

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA ELÉTRICA**

Cesar Roberto Pamplona Filho

***SHELL* PARA CRIAÇÃO AUTOMÁTICA DE SISTEMAS  
ESPECIALISTAS COM INFERÊNCIA EM PROLOG E  
TRATAMENTO DE INCERTEZA POR FATORES DE CERTEZA**

Florianópolis

2012



Cesar Roberto Pamplona Filho

***SHELL* PARA CRIAÇÃO AUTOMÁTICA DE SISTEMAS  
ESPECIALISTAS COM INFERÊNCIA EM PROLOG E  
TRATAMENTO DE INCERTEZA POR FATORES DE CERTEZA**

Dissertação submetida ao Programa de  
Pós-Graduação em Engenharia Elétrica  
da Universidade Federal de Santa  
Catarina para a obtenção do grau de  
Mestre em Engenharia Elétrica  
Orientador: Prof. Dr. Fernando  
Mendes de Azevedo

Florianópolis

2012

Pamplona Filho, Cesar Roberto

Shell para criação automática de sistemas especialistas com inferências em prolog e tratamento de incerteza por fatores de certeza [dissertação] / Cesar Roberto Pamplona Filho ; orientador, Fernando Mendes de Azevedo - Florianópolis, SC, 2012.

84 p. ; 21cm

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Elétrica.

Inclui referências

1. Engenharia Elétrica. 2. Sistemas Especialistas. 3. Shell. 4. Prolog. I. Azevedo, Fernando Mendes de . II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

Cesar Roberto Pamplona Filho

***SHELL* PARA CRIAÇÃO AUTOMÁTICA DE SISTEMAS  
ESPECIALISTAS COM INFERÊNCIA EM PROLOG E  
TRATAMENTO DE INCERTEZA POR FATORES DE CERTEZA**

Esta Dissertação foi julgada adequada para obtenção do título de Mestre em Engenharia Elétrica, área de concentração em Engenharia Biomédica, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

Florianópolis, 29 de Março de 2012.

---

Prof. Dr. Patrick Kuo Peng  
Coordenador do Curso

**Banca Examinadora:**

---

Prof. Dr. Fernando Mendes de Azevedo  
Orientador

---

Prof. PhD. Jefferson Luiz Brum Marques  
Universidade Federal de Santa Catarina

---

Profa. Dra. Daniela Ota Hisayasu Suzuki  
Universidade Federal de Santa Catarina

---

Profa. Dra. Fernanda Isabel Marques Argoud  
Instituto Federal de Educação, Ciências e Tecnologia de Santa Catarina

## AGRADECIMENTOS

Aos meus pais, Cesar Roberto Pamplona e Maria Letícia Poli Pamplona, e à minha irmã Annelise Poli Pamplona, pelo companheirismo e incentivo durante toda esta minha caminhada.

Aos professores do Instituto de Engenharia Biomédica da UFSC, pelas aulas, debates e ensinamentos permeados de estímulo, que contribuíram não só para o meu amadurecimento acadêmico, mas também para a minha postura de vida.

Ao meu Orientador, Prof. Fernando Mendes de Azevedo, pela segura orientação, conselhos, ideias e sugestões, e também por todo o auxílio na resolução dos problemas durante todo o trabalho.

À Secretaria de Pós-Graduação, ao Instituto de Engenharia Biomédica da UFSC e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela oportunidade concedida de transformar um sonho em realidade.

Aos meus colegas do curso, que estiveram ao meu lado e que me incentivaram quando as dificuldades surgiam e a solução parecia inexistente.

Enfim, a todos que contribuíram de alguma forma para a realização desta pesquisa, expresso minha gratidão pela energia sempre revigorada provinda do calor humano e da solidariedade.



## RESUMO

Este trabalho apresenta o aprimoramento da ferramenta *shell Intellec System* para a criação automática de Sistemas Especialistas, com a utilização de fatores de certeza para tratamento da incerteza. A máquina de inferência foi desenvolvida em Prolog por se tratar de uma linguagem de programação lógica, utilizando a ferramenta *Visual Prolog*. A incerteza sobre o conhecimento do domínio é representada por fatores de certeza. Na *shell* as variáveis podem ser de natureza linguística, numérica e multimídia, como imagens e vídeos. A interface gráfica com o usuário foi remodelada e transcrita utilizando-se a ferramenta Borland C++ Builder para melhor integração com o usuário e a máquina de inferência PROLOG. Testes foram realizados para assegurar que os resultados apresentados pela *shell* fossem compatíveis com aqueles da versão anterior. Foi realizada, também, uma avaliação sobre usabilidade da *shell* por meio de um questionário chamado QUISS, baseado na norma ISO/IEC 9126, através de aplicação a um grupo de alunos, obtendo-se resultados promissores para a interface desenvolvida.

**Palavras-chave:** Sistemas Especialistas, *Shell*, Prolog, Fatores de Certeza, ISO/IEC 9126



## ABSTRACT

This work presents the improvement of *shell* program Intellec System for the automatic creation of expert systems with the use of certainty factors for the treatment of uncertainty. The inference engine was developed in Prolog because it is a logic programming language, Visual Prolog using the program. Uncertainty about the domain knowledge is represented by certainty factors. In the shell variables can be linguistic, numerical and multimedia such as images and videos. The graphical user interface has been redesigned and transcribed using the tool Borland C++ Builder for better integration with the user and the PROLOG inference engine. Tests were performed to ensure that the results presented by this *shell* are compatible with those of the previous version. It was also performed an evaluation of usability of the shell through a questionnaire called QUISS, based on the ISO / IEC 9126, by application to a group of students, yielding good results for interfacing developed for this shell.

**Keywords:** Expert Systems, Shell, Prolog, Certainty Factors, ISO/IEC 9126



## LISTA DE FIGURAS

Figura 1- Fluxograma de preenchimento da base de conhecimento. ....	31
Figura 2- Interface entre os módulos do <i>Intellec System</i> (FERRARI, 2005) .....	33
Figura 3 - Tela principal da base de conhecimento.....	35
Figura 4 - Tela com o nome do especialista e informações sobre o sistema .....	36
Figura 5 - Tela de cadastro de variável .....	37
Figura 6- Visualização do vídeo na consulta pelo player-padrão.....	38
Figura 7- Modelo de regra .....	38
Figura 8 - Tela do cadastro de regras .....	40
Figura 9 - Tela de escolha da variável <i>objetivo</i> .....	41
Figura 10- Tela de lista dos valores <i>verdade</i> .....	42
Figura 11 - Tela de cadastro de perguntas .....	43
Figura 12- Tela das perguntas executando uma consulta.....	44
Figura 13 - Tela das variáveis utilizadas pelo SE. ....	45
Figura 14 - Tela de regras processadas pelo SE.....	46
Figura 15 - Árvore de decisão para visualizar o raciocínio do SE. ....	47
Figura 16 - Gráfico de Satisfação Global.....	55



## **LISTA DE TABELAS**

Tabela 1- Comparativo dos resultados obtidos nos softwares .....	49
Tabela 2- Comparativo dos resultados obtidos nos softwares para o caso 1.....	50
Tabela 3- Resultados dos exames para o caso 1 .....	51
Tabela 4- Comparativo dos resultados obtidos nos softwares para o caso 2.....	51
Tabela 5- Resultados dos exames para o caso 2 .....	51
Tabela 6- Comparativo dos resultados obtidos nos softwares para o caso 3.....	52
Tabela 7- Resultados dos exames para o caso 3 .....	52
Tabela 8 - Comparativo por faixa de notas .....	54



# SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>17</b>
1.1. OBJETIVOS.....	19
1.1.1. OBJETIVO GERAL .....	19
<b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>21</b>
2.1. SISTEMAS ESPECIALISTAS .....	21
2.2. LÓGICA MATEMÁTICA.....	22
2.3. FATORES DE CERTEZA.....	25
2.4. C++ .....	26
2.5. VISUAL PROLOG .....	27
<b>3. METODOLOGIA E IMPLEMENTAÇÃO .....</b>	<b>29</b>
3.1. METODOLOGIA.....	29
3.2. IMPLEMENTAÇÃO .....	34
3.3. MÓDULO EDITOR DA BASE DE CONHECIMENTO .....	35
3.3.2. <i>Informações do Sistema Especialista</i> .....	35
3.3.3. <i>Cadastro de variáveis</i> .....	36
3.3.4. <i>Cadastro de Regras</i> .....	38
3.3.5. <i>Cadastro de objetivos</i> .....	40
3.3.6. <i>Cadastro de Valores Verdade</i> .....	41
3.3.7. <i>Cadastro de perguntas</i> .....	42
3.4. MÓDULO DE EXECUÇÃO .....	43
3.5. MÓDULO DE DEPURAÇÃO.....	44
<b>4. AVALIAÇÃO .....</b>	<b>48</b>
<b>5. CONSIDERAÇÕES FINAIS E CONCLUSÕES .....</b>	<b>57</b>
<b>6. REFERÊNCIAS .....</b>	<b>59</b>
<b>7. APÊNDICE A – QUESTIONÁRIO DE AVALIAÇÃO.....</b>	<b>62</b>
<b>ANEXO A – BASE DE CONHECIMENTO .....</b>	<b>71</b>
<b>ANEXO B – BASE DE CONHECIMENTO : DIAGNÓSTICO PARA AS DOENÇAS ARTRITE REUMATOIDE E LUPOS ERIMATOSO SISTEMICO. – INTELLEC SYSTEM .....</b>	<b>75</b>
<b>ANEXO B – BASE DE CONHECIMENTO : DIAGNÓSTICO PARA AS DOENÇAS ARTRITE REUMATOIDE E LUPOS ERIMATOSO SISTEMICO. – EXPERT SYSTEM .....</b>	<b>79</b>





## 1. INTRODUÇÃO

Na última década, a busca pela resolução de problemas complexos tem sido cada vez mais presente nas empresas, no meio acadêmico e também na vida diária das pessoas. Com isto, torna-se necessário o uso de ferramentas avançadas para tentar resolvê-los, viabilizando, assim, o uso das técnicas de Inteligência Artificial (IA) de forma a suprir algumas dessas necessidades (FERRARI, 2004).

Sabe-se que muitos problemas práticos, em particular nas áreas da saúde, apresentam difícil – ou até mesmo impossível – algoritmização. Neste caso, as técnicas tradicionais de construção de sistemas de apoio à decisão não são aplicáveis. A alternativa é o uso de Técnicas de Inteligência Artificial. Quando for possível criar um conjunto de regras de produção representando o conhecimento de um especialista, usa-se o paradigma simbólico da IA. Porém, quando o especialista não consegue articular em regras o conhecimento que ele possui do problema nem a forma de conseguir resolvê-lo, pode-se usar o paradigma conexionista se esse mesmo especialista for capaz de fornecer um conjunto de exemplos de casos bem solucionados que possam servir como conjunto de treinamento e validação.

Dentro do estudo da IA, no caso do paradigma simbólico, os sistemas especialistas (SE) aparecem como sendo uma das ferramentas mais usadas assim como de maior sucesso. Apesar de hoje existirem SE usando técnicas de outros paradigmas da IA, eles, surgiram inicialmente como resultado de estudos do paradigma simbólico da IA.

Segundo Fernandes (2004), os Sistemas Especialistas solucionam problemas que podem ser resolvidos apenas por pessoas que possuem o conhecimento sobre o domínio específico, ou seja, por especialistas reconhecidos no assunto e que são denominados *Especialistas de Domínio*.

Os SEs podem ser caracterizados como sistemas que reproduzem o conhecimento e a experiência de um especialista que foram adquiridos ao longo dos anos de trabalho e estudo. Para construir um Sistema Especialista, é necessário o auxílio de um especialista humano com conhecimento e experiência no domínio, o qual fornece as informações necessárias. Ou seja, além do Engenheiro de Conhecimento, que é o encarregado da construção e/ou implementação do sistema, existe a necessidade de uma forte interação com um Especialista de Domínio. Os especialistas têm a capacidade de resolver problemas difíceis, explicar os resultados obtidos, aprender, reestruturar o conhecimento e determinar as suas características relevantes; porém muitas vezes esses

mesmos especialistas possuem dificuldade para explicar seu modo de raciocínio de uma maneira analítica (FERNANDES, 2004). Provém daí a necessidade da presença do Engenheiro de Conhecimento que, além da construção / implementação do sistema, deve saber conduzir/direcionar/instruir/ajudar o Especialista de Domínio no sentido de conseguir extrair o conhecimento que este especialista possui sobre o problema em questão.

A motivação para o desenvolvimento desses sistemas foi, inicialmente, a resolução de problemas em uma área restrita do conhecimento humano no intuito de auxiliar na tomada de decisões e também de explicar como o sistema chegou a uma resposta, mostrando os passos de seu raciocínio.

Ao contrário dos sistemas tradicionais, os SEs possuem facilidades que aumentam a sua flexibilidade e eficiência, tais como a possibilidade de construção de novas regras e a tomada lógica de decisões sob imprecisão ou na ausência de informações. Nos sistemas tradicionais, o método de busca é realizado de acordo como o sistema foi codificado anteriormente, ou seja, toda lógica de busca e inferência deve ser criada nesse tipo de sistemas. Surgindo novos conhecimentos, o tradicional deve ser reescrito, enquanto o sistema especialista só adiciona as novas regras ou altera as existentes à base de conhecimento criado (FERNANDES, 2004).

O uso de SE para resolução de problemas é visto como a principal solução dentre nas ferramentas de IA por ser aplicável facilmente a um determinado problema em específico, enquanto outras ferramentas demandam um esforço superior para conseguir resultados similares.

Um dos grandes problemas da concepção e implementação de SE é a necessidade de conhecer as técnicas de IA, bem como de uma linguagem de programação direcionada a este tipo de aplicação, à semelhança do PROLOG. Desta forma, foi sugerida a construção de *shells*, ou seja, de programas que auxiliam pessoas sem grande conhecimento de técnicas de IA e sem nenhum conhecimento das linguagens de programação relacionadas à implementação de tais sistemas. Na realidade, na maioria dessas *shells*, os implementadores têm de se preocupar apenas com a construção da base de conhecimento. Toda a parte de interface homem- máquina, máquina de inferência, definições de tipos prototípicos de variáveis, perguntas padrões, etc, são geradas automaticamente pela *shell*.

O Instituto de Engenharia Biomédica (IEB-UFSC) da Universidade Federal de Santa Catarina (UFSC) tem desenvolvido

trabalhos nesta área há anos. Como resultado de diversos SE desenvolvidos, surgiu a necessidade do desenvolvimento de uma *shell* própria. Dois sistemas, o SSEF e o Intellec System I, foram desenvolvidos inicialmente, os quais deram origem às dissertações de mestrado de ANGELONI, em 2001, e de FERRARI, em 2005.

Estes sistemas, todavia, tiveram como propósito inicial demonstrar a possibilidade de sua implementação por meio do uso de uma máquina de inferência já existente (obtida de versão profissional e licenciada do *Visual Prolog*) para estudar e desenvolver protótipos de interfaces. Todavia, apesar de terem atendido às especificações iniciais, tais sistemas apresentavam alguns problemas, como por exemplo, a falta de interface amigável com o usuário, a impossibilidade de definir o tipo de variáveis (como vídeo) para serem apresentados na consulta e a falta de um gerador de uma árvore de decisão que mostrasse como o SE chegaria ao seu resultado. Ademais, os dois sistemas foram escritos em linguagem Delphi, o que acabou ocasionando algumas restrições durante suas implementações, sendo a de maior importância a de acesso dinâmico a DLL, pois a linguagem Delphi permite somente acesso estático declarado no início da aplicação.

Como consequência surgiu a necessidade de um novo *software* com uma nova linguagem e com novos tipos de variáveis que explorassem melhor a máquina de inferência Prolog, que apresentassem uma interface amigável e que fossem “escritos” numa linguagem de programação que permitisse mais recursos, no caso o C++. Enfim, um *software* que realmente atendesse às necessidades de rápida prototipagem aliada a um grande número de tipos de variáveis e dotado de uma interface homem/máquina que facilitasse o seu uso tornando-o amigável.

## 1.1.OBJETIVOS

Os objetivos deste trabalho são descritos a seguir.

### 1.1.1.Objetivo Geral

Pretendemos desenvolver um *software* definido como uma ferramenta *shell* para criação de Sistemas Especialista utilizando uma máquina de inferência Prolog e tratando a incerteza por Fatores de Certeza.

### 1.1.2.Objetivos Específicos

- Transcrever o sistema da linguagem Delphi para C++;
- Criar vídeos como “valores” de variáveis;
- Aumentar o número de tipos de imagens suportados pelo programa como “valores” de variáveis;
- Desenvolver uma interface mais amigável;
- Gerar árvore de decisão do SE;
- Validar os valores encontrados pelo Sistema Especialista;
- Avaliar a interface do sistema desenvolvido.

Nos capítulos a seguir são apresentadas a fundamentação teórica, a metodologia e respectiva implementação, a avaliação e, finalmente, as discussões e conclusões finais sobre o trabalho.

## 2.FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentada a fundamentação teórica sobre os temas abordados no trabalho.

### 2.1.SISTEMAS ESPECIALISTAS

Após os primeiros anos de estudos sobre a Inteligência Artificial (IA), em que se acreditava ser possível ter um programa capaz de resolver problemas em geral, com uma abrangência mais modesta de domínios, mas com resultados consideráveis, surgiu um campo a ser explorado, que ficou conhecido como Sistemas Especialistas (SEs). (BARRETO,1997).

O marco inicial da IA foi a conferência de verão no Dartmouth College, em 1956, que fica localizado no estado de New Hampshire nos Estados Unidos. Na proposta enviada a esta conferência, que foi submetida ao parecer da Fundação Rockefeller, consta a intenção (por parte dos autores da mesma) de realizar um estudo de dois meses com um grupo de dez pessoas sobre o tópico de inteligência artificial. Esta parece ser a primeira menção oficial à expressão “Inteligência Artificial” (BITTENCURT, 2001).

Na década de 1960 iniciou-se o desenvolvimento dos Sistemas Especialistas, pois se imaginava a possibilidade de, a partir de um pequeno conjunto de regras, de alguns fatos e de regras de inferência, além de um poderoso computador, criar um sistema com capacidade de mimetizar o raciocínio humano na resolução de problemas específicos numa área restrita do conhecimento.

No ano de 1964 foi construído o primeiro Sistema Especialista (SE), chamado de DENDRAL, na Universidade de Stanford. O sistema era capaz de encontrar as estruturas moleculares orgânicas a partir da espectrometria de massa das ligações químicas presentes em uma molécula desconhecida (FERRARI, 2005). Em 1968, o MACSYMA foi criado no Massachusetts Institute of Technology nos EUA, com a finalidade de auxiliar os matemáticos a resolver problemas complexos (FERRARI, 2005). De 1970 em diante, vários Sistemas Especialistas foram criados destacando-se, por sua importância, o MYCIN, o CADUCEUS, o EXPERT, o SOPHIE e o PROSPECTOR (ANGELONI, 2001).

O MYCIN foi criado na Universidade de Standorf, USA, em 1975, com o intuito de auxiliar os médicos a escolher o tratamento baseado em antibióticos para pacientes com bacteremia, meningite e cistite infecciosa em ambientes hospitalares (BITTENCURT, 2001). O PROSPECTOR foi baseado no MYCIN com a finalidade de auxiliar os geólogos na exploração mineral.

Na década de 1980 os desenvolvimentos com Sistemas Especialistas foram muitos, inclusive em aplicações comerciais. Isto se deve aos avanços das tecnologias de *hardware* ocorridas neste período. (HEINZLE; RABUSKE,1995)

No Brasil a Pontifícia Universidade Católica do Rio de Janeiro desenvolveu, em meados dos anos 80, um trabalho chamado SAFO que se destacou como sendo um Sistema Automático para Formalização do Conhecimento, com a finalidade de demonstrar teoremas matemáticos. (MONTELLO,1999)

Hoje o conceito de Sistemas Especialistas foi estendido para outros paradigmas da IA. Surgiram os Sistemas Especialistas baseados em Redes Neurais, os Sistemas Especialistas *Fuzzy*, e os Híbridos. Como consequência, os SEs tradicionais, citados anteriormente, passaram a ser conhecidos como *Sistemas Especialistas Baseados em Regras*. Neste trabalho utilizaremos apenas os SEs baseados em regras e, por consequência, vamos simplesmente nos referir a eles como Sistemas Especialistas.

## 2.2.LÓGICA MATEMÁTICA

A Prolog (PROgraming LOGic), como uma linguagem declarativa, oferece muitas características que facilitam a implementação de programas IA, principalmente da área de Sistemas Especialistas. Foi desenvolvida na França, por volta de 1970, por ALAIN COLMEAUER e seus colegas da Universidade de Marseille, só tendo, porém, se tornado conhecida após os japoneses haverem lançado seu programa de quinta geração de computadores.

O mecanismo de inferência básico do Prolog é a implementação do Teorema da Resolução estabelecido por ROBINSON em 1965 e que KOWALSKI provou ser utilizável na solução de uma grande classe de problemas. (FERRARI, 2005)

*Resolução* é o processo de eliminação definido pela inferência chamada *fórmula de resolução*

$$((\sim A \vee B) \wedge A) \rightarrow B \quad (1)$$

cujos membros esquerdo, pela distribuição de  $\wedge$ , resulta  $(\sim A \wedge A) \vee (B \wedge A)$  que, em virtude da falsidade da contradição  $(\sim A \wedge A)$  resulta em  $(B \wedge A)$ . Isto implica que B é verdade.

São também importantes as expressões de equivalência:

$$\sim(A \rightarrow B) \Leftrightarrow (\sim A \vee B) \Leftrightarrow A \wedge \sim B \quad (2)$$

Suponha a proposição  $A \rightarrow B$ , onde A é um conjunto de premissas, e onde o Prolog tenta demonstrar sua negação  $\sim(A \rightarrow B)$ , que é equivalente a  $A \wedge \sim B$ . Isto implica provar que B é falso, presumindo-se que A seja verdade (isto é para provar que  $\sim B$  existe no conjunto A). Se ocorre falha na prova, supõe-se que proposição é válida. Para realizar esta prova, o Prolog adiciona ao conjunto A a proposição  $\sim B$ ; e pela resolução e união, procura uma declaração vazia. Estas validações são obtidas somente em dois casos: primeiro, se B não é válido; segundo, se o conjunto A de regras não é completo. Desta maneira, todo o conjunto de regras e toda hipótese B resulta em uma solução. Se a solução é “B é verdade”, então B é verdade. Por outro lado, se a solução é “B é falso”, pode significar que somente o conjunto de regras não é suficiente para provar que B é verdade. (ALGARVE, 1995)

Mais ainda, Prolog utiliza a unificação para tratar de problemas com variáveis (cálculos dos predicados). Unificação é o processo de tentar tornar predicados iguais pela substituição de variáveis por argumentos e transportar estas substituições para a regra como um todo. Desta maneira, o cálculo dos predicados é transformado em cálculo proposicional ao qual se aplica a resolução. (ARARIBÓIA, 1988)

Nesta *shell* estão sendo implementadas três tipos de regras para computação da verdade (KOSKO, 1992). São elas: a regra Max-min, a regra do produto (probabilística) e a regra mista. Por exemplo, na regra  $(A \wedge B \wedge C) \rightarrow D$ , a verdade de D é igual à

- $\mu(D) = \min [\mu(\text{Regra}), \mu(A), \mu(B), \mu(C)]$  para regra min-max.
- $\mu(D) = \mu(\text{Regra}) \cdot \mu(A) \cdot \mu(B) \cdot \mu(C)$  para regra do produto.

- $\mu(D) = \mu(\text{Regra}) \cdot \min[\mu(A), \mu(B), \mu(C)]$  para regra mista.

Com a fórmula mista e a fórmula do produto, a verdade da conclusão sempre varia quando a verdade da regra varia. Por outro lado, a verdade da composição das premissas nunca é maior do que a verdade da menor das verdades dos componentes. A regra min-max, mais consistente, pode resultar na mesma verdade para a conclusão, mesmo quando se utilizam regras de diferentes verdades, apesar da diferença entre os mínimos das premissas. Em outras palavras, esta fórmula não é sensível a variações sobre o mínimo das premissas desde que a verdade da regra seja maior do que o mínimo das verdades das premissas.

A *shell* salva o resultado e gera uma falha após cada conclusão alcançada. A máquina de inferência do Prolog então inicia a pesquisa por outra solução. A falha se repete até que se alcance um resultado. Então, quando Prolog testa todos os caminhos possíveis para uma conclusão (intermediária ou final), sua verdade é computada pela

fórmula

$$\mu(A1 \vee A2 \vee \dots \vee An) = 1 - \prod_{k=1}^n [1 - \mu(A_k)] \quad (3)$$

para a regra dos produtos e mista,

$$\mu(A1 \vee A2 \vee \dots \vee An) = 1 - \min(1 - \mu(A_k)), k = 1, \dots, n \quad (4)$$

para regra min-max, onde  $\mu(A_k)$  são os valores de verdade de  $A_k$ , realizadas em duas formas diferentes. Estes cálculos e a pesquisa de todas as soluções implementam um “or” *fuzzy*. A fórmula(3) acima, expressa à aplicação do “and” *fuzzy*, junto com o complemento *fuzzy* definido por

$$\mu(\sim A) = 1 - \mu(A) \quad (5)$$

Estes cálculos mostram que todos os conectivos são implementados como operadores *fuzzy*. Após a realização destes cálculos, o programa pesquisa o resultado com maior valor de verdade (máximo) e o considera como a solução final. Ele também apresenta os próximos maiores valores de verdade.



### 2.3.FATORES DE CERTEZA

Um dos grandes problemas na implementação de sistemas especialistas, dentre outros, é o tratamento da incerteza e da imprecisão. A incerteza ocorre quando não se pode afirmar que uma proposição é verdadeira ou falsa; e a imprecisão pode ser definida quando o valor de uma proposição não for suficientemente determinado em relação a determinada escala. (PRADE, 1985)

Várias abordagens surgiram para tratar a incerteza e/ou a imprecisão. Uma delas propõe a utilização da Teoria da Probabilidade. Sistemas que utilizam essa abordagem também são conhecidos por *Sistemas Especialistas Probabilísticos*. O problema aqui é a determinação das probabilidades associadas, e o não esquecimento de que a soma de todas as probabilidades é sempre unitária.

Outra abordagem é o uso de Sistemas *Fuzzy*, o que originou os Sistemas Especialistas *Fuzzy*. A maior dificuldade, neste caso, é o processo de “fuzzificação” e o de “desfuzzificação”, este quando necessário. Existem outras abordagens, como a Teoria da Crença e os Fatores de Certeza. Neste trabalho foi utilizada a última para o tratamento da incerteza.

Os Fatores de Certeza foram propostos no MYCIN, Sistema Especialista que surgiu em 1976, conforme já mencionado, como modelo de tratamento de incerteza. A definição do fator de certeza(FC) é realizada em função de uma medida definida como *medida de crença* (MC) e em função de outra medida, conhecida como *medida de descrença* (MD), em uma hipótese H, para uma evidência E, conforme equação 2.2.2

$$FC(H,E) = MC(H,E) - MD(H,E) \quad (6)$$

onde

FC(H,E) é o fator de certeza na hipótese H para uma mesma evidência E;

MC(H,E) é a medida da crença em H, dado E. Este termo representa o recurso que mede até que ponto a evidência sustenta a hipótese;

MD(H,E) é a medida de descrença em H, dado E. Este termo representa o recurso que mede até que ponto a evidência sustenta a negação da hipótese;

As equações abaixo 7 e 8 definem as medidas de crença e descrença. (RICH E ROBINSON,1988).

$$MC(H, E) = \left\{ \begin{array}{l} 1 \text{ se } P(H) = 1 \\ \frac{\max \left[ P\left(\frac{H}{E}\right), P(H) \right] - P(H)}{\max[1, 0] - P(H)} \end{array} \right\} \text{ em outros casos} \quad (7)$$

$$MD(H, E) = \left\{ \begin{array}{l} 1 \text{ se } P(H) = 0 \\ \frac{\min \left[ P\left(\frac{H}{E}\right), P(H) \right] - P(H)}{\min[1, 0] - P(H)} \end{array} \right\} \text{ em outros casos} \quad (8)$$

onde  $P(H)$  é a probabilidade de ocorrência da hipótese  $H$ ;  
 $P(H/E)$  é a probabilidade condicional da ocorrência da hipótese  $H$ , dada a ocorrência da evidência  $E$ . (ANGELONI, 2001)

Como pode ser observado, o cálculo dos fatores de certeza também envolve probabilidades, o que nem sempre é fácil de computar. Todavia, existe uma abordagem para a determinação das probabilidades envolvidas que se baseia em uma das ideias subjacentes da Teoria da Crença: como os sistemas, de forma geral, são robustos, ao invés de se calcular as probabilidades para o subsequente cálculo das medidas de crença e de descrença para, então se obter o valor do fator de certeza, faz-se uma estimativa desses valores. Em outras palavras, é solicitado ao Especialista de Domínio responder baseando-se em seus vastos anos de experiência e nos muitos casos estudados qual seria sua estimativa, sua crença, de valores para essas variáveis. Claro, um erro estará associado. Todavia, na validação do sistema, os valores podem ser facilmente modificados de forma a serem minimizados.

#### 2.4.C++

A linguagem para desenvolvimento da interface com o usuário foi a C++, e como ambiente de desenvolvimento foi escolhida o Borland Builder C++ 6.0. Tal escolha se deu pelo fato de ser necessário transcreever o *Intellec System* para uma linguagem mais conhecida, a fim

de facilitar a manutenção do programa e também a sua continuidade em trabalhos futuros. Outro ponto levado em consideração na escolha pelo C++ foi a possibilidade de se fazer a integração com DLLs desenvolvidas em outras linguagens de programação.

O uso de componentes disponíveis no ambiente de desenvolvimento facilitou a criação das interfaces, podendo, assim, tornar seu uso pelos usuários mais intuitivo e também melhorar a manutenção do programa.

A utilização de componentes externos também facilita e ajuda o desenvolvimento, como a utilização do componente *ATImagebox* para visualização de diversos tipos de imagens. Com este componente é possível visualizar uma imagem definindo seu tamanho na tela, aplicar zoom, centralizar a imagem, aplicar uma escala de tamanho, alterar o seu tamanho e reexibir. Os tipos de imagens suportadas são BMP, JPG, ICO, GIF, PNG, WMF, TGA, etc.

A possibilidade de visualizar vídeos é outra vantagem da linguagem, mas como os vídeos geralmente são gerados de forma compactada, existe a necessidade de se conhecer os *codecs*, que são bibliotecas que descompactam os vídeos para reproduzi-los, tornando difícil a tarefa de deixá-lo disponíveis na aplicação. Torna-se inviável o seu uso por ser impraticável a pré-disponibilização destes *codecs* no sistema. Para resolver este problema, o *software* utiliza o player de vídeo padrão do sistema operacional, que já deve dispor do *codec* dos vídeos instalados e prontos para sua visualização.

## 2.5.VISUAL PROLOG

O PROLOG é projetado para lidar com problemas lógicos nos quais as decisões precisam ser feitas de forma ordenada, no sentido de fazer da "razão" do computador o seu caminho para uma solução. Ele é particularmente bem adaptado aos vários tipos de problemas na IA. (RICH; ROBINSON, 1988)

O compilador Visual Prolog é um sucessor do compilador Turbo Prolog criado em 1980, que foi o primeiro compilador Prolog. Desde então, o Prolog Development Center tem desenvolvido e melhorado esse compilador. (PROLOG DEVELOPER CENTER,1996)

Atualmente o Visual Prolog é um compilador que tem entre seus atributos:

- Criação de programas executáveis com interface ou DLLs;
- Resolução das referências cruzadas entre as declarações;
- Validação de dados de inicialização em construtores;
- Execução de predicado *resolução* (Executa a variável definida como meta do problema);
- Criação de janelas para interação com o usuário.

O Visual Prolog, também conhecido anteriormente como *PDC Prolog* e *Turbo Prolog*, é uma extensão orientada a objeto do Prolog. Nele podem-se construir aplicações gráficas, aplicativos de console e DLLs. Pode também ligar-se a componentes COM e às bases de dados através de ODBC.

O Visual Prolog é compilado e não interpretado, assim como as linguagens procedimentais tradicionais C, e isso proporciona a melhoria importante da conversão e velocidade de execução em relação ao Prolog tradicional, garantindo maior robustez nos aplicativos desenvolvidos.

### 3.METODOLOGIA E IMPLEMENTAÇÃO

Neste capítulo é descrita a metodologia utilizada para a especificação e implementação do sistema computacional proposto.

#### 3.1.METODOLOGIA

O Intellec System, em sua primeira versão, foi desenvolvido prevenindo três situações a serem realizadas pela *shell* no auxílio ao usuário não especialista em técnicas de inteligência artificial para construção de um sistema especialista. São elas: a criação de uma base de conhecimento, a execução do sistema especialista usando uma das bases de conhecimento criadas; e a depuração do sistema através da visualização pelo usuário de como o sistema especialista chegou a uma conclusão a partir da questão colocada e das inferências feitas pela máquina de inferência na base de conhecimentos.

O **módulo de criação** tem por finalidade permitir ao usuário criar bases de conhecimento, bem como editar, incluir e remover informações da referida base de conhecimento que compõe o Sistema Especialista. Na realidade, a *shell* permite criar diferentes bases de conhecimento. Ou seja, no início da execução da *shell*, a primeira atividade é a escolha da base de conhecimento a ser carregada no sistema computacional. Esta informação é importante, pois mostra uma das grandes potencialidades de sistema (o antigo e o novo), que é a possibilidade de implementar diferentes Sistemas Especialistas pela simples mudança da base de conhecimento ativa.

A base de conhecimento deve ser criada seguindo-se uma ordem de criação dos elementos que a compõem. O primeiro elemento é o nome do criador do sistema especialista e uma descrição sobre problema tratado pela referida base de conhecimento. O segundo elemento consiste na definição das variáveis, seus tipos e a definição de seus valores considerando-se que podem ser de diversos tipos. A seguir, é necessário definir quais as variáveis que serão os objetivos do sistema, meta essa que se constitui no terceiro elemento. Os objetivos ou metas podem variar em número, desde a unidade até a pluricidade, sendo este último no caso de Sistemas Especialistas mais complexos. No quarto elemento criamos as regras do sistema contendo as premissas e a sua

conclusão. Estas seriam as regras de produção, de forma a poderem ser tratadas por uma máquina de inferência do tipo PROLOG que utiliza uma única regra de inferência – a Resolução – e, para tal necessita tratar todas as variáveis, dependendo da questão colocada, pelo processo de unificação. No quinto elemento definimos o valor verdade. E por fim, definimos as perguntas a serem feitas aos usuários pelo sistema. Na realidade o sistema assume uma pergunta-padrão levando em consideração a regra associada. Todavia, de forma a facilitar a utilização do sistema por não especialistas e usuários sem grande familiaridade com o assunto tratado, o sistema permite que sejam, para cada regra, definidas perguntas personalizadas. Ademais, permite, ainda, que sejam criadas explicações do porquê da necessidade dessa pergunta. Este último mecanismo é de extrema importância, pois ajuda o usuário final a entender por que tais perguntas estão sendo feitas, contribuindo, com isso, adicionalmente, para ajudá-lo a entender o encadeamento seguido pelo sistema na realização de uma consulta e na construção de uma solução.

Para demonstrar o fluxo de criação das informações citadas acima, a Figura 1 abaixo apresenta os passos a serem seguidos para criação de uma base de conhecimento íntegra, em que a máquina de inferência irá interpretá-la corretamente.

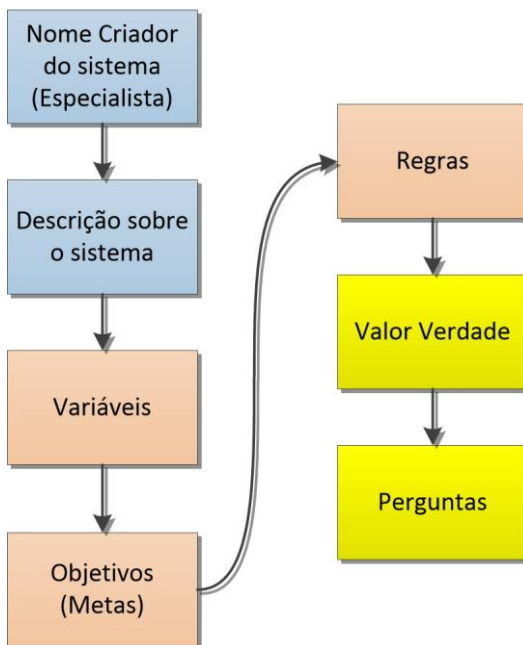


Figura 1- Fluxograma de preenchimento da base de conhecimento.

O **módulo de execução** é o módulo de consulta do sistema. Através das questões colocadas pelo sistema e as respostas dadas pelo usuário, o sistema gera uma ou várias soluções possíveis, cada uma acompanhada de seu fator de certeza, para o problema questionado. Ou seja, nessa consulta as informações são adquiridas por meio de perguntas feitas ao usuário para que o sistema possa inferir uma resposta.

No **módulo de depuração**, o usuário tem acesso, após o término da consulta, à base de conhecimento, às variáveis com seus respectivos valores, de acordo com as respostas dadas pelo usuário. Isto é feito através do acesso ao menu Depuração -> Histórico, variáveis da consulta. Pode-se também conferir as regras que foram aceitas pela máquina de inferência acessando-se o menu Depuração -> Histórico regras da consulta. Uma funcionalidade a mais consiste na possibilidade de se visualizar a árvore de decisão baseada na ordem em que as regras foram processadas e em que as perguntas foram realizadas. Esta pode ser acessada pelo menu Depuração -> Árvore de decisão.

A estrutura do Intellec System II foi desenvolvida utilizando-se um programa principal, que é o executável, e mais 2 DLLs. O programa principal contempla a criação e a edição das bases de conhecimento e da execução da consulta dos Sistemas Especialistas e acesso às funcionalidades de depuração.

A DLL, chamada de *MaquinaInferencia.dll*, contém a máquina de inferência responsável pelo processamento da base de conhecimento. A máquina de inferência foi desenvolvida em *Visual Prolog 5.2*. Foi gerada uma DLL para possibilitar o seu uso por outras linguagens de programação e facilidade de uso da linguagem Prolog.

A outra DLL, *ObterRespostas.dll*, apresenta as perguntas criadas pelo especialista ao usuário e retorna as suas respostas para a máquina de inferência. Esta foi desenvolvida no *Builder C++* para facilitar o desenvolvimento das telas de interface com o usuário.

O Intellec System II foi desenvolvido em outra linguagem de programação, C++, diferente da anterior, que utilizou a linguagem *Delphi*, disponível na ferramenta *Borland Delphi*. Esta mudança facilitou a comunicação com a DLL (máquina de inferência), desenvolvida em *Prolog*, com o ambiente de desenvolvimento da *shell*, possibilitando seu uso em qualquer parte da *shell*, instanciando dinamicamente em memória esta DLL. Outro motivo seria facilitar a continuidade do desenvolvimento da *shell* em trabalhos futuros, utilizando C++, sendo uma linguagem amplamente estudada, alto e baixo nível de programação, e também por maior número de desenvolvedores que a utilizam.

Os três módulos principais do sistema e suas inter-relações são mostrados na Figura 2 abaixo. Adicionalmente, a figura expõe como uma consulta é realizada utilizando-se as DLLs descritas acima. Na execução de um Sistema Especialista, todos os dados que estão presentes na base de conhecimento são transferidos para uma instância em memória da DLL da máquina de inferência. Para o usuário informar ao Sistema Especialista os fatos que ele ainda não possui, a própria instância em memória da máquina de inferência cria a instância em memória da DLL em que o usuário responde as questões e, após a confirmação da resposta, essa instância envia o valor da resposta à máquina de inferência e encerra a instância dessa DLL.



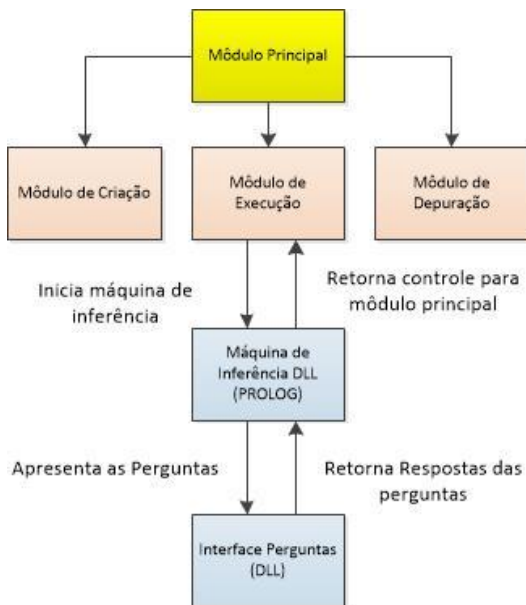


Figura 2- Interface entre os módulos do *Intellec System* (FERRARI, 2005)

Os elementos utilizados na construção e utilização dos SES no Intellec System são:

**Variáveis** – É uma estrutura utilizada para agrupar todos os valores possíveis de um argumento a ser passado para a máquina de inferência. Em sua definição escolhemos qual seu tipo, sua cardinalidade e sua lista de valores possíveis. São elementos de extrema importância no desenvolvimento da base de conhecimento, pois são utilizados pelas regras, perguntas e objetivos.

**Perguntas** - As perguntas servem para informar como os usuários interagem com o SE, revelando quais valores serão processados pelo SE para ser informada uma resposta do SE referente.

**Objetivos** - São definidos por variáveis já cadastradas na base de conhecimento e que são utilizadas não só para definir quais metas o SE deve alcançar como também para apresentar os resultados.

**Valores verdade** - Neste cadastro são criadas diversas faixas de valores que representam o nível de certeza do usuário sobre determinada pergunta.

**Informações** - As informações do SE são importantes para identificar qual especialista desenvolveu o sistema e também para informar qual o objetivo que o mesmo deve alcançar.

**Regras** – Quando um fato depende de outros fatos, podemos definir essas relações através de regras. As regras têm o formato *SE ... ENTÃO*, ou seja, são Regras de Produção a serem tratadas pela máquina de inferência. As regras são de extrema importância para um SE, pois nelas ficam armazenadas todas as informações sobre tomadas de decisão. Na realidade, num Sistema Especialista baseado em regras, o conhecimento do assunto está contido nessas regras. A implicação primeira é que o sistema terá melhor ou pior desempenho em função da capacidade do especialista de dominar a articulação de regras, de seu conhecimento do assunto e de sua conduta na realização de uma consulta (resolução de um problema). Ou seja, essas regras, além de representarem realmente as relações existentes entre as variáveis e as metas, devem, em princípio, apresentar completude e corretude. *Corretude* no sentido de realmente representarem de forma adequada as relações existentes entre as variáveis e as metas e não apresentarem conflitos ou contradições. *Completude* no sentido de realmente representarem o mais completamente possível essas relações. Ou seja, o espaço de estados do problema deve estar devidamente abrangido pelas regras da base de conhecimento.

### 3.2.IMPLEMENTAÇÃO

Neste item será mostrada a implementação da *shell Intellec System II* tanto em relação aos seus módulos constituintes como às funcionalidades acrescentadas e à parte funcional e operacional do sistema.

Como citado anteriormente, esta implementação foi baseada em versão anterior da *shell*. Ou seja, após diversos trabalhos usando a versão anterior, foi possível fazer uma análise crítica no sentido de se observar quais os pontos fortes e fracos da referida versão. Um dos pontos fracos foi a interface usuário-máquina. Outra foi a falta de alguns tipos de variáveis e de outros tipos de valores para variáveis. O sistema, então,

possui três partes principais: edição de bases de conhecimento, consulta das bases criadas e depuração das consultas realizadas.

Neste trabalho foi mantida a estrutura principal do sistema versão anterior (FERRARI, 2005) e implementadas as modificações necessárias conforme exposto no parágrafo anterior, além da mudança da linguagem de programação e da inclusão de outras funcionalidades.

### 3.3.MÓDULO EDITOR DA BASE DE CONHECIMENTO

Neste módulo, o sistema permite ao usuário criar a base com todas as informações, como variáveis, regras, perguntas, valores *verdade*, objetivos e informações sobre o SE, que são necessárias para criação e execução da base de conhecimento que compõe o Sistema Especialista.

Para iniciar a construção de um SE, devemos criar uma base de conhecimento acessando o menu Arquivo -> Novo. Então será mostrada uma tela com os elementos a serem criados e preenchidos, como mostrado na Figura 3 abaixo.

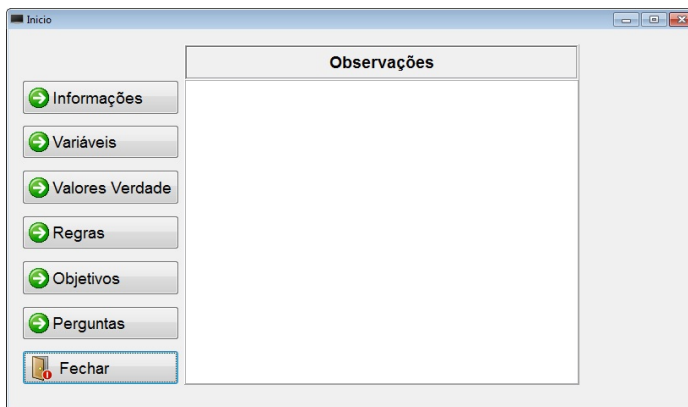


Figura 3 - Tela principal da base de conhecimento.

#### 3.3.2.Informações do Sistema Especialista

De forma a construir a base de conhecimento e prover todas as informações necessárias à construção do Sistema Especialista, primeiramente o campo *Nome do Especialista* deve ser preenchido. Em

seguida, o campo descrição do sistema deve ser preenchido com a descrição sobre o SE e qual o seu objetivo, conforme a Figura 4 abaixo:

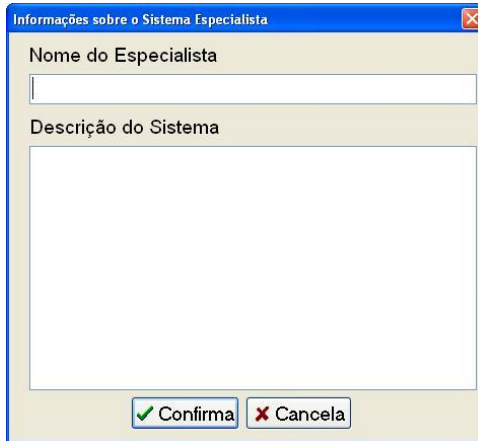
A imagem mostra uma janela de diálogo com o título "Informações sobre o Sistema Especialista". A janela possui um campo de texto para "Nome do Especialista" e um campo de texto maior para "Descrição do Sistema". Na base da janela, há dois botões: "Confirma" com um ícone de checkmark verde e "Cancela" com um ícone de X vermelho.

Figura 4 - Tela com o nome do especialista e informações sobre o sistema

### 3.3.3.Cadastro de variáveis

No cadastro da variável, primeiramente um nome deve ser escolhido para que a mesma possa ser encontrada pelos outros elementos da base de conhecimento, como regras e perguntas. Após, deve ser definido o número de respostas que podem ser atribuídas à variável. Podemos defini-la como:

- Univalorada – Apenas um valor pode ser escolhido quando a variável receber um valor.
- Multivalorada – Um ou mais valores podem ser selecionados para a variável.

A seguir, devem ser definidos também quais os valores que a variável pode assumir, podendo ser atribuído um ou mais valores de acordo com o tipo de variável escolhida, como mostra a Figura 5 abaixo.

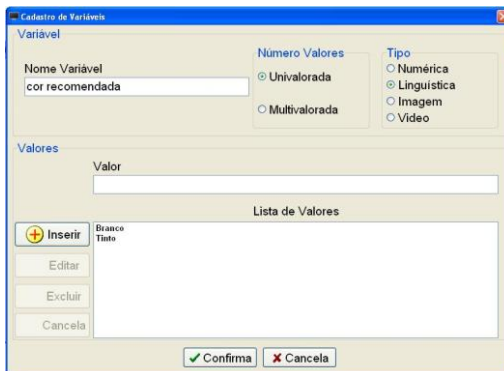


Figura 5 - Tela de cadastro de variável

Após, há outra opção a ser definida: é a qual tipo ela pertence e como o sistema deve tratá-la quando for utilizada. Os possíveis tipos estão descritos a seguir:

- Numérica – Um número pode ser atribuído como valor.
- Linguística – Os valores são definidos por palavras; ou seja, variáveis linguísticas possuem termos linguísticos como valores.
- Imagem – São escolhidas imagens que são atribuídas como valor.
- Vídeo – Arquivos contendo vídeos são definidos e atribuídos como valor.

As variáveis do tipo imagem e vídeo, ao invés de terem valores com termos linguísticos ou numéricos, constituem valores definidos por arquivos-multimídia que podem representar de uma maneira mais completa o assunto que se deseja abordar. As extensões de imagens suportadas são: BMP, JPG, ICO, GIF, PNG, WMF, TGA, etc. Já nos vídeos, as extensões suportadas são as principais que o computador do usuário suporta, como MPEG e AVI, dentre outras. A visualização das imagens e dos vídeos no Intellec System II é realizada pelos visualizadores-padrão do sistema operacional Windows, devendo conter os mesmos *codecs* para que seja possível reproduzir os vídeos. Abaixo, na Figura 6, é mostrada uma tela da consulta em que se pode observar a aplicação da variável do tipo vídeo. Para iniciar o vídeo, basta clicar em *Play Video*.

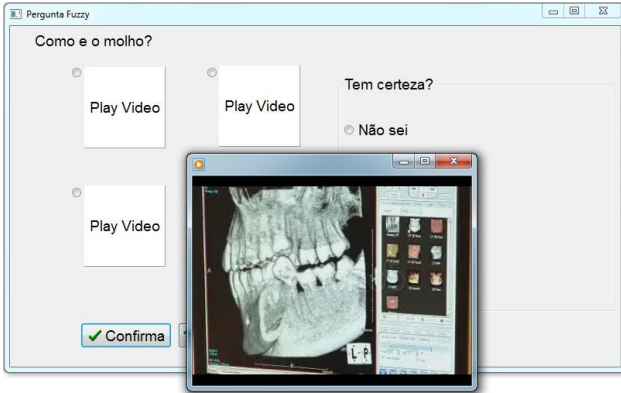


Figura 6- Visualização do vídeo na consulta pelo player-padrão.

É importante citar que esta variável do tipo vídeo e os novos tipos de imagens são funcionalidades novas que não existiam na versão anterior, e que surgiram como necessidade durante o trabalho realizado com a referida versão anterior.

Quando a variável é definida como numérica, o usuário não define uma lista, pois estes valores são passados ao SE em tempo de execução na consulta.

### 3.3.4. Cadastro de Regras

Para iniciar o cadastro de uma regra, o usuário deve selecionar a opção “regras”, onde se apresenta uma janela para escolher um modelo de regra. Nele o usuário pode escolher uma regra já inclusa para apenas alterar alguns itens, ou pode escolher “nenhum” para aparecer a tela de cadastro de regra em branco. A Figura 7 abaixo mostra a janela de modelo de regras.

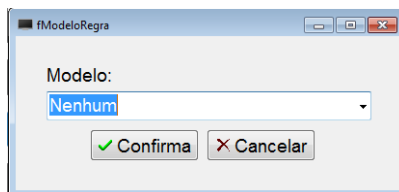


Figura 7- Modelo de regra

A primeira parte do cadastro de regra começa definindo as premissas que fazem parte da definição da regra. Para cadastrar uma premissa, o usuário deve preencher primeiramente a variável; em seguida, o tipo da associação do valor, que seria o sinal de atribuição, como igual ou diferente; e por fim, o valor da variável. Nessas premissas, o usuário escolhe a variável, sendo que todas já devem ter sido cadastradas.

A segunda parte do cadastro de regras é a inserção da conclusão da regra, que é inserida da mesma maneira que as premissas, mas logo abaixo.

Por último, o valor *verdade* deve ser definido em cada regra para ser usado como peso da regra perante o SE desenvolvido. O valor de verdade consiste num Fator de Certeza determinado, conforme discutido na seção de fundamentação teórica, pela certeza que um especialista de domínio tem sobre a situação em tela em função de seu conhecimento e experiência no assunto.

Abaixo, na Figura 8, é apresentada a tela de cadastro de regras com os campos a serem preenchidos pelo usuário.

Regra

Regra 2

Premissas

cor recomendada = Branco  
docura recomendada = Seco

Conclusão

Vinho=Chablis

Valor Verdade

0.9  
Exemplo : 0.5

Confirma Cancelar

Figura 8 - Tela do cadastro de regras

### 3.3.5. Cadastro de objetivos

Aqui escolhem-se os objetivos que o SE deve encontrar quando a consulta for realizada.

Para selecionar um ou mais objetivos, basta selecionar a variável na lista da esquerda e apertar no botão seta “>” para selecioná-la como objetivo, como mostra a tela abaixo na Figura 9.



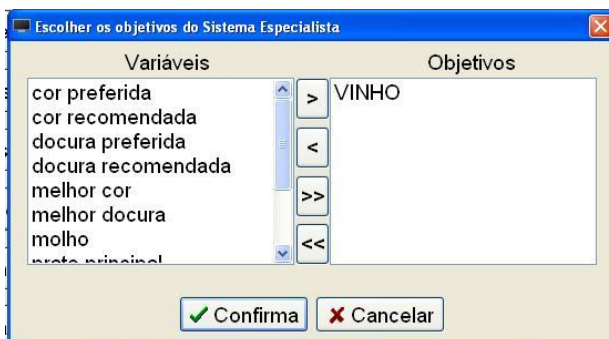


Figura 9 - Tela de escolha da variável *objetivo*

É importante frisar que o sistema, quando em operação de execução, procura primeiro provar o primeiro objetivo. No caso de existência de outros objetivos ou metas, o sistema continuará a consulta tentando provar o segundo objetivo, e assim sucessivamente. Ou seja, a ordem de definição dos objetivos é importante.

### 3.3.6. Cadastro de Valores *Verdade*

O nível de certeza sobre uma resposta de uma pergunta realizada ao usuário é necessário para o sistema saber quão precisa é aquela informação que o usuário está passando. Assim, o peso de resposta influencia no resultado final do usuário.

Para cadastrar, basta apertar o botão *incluir* e aparecerá a janela de cadastro dos valores *verdade*. Primeiramente deve ser definido um nome ao valor *verdade*; segundo, deve ser preenchido o campo com a pergunta que é apresentada ao usuário sobre a certeza de uma resposta. Depois devem ser inseridos os valores possíveis e o nome do termo, e, a seguir, um valor entre 0 e 1, após o quê deve-se apertar o botão *incluir*. Para editar selecione um dos termos e confirme a operação. Para excluir selecione o termo e aperte *excluir*. A Figura 10 abaixo mostra a tela de valores *verdade*.

Figura 10- Tela de lista dos valores *verdade*

O cálculo para a certeza final é dado pela fórmula a seguir:

$$CT = CT1 + CT2 - CT1 * CT2 = CT1 + (1 - CT1) * CT2 \quad (9)$$

onde,  $CT \leq 1$ ,  $CT1 \leq 1$  e  $CT2 \leq 1$ .

Pode-se observar, na fórmula, que o primeiro fator de certeza tem um peso muito grande na definição do valor final. Ou seja, durante a construção de uma base de conhecimento, deve-se ter cuidado na escolha da primeira variável. Se ela for das variáveis que podem ter altíssimos valores de fator de certeza, ela praticamente predominará sobre os fatores de certeza das outras variáveis, definindo (ou quase) o valor final.

### 3.3.7. Cadastro de perguntas

Primeiramente, deve ser descrita a pergunta que o SE apresentará ao usuário quando o mesmo estiver realizando uma consulta ao SE. A seguir, é preciso selecionar uma variável já cadastrada no SE para que os valores respondidos nas perguntas sejam armazenados nesta variável.

Se na pergunta for necessário utilizar um valor *verdade*, esta pergunta deve ser definida como uma pergunta *fuzzy*. Caso contrário, é definida como uma pergunta *crisp*. Pode-se ainda definir uma explicação sobre a pergunta ou sobre as possíveis respostas, auxiliando o usuário a responder a pergunta.

É possível também incluir um tutorial, sendo um arquivo como DOC, PDF ou HTML que contenha informações explicativas sobre a pergunta em questão. Esse texto, como já mencionado, é importante para que o usuário final possa acompanhar o encadeamento de inferências do sistema de forma a ter uma visão mais clara da lógica subjacente e aceitar ou não, com mais facilidade, a conclusão da consulta.

Na Figura 11 abaixo consta, como ilustração, uma tela do cadastro de perguntas onde são definidas as formas como estas serão apresentadas ao usuário.

A imagem mostra uma janela de software com o título "Pergunta". O formulário contém os seguintes elementos:

- Um campo de texto rotulado "Pergunta" com o conteúdo "Este prato tem vitela?".
- Um menu suspenso rotulado "Variável:" com o valor selecionado "tem vitela".
- Um menu suspenso rotulado "Valor Verdade" que está atualmente vazio.
- Um grupo de botões de opção rotulado "Tratamento de Incerteza", com "Crisp" selecionado e "Fuzzy" desselecionado.
- Um campo de texto rotulado "Explicação" contendo o texto: "Para recomendar uma cor e uma docura, preciso saber de que consta seu prato de carne."
- Um campo de texto rotulado "Tutorial" que está vazio, com um ícone de reticências (...) à direita.
- Dois botões na base: "Confirma" com um ícone de marca de seleção verde e "Cancela" com um ícone de uma cruz vermelha.

Figura 11 - Tela de cadastro de perguntas

### 3.4.MÓDULO DE EXECUÇÃO

Para iniciar este módulo, o usuário deve acessar o menu [Consulta]->[Iniciar consulta da base de conhecimento], e escolher qual o arquivo correspondente à base de conhecimento que deseja executar. Caso alguma base já esteja aberta no sistema, o usuário pode iniciar a consulta pelo menu, como já informado, ou pela pressão da tecla de atalho F9, que iniciará automaticamente a execução da consulta da base aberta. Iniciando-se a consulta, é apresentada ao usuário uma série de perguntas definidas na base, e, conforme as respostas indicadas pelo usuário, o sistema, ao final da consulta, apresentará um quadro de respostas com um percentual do fator certeza sobre as respostas encontradas.

Como mostrado na Figura 12 abaixo, é apresentada ao usuário a pergunta assim como suas opções de resposta; e se for uma pergunta que utiliza fator de certeza, o sistema apresenta quais são suas opções para determinar qual a crença do usuário sobre a sua resposta.

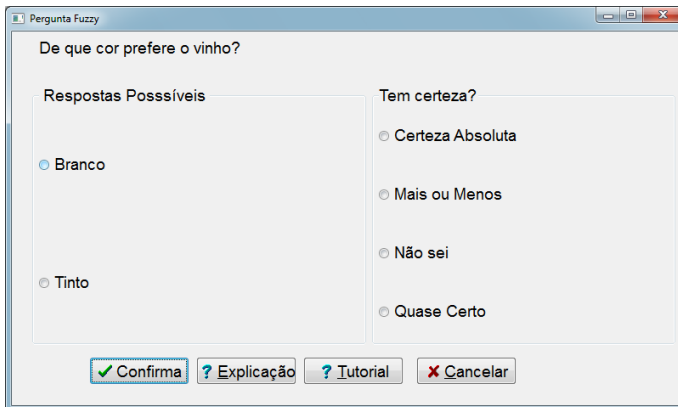


Figura 12- Tela das perguntas executando uma consulta.

### 3.5.Módulo de depuração

Neste módulo, o usuário tem a possibilidade de acessar as informações que foram processadas pela máquina de inferência, ou seja, seguir o encadeamento lógico do sistema que o fez chegar aos resultados finais. Esta funcionalidade é de extrema importância, pois pode servir para dar credibilidade ao sistema de forma “vis-à-vis” em relação ao usuário final. Ou seja, a existência desta finalidade permite que o usuário final

tenha várias possibilidades de entender o porquê da resposta do sistema bem como do fator de certeza associado.

O histórico de variáveis mostra os valores e os pesos que foram introduzidos nas perguntas realizadas ao usuário, e também as variáveis que a própria máquina de inferência utilizou para chegar às respostas. Na Figura 13 abaixo é mostrado um exemplo da tela que é gerada quando a opção de histórico de variáveis é selecionada.

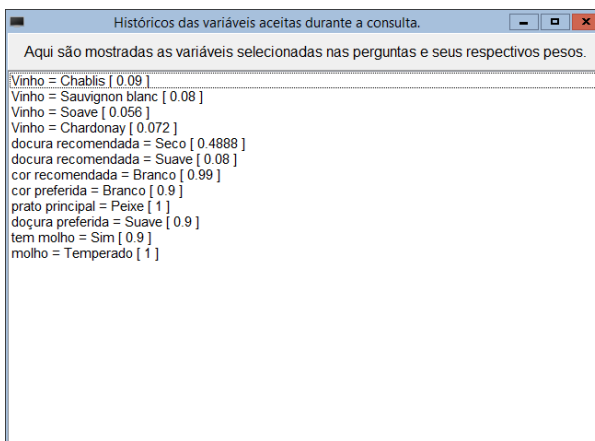


Figura 13 - Tela das variáveis utilizadas pelo SE.

Outra opção na depuração é a possibilidade de o usuário visualizar as regras processadas pela máquina de inferência para o SE chegar ao resultado. Assim, o usuário pode analisar as regras e conferir se o funcionamento do SE está de acordo com o esperado. Um exemplo na Figura 14 abaixo mostra como a tela contendo estas informações é apresentada.

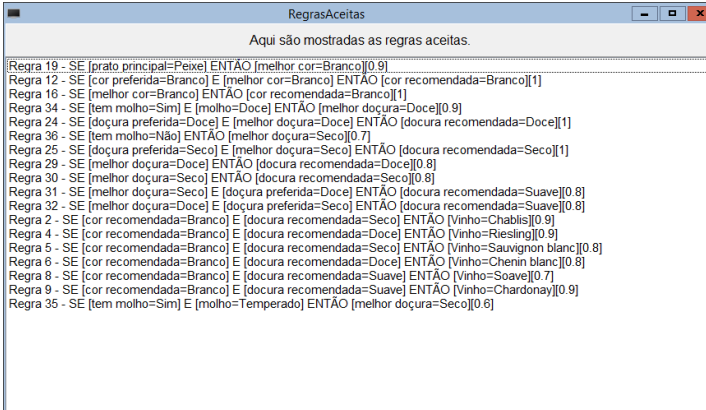


Figura 14 - Tela de regras processadas pelo SE.

Por fim, temos a árvore de decisão, gerada a partir das informações coletadas durante a execução do SE, para mostrar como são processadas as regras e variáveis do SE, e como a máquina de inferência apresenta o resultado do problema. Como podemos ver na figura 14 abaixo, as regras aceitas são marcadas com um círculo verde, e as regras rejeitadas são marcadas com um círculo vermelho; quanto às perguntas realizadas na consulta, estas são identificadas com um ponto de interrogação.

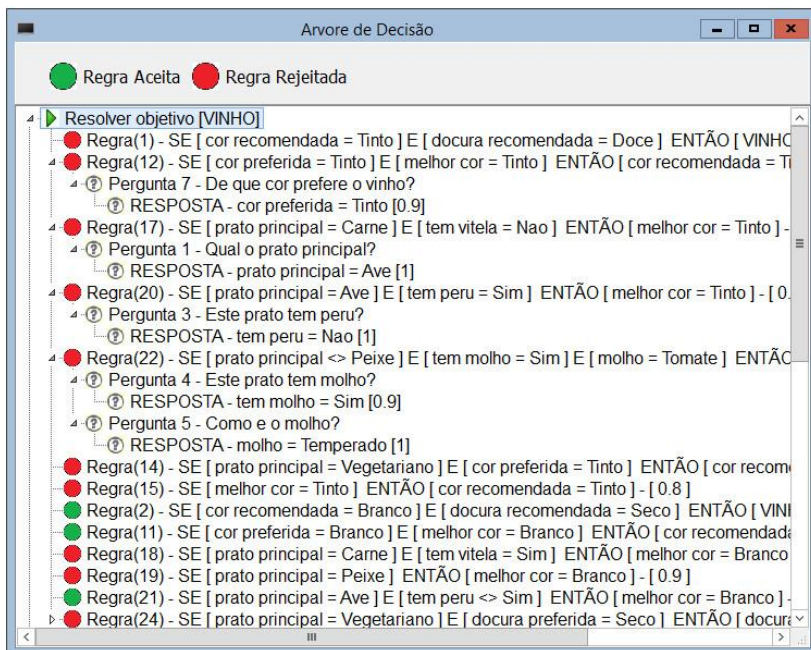


Figura 15 - Árvore de decisão para visualizar o raciocínio do SE.

## 4.AVALIAÇÃO

Neste capítulo são descritas as atividades realizadas na avaliação do Intellec System II.

Para avaliar a corretude do sistema, foram realizados testes entre o *Expert Sinta*, o *Intellec System* e também o *Intellec System II*. Nesta avaliação, os problemas utilizados para realizar estes testes nos três *softwares* foi A base de conhecimento disponibilizada por Araribóia(1989), cujo problema a ser resolvido é a escolha do melhor vinho para uma refeição e também a base de conhecimento criada por por Azevedo(1993), com a finalidade de indicar o diagnóstico para as doenças de Artrite Reumatoide e Lúpus Eritematoso Sistêmico.

Em cada um dos sistemas foram desenvolvidas praticamente as mesmas regras e variáveis, procurando assemelhar sempre o fator de certeza entre esses sistemas. Devido às limitações da *shell Intellec System* na versão I e II em relação à *shell Expert Sinta*, o número de regras variou. A ordem das regras foi mantida nas três implementações, já que esta ordem define a sequência das perguntas que serão feitas ao usuário.

Em cada uma das *shells*, foram realizados diversos testes com as bases criadas respectivamente para cada *shell*. Vários casos foram simulados para verificar se a máquina de inferência chegava a um resultado, e se eram os mesmos. As tabelas comparativas dos resultados são apresentadas. Deve-se levar em consideração o fato de que os valores de certeza para o *Expert Sinta* e *Intellec System II* variam de 0 a 100, e o *Intellec System*, entre 0 e 1.

A base de conhecimento desenvolvida a fim de se ter um sistema especialista para vinhos consiste em se escolher o melhor vinho para uma determinada refeição. O usuário dá à *shell* informações necessárias a respeito da refeição que será servida e o gosto de vinhos. A implementação desta base de conhecimento utilizada para os testes encontra-se no anexo A deste trabalho.

O teste envolvendo a primeira base de conhecimento que tem como objetivo a variável *vinho* considerando-se que a refeição tinha como prato principal peixe com molho temperado. A preferência do usuário quanto à cor do vinho foi a branca (vinho branco), com doçura suave. Neste caso foi considerado que as respostas informadas pelo usuário tinham seu valor *verdade* igual a 1 para o tipo de comida e tipo de molho. Já para o vinho, quanto à doçura e à cor, e quanto a molho na refeição, o valor *verdade* foi definido com 0,9. No *Expert Sinta* os



valores do fator de certeza foram definidos com 100 e 90 respectivamente.

Abaixo segue a tabela 1 com o comparativo dos resultados obtidos nos *softwares*.

**Tabela 1- Comparativo dos resultados obtidos nos softwares**

<i>Expert Sinta</i>	Resultado %	<i>Intellec System</i>	Resultado %	<i>Intellec System II</i>	Resultado %
Chablis	38,857	Chablis	0,410	Chablis	41,05
Sauvignon Blanc	32,721	Sauvignon Blanc	0,345	Sauvignon Blanc	34,50
		Chardonay	0,072	Chardonay	7,20
		Soave	0,056	Soave	5,60

Em relação à tabela acima apresentada, analisando os resultados apresentados pelas *shells*, verificamos que são semelhantes apesar de o *Expert Sinta* apresentar menos respostas, o que ocorre quando uma resposta feita ao usuário apresenta o valor de uma variável diferente do valor da premissa. Nesse caso esta regra é descartada, enquanto que no *Intellec*, ao invés de descartá-la, é somado 1 menos ao valor *verdade* fornecido pelo usuário. Isso se deve ao fato de o *Expert Sinta* estar verificando as premissas: se o valor fornecido pelo usuário para uma determinada variável for diferente do valor desta variável na premissa, a regra é descartada. No caso dos *Intellects*, quando a resposta obtida é diferente da resposta esperada, o valor *verdade* desta premissa é assumido com 1 menos o valor *verdade* fornecido pelo usuário, ao invés de descartar totalmente a premissa, como o que ocorre no *Expert Sinta*.

Agora um problema da área médica foi utilizado para executar novos testes comparativos entre as *shell's*. Este base é contida por 2 doenças, 14 sintomas e 2 exames clínicos. As doenças contidas nesta base de conhecimento são: Artrite Reumatoide (AR) e Lupus Erimatoso Sistemico (LES).

Artrite Reumatoide é uma doença inflamatória crônica de etiologia desconhecida, que afeta principalmente as membranas sinoviais, apesar de outros tecidos e órgãos também podem estar envolvidos. Há evidências de que um sistema imunológico desordenada desempenha um papel na patogênese.

Lupus Erimatoso Sistemico é uma doença inflamatória crônica de causa desconhecida que afeta a pele, articulações, rins, sistema nervoso e muitas vezes outros órgãos do corpo. É geralmente classificado com o colagénio ou distúrbios do tecido conjuntivo, e auto-imunidade estar associada na patogênese.

A segunda base de conhecimento utilizada na avaliação tem por objetivo indicar um diagnóstico para as doenças: Artrite reumatoide e Lúpus Erimatoso Sistemico, analisando 14 sintomas diferentes ( febre, mialgia, manifestações renais, manifestações pulmonares, manifestações do sistema nervoso central, alopecia, erupção borboleta, artrite, artralgia, fenômeno de Raynaud, fotosensibilidade, mão reumatoide, nódulos subcutâneos e rigidez matinal).

Outra maneira de se indicar qual o diagnóstico do paciente é através da realização de exames para cada doença. Caso o exame seja positivo para fator reumatoide e negativo para o exame ANA (*Antinuclear Antibodies*), indica que o diagnóstico para a doença é Artrite Reumatoide. Caso ocorrer o contrário, negativo para o fator reumatoide e positivo para o ANA, o diagnóstico para a doença é Lupus Erimatoso Sistemico.

Iniciando a consulta a esta base, os sintomas do caso avaliado são questionados ao usuário um a um, e então ao final é indicado uma sugestão para o diagnóstico

A segunda parte da consulta à base é destinada a obtenção dos resultados dos exames realizados pelo paciente, assim podendo indicar com precisão o diagnóstico do paciente.

Na realização do teste para o primeiro caso, foi definido como variável objetivo a sugestão para diagnóstico. Para encontra-lo foram definidos os seguintes sintomas para o paciente: rigidez matinal mialgia e manifestações renais que foram respondidos ao sistema com valor 1. Para os demais sintomas foram respondidos com valor 0.

Na tabela 2 que segue abaixo, apresentada o comparativo entre o Expert System e o Intellec System 2 para a sugestão de diagnóstico baseada nos sintomas apresentados.

**Tabela 2- Comparativo dos resultados obtidos nos softwares para o caso 1.**

<i>Expert Sinta</i>	Resultado %	<i>Intellec System II</i>	Resultado %
Artrite Reumatoide	94,4	Artrite Reumatoide	94,4
Lupus Eritematoso Sistemico	58,0	Lupus Eritematoso Sistemico	57,99

Abaixo na Tabela 3, consta o resultado comparativo referente ao exame do fator Reumatoide e o ANA. Foi definido para os exames os valores 1 para fator reumatoide com resultado positivo e 0 para ANA com resultado negativo.

**Tabela 3- Resultados dos exames para o caso 1**

<i>Expert Sinta</i>	<b>Resultado %</b>	<i>Intellec System II</i>	<b>Resultado %</b>
Artrite Reumatoide	100	Artrite Reumatoide	89,99

Como é apresentado nas tabelas 2 e 3, o diagnostico encontrando é para Artrite Reumatoide.

No segundo caso testado, os sintomas informados ao sistema foram: artralgia, erupção borboleta, fenômeno de Raynaud's e alopecia, como presentes neste caso respondidos a shell com o valor 1. Os sintomas restantes foram respondidos com o valor 0.

Na tabela 4 abaixo, apresentada o comparativo entre o Expert System e o Intellec System 2 para a sugestão de diagnostico baseada nos sintomas apresentados no caso 2.

**Tabela 4- Comparativo dos resultados obtidos nos softwares para o caso 2**

<i>Expert Sinta</i>	<b>Resultado %</b>	<i>Intellec System II</i>	<b>Resultado %</b>
Lupus Eritematoso Sistemico	99,25	Lupus Eritematoso Sistemico	99,25
Artrite Reumatoide	50	Artrite Reumatoide	50

Abaixo na Tabela 5, consta o resultado comparativo referente ao exame do fator Reumatoide e o ANA. Foi definido para os exames os valores 0 para fator reumatoide com resultado negativo e 1 para ANA com resultado positivo.

**Tabela 5- Resultados dos exames para o caso 2**

<i>Expert Sinta</i>	<b>Resultado %</b>	<i>Intellec System II</i>	<b>Resultado %</b>
Artrite Reumatoide	100	Artrite Reumatoide	89,99

Como é apresentado nas tabelas 4 e 5, o diagnostico encontrando é para Lupus Erimatoso Sistemico.

No terceiro e ultimo caso testado, os sintomas informados na execução da base foram: febre, artralgia, artrite manifestações renais e manifestações do sistema nervoso central, como presentes neste caso respondidos a shell com o valor 1. Os sintomas restantes foram respondidos com o valor 0.

Na tabela 6 que segue abaixo, são apresentados os valores comparativos entre o Expert System e o Intellec System 2 para a sugestão de diagnóstico baseada nos sintomas apresentados no caso 3.

**Tabela 6- Comparativo dos resultados obtidos nos softwares para o caso 3**

<i>Expert Sinta</i>	Resultado %	<i>Intellec System II</i>	Resultado %
Lupus Eritematoso Sistemico	91,6	Lupus Eritematoso Sistemico	91,6
Artrite Reumatoide	87,2	Artrite Reumatoide	87,2

Abaixo na Tabela 7, consta o resultado comparativo referente ao exame do fator Reumatoide e o ANA. Foi definido para os exames os valores 1 para fator reumatoide com resultado positivo e 0 para ANA com resultado negativo.

**Tabela 7- Resultados dos exames para o caso 3**

<i>Expert Sinta</i>	Resultado %	<i>Intellec System II</i>	Resultado %
Artrite Reumatoide	100	Artrite Reumatoide	89,99

Com base nos resultados apresentados na tabela 6, podemos observar que os valores são mais próximos que nos outros dois casos anteriores. Esta proximidade é justificada pelo fato que os sintomas apresentados ao *shell* são sintomas que podem ser identificados nas duas doenças quando elas estão em estágio inicial. Na tabela 7 acima podemos verificar que a *shell* indica o diagnóstico para Lupus Eritematoso Sistemico, mas no exame é indicado o diagnóstico como Artrite Reumatoide.

Os valores demonstrados acima indicam que os resultados são semelhantes em ambos as *shells* permitindo que o Intellec System 2 possa ser utilizado para criação de sistemas especialistas em vista que seus resultados coincidem com os do Expert System uma *shell* já renomada no meio acadêmico.

Outra avaliação foi realizada para medir a satisfação do usuário. Nesta avaliação foram considerados os seguintes procedimentos: definição do questionário de avaliação, apresentação do sistema, aplicação do questionário, identificação dos pontos fortes e fracos e novas oportunidades.

Para realizar a avaliação desse sistema, foi utilizada uma metodologia juntamente com um questionário chamado de *Questionnaire for User Interaction Satisfaction* (QUIS), que foi adaptado e criado na Universidade de *Maryland* - USA, e que faz parte da metodologia criada por Almeida (2011) para criação e desenvolvimento de *software* na área de engenharia biomédica, com o objetivo de avaliar a satisfação subjetiva dos usuários com aspectos específicos da interface homem-máquina.

Quanto ao grupo de usuários, foi definido que esta avaliação seria realizada com os alunos participantes do 3º Minicurso de Engenharia Biomédica na prática, realizado pelo Instituto de Engenharia Biomédica da UFSC, no período de 5 a 7 de agosto de 2010, a fim de obter uma primeira avaliação desta versão do sistema.

Esse minicurso é aberto a alunos da graduação dos cursos de engenharia que cursam entre 3ª e o 8ª semestre, com idades que variam de 19 a 24 com uma média geral de 22 anos. O total de participantes foi de 11 pessoas, sendo 9 do sexo masculino e duas do sexo feminino.

O QUIS contém originalmente doze seções de perguntas, sendo que cada seção corresponde a um aspecto sobre usabilidade de sistemas a ser avaliado. Para este trabalho foram selecionadas apenas cinco seções que se mostram pertinentes ao *software* desenvolvido, contendo um total de cinquenta e duas perguntas.

Estas seções selecionadas são respectivamente:

- Reações gerais dos usuários;
- Informações sobre a tela;
- Terminologia e Sistema de informação;
- Aprendizagem;
- Capacidades do Sistema.

Na primeira seção, que aborda as Reações Gerais dos Usuários Sistema, o usuário deve responder perguntas que abrangem uma visão geral sobre uso, aprendizado e interação do sistema que está sendo avaliado.

Sobre informações sobre tela, na seção seguinte, o usuário responde sobre tamanho e definição dos caracteres, *layouts* das telas e disposição das informações.

Na terceira seção, que contém questões sobre terminologia e sistema de informação, o usuário irá responder perguntas sobre as mensagens que aparecem na tela, instruções de correções de erros e instruções sobre como realizar as tarefas.

Já na próxima seção, as questões a serem respondidas são referentes à aprendizagem na utilização do sistema, como operação do sistema, descoberta de novas características e nome de comandos.

Na última seção os usuários devem responder a questões sobre as capacidades do sistema, como: se é confiável, quantidade de informação apresentada, velocidade e facilidade de operação, de acordo com o nível de experiência de cada um

A seguir, foi preparada uma aula sobre Sistemas Especialistas para ser ministrada no 3º Minicurso de Engenharia Biomédica na prática descrito abaixo, e também uma breve apresentação do sistema. Após a apresentação, foi sugerido aos alunos que recriassem parte de uma base de conhecimento já existente para iniciar a utilização do sistema.

No próximo passo, foi aplicado o questionário definido anteriormente com o objetivo de poder avaliar como o usuário utiliza o sistema e quais pontos ainda devem sofrer melhorias, baseando-se nas respostas e em críticas descritas ao fim de cada seção do questionário.

As respostas das questões devem ser expressas escolhendo-se um valor correspondente na faixa determinada entre 1 a 9. Outro valor a ser escolhido é o NA, que corresponde às perguntas cujos conteúdos foram interpretados pelo usuário como não aplicáveis ao assunto que estava sendo avaliado.

Para extrair os resultados desta avaliação, foi utilizada a escala de Likert, separando a faixa em quatro grupos, de 1 a 3, de 4 a 6, 7 a 9 e NA (Não Atende).

Obtendo as médias de cada grupo em cada seção, como é mostrado na tabela 8 abaixo, foi possível calcular a média geral da avaliação separada pelos grupos.

**Tabela 8 - Comparativo por faixa de notas**

Seções	1 a 3 (Ruim)	4 a 6 (Médio)	7 a 9 (Ótimo)	NA
1-Reações Gerais	4,55%	31,82%	63,64%	0,00%
2-Tela	9,85%	11,36%	77,27%	1,52%
3-Terminologia	6,06%	15,15%	67,42%	11,36%
4-Aprendizagem	1,01%	18,18%	80,81%	0,00%
5-Capacidades do Sistema	16,67%	15,91%	48,48%	18,94%
<b>Médias</b>	<b>7,63%</b>	<b>18,48%</b>	<b>67,37%</b>	<b>6,36%</b>

Esta avaliação apontou resultados que podemos avaliar. O primeiro deles se refere à seção **4 - Aprendizagem**, onde 80,81% das

notas foram ente 7 e 9, indicando que esta seção apresentou o melhor resultado quando comparado ao das outras seções.

Outro ponto indicado pelos resultados foi na seção **5 – Capacidades do Sistema**, onde foram somadas as porcentagens das notas com faixa de 1 a 3, de faixa 4 a 6 e ainda NA, apresentando um total de 51,52, indicando a insatisfação dos usuários com o sistema nesta seção. Porém, essas notas baixas podem ser provindas da inexperiência dos participantes com o sistema e com o assunto.

Fazendo a média dos resultados desses grupos, obtemos o gráfico na figura 14 abaixo, onde é possível identificar que em 67,37% das cinquenta e duas perguntas, os usuários responderam que consideraram o sistema *ótimo*, indicado pelo grupo de 7 a 9.

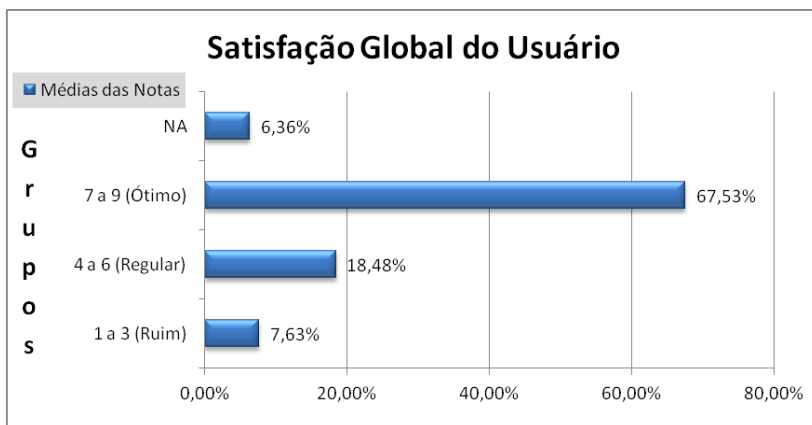


Figura 16 - Gráfico de Satisfação Global

Analisando os valores acima apresentados na Tabela 8 e na Figura 16, podemos verificar que a faixa de valores em 7 e 9 foi a que teve o maior número de ocorrências. Assim, somos levados a concluir que os usuários participantes desta avaliação consideraram o *software* “Ótimo”, de acordo com a escala criada para esta avaliação.

Algumas observações foram feitas sobre o *software*. A principal foi a falta de atalhos do teclado e do *mouse* para facilitar a criação da base de conhecimento no preenchimento dos campos necessários.

A última parte da avaliação foi a utilização do *Intellec System II* em uma dissertação de mestrado no programa de Pós-Graduação em

Ciências Médicas da UFSC: a de CUNHA (2012). Esse trabalho de mestrado consiste em criar base de conhecimento para uso em um Sistema Especialista, levando em conta que a avaliação funcional das atividades de levantar e sentar na cadeira é frequentemente utilizada por fisioterapeutas, como por exemplo, ao examinar pacientes com disfunções neurológicas e musculoesqueléticas. Tal base de conhecimento visa a auxiliar no suporte à avaliação das atividades de levantar e sentar na cadeira, baseando-se num protocolo clínico, por meio da utilização da *shell Intellec System II*. A referida dissertação representou um excelente mecanismo de validação ao presente trabalho, tendo nos levado à descoberta de muitos problemas, às vezes dos mesmos erros, que foram consertados ou contornados imediatamente. Ademais, serviu como um laboratório experimental para um melhor entendimento de como apresentar as novas funcionalidades e melhorar as já existentes.

O pedido de registro de patente deste *software* será solicitado por intermédio do Departamento de Propriedade Intelectual da UFSC.



## 5. CONSIDERAÇÕES FINAIS E CONCLUSÕES

A presente pesquisa teve como finalidade desenvolver uma *shell* para implementação de Sistemas Especialistas por usuários que não tenham grande experiência em técnicas de inteligência artificial, permitindo que pequenos sistemas possam ser construídos diretamente pelos especialistas de domínio. Também tem como finalidade fornecer uma ferramenta poderosa de desenvolvimento de SE para usuários especialistas que não disponham de grande tempo para prototipagem.

Para o desenvolvimento do *Intellec System II*, foi utilizado como base um sistema semelhante, o *Intellec System*, desenvolvido no mesmo laboratório de Informática médica do Instituto de Engenharia Biomédica (IEB) -UFSC.

Este último sistema foi avaliado em seu desempenho, principalmente em relação a seus pontos fortes e fracos, através da verificação dos trabalhos nele desenvolvidos em nosso laboratório, o que permitiu que ficasse claro o que poderia ser esperado de uma nova versão.

O novo sistema, apesar do mesmo nome, apresenta modificações substanciais em relação ao anterior: a linguagem de programação foi mudada, novas funcionalidades foram acrescentadas e uma nova interface homem-máquina foi desenvolvida. Foi, todavia, mantida a máquina de inferência da primeira versão – uma máquina Prolog, obtida do *software* Visual Prolog da empresa *Prolog Development Center A/S*, usada como uma DLL.

A avaliação de corretude obteve resultados iguais entre os dois *Intellecs*, e, comparando-o com o *Expert Sinta*, obtivemos respostas similares devido às diferenças entre as máquinas de inferência do *Expert Sinta* e as utilizadas nos *Intellecs*. Assim, podemos afirmar que uma mesma base de conhecimento pode ser desenvolvida em qualquer uma das três *shells*, uma vez que o resultado será o mesmo.

A avaliação de satisfação realizada neste trabalho utilizando o *Intellec System* tem a finalidade de verificar a usabilidade do sistema tanto com o especialista que vai criar o SE quanto com o usuário final que irá realizar as consultas dos SEs criados. Nesta avaliação foi apresentado um resultado que aponta o *software* com aproximadamente 68% de aprovação dos usuários que participaram da mesma.

Dentre as contribuições deste trabalho está uma melhoria significativa da interface, principalmente para o usuário especialista que teve uma melhoria na usabilidade. Foi possibilitada também a utilização de vídeos como valores de variáveis que são apresentados na consulta,

durante a qual podem ser exibidos. As variáveis também podem ser definidas como imagens, e neste trabalho foi possibilitado o uso de outros tipos de imagens, como JPEG, TIFF, PNG, dentre outras.

A máquina de inferência Prolog utilizada foi criada como uma DLL, tendo sido permitida sua utilização por outra linguagem como a C++ neste trabalho.

A geração de uma árvore de decisão também contribuiu para a realização da pesquisa, mostrando ao usuário como o SE chegou a uma conclusão, podendo ajudar no processo de criação da base de conhecimento a fim de chegar ao resultado esperado.

Como sugestão para trabalhos futuros, podemos citar:

- Processamento do arquivo contendo a base de conhecimento “arquivo.esp” através de um banco de dados para preservar os relacionamentos entre as entidades e manter a integridades das informações.
- Possibilidade de criação de regras com mais de uma conclusão, o que reduzirá o número de regras.
- Realização de exportação da base de conhecimento criada para o uso em outras linguagens utilizando-se das regras já validadas no *Intellec System II*.
- Permissão para construir regras com o conectivo “ou” para reduzir o número de regras.
- Criação de um tutor inteligente para auxiliar a criação do SE. Este tutor poderá auxiliar todos os usuários a criar suas bases de conhecimento fornecendo informações passo a passo.

## 6.REFERÊNCIAS

ALGARVE, A. S.; DE AZEVEDO, .F; BARRETO, J. M. **Prolog Implementation of a Shell for Developing Fuzzy Expert System**. In: Electro 95, Punta Arenas, Chile, 1995.

ALMEIDA, M. A. F.;AZEVEDO, F. M. ou de Azevedo, F.M. **Avaliação de sistema de hipermídia adaptativos educacionais: uma aplicação em engenharia biomédica**. 2011. Tese (Doutorado em Programa de Pós Graduação em Engenharia Elétrica) - Universidade Federal de Santa Catarina.

ANGELONI, M.N.M; **Fuzzy Expert System Shell Implemented Using Prolog**. In: Chicago 2000 – World Congress on Medical Physics and Biomedical Engineering, Navy Pier, Chicago, IL, 23 a 28 julho de 2000. Abstract.

ANGELONI, M.N.M.; AZEVEDO, F.M., ALGARVE, A.S. **Implementação de um Shell para desenvolvimento de sistemas especialistas fuzzy usando Prolog**. IN: CBEB 2000 – XVII Congresso Brasileiro de Engenharia Biomédica, Florianópolis, 11 a 13 de setembro de 2000, pp. 1124-1127.

ARARIBÓIA, G.;1988. **Inteligência artificial: um curso prático**. Rio de Janeiro: Livros Técnicos e Científicos.

AZEVEDO, F. M.; FERRARI, G. L.; ANGELONI, M. N. M.; ARGOUD, F. I. M. **Shell para Desenvolvimento de Sistemas Especialistas Fuzzy usando Prolog**. IN: III Congresso Latino Americano de Engenharia Biomédica (Set. 2004: João Pessoa, Paraíba) Anais, Paraíba. Págs. 919-922.

AZEVEDO, F. M. **Contribution to the Study of Neural Networks In Dynamical Expert Systems**.1993.Tese (Doutorado no Institut d'Informatique, FUNDP, Belgica)

BARRETO, J. M.; 1997. **Inteligência artificial – no limiar do século XXI**. Florianópolis: PPP Edições 291pp.

BITTENCURT, G.; 2001. **Inteligência artificial: ferramentas e teorias**. Florianópolis: Editora da UFSC. 362 pp.

CUNHA, M. J.; AZEVEDO, F. M. ou de Azevedo, F.M. **Base de conhecimento para sistema especialista de suporte na avaliação das atividades de levantar e sentar - protocolo clínico**. 2012. Dissertação (Mestrado em Ciências Médicas) - Universidade Federal de Santa Catarina.

FERNANDES, A. M. R. **Inteligência artificial aplicada à saúde**. Florianópolis: Visual Books, 2004. 195 p.

FERRARI, G. L.; DE AZEVEDO, F. M. **Intellec: Shell para implementação de sistemas especialistas**. 2005. 73 f. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-graduação em Engenharia Elétrica

FERRARI, G.L.; ARGOUD, F. I. M.; AZEVEDO, F. M. **Shell para Desenvolvimento de Sistemas Especialistas Fuzzy – Estudo de Caso: Gastroenterologia**. IV Workshop de Informática aplicada à Saúde – CBComp (Outubro 2004: Itajaí, Santa Catarina). Anais, Santa Catarina, 2004 p. 583-588.

HAHNE, M. N. M. A.; AZEVEDO, F. M.; ALGARVE, A. S. **Implementação de um shell para desenvolvimento de sistemas especialistas fuzzy usando prolog**. 2001. 120f. Dissertação de Mestrado (Engenharia Elétrica) – Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-graduação em Engenharia Elétrica

HEINZLE, R.; RABUSKE, R. A. **Protótipo de uma ferramenta para criação de sistemas especialistas baseados em regras de produção**. 1995. 145f. Dissertação (Mestrado) - Universidade Federal de Santa Catarina. Centro Tecnológico.

KOSKO, B. **Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence**. Prentice-Hall. 1992.

MONTELLO, M. V. **Sistema Especialista para predição de complicações cardiovasculares integrado a um sistema de controle de pacientes portadores de Diabetes Mellitus**. 1999. 152f. Dissertação

de Mestrado (Engenharia Elétrica) – Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-graduação em Engenharia Elétrica.

NOGUEIRA, J. H. M.; ANDRADE E SILVA, R. B., ALCÂNTARA, J. F. L., HOLANDA, S. C., ANDRADE, R. C.; 1996. **Expert Sinta: uma ferramenta visual geradora de Sistemas Especialistas**. VI Semana de Informática, Universidade Federal da Bahia, 1996.

PAMPLONA FILHO, C. R. ; FERRARI, G. L. ; CUNHA, M. J. ; AZEVEDO, F. M. Intellec System: **Shell for expert systems creation with fuzzy inference machine developed in prolog**. In: International Conference on System and Engineering in Taiwan 2010, 2010, Taipé., 2010.

PRADE, H. **A computacional approach to approximate and plausible reasoning with applications to Expert Systems**. IEEE Transactions on pattern analysis and machine intelligence , v. PAMI-7 , n. 3, May. 1985, p. 260-283.

RICH, K.; ROBINSON, P. B. **Using turbo prolog**. ed.2 Berkeley: Osborne McGraw-Hill, 1988. 369p.

PROLOG DEVELOPER CENTER. **Visual Prolog language tutorial**. Copenhagen, 1996.

## **7.APÊNDICE A – Questionário de Avaliação**

## INSTRUMENTO DE AVALIAÇÃO DA SATISFAÇÃO DO USUÁRIO

Adaptado do “Questionnaire for User Interaction Satisfaction” (QUIS)

Olá,

Primeiro, temos algumas questões iniciais antes de começarmos o questionário. Por favor, responda às seguintes perguntas sobre você:

**Qual a sua idade?** \_\_\_\_\_

Sexo:

Feminino

Masculino

Por favor, entre com o nome do sistema avaliado: \_\_\_\_\_

Parte 1: Sua experiência com sistemas

**Há quanto tempo você tem utilizado este sistema?**

Menos de 1 hora

1 hora a 1 dia

1 dia a 1 semana

1 semana a 1 mês

1 a 6 meses

6 meses a 1 ano

1 a 2 anos

2 a 3 anos

3 anos ou mais

**1.2 Em média, quanto tempo você gasta por semana com este sistema?**

Menos de 1 hora

1 a 4 horas

4 a 10 horas

Acima de 10 horas

Parte 3: Reações gerais dos usuários

**Por favor, marque os números que refletem mais adequadamente as suas impressões sobre a utilização do sistema avaliado. Marque NA quando não for aplicável.**

Exemplo:

Você pensa que o sistema é:

Terrível											Maravilhoso	
	1	2	3	4	5	6	7	8	9			NA

3.1. Você pensa que o sistema é:

Terrível											Maravilhoso	
	1	2	3	4	5	6	7	8	9			NA

3.2. Em relação à interação, o sistema é:

Frustrante											Excitante	
	1	2	3	4	5	6	7	8	9			NA

3.3. Em relação ao uso geral o sistema é:

Tedioso											Estimulante	
	1	2	3	4	5	6	7	8	9			NA

3.4. Em relação ao manuseio o sistema é:

Difícil											Fácil	
	1	2	3	4	5	6	7	8	9			NA

3.5. Em relação a seu aprendizado após usar o sistema, você teve um:

Domínio Inadequado											Domínio Adequado	
	1	2	3	4	5	6	7	8	9			NA

3.6. A concepção geral do sistema é:

Rígida											Flexível	
	1	2	3	4	5	6	7	8	9			NA

Escreva comentários sobre este tópico:

Parte 4: Tela

Por favor, marque os números que refletem mais adequadamente as suas impressões sobre a utilização do sistema avaliado. Marque NA quando não for aplicável.



4.1 Os caracteres na tela do computador estão:

Difíceis de Ler											Fáceis de Ler	
	1	2	3	4	5	6	7	8	9			NA

4.1.1. A imagem dos caracteres está:

Esmacida											Definida	
	1	2	3	4	5	6	7	8	9			NA

4.1.2. As formas de caracteres (fontes) estão:

Pouco Legíveis											Muito Legíveis	
	1	2	3	4	5	6	7	8	9			NA

4.2. Os destaques na tela foram:

Inúteis											Úteis	
	1	2	3	4	5	6	7	8	9			NA

4.2.3. O uso de negrito foi:

Inútil											Útil	
	1	2	3	4	5	6	7	8	9			NA

4.3. Os *layouts* de tela foram úteis?

Nunca											Sempre	
	1	2	3	4	5	6	7	8	9			NA

4.3.1. A quantidade de informação mostrada na tela foi :

Inadequada											Adequada	
	1	2	3	4	5	6	7	8	9			NA

4.3.2. A disposição da informação na tela foi :

Ilógica											Lógica	
	1	2	3	4	5	6	7	8	9			NA

4.4. A sequência de telas é:

Confusa											Clara	
	1	2	3	4	5	6	7	8	9			NA

4.4.1. A próxima tela na sequência é:

Imprevisível										Previsível	
	1	2	3	4	5	6	7	8	9		NA

4.4.2 O retorno à tela anterior é:

Impossível										Fácil	
	1	2	3	4	5	6	7	8	9		NA

4.4.3. A progressão de trabalhos relacionados com as tarefas foi :

Confusa										Claramente definida	
	1	2	3	4	5	6	7	8	9		NA

Escreva comentários sobre este tópico:

## Parte 5: Terminologia e Sistema de Informação

Por favor, marque os números que refletem mais adequadamente as suas impressões sobre a utilização do sistema avaliado. Marque NA quando não for aplicável.

5.1. O uso de terminologia em todo o sistema foi:

Inconsistente										Consistente	
	1	2	3	4	5	6	7	8	9		NA

5.2. A terminologia se relaciona bem com as atividades feitas?

Apropriadamente										Com muita frequência	
	1	2	3	4	5	6	7	8	9		NA

5.3 As mensagens que aparecem na tela são:

Confusas										Claras	
	1	2	3	4	5	6	7	8	9		NA

5.3.1 A posição de instruções na tela é:

Inconsistente										Consistente	
	1	2	3	4	5	6	7	8	9		NA

5.4 As mensagens que aparecem na tela são:

Confusas											Claras	
	1	2	3	4	5	6	7	8	9			NA

5.4.1 As instruções para os comandos e funções são:

Confusas											Claras	
	1	2	3	4	5	6	7	8	9			NA

5.4.2 As instruções para correção de erros são:

Confusas											Claras	
	1	2	3	4	5	6	7	8	9			NA

5.5 O computador mantém você informado sobre o que está acontecendo?

Nunca											Sempre	
	1	2	3	4	5	6	7	8	9			NA

5.5.2 A realização de uma operação conduz a um resultado previsível?

Nunca											Sempre	
	1	2	3	4	5	6	7	8	9			NA

5.5.4 A duração de atraso entre as operações é:

Inaceitável											Aceitável	
	1	2	3	4	5	6	7	8	9			NA

5.6 As mensagens de erros são:

Inúteis											Úteis	
	1	2	3	4	5	6	7	8	9			NA

5.6.1 As mensagens de erros clarificam o problema?

Nunca											Sempre	
	1	2	3	4	5	6	7	8	9			NA

5.6.2 As frases de mensagens de erro são:

Desagradáveis											Agradáveis	
	1	2	3	4	5	6	7	8	9			NA

Faça comentários adicionais sobre este tópico aqui:

Parte 6: Aprendizagem

**Por favor, marque os números que refletem mais adequadamente as suas impressões sobre a utilização do sistema avaliado. Marque NA quando não for aplicável.**

6.1 A aprendizagem para operar o sistema foi:

Difícil											Fácil	
	1	2	3	4	5	6	7	8	9			NA

6.1.3. O tempo para aprender a usar o sistema foi:

Lento											Rápido	
	1	2	3	4	5	6	7	8	9			NA

6.2. A exploração de características por tentativa e erro foi:

Desanimadora											Encorajadora	
	1	2	3	4	5	6	7	8	9			NA

6.2.1 A exploração de características foi:

Arriscada											Segura	
	1	2	3	4	5	6	7	8	9			NA

6.2.2 A descoberta de novas características foi:

Difícil											Fácil	
	1	2	3	4	5	6	7	8	9			NA

6.3 A memorização de nomes e uso de comandos foi:

Difícil											Fácil	
	1	2	3	4	5	6	7	8	9			NA

6.4 As tarefas podem ser desenvolvidas de maneira direta?

Nunca											Sempre	
	1	2	3	4	5	6	7	8	9			NA

6.4.1 O número de passos por tarefa foi:

Muito											Pouco	
	1	2	3	4	5	6	7	8	9			NA

6.4.2 Os passos para completar uma tarefa seguiram uma seqüência lógica?

Nunca											Sempre	
	1	2	3	4	5	6	7	8	9			NA

Faça comentários adicionais sobre este tópico aqui:

## Parte 7: Capacidades do Sistema

**Por favor, marque os números que refletem mais adequadamente as suas impressões sobre a utilização do sistema avaliado. Marque NA quando não for aplicável.**

7.1 A velocidade do sistema foi:

Lenta											Rápida	
	1	2	3	4	5	6	7	8	9			NA

7.1.2. A quantidade de informação apresentada foi feita de forma

Lenta											Rápida	
	1	2	3	4	5	6	7	8	9			NA

7.2 O sistema é confiável?

Nunca											Sempre	
	1	2	3	4	5	6	7	8	9			NA

7.2.1 As operações eram:

Dependentes											Independentes	
	1	2	3	4	5	6	7	8	9			NA

7.2.2 As falhas no sistema ocorreram:

Frequentemente											Raramente	
	1	2	3	4	5	6	7	8	9			NA

7.2.3 O sistema emitiu mensagens de avisos?

Nunca											Sempre	
	1	2	3	4	5	6	7	8	9			NA

7.3.1 Os dispositivos de *hardware* tendem em ser:

Ruidosos											Silenciosos	
	1	2	3	4	5	6	7	8	9			NA

## 7.3.2. Os sons gerados são:

Irritantes											Agradáveis	
	1	2	3	4	5	6	7	8	9			NA

## 7.4.1. A correção de erros tipográficos é:

Complexa											Simple	
	1	2	3	4	5	6	7	8	9			NA

## 7.4.2. A habilidade de executar operações de desfazer é:

Inadequada											Adequada	
	1	2	3	4	5	6	7	8	9			NA

## 7.5. A facilidade de operação depende do nível de experiência?

Nunca											Sempre	
	1	2	3	4	5	6	7	8	9			NA

## 7.5.2 O sistema permite uso de atalhos?

Com dificuldades											Facilmente	
		1	2	3	4	5	6	7	8	9		NA

Faça comentários adicionais sobre este tópico aqui:

## **ANEXO A – Base de Conhecimento**

introducao(["Eu sou um especialista em Vinhos.", "Espero que eu possa ajuda'-lo em sua escolha de um vinho apropriado para o seu jantar."])

nomeesp("Vinho")

fator\_certeza("fator1", "Tem certeza?":["Não sei", "Não", "Sim"],["0.5", "0", "1"])

fator\_certeza("fator2", "Tem certeza?":["Certeza Absoluta", "Mais ou Menos", "Não sei", "Quase Certo"],["0.9", "0.6", "0.5", "0.7"])

pergunta(1, "Qual o prato

principal?":["Vegetariano", "Ave", "Peixe", "Carne"],"", [], [], "prato principal")

pergunta(2, "Este prato tem vitela?":["Sim", "Nao"],"", [], [], "tem vitela")

pergunta(3, "Este prato tem peru?":["Sim", "Nao"],"", [], [], "tem peru")

pergunta(4, "Este prato tem molho?":["Sim", "Nao"], "Tem certeza?":["Certeza Absoluta", "Mais ou Menos", "Não sei", "Quase Certo"],["0.9", "0.6", "0.5", "0.7"], "tem molho")

pergunta(5, "Como e o

molho?":["Tomate", "Doce", "Temperado"],"", [], [], "molho")

pergunta(6, "Como prefere o vinho?":["Seco", "Suave", "Doce"], "Tem certeza?":["Certeza Absoluta", "Mais ou Menos", "Não sei", "Quase Certo"],["0.9", "0.6", "0.5", "0.7"], "docura preferida")

pergunta(7, "De que cor prefere o vinho?":["Branco", "Tinto"], "Tem certeza?":["Certeza Absoluta", "Mais ou Menos", "Não sei", "Quase Certo"],["0.9", "0.6", "0.5", "0.7"], "cor preferida")

regra(1, "Sim", "VINHO", "gamay", 0.9, ["Sim", "cor recomendada", "Tinto", "Sim", "docura recomendada", "Doce"])

regra(2, "Sim", "VINHO", "chablis", 0.95, ["Sim", "cor recomendada", "Branco", "Sim", "docura recomendada", "Seco"])

regra(3, "Sim", "VINHO", "cabernet sauvignon", 0.85, ["Sim", "cor recomendada", "Tinto", "Sim", "docura recomendada", "Seco"])

regra(4, "Sim", "VINHO", "riesling", 0.9, ["Sim", "cor recomendada", "Branco", "Sim", "docura recomendada", "Doce"])

regra(5, "Sim", "VINHO", "sauvignon blanc", 0.8, ["Sim", "cor recomendada", "Branco", "Sim", "docura recomendada", "Seco"])

regra(6, "Sim", "VINHO", "chenin blanc", 0.8, ["Sim", "cor recomendada", "Branco", "Sim", "docura recomendada", "Doce"])

regra(7, "Sim", "VINHO", "pinot noir", 0.9, ["Sim", "cor recomendada", "Tinto", "Sim", "docura recomendada", "Suave"])

regra(8, "Sim", "VINHO", "soave", 0.7, ["Sim", "cor recomendada", "Branco", "Sim", "docura recomendada", "Suave"])

regra(9, "Sim", "VINHO", "chardonay", 0.9, ["Sim", "cor recomendada", "Branco", "Sim", "docura recomendada", "Suave"])

regra(10, "Sim", "VINHO", "zinfandel", 0.85, ["Sim", "cor recomendada", "Tinto", "Sim", "docura recomendada", "Suave"])

regra(11, "Sim", "cor recomendada", "Branco", 1, ["Sim", "cor preferida", "Branco", "Sim", "melhor cor", "Branco"])

regra(12, "Sim", "cor recomendada", "Tinto", 1, ["Sim", "cor preferida", "Tinto", "Sim", "melhor cor", "Tinto"])



regra(13,"Sim","cor recomendada","Branco",1,["Sim","prato principal","Vegetariano","Sim","cor preferida","Branco"])  
 regra(14,"Sim","cor recomendada","Tinto",1,["Sim","prato principal","Vegetariano","Sim","cor preferida","Tinto"])  
 regra(15,"Sim","cor recomendada","Tinto",0.8,["Sim","melhor cor","Tinto"])  
 regra(16,"Sim","cor recomendada","Branco",0.8,["Sim","melhor cor","Branco"])  
 regra(17,"Sim","melhor cor","Tinto",0.9,["Sim","prato principal","Carne","Sim","tem vitela","Nao"])  
 regra(18,"Sim","melhor cor","Branco",0.6,["Sim","prato principal","Carne","Sim","tem vitela","Sim"])  
 regra(19,"Sim","melhor cor","Branco",0.9,["Sim","prato principal","Peixe"])  
 regra(20,"Sim","melhor cor","Tinto",0.8,["Sim","prato principal","Ave","Sim","tem peru","Sim"])  
 regra(21,"Sim","melhor cor","Branco",0.7,["Sim","prato principal","Ave","Nao","tem peru","Sim"])  
 regra(22,"Sim","melhor cor","Tinto",0.7,["Nao","prato principal","Peixe","Sim","tem molho","Sim","Sim","molho","Tomate"])  
 regra(23,"Sim","docura recomendada","Doce",0.7,["Sim","prato principal","Vegetariano","Sim","docura preferida","Doce"])  
 regra(24,"Sim","docura recomendada","Seco",0.7,["Sim","prato principal","Vegetariano","Sim","docura preferida","Seco"])  
 regra(25,"Sim","docura recomendada","Suave",0.7,["Sim","prato principal","Vegetariano","Sim","docura preferida","Suave"])  
 regra(26,"Sim","docura recomendada","Doce",0.8,["Sim","melhor docura","Doce"])  
 regra(27,"Sim","docura recomendada","Seco",0.8,["Sim","melhor docura","Seco"])  
 regra(28,"Sim","docura recomendada","Suave",0.8,["Sim","melhor docura","Seco","Sim","docura preferida","Doce"])  
 regra(29,"Sim","docura recomendada","Suave",0.8,["Sim","melhor docura","Doce","Sim","docura preferida","Seco"])  
 regra(30,"Sim","docura recomendada","Suave",0.7,["Sim","melhor docura","Suave"])  
 regra(31,"Sim","melhor docura","Doce",0.9,["Sim","tem molho","Sim","Sim","molho","Doce"])  
 regra(32,"Sim","melhor docura","Seco",0.6,["Sim","tem molho","Sim","Sim","molho","Temperado"])  
 regra(33,"Sim","melhor docura","Seco",0.7,["Sim","tem molho","Nao"])  
 regra(34,"Sim","melhor docura","Doce",0.5,["Sim","tem molho","Sim","Sim","molho","Tomate"])  
 explicacao(1,["Para recomendar um vinho, preciso","recomendar uma docura e uma cor","apropriadas. Para isto, preciso saber","qual o prato principal de seu jantar."])

```

explicacao(2,["Para recomendar uma cor e uma docura", "preciso saber de que
consta seu prato", "de carne."])
explicacao(3,["O peru e a unica ave que pede um", "vinho tinto, por isso
preciso saber se", "ha peru neste prato."])
explicacao(4,["O molho e importante na escolha de", "um vinho, por isso
preciso saber se", "tem molho no prato e qual o tipo de", "molho."])
explicacao(5,["O molho e importante na escolha de", "um vinho, por isso
preciso saber se", "tem molho no prato e qual o tipo de", "molho."])
explicacao(6,["Para recomendar uma das marcas", "conhecidas de
vinho", "preciso recomendar a cor", "e a docura do referido vinho."])
explicacao(7,["A fim de determinar a melhor cor", "e a melhor doc para o
vinho", "que vai acompanhar o seu jantar", "preciso de todos os dados sobre", "o
prato principal e sobre", "o molho que vai ser usado."])
tutorial(1, "")
tutorial(2, "")
tutorial(3, "")
tutorial(4, "")
tutorial(5, "")
tutorial(6, "")
tutorial(7, "")
variavel("cor preferida", ["Branco", "Tinto"], "Univalorada", "Linguística")
variavel("cor recomendada", ["Branco", "Tinto"], "Univalorada", "Linguística")
variavel("docura
preferida", ["Seco", "Suave", "Doce"], "Univalorada", "Linguística")
variavel("docura
recomendada", ["Seco", "Suave", "Doce"], "Univalorada", "Linguística")
variavel("melhor cor", ["Branco", "Tinto"], "Univalorada", "Linguística")
variavel("melhor
docura", ["Seco", "Doce", "Suave"], "Univalorada", "Linguística")
variavel("molho", ["Tomate", "Doce", "Temperado"], "Univalorada", "Linguística")
)
variavel("prato
principal", ["Vegetariano", "Ave", "Peixe", "Carne"], "Univalorada", "Linguística")
variavel("tem molho", ["Sim", "Nao"], "Univalorada", "Linguística")
variavel("tem peru", ["Sim", "Nao"], "Univalorada", "Linguística")
variavel("tem vitela", ["Sim", "Nao"], "Univalorada", "Linguística")
variavel("VINHO", ["gamay", "chablis", "cabernet
sauvignon", "riesling", "sauvignon blanc", "chenin blanc", "pinot
noir", "soave", "chardonay", "zinfandel"], "Multivalorada", "Linguística")
meta("cor recomendada")
meta("docura recomendada")
meta("VINHO")
historia([])

```

**ANEXO B – Base de conhecimento : Diagnóstico para as doenças  
Artrite Reumatoide e Lupos Erimatoso Sistemico. – Intellec System**

introducao(["Base de conhecimento sobre a Artrite Reumatoide e Lúpus", "Eritematoso Sistêmico"])  
 nomeesp("Cesar Roberto Pamplona Filho")  
 pergunta(1, "O paciente está com febre ?", [""], "", [], [], "febre")  
 pergunta(2, "O paciente apresenta sintomas de artralgia ?", [""], "", [], [], "artralgia")  
 pergunta(3, "O paciente apresenta artrite ?", [""], "", [], [], "artrite")  
 pergunta(4, "O paciente apresenta rigidez matinal ?", [""], "", [], [], "rigidezmatinal")  
 pergunta(5, "O paciente apresenta mialgia ?", [""], "", [], [], "mialgia")  
 pergunta(6, "O paciente apresenta nodulos subcutaneos ?", [""], "", [], [], "nodulo\_subcutaneo")  
 pergunta(7, "O paciente apresenta erupcao borboleta ?", [""], "", [], [], "erupcao borboleta")  
 pergunta(8, "O paciente apresenta fenômeno de Raynaud ?", [""], "", [], [], "raynaud")  
 pergunta(9, "O paciente apresenta fotosensibilidade?", [""], "", [], [], "fotosensibilidade")  
 pergunta(10, "O paciente apresenta alopecia ?", [""], "", [], [], "alopecia")  
 pergunta(11, "O paciente apresenta manifestações renais ?", [""], "", [], [], "manifestacoesrenais")  
 pergunta(12, "O paciente apresenta manifestações do sistema nervoso central ?", [""], "", [], [], "manifestacoessnc")  
 pergunta(13, "O paciente apresenta manifestações pulmonares ?", [""], "", [], [], "manifestacoespulmonares")  
 pergunta(14, "O paciente apresenta mão reumatóide ?", [""], "", [], [], "maoreumatoide")  
 pergunta(15, "O teste para fator reumatóide é positivo ?", [""], "", [], [], "exame\_fator\_reumatoide")  
 pergunta(16, "O teste FAN (Anticorpos Antinúcleo) é positivo?", [""], "", [], [], "exame\_ana")  
 regra(1, "Sim", "SugestãoDiagnóstico", "Artrite Reumatóide", 0.2, ["Sim", "febre", "Sim"])  
 regra(2, "Sim", "SugestãoDiagnóstico", "Lúpus Eritematoso Sistêmico", 0.2, ["Sim", "febre", "Sim"])  
 regra(3, "Sim", "SugestãoDiagnóstico", "Artrite Reumatóide", 0.5, ["Sim", "artralgia", "Sim"])  
 regra(4, "Sim", "SugestãoDiagnóstico", "Lúpus Eritematoso Sistêmico", 0.5, ["Sim", "artralgia", "Sim"])  
 regra(5, "Sim", "SugestãoDiagnóstico", "Artrite Reumatóide", 0.5, ["Sim", "artrite", "Sim"])  
 regra(6, "Sim", "SugestãoDiagnóstico", "Lúpus Eritematoso Sistêmico", 0.5, ["Sim", "artrite", "Sim"])  
 regra(7, "Sim", "SugestãoDiagnóstico", "Artrite Reumatóide", 0.9, ["Sim", "rigidezmatinal", "Sim"])

regra(8,"Sim","SugestãoDiagnóstico","Artrite Reumatóide",0.3,["Sim","mialgia","Sim"])  
regra(9,"Sim","SugestãoDiagnóstico","Lúpus Eritematoso Sistêmico",0.3,["Sim","mialgia","Sim"])  
regra(10,"Sim","SugestãoDiagnóstico","Artrite Reumatóide",0.5,["Sim","nodulo\_subcutaneo","Sim"])  
regra(11,"Sim","SugestãoDiagnóstico","Lúpus Eritematoso Sistêmico",0.9,["Sim","erupcao borboleta","Sim"])  
regra(12,"Sim","SugestãoDiagnóstico","Lúpus Eritematoso Sistêmico",0.5,["Sim","raynaud","Sim"])  
regra(13,"Sim","SugestãoDiagnóstico","Lúpus Eritematoso Sistêmico",0.5,["Sim","fotosensibilidade","Sim"])  
regra(14,"Sim","SugestãoDiagnóstico","Lúpus Eritematoso Sistêmico",0.7,["Sim","alopecia","Sim"])  
regra(15,"Sim","SugestãoDiagnóstico","Artrite Reumatóide",0.2,["Sim","manifestacoesrenais","Sim"])  
regra(16,"Sim","SugestãoDiagnóstico","Lúpus Eritematoso Sistêmico",0.4,["Sim","manifestacoesrenais","Sim"])  
regra(17,"Sim","SugestãoDiagnóstico","Artrite Reumatóide",0.2,["Sim","manifestacoessnc","Sim"])  
regra(18,"Sim","SugestãoDiagnóstico","Lúpus Eritematoso Sistêmico",0.3,["Sim","manifestacoessnc","Sim"])  
regra(19,"Sim","SugestãoDiagnóstico","Artrite Reumatóide",0.3,["Sim","manifestacoespulmonares","Sim"])  
regra(20,"Sim","SugestãoDiagnóstico","Lúpus Eritematoso Sistêmico",0.2,["Sim","manifestacoespulmonares","Sim"])  
regra(21,"Sim","SugestãoDiagnóstico","Artrite Reumatóide",0.9,["Sim","maoreumatoide","Sim"])  
regra(22,"Sim","DiagnósticoExames","Artrite Reumatóide",0.9,["Sim","exame\_fator\_reumatoide","Sim"])  
regra(23,"Sim","DiagnósticoExames","Lúpus Eritematoso Sistêmico",0.9,["Sim","exame\_ana","Sim"])  
explicacao(1,[""])  
explicacao(2,[""])  
explicacao(3,[""])  
explicacao(4,[""])  
explicacao(5,[""])  
explicacao(6,[""])  
explicacao(7,[""])  
explicacao(8,[""])  
explicacao(9,[""])  
explicacao(10,[""])  
explicacao(11,[""])  
explicacao(12,[""])  
explicacao(13,[""])

```

explicacao(14,[""])
explicacao(15,[""])
explicacao(16,[""])
tutorial(1, "")
tutorial(2, "")
tutorial(3, "")
tutorial(4, "")
tutorial(5, "")
tutorial(6, "")
tutorial(7, "")
tutorial(8, "")
tutorial(9, "")
tutorial(10, "")
tutorial(11, "")
tutorial(12, "")
tutorial(13, "")
tutorial(14, "")
tutorial(15, "")
tutorial(16, "")
variavel("febre",[""], "Univalorada", "Linguística")
variavel("artralgia",[""], "Univalorada", "Linguística")
variavel("artrite",[""], "Univalorada", "Linguística")
variavel("rigidezmatinal",[""], "Univalorada", "Linguística")
variavel("mialgia",[""], "Univalorada", "Linguística")
variavel("erupcao borboleta",[""], "Univalorada", "Linguística")
variavel("raynaud",[""], "Univalorada", "Linguística")
variavel("fotosensibilidade",[""], "Univalorada", "Linguística")
variavel("alopecia",[""], "Univalorada", "Linguística")
variavel("manifestacoessnais",[""], "Univalorada", "Linguística")
variavel("manifestacoessnc",[""], "Univalorada", "Linguística")
variavel("manifestacoespulmonares",[""], "Univalorada", "Linguística")
variavel("maoreumatoide",[""], "Univalorada", "Linguística")
variavel("SugestãoDiagnóstico",["Artrite Reumatóide", "Lúpus Eritematoso
Sistêmico"], "Univalorada", "Linguística")
variavel("DiagnósticoExames",["Lúpus Eritematoso Sistêmico", "Artrite
Reumatóide"], "Univalorada", "Linguística")
variavel("exame_fator_reumatoide",[""], "Univalorada", "Linguística")
variavel("exame_ana",[""], "Univalorada", "Linguística")
variavel("nodulo_subcutaneo",[""], "Univalorada", "Linguística")
meta("SugestãoDiagnóstico")
meta("DiagnósticoExames")
historia([])

```

**ANEXO B – Base de conhecimento : Diagnóstico para as  
doenças Artrite Reumatoide e Lupos Erimatoso Sistemico. –  
Expert System**

---

## SOBRE O SISTEMA ESPECIALISTA

- Nome: Diagnostico Diferencial AR x LES
- Autores: Fernando Mendes de Azevedo
- Resumo:

Este sistema foi implementado apenas com a finalidade de servir de referencia para uma comparacao entre uma mesma base de conhecimento em dois sistemas shell diferentes. Portanto nao existe a menor possibilidade de ser considerada correta e/ou completa do

Operador de maior precedência: conjunção

Fator de confiança mínimo para aceitação de regra: 10

As premissas devem ser avaliadas por completo.

---

## VARIÁVEIS

Alopecia

Tipo:

univalorada

ANA (Antinuclear antibodies)

Tipo:

univalorada

Artralgia

Tipo:

univalorada

Artrite

Tipo:

univalorada

Diagnostico Sugestão

Valores:

Lupus Eritematoso Sistemico

Artrite Reumatóide

Tipo:

multivalorada

Diagnóstico Final

Valores:

Artrite Reumatóide

Lupus Eritematoso Sistemico

Tipo:

multivalorada

Erupção Borboleta

Tipo:

univalorada



Febre

Tipo:

multivalorada

Fenomeno de Raynaud

Tipo:

univalorada

Fotosensibilidade

Tipo:

univalorada

Manifestações Pulmonares

Tipo:

univalorada

Manifestações do Sistema Nervoso Central

Tipo:

univalorada

Manifestações Renais

Tipo:

univalorada

Mialgia

Tipo:

univalorada

Mão Reumatóide

Tipo:

univalorada

Nodulos Subcutâneos

Tipo:

univalorada

Rigidez Matinal

Tipo:

univalorada

Teste para Fator Reumatoide

Tipo:

univalorada

## OBJETIVOS

Diagnostico Sugestão

Diagnóstico Final

## REGRAS

Regra 1

SE Febre = Sim

ENTÃO Diagnostico Sugestão = Artrite Reumatóide CNF 20%

Diagnostico Sugestão = Lupus Etitematoso Sistemico CNF 20%

## Regra 2

SE Artralgia = Sim

ENTÃO Diagnostico Sugestão = Artrite Reumatóide CNF 50%

Diagnostico Sugestão = Lupus Etitematoso Sistemico CNF 50%

## Regra 3

SE Artrite = Sim

ENTÃO Diagnostico Sugestão = Artrite Reumatóide CNF 50%

Diagnostico Sugestão = Lupus Etitematoso Sistemico CNF 50%

## Regra 4

SE Rigidez Matinal = Sim

ENTÃO Diagnostico Sugestão = Artrite Reumatóide CNF 90%

## Regra 5

SE Mialgia = Sim

ENTÃO Diagnostico Sugestão = Artrite Reumatóide CNF 30%

Diagnostico Sugestão = Lupus Etitematoso Sistemico CNF 30%

## Regra 6

SE Nodulos Subcutâneos = Sim

ENTÃO Diagnostico Sugestão = Artrite Reumatóide CNF 50%

## Regra 7

SE Erupção Borboleta = Sim

ENTÃO Diagnostico Sugestão = Lupus Etitematoso Sistemico CNF 90%

## Regra 8

SE Fenomeno de Raynaud = Sim

ENTÃO Diagnostico Sugestão = Lupus Etitematoso Sistemico CNF 50%

## Regra 9

SE Fotosensibilidade = Sim

ENTÃO Diagnostico Sugestão = Lupus Etitematoso Sistemico CNF 50%

## Regra 10

SE Alopecia = Sim

ENTÃO Diagnostico Sugestão = Lupus Etitematoso Sistemico CNF 70%

## Regra 11

SE Manifestações Renais = Sim

ENTÃO Diagnostico Sugestão = Artrite Reumatóide CNF 20%

Diagnostico Sugestão = Lupus Etitematoso Sistemico CNF 40%

## Regra 12

SE Manifestações do Sistema Nervoso Central = Sim

ENTÃO Diagnostico Sugestão = Artrite Reumatóide CNF 20%

Diagnostico Sugestão = Lupus Eritematoso Sistemico CNF 30%

## Regra 13

SE Manifestações Pulmonares = Sim

ENTÃO Diagnostico Sugestão = Artrite Reumatóide CNF 30%

Diagnostico Sugestão = Lupus Eritematoso Sistemico CNF 20%

## Regra 14

SE Mão Reumatóide = Sim

ENTÃO Diagnostico Sugestão = Artrite Reumatóide CNF 90%

## Regra 15

SE Teste para Fator Reumatóide = Sim

ENTÃO Diagnóstico Final = Artrite Reumatóide CNF 100%

## Regra 16

SE ANA (Antinuclear antibodies) = Sim

ENTÃO Diagnóstico Final = Lupus Eritematoso Sistemico CNF 100%

## PERGUNTAS

Variável:Alopecia

Pergunta:"Possui alopecia ?"

Variável:ANA (Antinuclear antibodies)

Pergunta:"Teste ANA ?"

Variável:Artralgia

Pergunta:"Possui artralgia ?"

Variável:Artrite

Pergunta:"Possui Artrite ?"

Variável:Erupção Borboleta

Pergunta:"Possui Erupção borboleta ?"

Variável:Febre

Pergunta:"Possui Febre ?"

Variável:Fenomeno de Raynaud

Pergunta:"Possui Fenomeno de Raynaud ?"

Variável:Fotosensibilidade

Pergunta:"Possui Fotosensibilidade?"

Variável:Manifestações Pulmonares

Pergunta:"Possui Manifestacoes Pulmonares ?"

Variável:Manifestações do Sistema Nervoso Central

Pergunta:"Possui manifestacoes do Sistema Nervoso ?"

Variável:Manifestações Renais

Pergunta:"Possui manifestações Renais ?"

Variável:Mialgia

Pergunta:"Possui Mialgia ?"

Variável:Mão Reumatóide

Pergunta:"Possui mão Reumatoide ?"

Variável:Nodulos Subcutâneos

Pergunta:"Possui nodulos subcutaneos ?"

Variável:Rigidez Matinal

Pergunta:"Possui rigidez Matinal ?"

Variável:Teste para Fator Reumatoide

Pergunta:"Teste Fator Reumatoide ?"