

**UNIVERSIDADE FEDERAL DE SANTA CATARINA - UFSC**  
**DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E SISTEMAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**UMA METODOLOGIA DE USO DE TÉCNICAS DE INDUÇÃO PARA  
CRIAÇÃO DE REGRAS DE SISTEMAS ESPECIALISTAS**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia de Produção.

Orientador: Prof. Bruno Hartmut Kopittke, Dr.

**ALEXSANDRA FAISCA NUNES DE OLIVEIRA**

**FLORIANÓPOLIS - SC - BRASIL**

**Abril / 2001**

ALEXSANDRA FAISCA NUNES DE OLIVEIRA

**UMA METODOLOGIA DE USO DE TÉCNICAS DE INDUÇÃO PARA CRIAÇÃO  
DE REGRAS DE SISTEMAS ESPECIALISTAS**

Esta Dissertação foi julgada adequada para obtenção do grau de **Mestre em Engenharia de Produção** - Área de Concentração: Engenharia de Avaliação e Inovação Tecnológica e aprovada em sua forma final pelo **Programa de Pós-Graduação em Engenharia de Produção** da Universidade Federal de Santa Catarina.


Florianópolis, 16 de Abril de 2001.



---

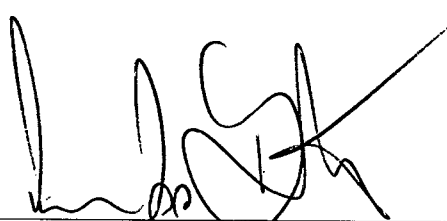
Prof. Ricardo Miranda Barcia, Ph.D  
Coordenador do Programa

Banca Examinadora:



---

Prof. Bruno Hartmut Kopittke, Dr.  
Orientador



---

Armando Luiz Dettmer, Dr.



---

Prof.<sup>a</sup> Silvia Modesto Nassar, Dr.<sup>a</sup>



Dedico este trabalho aos meus futuros filhos,  
que cresçam tendo dentro de si o valor do  
estudo e a importância de se fazer amigos.

## AGRADECIMENTOS

Agradeço a todos que de alguma forma contribuíram para a realização deste trabalho. Em especial agradeço:

A Deus, que nada é impossível, por me dotar de saúde, sabedoria, persistência, e de tornar disponível e ao meu alcance os meios necessários para a realização deste objetivo, além de me ajudar a transpor os obstáculos que surgiram durante esta caminhada.

Aos meus pais, Itamar e Helena, mesmo estando longe, sempre procuraram me incentivar, muitas vezes tomando para eles a minha ansiedade e preocupação.

Ao meu esposo, Rodrigo, pela sua participação e colaboração para a conclusão deste trabalho não me deixando desanimar.

Ao meu orientador, professor Bruno Hartmut Kopittke, pela oportunidade, apoio, paciência e compreensão durante este período.

Ao amigo e colaborador Armando Luiz Dettmer, pela confiança depositada para a realização deste trabalho e pelos inúmeros diálogos e discussões sobre o mesmo, sendo inestimável a sua participação para este se tornar realidade.

A professora Sílvia Modesto Nassar pelas sugestões feitas para a melhoria deste trabalho.

Ao amigo Janderson de Lima Reis, que tornou possível a concretização da ferramenta GARP através das linhas de código em Delphi programadas.

Ao professor Luís M. Ribeiro Vieira que por meio de trocas de mensagens via Internet (através do acesso a página da Instituição<sup>1</sup>) forneceu a referência bibliográfica necessária para o cálculo da probabilidade e confiança das regras, mostrando-se prestativo com a minha solicitação.

A Capes pelo apoio financeiro, possibilitando assim o desenvolvimento deste trabalho.

---

<sup>1</sup> Instituto Superior de Economia e Gestão – ISEG, *home-page* <http://www.iseg.utl.pt/pinicio.html>. Tal instituto faz parte da Universidade Técnica de Lisboa. O professor Luís Vieira, *e-mail*: [lmrv@iseg.utl.pt](mailto:lmrv@iseg.utl.pt), ministra a disciplina de Sistemas de Informação Estratégica, *home-page* <http://pascal.iseg.utl.pt/~sie/ini.htm> no curso de Pós-graduação em Tecnologias e Sistemas de Informação para as Organizações que o ISEG oferece.

## SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>vii</b>
<b>LISTA DE FIGURAS .....</b>	<b>ix</b>
<b>LISTA DE TABELAS.....</b>	<b>xi</b>
<b>RESUMO.....</b>	<b>xii</b>
<b>ABSTRACT .....</b>	<b>xiii</b>
<b>1. INTRODUÇÃO .....</b>	<b>14</b>
<b>1.1 Contextualização .....</b>	<b>14</b>
<b>1.2 Definição do Problema .....</b>	<b>15</b>
<b>1.3 Justificativas .....</b>	<b>16</b>
<b>1.4 Objetivos do Trabalho.....</b>	<b>16</b>
<b>1.5 Limitações do Trabalho .....</b>	<b>17</b>
<b>1.6 Conteúdo e Organização do Trabalho.....</b>	<b>18</b>
<b>2. REVISÃO DA LITERATURA .....</b>	<b>20</b>
<b>2.1 Descoberta de Conhecimento em Base de Dados .....</b>	<b>20</b>
2.1.1 Conceitos de DCBD.....	21
2.1.2 O Processo de DCBD.....	23
2.1.3 Mineração de Dados.....	26
<b>2.2 Indução de Árvores de Decisão.....</b>	<b>27</b>
<b>2.3 Sistema ID3.....</b>	<b>30</b>
2.3.1 ID3: Caso Especial.....	37
<b>2.4 Extração de Regras.....</b>	<b>39</b>
2.4.1 Cálculo da Probabilidade e Confiança .....	40
<b>3. A METODOLOGIA PROPOSTA.....</b>	<b>42</b>
<b>3.1 Contextualização .....</b>	<b>42</b>
3.1.1 Termos AM versus BD.....	43
<b>3.2 As Etapas da Metodologia Proposta.....</b>	<b>43</b>
3.2.1 Origem dos Dados.....	45
3.2.2 Preparação dos Dados .....	47

3.2.3	Geração da Árvore de Decisão .....	52
3.2.4	Extração de Regras e Cálculo de Probabilidades.....	57
<b>4.</b>	<b>A FERRAMENTA GARP .....</b>	<b>60</b>
4.1	Ambiente.....	60
4.2	Dados.....	61
4.3	Funcionamento do Sistema .....	61
4.4	Aplicação da Metodologia Proposta através do GARP.....	67
4.5	Constatações sobre a Aplicação .....	77
4.5.1	Utilizando Dados Fictícios .....	78
4.5.2	Utilizando Dados do GI-EPS.....	85
<b>5.</b>	<b>CONCLUSÃO .....</b>	<b>92</b>
5.1	Conclusões .....	92
5.2	Sugestões para Trabalhos Futuros .....	93
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>94</b>

## LISTA DE ABREVIATURAS E SIGLAS

### — *A* —

**AM** – Aprendizado de Máquina.

### — *B* —

**BD** – Banco de Dados.

### — *C* —

**CLS** – *Concept Learning System*.

**CTC** – Centro Tecnológico.

### — *D* —

**DCBD** – Descoberta de Conhecimento em Base de Dados.

### — *E* —

**EPS** – Engenharia de Produção e Sistemas.

### — *G* —

**GARP** – Gerador Automático de Regras Probabilísticas.

**GI-EPS** – Gestão Industrial – Engenharia de Produção e Sistemas.

### — *I* —

**IA** – Inteligência Artificial.

**ID3** – *Itemized Dichotomizer 3*.

### — *K* —

**KDD** – *Knowledge Discovery in Databases*.

### — *L* —

**LJE** – Laboratório de Jogos de Empresas.

— *M* —

**MB** – *Mega Bytes*.

**MD** – *Mineração de Dados*.

— *P* —

**PPGEP** – *Programa de Pós-Graduação em Engenharia de Produção*.

— *R* —

**RAM** – *Random Access Memory*.

— *S* —

**SAD** – *Sistema de Apoio a Decisão*.

**SC** – *Santa Catarina*.

**SGBD** – *Sistema de Gerenciamento de Bancos de Dados*.

**SPIRIT** – *Symmetrical Probabilistic Intensional Reasoning in Inference Networks in Transition*.

**SQL** – *Structured Query Language*.

— *T* —

**TDIDT** – *Top Down Induction Decision Tree*.

— *U* —

**UFSC** – *Universidade Federal de Santa Catarina*.

## LISTA DE FIGURAS

<b>Figura 2.1 – Os passos do processo de DCBD (Fayyad et al, 1996).....</b>	<b>25</b>
<b>Figura 2.2 - Árvore de decisão para o diagnóstico do motor.....</b>	<b>29</b>
<b>Figura 2.3 – Estratégia de busca do ID3 (Holsheimer e Siebes, 1991) .....</b>	<b>32</b>
<b>Figura 2.4 – Estrutura da árvore de decisão para os objetos em C.....</b>	<b>33</b>
<b>Figura 3.1 - Visão macro da metodologia.....</b>	<b>44</b>
<b>Figura 3.2 – Origem dos Dados (metodologia).....</b>	<b>45</b>
<b>Figura 3.3 – Preparação dos Dados (metodologia) .....</b>	<b>48</b>
<b>Figura 3.4 – Formação da árvore de decisão .....</b>	<b>56</b>
<b>Figura 3.5 – Formação da árvore de decisão: criação de uma folha.....</b>	<b>57</b>
<b>Figura 4.1 – Seqüência de passos (interface - GARP).....</b>	<b>62</b>
<b>Figura 4.2 – Ícone de acesso a ferramenta GARP .....</b>	<b>63</b>
<b>Figura 4.3 – Tela de abertura da ferramenta.....</b>	<b>63</b>
<b>Figura 4.4 – Interface inicial do protótipo GARP .....</b>	<b>64</b>
<b>Figura 4.5 – Salvando um projeto .....</b>	<b>65</b>
<b>Figura 4.6 – Mensagem de aviso opção salvar: arquivo já existente.....</b>	<b>65</b>
<b>Figura 4.7 – Interface da opção imprimir (relatório) .....</b>	<b>66</b>
<b>Figura 4.8 – Acesso a uma base de dados com restrições (usuário / senha).....</b>	<b>68</b>
<b>Figura 4.9 – Seleção da tabela que contém as informações desejadas .....</b>	<b>68</b>
<b>Figura 4.10 – Seleção dos campos que se deseja .....</b>	<b>69</b>
<b>Figura 4.11 – Preparação dos dados: seleção de algum conjunto específico de valores ..</b>	<b>70</b>
<b>Figura 4.12 – Tipos de regras disponíveis para seleção .....</b>	<b>71</b>
<b>Figura 4.13 – Seleção dos dados dos pacientes com idades acima de 45 anos.....</b>	<b>71</b>
<b>Figura 4.14 – Discretização dos atributos (criação de classes).....</b>	<b>72</b>
<b>Figura 4.15 – Mensagem de erro de discretização. ....</b>	<b>73</b>
<b>Figura 4.16 – Escolha do conceito a ser instruído e tamanho inicial da amostra. ....</b>	<b>74</b>
<b>Figura 4.17 – Mensagem de identificação de conflito .....</b>	<b>75</b>
<b>Figura 4.18 – Visualização dos dados de acordo com os passos anteriores.....</b>	<b>75</b>
<b>Figura 4.19 – Árvore de decisão por indução.....</b>	<b>76</b>
<b>Figura 4.20 – Regras e probabilidades retiradas da árvore de decisão .....</b>	<b>77</b>
<b>Figura 4.21 – Objetos tomados para a janela inicial de 80 % (primeira aplicação).....</b>	<b>79</b>
<b>Figura 4.22 – Árvore de decisão por indução gerada (primeira aplicação).....</b>	<b>79</b>

## LISTA DE TABELAS

<b>Tabela 2.1 – Histórico do motor .....</b>	<b>28</b>
<b>Tabela 2.2 – Subconjunto <math>C_1</math> temperatura igual a alta.....</b>	<b>36</b>
<b>Tabela 2.3 – Subconjunto <math>C_2</math> temperatura igual a normal .....</b>	<b>37</b>
<b>Tabela 2.4 – Subconjunto <math>C_3</math> temperatura igual a baixa .....</b>	<b>37</b>
<b>Tabela 2.5 – Probabilidades e confianças associadas a cada regra (motor).....</b>	<b>41</b>
<b>Tabela 3.1 - Termos BD versus AM .....</b>	<b>43</b>
<b>Tabela 3.2 – Histórico de pacientes sobre avaliação do nível de pressão (fictício) .....</b>	<b>49</b>
<b>Tabela 3.3 – Histórico de pacientes sobre avaliação do nível de pressão (fictício) - discretizado ....</b>	<b>51</b>
<b>Tabela 3.4 – Subconjunto <math>C_1</math> gerado da partição no atributo raiz ativfísica (diária).....</b>	<b>55</b>
<b>Tabela 3.5 – Subconjunto <math>C_2</math> gerado da partição no atributo raiz ativfísica (semanal) ..</b>	<b>55</b>
<b>Tabela 3.6 - Subconjunto <math>C_3</math> gerado da partição no atributo raiz ativfísica (ocasional). </b>	<b>56</b>
<b>Tabela 3.7 - Subconjunto <math>C_4</math> gerado da partição no atributo raiz ativfísica (nunca).....</b>	<b>56</b>
<b>Tabela 3.8 – Probabilidades associadas a cada regra (nível de pressão).....</b>	<b>58</b>
<b>Tabela 4.1 – Exemplos de acionamento de greve.....</b>	<b>84</b>
<b>Tabela 4.2 - Exemplos dos dados analisados (baseGIP.dbf).....</b>	<b>85</b>
<b>Tabela 4.3 – Descrição dos dados analisados (baseGIP.dbf) .....</b>	<b>86</b>



## RESUMO

### UMA METODOLOGIA DE USO DE TÉCNICAS DE INDUÇÃO PARA CRIAÇÃO DE REGRAS DE SISTEMAS ESPECIALISTAS

Autora: Alexsandra Faisca Nunes de Oliveira

Orientador: Dr. Bruno Hartmut Kopittke

O presente trabalho relata a necessidade, na era atual, da utilização de sistemas especialistas para auxiliar os tomadores de decisão empresarial, pois a enorme quantidade de informações encontradas nas bases de dados das empresas torna a análise destas sem a ajuda da informática inviável, uma vez que a capacidade de inferência humana é limitada. Além da precisão da extração de conhecimento novo útil a partir das informações contidas nestas bases de dados de forma automática computacionalmente, pois tal conhecimento pode ser utilizado para a formação da base de conhecimento de um sistema especialista. Proceceu-se então, a busca na literatura para a realização desta tarefa, encontrando a área de descoberta de conhecimento em base de dados como orientação para tal, a qual propõe a aplicação de algoritmos de mineração de dados, além de atividades de pré-processamento dos dados e pós-processamento do conhecimento descoberto, entre outras. Dentre os algoritmos de mineração de dados encontrados destaca-se o ID3, o qual realiza a aprendizagem supervisionada a partir de exemplos, representando o conhecimento descoberto através de árvore de decisão. Fazendo a “leitura” da árvore pode-se representar este conhecimento na forma de regras e com parte do embasamento teórico de regras de associação calcular o suporte (probabilidade) e a confiança de cada regra. Assim, este estudo possibilitou a criação de uma metodologia de uso de técnicas de indução para criação de regras de sistemas especialistas. Tal metodologia conduziu a criação de um protótipo de software, denominado GARP, que proporciona a geração automática de regras probabilísticas podendo ser usadas em qualquer *shell* de sistemas especialistas baseada em regras. Para validação desta metodologia, o protótipo criado foi submetido a testes utilizando-se bases de dados fictícios como também, uma aplicação real do jogo de empresas GI-EPS. Por fim, são apresentadas algumas constatações referentes a aplicação desta metodologia em relação ao algoritmo de mineração de dados utilizado, o ID3.

**Palavras-chave:** sistema especialista, mineração de dados, ID3, jogo de empresas.

## ABSTRACT

### A METHODOLOGY OF USE OF TECHNIQUES OF INDUCTION FOR CREATION OF RULES OF EXPERT SYSTEMS

Author: Alexsandra Faisca Nunes de Oliveira

Adviser: Dr. Bruno Hartmut Kopittke

The present work tells the need, in the current era, of the use of expert systems to aid the maker of managerial decision, because the enormous amount of information found in the databases of the business turns the analysis of these without the help of the unviable computer science, once the capacity of human inference is limited. Besides the precision of the extraction of useful new knowledge starting from the information contained in these databases in an automatic way through the computer, because such knowledge can be used for the formation of the base of knowledge of a expert system. It was proceeded then, the search in the literature for the accomplishment of this task, finding the area of knowledge discovery in database as orientation for such, which proposes the application of algorithms of data mining, besides activities of pre-processing of the data and post-processing of the discovered knowledge, among another. Among the algorithms of data mining of found stand out the ID3, which it accomplishes the learning supervised starting from examples, representing the knowledge discovered through decision tree. Making the "reading" of the tree this knowledge can be represented in the form of rules and with part of the theoretical foundation of association rules to calculate the support (probability) and the confidence of each rule. Thus, this study turned possible the creation of a methodology of use of induction techniques for creation of rules of expert systems. Such methodology drove the creation of a software prototype, denominated GARP, that provides the automatic generation of rules probabilistic could be used in any shell of expert systems based on rules. For validation of this methodology, the prototype servant was submitted it you test being used bases of fictitious data as well as, a real application of the game of business GI-EPS. Finally, some referring verifications are presented the application of this methodology in relation to the used algorithm of data mining, the ID3.

**Key-words:** expert system, data mining, ID3, game of business.

## 1. INTRODUÇÃO

---

*Nesta introdução ao presente trabalho, que relata uma metodologia de uso de técnicas de indução para a criação de regras de sistemas especialistas, apresenta-se uma visão geral da essência deste, bem como, de forma mais detalhada: a definição do problema, justificativas, objetivos, limitações e por fim, descreve-se sucintamente o seu conteúdo e a sua organização.*

---

### 1.1 Contextualização

---

Atualmente, tomar decisões nas diversas atividades empresariais está relacionado diretamente com a capacidade de extrair conhecimento de dados, de fatos concretos.

A capacidade de inferência humana é limitada - devido a grandes quantidades de dados o processo de inferência se torna moroso, incerto e até inconclusivo - e é influenciada pela forma como o homem interpreta os fatos a ele apresentados ou mesmo pela forma que estão sendo apresentados. Por outro lado, aliado ao homem está o computador, máquina capaz de realizar o processamento de grandes quantidades de dados com maior rapidez, segurança e exatidão inigualáveis ao ser humano.

Dada sua limitação, o comportamento normal do ser humano é bloquear, inconscientemente, o excesso de informação, tratando de apenas alguns aspectos dos fatos (Torres, 1995).

Com a rápida evolução e mudanças tecnológicas, massificou-se o acesso às novas tecnologias de hardware, softwares, técnicas de Inteligência Artificial, redes de computadores, Internet, e o processamento informatizado nas diversas áreas empresariais tanto em nível de produção de bens de consumo como o de prestação de serviços, provocando ao longo dos

anos, o surgimento de grandes volumes de dados com históricos completos sobre clientes, produtos e transações para as empresas.

Assim, ressalta-se a importância de como coletar, armazenar e processar tais informações computacionalmente, de forma que venham a auxiliar os administradores nos processos de tomadas de decisões empresarias, como também, buscar novos padrões a fim de gerar conhecimento novo através destas informações.

Atualmente, as funções operacionais, gerenciais e de planejamento estratégico de determinadas empresas, visando alcançarem seus objetivos e metas, buscam o suporte de sistemas de informações baseados em computador em suas tomadas de decisões, principalmente nos sistemas especialistas do tipo SAD – Sistema de Apoio a Decisão, uma vez que a economia atual dá importância ao capital intelectual, e o seu incremento, pode ser dado a partir da utilização destes sistemas especialistas.

É neste contexto que se insere a presente dissertação, propondo uma metodologia para análise de dados e construção de padrões na forma de regras e probabilidades, de modo que estas novas informações possam auxiliar o especialista do domínio (pessoa que detém o conhecimento) em sua tomada de decisão, ou seja, na formação da base de conhecimento de um sistema especialista do tipo SAD.

## **1.2 Definição do Problema**

---

Em virtude da grande quantidade de variáveis envolvidas em determinados processos de tomada de decisão, humanamente é difícil fazer a distinção de quais variáveis e valores interferem de forma direta ou indireta nestes. Como também, algumas vezes, o especialista do domínio não consegue explicitar, ou até mesmo justificar, o seu conhecimento de forma clara o suficiente para colocá-lo na forma de regras para o auxílio na tomada de decisão e conseqüentemente criação do sistema especialista do tipo SAD, fazendo com que o subjetivismo impere no processo de tomada de decisão podendo gerar conclusões incorretas sobre este.

## 1.3 Justificativas

---

Em um processo de tomada de decisão quando existir uma base de dados bastante grande, de modo que o especialista do domínio encontre dificuldades em analisá-los, ou até mesmo, tomar uma decisão sem contemplar todos os fatos envolvidos no processo em análise, fazendo com que não haja um conhecimento completo de como cada variável do processo se interage com as demais, como também, o seu grau de importância dentro do processo decisório, surge a necessidade de que tal base de eventos seja analisada com o auxílio da informática.

O problema de pesquisa se mostra importante, uma vez que existe um reconhecimento explícito na literatura das aplicações para as empresas atuais e para a ciência da descoberta de conhecimento em base de dados, ora referenciada simplesmente como mineração de dados (*data mining*), ora relacionada diretamente com a aprendizagem a partir de exemplos. Isso ocorre devido a DCBD (descoberta de conhecimento em base de dados) utilizar entre outras coisas, técnicas de aprendizado de máquina, área da inteligência artificial, e de conceitos estatísticos para lidar com a incerteza relacionada às descobertas. Essa afirmação se fundamenta nas referências bibliográficas deste trabalho, (Bispo, 1999), (Feldens, 1997), (Gilleron, 2000), (Mannila, 1996), (Nimer, 1998) (IT Mídia, 2000), (Fayyad, Haussler & Stolorz, 1996) entre outras.

## 1.4 Objetivos do Trabalho

---

O objetivo geral deste trabalho concentra-se no desenvolvimento de uma ferramenta computacional capaz de gerar conhecimento na forma de regras mais a probabilidade associada a cada regra, através de indução de árvores de decisão, a partir de massas de dados acessíveis ao computador e, tendo como perspectivas futuras o ambiente para testes, aplicações, e validação da metodologia proposta o Jogo de Empresas: GI-EPS – Gestão Industrial da Engenharia de Produção e Sistemas.

A partir do exposto até então, pode-se listar os seguintes objetivos específicos:

- ↳ Estruturar, modelar e desenvolver uma ferramenta computacional capaz de gerar conhecimento automatizado na forma de regras probabilísticas a partir de uma base de

dados ou simplesmente a partir de exemplos, onde estes por sua vez, encontram-se armazenados em uma tabela. Tal ferramenta será denominada GARP (Gerador Automático de Regras Probabilísticas).

- ↪ Implementar o GARP com uma interface amigável, proporcionando de forma autocontida, uma fácil compreensão aos usuários de sua utilização e de todos os passos que este abrange.
- ↪ Construir o GARP de forma que o conhecimento gerado possa ser utilizado em qualquer shell de sistemas especialistas em que haja compatibilidade com a base de conhecimentos (regra e probabilidade) gerada pela ferramenta.
- ↪ Mostrar evidências de que o sistema desenvolvido poderá ser aplicado e validado no ambiente do Jogo de Empresas GI-EPS.

## 1.5 Limitações do Trabalho

---

O trabalho, que ora se apresenta, tem destaque para as seguintes limitações:

- A ferramenta GARP está implementada nesta primeira versão de forma sucinta, isto é, estão implementadas as funções consideradas fundamentais do processo de descoberta de conhecimento de base de dados em si, para proporcionar a aplicabilidade da mesma. Assim, funções não menos importantes mas que não inviabilizam a sua utilização, nesse primeiro momento, não estão implementadas, tais como: a geração de relatórios, a possibilidade de se trabalhar com mais de uma tabela de dados no processo de descoberta, o menu ajuda; além de algumas características que poderiam ser ampliadas no uso do algoritmo ID3 como: possibilidade de trabalhar com ruído (conflito) nos dados e dados incompletos, e implementar o processo de classificação para mais de duas classes para o conceito aprendido. Nesta versão o GARP está detectando conflito nos dados e informando ao usuário. Já com os dados incompletos, este trabalha como se fossem valores nulos e o processo de classificação está sendo realizado apenas com duas classes para o conceito. Estas e outras considerações de modelagem, funcionamento e implementação da ferramenta encontram-se disponíveis no capítulo 3 e 4.
- O processo de discretização de dados contínuos para a criação de classes, no GARP, é realizado pelo usuário, ou seja, este não ocorre no sistema de forma automatizada por opção de deixar liberdade ao usuário nesta tarefa.

- Apesar da possibilidade de utilização da base de conhecimentos gerada pelo GARP por qualquer shell de sistema especialista compatível com a mesma, nenhum método computacional foi analisado e implementado de modo a permitir a importação automatizada da base de conhecimentos de uma ferramenta à outra.
- Neste primeiro momento, não se justifica a realização de testes de desempenho e otimização dos algoritmos implementados para a ferramenta GARP.

Portanto, a ferramenta desenvolvida GARP constitui um protótipo, ou seja, está aberta a incorporação de novas funções.

## 1.6 Conteúdo e Organização do Trabalho

---

O presente trabalho está organizado em cinco capítulos, de forma que a seqüência da disposição das informações nesse roteiro possa oferecer um melhor entendimento de seu conteúdo. Sendo assim, a estrutura do trabalho apresenta-se da seguinte forma:

- Capítulo 1- Introdução: expõe uma visão geral do contexto em que este trabalho se insere, enfatizando a capacidade de inferência humana, a tomada de decisões empresariais, a era tecnológica atual que proporciona as empresas grandes capacidades de armazenamentos de informações e a extração de conhecimento novo útil a partir destas informações. São apresentados, em seguida, o problema de pesquisa, as justificativas para solucioná-lo, os objetivos do trabalho, e suas limitações.
- Capítulo 2 – Revisão da Literatura: um apanhado na literatura, referindo-se ao embasamento teórico sobre o qual está fundamentada a pesquisa, abordando o tema relativo à descoberta de conhecimento em base de dados, explorando seu conceito e processo, a técnica de indução de árvores de decisão e o sistema ID3, além da extração de regras da árvore de indução e os cálculos de probabilidade (suporte) e confiança de cada regra baseados na técnica de regras de associação.
- Capítulo 3 – A Metodologia Proposta: uma descrição detalhada de todos os passos que a metodologia envolve, além de, inicialmente, ser enfatizado o contexto em que esta se insere.
- Capítulo 4 – A Ferramenta GARP: é realizada a aplicação da ferramenta GARP em base de dados fictícios e em uma base de dados de uma aplicação real do jogo de empresas GI-

EPS, são apresentadas todas as interfaces da ferramenta e suas funções, além de algumas constatações sobre a sua aplicação.

- **Capítulo 5 – Conclusão:** é reservado para as considerações finais, abordando os resultados do trabalho em função dos objetivos propostos, as conclusões e as recomendações para trabalhos futuros.



## 2. REVISÃO DA LITERATURA

---

*Neste capítulo é elaborada a revisão bibliográfica dos conteúdos relativos a descoberta de conhecimento em base de dados, além dos relativos a indução de árvores de decisão, focalizando principalmente a utilização de árvores de decisão para representação do conhecimento elucidado através do sistema ID3, bem como, a extração de regras da árvore de decisão e os cálculos de suporte e confiança de cada regra.*

---

### 2.1 Descoberta de Conhecimento em Base de Dados

---

Em um *workshop* realizado em Detroit na data de 20 de agosto de 1989, o termo descoberta de conhecimento em base de dados, também conhecido pela sigla KDD (*Knowledge Discovery Database*), foi criado para enfatizar que o conhecimento é o produto final da descoberta de dados dirigidos. Sendo este popularizado na área de inteligência artificial e aprendizado de máquina (Fayyad et al, 1996).

A tarefa de encontrar padrões úteis em dados recebeu uma variedade de nomes dentre eles tem-se: mineração de dados (*data mining*), extração de conhecimento, descoberta de informação, arqueologia de dados, entre outros. O termo mineração de dados ganhou popularidade no campo de banco de dados, pois este tem sido usado principalmente por estatísticos, analista de dados e pela comunidade de sistemas de informações gerenciais (ibid).

De acordo com os mesmos autores, a descoberta de conhecimento em base de dados, (DCBD), refere-se a todo o processo de descoberta de conhecimento útil a partir dos dados, enquanto mineração de dados refere-se em particular a uma parte deste processo, pois este é a aplicação de algoritmos específicos para a extração de padrões de dados.

O relatório do *workshop*, (Piatetsky-Shapiro, 1991), relata que a descoberta de conhecimento em base de dados utiliza-se de vários campos incluindo sistemas especialistas, aprendizado de máquina (*machine learning*), banco de dados inteligente, aquisição de conhecimento, raciocínio baseado em casos e estatística.

## 2.1.1 Conceitos de DCBD

---

Wüthrich (1996) expõe o que é descoberta de conhecimento e por que em base de dados, dizendo que, a descoberta de conhecimento denota o processo de gerar automaticamente informações representadas de forma “compreensíveis” ao ser humano a partir de exemplos contidos em uma base de dados. Por informação compreensível entende-se como sendo declarações gerais sobre as características do objeto em análise, por exemplo:

- O objeto em análise é um pássaro:  
 $\text{é\_animal}(x), \text{tem\_asas}(x), \text{pode\_voar}(x) \rightarrow \text{é\_pássaro}(x)$
- O objeto em análise é a competitividade de uma empresa:  
 $\text{é\_empresa}(x), \text{gerenciada\_inteligentemente}(x) \rightarrow \text{é\_competitiva}(x)$

Segundo o mesmo autor, para que o processo de descoberta automática aconteça, geralmente é necessário que se tenha:

- Um objetivo, que é geralmente representado por um atributo da base de dados. Por exemplo, um atributo que indique que uma empresa é competitiva, de forma que o sistema poderia descobrir regras que descrevam uma empresa competitiva;
- Um conjunto de treinamento que, são as informações representadas na forma de atributos e seus respectivos valores em uma base de dados. De acordo com o exemplo anterior, estas informações seriam sobre o grau de competitividade de diversas empresas em questão.

Ainda, de acordo com Wüthrich (1996) esta abordagem é:

- Simbólica: porque a linguagem de representação do conhecimento ou informação é simbólica e não numérica (apesar de mais tarde também poderem ser adicionados componentes numéricos como probabilidades ou pesos).

- Supervisionada: uma vez que é dado um objetivo.
- Orientada a base de dados: uma vez que os exemplos estão armazenados em uma base de dados e a linguagem de representação pode ser linguagens de consultas de sistemas de banco de dados.
- Extensível: no sentido que em adição para descoberta de regras, engenheiros de conhecimento ou administradores de banco de dados podem adicionar outras regras manualmente e, desta forma, complementar o conhecimento gerado automaticamente.

Já para Fayyad et al (1996), a descoberta de conhecimento em base de dados “é o processo não trivial de identificação válida, nova, potencialmente útil e fundamentalmente de padrões compreensíveis nos dados”.

Desta definição os autores explicam que, dados são um conjunto de fatos, isto é, casos na base de dados, e padrão é uma expressão em alguma linguagem descrevendo um subconjunto dos dados ou um modelo aplicável para este subconjunto. Assim como, extração de um padrão também designa um modelo apropriado de dados, ou em geral uma descrição de alto nível do conjunto de dados. O termo processo implica que DCBD é composta por muitos passos, todos repetidos em múltiplas iterações. O termo não trivial significa que alguma técnica de busca ou inferência é envolvida, isto é, o processo não é uma computação direta de quantidades pré-definidas como calcular a média de valores. Os padrões descobertos devem ser válidos com algum grau de certeza. Também, que os padrões devem ser novos (pelo menos para o sistema, e preferencialmente para o usuário) e potencialmente úteis, isto é, levar algum benefício para o usuário / tarefa. Finalmente, os padrões devem ser compreensíveis, se não imediatamente então depois de algum pós-processamento.

Frawley et al (1992), definem a descoberta de conhecimento como a extração não trivial implícita, previamente desconhecida, e potencialmente útil de informações a partir de dados. E dado um conjunto de fatos (dados)  $F$ , uma linguagem  $L$ , e alguma medida de certeza  $C$ , é definido *padrão* como uma declaração  $S$  em  $L$  que descreve relacionamentos entre um subconjunto  $F_S$  de  $F$  com uma certeza  $c$ , tal que  $S$  é o simplificador (em algum sentido) da enumeração de todos os fatos em  $F_S$ . Um padrão que é interessante (de acordo com alguma medida de interesse imposta pelo usuário) e certeza suficiente (de novo de acordo com algum critério do usuário) é chamado conhecimento. A saída de um programa que monitora um conjunto de fatos em uma base de dados e produz padrões neste sentido, acima, é descoberta de conhecimento.

Os mesmos autores relatam que a descoberta de conhecimento em base de dados exhibe quatro características principais, as quais resumidamente são:

- Linguagem de alto nível: descoberta de conhecimento é representada em uma linguagem de alto nível. Ela não precisa ser diretamente usada por humanos, mas sua expressão deve ser compreensível pelos mesmos.
- Acurácia (precisão): descobertas retratam precisamente os conteúdos da base de dados. O tamanho o qual esta retratabilidade é imperfeita é expressa por medidas de certeza.
- Resultados interessantes: descoberta de conhecimento é interessante de acordo com as tendências definidas pelo usuário. Em particular, ser interessante implica que os padrões são novos e potencialmente úteis, e o processo de descoberta é não trivial.
- Eficiência: o processo de descoberta é eficiente. Tempo de execução para grandes bases de dados é previsível e aceitável.

Os sistemas e técnicas descritos em descoberta de conhecimento em base de dados geralmente esforçam-se para satisfazer estas características. Entretanto, as abordagens utilizadas são bastante diversas. A maior parte é baseada em métodos de aprendizado de máquina que têm sido intensificados para melhor resolverem problemas particulares de descoberta em base de dados (ibid).

## 2.1.2 O Processo de DCBD

---

Segundo Fayyad et al (1996), o processo de descoberta de conhecimento em base de dados é interativo e iterativo, envolvendo vários passos sequenciais, com muitas decisões sendo tomadas pelo usuário, podendo a qualquer passo retornar a passos anteriores buscando novos resultados. Resumidamente e de forma ampla o processo envolve:

- ❶ Compreensão do domínio da aplicação e identificação do objetivo do processo de DCBD.

- ② Criação de um conjunto de dados alvos: seleção do conjunto de dados nos quais a descoberta será executada.
- ③ Pré-processamento e limpeza dos dados: operações básicas como remoção de ruído se apropriado, coletar as informações necessárias para o modelo, tratar campos de dados ausentes.
- ④ Projeção e redução dos dados: encontrar características úteis para representar os dados dependendo do objetivo da tarefa. Utilização de métodos de redução ou transformação para reduzir o número efetivo de variáveis.
- ⑤ Combinar o objetivo do processo de DCBD com um método particular de data mining, isto é, sumarização, classificação, regressão, clusterização etc.
- ⑥ Escolha do algoritmo de data mining: selecionar o método (ou os métodos) a ser utilizado para a busca de padrões nos dados.
- ⑦ Aplicação do algoritmo de data mining: busca por padrões interessantes representados em uma forma em particular ou num conjunto de representações como: regras ou árvores de classificação, regressão, clusterização, etc.
- ⑧ Interpretação dos padrões minerados: possibilidade de retornar para qualquer um dos passos anteriores para iterações adicionais. E este passo pode também envolver a forma de visualização dos padrões / modelos extraídos.
- ⑨ Consolidação do conhecimento descoberto: incorporar este conhecimento em outro sistema para ações adicionais, ou simplesmente realizar sua documentação relatando partes interessantes. Também inclui detecção e resolução de conflitos com o conhecimento prévio do próprio usuário (especialista do domínio) ou do extraído.

A figura 2.1 apresenta um resumo dos passos que compreendem o processo de DCBD.

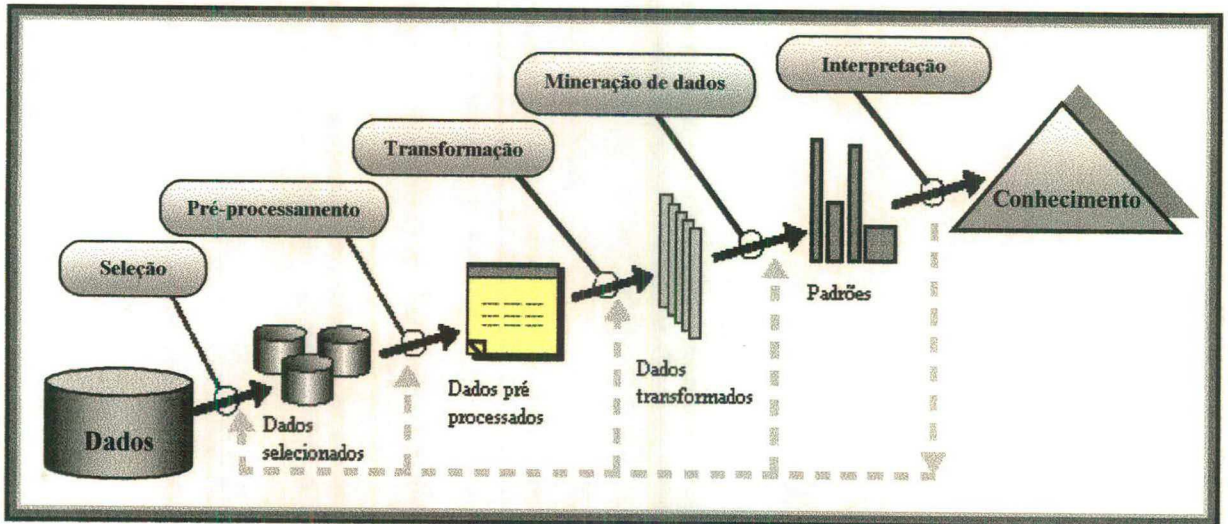


Figura 2.1 – Os passos do processo de DCBD (Fayyad et al, 1996).

Variantes das etapas do processo de DCBD são encontradas em Mannila (1996) e Feldens (1997), porém apenas com algumas suprimissões e/ou aglutinações destas etapas ou sinônimos de nomenclatura.

Assim, pode-se dizer que como a DCBD consiste da descoberta de conhecimento útil em dados, basicamente o seu processo envolve várias etapas: desde a compreensão do domínio da aplicação pelo usuário, a identificação do objetivo do processo, a seleção dos dados para a descoberta, a preparação dos dados (eliminação de ruído, limpeza de erros, lidar com dados ausentes), a transformação dos dados (criação de classes e/ou discretização de atributos quantitativos), a aplicação de algoritmos de mineração de dados (extração de padrões nos dados), até a interpretação ou avaliação dos padrões descobertos (visualização, ordenação por algum critério, criação de relatórios, validação do conhecimento descoberto através de algum método estatístico ou por um especialista). Todas estas etapas são realizadas pelo usuário interagindo com o sistema DCBD, podendo para algumas existir métodos automatizados ou não. No entanto, caso em algum momento o usuário perceba que os padrões gerados não estão de acordo com o seu conhecimento a priori, ou que existem conflitos, ou a necessidade de se testar a geração de novos padrões, este por sua vez, pode recorrer a etapas anteriores do processo retomando a sua execução, caracterizando assim, a interatividade e a iteratividade do processo.



## 2.1.3 Mineração de Dados

---

Todas as etapas (passos) do processo de descoberta de conhecimento em base de dados são importantes para o sucesso de sistemas de DCBD na prática, no entanto, encontra-se na literatura maior atenção ao componente de mineração de dados (Fayyad et al, 1996).

A mineração de dados é um passo no processo de DCBD consistindo da aplicação de algoritmos de descoberta e análise de dados que sob limitações aceitáveis de eficiência computacional, produz uma enumeração particular de padrões sobre os dados (ibid).

Os algoritmos utilizados em mineração de dados estão diretamente relacionados com as técnicas / métodos que estes implementam, além do tipo de conhecimento que se pretende minerar.

De acordo com Chen et al. (1996), diferentes critérios podem ser usados para classificar os sistemas de mineração de dados, tais como:

- Tipos de base de dados: os sistemas de mineração de dados podem ser classificados de acordo com o tipo de base de dados em que estão sendo executados. Por exemplo, um sistema é considerado um minerador de dados relacional se o conhecimento descoberto provir de uma base de dados relacional, ou um minerador orientado a objetos se executado sobre uma base de dados orientada a objetos.
- Tipos de conhecimento: vários tipos de conhecimento podem ser descobertos pelos sistemas de mineração de dados, incluindo regras de associação, regras de classificação, clusterização (agrupamentos), entre outros. Feldens (1997), aborda os tipos de conhecimento descoberto acrescentando: correlação, dependência (regra curta), descrição de conceitos, detecção de seqüências, detecção de desvios e regressão.
- Tipos de técnicas: a escolha da técnica a ser utilizada está diretamente relacionada com o tipo de conhecimento que se pretende minerar. Para um tipo de conhecimento em particular têm-se diferentes abordagens, tais como, aprendizado de máquina, estatística, e orientada a grandes bases de dados, além da integração destas. Essas são comparadas e utilizadas com ênfase nas questões de base de dados como eficiência e escalabilidade computacional.

Dentre as técnicas encontradas na literatura pode-se destacar: regras de associação, árvores de decisão, indução de árvores de decisão, redes neurais artificiais, regras de indução, lógica difusa, entre outras.

Holsheimer et al. (1991), discute sobre alguns algoritmos / sistemas de mineração de dados, dentre eles: ID3, AQ15, CN2, DBLearn, Meta-Dendral, RADIX / RX, Bacon e KEDS.

Contudo no presente trabalho serão abordadas somente as técnicas utilizadas, indução de árvores de decisão e regras de associação, sendo esta última apenas referenciada com o objetivo de calcular as probabilidades associadas às regras extraídas da árvore de decisão. Como também, apresentado em detalhes parte do sistema ID3.

## 2.2 Indução de Árvores de Decisão

---

Segundo Durkin (1991), indução é o processo de raciocínio de um dado conjunto de fatos para princípios gerais ou regras. Indução é preciosa se existem exemplos para criar um processo decisório padrão.

Tomada a seguinte tarefa de indução: determinar o diagnóstico de um motor de fábrica em uma linha de produção, ou seja, dizer o que caracteriza um motor bom ou ruim. Para tal, necessita-se de exemplos, casos passados, que determinam o diagnóstico do motor. Esses exemplos são observações, um histórico, sobre o estado do motor em relação a algumas características do ambiente (linha de produção e o estado do motor) que influenciam no seu estado. A tabela 2.1 representa o histórico do motor.



**Tabela 2.1** – Histórico do motor

VELOCIDADE DA LINHA	IDADE	TEMPERATURA	MOTOR
baixa	velho	alta	ruim
baixa	velho	normal	ruim
normal	novo	normal	bom
normal	velho	alta	ruim
alta	velho	alta	ruim
alta	novo	normal	bom
normal	novo	normal	bom
baixa	novo	alta	ruim
baixa	novo	baixa	ruim

(Fonte: Durkin, 1991)

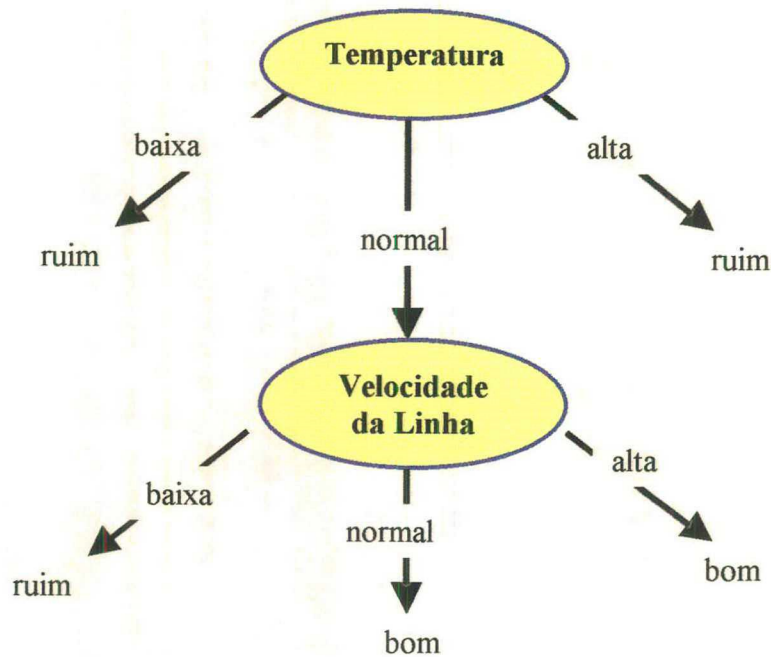
De acordo com a tabela 2.1, cada linha da tabela corresponde a um exemplo de um estado do motor, como também chamado de um objeto. Cada objeto ou cada exemplo está representado em termos de seus atributos – velocidade da linha, idade do motor, temperatura do motor e o estado do motor, com seus respectivos valores associados. O atributo motor como é o objetivo da tarefa de indução, proposta inicialmente, é também chamado de classe ou conceito que será aprendido pelo processo, assim como, de acordo com os seus valores, observa-se existirem duas classes: “classe bom” e “classe ruim”. O conjunto destes objetos (tabela 2.1) é chamado de conjunto de treinamento.

De acordo com Quinlan (1985), a tarefa de indução é desenvolver regras de classificação que podem determinar a classe de qualquer objeto através dos valores de seus atributos. Os objetos são descritos em termos de uma coleção de atributos. Cada atributo mede alguma característica importante do objeto. Cada objeto no universo (domínio da aplicação) pertence a um de um conjunto de classes mutuamente exclusivas e a classe de qualquer objeto do conjunto de treinamento é conhecida. Se o conjunto de treinamento contém dois objetos que têm valores idênticos para cada atributo e pertencerem a classes diferentes, isto é, claramente impossível à diferenciação entre estes objetos com referência somente em seus dados atributos. E neste caso, os atributos serão considerados inadequados para o conjunto de treinamento e conseqüentemente para a tarefa de indução.

Segundo Thompson (1986), um conflito ocorre quando dois exemplos contêm valores idênticos para todos os seus atributos, mas valores de classes diferentes. Um conflito normalmente significa que os atributos escolhidos são inadequados para a tarefa de classificação. Pode-se remover este problema introduzindo atributos adicionais, o que é uma tarefa para o especialista do domínio.

Indução de árvores de decisão consiste destas regras de classificação serem expressas através de uma árvore de decisão, sendo a representação em árvore de decisão equivalente às regras de classificação.

A partir do conjunto de treinamento da tabela 2.1, a tarefa de indução proposta inicialmente pode ser concluída com a árvore de decisão da figura 2.2, pois esta classifica corretamente cada objeto do conjunto de treinamento.



**Figura 2.2** - Árvore de decisão para o diagnóstico do motor

As folhas da árvore de decisão são os nomes de classe, os nós representam testes baseados nos atributos com ramos rotulados, com os possíveis valores do atributo, para um resultado de classificação. Para classificar um objeto, começa-se da raiz da árvore, avalia-se o teste, ou seja, o nó da árvore é comparado com o respectivo atributo do objeto em questão, partindo pelo ramo determinado pelo valor do atributo do objeto que se pretende classificar, e o processo continua até que uma folha seja encontrada, na qual o objeto é afirmado a pertencer à classe nomeada pela folha.

Quinlan (1985) afirma que, em uma tarefa de indução, se os atributos são adequados, ou seja, não existirem conflitos, sempre é possível construir uma árvore de decisão que corretamente classifica cada objeto do conjunto de treinamento, podendo existir



várias (muitas) árvores corretas para um mesmo conjunto de treinamento, porém em uma quantidade finita de árvores.

A essência da indução é mover além do conjunto de treinamento, isto é, construir uma árvore de decisão que corretamente classifica não só objetos do conjunto de treinamento, mas outros (não vistos) objetos, sendo que para fazer isso, a árvore de decisão tem que capturar alguma relação significativa entre a classe de um objeto e os valores de seus atributos (ibid).

Árvores de complexidades menores tendem a classificar corretamente um número maior de objetos, fora do conjunto de treinamento. Sendo o ID3 um sistema projetado para formar árvores de decisão simples, no entanto, não pode garantir que forma a árvore de decisão mais simples (ibid).

## 2.3 Sistema ID3

---

O sistema ID3 (*Itemized Dichotomizer 3*) foi desenvolvido por Quinlan em 1979, é um de uma série de programas projetados a partir do seu precursor, o sistema CLS (*Concept Learning System*) de Hunt. São sistemas de aprendizagem para tarefas de classificação a partir de exemplos. Tendo como característica em comum a representação do conhecimento adquirido como árvores de decisão, onde estas são construídas começando da sua raiz e procedendo até suas folhas (Quinlan, 1985).

A estrutura básica do ID3 é iterativa. Um subconjunto do conjunto de treinamento chamado “janela” é ao acaso escolhido (randomicamente) e uma árvore de decisão a partir deste é formada; esta árvore classifica corretamente todos os objetos da janela. Todos os outros objetos no conjunto de treinamento são então classificados usando a árvore. Se a árvore dá resposta correta para todos estes objetos então esta está correta para o conjunto de treinamento inteiro e o processo termina. Se não, uma seleção dos objetos incorretamente classificados é adicionada a janela e o processo continua, gerando uma nova árvore. A evidência empírica sugere que uma árvore de decisão correta normalmente seja encontrada mais depressa por este método iterativo, uso de uma janela, que pela formação de uma árvore diretamente de todo conjunto de treinamento (ibid).

Mas, como formar uma árvore de decisão para uma coleção arbitrária  $C$  de objetos? Para tal, o algoritmo ID3 baseia-se no algoritmo CLS como é mostrado a seguir, de acordo com Durkin (1991):

☒ **Algoritmo CLS:**

Tem-se:  $C$  = um conjunto de objetos (chamado conjunto de treinamento).

$v$  = total de valores possíveis distintos do atributo  $A$ .

$A$  = um atributo sendo o possível raiz (um teste).

Objetivo: classificar os objetos em duas classes:  $P$  (positiva) ou  $N$  (negativa).

**INÍCIO**

1. **SE** todos os exemplos em  $C$  são positivos **ENTÃO**

Cria um nó  $P$  e para.

**SE NÃO**

**SE** todos os exemplos em  $C$  são negativos **ENTÃO**

Cria um nó  $N$  e para.

**SE NÃO**

Selecione (usando um critério heurístico) um atributo  $A$ , com valores  $A_1, A_2, \dots, A_v$  e cria um nó de decisão (teste / raiz).

2. Divida os exemplos do conjunto de treinamento  $C$  em subconjuntos  $C_1, C_2, \dots, C_v$  de acordo com os valores de  $A$ .
3. Aplique o algoritmo recursivamente para cada um dos conjuntos  $C_i$ .

**FIM.**

☒ **Algoritmo ID3:**

Tem-se  $W$  = um subconjunto do conjunto de treinamento, chamado janela.

**INÍCIO**

1. Selecione randomicamente um subconjunto de tamanho  $w$  do conjunto inteiro (completo) dos exemplos de treinamento,  $C$ .
2. Aplique o algoritmo **CLS** para formar a árvore de decisão para a janela.
3. Verifique se a árvore de decisão classifica corretamente os demais objetos do conjunto de treinamento, procurando os objetos incorretamente classificados.

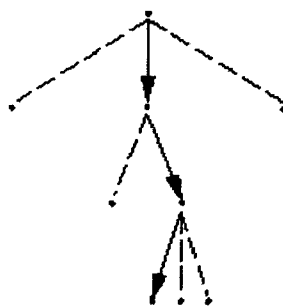
4. **SE** existem alguns objetos incorretamente classificados **ENTÃO**

insira alguns deles dentro do conjunto janela e repita a partir do **passo 2**.

**SE NÃO** pare e mostre a última árvore de decisão.

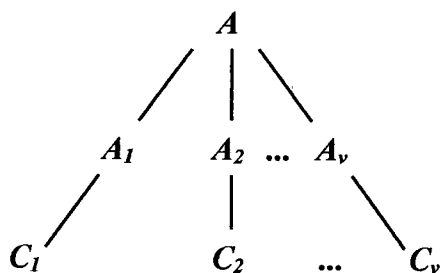
**FIM.**

O algoritmo CLS é uma sub-rotina do algoritmo ID3. O algoritmo ID3 para a formação da árvore de decisão trabalha em profundidade, com uma estratégia de busca *top-down* e *hill climber*, (Holsheimer e Siebes, 1991), como é mostrado na figura 2.3, além de utilizar-se da idéia básica “dividir e conquistar” (Quinlan, 1985).



**Figura 2.3** – Estratégia de busca do ID3 (Holsheimer e Siebes, 1991)

Assim, para a formação da árvore de decisão o conjunto de treinamento  $C$  é dividido em subconjuntos  $C_i$  onde esta divisão é feita de acordo com o atributo  $A$ , escolhido para raiz da árvore, e seus possíveis valores distintos do atributo, ou seja, cada  $C_i$  conterà apenas os objetos de  $C$  com valores  $A_i$  de  $A$  e conseqüentemente cada  $C_i$  será menor que  $C$ . Sendo que, a partir do atributo raiz  $A$  partem os ramos rotulados com os possíveis valores distintos de  $A$  para cada subconjunto  $C_i$ , isso é representado graficamente na figura 2.4. A idéia básica é “dividir e conquistar”, isto é, proceder com esta escolha da raiz e conseqüentemente com a divisão do conjunto em subconjuntos para cada  $C_i$ , um por vez (estratégia de busca *top-down* e *hill climber*), até encontrar uma folha para este. O resultado final será uma árvore para  $C$ , pois esta estratégia renderá subconjuntos até que satisfaçam a exigência de uma classe para uma folha, ou seja, até que nestes subconjuntos tenham objetos pertencentes apenas a uma única classe, logo uma folha é encontrada.



**Figura 2.4** – Estrutura da árvore de decisão para os objetos em  $C$

De acordo com Quinlan (1985), como o sistema ID3 foi projetado para construir árvores de decisão simples, a escolha do teste (atributo  $A$ ) é crucial, sendo que o ID3 adota uma heurística baseada na informação que depende de duas suposições. Assumindo que  $C$  contém  $p$  objetos de classe  $P$  e  $n$  objetos de classe  $N$ , em uma tarefa de indução de duas classes, as suposições são:

- (1) Qualquer árvore de decisão correta para  $C$  classificará objetos na mesma proporção como a representação deles em  $C$ . Um objeto arbitrário será determinado para pertencer à classe  $P$  com probabilidade de  $p/(p+n)$  e para pertencer à classe  $N$  com probabilidade de  $n/(p+n)$ ;
- (2) Quando uma árvore de decisão é usada para classificar um objeto, retorna uma classe. Assim, uma árvore de decisão pode ser considerada como uma fonte de uma mensagem “ $P$ ” ou “ $N$ ”, com a informação esperada necessária para gerar essa mensagem dada por:

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n} \quad (1)$$

A primeira suposição de Quinlan trata da frequência de ocorrência de cada classe no conjunto de treinamento  $C$ , ou seja, a probabilidade de um objeto pertencer à classe  $P$  e a probabilidade de um objeto pertencer a classe  $N$ .

A segunda suposição de Quinlan se refere à entropia de classificação para o conjunto de treinamento  $C$ , ou seja, a quantia de incerteza de informação contida neste conjunto com relação a esse determinar qual classe (à classe  $P$  ou à classe  $N$ ) que pertence um determinado objeto.

O ID3 utiliza-se da entropia, medida da quantia de incerteza de informação sobre a classificação contida no conjunto de treinamento  $C$ , como também, a contida em cada subconjunto de  $C$ ,  $C_i$ , dada a escolha do atributo  $A$  para particionar  $C$ . Isto é calculado para

determinar se tal atributo  $A$  escolhido levará a uma subárvore de maior ganho de informação, ou seja, menor incerteza na classificação dos objetos realizada por esta subárvore. Como é mostrado a seguir:

De acordo com Quinlan (1985), se o atributo  $A$  com valores  $\{A_1, A_2, \dots, A_v\}$  é usado para a raiz da árvore de decisão, particionará  $C$  em subconjuntos  $\{C_1, C_2, \dots, C_v\}$  onde  $C_i$  contém objetos de  $C$  que tem valor  $A_i$  de  $A$ . Sendo que  $C_i$  contém  $p_i$  objetos de classe  $P$  e  $n_i$  objetos de classe  $N$ . A informação esperada requerida para a subárvore para  $C_i$  é  $I(p_i, n_i)$ . A informação esperada requerida para a árvore com o atributo  $A$  como raiz é então obtida como a média ponderada:

$$E(A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} I(p_i, n_i) \quad (2)$$

Onde: o peso para o  $i$ -ésimo ramo é a proporção dos objetos em  $C$  que pertencem a  $C_i$ , ou seja,  $(p_i + n_i) / (p + n)$ .

Em outras palavras, da equação (2), o  $I(p_i, n_i)$  é a entropia de classificação para cada subárvore formada pela escolha do atributo  $A$  como raiz, ou seja, para cada subconjunto  $C_i$ . E o  $E(A)$  é a entropia de classificação para a árvore formada pela escolha do atributo  $A$  como raiz, ou seja, é a soma das entropias de cada subárvore multiplicada pela probabilidade (quer dizer, a frequência de ocorrência) dos objetos de  $C_i$  em  $C$ . Note que a figura 2.4 representa a estrutura da árvore de decisão para qual é calculado o  $E(A)$ .

O ganho de informação pela ramificação em  $A$  é então calculado (Quinlan, 1985):

$$Gain(A) = I(p, n) - E(A) \quad (3)$$

A equação (3) expressa que o ganho de informação pela ramificação em  $A$  é igual a entropia de classificação contida em todo conjunto de treinamento  $C$ , menos a entropia de classificação contida na árvore formada pela partição do conjunto  $C$  no atributo  $A$ .

De acordo com Quinlan (1985), uma boa regra prática parece ser escolher aquele atributo para ramificar a árvore em que o ganho de informação é maior. Desde que  $I(p, n)$  é constante para todos os atributos, maximizando o ganho é equivalente a minimizar  $E(A)$ . Assim, o ID3 examina todos os atributos candidatos e escolhe  $A$  para maximizar o  $Gain(A)$ , forma a árvore como mencionado anteriormente, e então usa este mesmo processo recursivamente para formar árvores de decisão para os subconjuntos residuais  $C_1, C_2, \dots, C_v$ .

Observe que os atributos candidatos a serem a possível raiz, são todos os atributos do conjunto de treinamento  $C$ , exceto o atributo objetivo, ou seja, o conceito que será aprendido pelo sistema, já que este será as folhas da árvore, rotuladas com a sua classe.

Calcular-se-á a seguir a árvore de decisão para a tarefa de indução proposta no início da seção 2.2 Indução de Árvores de Decisão de acordo com as suposições de Quinlan, apresentadas anteriormente.

Sendo  $C$  o conjunto de objetos da tabela 2.1, com um total de 9 objetos, destes 3 são da classe bom e 6 são da classe ruim. Assim, a informação requerida para a classificação é:

$$I(p, n) = -\frac{3}{9} \log_2 \frac{3}{9} - \frac{6}{9} \log_2 \frac{6}{9} = 0,9183 \text{ bits}$$

Agora considerando o atributo temperatura como possível raiz com os seus valores {alta, normal, baixa}, tem-se o cálculo da informação requerida para a classificação nestes subconjuntos  $C_i$  gerados pela partição no atributo temperatura:

- Para  $C_1$ : 4 dos 9 objetos em  $C$  tem temperatura igual ao primeiro valor (alta), sendo todos eles da classe ruim, então:

$$p_1 = 0 \quad n_1 = 4 \quad I(p_1, n_1) = 0 \text{ bits}$$

- Para  $C_2$ : 4 dos 9 objetos em  $C$  tem temperatura igual ao segundo valor (normal), sendo 3 deles da classe bom e 1 deles da classe ruim, então:

$$p_2 = 3 \quad n_2 = 1 \quad I(p_2, n_2) = 0,8113 \text{ bits}$$

- Para  $C_3$ : 1 dos 9 objetos em  $C$  tem temperatura igual ao terceiro valor (baixa), sendo este da classe ruim, então:

$$p_3 = 0 \quad n_3 = 1 \quad I(p_3, n_3) = 0 \text{ bits}$$

A informação esperada requerida para a árvore tendo este atributo (temperatura) como raiz é:

$$E(\text{temperatura}) = \frac{4}{9} I(p_1, n_1) + \frac{4}{9} I(p_2, n_2) + \frac{1}{9} I(p_3, n_3) = 0,3605 \text{ bits}$$



O ganho de informação pela ramificação neste atributo temperatura é:

$$Gain(temperatura) = 0,9183 - 0,3605 = 0,5577$$

Análise similar faz-se para calcular os ganhos de informação para os demais atributos da tabela 2.1, assim tem-se:

$$Gain(velocidade\ da\ linha) = 0,3900\ bits$$

$$Gain(idade) = 0,3789\ bits$$

Desta forma o método usado no ID3 de formação da árvore de decisão através do atributo de maior ganho de informação escolhe temperatura como o atributo para a raiz da árvore de decisão. Os objetos do conjunto de treinamento  $C$  são divididos então em subconjuntos de acordo com os valores do atributo temperatura, resultando em três subconjuntos os quais são apresentados respectivamente nas tabelas 2.2, 2.3 e 2.4. E uma árvore de decisão para cada subconjunto é induzida de um modo similar. No entanto, tomando o subconjunto  $C_1$  observa-se que todos os objetos contidos neste subconjunto pertencem somente a uma classe, classe ruim, logo o processo termina para este subconjunto e uma folha é encontrada rotulada com o nome da classe. Desta mesma forma ocorre com o subconjunto  $C_3$ . Já o subconjunto  $C_2$  contém objetos das duas classes então o processo se repete para este de modo similar ao apresentado anteriormente.

A figura 2.2 (pg. 29) mostra a árvore de decisão final gerada pelo ID3 para este conjunto de treinamento. Observe que o atributo idade não aparece na árvore de decisão. Isto se deve ao fato do ID3 ter considerado este atributo irrelevante para esta tarefa de classificação.

Tabela 2.2 – Subconjunto  $C_1$  temperatura igual a alta

VELOCIDADE DA LINHA	IDADE	MOTOR
baixa	velho	ruim
normal	velho	ruim
alta	velho	ruim
baixa	novo	ruim

**Tabela 2.3** – Subconjunto  $C_2$  temperatura igual a normal

VELOCIDADE DA LINHA	IDADE	MOTOR
baixa	velho	ruim
normal	novos	bom
alta	novos	bom
normal	novos	bom

**Tabela 2.4** – Subconjunto  $C_3$  temperatura igual a baixa

VELOCIDADE DA LINHA	IDADE	MOTOR
baixa	novos	ruim

Observação: uma vez escolhido um atributo para raiz da árvore este não aparecerá mais no caminho da raiz até uma folha. Por isso, não há a necessidade do atributo temperatura aparecer nos subconjuntos. Tal afirmação é reforçada matematicamente, pois tomando o segundo subconjunto  $C_2$ , no momento de se calcular  $I(p,n)$  para este tem-se  $p=3$  e  $n=1$ , logo  $I(p,n) = 0,8113 \text{ bits}$  e calculando o  $E(A)$  para o atributo temperatura obtém-se  $E(\text{temperatura}) = 0,8113 \text{ bits}$ . Assim o ganho de informação deste atributo para este subconjunto  $C_2$  é nulo.

### 2.3.1 ID3: Caso Especial

---

Segundo Quinlan (1985), um caso especial surge se  $C$  não contém nenhum objeto com algum particular valor  $A_i$  de  $A$  dando um  $C_i$  vazio. O ID3 rotula tal folha como “nulo” de forma que isto não classifica qualquer objeto que chega até aquela folha, ou seja, ocorre falha na classificação. Uma solução melhor generalizaria do conjunto  $C$  do qual  $C_i$  veio, e nomeia para esta folha a classe mais frequente em  $C$ .

No entanto, em ambas as soluções propostas acima, o ID3 gera a regra equivalente ao caso com probabilidade zero e com confiança zero, ou seja, o processo gera uma regra onde não existe pelo menos um objeto no conjunto de treinamento que satisfaça essa regra. Isso foi constatado nas aplicações realizadas através do GARP (ver capítulo 4 pg. 83 e 84).

De modo a esclarecer melhor este caso especial é apresentado a seguir o que este representa em termos de cálculos realizados pelo ID3.

Para cada atributo, possível raiz, ao calcular a informação esperada para cada valor de atributo, sendo  $n_i = 0$  ou  $p_i = 0$ , então  $I(p_i, n_i) = 0$ , desta forma tem-se:

Se  $(p_i = 0)$  ou  $(n_i = 0)$  então

$$I(p_i, n_i) := 0$$

Se não

$$I(p_i, n_i) := -\frac{p_i}{p_i + n_i} \log_2 \frac{p_i}{p_i + n_i} - \frac{n_i}{p_i + n_i} \log_2 \frac{n_i}{p_i + n_i}$$

Fim se;

O caso especial surge quando o atributo de maior ganho, ou seja, a raiz escolhida é, por exemplo, um atributo  $A$  com valores  $\{A_1, A_2, \dots, A_v\}$  onde algum  $p_i + n_i = 0$  deste atributo, resultando num  $C_i$  vazio, então se aplica a seguinte solução:

Tomados os valores de  $p$  e de  $n$  do conjunto do qual  $C_i$  veio, tem-se:

Se  $n > p$  então

$$\text{Folha} := N$$

Se não

Se  $n = p$  então

$$\text{Folha} := \text{rand}(\text{dos possíveis valores de classe})^*$$

Se não

$$\text{Folha} := P$$

Fim se;

Fim se;

---

\* Entende-se como  $\text{rand}(x)$  uma função que retorna um valor randômico do conjunto  $x$ . Neste caso como são duas classes um valor  $P$  ou  $N$ .

## 2.4 Extração de Regras

---

Segundo Thompson e Thompson (1986), embora as árvores mostrem os relacionamentos que existem entre os vários atributos, elas podem ser muito difíceis de serem manipuladas. Uma estrutura que pode representar a informação de forma semelhante, mas mais fácil de ser usada é a regra.

De acordo com Durkin (1991), da árvore de decisão da figura 2.2 (pg. 29) pode-se extrair as seguintes regras:

- ✓ *Se temperatura = baixa ou temperatura = alta então motor = ruim*
- ✓ *Se temperatura = normal e velocidade da linha = baixa então motor ruim*
- ✓ *Se temperatura = normal e velocidade da linha = normal ou velocidade da linha = alta então motor = bom*

Assim, observa-se que o conhecimento representado a partir de uma árvore de decisão pode ser extraído sob a forma de regras. A premissa ou condição da regra (lado esquerdo) corresponde aos atributos, exceto o atributo de classe, com seus respectivos valores associados enquanto a conclusão da regra ao conceito que é instruído ou ao nó objetivo (atributo de classe com seus valores). A retirada das regras da árvore de decisão é realizada a partir da raiz seguindo por um ramo até encontrar um nó folha. Quando valores diferentes de um mesmo atributo levarem a folhas de mesmo rótulo, ou seja, mesmo valor de classe, usa-se um conectivo *ou*. Quando por um ramo para chegar a uma folha encontrar-se um outro atributo usa-se o conectivo *e*.

No entanto, pode-se criar regras apenas com o conectivo *e*, sendo que desta forma, o número total de regras corresponderá ao número total de nós folha da árvore de decisão. De acordo com a árvore de decisão da figura 2.2 (pg. 29) têm-se as seguintes regras:

- ✓ *Se temperatura = baixa então motor = ruim*
- ✓ *Se temperatura = normal e velocidade da linha = baixa então motor ruim*
- ✓ *Se temperatura = alta então motor = ruim*
- ✓ *Se temperatura = normal e velocidade da linha = normal então motor = bom*
- ✓ *Se temperatura = normal e velocidade da linha = alta então motor = bom*

## 2.4.1 Cálculo da Probabilidade e Confiança

---

Para calcular a probabilidade de uma regra, aplicam-se os conceitos de *suporte* e *confiança* apresentados por Agrawal et al. (1993), em regras de associação, sendo estes estendidos para as regras geradas para o conjunto de treinamento  $C$ .

Assim, como uma regra gerada do conjunto de treinamento  $C$  é uma implicação do tipo  $A \Rightarrow B$ , onde  $A \subset C$ ,  $B \subset C$  e  $A \cap B = \emptyset$ , o *suporte* e a *confiança* desta regra são então calculados como:

$$\text{Suporte} = \frac{AB}{(p+n)} \quad (4)$$

$$\text{Confiança} = \frac{AB}{A} \quad (5)$$

onde:

- $AB$  = total de objetos (registros) que verificam a condição e a conclusão da regra do conjunto de treinamento  $C$ .
- $(p+n)$  = total de objetos (registros) do conjunto de treinamento  $C$ .
- $A$  = total de objetos (registros) que verificam a condição da regra.

Desta forma, o *suporte* de uma regra corresponde a sua probabilidade dentro do conjunto de treinamento  $C$ , ou seja, a proporção dos objetos em  $C$  onde a condição e a conclusão da regra se verificam. A *confiança* de uma regra é a medida da força da regra, ou seja, a sua validade estatística, sendo calculada como a proporção dos objetos do conjunto de treinamento  $C$  que satisfazem a condição e a conclusão da regra dividido pela proporção dos objetos em  $C$  que satisfazem a condição da regra. Assim, a *confiança* é um valor maior ou igual a zero e menor ou igual a 1.

Caso todos os objetos com a característica  $A$  pertençam à classe  $B$ , a *confiança* da regra será igual a 1 ou 100%.

Para exemplificar os conceitos apresentados, são mostradas na tabela 2.5 a probabilidade (suporte) e a confiança associadas a cada regra obtida da árvore de decisão do diagnóstico do motor de fábrica (fig. 2.2):

Tabela 2.5 – Probabilidades e confianças associadas a cada regra (motor)

Regra	Suporte	Confiança
Se temperatura = baixa então motor = ruim	11,11%	100%
Se temperatura = normal e velocidade da linha = baixa então motor ruim	11,11%	100%
Se temperatura = alta então motor = ruim	44,44%	100%
Se temperatura = normal e velocidade da linha = normal então motor = bom	22,22%	100%
Se temperatura = normal e velocidade da linha = alta então motor = bom	11,11%	100%

## 3. A METODOLOGIA PROPOSTA

---

*Neste capítulo apresenta-se a metodologia proposta de uso de técnicas de indução para criação de regras de sistemas especialista, enfatizando o contexto em que esta se insere e descrevendo detalhadamente todos os passos que a envolvem.*

---

### 3.1 Contextualização

---

De acordo com a definição do problema relatada no capítulo 1 item 1.2 e com a revisão bibliográfica apresentada no capítulo 2, propõe-se uma metodologia de uso de técnicas de indução para criação de regras de sistemas especialistas. Esta metodologia se insere no contexto de DCBD, pois utiliza-se de recursos de inteligência artificial, técnicas de aprendizado de máquina, estatística e banco de dados.

O fato de se estar buscando uma forma automatizada computacionalmente de criar regras para sistemas especialistas toma-se como referência a base teórica da inteligência artificial, uma vez que esta utiliza o computador como ferramenta capaz de solucionar problemas que somente pessoas especialistas no assunto (domínio) resolveriam.

De acordo com Levine (1988), “a inteligência artificial (IA) é simplesmente uma maneira de fazer o computador pensar inteligentemente”, sendo que esta se preocupa com a forma de aquisição, armazenamento, representação e processamento do conhecimento humano através do computador.

A utilização de aprendizado de máquina (AM), área da inteligência artificial (IA), a qual estuda processos de aprendizagem que podem ser realizados através do computador, (Holsheimer e Siebes, 1991), faz-se presente devido à busca de aquisição de conhecimento (regras mais probabilidades) através de aprendizagem por indução. Para tal, encontrou-se na literatura o sistema ID3, o qual a partir de um conjunto de exemplos cria uma árvore de

decisão por indução, e há a possibilidade de se passar estas informações representadas na forma de árvore para regras.

A estatística se faz presente devido ao cálculo das probabilidades associadas a cada regra em relação às informações contidas na base de dados.

Além de recursos de banco de dados, já que as informações necessárias para a criação da base de conhecimento estão armazenadas em uma base de dados.

### 3.1.1 Termos AM versus BD

---

Em virtude do relacionamento da metodologia proposta, para o uso de técnicas de indução para criação de regras de sistemas especialistas, com aprendizado de máquina (AM) e com banco de dados (BD), modelo relacional, surgem alguns termos, de ambas as áreas, referenciados durante a mesma que se tornam equivalentes dentro deste contexto. A tabela 3.1 lista estas equivalências, informalmente, de acordo com cada área.

**Tabela 3.1 - Termos BD versus AM**

<b>BANCO DE DADOS (BD)</b>	<b>APRENDIZADO DE MÁQUINA (AM)</b>
Tabela, dados relacionais	Conjunto de exemplos, conjunto de treinamento (também organizados em uma tabela)
Campo, atributo (coluna)	Característica, atributo
Valor do campo	Valor do atributo
Registro, tupla (linha)	Objeto, exemplo
Domínio do campo	Possíveis valores do atributo

## 3.2 As Etapas da Metodologia Proposta

---

Nesta seção parte-se a descrever a metodologia proposta passo a passo. São apresentados todos os passos que a envolvem. Uma visão macro desta metodologia é mostrada na figura 3.1, onde o processo inicia com a origem dos dados, escolha do usuário pelos dados que contém as informações pertinentes ao domínio do sistema especialista, logo



após, há a preparação destes dados, para então a aplicação do algoritmo de mineração, criando por indução uma árvore de decisão e, a partir desta, são retiradas as regras e calculadas as probabilidades. Logo, o processo é iterativo.



**Figura 3.1** - Visão macro da metodologia

A metodologia propõe-se a apoiar a construção de sistemas especialistas, possibilitando que, quando houver uma base de casos disponível, parte da base de conhecimento seja adquirida automaticamente na forma de regras a partir destes. A base de casos trata-se de objetos (exemplos) representados por um conjunto de propriedades ou atributos com os seus respectivos valores associados. O domínio da aplicação do sistema é de propósito geral, no entanto, todas as aplicações envolvem classificação. Para mais detalhes sobre a tarefa de classificação ver capítulo 2 item 2.2 Indução de Árvores de Decisão.

### 3.2.1 Origem dos Dados

---

Esta é a primeira fase da metodologia proposta e é muito importante. Nesta etapa o usuário irá selecionar os dados que ele deseja utilizar para a descoberta. Os dados, a nível de descoberta, são considerados como sendo dos tipos: numérico - para informação quantitativa, caracter - para informação qualitativa e data - para informação temporal. No entanto, estas considerações ficam transparentes ao usuário quando este utiliza o GARP. As atividades realizadas pelo usuário, ilustradas na figura 3.2, nesta primeira etapa do processo são:

- ❶ Seleção da base de dados;
- ❷ Seleção da tabela;
- ❸ Seleção dos campos da tabela (todos ou alguns);

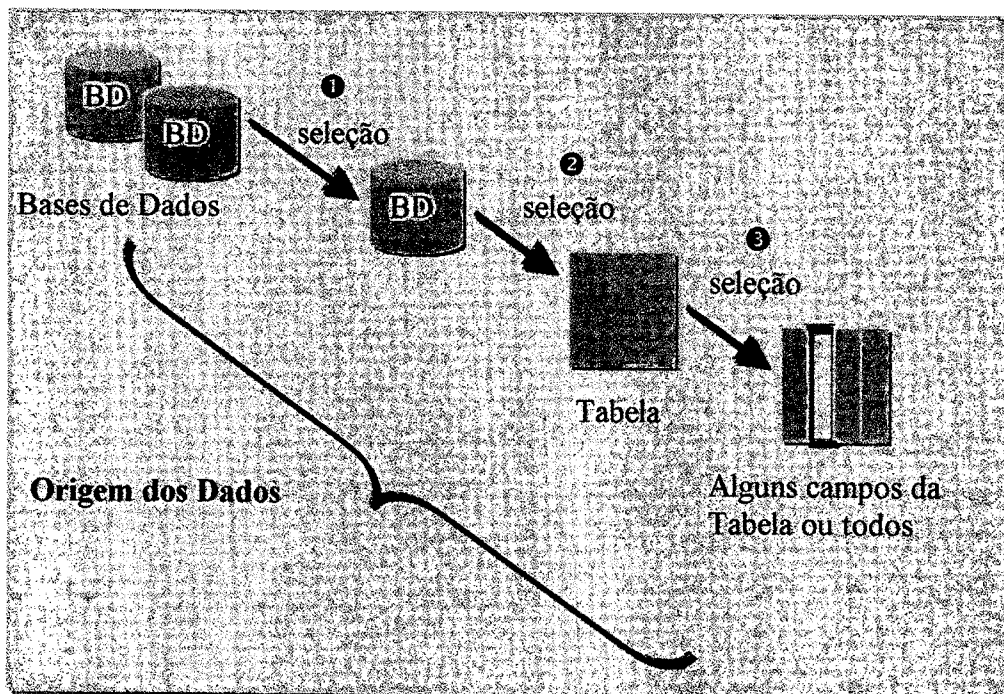


Figura 3.2 – Origem dos Dados (metodologia)

Neste processo de aprendizagem por indução a qualidade dos dados é muito importante, sendo essa vista de duas formas: primeira que os dados tenham um verdadeiro significado para a descoberta ou para a classificação que se queira fazer, segunda que estes dados sejam confiáveis. Pois, é a partir deles que o sistema de mineração chegará a uma generalização sobre as informações extraídas da base de dados.

O algoritmo utilizado para a mineração de dados, o ID3, é capaz de diagnosticar atributos irrelevantes e os descartar para a criação da árvore. Isto é um ponto muito positivo do algoritmo, pois pode reduzir a quantidade de atributos (variáveis) utilizados no sistema especialista. Mas, não exclui a necessidade de, no mínimo, o usuário optar por dados que sejam do domínio que ele deseja trabalhar no sistema especialista. Conclui-se então que devido a esta característica do algoritmo parece mais produtivo escolher atributos a mais do que a menos.

No entanto, é bom salientar que com a adição de novas informações (atributos), o atributo que antes destas mostrou-se irrelevante, pode vir a se tornar importante para a tarefa de classificação, (Durkin, 1991).

Segundo Quinlan (1985), o poder dos sistemas da família TDIDT (*Top Down Induction Tree*), sendo que o ID3 faz parte desta família, vem de um banco de dados existente que forma um histórico de observações, como os registros dos históricos de pacientes em alguma área médica de um centro de diagnóstico. Objetos deste tipo dão um quadro estatístico de confiança, mas desde que eles não sejam de qualquer forma organizados, eles podem ser redundantes ou podem omitir casos incomuns que não foram encontrados durante o período de manutenção do registro.

Dois autores, Genaro (1986) e Rabuske (1995), citam a mesma frase que Feigenbaum disse em uma conferência sobre IA, em 1977: “a potência de um sistema especialista deriva do conhecimento que ele possui e não de formalismos e esquemas específicos que ele emprega”.

Desta forma, já que o conhecimento para o futuro sistema especialista irá partir destas informações da base de dados, pelo menos parte deste, pois nada impede que o usuário acrescente alguns fatos e / ou regras a base de conhecimento, é de suma importância a qualidade destas e que no mínimo o usuário saiba relacionar o domínio que ele deseja trabalhar no sistema especialista com alguma base de dados e conseqüentemente com uma tabela da base.

Por outro lado, a metodologia proposta também permite que os exemplos utilizados, representados na forma de tabela, sejam um conjunto cuidadosamente selecionado de exemplos preparados por um especialista do domínio, cada um com alguma relevância particular para uma regra de classificação completa e correta. Em alguns casos, poderia ser trabalhoso para o especialista evitar redundância e incluir exemplos de casos raros, (Quinlan, 1985).

Voltando para as atividades realizadas pelo usuário nesta etapa do processo, citadas no início desta seção, e considerando o exposto anteriormente, tem-se as seguintes considerações a serem feitas em cada atividade:

A escolha da base de dados é trivial, basta o usuário ter em mente o domínio da aplicação que ele deseja trabalhar e selecionar a respectiva base de dados. A seleção da tabela, e após, os seus campos, parte do pressuposto que se está trabalhando com banco de dados relacional e, portanto as informações encontram-se armazenadas em tabelas (linhas e colunas de dados). As linhas representam os registros (conjuntos de informações sobre elementos distintos) e as colunas representam os campos (atributos específicos de um registro). Ver tabela 3.1, para maiores esclarecimentos das equivalências dos termos utilizados durante a metodologia.

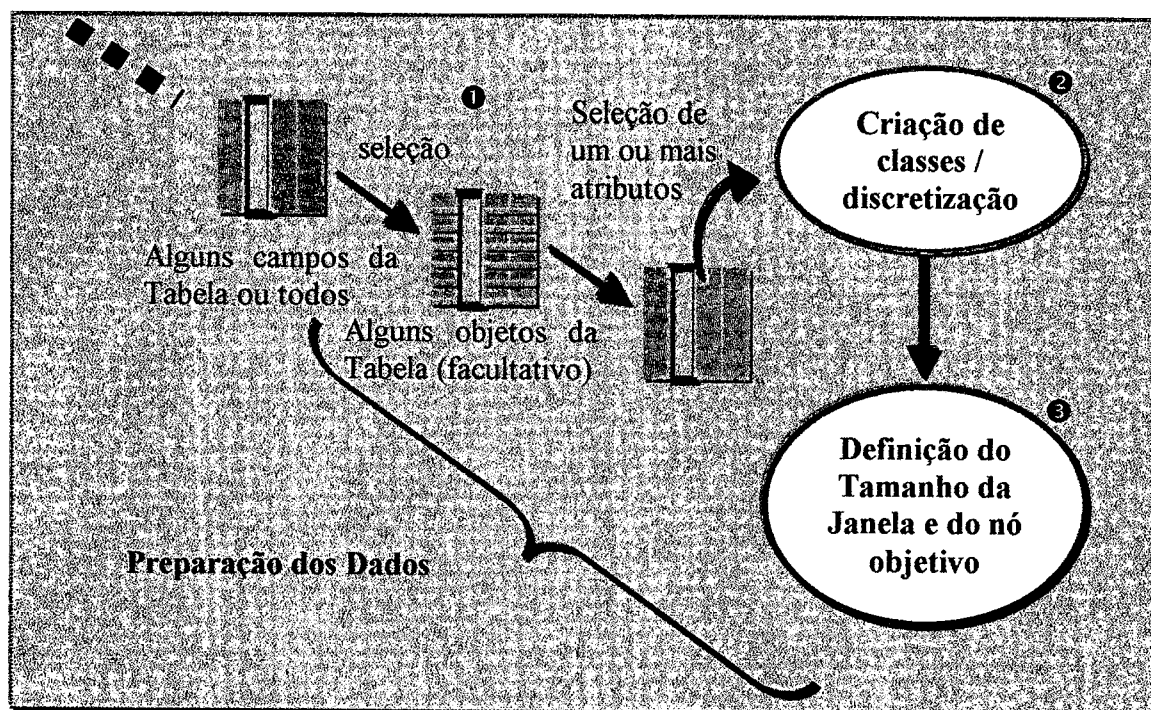
Assim, a seleção dos campos da tabela corresponde ao usuário informar com quais colunas ou atributos ele fará o processo de busca, tendo em mente qual o conceito que o sistema irá aprender e através de quais atributos pretende-se fazer esta aprendizagem. Maiores detalhes sobre aprendizagem e conceito, ver capítulo 2 item 2.2 Indução de Árvores de Decisão e nesta seção item 3.2.2 Preparação dos Dados – escolha do nó objetivo.

### **3.2.2 Preparação dos Dados**

---

Definida a origem dos dados, na fase anterior, esta compreende a preparação dos dados a serem utilizados no processo de aprendizagem. A preparação envolve três etapas distintas, citadas abaixo e ilustradas pela figura 3.3:

- ❶ Seleção de algum conjunto de registros (facultativo);
- ❷ Discretização dos dados ou definição de classes;
- ❸ Definição do tamanho da Janela e escolha do nó objetivo.



**Figura 3.3 – Preparação dos Dados (metodologia)**

Dentre as etapas, citadas anteriormente, a seleção de algum conjunto específico de registros, parte do pressuposto que se está trabalhando com banco de dados relacional e desta forma, a seleção de algum conjunto de registros destes está se referindo a escolha de algumas linhas da tabela e não se utilizar todas. Esta seleção, de algumas linhas da tabela, objetiva algum interesse do usuário em alguns objetos em particular. Por exemplo, dentro de um domínio médico, sobre fatores que influenciam o nível de pressão arterial do paciente, poderia selecionar apenas os pacientes acima de 40 anos, não importando os pacientes de outra faixa etária. Esta etapa, portanto, é facultativa.

Para a realização da seleção, exemplificada acima, note que o atributo a ser utilizado é o atributo *IDADE*, e a regra a ser usada deste com seus valores é *MAIOR QUE 40*, (ver possíveis exemplos de dados, fictícios, na tabela 3.2).

Desta forma, a metodologia fornece algumas regras possíveis de serem utilizadas pelo usuário, disponibilizando também os valores distintos de cada atributo, de modo que fique transparente para o mesmo a utilização de comandos SQL para a realização destas. A construção de uma seleção de exemplos é realizada respeitando o seguinte modelo: escolha do <ATRIBUTO> e escolha da <REGRA e VALOR ASSOCIADO>. As regras possíveis são:

- Igual a;
- Diferente de;
- Entre;

- Maior ou igual a;
- Maior que;
- Menor ou igual a;
- Menor que.

A segunda etapa desta fase de preparação dos dados corresponde a discretização destes ou a criação de classes. A discretização corresponde a ação de transformar valores contínuos de atributos em valores discretos. Por exemplo, voltando ao exemplo médico do nível de pressão arterial, poder-se-ia ter um atributo denominado IDADE, com todas as idades dos pacientes do histórico hospitalar. Para melhor utilizar-se do processo de classificação para a criação das regras trabalhar-se-ia com faixas etárias em vez de números contínuos. Observe o exemplo fictício abaixo da tabela 3.2:

**Tabela 3.2** – Histórico de pacientes sobre avaliação do nível de pressão (fictício)

SAL	GORDURA	FUMANTE	BEBIDA	ATIVFÍSICA	IDADE	NÍVEL
Nada	Frequente	Era	Regular	3	45	Normal
Moderado	Frequente	Era	Ocasional	1	64	Normal
Moderado	Por vezes	Era	Anterior	4	42	Anormal
Nada	Por vezes	Nunca	Nunca	2	55	Normal
MuitoPouco	Frequente	Nunca	Regular	1	54	Normal
Bastante	Frequente	Regular	Regular	4	57	Anormal
MuitoPouco	Frequente	Era	Regular	1	38	Normal
Moderado	Frequente	Era	Regular	2	60	Normal
MuitoPouco	Nunca	Era	Regular	3	53	Normal
Nada	Por vezes	Regular	Regular	2	41	Normal
Nada	Frequente	Era	Nunca	4	31	Anormal

De acordo com a tabela 3.2, os atributos sal e gordura relatam a quantidade destes na comida do paciente. O atributo fumante armazena valores referentes à frequência com que o paciente fuma (era fumante ou fumante regular) ou se ele nunca fumou. O atributo bebida está se referindo ao consumo de bebida alcoólica no dia-a-dia do paciente. O atributo ativfísica corresponde a prática de atividades físicas do paciente, observe que este está em algum código do hospital pois, estão representados com números. O atributo idade corresponde a idade do paciente e o atributo nível, ao nível da pressão arterial deste paciente. Logo, cada linha da tabela corresponde as características que irão determinar o nível de pressão arterial.

Nesta etapa de discretização o objetivo poderia ser transformar o atributo IDADE em valores discretos. Assim criar-se-iam faixas etárias para representar os valores do atributo

ou então nomes que expressem o seu valor. A construção de uma discretização de um atributo ou criação de classes é realizada respeitando o seguinte modelo: escolha do <ATRIBUTO>, escolha do <RELACIONAMENTO>, informação do <VALOR CONTÍNUO> e informação do <VALOR DISCRETO>. Para tal, a metodologia disponibiliza ao usuário os seguintes relacionamentos:

- Igual a;
- Entre;
- Maior ou igual a;
- Maior que;
- Menor ou igual a;
- Menor que.

De acordo com o exposto acima, poder-se-ia criar as seguintes classes para o atributo IDADE: meia idade e idoso. Ou então, com faixas etárias explícitas, tais como: 40 a 45, 45 a 55, 55 a 60, e assim, como melhor convir ao domínio da aplicação. No entanto, não se pode criar classes sem que pelo menos um valor de atributo do conjunto de dados esteja associado a esta.

O exemplo citado é apenas ilustrativo, mas imagine em um histórico de pacientes real onde pode-se ter idades de pacientes desde 10 até 100 anos, por exemplo, e milhares destes, assim, faz-se necessário a discretização para melhor generalizar estas informações para regras que é o objetivo do processo.

Já o atributo ATIVFÍSICA para melhor representá-lo nas futuras regras geradas pela metodologia pode-se criar classes de acordo com que cada número significa dentro do domínio da aplicação. Por exemplo, ter-se-ia as seguintes classes: diária, semanal, ocasional e nunca, de acordo com os valores associados, respectivamente, 1, 2, 3 e 4. Para tal, procede-se da seguinte forma: escolha do atributo ATIVFÍSICA, escolha do relacionamento IGUAL A, informação do valor contínuo 1, informação do valor discreto DIÁRIA. Desta forma faz-se para todos os valores desejados.

A metodologia proposta deixa a cargo do usuário a tarefa de discretização ou criação de classes, dando-lhe liberdade de fazê-la como melhor lhe convir. Assim, conforme a aplicação, um mesmo conjunto de exemplos pode servir a tarefas de classificação diferentes. Por exemplo, poder-se-ia ter um atributo altura discretizado como baixa, média e alta, onde estas noções de valores variam se, está se considerando alturas para um time de basquete ou alturas de uma determinada faixa etária de adolescentes em geral. Logo, esta etapa é opcional,

pode acontecer de, às vezes, não ser necessária dependendo do conjunto de exemplos e a aplicação do usuário.

Para o exemplo da tabela 3.2 adotar-se-á a discretização do atributo ativfísica, e do atributo idade como apresentado na tabela 3.3.

**Tabela 3.3** – Histórico de pacientes sobre avaliação do nível de pressão (fictício) - discretizado

SAL	GORDURA	FUMANTE	BEBIDA	ATIVFÍSICA	IDADE	NIVEL
Nada	Frequente	Era	Regular	ocasional	entre 40 e 45	Normal
Moderado	Frequente	Era	Ocasional	diária	entre 60 e 65	Normal
Moderado	Por vezes	Era	Anterior	nunca	entre 40 e 45	Anormal
Nada	Por vezes	Nunca	Nunca	semanal	entre 51 e 55	Normal
MuitoPouco	Frequente	Nunca	Regular	diária	entre 51 e 55	Normal
Bastante	Frequente	Regular	Regular	nunca	entre 56 e 59	Anormal
MuitoPouco	Frequente	Era	Regular	diária	entre 31 e 39	Normal
Moderado	Frequente	Era	Regular	semanal	entre 60 e 65	Normal
MuitoPouco	Nunca	Era	Regular	ocasional	entre 51 e 55	Normal
Nada	Por vezes	Regular	Regular	semanal	entre 40 e 45	Normal
Nada	Frequente	Era	Nunca	nunca	entre 31 e 39	Anormal

Após a realização das etapas anteriores, o conjunto de casos (tabela 3.3) será referenciado como conjunto de treinamento “C”. A próxima etapa da preparação dos dados, após a discretização, é a definição do tamanho da Janela e a escolha do nó objetivo. Esta, por sua vez, está diretamente relacionada ao algoritmo de mineração de dados utilizado, o ID3, e é obrigatória. A definição do tamanho da Janela corresponde a definir a porcentagem de objetos deste conjunto de treinamento que será utilizada inicialmente para a formação da árvore de decisão por indução. E a escolha do nó objetivo consiste em determinar qual o conceito que será aprendido pela tarefa de indução.

Adotar-se-á uma Janela de tamanho 100 %, ou seja, tomar-se-á todos os exemplos da tabela 3.3 para geração da árvore de decisão por indução. E tendo como objetivo para a tarefa de indução o diagnóstico do nível de pressão arterial tomar-se-á como nó objetivo o atributo nível com as classes normal e anormal.

Definido o objetivo, faz-se necessária a verificação de conflitos no conjunto de treinamento, uma vez que a discretização pode originar conflitos antes não existentes e existindo conflitos no conjunto de treinamento os objetos em conflito são considerados inadequados para a tarefa de indução, não sendo possível assim, gerar a árvore de decisão.



Dois ou mais objetos (registros) do conjunto de treinamento estão em conflito quando tiverem para todos os seus campos valores de campos iguais e tiverem o valor do nó objetivo diferente.

Quando ocorrerem conflitos e estes serem frutos de uma discretização aconselha-se a revisão da discretização, e para procurar eliminá-lo a criação de mais classes para o atributo discretizado ou se não, a inclusão de mais um atributo (campo) para a tarefa de classificação, voltando assim, a fase de origem dos dados, seleção dos campos da tabela. Esta tarefa é aconselhada a um especialista do domínio da aplicação.

### 3.2.3 Geração da Árvore de Decisão

---

Nesta fase da metodologia é apresentado em detalhes como ocorre a criação da árvore de decisão através do algoritmo ID3, ou melhor, como é feita a aprendizagem do conceito e representada na forma de árvore de decisão. Aplicar-se-á um exemplo, passo a passo, para melhor compreender os detalhes de funcionamento do algoritmo.

Os passos que são apresentados a seguir, para a formação da árvore de decisão, são realizados inicialmente para o conjunto  $C$ , tabela 3.3 (Janela de tamanho 100%), e logo após para cada subconjunto de  $C$ , sucessivamente, até encontrar uma folha para cada subconjunto. A título de exemplificação, tomar-se-á o conjunto de treinamento do nível de pressão arterial, o qual foi utilizado na etapa anterior.

Aplicando as suposições de Quinlan, (capítulo 2 pg. 33), para a formação da árvore de decisão por indução para uma tarefa de classificação de duas classes tem-se:

#### ⊗ 1<sup>o</sup> PASSO:

1. Contar o número total de  $p$  do conjunto de treinamento.
2. Contar o número total de  $n$  do conjunto de treinamento.

3. Calcular a probabilidade de um objeto pertencer a classe P:  $P(O_i(P)) = \frac{P}{(p+n)}$

onde  $(p+n)$  = total de objetos do conjunto de treinamento.

4. Calcular a probabilidade de um objeto pertencer a classe N:  $P(O_i(N)) = \frac{n}{(p+n)}$ ,

onde  $(p+n)$  = total de objetos do conjunto de treinamento.

De acordo com o exemplo do nível de pressão arterial,  $p$  corresponde ao número de objetos que pertencem a classe Normal e  $n$  ao número de objetos que pertencem a classe Anormal. Desta forma, sendo  $C$  o conjunto de objetos da tabela 3.3 com um total de 11 objetos, destes 8 são da classe Normal logo,  $p = 8$  e 3 são da classe Anormal logo,  $n = 3$  e assim o cálculo da probabilidade de um objeto pertencer a classe Normal é igual a  $8/11$  e de pertencer a classe Anormal é igual a  $3/11$ .

### ☒ 2<sup>o</sup> PASSO

Calcular a quantia de incerteza de informação contida neste conjunto  $C$  com relação a este determinar qual classe (Normal ou Anormal) pertence um objeto, ou seja, a entropia de classificação para o conjunto de treinamento  $C$ , tomando a equação (1) pg. 33, têm-se:

$$I(p, n) = -\frac{8}{11} \log_2 \frac{8}{11} - \frac{3}{11} \log_2 \frac{3}{11}$$

Assim,  $I(8, 3) = 0,8454 \text{ bits}$ .

### ☒ 3<sup>o</sup> PASSO

Para cada atributo (campo), exceto o atributo de classe (conceito a ser aprendido), calcular a informação esperada necessária para cada valor do atributo, ou seja, a entropia de classificação para cada subconjunto  $C_i$ . Têm-se os seguintes atributos, do exemplo do nível de pressão, com seus respectivos valores associados:

- sal (nada, moderado, muito pouco, bastante);
- gordura (frequente, por vezes, nunca);
- fumante (anterior, nunca, regular);
- bebida (regular, ocasional, anterior, nunca);
- ativfísica (diária, semanal, ocasional, nunca);
- idade (entre 31 e 39, entre 40 e 45, entre 51 e 55, entre 56 e 59, entre 60 e 65).

Assim, para o atributo fumante tem-se:

- $p_1 = 5$  e  $n_1 = 2$  e  $I(p_1, n_1) = I(5, 2) = 0,8631 \text{ bits}$ .
- $p_2 = 2$  e  $n_2 = 0$  e  $I(p_2, n_2) = I(2, 0) = 0 \text{ bits}$ .
- $p_3 = 1$  e  $n_3 = 1$  e  $I(p_3, n_3) = I(1, 1) = 1,00 \text{ bits}$ .

Analogamente calcular para os demais atributos.

#### ☒ 4<sup>o</sup> PASSO

Calcular a informação esperada necessária para a árvore com o atributo  $A$  como raiz através da média ponderada, equação (2) da pg. 34:

$$E(\textit{fumante}) = \sum_{i=1}^3 \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

onde:  $\textit{fumante}$  = atributo, possível raiz.

3 = número total dos possíveis valores para o atributo  $\textit{fumante}$ .

Desta forma,

$$E(\textit{fumante}) = \frac{7}{11} I(p_1, n_1) + \frac{2}{11} I(p_2, n_2) + \frac{2}{11} I(p_3, n_3) =$$

$$E(\textit{fumante}) = \frac{7}{11} 0,8631 + \frac{2}{11} 0 + \frac{2}{11} 1 =$$

$$E(\textit{fumante}) = 0,7311 \textit{ bits}$$

Fazer este cálculo para cada atributo do passo 3. Está-se procurando a possível raiz.

Assim, tem-se:

- $E(\textit{sal}) = 0,5455 \textit{ bits}$
- $E(\textit{gordura}) = 0,7997 \textit{ bits}$
- $E(\textit{fumante}) = 0,7311 \textit{ bits}$
- $E(\textit{bebida}) = 0,5583 \textit{ bits}$
- $E(\textit{ativfísica}) = 0,00 \textit{ bits}$
- $E(\textit{idade}) = 0,4323 \textit{ bits}$

Como o  $E(\textit{ativfísica})$  é igual a zero, quer dizer que o nível de incerteza da classificação de um objeto utilizando este atributo  $\textit{ativfísica}$  é igual a zero.

#### ☒ 5<sup>o</sup> PASSO

Calcular o ganho de informação pela ramificação em  $A$ , tomando a equação (3) da pg. 34, tem-se:

$$\textit{Gain}(\textit{fumante}) = I(8,3) - E(\textit{fumante})$$

$$\textit{Gain}(\textit{fumante}) = 0,8454 - 0,7311$$

$$Gain(\text{fumante}) = 0,1143 \text{ bits}$$

Realizar este cálculo para todos os atributos do passo 3, ou seja, as possíveis raízes. Assim, tem-se:

- $Gain(\text{sal}) = 0,2999 \text{ bits}$
- $Gain(\text{gordura}) = 0,0456 \text{ bits}$
- $Gain(\text{fumante}) = 0,1143 \text{ bits}$
- $Gain(\text{bebida}) = 0,2870 \text{ bits}$
- $Gain(\text{ativfisica}) = 0,8454 \text{ bits}$
- $Gain(\text{idade}) = 0,4131 \text{ bits}$ .

### ⊗ 6º PASSO

Escolher o atributo de maior ganho de informação para ser a raiz da árvore. Será o atributo para ramificar a árvore e conseqüentemente dividir o conjunto  $C$  em subconjuntos  $C_i$ .

De acordo com os valores anteriores verifica-se que o atributo de maior ganho é *ativfisica*, portanto é o atributo mais informativo para a tarefa de classificação.

### ⊗ 7º PASSO

Dividir o conjunto de treinamento  $C$  (tabela 3.3) em subconjuntos de acordo com os possíveis valores do atributo raiz escolhido no passo 6. Logo em cada subconjunto tem-se apenas um valor distinto deste atributo raiz.

De acordo com o exemplo utilizado, os subconjuntos  $C_i$  gerados são mostrados nas tabelas 3.4, 3.5, 3.6 e 3.7 a seguir:

**Tabela 3.4** – Subconjunto  $C_1$  gerado da partição no atributo raiz *ativfisica* (diária)

SAL	GORDURA	FUMANTE	BEBIDA	IDADE	NIVEL
Moderado	Frequente	Era	Ocasional	entre 60 e 65	Normal
MuitoPouco	Frequente	Nunca	Regular	entre 51 e 55	Normal
MuitoPouco	Frequente	Era	Regular	entre 31 e 39	Normal

**Tabela 3.5** – Subconjunto  $C_2$  gerado da partição no atributo raiz *ativfisica* (semanal)

SAL	GORDURA	FUMANTE	BEBIDA	IDADE	NIVEL
Nada	Por vezes	Nunca	Nunca	entre 51 e 55	Normal
Moderado	Frequente	Era	Regular	entre 60 e 65	Normal
Nada	Por vezes	Regular	Regular	entre 40 e 45	Normal

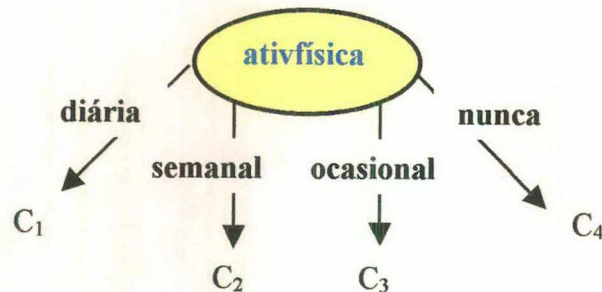
**Tabela 3.6** - Subconjunto  $C_3$  gerado da partição no atributo raiz ativfísica (ocasional)

SAL	GORDURA	FUMANTE	BEBIDA	IDADE	NÍVEL
Nada	Frequente	Era	Regular	entre 40 e 45	Normal
Muito Pouco	Nunca	Era	Regular	entre 51 e 55	Normal

**Tabela 3.7** - Subconjunto  $C_4$  gerado da partição no atributo raiz ativfísica (nunca)

SAL	GORDURA	FUMANTE	BEBIDA	IDADE	NÍVEL
Moderado	Por vezes	Era	Anterior	entre 40 e 45	Anormal
Bastante	Frequente	Regular	Regular	entre 56 e 59	Anormal
Nada	Frequente	Era	Nunca	entre 31 e 39	Anormal

Observe como fica a representação destes subconjuntos em árvore de decisão, figura 3.4.

**Figura 3.4** – Formação da árvore de decisão

### ☒ 8º PASSO

Verificar os valores do atributo de classe no subconjunto  $C_i$ :

- ☞ Se estes pertencem somente a uma classe então uma folha é encontrada do nó raiz (atributo que gerou a divisão) e é rotulada com o nome da classe. O valor do atributo que particionou o subconjunto  $C_i$  torna-se o ramo da árvore até esta folha e a seqüência de passos termina para este subconjunto.
- ☞ Se não, repetir a seqüência de passos da geração da árvore de decisão a partir do 1º passo, tomando  $C_i$  como  $C$ .

Fazer isto para todos os subconjuntos divididos (um subconjunto por vez).

No exemplo utilizado todos os subconjuntos geraram partições em que existe apenas um valor de classe, logo a seqüência de passos para a geração da árvore termina.



Observe na figura 3.5 como fica a formação da árvore de decisão. Verifica-se que para o conjunto de treinamento da pressão arterial (tabela 3.3), o sistema ID3 descartou os atributos: sal, gordura, fumante, bebida e idade, considerando-os irrelevantes para a tarefa de classificação.



**Figura 3.5** – Formação da árvore de decisão: criação de uma folha

Como para a formação da árvore de decisão da figura 3.5 partiu-se de uma Janela de tamanho 100% do conjunto de treinamento  $C$ , a referida árvore garante a correta classificação de todos os objetos envolvidos na sua formação.

### 3.2.4 Extração de Regras e Cálculo de Probabilidades

---

Realizada a aprendizagem na fase anterior, o conhecimento descoberto encontra-se representado na forma de árvore de decisão. Como o objetivo é a representação deste na forma de regras, nesta fase da metodologia apresenta-se a transformação do conhecimento descoberto para regras, como também, os cálculos necessários para encontrar a probabilidade associada a cada regra.

Para formação das regras a partir da árvore de decisão da figura 3.5 toma-se o atributo raiz *ativfisica* juntamente com o seu valor associado (ramo) como parte da premissa da regra, e a parte conclusiva da regra sendo o atributo de classe (conceito a ser instruído) juntamente com o seu valor (folha).

Desta forma, a partir da raiz da árvore de decisão da figura 3.5 constroem-se as seguintes regras, (maiores detalhes ver pg. 39):

- ✓ Se ATIVFÍSICA = diária então NÍVEL = Normal
- ✓ Se ATIVFÍSICA = semanal então NÍVEL = Normal
- ✓ Se ATIVFÍSICA = ocasional então NÍVEL = Normal
- ✓ Se ATIVFÍSICA = nunca então NÍVEL = Anormal

Extraídas as regras da árvore de decisão, passa-se agora para a última tarefa do processo: o cálculo da probabilidade (suporte) de cada regra de acordo com a equação (4) e o cálculo da confiança de acordo com a equação (5), ambos apresentados na pg. 40.

Tomando a primeira regra:

- ✓ Se ATIVFÍSICA = diária então NÍVEL = Normal

Tem-se que:

$$\text{Suporte} = \frac{3}{11} = 0,2727 \quad \text{e} \quad \text{Confiança} = \frac{3}{3} = 1$$

Pois, do total de 11 objetos do conjunto de treinamento  $C$  (tabela 3.3) existem 3 objetos onde se verificam a condição e a conclusão da regra, ou seja, o atributo ativfísica é igual a diária e o atributo nível é igual a normal, logo a probabilidade (suporte) para esta regra é igual a  $3/11$  ou 27,27%. E existem apenas 3 objetos onde se verifica a condição da regra, ou seja, o atributo ativfísica ser igual a diária, logo a confiança da regra é igual a  $3/3$  ou 100%.

Desta forma, calcula-se o suporte e a confiança para todas as regras extraídas da árvore de decisão em relação ao conjunto de treinamento  $C$  (tabela 3.3).

Para o exemplo do nível de pressão arterial, são mostradas na tabela 3.8 as probabilidades (suporte) associadas a cada regra e a sua confiança:

**Tabela 3.8** – Probabilidades associadas a cada regra (nível de pressão)

Regra	Suporte (%)	Confiança (%)
Se ATIVFÍSICA = diária então NÍVEL = Normal	27,27	100
Se ATIVFÍSICA = semanal então NÍVEL = Normal	27,27	100
Se ATIVFÍSICA = ocasional então NÍVEL = Normal	18,18	100
Se ATIVFÍSICA = nunca então NÍVEL = Anormal	27,27	100

Portanto, finaliza-se a descrição das etapas da metodologia proposta de uso de técnicas de indução para criação de regras de sistemas especialistas. Sendo assim, conclui-se que a metodologia proposta gerou parte da base de conhecimento para um sistema especialista baseado em regras com suas respectivas probabilidades, onde esta por sua vez, pode ser analisada por um especialista do domínio para validação ou inclusão de novas regras a base de conhecimento.



A partir desta metodologia criou-se um protótipo de software, que é apresentado no capítulo 4, para que parte da base de conhecimento seja adquirida automaticamente na forma de regras mais probabilidades associadas.



## 4. A FERRAMENTA GARP

---

*Neste capítulo é apresentado o protótipo implementado, de acordo com a metodologia proposta de uso técnicas de indução para a criação de regras de sistemas especialistas, ressaltando o ambiente do sistema, os dados utilizados a tipo de exemplificação, o seu funcionamento passo a passo, e por fim, algumas constatações referentes a sua aplicação realizada em dados fictícios e em uma base de dados real do Jogo de Empresas GI-EPS.*

---

### 4.1 Ambiente

---

A ferramenta GARP – Gerador Automático de Regras Probabilísticas pode ser utilizada em microcomputadores IBM-PC, sob ambiente Windows. A plataforma mínima necessária é um 386 com 4 MB de RAM, sendo recomendado a utilização de um Pentium, com 32 MB (ou mais) de RAM para a exploração de bases de médio porte. A implementação foi feita em Delphi, linguagem utilizada pelo grupo de pesquisa do Laboratório de Jogos de Empresas – LJE.

Além disso, a escolha foi influenciada pela padronização dentro do grupo de pesquisa, o que beneficia a reusabilidade de objetos, e pela disponibilidade do Delphi como linguagem adequada para o desenvolvimento do protótipo, haja vista a possibilidade de conectá-lo com servidores SQL e praticamente todo tipo de base de dados.

Desta forma, a ferramenta GARP pode ser utilizada em uma arquitetura cliente / servidor como em uma base de dados local, independente do tipo de banco de dados associado.

## 4.2 Dados

---

A base de dados a ser analisada a título de exemplificação de uso da ferramenta GARP corresponde ao exemplo fictício da pressão arterial de pacientes de um histórico hospitalar, tabela 3.2, apresentada no capítulo 3.

Neste primeiro momento está trabalhando-se apenas com uma tabela de dados, quando se desejar utilizar dados de duas tabelas para a descoberta de regras, o sistema será aplicado a uma relação resultante da junção das duas tabelas.

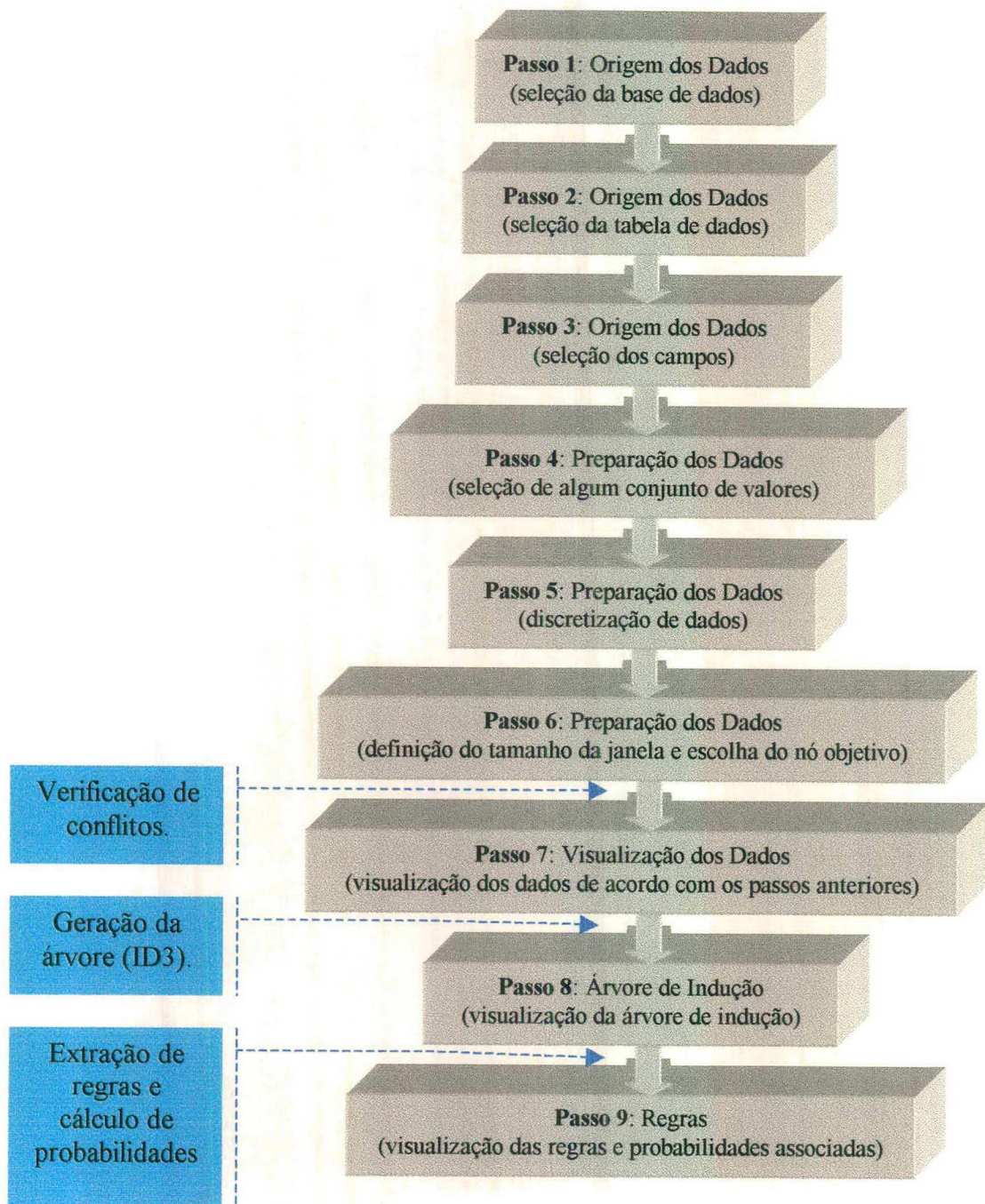
## 4.3 Funcionamento do Sistema

---

Esta seção demonstra a aplicação da metodologia de uso de técnicas de indução para a criação de regras de sistemas especialistas, através do protótipo implementado GARP – Gerador Automático de Regras Probabilísticas, objetivando apresentar o seu funcionamento destacando a sua interface, funções e as interações que o usuário possa a vir fazer com o sistema.

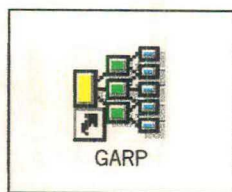
A interface do GARP está estruturada em 9 passos. A seqüência dos passos e sua principal função são mostradas na figura 4.1.





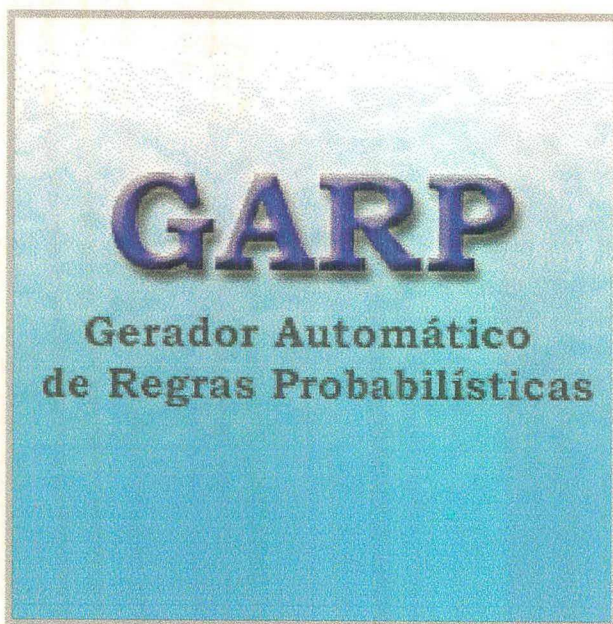
**Figura 4.1** – Seqüência de passos (interface - GARP)

O acesso ao sistema ocorre pelo ícone mostrado na figura 4.2, a partir do desktop do microcomputador.



**Figura 4.2** – Ícone de acesso a ferramenta GARP

Desta forma é executado o protótipo apresentando a tela de abertura, ver figura 4.3, e logo após a sua interface inicial, ver figura 4.4.



**Figura 4.3** – Tela de abertura da ferramenta

Na interface inicial (figura 4.4) pode-se observar que o protótipo dispõe de um menu superior com as seguintes opções: novo projeto, abrir um projeto já existente, salvar, imprimir e ajuda. A navegação pelo sistema ocorre pelos botões (sair, anterior e próximo) apresentados na parte inferior da interface, sendo que a qualquer momento o usuário pode optar por sair do sistema. O botão próximo sempre executa a(s) operação(ções) relacionada(s) com o passo do software em que o usuário se encontra e o botão anterior possibilita que o usuário volte ao passo anterior executado.



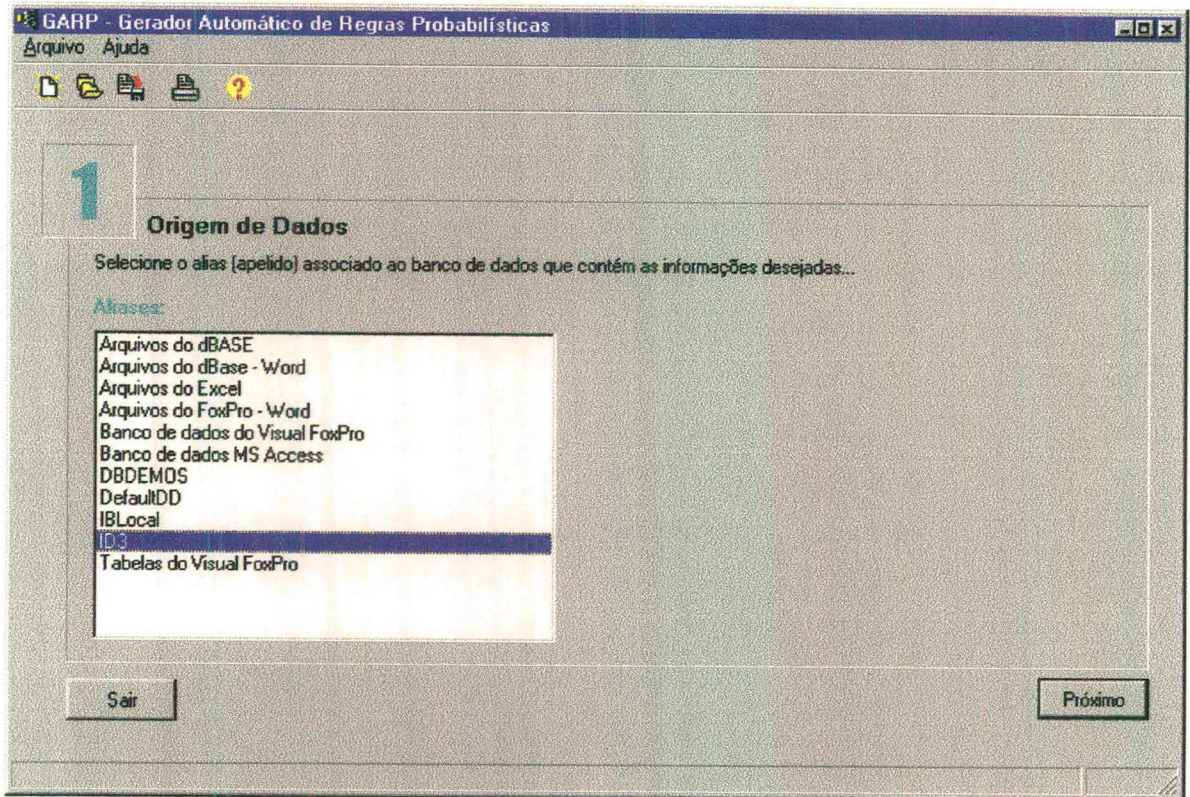
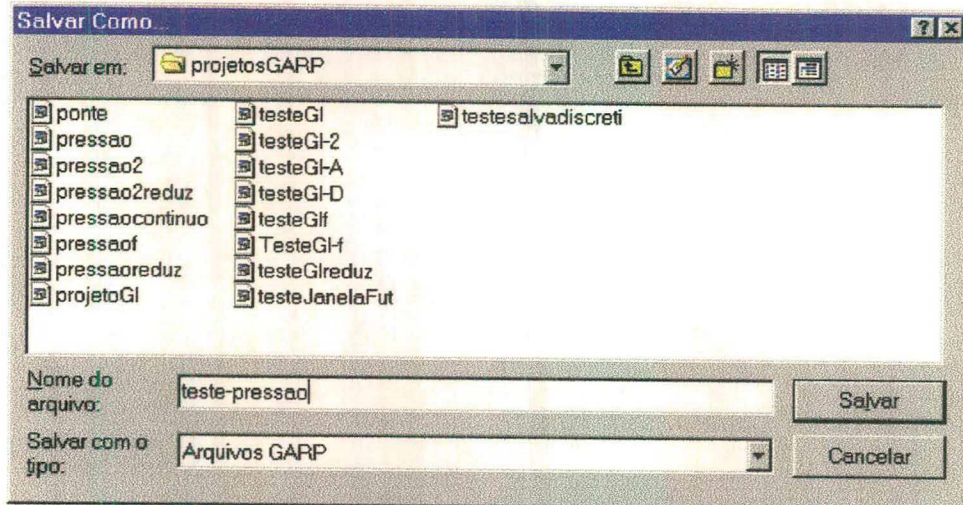


Figura 4.4 – Interface inicial do protótipo GARP

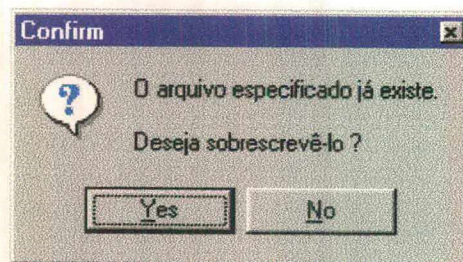
Ao optar por um novo projeto, independente de qual interface (passo) o usuário esteja, automaticamente o sistema volta ao passo 1 (figura 4.4) e fica por responsabilidade do usuário em escolher a opção salvar, para poder salvar seu projeto atual antes de inicializar outro projeto. Como mostra a figura 4.5 os projetos salvos recebem uma extensão própria do sistema denominada **grp**, identificando um arquivo do protótipo GARP, e fica a critério do usuário informar um nome ao arquivo. Caso o nome informado pelo usuário já exista o sistema mostra uma mensagem de aviso, ver figura 4.6. Nesta versão do protótipo o projeto é salvo até o **Passo 5 - Preparação dos Dados: Discretização**, em um único arquivo aberto apenas pelo próprio sistema.





**Figura 4.5** – Salvando um projeto

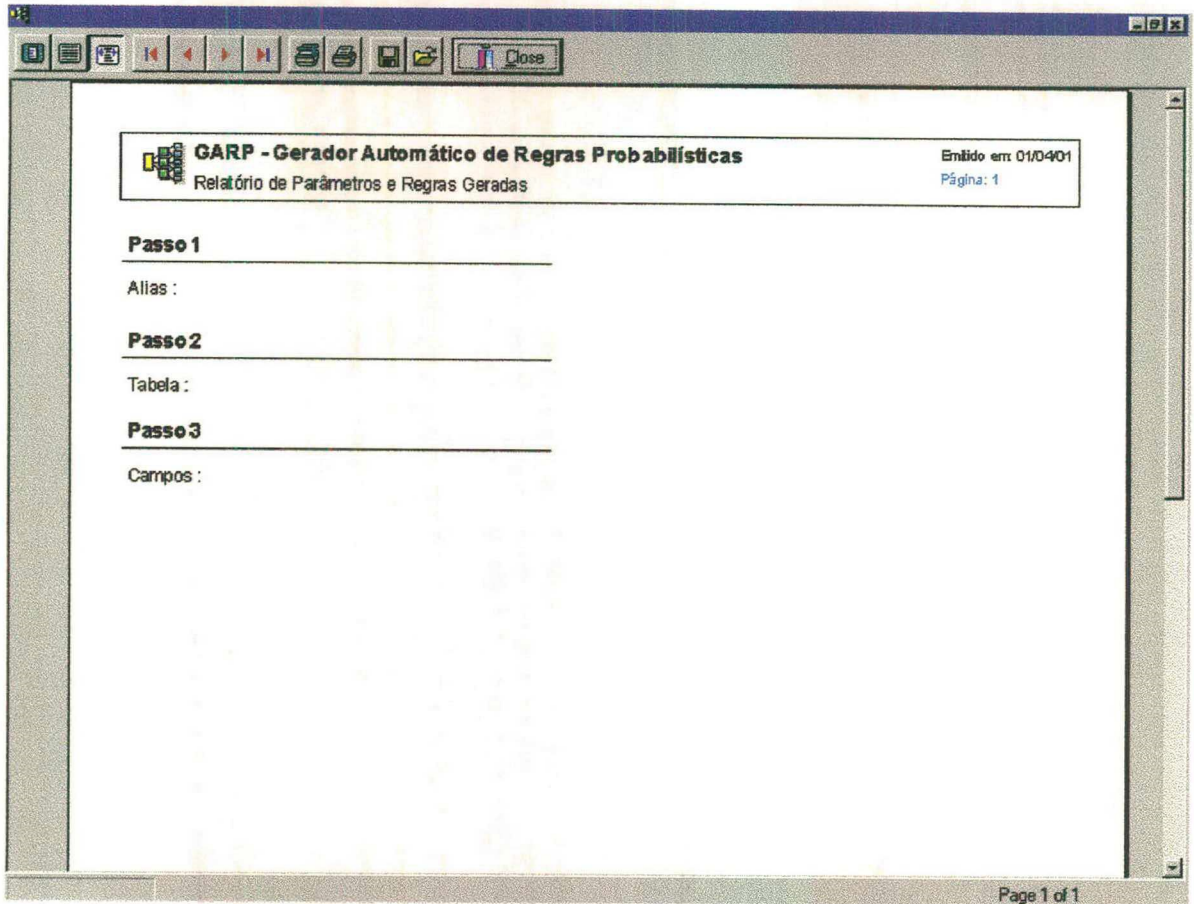
Para abrir um projeto já existente o usuário deve acessar a opção de menu abrir e somente se o usuário tiver acesso a base de dados e a tabela que gerou o projeto é que o sistema permitirá a abertura do projeto, caso contrário, apresentar-lhe-á uma mensagem de erro avisando que os dados de acesso não foram encontrados. Caso o sistema esteja sendo utilizado em uma base de dados sem restrições de acesso, mesmo assim o projeto só será aberto se ainda existir a fonte de dados que o gerou. Isto impede que o projeto seja alterado independentemente dos dados que o gerou, garante uma certa integridade entre os dados e o projeto. O projeto só existe se os “dados” (tabela) ainda existem. Já alterações nos dados (registros da tabela) ficam a encargo do banco de dados (SGBD) controlar, não cabe a ferramenta diagnosticar se houve violação ou mudanças nos registros.



**Figura 4.6** – Mensagem de aviso opção salvar: arquivo já existente

A opção de menu imprimir permite que o usuário imprima um relatório sobre o projeto realizado, além de disponibilizar-lhe outras funções referentes ao relatório gerado pelo sistema. Observe no alto desta interface (ver figura 4.7) as opções de menu disponíveis, dentre elas, as opções: salvar e abrir. Logo, o usuário também tem a opção de salvar em um arquivo

próprio do sistema o relatório gerado pelo projeto e quando desejar abri-lo deve fazê-lo através desta interface. Nesta versão o conteúdo do relatório está apenas como um esboço. O principal objetivo com o relatório é de documentar os passos realizados no projeto, salvando-os para futuras consultas, inclusive deixando registrado os resultados obtidos como as regras e probabilidades alcançadas. A abertura de um relatório já existente, de um projeto, independe do acesso a sua base de dados.



**Figura 4.7** – Interface da opção imprimir (relatório)

Como desenvolver um novo projeto utilizando os passos disponíveis na ferramenta GARP é mostrado no próximo item 4.4.



## 4.4 Aplicação da Metodologia Proposta através do GARP

---

De acordo com os dados apresentados no item 4.2 Dados, é realizada uma aplicação da metodologia de uso de técnicas de indução para criação de regras (ver capítulo 3) fazendo-se uso da ferramenta GARP. Nesta aplicação objetiva-se encontrar alguma relação entre os dados, conhecimento novo e útil, e conseqüentemente criação das regras e probabilidades para um sistema especialista do tipo SAD (Sistema de Apoio a Decisão), podendo ser desenvolvido com a *shell* SPIRIT (Rödder, 1995). Além de apresentar as possibilidades de interação do usuário com a ferramenta, passo a passo, nesta descoberta de conhecimento.

### ⇒ PASSO 1:

Inicialmente o usuário escolhe a base de dados que irá trabalhar, ou seja, onde estão armazenadas as informações que ele necessita. Neste caso não há a necessidade de se saber fisicamente em qual winchester e/ou em qual parte está localizado o banco de dados, pois o Delphi possui o recurso de utilizar apelidos (aliases) para os mesmos, tornando transparente para o usuário a sua localização física. Ver na figura 4.4 a escolha realizada da base de dados.

Quando o acesso ao banco possui restrições (usuário / senha), estas são solicitadas no momento em que a seleção do banco for feita, ver figura 4.8.

### ⇒ PASSO 2:

Neste momento o usuário escolhe em qual tabela estão os dados em que ele deseja trabalhar. (Figura 4.9). Observe que o botão detalhes permite visualizar os campos da tabela.



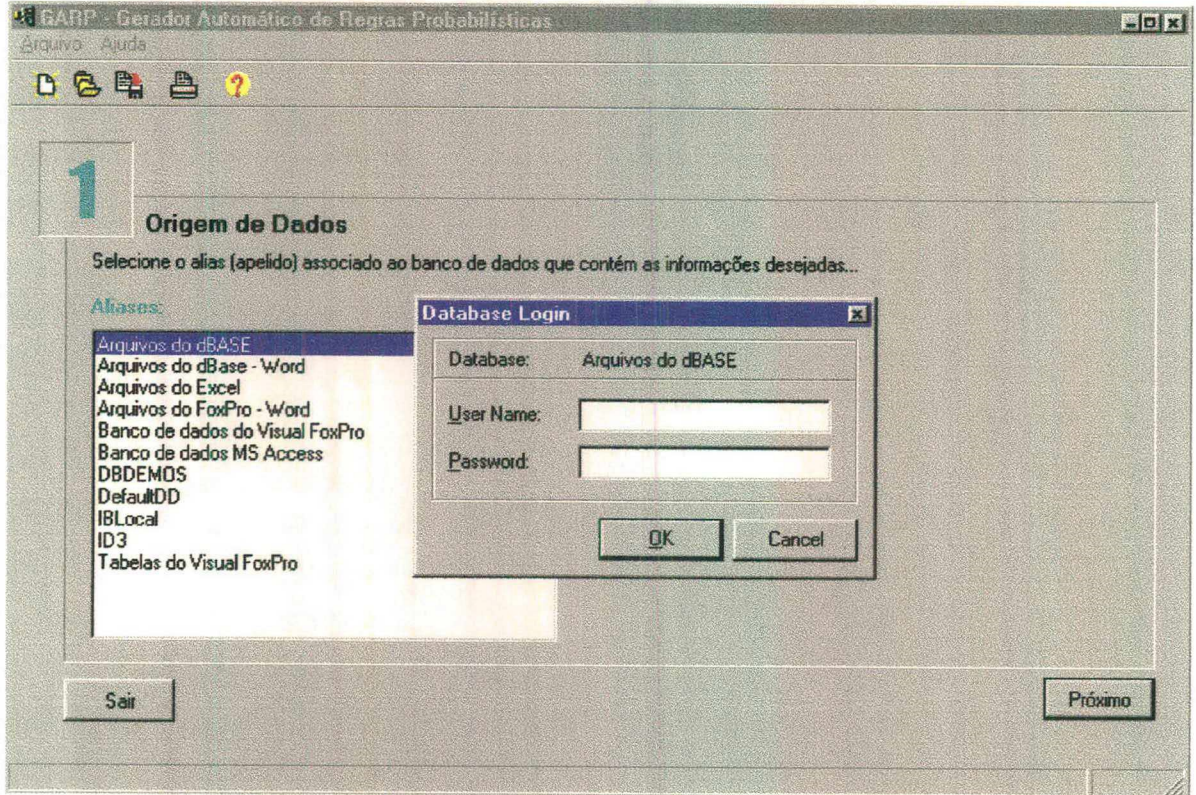


Figura 4.8 – Acesso a uma base de dados com restrições (usuário / senha)

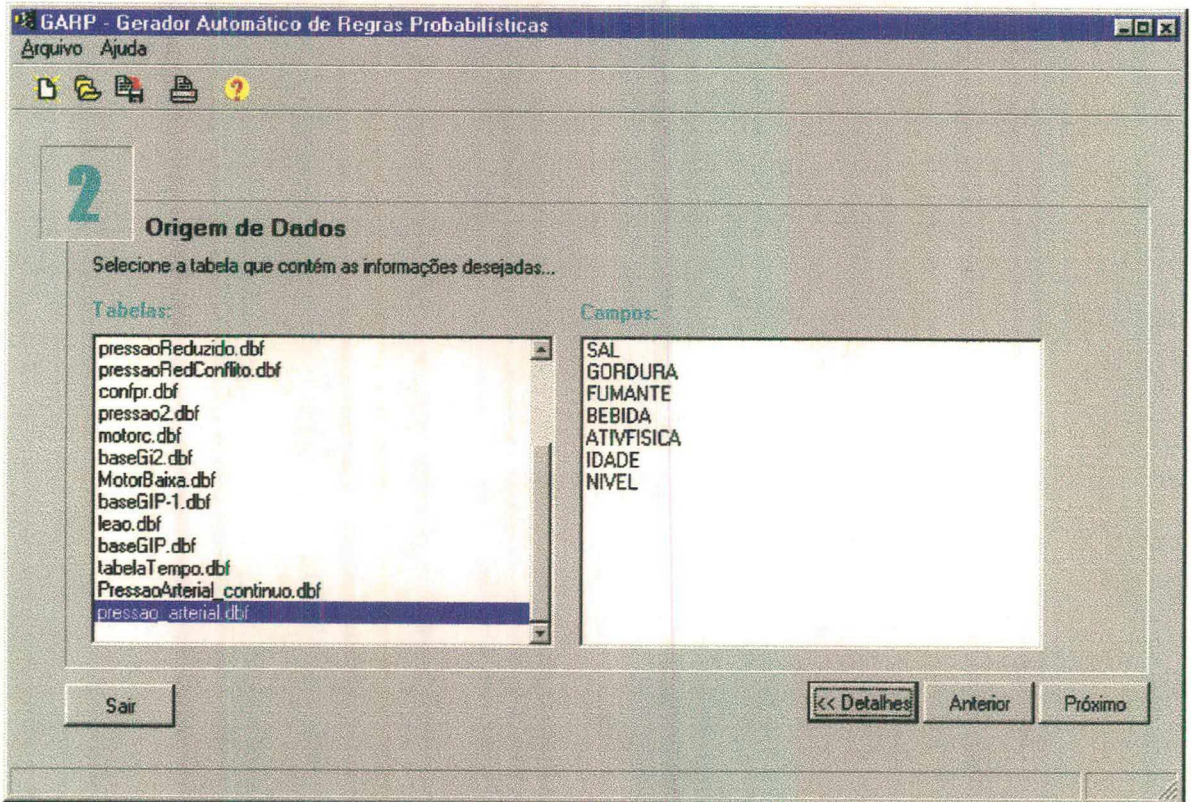


Figura 4.9 – Seleção da tabela que contém as informações desejadas



### ⇒ PASSO 3:

Neste passo o usuário seleciona os campos da tabela que deseja utilizar, podendo selecionar alguns ou todos. Deve-se ter em mente qual o conceito que o sistema irá aprender e através de quais atributos pretende-se fazer esta aprendizagem. Ver figura 4.10.

Até o momento, ainda não se salvou o projeto. Como esta ação pode ser feita em qualquer passo do mesmo, pode-se fazê-la agora, escolhendo o item de menu salvar ou opção Arquivo: salvar.

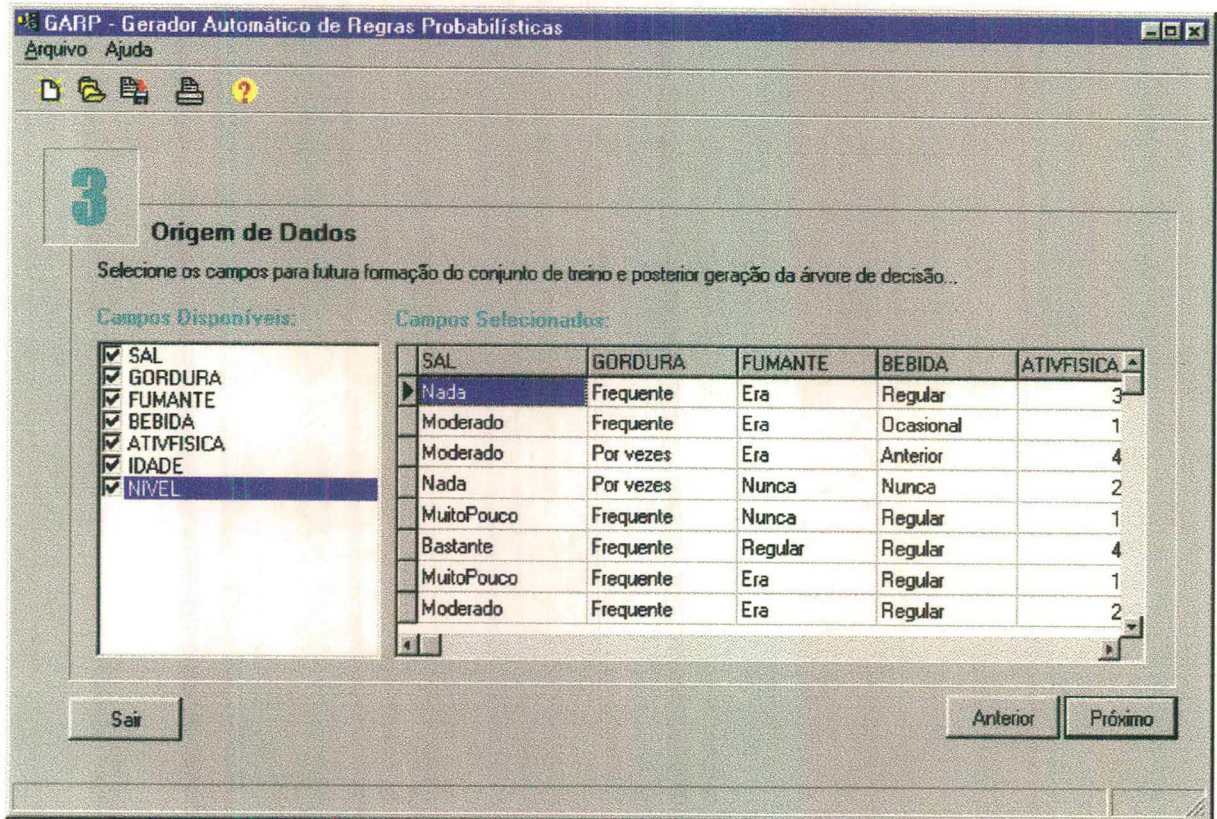


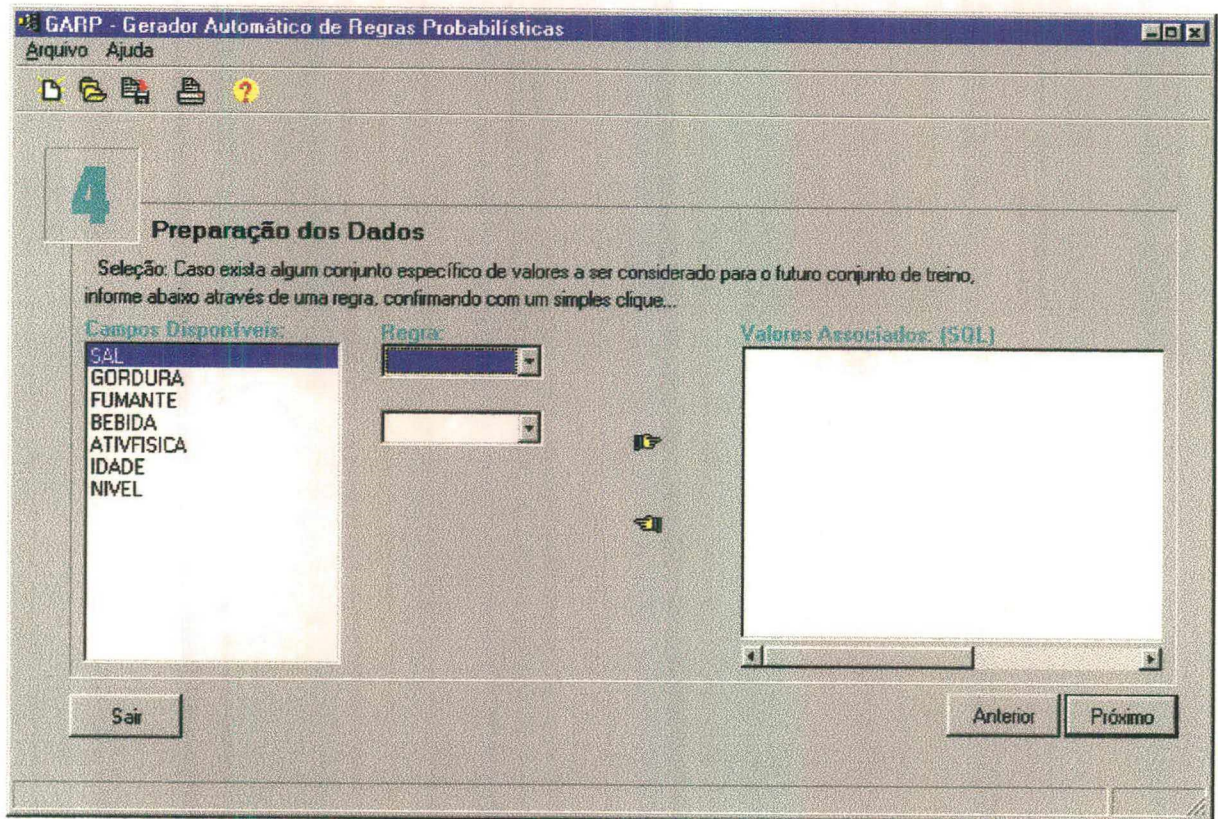
Figura 4.10 – Seleção dos campos que se deseja

### ⇒ PASSO 4:

Definida a origem dos dados nos passos anteriores, neste passo o usuário pode fazer alguma “query” (consulta) selecionando um determinado conjunto de dados (registros) dentro do universo disponível. Esta consulta é independente do usuário saber ou não comandos SQL, o sistema se encarrega desta busca. Basta fazer a seleção de qual campo se deseja restringir o domínio e de que forma, escolhendo a regra e os limites da seleção. Logo após, confirmar a seleção acionando o ícone “mãozinha” sentido para os valores associados (SQL), caso desista desta seleção deve-se selecionar o valor SQL que se pretende eliminar e



em seguida clicar na “mãozinha” sentido inverso ao anterior. Uma observação a ser considerada no protótipo é que valores numéricos da tabela a ser utilizada na formação da regra nesta seleção (SQL) só estão sendo aceitos com ponto na casa decimal em vez de vírgula. Na busca de conhecimento que está sendo realizada optou-se por não fazer restrições no espaço de busca. Ver figura 4.11.



**Figura 4.11** – Preparação dos dados: seleção de algum conjunto específico de valores

Logo, a realização deste passo pelo usuário é opcional. Ele pode optar por utilizar todos os dados (tabela inteira) em relação aos campos selecionados no passo 3, ou fazer alguma restrição no seu espaço de busca. Por exemplo, se fosse uma tabela de dados de uma seguradora, poderia estar objetivando apenas os registros do ano de 1986 ou todos os clientes com mais de 30 anos e menos de 50 anos de idade e que possuem um Chevette 77 preto.

No momento da escolha do campo e de uma regra os possíveis valores distintos do campo associado a esta são listados para facilitar a seleção de valores para o usuário. As regras disponíveis no sistema são apresentadas na figura 4.12. Caso o usuário opte, por exemplo, a trabalhar apenas com os dados de pacientes acima de 45 anos, poderia fazer esta seleção como é mostrada na figura 4.13. Apesar desta seleção ser possível trabalhar-se-á com todos os dados da tabela 3.2.



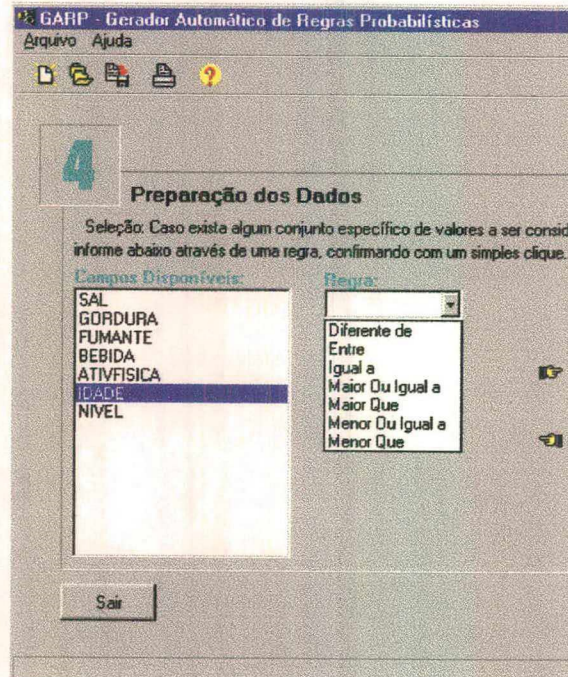


Figura 4.12 – Tipos de regras disponíveis para seleção

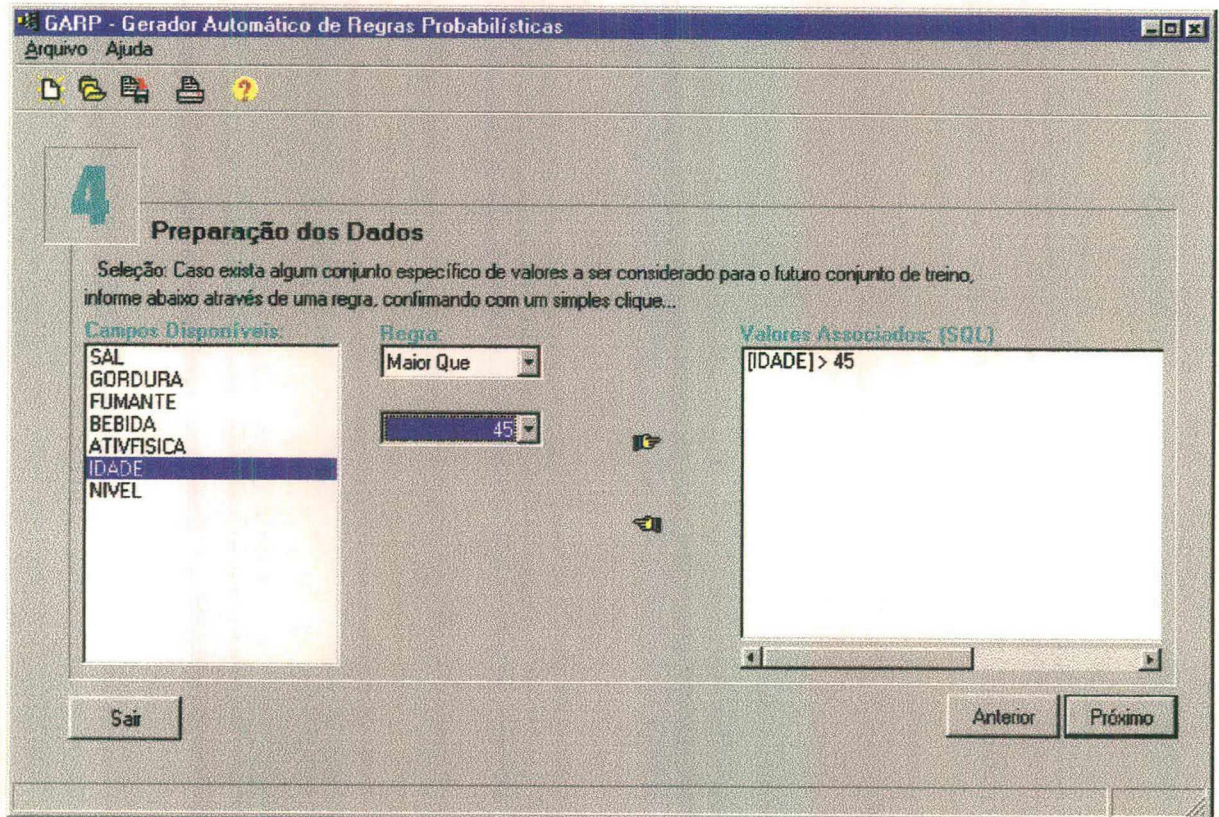
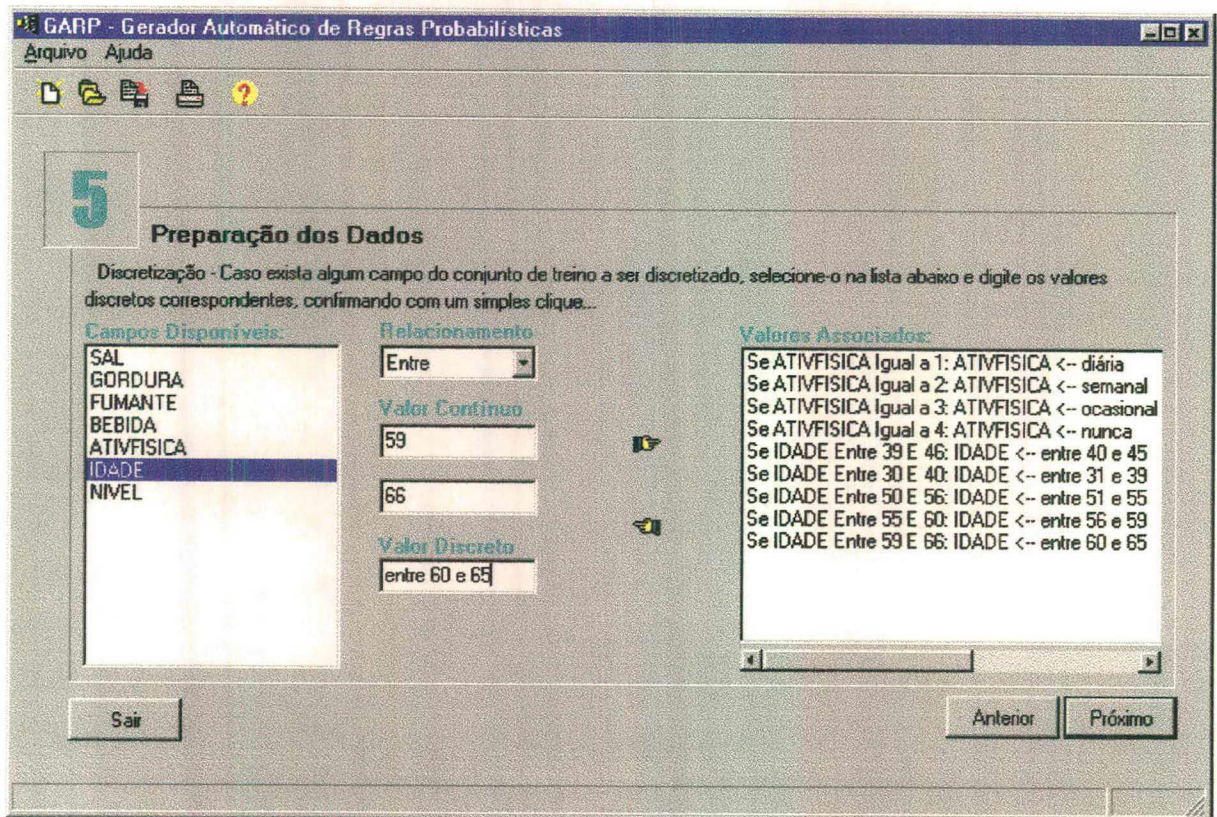


Figura 4.13 – Seleção dos dados dos pacientes com idades acima de 45 anos



## ➤ PASSO 5:

Este passo ainda abrange a preparação dos dados. No entanto com uma função especial, transformar valores contínuos de atributos em valores discretos, ou se o usuário preferir pode utilizá-lo para a criação de classes independente do tipo do atributo, mas sim, por desejar tornar os valores dos atributos mais significativos para a tarefa que está resolvendo. Observe na figura 4.14 que o usuário deve selecionar o campo a ser discretizado, escolher o relacionamento pertinente a discretização desejada, preencher o valor contínuo e o valor discreto que se deseja associar. A utilização desta interface é muito semelhante a do passo 4 (anterior).



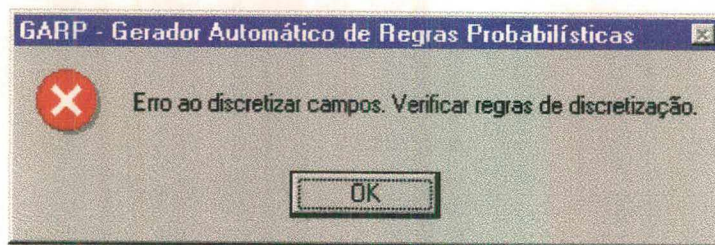
**Figura 4.14 – Discretização dos atributos (criação de classes)**

Para os dados que estão sendo utilizados optou-se por fazer duas discretizações: uma para o atributo atívfisica e outra para o atributo idade. Tomando o atributo idade como exemplo, a sua discretização pode ser feita pelo usuário através da escolha deste atributo, do relacionamento ENTRE e pela informação dos valores contínuos 59 e 66 em seguida informando o valor discreto associado “entre 60 e 65”. Observe que no momento de informar



os valores contínuos foram informados 59 e 66, e não 60 e 65, isto faz-se necessário devido o relacionamento ENTRE não incluir os extremos.

Sugere-se a criação de dez classes no máximo para um mesmo atributo. A tarefa de discretização ou criação de classes é de responsabilidade do usuário fazê-la da melhor forma que convir ao seu objetivo com o projeto que está desenvolvendo, no entanto, o sistema emite uma mensagem de erro de discretização caso o usuário crie uma classe onde não exista pelo menos um valor correspondente a esta na base de dados. Ver figura 4.15. Maiores detalhes sobre a tarefa de discretização ver capítulo 3.



**Figura 4.15** – Mensagem de erro de discretização.

#### ➔ PASSO 6:

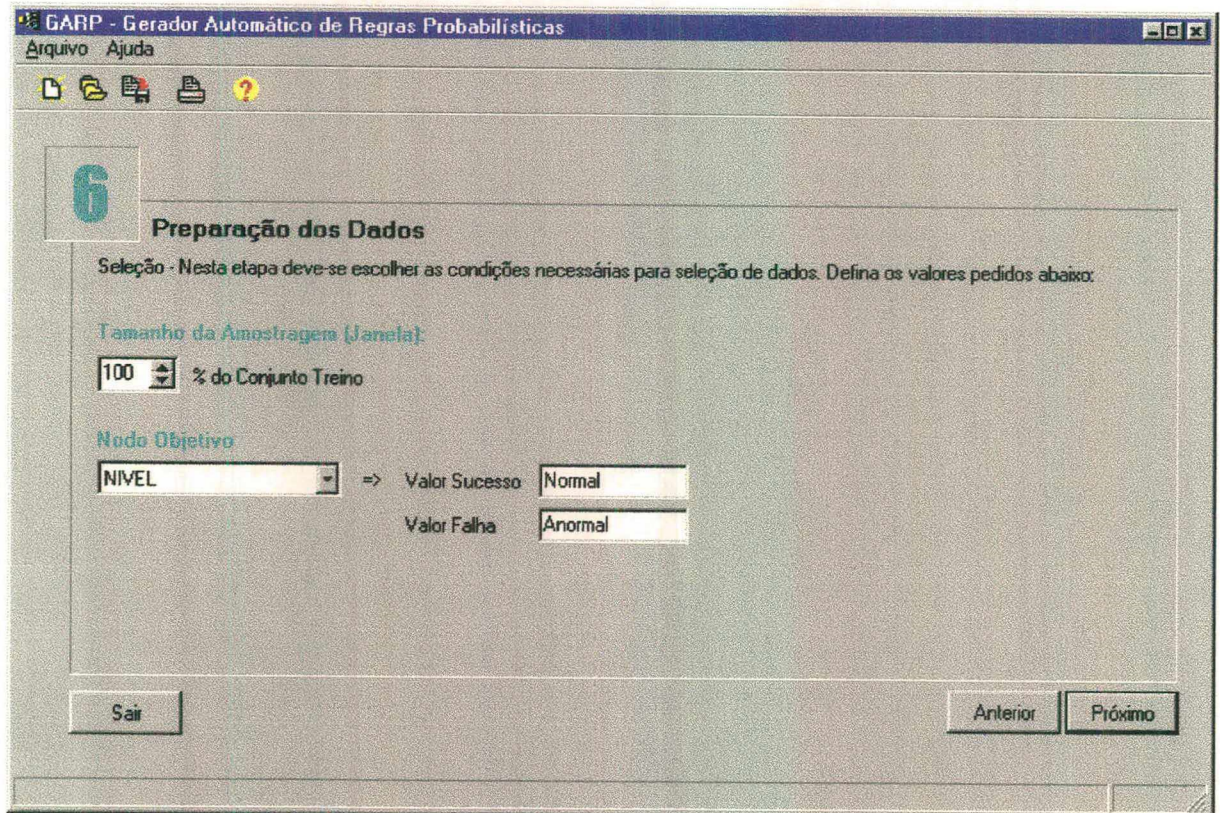
Esta etapa do processo está relacionada diretamente com o algoritmo ID3, onde o usuário deve informar qual o tamanho inicial da janela, para a geração da árvore e qual é o conceito a ser aprendido, isto é, qual o nó objetivo e os seus respectivos valores de classe associados. Uma observação a ser considerada neste protótipo é que os valores de classe associados ao nó objetivo devem ser digitados da mesma forma (letras maiúsculas e minúsculas) em que estes se encontram no conjunto de treinamento. A figura 4.16 apresenta esta etapa do processo, onde foram determinados:

- Tamanho da Janela: 100 %
- Nó objetivo: NÍVEL
- Valores de classe associados: Normal e Anormal.

Com a escolha do nó objetivo, nível de pressão arterial, objetiva-se identificar regras de classificação a partir dos demais atributos (premissa da regra) para determinar este conceito (conclusão da regra): pressão arterial normal, pressão arterial anormal. Neste caso diz-se que o sistema irá aprender por indução o que determina uma pressão arterial normal ou



anormal, dentro do contexto em que está sendo aplicado, pois a base de dados é formada por vários exemplos (objetos) com estas características. Detalhes sobre a tarefa de indução e sobre o algoritmo ID3, ver capítulo 2.



**Figura 4.16** – Escolha do conceito a ser instruído e tamanho inicial da amostra.

Na transição do Passo 6 ao Passo 7 ocorre a verificação de conflitos. Caso ocorra conflito, este é informado ao usuário ainda no Passo 6, de acordo com figura 4.17 e ao pressionar o botão OK são listados os objetos que se encontram em conflito. Nesta versão do protótipo cabe ao usuário (especialista do domínio) eliminar o conflito adicionando novos atributos para a tarefa de indução. Para tal, o especialista deve retornar ao passo 3 e selecionar mais um campo (atributo), isso pode ser feito caso na sua realização deste passo 3 anteriormente não tenha selecionado todos os campos da tabela. E caso o conflito tenha surgido como fruto da discretização realizada, o especialista pode refiná-la, ou seja, diminuir os intervalos de classes, criando assim mais classes, a ponto de eliminar o conflito. De modo geral um conflito ocorre quando dois ou mais objetos possuem valores idênticos para todos os atributos, mas com valores diferentes para o atributo objetivo. Maiores detalhes sobre conflitos em dados ver capítulo 2.



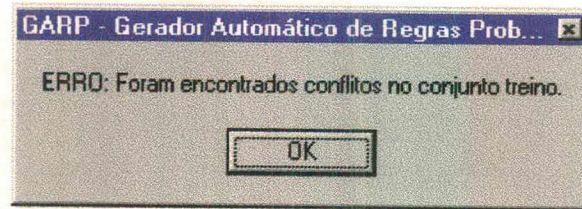


Figura 4.17 – Mensagem de identificação de conflito

### ⇒ PASSO 7:

A função deste passo 7 é permitir ao usuário a visualização da amostragem, neste caso janela 100 %, selecionada no passo 6. Esta amostragem corresponde à visualização dos dados (atributos e seus valores) de acordo com todos os passos realizados anteriormente. A figura 4.18 exemplifica esta interface.

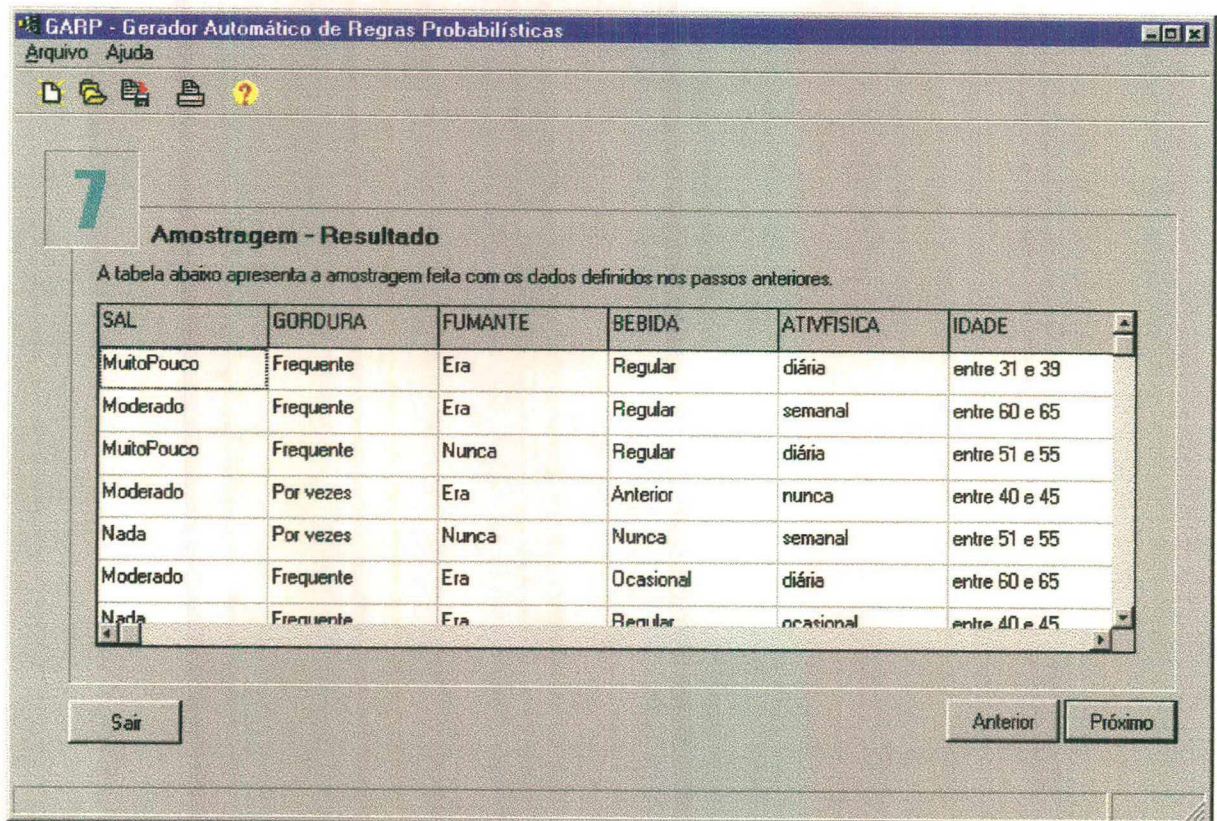


Figura 4.18 – Visualização dos dados de acordo com os passos anteriores

Na visualização da janela o usuário pode ver se na seleção (randômica, sem reposição) dos objetos do conjunto de treinamento ocorreu do sistema selecionar apenas objetos de uma única classe do conceito a ser instruído, pois assim a janela não está adequada para a formação da árvore de indução. No entanto, se isso vier a acontecer o usuário é



informado, devendo retornar ao passo anterior (passo 6) e novamente acionar o botão próximo, gerando desta forma uma nova amostragem. Observe que uma vez discretizados os objetos (passo 5), esses se apresentam nos demais passos (passos 7, 8 e 9) nesta forma discretizada.

### ➔ PASSO 8:

Neste passo o usuário tem a visualização da árvore de classificação gerada pelo processo. A leitura da árvore é feita da raiz (nó mais alto) percorrendo um ramo, descendo pelo nó seguinte, se houver, até chegar a uma folha (final de um ramo). As folhas correspondem as instâncias da classe do nó objetivo. Por exemplo, da árvore exemplificada na figura 4.19 tem-se a seguinte regra: Se ATIVFÍSICA = diária então NÍVEL = Normal. No próximo passo são apresentadas todas as regras possíveis retiradas desta árvore.

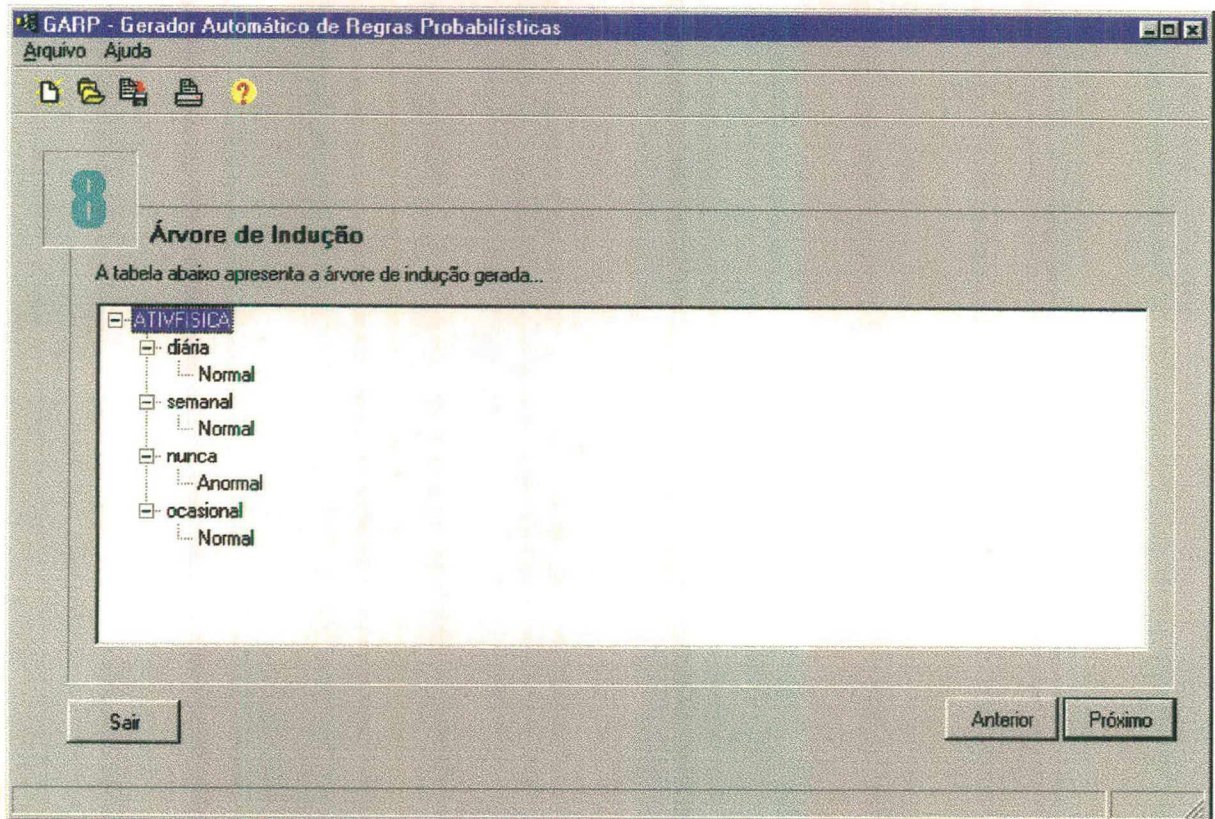


Figura 4.19 – Árvore de decisão por indução



## ➔ PASSO 9

Este é o último passo do processo. O usuário pode visualizar as regras retiradas da árvore de decisão com suas respectivas probabilidades e a confiança da regra, figura 4.20. Chega-se ao objetivo final do processo: regras com suas respectivas probabilidades para formar a base de conhecimento de um sistema especialista. Maiores detalhes sobre a teoria de suporte e confiança ver capítulo 2.

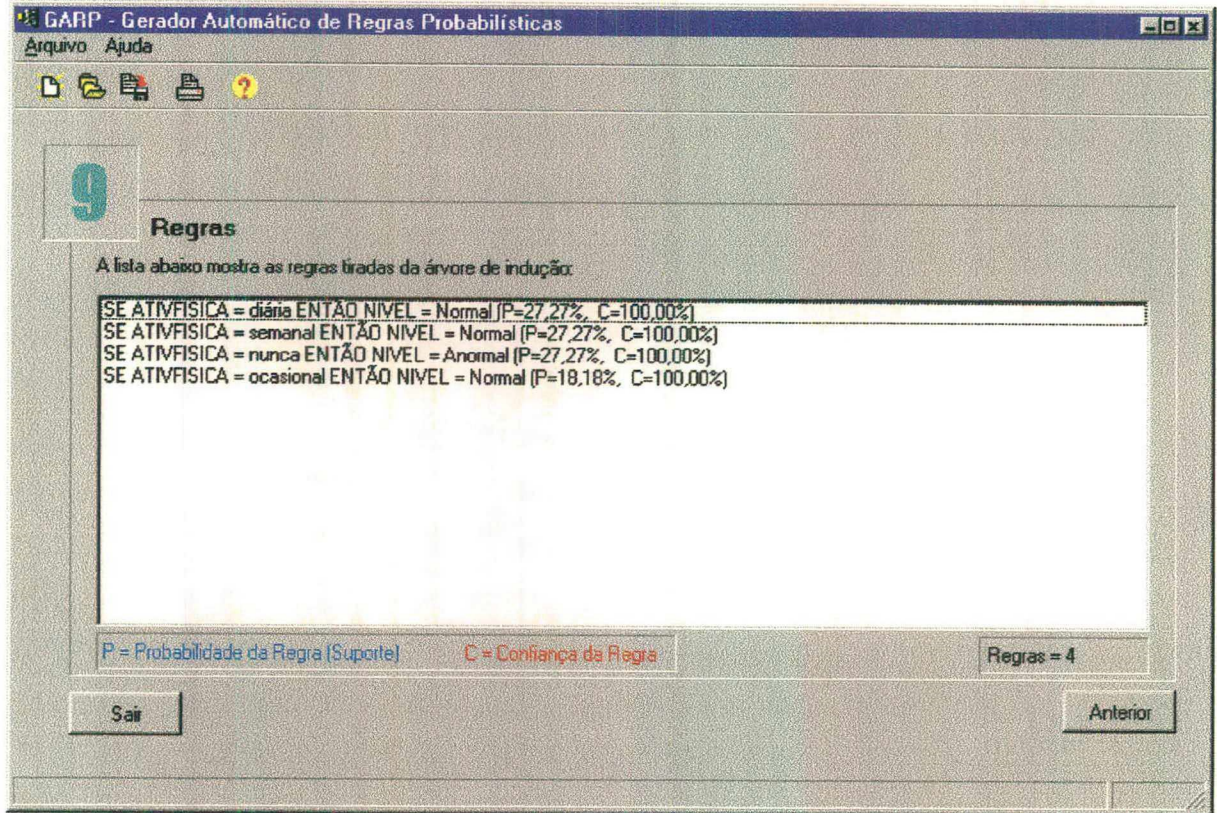


Figura 4.20 – Regras e probabilidades retiradas da árvore de decisão

## 4.5 Constatações sobre a Aplicação

Depois de realizadas várias aplicações da metodologia proposta, através do GARP, chegou-se a algumas constatações referentes ao algoritmo de mineração de dados utilizado e a evidência de que a metodologia pode ser aplicada e validada no ambiente do jogo de empresas GI-EPS.



### 4.5.1 Utilizando Dados Fictícios

---

A metodologia de uso de técnicas de indução para a criação de regras de sistemas especialistas, ora proposta, foi aplicada com êxito em todos os exemplos tomados para teste de base de dados fictícios, apesar deste trabalho mostrar passo a passo apenas uma aplicação (o exemplo do nível de pressão arterial).

Com a utilização do protótipo de software GARP, criado a partir da metodologia proposta para que parte da base de conhecimento seja adquirida automaticamente na forma de regras mais probabilidades associadas, foram realizadas quatro aplicações da metodologia para uma mesma base de dados, tendo o mesmo conjunto de treinamento (tabela 2.1), com o mesmo objetivo de classificação: o diagnóstico de um motor de fábrica, sendo que nestas observou-se algumas constatações referentes ao algoritmo de mineração de dados utilizado, o ID3. Tais constatações são listadas a seguir e partem da seguinte observação: os objetos tomados inicialmente para a janela do algoritmo ID3 podem influenciar na árvore resultante.

- 1) O tamanho da janela nem sempre interfere na árvore resultante, mas sim, quais exemplos (objetos) estão sendo utilizados inicialmente para a geração da árvore de decisão.

Observou-se essa primeira constatação em duas aplicações, onde tomou-se porcentagens iguais a 80 % do conjunto de treinamento inicialmente para a janela, para cada aplicação, e obteve-se árvores diferentes. As janelas iniciais em cada aplicação com as suas respectivas árvores e posteriores regras e probabilidades são apresentadas para a primeira aplicação nas figuras 4.21, 4.22 e 4.23 e para a segunda aplicação nas figuras 4.24, 4.25 e 4.26.

Além de, em outras duas aplicações, uma com janela de 60 % e outra com janela de 100 % do conjunto de treinamento, obteve-se as mesmas regras da segunda aplicação (figura 4.26). Os objetos tomados inicialmente para a janela de 60 % do conjunto de treinamento são apresentados na figura 4.27.



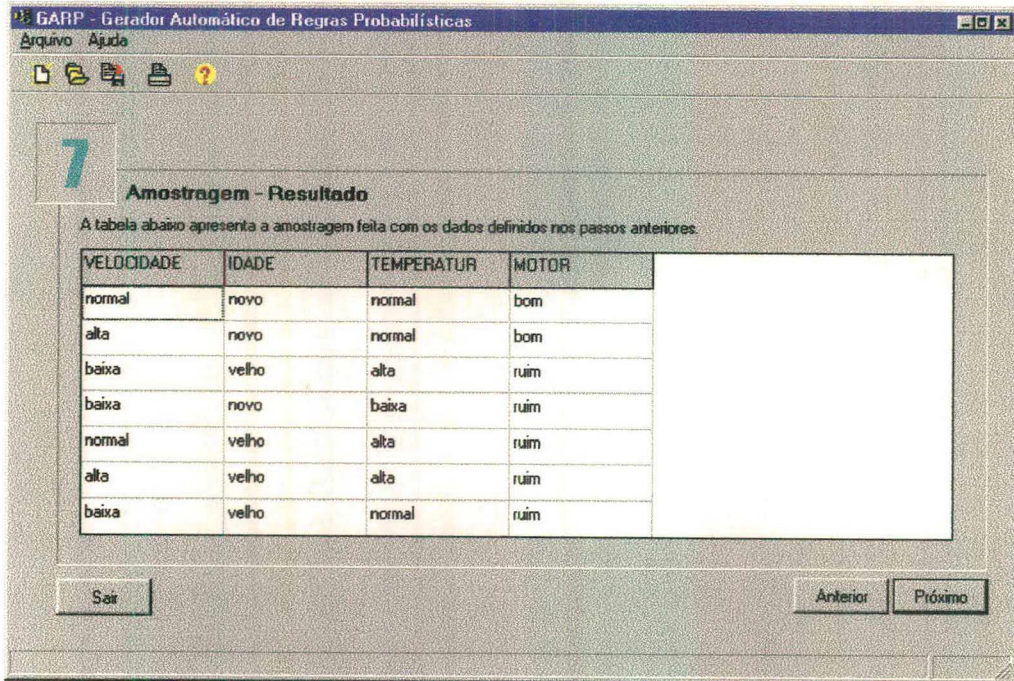


Figura 4.21 – Objetos tomados para a janela inicial de 80 % (primeira aplicação)

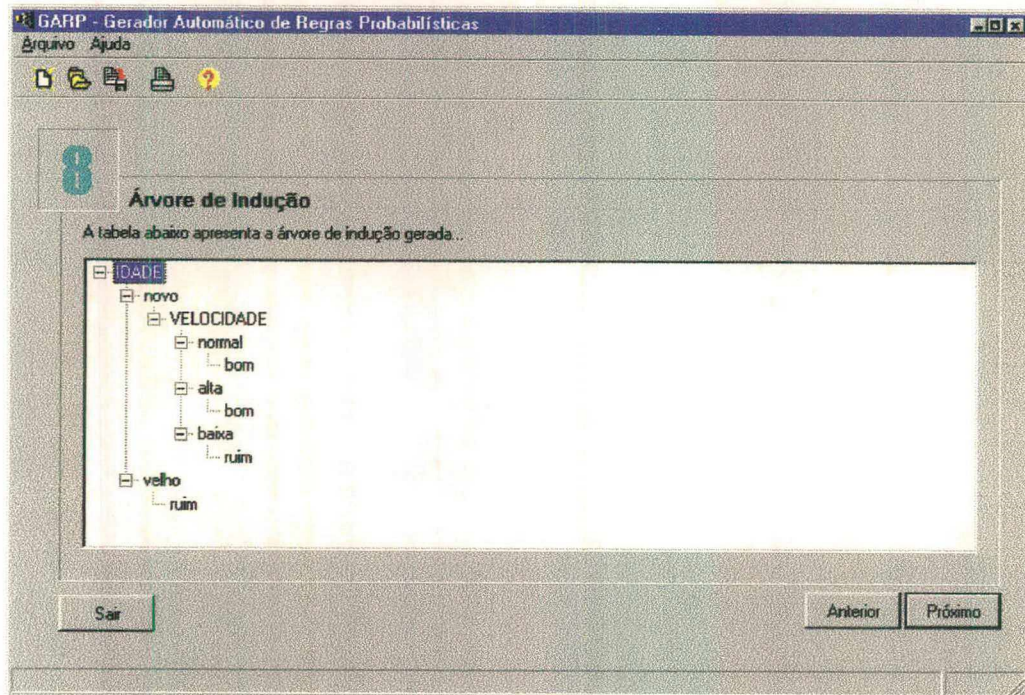


Figura 4.22 – Árvore de decisão por indução gerada (primeira aplicação)



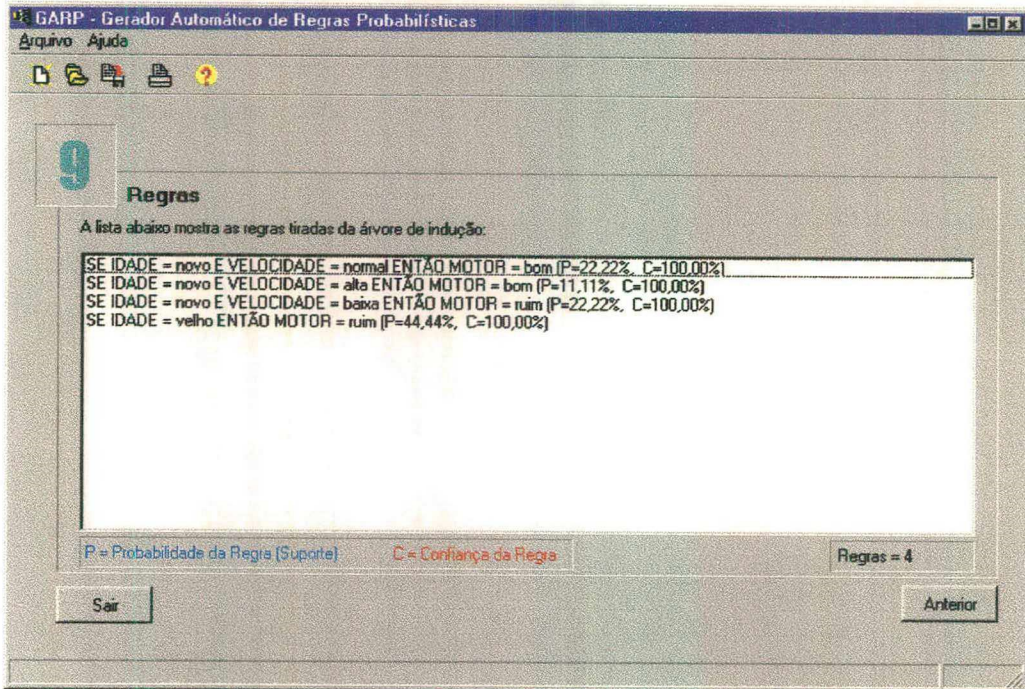


Figura 4.23 – Regras a partir da árvore de indução (primeira aplicação)

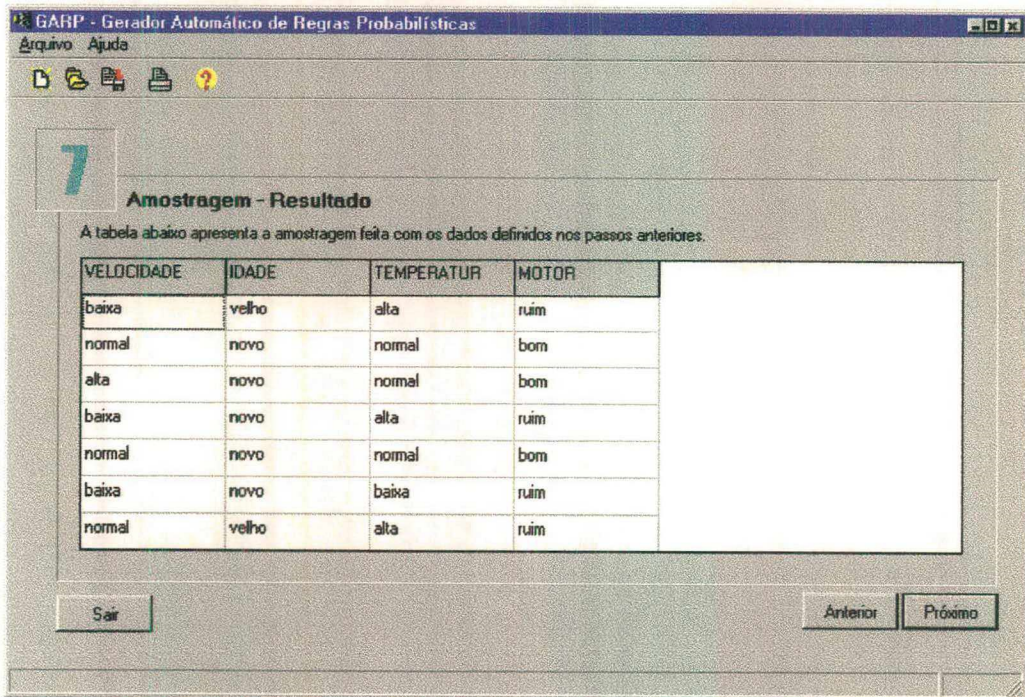


Figura 4.24 - Objetos tomados para a janela inicial de 80 % (segunda aplicação)



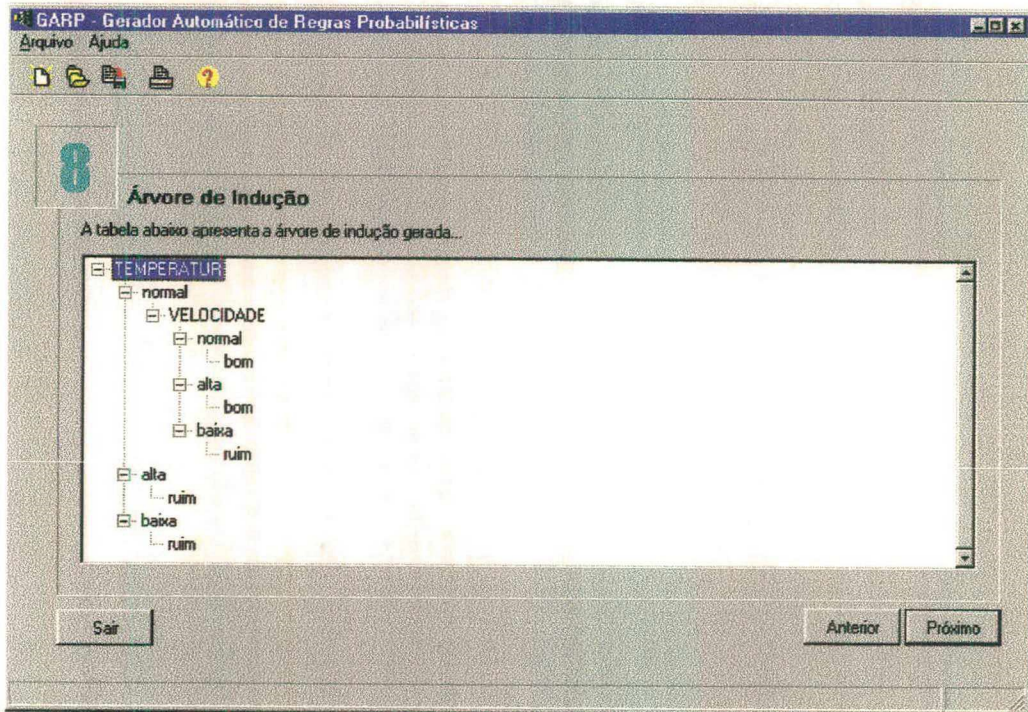


Figura 4.25 – Árvore de decisão por indução gerada (segunda aplicação)

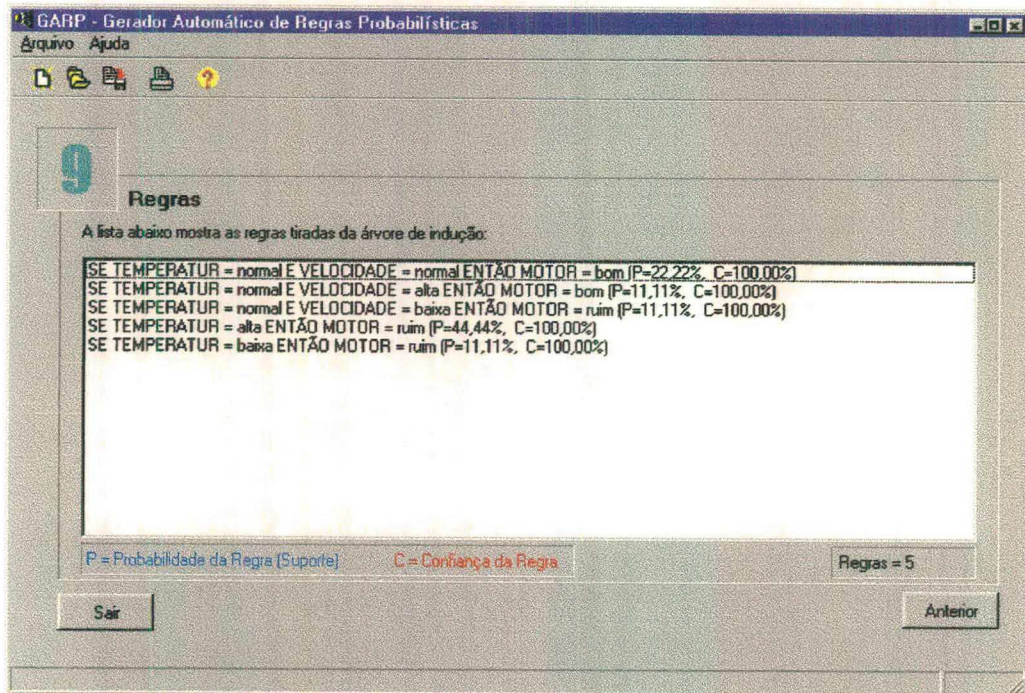
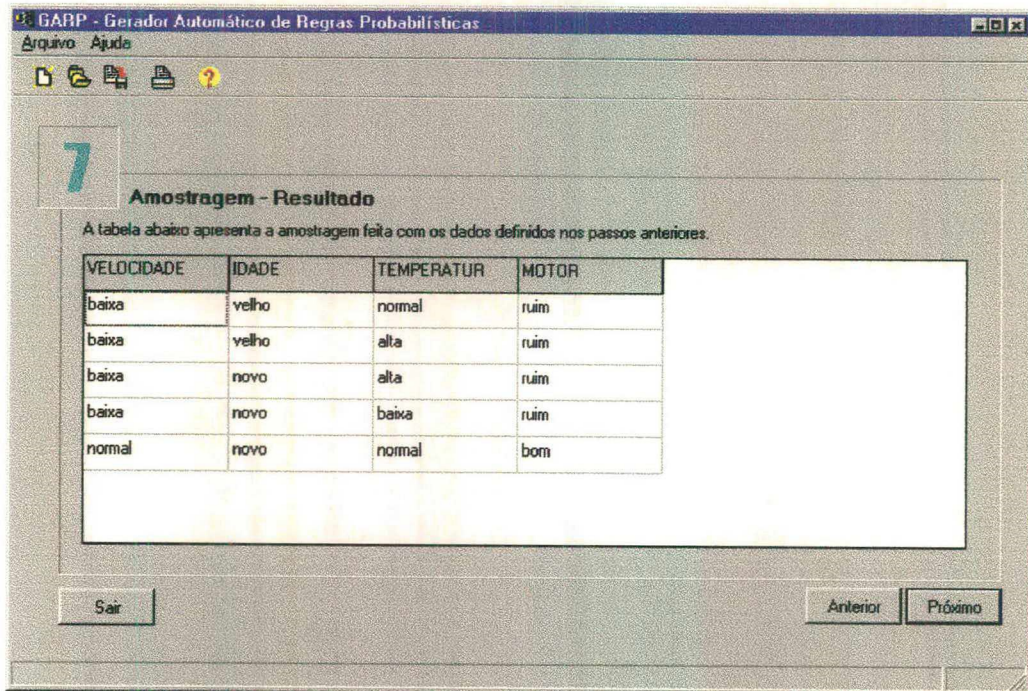


Figura 4.26 – Regras a partir da árvore de indução (segunda aplicação)





**Figura 4.27** – Objetos tomados para a janela inicial de 60 % (terceira aplicação)

- 2) Dependendo dos objetos tomados na janela inicial para a formação da árvore de indução, o algoritmo ID3 pode gerar árvores de complexidades diferentes.

A segunda constatação pode ser observada nas figuras 4.22 e 4.24. Pois a árvore da figura 4.22 é menor (mais simples) que a árvore da figura 4.24, sendo que da árvore da figura 4.22 extrai-se 4 regras (figura 4.23) e da árvore da figura 4.23 extrai-se 5 regras (figura 4.26).

- 3) Os objetos tomados para a janela inicial para a formação da árvore de indução, através do algoritmo ID3, podem interferir na seleção de qual atributo é considerado irrelevante e qual atributo é considerado o mais informativo, ambos para a tarefa de classificação, ou seja, para a geração da árvore de decisão.

Essa terceira constatação pode ser averiguada nas figuras 4.22 e 4.25, onde para a formação da árvore de decisão da figura 4.22 foram tomados para a janela inicial os objetos da figura 4.21, sendo o atributo TEMPERATURA considerado irrelevante para a tarefa de classificação, uma vez que esse não aparece na árvore e o atributo IDADE considerado o mais informativo, ou seja, o atributo de maior importância para a tarefa de classificação, uma vez que esse consiste da raiz da árvore de indução (primeiro atributo do topo da árvore). Já para a formação da árvore da figura 4.25 foram tomados para a janela inicial os objetos da figura 4.24 (por exemplo, segunda aplicação), sendo o atributo IDADE considerado irrelevante para



a tarefa de classificação, uma vez que esse não aparece na árvore e o atributo TEMPERATURA considerado o mais importante, uma vez que esse se encontra na raiz da árvore.

Com o exposto nas constatações mencionadas, conclui-se que apesar de terem sido encontradas duas árvores diferentes (figura 4.22 e figura 4.25) para o mesmo conjunto de treinamento, com a mesma tarefa de classificação, ambas classificam corretamente todos os objetos do conjunto de treinamento, logo, ambas as árvores estão corretas para o conjunto de treinamento.

A diferença encontrada nos conjuntos de regras, ou seja, duas árvores de decisão diferentes para o mesmo conjunto de treinamento e para a mesma tarefa de classificação, se dá pelo fato do funcionamento do algoritmo ID3, onde a seleção de objetos para a janela ocorre de forma randômica e primeiramente é formada a árvore de indução para a janela, se esta classifica corretamente todos os demais exemplos do conjunto de treinamento o algoritmo termina, e caso isso não ocorra, são acrescentados à janela uma seleção dos exemplos não classificados corretamente, aumentando o tamanho desta (a janela pode crescer até atingir o tamanho do conjunto de treinamento inteiro), e uma nova árvore de decisão é formada.

Logo, pode ocorrer dos objetos selecionados aleatoriamente para a janela inicial, ou seja, para a formação da árvore de decisão, já serem suficientes para a árvore formada classificar corretamente todos os objetos do conjunto de treinamento, e em outra seleção destes não.

A seguir é mostrada a constatação sobre o ID3 referente ao caso especial apresentado no capítulo 2 item 2.3.1.

Em uma aplicação do GARP para o conjunto de objetos da tabela 4.1, tendo como objetivo de classificação o atributo greve com os valores de classe (sim, não), foram geradas as regras da figura 4.28.



Tabela 4.1 – Exemplos de acionamento de greve

DIASEM	GREVE	HORA	TEMPO	FAIXA5	ENTPO
DOMINGO	NAO	16-21	CHUV	SUL	DENSO
DOMINGO	NAO	21-7	SOL	FECHADA	FLUIDO
SEXTA	SIM	21-7	CHUV	SUL	FLUIDO
SEXTA	SIM	16-21	NUB	NORTE	FILA-PEQ
SEXTA	NAO	10-16	CHUV	SUL	DENSO
SEXTA	NAO	7-10	NUB	NORTE	FILA-GR
OUTRA	SIM	7-10	SOL	FECHADA	FILA-GR
OUTRA	SIM	10-16	NUB	SUL	FILA-PEQ
OUTRA	NAO	16-21	SOL	FECHADA	DENSO
OUTRA	NAO	21-7	CHUV	NORTE	FLUIDO

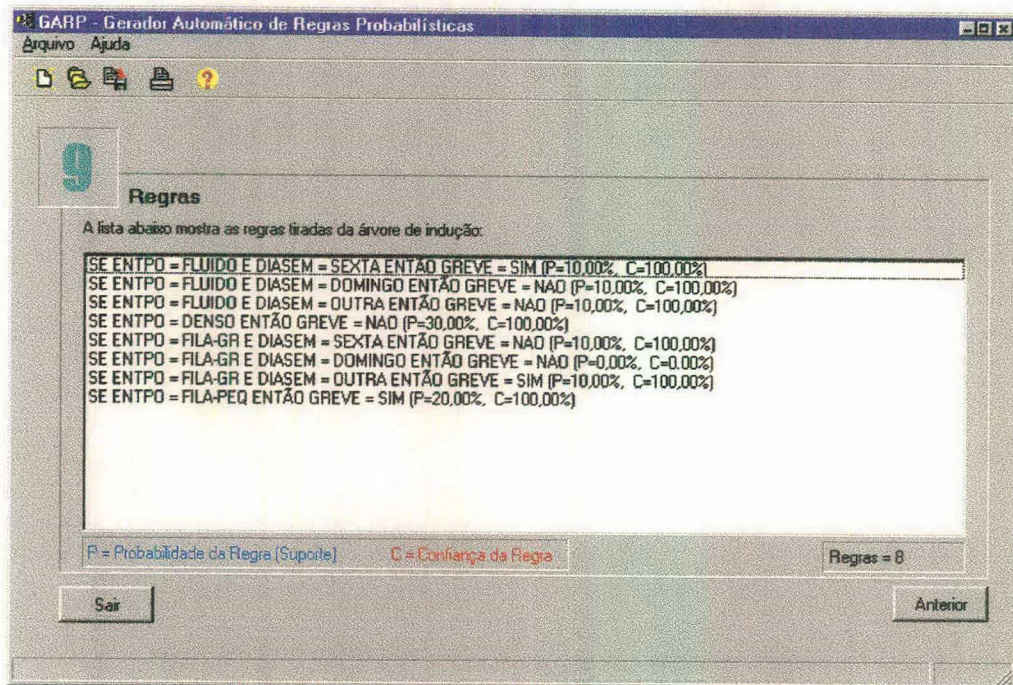


Figura 4.28 – Regras geradas pelo ID3 (caso especial)

Observa-se nessa aplicação que o ID3 gerou uma regra com probabilidade zero e com confiança também zero, ou seja, uma regra onde não existe pelo menos um objeto no conjunto de treinamento que a satisfaça. Apesar das regras classificarem corretamente todos os objetos do conjunto de treinamento, e desta forma, a árvore de decisão gerada estar correta para o mesmo.

Esse fato ocorreu devido ao atributo diasem ter sido escolhido pelo ID3 como raiz da subárvore gerada para  $C_2$  e esse não conter pelo menos um objeto com o valor domingo, assim constatou-se o caso especial relatado no capítulo 2 na seção 2.3.1 e que esse leva a regra com probabilidade igual a zero e confiança também igual a zero, além de que, isso



surgiu em uma subárvore, ou seja, de um subconjunto do conjunto de treinamento, durante o processo de geração da árvore de decisão.

Como o cálculo da probabilidade das regras faz parte da metodologia proposta, caso o ID3 gere uma regra com probabilidade e confiança iguais a zero, isso é um alerta para o especialista do domínio que para tal regra não existe objeto no conjunto de treinamento que a contemple.

## 4.5.2 Utilizando Dados do GI-EPS

Neste tópico será evidenciado que o sistema desenvolvido pode ser aplicado e validado no ambiente do jogo de empresas GI-EPS.

A base de dados a ser analisada contém um conjunto de decisões de uma determinada empresa, em diversos períodos, numa determinada aplicação real do Jogo de Empresas – GI-EPS com um total de 1808 registros. Para efeito didático, na tabela 4.2, são mostrados alguns exemplos destas decisões, porém a aplicação do GARP será realizada sobre todos os registros. Cada linha corresponde a um objeto (uma decisão da empresa) e cada coluna corresponde a um atributo com os seus respectivos valores (uma característica) em relação ao objeto.

**Tabela 4.2 - Exemplos dos dados analisados (baseGIP.dbf)**

PRECO	PROP	PRAZO	SAZON	CONJUNT	MERC	NEMP	VDEM
5,71	4,00	10,00	0,30	0,00	3,48	6	-9,27
5,71	4,00	30,00	0,30	0,00	3,48	6	-8,13
5,71	2,75	45,00	0,30	0,00	3,48	6	-19,50
8,57	4,00	30,00	0,30	0,00	3,48	6	-15,20
3,89	4,00	30,00	0,30	0,00	3,48	6	-3,20
4,29	4,00	12,00	0,30	0,00	3,48	6	2,84
0,00	4,90	10,00	0,30	0,00	3,48	6	27,26
-16,07	9,64	58,00	0,30	0,00	7,19	9	37,28
-10,47	9,08	60,00	0,30	0,00	7,19	9	9,52
2,65	9,37	64,00	0,30	0,00	7,19	9	-33,35
-1,63	8,44	60,00	0,30	0,00	7,19	9	-19,63
-1,77	9,02	58,00	0,30	0,00	7,19	9	-16,83
0,00	9,57	34,00	0,30	0,00	7,19	9	-27,81
-8,33	9,71	46,00	0,30	0,00	7,19	9	-5,60
-24,43	7,92	20,00	0,30	0,00	7,19	9	119,81



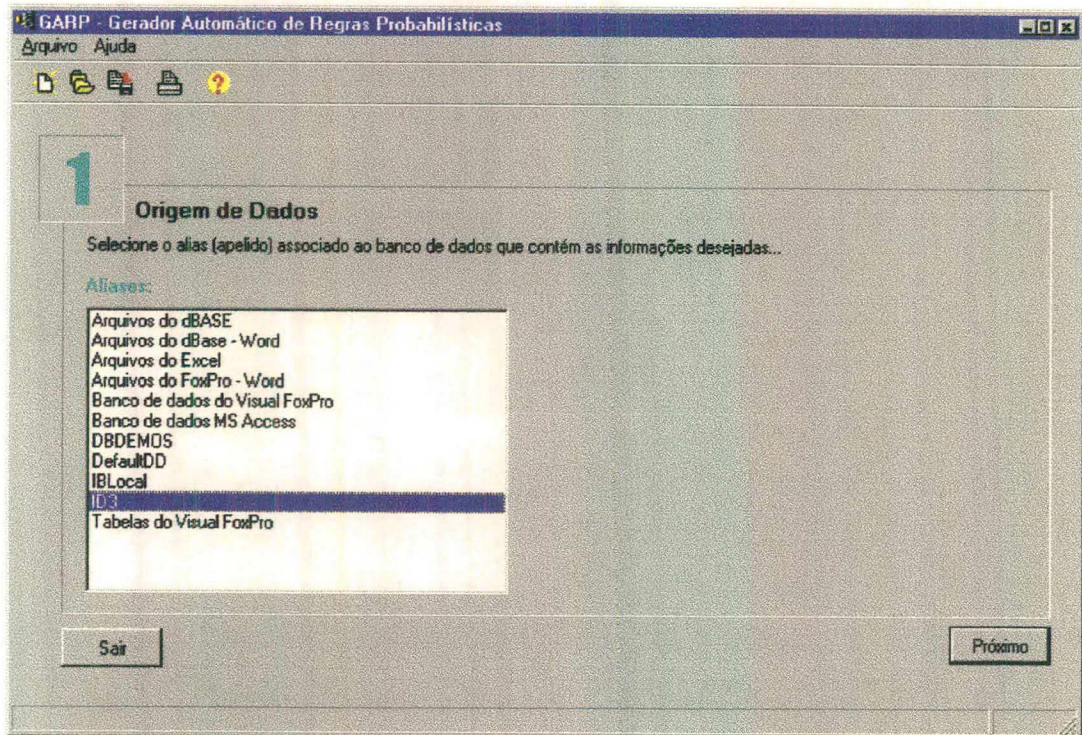
Dentro do contexto da simulação, a tabela 4.3 apresenta a descrição dos atributos utilizados.

**Tabela 4.3 – Descrição dos dados analisados (baseGIP.dbf)**

Atributo	Descrição
PRECO	Variação do preço de venda do produto
PROP	Número de módulos de propaganda
PRAZO	Prazo dado aos clientes na compra do produto
SAZON	Sazonalidade
CONJUNT	Conjuntura econômica
MERC	Crescimento do mercado consumidor
NEMP	Número de empresas participantes
VDEM	Variação da demanda

A seguir são apresentados os passos da aplicação da metodologia proposta através do GARP, utilizando todos os 1808 registros da base de dados do GI-EPS, tendo como objetivo identificar o que determina (quais regras e probabilidades associadas) uma variação de demanda positiva e uma variação de demanda negativa num período de sazonalidade.

➔ PASSO 1:



**Figura 4.29 – Seleção da base de dados da aplicação**



➔ PASSO 2:

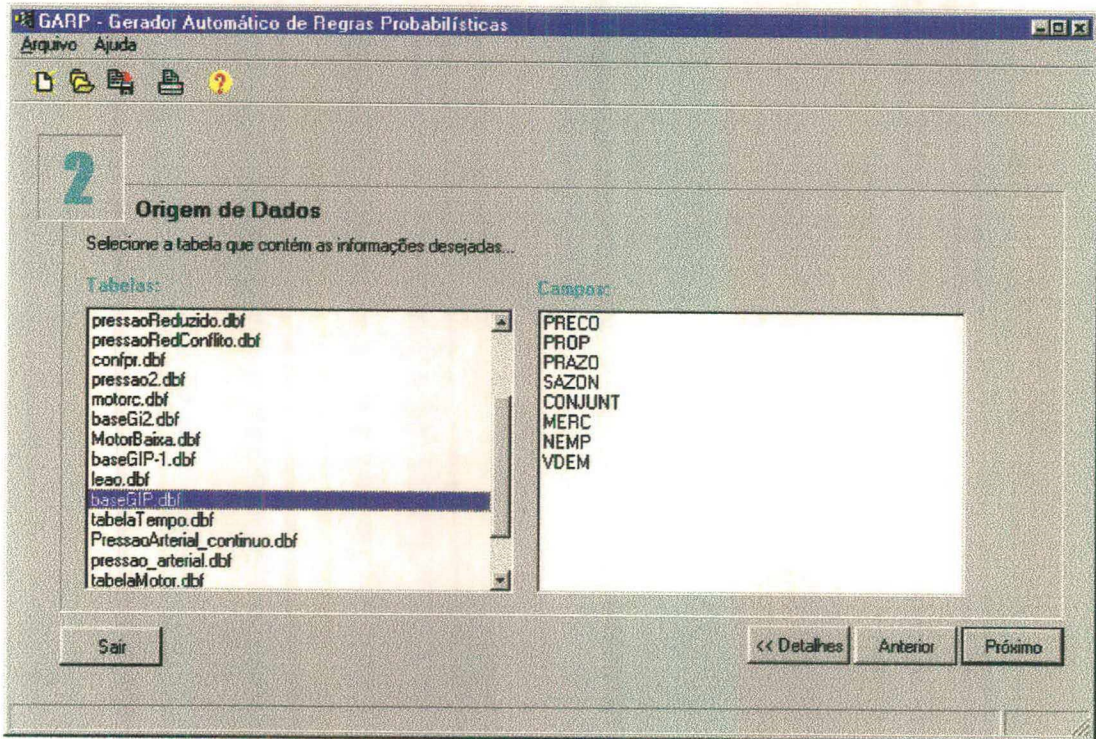


Figura 4.30 – Seleção da tabela que contém as informações desejadas

➔ PASSO 3:

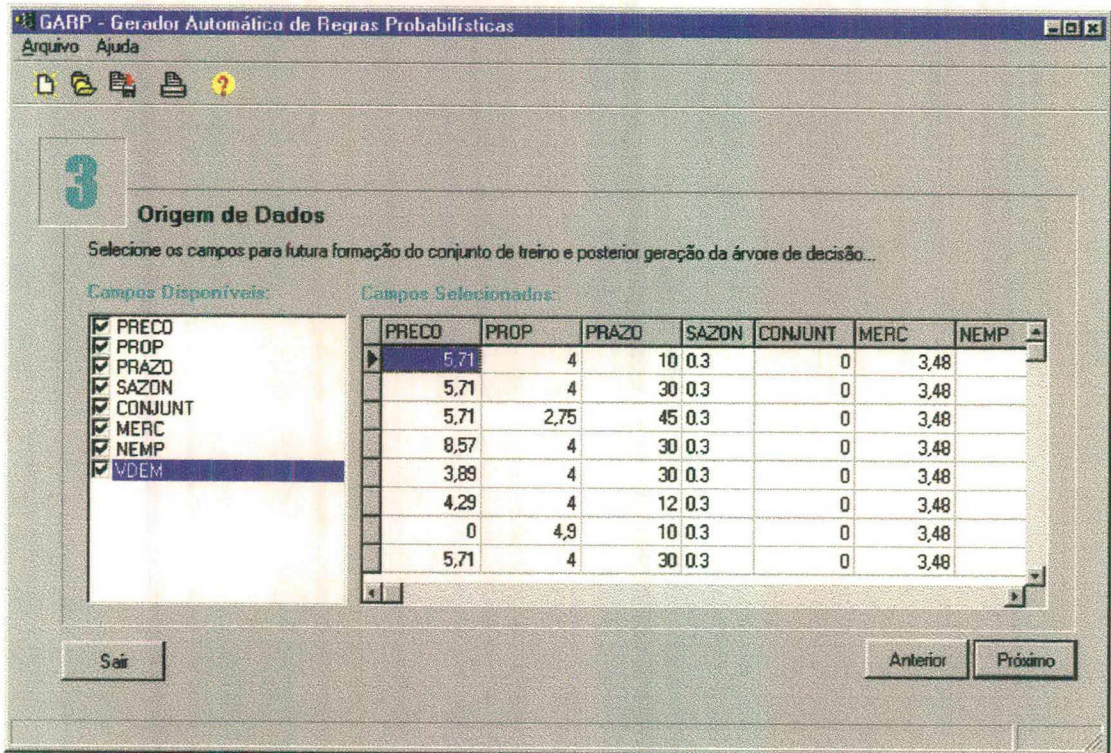


Figura 4.31 – Seleção dos campos que se deseja



➤ PASSO 4:

Como o objetivo da tarefa de classificação é identificar o que determina uma variação de demanda positiva e uma variação de demanda negativa num período de sazonalidade (atributo SAZON igual a 0,7), a figura 4.32 mostra o procedimento para a seleção destes objetos.

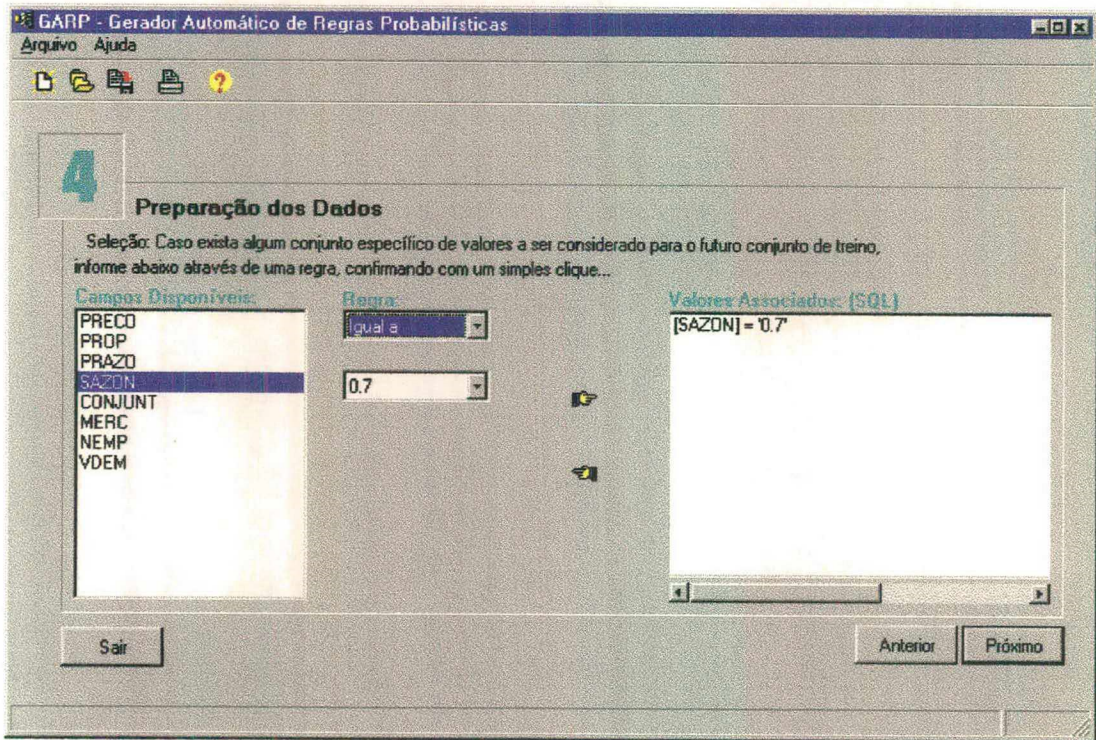


Figura 4.32 – Seleção dos dados de período sazonal

➤ PASSO 5:

Como o objetivo da tarefa de classificação é identificar o que determina uma variação de demanda positiva e uma variação de demanda negativa num período de sazonalidade, tem-se que discretizar o atributo VDEM nos valores de classes associados: positiva e negativa. A figura 4.33 mostra como proceder para tal.



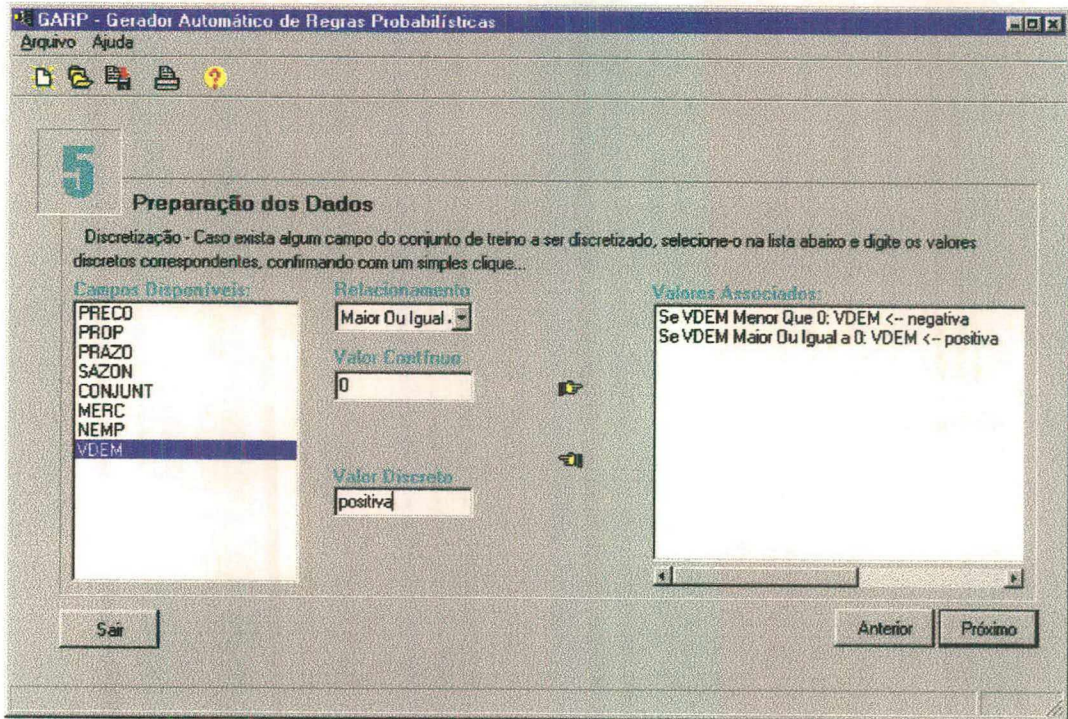


Figura 4.33 – Discretização dos atributos (criação de classes)

➤ PASSO 6:

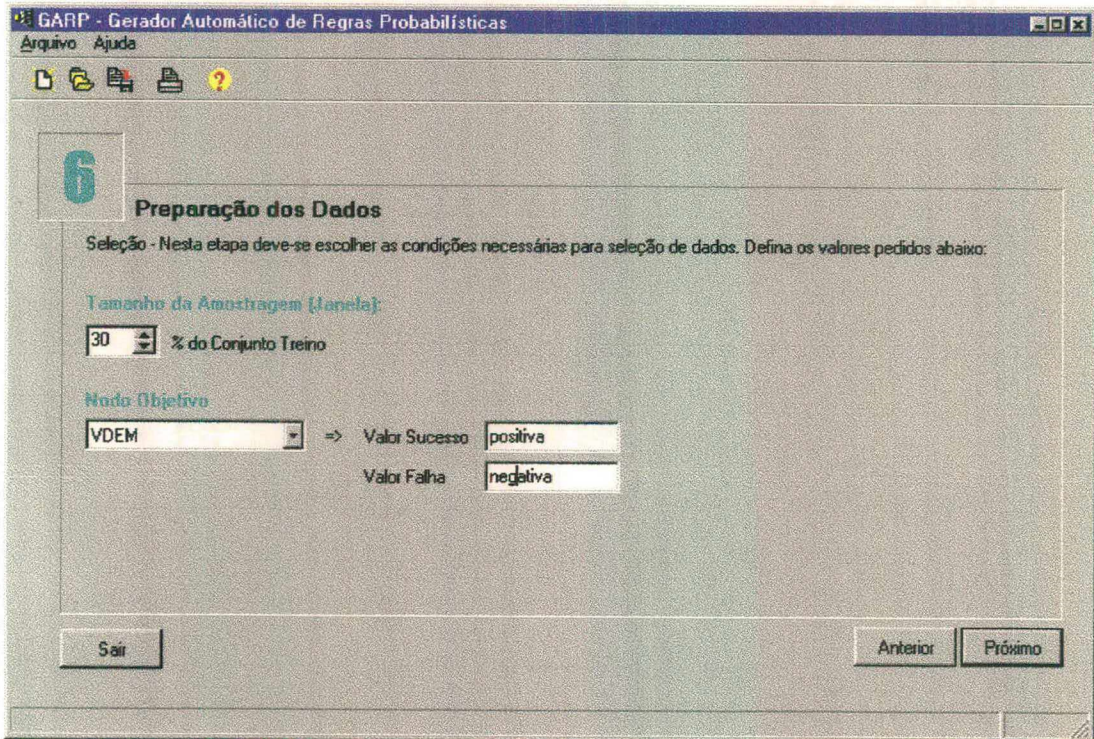
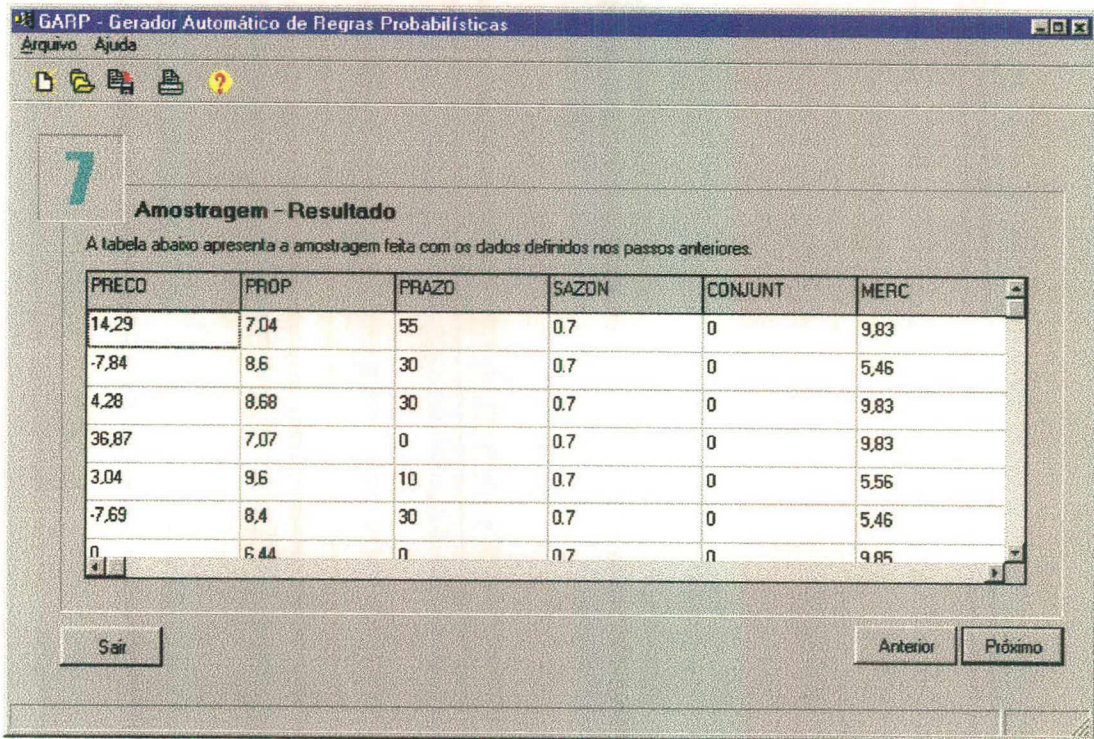


Figura 4.34 – Escolha do conceito a ser instruído e tamanho inicial da janela.



➔ PASSO 7:



A tabela abaixo apresenta a amostragem feita com os dados definidos nos passos anteriores.

PRECO	PROP	PRAZO	SAZON	CONJUNT	MERC
14,29	7,04	55	0,7	0	9,83
-7,84	8,6	30	0,7	0	5,46
4,28	8,68	30	0,7	0	9,83
36,87	7,07	0	0,7	0	9,83
3,04	9,6	10	0,7	0	5,56
-7,69	8,4	30	0,7	0	5,46
n	6,44	n	0,7	n	9,85

Figura 4.35 – Visualização dos dados de acordo com os passos anteriores

➔ PASSO 8:

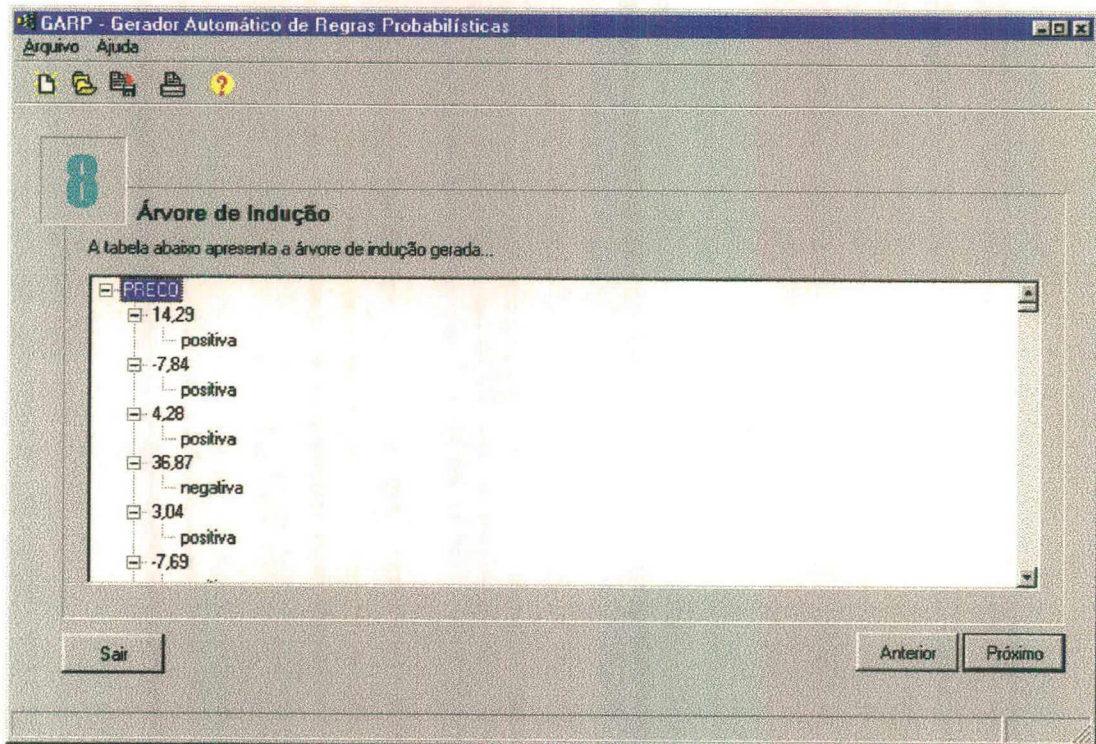
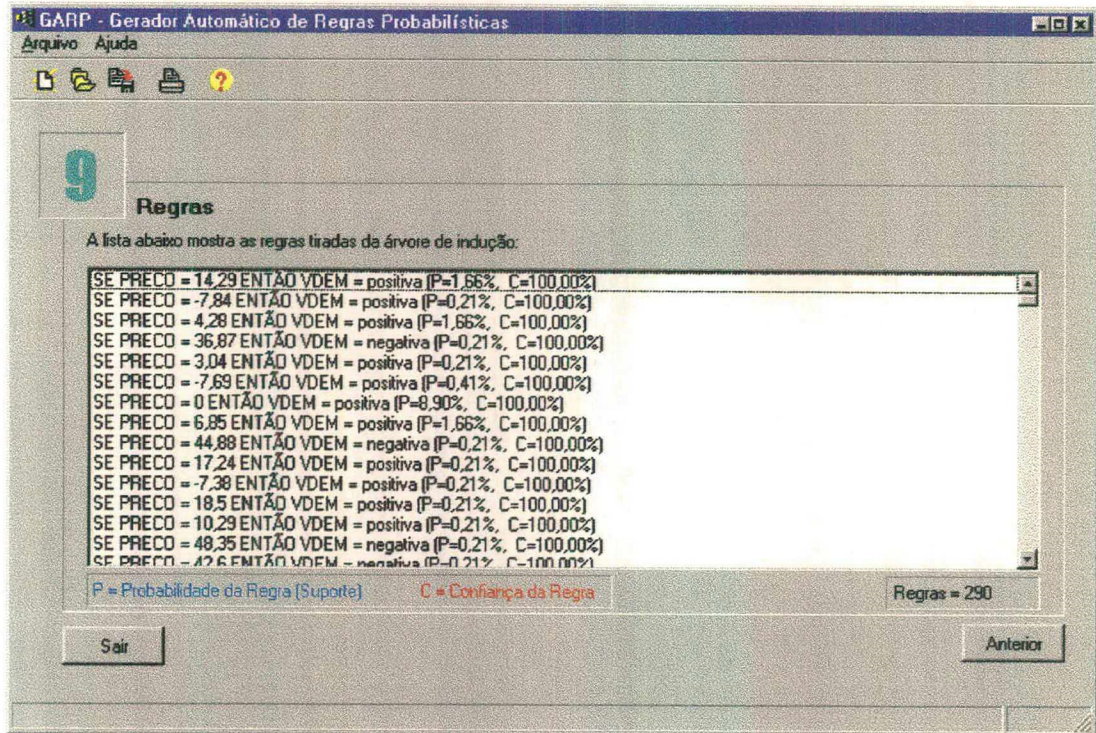


Figura 4.36 – Árvore de decisão por indução (parcial)



➤ PASSO 9:

Dentre todas as regras (290) geradas pelo GARP, o único atributo que este considerou importante para a tarefa de classificação foi o atributo PRECO. Todos os demais atributos foram considerados irrelevantes para a tarefa de classificação. Algumas das regras geradas são apresentadas na figura 4.37.



**Figura 4.37** – Regras e probabilidades retiradas da árvore de decisão

Sendo assim, evidencia-se que a metodologia proposta pode ser aplicada e validada no ambiente do jogo de empresas GI-EPS.



## 5. CONCLUSÃO

---

*Neste capítulo são apresentadas as conclusões deste trabalho em relação aos objetivos inicialmente propostos, além de serem apontadas direções de pesquisas encontradas durante o desenvolvimento deste.*

---

### 5.1 Conclusões

---

A metodologia proposta neste trabalho cumpriu com o objetivo de gerar conhecimento na forma de regras mais a probabilidade associada a cada regra, através de indução de árvores de decisão, além dessa sistematizar as informações, organizando-as de forma clara e simples, como também, formalizando-as, registrando todas as etapas realizadas, facilitando assim a avaliação das ações relacionadas às informações pelos decisores.

A participação do especialista do domínio é de fundamental importância, uma vez que este analisa as diferentes regras geradas pelo sistema, e opta pela utilização do conjunto de regras (árvore de indução) que melhor lhe convier para o domínio da aplicação do sistema especialista em questão. Recomenda-se a realização de algumas aplicações da metodologia a um mesmo domínio (conjunto de treinamento), tomando-se porcentagens diferentes para a janela, pois se constatou que o ID3 é capaz de gerar árvores diferentes para um mesmo conjunto de treinamento numa mesma tarefa de classificação.

No que se refere ao protótipo de software desenvolvido e utilizado neste trabalho, ou seja, o GARP, é inquestionável a facilidade que o mesmo proporciona, tanto em termos de eliminação de cálculos matemáticos que seriam feitos manualmente, bem como, o tempo que se iria despendar para efetuá-los, quanto da construção por indução das árvores de decisão e de suas regras, facilitando assim a compreensão dos padrões extraídos da base de dados analisada. A sua construção permite que o conhecimento gerado possa ser utilizado em

qualquer shell de sistema especialista em que haja compatibilidade com a base de conhecimentos (regra e probabilidade) gerada pela ferramenta. Além da simplicidade e da clareza de sua interface também poder ser considerada um dos pontos fortes deste trabalho.

Na aplicação em um caso real do jogo de empresas GI-EPS, a metodologia proposta apresentou uma boa performance na obtenção dos resultados, porém testes, aplicações e validações do GARP devem ser ratificados em trabalhos futuros como proposto inicialmente no objetivo deste trabalho.

Como ponto fraco, cita-se que o GARP está inicialmente limitado a trabalhar com apenas duas classes no conceito a ser instruído, já que em uma aplicação real, isso limita as possibilidades do especialista encontrar padrões nos dados a serem analisados. No entanto, a teoria apresentada pelo ID3 pode ser estendida para trabalhar com mais de duas classes.

## 5.2 Sugestões para Trabalhos Futuros

---

A partir do desenvolvimento deste, surgiram algumas direções de investigações possíveis de serem tomadas como sugestões para trabalhos futuros:

- ⊗ Tratamento de conflito e dados incompletos. O sistema ID3 pode suportar tais tarefas (Quinlan, 1985).
- ⊗ Acrescentar na metodologia a possibilidade de através desta realizar-se a descoberta de conhecimento fazendo-se uso ao mesmo tempo de mais de uma tabela de dados.
- ⊗ Acrescentar métodos de discretização automática.
- ⊗ Acrescentar novos algoritmos de mineração, para a tarefa de aprendizagem de conceitos.
- ⊗ Possibilitar a aprendizagem de conceitos para mais de duas classes, estendendo para tal a teoria apresentada pelo sistema ID3.
- ⊗ Análise detalhada da compatibilidade da metodologia proposta com a *shell* SPIRIT – utilizada para o desenvolvimento de sistemas especialistas probabilísticos (Rödder, 1995).
- ⊗ Possibilitar a importação das regras e probabilidades geradas de forma automatizada a *shell* SPIRIT.
- ⊗ Estudo detalhado das implicações do uso desta metodologia para a geração de regras de sistemas especialistas, fazendo comparações com sistemas especialistas baseados em regras já existentes.
- ⊗ Testar, aplicar e validar o sistema desenvolvido no ambiente do jogo de empresas GI-EPS.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABT, Clark C. **Jogos Simulados: Estratégia e Tomada de Decisão**. Rio de Janeiro: J. Olympio, 1974.

AGRAWAL, Rakesh; IMIELINSKI, Tomasz; SWAMI, Arun. **Mining Association Rules between Sets of Items in Large Databases**. Proceedings of the ACM SIGMOD Conference Washington DC, USA, May 1993. Disponível em <http://citeseer.nj.nec.com/agrawal93mining.html>

ALI, Kamal; MANGANARIS, Stefanos; SRIKANT, Ramakrishnan. **Partial Classification using Association Rules**. AAI, 1997. Disponível em <http://citeseer.nj.nec.com/ali97partial.html>

BISPO, Carlos Alberto F; CAZARINI, Edson Walmir. Análises Sofisticadas com o On-Line Analytical Processing. **Developers Magazine**. Ano 3, n. 32, p. 28–31, Abr. 1999.

\_\_\_\_\_. Transformando Dados em Informações Via Data Mining. **Developers Magazine**. Ano 3, n. 29, p. 36–38, Jan. 1999.

CHEN, Ming-Syan; HAN, Jiawei; YU, Philip S. **Data Mining: An Overview from Databases Perspective**. 1996. Disponível em <http://citeseer.nj.nec.com/5126.html>

CIELO, Ivã Rafael. Como a TI pode Contribuir com a sua Empresa. Site da DW Brasil. Visitado em 29 Jan. 2001. [http://www.datawarehouse.inf.br/bi/body\\_bi.html](http://www.datawarehouse.inf.br/bi/body_bi.html)

CIELO, Ivã Rafael; PAZ, Luiz Cláudio. Data Mining. Site da DW Brasil. Visitado em 29 Jan. 2001. [http://www.datawarehouse.inf.br/mining/body\\_mining.html](http://www.datawarehouse.inf.br/mining/body_mining.html)

\_\_\_\_\_. Data Warehouse. Site da DW Brasil. Visitado em 29 Jan. 2001. [http://www.datawarehouse.inf.br/dw/body\\_dw.html](http://www.datawarehouse.inf.br/dw/body_dw.html)



CIENTISTAS do Projeto Genoma Usam Tecnologia de Data Mining. Site da IT Web Brasil.  
**IT Mídia Ltda.** Publicado em 03 Ago. 2000.  
<http://www.itweb.com.br/empresas/noticias/artigo.asp?id=5902>

DALFOVO, Oscar; GRIPA, Robson. Data Warehouse: Usando a Técnica de Cubo de Decisão. **Developers Magazine**. Ano 3, n. 32, p. 12–17, Abr. 1999.

DENIS, François; GILLERON, Rémi. **Apprentissage à partir d'Exemples**. Notes de cours. Université Charles de Gaulle, Lille 3. Avril, 2000.

DETTMER, Armando Luiz. **Concebendo um Laboratório de Engenharia de Produção Utilizando um Jogo de Empresas**. Florianópolis, 2001. Tese (Doutorado em Engenharia de Produção) – Curso de Pós Graduação em Engenharia de Produção. Universidade Federal de Santa Catarina – UFSC.

DURKIN, John. Designing an Induction Expert System. **AI Expert**. p. 29–35. Dec. 1991.

\_\_\_\_\_. **Expert Systems: Design and Development**. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1994.

FAYYAD, U. M. et al. From Data Mining to Knowledge Discovery: An Overview. In: FAYYAD, U. M. et al. **Advances in Knowledge Discovery and Data Mining**. Menlo Park, CA: AAAI Press, 1996.

FAYYAD, Usama; HAUSSLER, David; STOLORZ, Paul. **KDD for Science Data Analysis: Issues and Examples**. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, Aug. 1996, AAAI Press. Disponível em <http://citeseer.nj.nec.com/fayyad96kdd.html>

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. **Knowledge Discovery and Data Mining: Towards a Unifying Framework**. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, Aug. 1996, AAAI Press. Disponível em <http://citeseer.nj.nec.com/fayyad96knowledge.html>

FELDENS, Miguel Artur. **Engenharia da Descoberta de Conhecimento em Base de Dados: Estudo e Aplicação na Área de Saúde**. Porto Alegre, 1997. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós Graduação em Ciência da Computação. Universidade Federal do Rio Grande do Sul – UFRGS.

FRAWLEY, William J; PIATETSKY-SHAPIO, Gregory; MATHEUS, Christopher J. Knowledge Discovery in Databases: An Overview. AAAI Press, Menlo Park, California, **AI MAGAZINE**. p. 57–70. FALL 1992.

GESTWICKI, Paul. **ID3: History, Implementation, and Applications**. Oct. 1997. Disponível em <http://citeseer.nj.nec.com/398697.html>

GONÇALVES, Alexandre Leopoldo. **Utilização de Técnicas de Mineração de Dados em Bases C&T: Uma Análise dos Grupos de Pesquisa no Brasil**. Florianópolis, 2000. Dissertação (Mestrado em Engenharia de Produção) – Curso de Pós Graduação em Engenharia de Produção. Universidade Federal de Santa Catarina – UFSC.

GUROVITZ, Helio. O que Cerveja tem a Ver com Fraldas? Não é charada. Entenda como o Armazém de Dados pode Aumentar a Produtividade a partir de Informações Aparentemente Desconexas. **Info Exame**. Abr. 1997. Disponível em: <http://www2.uol.com.br/info/exame/armaz.html>.

HAN, Jiawei; CAI, Yandong; CERCONE, Nick. **Knowledge Discovery in Databases: An Attribute-Oriented Approach**. Proceedings of the 18<sup>th</sup> VLDB Conference Vancouver, British Columbia, Canada, 1992. Disponível em: <http://citeseer.nj.nec.com/hah92knowledge.html>

HERMENEGILDO, Jorge Luiz Silva. **A Utilização da Padronização como Ferramenta da Qualidade Total para o Desenvolvimento de Jogos de Empresas**. Florianópolis, 1996. Dissertação (Mestrado em Engenharia de Produção) – Curso de Pós Graduação em Engenharia de Produção. Universidade Federal de Santa Catarina – UFSC.

HOLSHEIMER, Marcel; SIEBES, Arno. **Data Mining: The Search for Knowledge in Databases**. Amsterdam, The Netherlands, 1991. Disponível em <http://citeseer.nj.nec.com/holsheimer91data.html>

INMON, W. H. **Como Construir o Data Warehouse**. 2. ed. Rio de Janeiro: Campus, 1997.

KOPITTKE, Bruno Hartmut. **GI-EPS: Manual do Jogador**. Florianópolis, 1989. Departamento de Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina – UFSC.

\_\_\_\_\_. **Jogos de Empresas: Novos Desenvolvimentos**. Florianópolis, 1991. (Publicação interna). Departamento de Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina – UFSC.

KOPITTKE, Bruno Hartmut; DETTMER, Armando Luiz; GARTNER, Ivan Ricardo. **Um Sistema Inteligente de Apoio à Decisão Estratégica Baseado em Inferência Probabilística**. 7<sup>o</sup> Congresso Nacional de Investigação Operacional Aveiro, Portugal. Abr. 1996.

LANGLEY, Pat; SIMON, Herbert A. **Applications of Machine Learning and Rule Induction**. 1995. Disponível em <http://citeseer.nj.nec.com/109872.html>

LEVINE, Robert I. **Inteligência Artificial e Sistemas Especialistas**. São Paulo: McGraw-Hill, 1988.

MANNILA, Heikki. **Data Mining: Machine Learning, Statistics, and Databases**. Eight International Conference on Scientific and Statistical Database Management, Stockholm, June, 1996, p. 1-8. Disponível em <http://citeseer.nj.nec.com/52294.html>

\_\_\_\_\_. **Methods and Problems in Data Mining**. Proceedings of the International Conference on Database Theory, Delphi, Greece, Jan. 1997, F. Afrati and P. Kolaitis (ed.), Springer-Verlag. Disponível em <http://citeseer.nj.nec.com/mannila97method.html>



- MECHELN, Pedro José Von. **SAP1-GI: Sistema de Apoio ao Planejamento no Processo de Tomada de Decisão do Jogo de Empresas GI-EPS**. Florianópolis, 1997. Dissertação (Mestrado em Engenharia de Produção) – Curso de Pós Graduação em Engenharia de Produção. Universidade Federal de Santa Catarina – UFSC.
- MENCONI, Darlene. A Mineração de Informações: Vender Guarda-chuva onde não Chove? O Data Mining Ajuda a Conhecer os Consumidores e Acertar no Alvo. **Info Exame**. Ano 13, n. 144, p. 92-93, Mar. 1998.
- MOREIRA, Maria Isabel. A fase Adulta do SQL: A Versão 2000 Coloca o Produto da Microsoft ao Lado dos Grandes Gerenciadores de Bancos de Dados Oracle e DB2. **Info Exame**. Ano 15, n. 175, p. 166-167, Out. 2000.
- NIMER, Fernando; SPANDRI, Luiz Carlos. Obtendo Vantagem Competitiva com Data Mining. **Developers Magazine**. Ano 2, n. 18, p. 30 e 31, Fev. 1998.
- O ABC da Mineração de Dados: Como o Data Mining pode Ajudar os Executivos a Tomar Decisões bem Fundamentadas. **Info Exame**. Ano 13, n. 154, p. 22–23, Jan. 1999.
- O DATA MINING chega à Web. Site da IT Web Brasil. **Computer Reseller News Brasil**. Publicado em 16 de Agosto 2000. <http://www.crn.com.br/noticias/artigo.asp?id=5892>
- PASSOS, Emmanuel Lopes. **Inteligência Artificial e Sistemas Especialistas ao Alcance de Todos**. Rio de Janeiro: LCT - Livros Técnicos Científicos Ltda, 1989.
- PEREIRA, Max Roberto. Data Warehouse: Otimizando seu Desempenho. **Developers Magazine**. Ano 3, n. 32, p. 22–26, Abr. 1999.
- PIATETSKY-SHAPIRO, Gregory. Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop. **AI Magazine**, v. 11, n. 5, p. 68–70, Jan. 1991. Disponível em <http://www.kdnuggets.com/meetings/kdd89/kdd-89-report-aimag.html>
- QUINLAN, J. Ross. **Induction of Decision Tree**. Machine Learning, 1:81-106, 1986.

- RABUSKE, Renato Antônio. **Inteligência Artificial**. Florianópolis: UFSC, 1995.
- REGGIANI, Lucia. Pesquisa aos Montes: Os Sites que Coletam as Opiniões dos Internautas Proliferam mais que Cogumelos. **Info Exame**. Ano 16, n. 179, p. 22-23, Fev. 2001.
- RICH, Elaine; KNIGHT, Kevin. **Inteligência Artificial**. 2. ed. São Paulo: Makron Books, 1993.
- RÖDDER, Wilhelm; KOPITTKE, Bruno Hartmut.; KULMANN, Friedhelm. **Sistemas Especialistas Probabilísticos**. Texto para o Ensino à Distância – Feito em Cooperação com a FernUniversität Hagen. Florianópolis, 1997. Departamento de Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina – UFSC.
- RÖDDER, Wilhelm; MEYER, Carl-Heinz. **Probabilistic Reasoning and Inductive Bayes Learning at Minimal Relative Entropy**. Department of Mathematical Economics. FernUniversität Hagen, Germany, 1995.
- SOUZA, Otávio Roberto Martins de. **Mineração de Dados de um Plano de Saúde para Obter Regras de Associação**. Florianópolis, 2000. Dissertação (Mestrado em Engenharia de Produção) – Curso de Pós Graduação em Engenharia de Produção. Universidade Federal de Santa Catarina – UFSC.
- SPIRIT 1. EXE. Versão 0.90 – 16 bit. **Shell para Criação de Sistemas Especialistas Probabilísticos**. Responsável: Prof. Dr. W. Rödder. FernUniversität Gesamthochschule in Hagen – Alemanha. <http://lje.eps.ufsc.br/spirit/>
- SPRAGUE, Ralph H.; WATSON, Hugh J. **Sistema de Apoio à Decisão: Colocando a Teoria em Prática**. 2. ed. Rio de Janeiro: Campus, 1991.
- SRIKANT, Ramakrishnan; AGRAWAL, Rakesh. **Mining Quantitative Association Rules in Large Relational Tables**. Proceedings of the ACM SIGMOD Conference on Management of Data, p. 1–12, Montreal, Canada, Jun. 1996. Disponível em <http://citeseer.nj.nec.com/srikant96mining.html>

- THOMPSON, Beverly; THOMPSON, William. Finding Rules in Data. **Byte Magazine**, p. 149–158. Nov. 1986.
- TORRES, Norberto A. **Competitividade Empresarial com a Tecnologia de Informação**. São Paulo: Makron Books, 1995.
- VASCONCELLOS, João Marcos. Implementando um Data Warehouse Incremental. **Developers Magazine**. Ano 3, n. 32, p. 18–20, Abr. 1999.
- WILHELM, Pedro Paulo Hugo. **Uma Nova Perspectiva de Aproveitamento e Uso dos Jogos de Empresas**. Florianópolis, 1997. Tese (Doutorado em Engenharia de Produção) – Curso de Pós Graduação em Engenharia de Produção. Universidade Federal de Santa Catarina – UFSC.
- WÜTHRICH, Beat. **Knowledge Discovery in Databases**. Kowloon, Hong Kong: The Hong Kong University of Science and Technology, 1996. Disponível em <http://citeseer.nj.nec.com/89234.html>