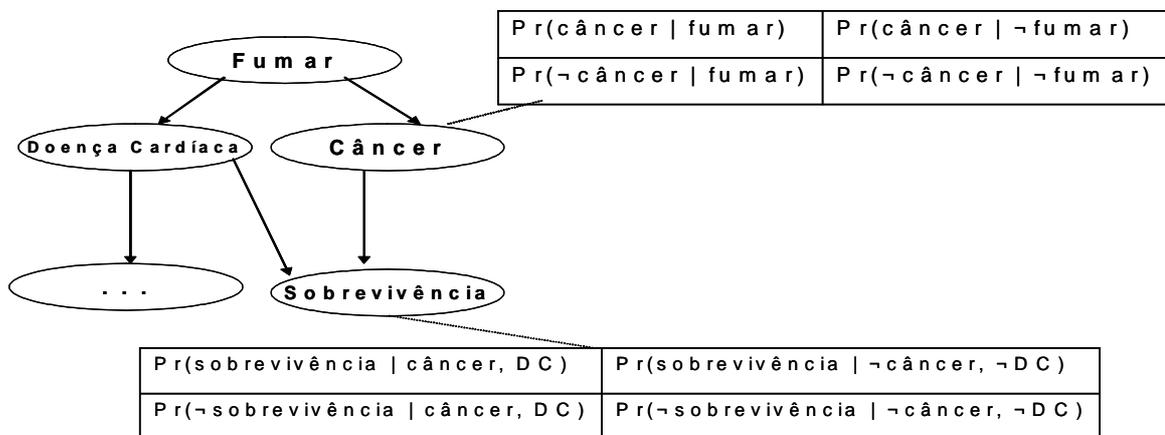


Figura 1 – Exemplo de rede Bayesiana.

2.3 Etapas para Criação Manual de Redes Bayesianas

Em [13] tem-se a enumeração e descrição de cinco etapas para a criação de uma Rede Bayesiana relacionada à medicina e cuidados médicos, sendo que tais etapas são listadas abaixo:

1. **SELEÇÃO DE VARIÁVEIS RELEVANTES:** como cada nó do grafo representa uma variável relevante do processo sobre consideração, faz-se necessário fazer um levantamento de todas as possíveis variáveis que compõem o processo em questão. Para tal costuma-se realizar entrevistas com especialistas na área do processo, levantando assim todas as possíveis variáveis.
2. **IDENTIFICAÇÃO DOS RELACIONAMENTOS ENTRE AS VARIÁVEIS:** após a identificação das variáveis que fazem parte do processo, é necessário analisar como tais variáveis se relacionam, ou seja, quais são as casualidades presentes em

cada variável que fazem com que determinadas variáveis tenham ligações através de arcos e outras não. Normalmente essas casualidades estão relacionadas com o conhecimento do especialista sobre o acontecimento de determinados eventos, perguntas do tipo “*O que causa tal efeito?*” e “*Que manifestação essa causa pode ter?*” são mostradas em [13] como boas opções para se extrair tais tipos de relacionamentos.

3. IDENTIFICAÇÃO DE PROBABILIDADES QUALITATIVAS E RESTRIÇÕES LÓGICAS: identificar o tipo de distribuição de probabilidade que rege o processo pode ajudar a avaliar e verificar as probabilidades requeridas para a construção da rede. Já as restrições lógicas ajudam a limitar o universo de probabilidades que tem de ser avaliadas pela rede, exemplo, se na Figura 1 as chances de sobrevivência só pudessem ser avaliadas se a pessoa \neg câncer e \neg DC, então a tabela de probabilidade associada ao nodo sobrevivência teria a metade do seu tamanho.
4. AVALIAÇÃO DAS PROBABILIDADES: nesse estágio a distribuição de probabilidade $\Pr(V_i | \mathbf{p}(V_i))$ é atribuída a cada nó do grafo. Tais probabilidades são obtidas na maioria das vezes dos especialistas no processo sobre questão ou então podem ser levantadas a partir de um conjunto de dados sobre o processo.
5. ANÁLISE DE SENSIBILIDADE E AVALIAÇÃO: no quarto estágio a rede já está totalmente montada, o que falta então é verificar sua validade. Pode-se através de dados reais de pacientes fazer um teste de sensibilidade da rede para verificar se as distribuições de probabilidades subjacentes possuem alguma inexatidão. A avaliação pode ser feita utilizando-se dados de um paciente na rede recém criada, então as mesmas informações são submetidas a outro sistema probabilístico, comparando-se os resultados.

2.4 Propagação de Evidências

“A propagação é um mecanismo que permite avaliar as probabilidades a posteriori das variáveis da rede. Para se manter a consistência da crença sobre as variáveis do modelo, sempre que surge uma nova evidência a favor de uma variável a crença nesta variável é alterada e esta nova crença propagada por toda a rede. Com isso, toda a rede atualiza sua crença e os efeitos da nova evidência são absorvidos por todas as variáveis, fazendo com que a representação continue consistente” [15].

Tal passagem de novas evidências, ou mensagens, pode ser realizada através de troca de mensagens entre os nós da rede.

2.5 Ferramentas para Modelagem de Redes Bayesianas

Dependendo do cenário que se deseja modelar, uma Rede Bayesianas pode ser criada com dezenas ou até centenas de nodos. Sendo assim para facilitar a representação gráfica de tais estruturas existem ferramentas computacionais capazes de auxiliar no projeto e validação de Redes Bayesianas. A seguir tem-se a descrição da ferramenta utilizada para a criação da rede que representa o Interrogatório Sintomatológico.

2.5.1 Microsoft Belief Networks

É um *software* para a plataforma *Windows* que dá suporte a criação, manipulação e avaliação de Redes Bayesianas. O *software* representa a rede graficamente através de um diagrama, onde as variáveis são exibidas como elipses, os nodos, e as dependências condicionais são exibidas como arcos entre as variáveis.

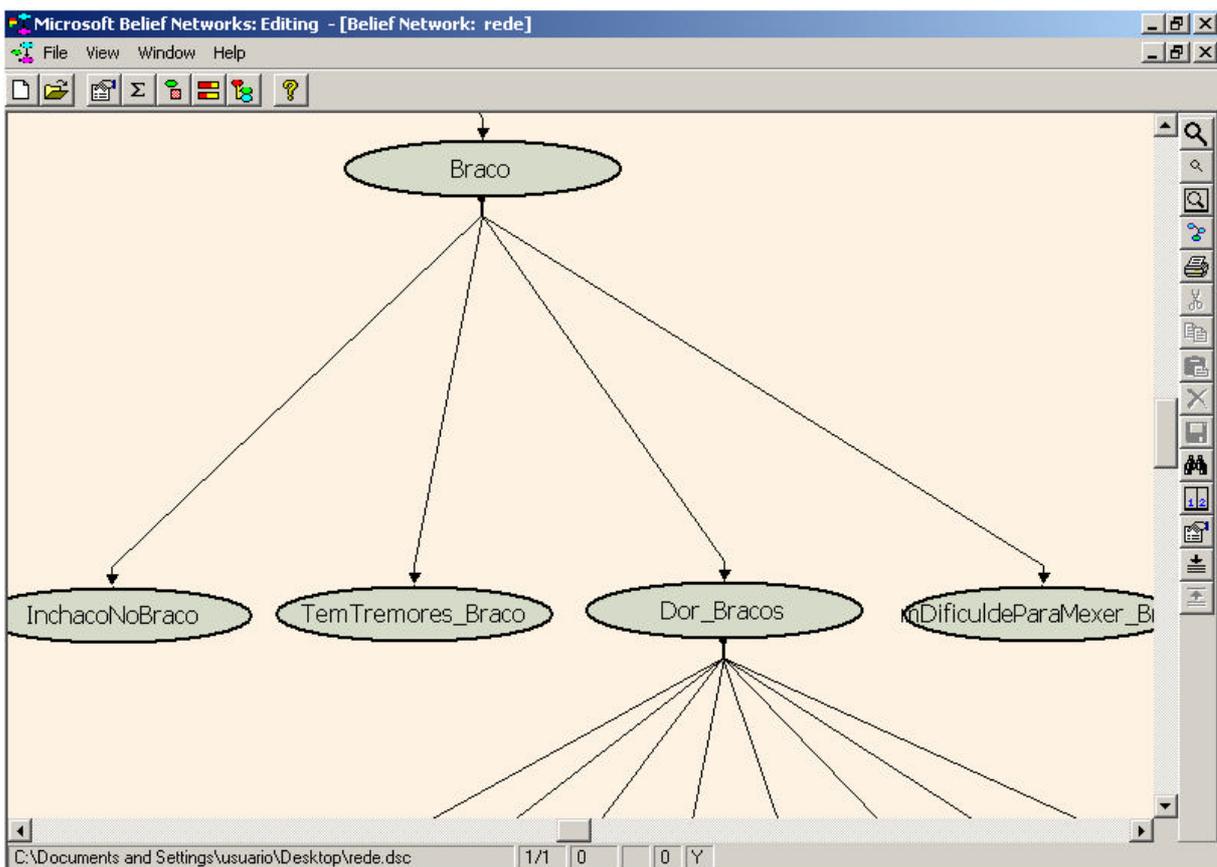


Figura 2 – Exemplo da Estrutura Gerada pela Ferramenta.

Cada nodo pode ser selecionado e então atribuído ao mesmo a sua tabela de probabilidades assim como definidas as evidências que podem ser observadas no cenário que se está modelando.

O *Microsoft Belief Networks* permite que a estrutura gráfica gerada seja armazenada em um arquivo em disco com a opção de escolha do tipo de formato que será dado ao arquivo gerado. Para a execução deste trabalho foi escolhido o formato .dsc, um formato de texto estruturado que pode ser visualizado em seus componentes principais (Figura 3 e 4). A versão utilizada do *software* foi 1.4.2.

Como principais vantagens do *Microsoft Belief Networks* temos a sua facilidade para trabalhar com vários nodos, permitindo a manipulação gráfica da maioria de suas funções, a possibilidade de se realizar inferências sobre novas evidências inseridas no modelo, pois o mesmo utiliza o algoritmo de propagação de evidências árvore de junção. Outra vantagem é que se trata de um *software* que pode ser utilizado livremente.

```

node AchouQueOCoracaoParouDeBater
{
  name = "Achou que o Coracao Parou de Bater";
  type = discrete[2]
  {
    "Yes",
    "No"
  };
  position = (570789, -73999);
}

```

Figura 3 – Representação do nodo no arquivo dsc.

```

probability(AchouQueOCoracaoParouDeBater|Peito)
{
  (0) = 0.5, 0.5;
  (1) = 0.5, 0.5;
}

```

Figura 4 – Representação da tabela de probabilidade do nodo.

Uma das desvantagens dessa ferramenta é que a mesma não gera o modelo em forma de estrutura de dados para nenhuma linguagem específica, usa um formato proprietário no arquivo .dsc. Sendo assim foi necessária a implementação de um pequeno tradutor que será descrito na seção seguinte.

2.6 O TradutorDSC

O tradutor criado para a execução desse trabalho é bastante simplificado, pois deverá ser portado para execução em ambientes com recursos computacionais restritos, como celulares e *PDA*s. O mesmo se baseia nas características particulares da Rede Bayesiana do Interrogatório Sintomatológico. A listagem do código fonte do tradutor pode ser vista no Anexo B. Sendo que o mesmo foi implementado usando a tecnologia *Java*.

O tradutor assume algumas características importantes encontradas na rede modelada:

- Todo nodo possui apenas um único nodo antecessor (Pai).
- As matrizes de probabilidades no arquivo DSC não podem ultrapassar o comprimento da linha, ou seja, não pode haver um matriz com quebra de linha.
- Linhas do tipo: *Type*, *Position* e *Default* são ignoradas pelo tradutor.

Sendo que *Type* define a quantidade de evidências que podem ser observadas em um nodo. O tradutor simplesmente começa a leitura das evidências quando encontra o *token* { e finaliza a leitura das mesmas quando encontra o *token* } ;.

Linhas definidas como *Position* só são utilizadas pelo *Microsoft Belief Networks* para gerar a visualização do grafo na posição correta da tela.

Já as linhas *Default* são linhas que definem probabilidades padrão para o nodo, economizando assim tempo na digitação, bastando apenas escolher a probabilidade padrão e a tabela é preenchida, sendo que no Interrogatório Sintomatológico tal tipo de probabilidade não foi utilizado nenhuma vez.

Na Figura 5 é mostrado um diagrama de classes de como foi implementada a interação do tradutor com o restante das classes que permitem o funcionamento do tradutor.

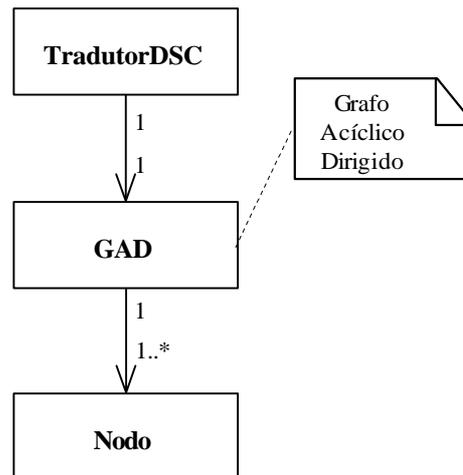


Figura 5 – Diagrama de Classes

Assim com o uso do TradutorDSC é possível portar a estrutura gerada pelo *Microsoft Belief Networks* para a linguagem *Java* permitindo assim a obtenção de uma representação da rede que possa ser executada nos algoritmos propostos nas próximas seções, que darão suporte a decisão.

2.7 Suporte a Decisão

O uso das Redes Bayesianas para a modelagem do Interrogatório Sintomatológico tem como principal objetivo automatizar e otimizar o procedimento de anamnese usada pelos médicos. Tais características no entanto, não garantem uma precisão para a geração de um diagnóstico, que no cenário médico é uma decisão.

No entanto, tais redes podem gerar informações de grande valia para módulos especializados na tomada de decisões, ou seja, a idéia é que as Redes Bayesianas possam se enquadrar em um nível abaixo dos módulos de decisão dos Sistemas de Monitoramento da Saúde Humana, formando assim o suporte necessário para a correta operação dos módulos de decisão.

Na Tabela 1 observam-se os níveis assim como as técnicas mais comumente utilizadas em cada um dos níveis da Fusão de Dados. Sendo que o seguinte trabalho se encaixa no nível 2. Assim sua saída de dados pode passar para módulos do nível 3, onde esses seriam responsáveis por gerar uma saída mais refinada para um módulo de decisão, ou mesmo diretamente para algum especialista humano, no caso um médico. Os níveis podem de acordo com [1] ser assim explicados: Nível 1 é responsável pelo pré-processamento do sinal, ou seja obtenção do dado. Nível 2 é responsável por gerar interpretações dos dados, ou seja criar

variáveis. Nível 3 é um nível de mistura, onde dados e variáveis podem ser agregados para obtenção de informações de maior valor.

Tabela 1 - Técnicas Comuns em Fusão de Dados.

Nível de Fusão	Técnicas
Nível 1	FOM (<i>Figure of Merit</i>)
	Técnicas de Gating
	Filtro de Kalman
Nível 2	Teoria de Decisão Bayesiana
	DSER (Argumentação Evidencial Dempster-Schafer)
	Redes Neurais
	Métodos de Clusterização
Nível 3	Sistemas Especialistas
	Arquitetura Blackboard
	Lógica Fuzzy
	Raciocínio Baseado em Casos

2.8 Coleta de Evidências: Algoritmos em Grafos

Uma Rede Bayesiana como definida no item 2.2 é um grafo acíclico e dirigido, sendo então possível gerar algoritmos capazes de percorrer o grafo seguindo algum critério de busca. Durante o trajeto pelo grafo é possível realizar a coleta de evidências fornecidas pelo usuário do sistema, tais grupos de evidências fornecem a base para a geração de um raciocínio evidencial nos módulos de decisão.

Surge agora a questão: qual seria o melhor algoritmo para realizar a coleta de evidências pelo grafo?

Levando-se em conta que a rede assume a forma de um grafo acíclico, pode-se então pensar em usar algoritmos para caminhamentos em árvores. Para este trabalho foram avaliados dois algoritmos: um algoritmo de busca em profundidade [5] e um outro que utiliza uma heurística para gerar o caminhamento.

Definidos os algoritmos que serão analisados, basta agora definir qual será o parâmetro que definirá qual terá o “melhor” ou “pior” desempenho no caminhamento pelo grafo. Uma característica do Interrogatório Sintomatológico é que o mesmo assume uma visão de alto nível do possível problema do paciente e então vai refinando tal visão através de perguntas cada vez mais específicas. Portanto, o parâmetro de desempenho será qual dos algoritmos consegue chegar até as perguntas mais específicas de forma mais rápida sendo que

a constatação é a velocidade de convergência que será dada pela notação *big-Oh*, ou seja, qual a função limitante superior do algoritmo.

2.8.1 Busca em Profundidade

Tal técnica de pesquisa consiste em visitar todos os nós de um ramo até atingir os nós terminais, repetindo o processo em todos os ramos [5].

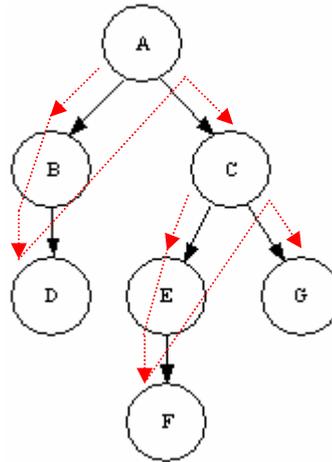


Figura 6 – Busca em Profundidade

A técnica original de busca em profundidade procura alguma chave de pesquisa pré-definida. Sendo assim para o seu uso no grafo do interrogatório faz-se necessário a execução de perguntas referentes aos nodos imediatamente posteriores a pergunta anterior. Ex: Caso a pergunta C da Figura 6 tenha sido feita, seria necessário realizar a pergunta referente ao nodo E e G, pois o algoritmo original não prevê qualquer tipo de heurística para escolher a pergunta mais apropriada.

A complexidade computacional do algoritmo de busca em profundidade, especificamente para o cenário modelado é $O(n)$, pois o número de iterações do algoritmo fica restrito ao número de nodos encontrados em seu trajeto, sendo que o grafo utilizado para o caminhamento não possui arestas cíclicas.

2.8.2 Heurística Proposta

A heurística proposta a seguir procura criar um algoritmo para coleta de evidências na Rede Bayesiana de modo eficiente, tendo em vista que o mesmo procura imitar de maneira

simplificada o tipo de escolha que o médico realiza na execução do Interrogatório Sintomatológico.

A diferença básica entre esse algoritmo e o da seção anterior, é a suposição de que o mesmo utiliza um Banco de Conhecimento sobre Eventos, assim sempre que uma nova evidência é coletada a mesma é pesquisada no Banco de Conhecimento sobre Eventos, definindo assim qual das próximas perguntas são mais prováveis de serem feitas levando-se em conta a evidência anterior. Assim o usuário do sistema não será sobrecarregado com inúmeras perguntas, o usuário só terá que responder a pergunta mais provável de acordo com seu problema.

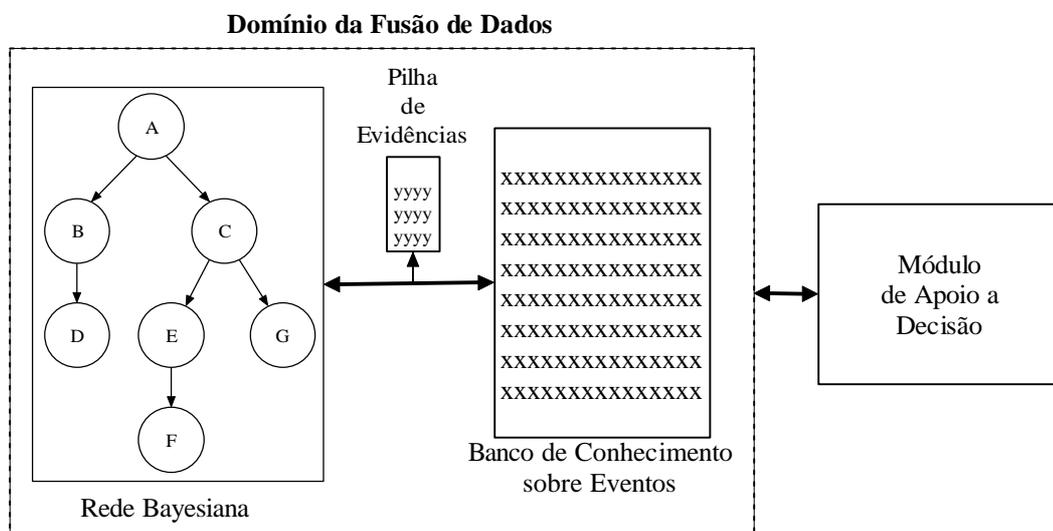


Figura 7 – Visão de Alto Nível do Domínio da Fusão de Dados

Na Figura 7 pode-se observar como o Domínio da Fusão de Dados se relaciona com o módulo de Apoio à decisão. O objetivo principal do algoritmo proposto é preencher a estrutura denotada como Pilha de Evidências, para tal o mesmo deverá constantemente procurar a próxima pergunta com a maior probabilidade de ser a escolha correta, baseado no que se conhece sobre o evento que acabou de ser observado. Para isso cria-se uma estrutura adicional chamada de Banco de Conhecimento sobre Eventos.

Analisando-se a Figura 8 vê-se o funcionamento interno da heurística proposta: a questão A é feita ao usuário, sendo que o mesmo diz qual a evidência condiz com seu estado de saúde, Pouca Dor de Cabeça, por exemplo. Agora sabendo que o usuário queixa-se de dor de cabeça, o algoritmo vai realizar uma busca no Banco de Conhecimento sobre Eventos para encontrar as Crenças que fortalecem ou enfraquecem a possibilidade de se executar a pergunta

B ou C. No exemplo, como a evidência Pouca Dor de Cabeça, foi verdadeira, então as crenças relevantes a escolha serão as que possuem tal evidência como verdadeira, ou seja, reduz-se então o número de itens que terão de ser pesquisados no Banco de Conhecimento sobre Eventos. Logo restam apenas duas opções: $p(A) = \text{verdade} \rightarrow c(C) = 7$ e $p(A) = \text{verdade} \rightarrow c(B) = 4$. Assim a próxima pergunta realizada será a C pois possui a maior crença.

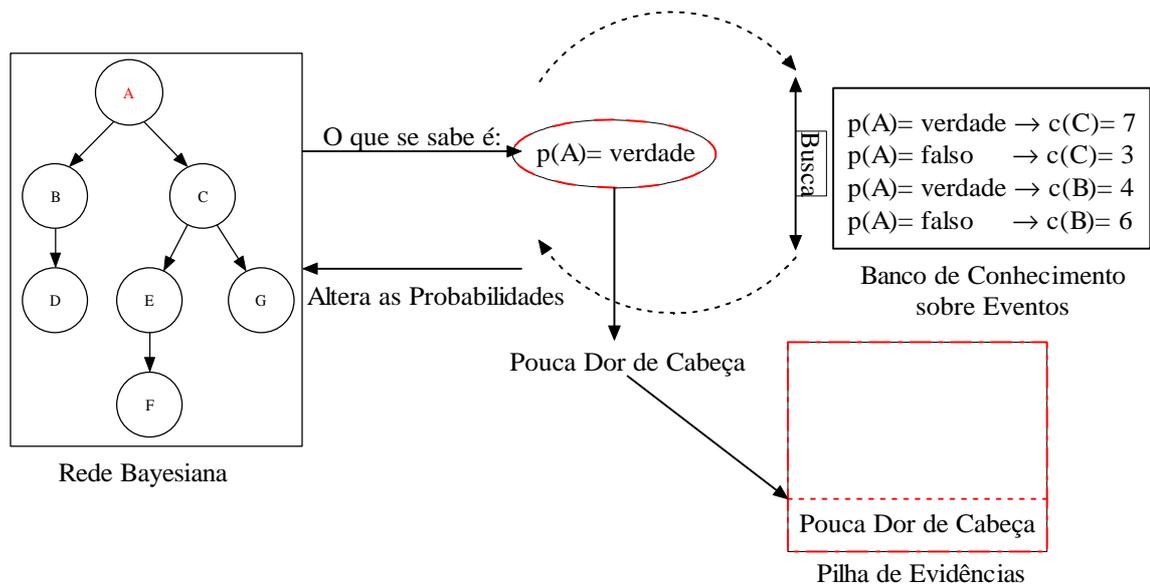


Figura 8 – Heurística para Busca no Grafo, Passo 1

Em uma segunda iteração o algoritmo irá realizar a pergunta C ao usuário, no exemplo a evidência coletada é falsa, no caso Sem Tontura, assim uma busca procurando por todas as ocorrências no Banco de Conhecimento sobre Eventos onde a evidência ocorre, retorna duas possíveis escolhas, assim como no passo anterior escolhe-se a de maior crença, ou seja $c(G) = 5$. Logo a próxima pergunta seria a do nodo G.

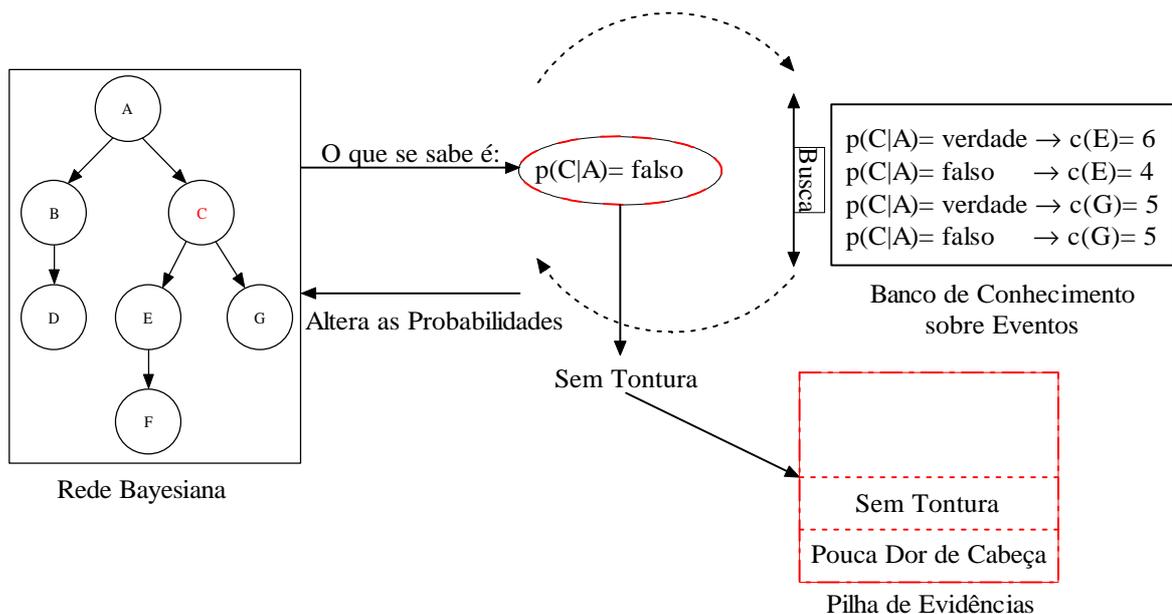


Figura 9 - Heurística para Busca no Grafo, Passo 2

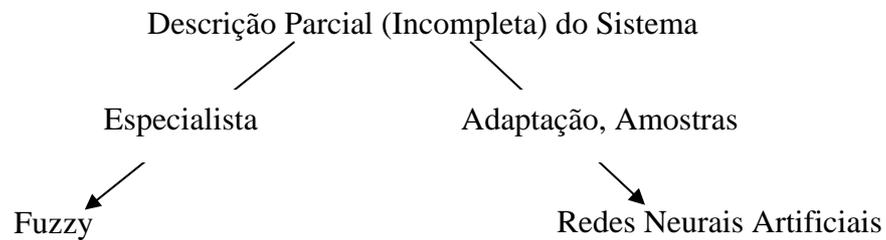
Assim o algoritmo continuaria sua execução até encontrar um nodo folha. Quando um nodo folha for alcançado, tem-se na pilha de evidências todos os fatos pertinentes a queixa do usuário, sendo que pode ocorrer que durante a execução do interrogatório, o questionamento tenha que seguir por bifurcações no grafo, neste caso pode-se utilizar de pilhas intermediárias em cada sub-grafo da estrutura.

Pode-se imaginar o algoritmo proposto acima como uma espécie de busca em um grafo com pesos nas arestas. Sendo que os pesos mudam de maneira dinâmica, de acordo com o que foi observado na coleta da evidência e que essa mesma observação muda a crença (pesos) das próximas perguntas. A complexidade computacional do algoritmo é igual ao de busca em profundidade, logo a complexidade é $O(n)$.

2.9 Sumário e Conclusões

Problemas que apresentam a incerteza como uma de suas principais características, são fortes candidatos a serem resolvidos usando a Teoria de Decisão Bayesiana, sendo que a mesma pode ser estendida para representar o problema na forma de um Grafo Acíclico e Dirigido. Viu-se ainda que a criação de Redes Bayesianas segue algumas etapas e algumas dessas etapas são totalmente dependentes do conhecimento de especialistas humanos em determinada área do conhecimento. Assim a Rede Bayesiana proposta neste trabalho se encaixa na classificação de sistemas especialistas definida por Bauchspiess [9], pois a mesma

foi construída com a ajuda de conhecimento de especialistas e não com amostras de dados, Figura 10.



Fonte:(Bauchspiess,2002)

Figura 10 – Tipo de Problema Estudado e Abordagem Utilizada

Para a modelagem de Redes Bayesianas usou-se o *software Microsoft Belief Networks* que permite a criação, manipulação e teste da estrutura gerada. Também foi constatado que o *software* não dá suporte a geração de arquivos com estruturas de dados prontas para serem usadas em nenhuma linguagem de programação. Sendo assim foi necessária a criação de um pequeno tradutor para portar a estrutura gerada no *Microsoft Belief Networks* para a linguagem *Java*.

Ambos os algoritmos possuem a mesma complexidade computacional, ou seja, $O(n)$. O que pode deixar o algoritmo que utiliza uma heurística em vantagem é o fato de que o usuário terá que avaliar uma evidência que tem maior probabilidade de condizer com sua condição de saúde, e não mais avaliar evidência por evidência apresentada pelo interrogatório. Sendo que quando a evidência suposta pelo sistema não condizer com o que o usuário esteja vivenciando, então tal vantagem desaparece, pois o usuário terá de avaliar todas as possíveis evidências assim como no algoritmo de busca em profundidade.

Foi mostrado ainda que o TradutorDSC assume que algumas características importantes da Rede Bayesiana que representa o Interrogatório Sintomatológico estejam presentes. No próximo capítulo segue-se uma explicação de como foi feita a modelagem da rede e suas principais características. E como o módulo de Fusão de Dados que dá o suporte à decisão foi concebido e qual dos dois algoritmos obteve um melhor desempenho.

CAPÍTULO 3

3.1 Modelando o Interrogatório Sintomatológico

O Interrogatório Sintomatológico é um conjunto de questões pertinentes à saúde do paciente, que tem como objetivo auxiliar o médico na realização da anamnese. As perguntas que compõem o interrogatório são organizadas de forma hierárquica, ou seja, permitem uma abordagem do tipo *top-down* do problema vivido pelo paciente. O mais importante neste tipo de hierarquia é que a mesma pode ser facilmente representada sob a forma de um grafo.

A seguir tem-se um pequeno trecho do Interrogatório Sintomatológico, ilustrando como o mesmo pode ser transformado em um grafo, sendo que o interrogatório completo pode ser visto no Anexo A. O Interrogatório Sintomatológico utilizado foi obtido em [16].

3. Cabeça

- 3.1. Onde fica? no rosto (inclui a testa e o ouvido) resto da cabeça
 - 3.1.1. Tem dor de cabeça? Sim Não
 - 3.1.1.1. Começou: de repente gradualmente
 - 3.1.1.2. Intensidade: fraca moderada intensa muito intensa
 - 3.1.1.3. Dor: em queimação em pontadas pulsátil em aperto fisgada em peso contínua
 - 3.1.1.4. Há quanto tempo? Menos de 12 horas entre 12 e 24 horas há mais de 1 dia
 - 3.1.1.5. Onde dói? Toda cabeça na testa no lado esquerdo no lado direito na nuca
 - 3.1.1.6. Irradiação para: testa parte de trás da cabeça nuca boca
 - 3.1.1.7. Tem relação com: ansiedade menstruação leitura tosse exercício posição e/ou movimentos nenhuma alternativa anterior
 - 3.1.1.8. Está associado com: distúrbios visuais ânsia de vômito vômito

Figura 11 – Parte do Interrogatório Sintomatológico

Na Figura 11 pode-se observar a estrutura de tópicos do interrogatório, evidenciando assim sua natureza hierárquica, vale lembrar que os itens em sublinhado são as evidências que podem ou não ser observadas pelo paciente. Assim o grafo só representa as perguntas, sendo que as evidências ficam inseridas dentro de cada nodo do grafo. Abaixo se encontra a estrutura gráfica da Figura 11. Note no entanto que não há qualquer tipo de exibição das evidências nos nodos do grafo, tais evidências ficam armazenadas na estrutura de cada nodo (campo *State*), como pode ser visto na Figura 13 as evidências relativa a pergunta 3.1.1.1. Para simplificação da Figura 12 temos somente o item 3.1.1.1 e o item 3.1.1.8 como nodos folhas.

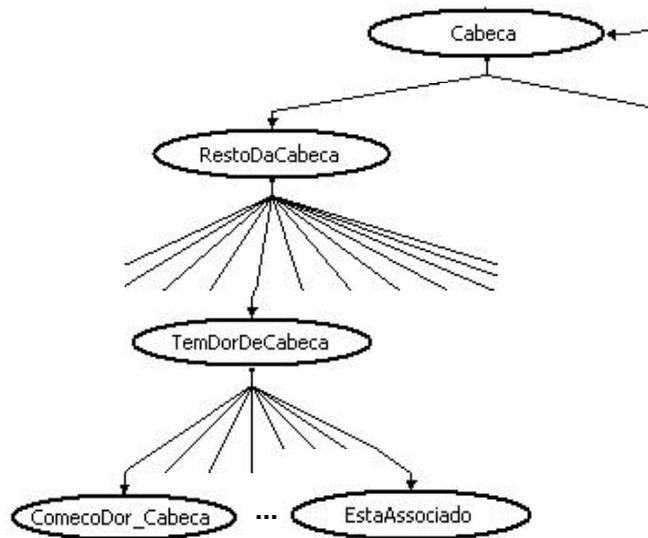


Figura 12 – Representação Gráfica da Figura 11.

Node		Model	Property Types
Attribute	Value		
Name	ComecoDor_Cabeca		
Description	Comeco da Dor		
States	Name		
State(0)	Repentinamente		
State(1)	Gradualmente		
State(2)	<unassigned>		
Properties	Value		
MS_Asym	<unassigned>		

Figura 13 – Propriedades do Nodo e Exibição das Evidências.

Uma característica importante do interrogatório é que o mesmo só gera um antecessor para cada nodo, ou seja, cada nodo filho ou conjunto de nodos filhos no mesmo nível irão possuir apenas um nodo predecessor (pai). Garantindo assim um único trajeto possível até um nodo folha.

3.2 Implementando o Suporte a Decisão

Os algoritmos responsáveis pelo funcionamento do módulo de suporte a decisão foram implementados utilizando-se a linguagem *Java*. Os arquivos que representam as classes criadas são mostrados abaixo, seguidos de uma breve descrição:

1. `Principal.java` : Classe principal para a chamada das outras classes, Anexo G.
2. `TradutorDSC.java` : Classe que implementa o tradutor.
3. `Nodo.java` : Classe que implementa o `Nodo`.
4. `GAD.java` : Classe que implementa o Grafo Acíclico e Dirigido.
5. `BProfundidade.java` : Classe que implementa o algoritmo de busca em profundidade.
6. `BHeuristica.java` : Classe que implementa o algoritmo com a heurística proposta.

3.2.1 A Classe TradutorDSC

A classe que implementa o tradutor é responsável por portar a estrutura gerada pelo *Microsoft Belief Networks* para uma estrutura de dados dinâmica, permitindo assim sua manipulação pelos algoritmos.

O algoritmo supõe que há no mesmo diretório de sua execução um arquivo chamado `rede.dsc`, que possui a implementação da Rede Bayesiana modelada. Como principal característica técnica do TradutorDSC tem-se o uso da classe `StringTokenizer` que é responsável por separar as linhas lidas do arquivo `rede.dsc` em *tokens* que podem ser analisados, tornando possível então montar a estrutura de dados que representa o grafo em memória para ser manipulado pelos algoritmos.

Note no diagrama de classes reproduzido novamente na Figura 14, que cada vez que uma nova instância do TradutorDSC é criada, o mesmo se encarrega de criar um novo objeto da classe GAD. Assim o modelo dá liberdade de se criar em memória vários objetos GAD, sendo que cada um desses objetos pode representar uma Rede Bayesiana com características práticas diferentes. Ou seja, pode-se ter a Rede do Interrogatório Sintomatológico e mais uma rede para fazer inferências sobre ajustes em dosagens de insulina, carboidratos ingeridos e atividades físicas, por exemplo como mostrado em [17].

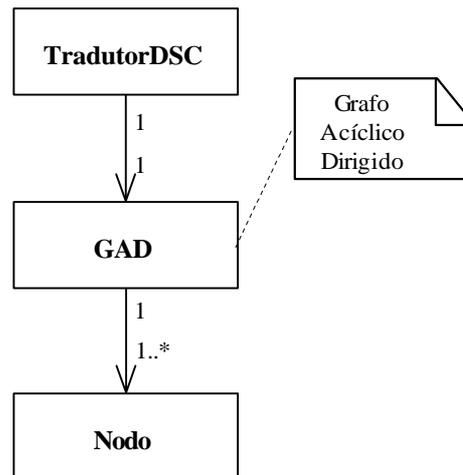


Figura 14 – Diagrama de Classes

A classe TradutorDSC possui um único método chamado `ListarArquivo()` que é responsável por iniciar toda a execução da leitura do arquivo `rede.dsc` e montagem das estruturas de dados. A listagem do código fonte do TradutorDSC pode ser vista no Anexo B.

3.2.2 A Classe Nodo

A classe Nodo é responsável por gerar todas as estruturas de dados e métodos para a criação e manutenção dos nodos em memória. Sua estrutura de dados pode ser vista na Figura 15.

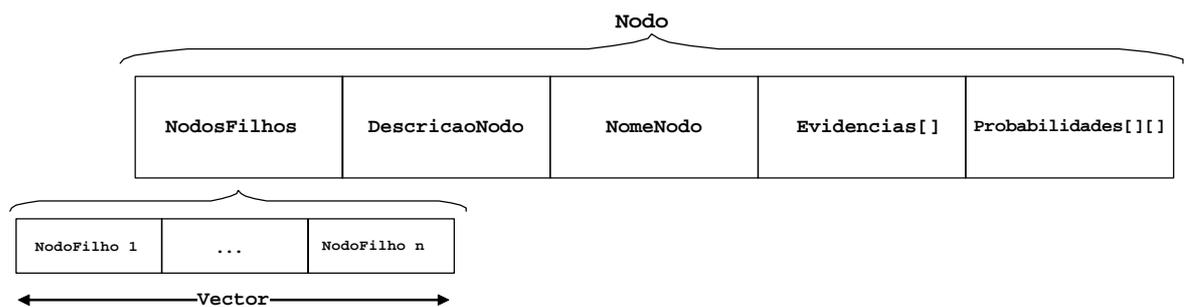


Figura 15 – Estrutura de Dados da Classe Nodo

O item de dado crucial para o funcionamento da estrutura mostrada acima é o campo `NodosFilhos` onde o mesmo armazena referências para os filhos do nodo em questão. Sendo que `NodosFilhos` é um campo do tipo *Vector*, uma estrutura de dados presente na linguagem *Java* que permite um crescimento dinâmico da estrutura.

Os métodos da classe nodo são descritos a seguir:

```
public Nodo(String Desc,String NomeN)
```

Método construtor do `Nodo`, recebe como parâmetros de construção duas *Strings* uma `Desc` (Descrição) e um `NomeN` (Nome do `Nodo`) que é a chave para pesquisa o grafo.

```
public JTextArea imprimeEvidencias()
```

Método que percorre a lista de evidências do `nodo` em questão gerando um objeto do tipo `JTextArea` contendo todas as evidências do `nodo`.

```
public int quantasEvidencias()
```

Método que retorna um inteiro com o número de evidência contidas no `nodo`.

```
public int quantosFilhos()
```

Método que retorna um inteiro com o número de filhos presentes no `nodo`.

```
public boolean temFilhos()
```

Método que retorna um *boolean* indicando se o `nodo` possui ou não filhos.

```
public void recebeFilho(Nodo ref)
```

Método que anexa um novo filho ao `nodo` em questão. Uma referência para o novo filho deve ser passada como parâmetro de entrada em `ref` que é do tipo `Nodo`.

```
public void removeFilho(Nodo ref)
```

Método para remover um filho do `nodo` em questão. Uma referência para o `nodo` que será removido deve ser passada com parâmetro de entrada em `ref` que é do tipo `Nodo`.

```
public int ondeEsta(Nodo ref)
```

Método que retorna um inteiro indicando a posição do `nodo` passado com referência no vetor de filhos.

```
public void fixarEvidencias(String evi[])
```

Método que atribui evidências ao `nodo`. Um vetor de *String* com as evidências é passado como parâmetro de entrada.

```
public void fixarProbabilidades(float p[][])
```

Método que atribui probabilidades ao nodo. Uma matriz de *Float* é passada com as probabilidades como parâmetro de entrada.

```
public JTextArea listarFilhos()
```

Método que retorna um objeto do tipo `JTextArea` contendo todos os filhos do nodo em questão assim como suas posições no vetor de nodos filhos.

A classe `Nodo` juntamente com seus métodos é uma classe basicamente utilizada pela classe `TradutorDSC`. Mas note que todos os métodos foram definidos como públicos, sendo assim em qualquer lugar do programa onde se tenha uma referência para um objeto do tipo `Nodo` os métodos podem ser invocados. Por exemplo: a qualquer referência de `Nodo` pode ser feita uma requisição para pedir a lista de seus filhos. Ex:

```
nodoInstanciado.listarFilhos();
```

Sendo que é retornado um objeto do tipo `JTextArea` que deve ser armazenado em um objeto do mesmo tipo ou mandado diretamente para um método de exibição em tela que suporte a classe de objeto utilizada.

A listagem de todo o código da classe `Nodo` pode ser vista no Anexo C.

3.2.3 A Classe GAD

Responsável pela manutenção e criação de Grafos Acíclicos e Dirigidos na memória, é basicamente uma classe que assimila referencias para objetos do tipo `Nodo`, criando assim uma coleção de nodos que juntos formam um grafo.

Sua estrutura de dados pode ser observada na Figura 16. Onde vale destacar a capacidade de auto-ajuste do campo `Nodo` definido como um tipo *Vector*.

Como as classes `TradutorDSC`, `GAD` e `Nodo` são utilizadas de maneira que o `TradutorDSC` gere instâncias das duas últimas classes respectivamente, observe a Figura 14. É possível realizar um agrupamento hierárquico de tais informações para seu uso nos algoritmos.

Ex:

```
TradutorDSC.Grafo.acharNo("Problema");
```

Onde temos a indicação do pacote `TradutorDSC`, seguido de uma indicação que se deseja acessar o membro `Grafo`, um objeto do tipo `GAD` criado dentro do próprio `TradutorDSC`. E em seguida é invocado o método `acharNo` do objeto `Grafo`, sendo que o método `acharNo` recebe como parâmetro uma *String*, no caso o "Problema" que é a raiz do grafo, de acordo com o que foi modelado no Interrogatório Sintomatológico.

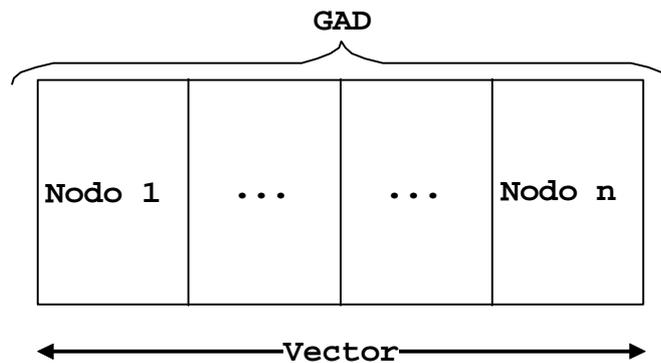


Figura 16 – Estrutura de Dados da Classe `GAD`.

Os métodos que dão funcionalidades a classe `GAD` são descritos a seguir:

```
public GAD()
```

Método construtor da classe `GAD`.

```
public void Insere(Nodo ref)
```

Método para inserir um novo nodo ao grafo.

```
public int ContaNos()
```

Método que retorna um inteiro com o número de nodos contidos no grafo.

```
public void imprimeGAD()
```

Método para criar uma listagem com todos os nodos contidos no grafo.

```
public Nodo acharNo(String chave)
```

Método para procurar por um determinado nodo no grafo, sendo o mesmo definido por uma *String* de busca, no caso do Interrogatório Sintomatológico usa-se o nome do nodo (`NomeNodo`) como chave de busca.

A listagem de todo o código fonte da classe GAD pode ser vista no Anexo D.

3.3 O Protótipo: BProfundidade.java

O algoritmo de busca em profundidade, como mostrado na seção 2.8.1. Foi implementado para usar a estrutura do grafo gerada pelo TradutorDSC.java. O protótipo inicia sua execução perguntando ao usuário se o mesmo possui algum problema, sendo essa a primeira pergunta do Interrogatório Sintomatológico e conseqüentemente a raiz do grafo. Quando um objeto da classe BProfundidade é instanciado, o mesmo assume que há um outro objeto da classe TradutorDSC já instanciado.



Figura 17 – Primeira Pergunta ao Usuário.

Supondo que o usuário responda sim, ou seja, 1 o sistema exibirá uma nova tela perguntando qual das seguintes evidências pode ser observada, Figura 18, sendo que somente uma pode ser escolhida. Afinal neste momento o algoritmo está esperando uma entrada para definir qual ramo do grafo irá seguir.

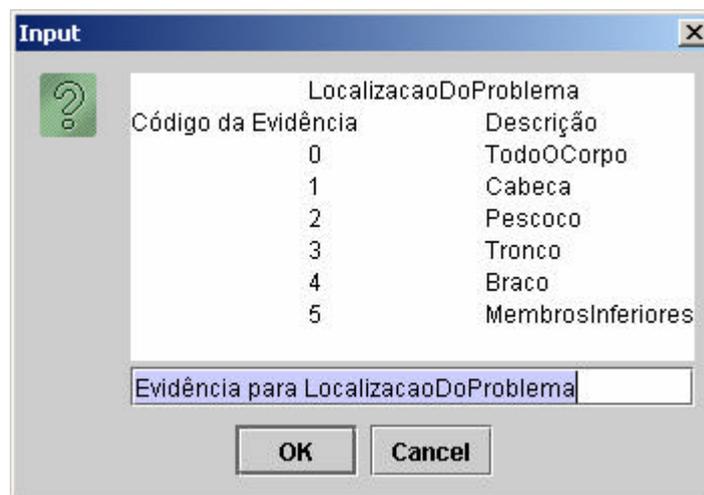


Figura 18 – Lista de Evidências para o Usuário.

Definido o primeiro trajeto do percurso, o algoritmo chega a um novo nodo pai e se o nodo possuir evidências que não sejam do tipo “Sim ou Não”, o mesmo irá então exibir uma lista com as evidências que podem ser observadas. Caso o nodo possua as evidências do tipo “Sim ou Não” então são exibidos os filhos do nodo como evidências. Note que quando se caminha para um nodo com evidências do tipo “Sim ou Não” é considerado que o usuário considera a evidência como presente, ou seja, “Sim”.

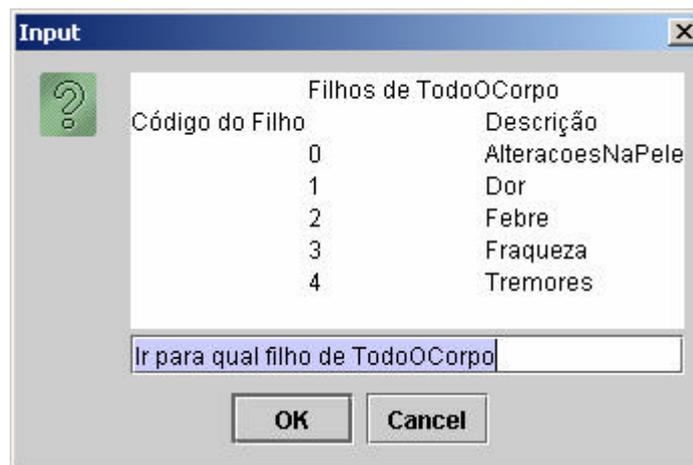


Figura 19 – Nodo com Evidências Tipo “Sim ou Não”.

A Figura 19 mostra o nodo chamado TodoOCorpo que possui evidências do tipo “Sim ou Não” e seus filhos são então exibidos para o usuário como evidências que podem ou não ser observadas.

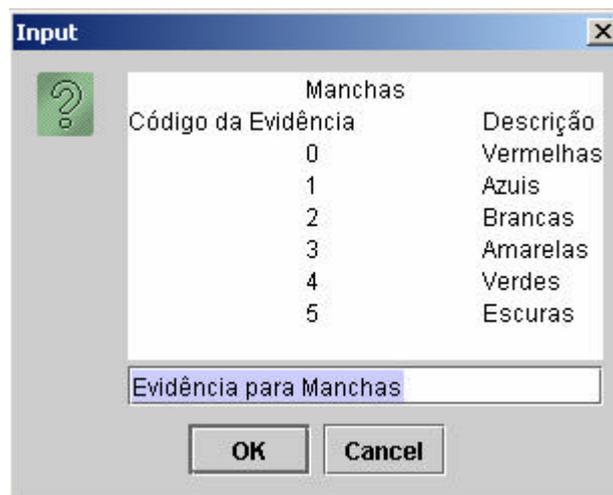


Figura 20 – Nodo com Evidências Diversas.

Já a Figura 20 mostra as evidências do nodo “Mancha” que não possui evidências do tipo “Sim ou Não”, logo são exibidas as evidências diversas que compõem a semântica do nodo. Observe no entanto, que o algoritmo segue durante toda a execução questionando o usuário sobre a opção que condiz com seu estado de saúde, em alguns momentos podem ser exibidas mais de oito evidências, o que torna a escolha da evidência correta um pouco demorada, pois o usuário tem que avaliar mentalmente todas as opções.

Após atingir um nodo folha (terminal), o algoritmo então questiona o usuário se há algum outro problema que não foi abordado no conjunto de questões realizadas. Sendo a tela de tal opção mostrada na Figura 21.



Figura 21 – Tela de Questionamento sobre o Final das Perguntas.

Caso o usuário responda “Sim”, o sistema então voltará a exibir a tela apresentada na Figura 18, e todo o procedimento descrito acima é realizado novamente. Caso a opção seja de parar o interrogatório, ou seja, selecionar a opção “Não” então uma caixa contendo todas as evidências observadas é exibida ao usuário do sistema, Figura 22.

Note ainda que na Figura 22, tem-se relatadas as evidências correspondentes a duas rodadas de perguntas, indicadas pela frase: “Localização do Problema” que ocorre duas vezes na figura.

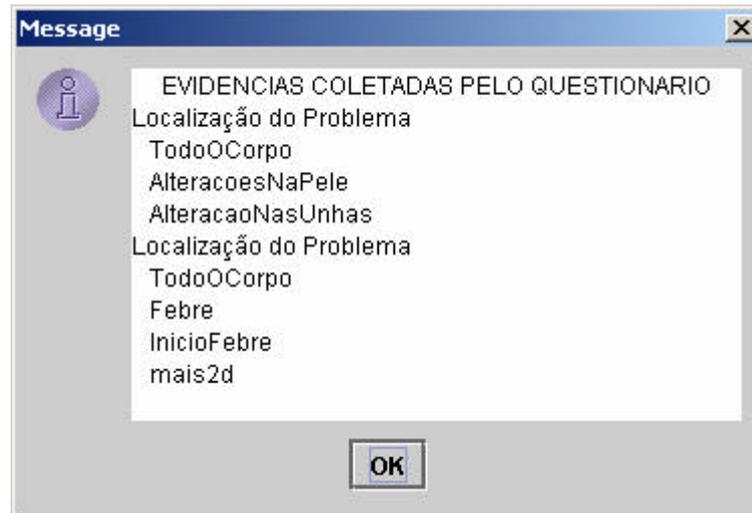


Figura 22 – Tela de Evidências Coletadas pelo Protótipo.

3.4 O Protótipo: BHeuristica.java

O protótipo que realiza o Interrogatório Sintomatológico utiliza-se de um Banco de Conhecimento sobre Eventos, para assim procurar otimizar a quantidade de evidência que o usuário do sistema deve avaliar, ou seja, em vez de o usuário ter de avaliar todas as possíveis evidências, o mesmo terá uma evidência com maior probabilidade de ser a correta, de acordo com o que se coletou no Banco de Conhecimento sobre Eventos.

Como a construção do Banco de Conhecimento sobre Eventos depende exclusivamente do conhecimento de um especialista médico para quantificar as probabilidades das crenças e o tipo de relação que existe quando uma certa evidência é observada em conjunto com outras. O protótipo usa um método chamado `simulaBancoDeConhecimento()` para simular as possíveis crenças para as evidências futuras.

O método `simulaBancoDeConhecimento()` gera para cada evidência que será observada um número aleatório entre 0 e 10, para assim quantificar as crenças para as evidências, sendo que a primeira maior crença é sempre a escolhida como sugestão de avaliação pelo usuário. O restante do algoritmo é idêntico ao que simula uma busca em profundidade, sendo que a seguir serão explicados somente os pontos que diferem do algoritmo comentado anteriormente.

No questionamento sobre qual a localização do problema, o algoritmo já pode através do Banco de Conhecimento sobre Eventos, determinar qual das partes do corpo listadas possuem uma maior chance de ser realmente a problemática para o usuário. Sendo que na Figura 23 o sistema supõe através da geração de números aleatórios que o problema se encontra na maior parte das vezes na “Cabeça”.

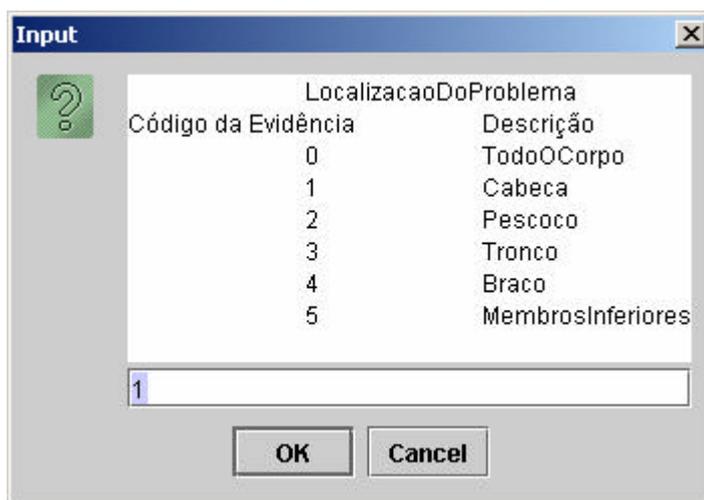


Figura 23 – Sugestão de Evidência para Avaliação pelo Usuário.

O procedimento de perguntas e sugestões é realizado até que o algoritmo atinja um nodo folha, sendo então solicitado que o usuário diga se quer ou não continuar com o questionário. Assim como ocorreu na Figura 21.

3.5 Resultados

Para avaliar o resultado teve-se em mente o que foi proposto na seção 2.8 : “Portanto o parâmetro de desempenho será qual dos algoritmos consegue chegar até as perguntas mais específicas de forma mais rápida”. Outro fator que deve ser levando em consideração na avaliação da velocidade de convergência para um nodo folha é o fato de que as chaves de busca são constantemente fornecidas pelo usuário do sistema. O que limita a velocidade é o próprio usuário, pois sua capacidade de escolha e avaliação das alternativas apresentadas pelo sistema definem em maior parte a velocidade do sistema.

O primeiro algoritmo, o algoritmo de busca em profundidade, tem um desempenho aceitável, desde que o número de alternativas apresentadas não seja muito grande, o que consome muito tempo por parte do usuário na avaliação de cada uma das alternativas.

Já o algoritmo que é proposto neste trabalho, que utiliza um Banco de Conhecimento sobre Eventos, tem uma convergência até os nodos folhas mais rápida, pois apresenta ao usuário uma sugestão da alternativa que deve ser avaliada primeiramente, sendo que tal sugestão se baseia em conhecimentos probabilísticos sobre o que foi observado em eventos passados, portanto existe a possibilidade da sugestão não condizer com o que o usuário possa estar vivenciando no momento, neste caso a eficiência do algoritmo se iguala ao algoritmo de busca em profundidade, pois o usuário terá de avaliar todas as alternativas.

3.5 Sumário e Conclusões

O Interrogatório Sintomatológico foi modelado para garantir uma representação *top-down* do problema. Sendo que o mesmo pode ser representado graficamente como um grafo acíclico e dirigido. Observou-se no entanto que para facilitar a modelagem do mesmo seria mais interessante estruturar o grafo em níveis, com pode ser visto na Figura 24. No entanto tal estruturação só tem objetivos práticos para sua manipulação por seres humanos. Sendo que o mesmo grafo pode ser também visualizado na Figura 25, só que sem qualquer tipo de estruturação.

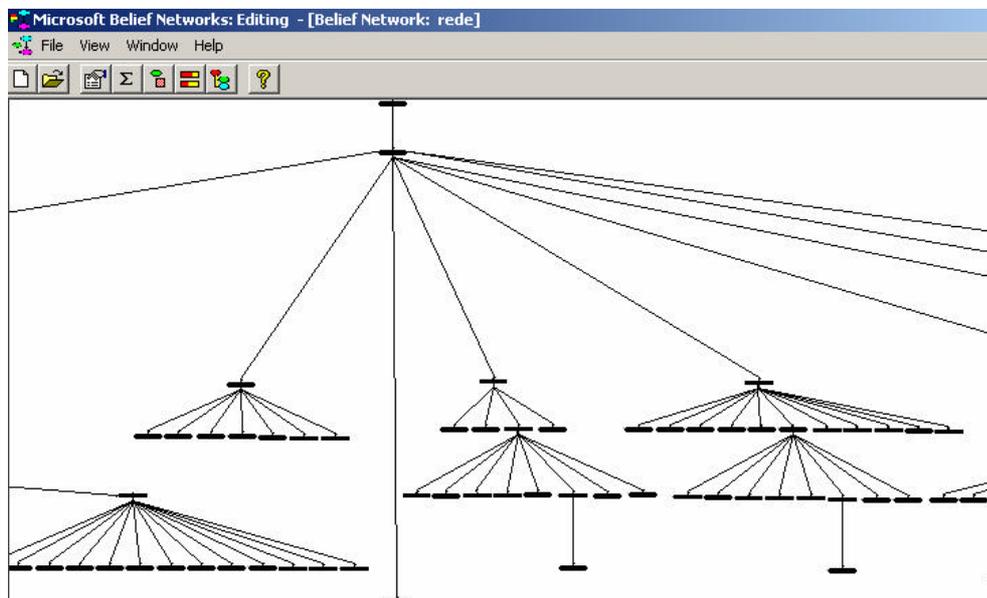


Figura 24 – Representação do Grafo em Níveis Estruturados

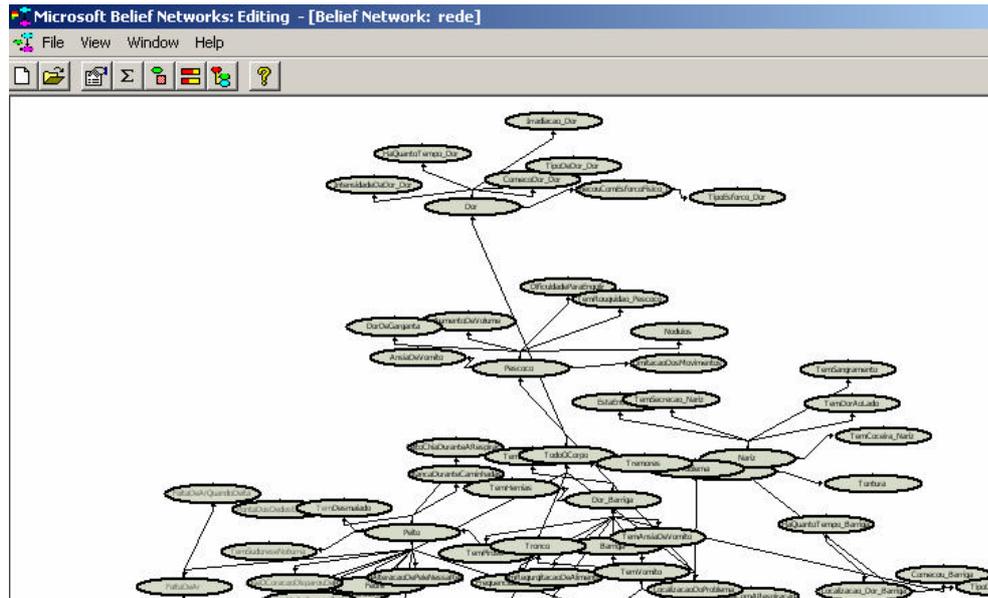


Figura 25 – Representação do Grafo sem os Níveis Estruturados.

O tradutor proposto pode trabalhar com qualquer um dos tipos de representações discutidas acima, pois o mesmo não assume qualquer ação sobre a posição dos nodos. Somente requer que condições especiais, discutidas na seção 2.6, estejam presentes na modelagem da rede.

O suporte a decisão foi implementado usando as classes: `Principal.java`, `TradutorDSC.java`, `Nodo.java`, `GAD.java`, `BProfundidade.java`, `BHeuristica.java`. Onde suas estruturas de dados dinâmicas podem trabalhar em conjunto para a execução dos algoritmos de busca em profundidade e o que utiliza um Banco de Conhecimento sobre Eventos proposto neste trabalho.

Viu-se ainda que ambos os algoritmos baseiam seus critérios de pesquisa exclusivamente nas chaves de busca fornecidas pelo usuário, sendo este o principal fator limitante da velocidade de execução dos mesmos. O fato de algumas evidências gerarem um número razoavelmente grande de novas evidências para serem avaliadas, deixa o algoritmo que se apóia em um Banco de Conhecimento sobre Eventos em vantagem em relação ao de busca em profundidade, pois assim há uma melhora significativa no tempo de resposta do usuário para o sistema.

No capítulo seguinte tem-se as conclusões obtidas na realização deste relatório, assim como propostas para trabalhos futuros, buscando a melhoria do que foi apresentado.

CAPÍTULO 4

4.1 Considerações Finais

Como o trabalho foi dividido basicamente em duas frentes de ação: modelagem da Rede Bayesiana e criação dos algoritmos para manipulação da mesma. Têm-se duas vertentes para serem consideradas.

Na Modelagem da Rede Bayesiana percebeu-se que a compreensão do domínio ao qual pertence o cenário que se está modelando é importante, pois assim pode-se gerar um modelo que represente o cenário real com quantidade de detalhes suficientes para garantir a veracidade do modelo.

A proposta do algoritmo que usa o Banco de Conhecimento sobre Eventos só foi possível como consequência natural da assimilação da natureza do problema que se está modelando, sendo que o mesmo procura somente simular o pensamento probabilístico que o médico realiza durante a execução do Interrogatório Sintomatológico ao paciente.

4.2 Propostas para Trabalhos Futuros

A seguir lista-se um pequeno conjunto de sugestões para trabalhos que podem ser realizados com o intuito de melhorar o que já foi feito.

1. Com a aplicação das definições da *Pervasive Computing* para gerar o *Wearable System*, sistemas que estão intrinsecamente relacionados com os usuários, são necessárias mudanças nas *interfaces* entre homem e máquina, o tipo de interfaces WIMP, *Windows-Icons-Menus-Pointers*, será suficiente para esse tipo de aplicação?
2. Para garantir que os Sistemas para Monitoramento da Saúde Humana tenham um grande alcance no que tange ao número de usuários, os mesmos devem ser capazes de serem executados em vários tipos de dispositivos móveis com capacidade de processamento e armazenamento, como celulares, por exemplo. Sendo que para isso é necessária a conversão de seu código, assim como os códigos dos algoritmos propostos aqui para uma plataforma que suporte tais dispositivos, como *J2ME*.
3. Definição de um módulo para Fusão de Dados, com interfaces bem definidas, permitindo assim a implementação das mais diversas técnicas de modelagem e implementação das mesmas como mostrado na Tabela 1. Sendo que tal estrutura se

assimilaria a técnica de *micro-kernel*, utilizada em sistemas operacionais modernos. Facilitando assim a manutenção e criação de novas funcionalidades. Criando-se uma arquitetura baseada em componentes plugáveis, Figura 26.

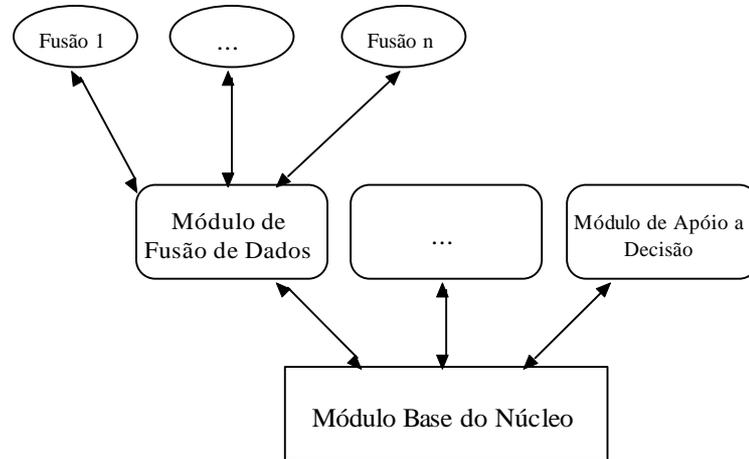


Figura 26 – Módulos Plugáveis.

4. Melhoramento do TradutorDSC para que o mesmo torne-se um tradutor genérico, sendo capaz de trabalhar com redes que possuam mais de um pai por filho, sem limite na linha de probabilidades e criação de uma interface onde seja possível indicar um arquivo do tipo dsc que será carregado para a memória. Permitindo assim o uso de várias Redes Bayesianas ou diferentes formatos de rede.
5. Alteração e criação de um novo método na classe `GAD.java`, para que a mesma implemente algum tipo de algoritmo para propagação de evidências em Redes Bayesianas, permitindo assim que o TradutorDSC e a classe `GAD` possam tornar-se pacotes genéricos para os mais diversos tipos de ambientes modelados.
6. Troca da classe `StringTokenizer` no TradutorDSC, por uma alternativa com suporte na arquitetura *J2ME*, garantindo assim a portabilidade dos pacotes, pois a classe `StringTokenizer` será descontinuada no futuro.
7. Criação de pilhas de evidências intermediárias em cada bifurcação do grafo, diminuindo assim a penalização pela escolha de um caminho incorreto. Ou seja, não será necessário executar o interrogatório desde o início, o mesmo pode ser reiniciado a partir de um ramo intermediário.
8. Criação de um novo módulo para Fusão de Dados, com o objetivo de receber o conteúdo da pilha de evidências e através, de técnicas de *fuzzificação* e *defuzzificação*,

gerar inferências através de um banco de regras para lógica *fuzzy*. Podendo assim criar algum tipo de diagnóstico.

9. Validação dos modelos criados com o auxílio de um médico e utilizando uma população de seres humanos para a criação de um Banco de Conhecimento sobre Eventos real.
10. Uso de sensores biológicos como por exemplo: pressão sanguínea, pulsação, temperatura, glicose no sangue, etc. Garantindo assim que muitas das perguntas feitas pelos algoritmos de caminhamento no grafo sejam puladas, pois a evidência pode ser colida por um sensor artificial, aumentando assim a velocidade na execução dos algoritmos e sua precisão [16].

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] CARVALHO, S. Hervaldo; HEINZELMANN, B. Wendi; MURPHY, Amy L.; COELHO, Claudionor J. N. A General Data Fusion Architecture.
- [2] WALD, L. Definitions and Terms of Reference in Data Fusion.
- [3] <http://www.data-fusion.org/article.php?sid=70> – Acessado 13/04/2004
- [4] Russel, S; Norvig P. Inteligência Artificial, Tradução da Segunda Edição, Editora Campus.
- [5] Forbellone, André L. V.; Eberspächer, Henri F., Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados Ed. Makron Books.
- [6] Pearl, J; Russel, S. Bayesian Networks, Technical Report, R-277, Novembro 2000
- [7] Porto, Celmo C. Semiologia Médica. Editora Guanabara Koogan, Terceira Edição.
- [8] Korhonen, I; Bardram, J. Guest Editorial Introduction to the Especial Section on Pervasive Healthcare., IEEE Transactions on Information Technology in Biomedicine, Vol. 8, N° 3, Setembro 2004
- [9] BAUCHSPIESS, Adolfo. Mini-Curso Sistemas Inteligentes – Redes Neurais e Lógica Fuzzy, Unb, Brasília 2002.
- [10] BILLINGHURST, Mark. Wearable Appliances; The Future of Wearable Computing, Human Interface Technology Laboratory, University of Washington.
- [11] MANN, Steve. Wearable Computing: Toward Humanistic Intellingence, University of Toronto.
- [12] DEVAUL, Rich; SUNG, Michael; GIPS, Jonathan; PENTLAND, Alex “Sandy”. MIThril 2003: Applications and Architecture. Media Laboratory, Massachusetts Institute of Technology.
- [13] Artificial Intelligence in Medicine 30 (2004) 2001-214. Bayesian networks in biomedicine and health-care.

[14] AZUAJE , Francisco; DUBITZKY, Werner; ADAMSON, Kenny. Improving Clinical Decision Support Through Case-Based Data Fusion.

[15] Carneiro, Alexandre L; Silva, Wagner T. Introdução a redes Bayesianas. Relatório de Pesquisa CIC/UnB – 09/99

[16] BARBOSA, M. G. A. Talles; SENE, Jr. G. Iwens; CASTRO, S. S. Liana; BRANISSO, J. P. Henrique; FIGUEREDO, C. Érika; CARVALHO, S. Hervaldo; ROCHA, F. Adson; NASCIMENTO, A. O. Francisco. Sistema Pessoal Móvel de Monitoração da Saúde: Algoritmo para Captura Inteligente de Sintomas.

[17] NETO, C. S. Olegário; JOSÉ, B. Alexandre; ALVES, H. J. Paulo; BARBOSA, M. G. A. Talles; SENE, Jr. G. Iwens; FERREIRA, O. José; ALCÂNTARA, H. Uma Proposta para o Melhoramento do *GlucoTools*.

ANEXOS

Anexo A – Interrogatório Sintomatológico

1. Vc está sentindo alguma coisa diferente?
ou tem algo que te incomoda?
ou tem algum problema? Sim Não
2. Qual a localização?
3. Todo o corpo
4. Cabeça
5. PESCOÇO
6. Tronco
7. Braço
8. Pernas
4. Todo o corpo
 - 4.1. Tem febre? Sim Não
 - 4.1.1. Qual a temperatura? 37°C 37,5°C 38°C 38,5°C 39°C 39,5°C 40°C
40,5°C 41°C
 - 4.1.2. Começou há: 1h 3h 6h 12h 1dia 2 dias mais de 2 dias
 - 4.1.3. Qual o horário em que aparece mais? O dia inteiro manhã tarde noite
madrugada
 - 4.2. Tem dor? Sim Não
 - 4.2.1. Começou: de repente gradualmente
 - 4.2.2. Intensidade: fraca moderada intensa muito intensa
 - 4.2.3. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em
peso contínua
 - 4.2.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias
mais de 2 dias
 - 4.2.5. Irradiação (????)
 - 4.2.6. Começou com esforço físico ? Sim Não
 - 4.2.6.1. pequenos esforços médios esforços grandes esforços
 - 4.3. Tem fraqueza? Sim Não
 - 4.3.1. Sentiu tontura? Sim Não
 - 4.3.1.1. Teve necessidade de se apoiar? Sim Não
 - 4.3.1.2. Caiu sem perder a consciência? Sim Não
 - 4.3.1.3. Chegou a perder a consciência? Sim Não
 - 4.4. Tem tremores? Sim Não
 - 4.4.1. São: leves fortes muito forte
 - 4.5. Tem alterações da pele? Sim Não
 - 4.5.1. Tem coceira? Sim Não
 - 4.5.2. Tem manchas? Sim Não
 - 4.5.2.1. São: Vermelhas azuis brancas amarelas verdes escuras
 - 4.5.3. Tem feridas? Sim Não
 - 4.5.3.1. Sangram? Sim Não
 - 4.5.3.2. Tem crostas? Sim Não
 - 4.5.3.3. Tem pus? Sim Não
 - 4.5.3.4. Tem descamação? Sim Não
 - 4.5.4. Tem caroços? Sim Não
 - 4.5.5. Sua muito (anormal)? Sim Não
 - 4.5.6. Tem alterações nos pelos? Sim Não
 - 4.5.6.1. Tem: queda de pelos excesso de pelos

4.5.7. Tem alterações nas unhas? Sim Não

4.5.8. Tem veias visíveis? Sim Não

4.5.8.1. deitado? Em pé?

5. Cabeça

5.1. Onde fica? na face (inclui o ouvido) .2 resto da cabeça .1

5.1.1. Tem dor de cabeça? Sim Não

5.1.1.1. Começou: de repente gradualmente

5.1.1.2. Intensidade: fraca moderada intensa muito intensa

5.1.1.3. Dor: em queimação em pontadas pulsátil em aperto fisgada em peso contínua

5.1.1.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias mais de 2 dias

5.1.1.5. Onde dói? Toda cabeça na testa no lado esquerdo no lado direito na nuca

5.1.1.6. Irradiação (????)

5.1.1.7. Tem relação com: ansiedade menstruação leitura tosse exercício posição e/ou movimentos (pergunta tipo marcar x)

5.1.1.8. Está associado com: distúrbios visuais ânsia de vômito vômito

5.1.2. Tem tontura? Sim Não

5.1.3. Tem vertigens? Sim Não

5.1.4. Bateu a cabeça? Sim Não

5.1.5. Tem perda de memória? Sim Não

5.1.5.1. de fatos antigos de fatos recentes

5.1.6. Tem coceira? Sim Não

5.1.7. Tem insônia? Sim Não

5.1.8. Tem sonolência? Sim Não

5.1.9. Tem alguma ferida na cabeça? Sim Não

5.1.9.1. Sai sangue? Sim Não

5.1.9.2. Sai pus? Sim Não

5.1.9.3. Há perda de cabelo no local? Sim Não

5.1.9.4. Tem descamação? Sim Não

5.1.10. Há perda excessiva de cabelo? Sim Não

5.1.11. Tem algum caroço? Sim Não

5.1.11.1. é doloroso? Sim Não

5.1.12. Você compreende bem o sentido das palavras? Sim Não

5.2. Na face (inclui o ouvido):

5.2.1. Fica: na pele (.1) nos olhos (.2) nos ouvidos (.3) no nariz (.4) na boca (.5) (alternativas não excludentes)

5.2.1.1. Na pele:

5.2.1.1.1. Tem inchaço? Sim Não

5.2.1.1.2. Tem coceira? Sim Não

5.2.1.1.3. Tem manchas? Sim Não

5.2.1.1.3.1. São: Vermelhas azuis brancas amarelas verdes escuras

5.2.1.1.4. Tem feridas? Sim Não

5.2.1.1.4.1. Sangram? Sim Não

5.2.1.1.4.2. Tem crostas? Sim Não

5.2.1.1.4.3. Tem pus? Sim Não

5.2.1.1.4.4. Tem descamação? Sim Não

- 5.2.1.1.5. Tem caroços? Sim Não
- 5.2.1.1.6. Sua pele é: seca normal oleosa
- 5.2.1.1.7. Tem alterações nos pelos? Sim Não
- 5.2.1.1.7.1. Tem: queda de pelos excesso de pelos
- 5.2.1.2. Nos olhos:
- 5.2.1.2.1. Está com dificuldade de enxergar? Sim Não
- 5.2.1.2.1.1. de longe de perto
- 5.2.1.2.2. Sua vista está embaçada? Sim Não
- 5.2.1.2.3. Vê pontos na visão? Sim Não
- 5.2.1.2.4. Tem visão das coisas? Sim Não
- 5.2.1.2.5. Tem inchaço? Sim Não
- 5.2.1.2.6. Tem vermelhidão? Sim Não
- 5.2.1.2.7. Tem coceira? Sim Não
- 5.2.1.2.8. A parte branca do seu olho está amarela? Sim Não
- 5.2.1.2.9. Tem hipersensibilidade à luz? Sim Não
- 5.2.1.2.10. Tem secreção ocular? Sim Não
- 5.2.1.2.11. Seu olho está seco? Sim Não
- 5.2.1.2.12. Tem dor? Sim Não
- 5.2.1.2.12.1. Começou: de repente gradualmente
- 5.2.1.2.12.2. Intensidade: fraca moderada intensa muito intensa
- 5.2.1.2.12.3. Dor: em queimação em pontadas pulsátil em aperto
fisgada em peso contínua
- 5.2.1.2.12.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h
1dia 2 dias mais de 2 dias
- 5.2.1.2.12.5. Irradiação (????)
- 5.2.1.2.12.6. Tem relação com: ansiedade menstruação leitura tosse
exercício posição e/ou movimentos (pergunta tipo marcar x)
- 5.2.1.2.13. Tem alucinações visuais? Sim Não
- 5.2.1.3. Nos ouvidos:
- 5.2.1.3.1. Tem dor? Sim Não
- 5.2.1.3.1.1. Começou: de repente gradualmente
- 5.2.1.3.1.2. Intensidade: fraca moderada intensa muito intensa
- 5.2.1.3.1.3. Dor: em queimação em pontadas pulsátil em aperto
fisgada em peso contínua
- 5.2.1.3.1.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h
1dia 2 dias mais de 2 dias
- 5.2.1.3.1.5. Irradiação (????)
- 5.2.1.3.1.6. Tem relação com: ansiedade menstruação leitura tosse
exercício posição e/ou movimentos (pergunta tipo marcar x)
- 5.2.1.3.2. Tem zumbido? Sim Não
- 5.2.1.3.2.1. Está associado a exposição ao barulho? Sim não
- 5.2.1.3.3. Tem secreção? Sim Não
- 5.2.1.3.3.1. É: cera líquida sangue pus
- 5.2.1.3.4. Você está ouvindo bem? (ou as pessoas estão tendo dificuldade
em se comunicar com você?) Sim Não
- 5.2.1.3.5. Você compreende bem o sentido das palavras? Sim Não
- 5.2.1.4. No nariz:
- 5.2.1.4.1. Tem dor ao lado do nariz? Sim Não
- 5.2.1.4.2. Tem sangramento nasal? Sim Não

- 5.2.1.4.3. Tem secreção nasal? Sim Não
 - 5.2.1.4.3.1.É: líquida catarro
- 5.2.1.4.4. Tem coceira? Sim Não
- 5.2.1.4.5. Está entupido? Sim Não
- 5.2.1.5. Na boca (inclui os dentes):
 - 5.2.1.5.1. Tem dor? Sim Não
 - 5.2.1.5.1.1. de dente na gengiva em outro lugar da boca
 - 5.2.1.5.2. Tem sangramento? Sim Não
 - 5.2.1.5.3. Tem feridas? Sim Não
 - 5.2.1.5.4. Tem manchas? Sim Não
 - 5.2.1.5.4.1. São: Vermelhas brancas amarelas escuras
 - 5.2.1.5.5. Tem boca seca? Sim Não
 - 5.2.1.5.6. Tem salivação em excesso? Sim Não
 - 5.2.1.5.7. Tem mau hálito constante? Sim Não
 - 5.2.1.5.8. Tem alteração da sensibilidade da língua? Sim Não
 - 5.2.1.5.9. Está com dificuldade na fala? Sim Não
 - 5.2.1.5.10. Tem dor de garganta? Sim Não (testar!)
 - 5.2.1.5.11. Tem dificuldade para engolir? Sim Não
 - 5.2.1.5.12. Tem rouquidão? Sim Não
 - 5.2.1.5.13. Tem muita sede? Sim Não
 - 5.2.1.5.14. Está com dificuldade para falar? Sim Não

6. Pescoço:

- 6.1. Tem dor de garganta? Sim Não
- 6.2. Tem nódulos? Sim Não
- 6.3. Tem aumento de volume do pescoço? Sim Não
- 6.4. Tem limitação dos movimentos? Sim Não (ou rigidez?)
- 6.5. Tem rouquidão? Sim Não
- 6.6. Tem dificuldade para engolir? Sim Não
- 6.7. Tem ânsia de vômitos? Sim Não

7. Tronco:

- 7.1. O que você sente está: no peito (acima das costelas)(.1) na barriga(abaixo das costelas)(.2) nas costas(.3) na região da genitália e virilha(.4)
 - 7.1.1. No peito:
 - 7.1.1.1. Você está sentindo dor? Sim Não
 - 7.1.1.1.1. Essa dor varia com a respiração? Sim Não
 - 7.1.1.1.2. Onde se localiza a dor ????
 - 7.1.1.1.3. Começou: de repente gradualmente
 - 7.1.1.1.4. Intensidade: fraca moderada intensa muito intensa
 - 7.1.1.1.5. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em peso contínua
 - 7.1.1.1.6. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h
1dia 2 dias mais de 2 dias
 - 7.1.1.1.7. Irradia: Sim Não
 - 7.1.1.1.7.1. Irradia para: o pescoço o ombro esquerdo o ombro direito braço esquerdo braço direito barriga costas cabeça
 - 7.1.1.1.8. Começou com esforço físico ? Sim Não
 - 7.1.1.1.8.1. pequenos esforços médios esforços grandes esforços
 - 7.1.1.2. Você já sentiu que o seu coração disparou de repente? Sim Não

- 7.1.1.3. Você já achou que seu coração tinha parado de bater? Sim não
- 7.1.1.4. Você tem tosse ? Sim não
- 7.1.1.4.1. A tosse: é seca tem escarro branco tem escarro amarelo
- 7.1.1.4.2. No escarro tem sangue? Sim não
- 7.1.1.5. Tem falta de ar? Sim não
- 7.1.1.5.1. Essa falta de ar ocorre: em repouso ao esforço mínimo ao esforço moderado ao esforço grande
- 7.1.1.5.2. Tem falta de ar quando deita? Sim não
- 7.1.1.5.3. Acorda à noite com falta de ar? Sim não
- 7.1.1.6. O seu peito chia durante a respiração? Sim não
- 7.1.1.7. A ponta dos seus dedos está azul? Sim não
- 7.1.1.8. Começa a mancar durante a caminhada ? Sim não
- 7.1.1.9. Tem varizes nas pernas? Sim não
- 7.1.1.10. Tem desmaiado? Sim não (ver cabeça?)
- 7.1.1.11. Sente tonturas quando levanta? Sim não
- 7.1.1.11.1. Sente tontura minutos depois de se levantar? Sim não
- 7.1.1.12. Tem sudorese noturna? Sim não
- 7.1.1.13. Passou muito tempo sentada ? Sim não
- 7.1.1.14. tem alguma alteração da pele nessa região? Sim não
- 7.1.2. Na barriga:
- 7.1.2.1. Tem dor ? Sim não
- 7.1.2.1.1. Onde se localiza essa dor?

— — —
— — —
- ou na boca do estômago do seu lado direito do seu lado esquerdo
- na região do umbigo do seu lado direito do seu lado esquerdo
- na região abaixo do umbigo do seu lado direito do seu lado esquerdo
- 7.1.2.1.2. Começou: de repente gradualmente
- 7.1.2.1.3. Intensidade: fraca moderada intensa muito intensa
- 7.1.2.1.4. Dor: em queimação em pontadas pulsátil cólica em aperto
fisgada em peso contínua
- 7.1.2.1.5. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h
1dia 2 dias mais de 2 dias
- 7.1.2.1.6. Irradiação (????)
- 7.1.2.1.7. Começou com esforço físico ? Sim Não
- 7.1.2.1.7.1. pequenos esforços médios esforços grandes esforços
- 7.1.2.2. Tem pirose ? (sensação de queimação da garganta à boca do estômago ou tem azia)
- 7.1.2.3. O seu apetite está: normal aumentado diminuído
- 7.1.2.4. Tem ânsia de vômito? Sim não
- 7.1.2.5. Tem vômitos? Sim não
- 7.1.2.5.1. O vômito tem sangue? Sim não
- 7.1.2.5.2. O vômito tem resto de alimentos? Sim não
- 7.1.2.6. Tem regurgitação de alimentos? Sim não
- 7.1.2.7. A frequência com que está evacuando está: normal aumentada (diarréia)
diminuída (constipação)
- 7.1.2.8. Tem hemorróidas? Sim não
- 7.1.2.9. Tem hérnias? Sim Não
- 7.1.3. Nas costas:

7.1.3.1. Tem dor? Sim Não

7.1.3.1.1. De que lado? Na coluna lado direito lado esquerdo toda as costas

7.1.3.1.2. qual a parte? Da metade pra cima da metade pra baixo

7.1.3.1.3. Começou: de repente gradualmente

7.1.3.1.4. Intensidade: fraca moderada intensa muito intensa

7.1.3.1.5. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em peso contínua

7.1.3.1.6. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h
1dia 2 dias mais de 2 dias

7.1.3.1.7. Irradiação (????)

7.1.3.1.8. Começou com esforço físico ? Sim Não

7.1.3.1.8.1. Pequenos esforços médios esforços grandes esforços

7.1.4. Na região da genitália e virilha:

7.1.4.1. Tem dor pra urinar? Sim Não

7.1.4.2. A quantidade da urina é: normal aumentada diminuída

7.1.4.3. A cor da urina está: amarelo clara amarelo escura avermelhada
branca preta (cor de coca-cola)

7.1.4.4. Acorda à noite para urinar? Sim Não

7.1.4.5. Tem urgência urinária? Sim Não

7.1.4.6. Elimina pedras na urina? Sim Não

Sexo feminino:

7.1.4.7. A menstruação está regular? Sim Não

7.1.4.8. Quando foi a última menstruação? Dd/mm/aaaa não sei

7.1.4.9. A duração é: menor que 2 dias de 3 a 7 dias maior que 8 dias

7.1.4.10. A relação sexual com seu parceiro é dolorosa? Sim Não

7.1.4.11. Está com corrimento? Sim Não

7.1.4.11.1. branco amarelado com sangue

7.1.4.12. Tem coceira na vagina? Sim Não

7.1.4.13. Tem alteração da libido? Sim Não

7.1.4.14. Tem feridas? Sim Não

Sexo masculino:

7.1.4.15. Tem alteração da libido? Sim Não

7.1.4.16. Tem dificuldade de ereção? Sim Não

7.1.4.17. Tem dor nos testículos? Sim Não

7.1.4.18. Tem caroços nos testículos? Sim Não

7.1.4.19. Tem corrimentos no pênis? Sim Não

7.1.4.20. Tem ejaculação: normal precoce dolorosa

7.1.4.21. Tem feridas? Sim Não

8. Nos braços:

8.1. Tem inchaço? Sim Não

8.2. Tem dor? Sim Não

8.2.1. Começou: de repente gradualmente

8.2.2. Intensidade: fraca moderada intensa muito intensa

8.2.3. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em peso contínua

- 8.2.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias
mais de 2 dias
- 8.2.5. Irradiação (????)
- 8.2.6. Começou com esforço físico ? Sim Não
- 8.2.6.1. pequenos esforços médios esforços grandes esforços
- 8.2.7. Localização: ombro braço cotovelo antebraço punho mão dedos
- 8.2.8. Sofreu alguma pancada? Sim Não
- 8.3. Tem tremores? Sim Não
- 8.4. Tem dificuldade para mexer? Sim Não
9. Nos membros inferiores:
- 9.1. Tem inchaço? Sim Não
- 9.2. Tem dor? Sim Não
- 9.2.1. Começou: de repente gradualmente
- 9.2.2. Intensidade: fraca moderada intensa muito intensa
- 9.2.3. Dor: em queimação em pontadas pulsátil cólica em aperto fisgada em peso contínua
- 9.2.4. Há quanto tempo? 5min 15min 30min 1h 3h 6h 12h 1dia 2 dias
mais de 2 dias
- 9.2.5. Irradiação (????)
- 9.2.6. Começou com esforço físico ? Sim Não
- 9.2.6.1. pequenos esforços médios esforços grandes esforços
- 9.2.7. Localização: coxa joelho batata da perna pé tornozelo
- 9.2.8. Sofreu alguma pancada? Sim Não
- 9.3. Tem tremores? Sim Não
- 9.4. Tem dificuldade para andar? Sim Não
- 9.5. Começa a mancar durante a caminhada ? Sim não
- 9.6. Tem fraqueza nas pernas? Sim não
- 9.7. A panturrilha dói quando você aperta? Sim Não
- 9.8. A panturrilha está solta? Sim não
- 9.9. Consegue dobrar o pé para frente e para trás sem dor? Sim Não
- 9.10. Tem dormência constante? Sim Não
- 9.11. Tem varizes? Sim Não

Finalmente, como você está se sentindo? Alegre triste Disposto cansado
ansioso

Anexo B – TradutorDSC.java

```

import java.io.*;
import java.util.*;

public class TradutorDSC
{
    private File NomeArquivo = new File("rede.dsc");
    private BufferedReader entrada;
    private StringBuffer buffer = new StringBuffer();
    private String texto = new String();
    private StringTokenizer tokens = new StringTokenizer(texto);
    private String tk = new String();
    //Nodos, evidencias e probabilidades sao variaveis de depuracao
    private int nodos,evidencias,probabilidades,aux;
    private boolean PontoEVirgula = false;
    private String Pai,Filho,Raiz = new String();
    private Vector evi= new Vector();
    private String e[];
    private float p[];
    public Nodo raiz, pai, filho;

    public static GAD Grafo = new GAD();

    public void ListarArquivo()
    {
        try
        {
            entrada = new BufferedReader(new
FileReader(NomeArquivo));
        }
        //Pega excecoes na E/S

        catch(IOException ioExcecoes)
        {
            System.out.println("Erro na abertura do Arquivo!");
        }

        try
        {
            while( (texto = entrada.readLine()) != null)
            {
                tokens = new StringTokenizer(texto);
                //IMPORTANTE: A organizacao do arquivo .dsc garante q os nomes dos nodos
                //serao os primeiros a serem processados.
                while(tokens.hasMoreTokens())
                {
                    tk = tokens.nextToken();
                    //Para descoberta de variaveis e suas
evidencias

                    if (tk.equals("node"))
                    {
                        //Tem-se o nome do nodo em tk
                        tk = tokens.nextToken();

                        Nodo no = new Nodo(tk,tk);
                        Grafo.Insere(no);
                        nodos++;

                        //Pula as linha 'inuteis'
                        texto = entrada.readLine();
                    }
                }
            }
        }
    }
}

```

```

    texto = entrada.readLine();
    texto = entrada.readLine();
    texto = entrada.readLine();
    //Para na linha da primeira evidencia:
    texto = entrada.readLine();

um nodo
    //Pega todas a evidencias relacionadas a
    while (texto.endsWith(";") != true)
    {
        texto = texto.trim();

        //Acha a Aspas e retira da string
        texto =
texto.substring(texto.indexOf(34)+1,
        texto.indexOf(34,texto.indexOf(34)+1));

        evi.add(texto.trim());
        texto = entrada.readLine();
        evidencias++;
    }
    //Cria um novo array com o tamanho certo
    e = new String[evidencias];
    evi.toArray(e); //Transforma o Vector em
array

    //Aqui as evidencias entram no nodo
    Grafo.acharNo(tk).fixarEvidencias(e);
    //Zerar o vetor de evidencias para
receber

    //novas evidencias
    evi.removeAllElements();
    evidencias = 0; //Zera o contador de
evidencias
    }
    else
    {
/*LEMBRAR:Pensar em como colocar as probabilidades no Nodo
lembrando-se q precisamos achar o nodo correto no Grafo
Montar um Matriz com as probabilidades e passar para o Nodo
A matriz sera Nx2 sendo q N = numero de evidencias no vetor de evidencias!
Obs: Nx2 vale para o dominio do interrogatorio pois so ha um pai por
filho*/

    //Para descoberta das probabilidades das
variaveis
    aux = tk.indexOf("probability");//-1 se
nao achou

    if (aux != -1) //Achou
    {/*FUNCIONA POIS POR ENQUANTO SO TEMOS
UM PAI POR FILHO
INTERROGATORIO*/

        //Para achar o(s) Filho(s)
        //aux agora tem o inicio do Pai
        aux = tk.indexOf("|");

        //Se tiver pai

```

```

tk.substring(12,aux);
2-upla
tk.substring(aux+1,tk.indexOf(""));
Nodos
recebeFilho(Grafo.acharNo(Filho));

if (aux != -1)
{
    Filho =
        //Para achar o Pai
        //Sempre e o ultimo item na
        //(x|pai)
        Pai =
            //Acha o Pai no vetor de
            //e coloca seu filho
            Grafo.acharNo(Pai).
}
else
{//E a raiz e nao tem pai!
    Raiz =
    Pai = null;
    //Temos a raiz do Grafo!
}

//LEMBRAR!: Nesse ponto ja posso localizar o nodo no Grafo (Vetor)
//probability(xx|yy) xx representa o nome do nodo (SearchKey)
//xx e o Filho

//Pula as linha 'inuteis'
texto = entrada.readLine();
texto = entrada.readLine();

//IMPORTANTE: assume-se que as probabilidades não passam de um linha no
texto
//SENAO: gera erro no token

de probabilidades

StringTokenizer(texto);
(tokens.hasMoreTokens())

preciso pular as linhas
existem mesmo

if (texto.startsWith("\tdefault"))
//O default e desnecessario
    texto = entrada.readLine();
//Agora estamos na primeira linha

while(texto.endsWith("}") != true)
{
    tokens = new
    while
    {
        //Como e sem pai nao
        //pois elas nao
        if (Pai != null)
        {

```


Anexo C – Nodo.java

```

import javax.swing.JTextArea;
import java.util.*;

public class Nodo
{

    //Vetor de Ponteiros para os Filhos do Nodo
    public Vector NodosFilhos;

    public String DescricaoNodo; //Descricao Textual do Nodo
    public String NomeNodo; //Nome para identificacao do Nodo

    public String Evidencias[];
    public float Probabilidades[][];

//Metodos

    public Nodo(String Desc,String NomeN)
    {
        this.DescricaoNodo = new String(Desc);
        this.NomeNodo = new String(NomeN);
        this.NodosFilhos = new Vector(1);
    }

    public JTextArea imprimeEvidencias()
    {
        JTextArea vetorEvidencias = new JTextArea();
        vetorEvidencias.setText("\t" + this.DescricaoNodo +
            "\nCódigo da Evidência\tDescrição\n");

        for(int aux = 0; aux < this.Evidencias.length; aux++)
            vetorEvidencias.append("\t" + aux + "\t" +
                this.Evidencias[aux] + "\n");

        return(vetorEvidencias);
    }

    public int quantasEvidencias()
    {
        return(this.Evidencias.length);
    }

    public int quantosFilhos()
    {
        return(this.NodosFilhos.size());
    }

    public boolean temFilhos()
    {
        return(!(this.NodosFilhos.isEmpty()));
    }

    public void recebeFilho(Nodo ref) //Referencia do nodo filho
    {
        this.NodosFilhos.addElement(ref);
    }

    public void removeFilho(Nodo ref)
    {
        this.NodosFilhos.removeElement(ref);
    }
}

```

```
}

public int ondeEsta(Nodo ref) //Referencia do nodo filho
{
    return(this.NodosFilhos.indexOf(ref));
}

public void fixarEvidencias(String evi[])
{
    this.Evidencias = evi;
}

public void fixarProbabilidades(float p[[[]])
{
    this.Probabilidades = p;
}

public JTextArea listarFilhos()
{
    Nodo x;

    JTextArea filhosNodo = new JTextArea();
    filhosNodo.setText("\tFilhos de " + this.DescricaoNodo +
        "\nCódigo do Filho \tDescrição\n");

    for(int aux = 0; aux < this.NodosFilhos.size(); aux++)
    {
        x = (Nodo)this.NodosFilhos.elementAt(aux);
        filhosNodo.append("\t" + aux + "\t" +
            x.NomeNodo + "\n");
    }
    return(filhosNodo);
}
}
```

Anexo D – GAD.java

```

import java.util.*;

public class GAD //Grafo Aciclico Dirigido
{
    private Vector Nodos;
    private Nodo x;

    //Metodos
    public GAD()
    {
        this.Nodos = new Vector(1);
    }

    public void Insere(Nodo ref)
    {
        this.Nodos.addElement(ref);
    }

    public int ContaNos()
    {
        return(this.Nodos.size());
    }

    public void imprimeGAD()
    {
        for (int aux = 0; aux < this.Nodos.size();aux++)
        {
            //IMPORTANTE: Forcar o tipo (Nodo) +- 4 Horas
            //desp. p. descobrir isso!!!
            x = (Nodo)this.Nodos.elementAt(aux);
            System.out.println(x.NomeNodo);
        }
    }

    public Nodo acharNo(String chave)
    {
        int aux = 0;

        do
        {
            x = (Nodo)this.Nodos.elementAt(aux);
            aux++;
        }while(aux < this.Nodos.size() && (x.NomeNodo.equals(chave) !=
true));

        if(this.x.NomeNodo.equals(chave) == false)
            return null;
        else
            return x;
    }
}

```

Anexo E – BProfundidade.java

```

import javax.swing.JOptionPane;
import javax.swing.JTextArea;

public class BProfundidade
{
    private boolean maisProblemas = true;
    private boolean perguntar = true;
    private Nodo aux;
    private String opc;
    private int opcao;
    public JTextArea EvidenciasColetadas = new JTextArea();

    //public void ListarEvidencias()
    public void Buscar()
    {
        EvidenciasColetadas.setText(
            "      EVIDENCIAS COLETADAS PELO QUESTIONARIO      \n");

        //Raiz é 'Problema'
        //Objeto declarado e inicializado no TradutorDSC
        aux = TradutorDSC.Grafo.acharNo("Problema");
        opc = JOptionPane.showInputDialog("Você possui algum "
            + aux.DescricaoNodo + "? 1-Sim 2-Não");

        System.out.println("Voce escolheu o nodo " + aux.NomeNodo
            + "E temFilhos = " + aux.temFilhos());

        if (opc.equals("2"))
        {
            JOptionPane.showMessageDialog(null,"OK! Questionário
Encerrado");
            //Coloca a informacao na pilha de Evidencias

            EvidenciasColetadas.append("Sem Problemas\n");
            perguntar = false;
        }
        else
        {
            aux = TradutorDSC.Grafo.acharNo("LocalizacaoDoProblema");
            EvidenciasColetadas.append("Localização do Problema\n");
        }

        do
        {
            while(perguntar)
            {
                if(aux.temFilhos() == true)
                    perguntar = true;
                else
                    perguntar = false;

                //E se não tiver mais filhos mas tem evidencias?
                //Se for uma evidencia do tipo S/N
                if(aux.Evidencias[0].equals("Yes"))
                {

```

```

perguntas                                     //Entao deve-se listar todos os filhos como
                                              //Retorna uma referencia para
                                              //um JTextArea --> aux.imprimeEvidencias()

                                              if(aux.temFilhos() == true)
                                              {
filho de "                                     opc = JOptionPane.showInputDialog(null,
                                              aux.listarFilhos(),"Ir para qual
                                              + aux.DescricaoNodo);
                                              opcao = Integer.parseInt(opc);
pelo usuario                                 //Coloca aux como o novo nodo escolhido
                                              aux =
(Nodo)aux.NodosFilhos.elementAt(opcao);

                                              EvidenciasColetadas.append(" " +
aux.NomeNodo + "\n");

                                              System.out.println("Voce escolheu o nodo
"
                                              + aux.NomeNodo
                                              + " E temFilhos = " +
aux.temFilhos());

                                              }
                                              }
else
{
                                              //Retorna uma referencia para
                                              //um JTextArea --> aux.imprimeEvidencias()

                                              opc = JOptionPane.showInputDialog(null,
"                                              aux.imprimeEvidencias(),"Evidência para
                                              + aux.DescricaoNodo);
                                              opcao = Integer.parseInt(opc);
                                              //se o problema só tem duas evidencias y e n
                                              //então é necessário listar todos os filhos

para escolher                                //Vai para o nodo correspondente a evidência

                                              EvidenciasColetadas.append(" "
                                              + aux.Evidencias[opcao] + "\n");

                                              aux =
TradutorDSC.Grafo.acharNo(aux.Evidencias[opcao]);
                                              }
}

opc = JOptionPane.showInputDialog(null,
"Você tem mais algum problema? 1-Sim 2-Não");
opcao = Integer.parseInt(opc);

if(opcao == 1)
{
    maisProblemas = true;
    perguntar = true;
}

```


Anexo F – Bheuristica

```

import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import javax.swing.JLabel;
import java.lang.Math;

public class BHeuristica
{
    private boolean maisProblemas = true;
    private boolean perguntar = true;
    private Nodo aux;
    private String opc, saidaCrenças;
    private int opcao, maiorCrença;
    public JTextArea EvidenciasColetadas = new JTextArea();
    private JLabel Etiqueta = new JLabel();
    private int Crenças[];

    public void carregaBancoDeConhecimento()
    {
        /*Carregar de um arquivo todos as crenças relacionados a
        evidencia que foi observada. Tal banco e fornecido por
        um especialista no assunto*/
    }

    //Metodo para simular a geracao das crenças
    public int simulaBancoDeConhecimento()
    {
        return((int)(10 * Math.random()));
    }

    public void Buscar()
    {
        EvidenciasColetadas.setText(
            "      EVIDENCIAS COLETADAS PELO QUESTIONARIO      \n");

        JOptionPane.showMessageDialog(null, "Numero: "
            + simulaBancoDeConhecimento());

        //Raiz é 'Problema'
        //Objeto declarado e inicializado no TradutorDSC
        aux = TradutorDSC.Grafo.acharNo("Problema");
        opc = JOptionPane.showInputDialog("Você possui algum "
            + aux.DescricaoNodo + "? 1-Sim 2-Não");

        System.out.println("Voce escolheu o nodo " + aux.NomeNodo
            + "E temFilhos = " + aux.temFilhos());

        if (opc.equals("2"))
        {
            JOptionPane.showMessageDialog(null, "OK! Questionário
Encerrado");
            //Coloca a informacao na pilha de Evidencias

            EvidenciasColetadas.append("Sem Problemas\n");
            perguntar = false;
        }
        else
        {
            aux = TradutorDSC.Grafo.acharNo("LocalizacaoDoProblema");

```

```

EvidenciasColetadas.append("Localização do Problema\n");
    }
do
{
    while(perguntar)
    {
        if(aux.temFilhos() == true)
            perguntar = true;
        else
            perguntar = false;

        //E se não tiver mais filhos mas tem evidencias?
        //Se for uma evidencia do tipo S/N
        if(aux.Evidencias[0].equals("Yes"))
        {
            //Entao deve-se listar o filho mais provavel

            //Baseado no Banco de Conhecimento Sobre

            //Retorna uma referencia para
            //um JTextArea --> aux.imprimeEvidencias()

            if(aux.temFilhos() == true)
            {
                //Crenças = aux.listarFilhos();

                //Novo vetor de inteiros para Crenças
                Crenças = new int[aux.quantosFilhos()];

                System.out.println(aux.quantosFilhos());

                for(int i = 0;
                    i <= aux.listarFilhos().getLineCount() -
4; i++)
                    Crenças[i] =

simulaBancoDeConhecimento();

                saidaCrenças = "Crenças\n";

                //Coloca as crenças no TextArea e ja

                //Fazer para Evidencias tambem
                maiorCrença = 0;
                for(int i = 0; i < Crenças.length; i++)
                {
                    saidaCrenças += Crenças[i] + "\n";

                    if(Crenças[i] > Crenças[maiorCrença])
                        maiorCrença = i;
                }

                JOptionPane.showMessageDialog(null, saidaCrenças);

                opc = JOptionPane.showInputDialog(null,
                    aux.listarFilhos(), " +

maiorCrença);

                opcao = Integer.parseInt(opc);

```

```

//Coloca aux como o novo nodo escolhido
pelo usuario
    aux =
(Nodo)aux.NodosFilhos.elementAt(opcao);

EvidenciasColetadas.append(" " +
aux.NomeNodo + "\n");

System.out.println("Voce escolheu o nodo
"
+ aux.NomeNodo
+ " E temFilhos = " +
aux.temFilhos());
}
} //Aqui temos as evidencias de um nodo que devem
ser escolhidas
else
{
//Retorna uma referencia para
//um JTextArea --> aux.imprimeEvidencias()

//Novo vetor de inteiros para Crenças
Crenças = new int[aux.quantasEvidencias()];

System.out.println(aux.quantosFilhos());
System.out.println("Numero de Evidencias no NODO"
+ aux.quantasEvidencias());

for(int i = 0;
4; i++)
    i <= aux.imprimeEvidencias().getLineCount() -
Crenças[i] = simulaBancoDeConhecimento();

saidaCrenças = "Crenças\n";

//Coloca as crenças no TextArea e ja escolhe a
maior
    maiorCrença = 0;
for(int i = 0; i < Crenças.length; i++)
{
    saidaCrenças += Crenças[i] + "\n";

    if(Crenças[i] > Crenças[maiorCrença])
        maiorCrença = i;
}

opc = JOptionPane.showInputDialog(null,
maiorCrença);
aux.imprimeEvidencias(), "" +
opc = Integer.parseInt(opc);

EvidenciasColetadas.append(" "
+ aux.Evidencias[opcao] + "\n");

aux =
TradutorDSC.Grafo.acharNo(aux.Evidencias[opcao]);
}
}

```

```
opc = JOptionPane.showInputDialog(null,
    "Você tem mais algum problema? 1-Sim 2-Não");
opcao = Integer.parseInt(opc);

if(opcao == 1)
{
    maisProblemas = true;
    perguntar = true;

    aux =
TradutorDSC.Grafo.acharNo("LocalizacaoDoProblema");
    EvidenciasColetadas.append("Localização do
Problema\n");
}
else
{
    maisProblemas = false;
    perguntar = false;
}

}while(maisProblemas == true);

JOptionPane.showMessageDialog(null,EvidenciasColetadas);
}
}
```

Anexo G – Principal.java

```
import javax.swing.JOptionPane;

class Principal
{
    public static void main(String args[])
    {
        int opcao;
        String opc;
        opc = JOptionPane.showInputDialog(null,
            "1- BProfundidade; 2- BHeruristica");
        opcao = Integer.parseInt(opc);

        if (opcao == 1)
        {
            TradutorDSC Listagem = new TradutorDSC();
            Listagem.ListarArquivo();

            BProfundidade x = new BProfundidade();
            x.Buscar();
        }
        else
        {
            TradutorDSC Listagem = new TradutorDSC();
            Listagem.ListarArquivo();

            BHeuristica y = new BHeuristica();
            y.Buscar();
        }
        System.exit(0);
    }
}
```