

**ALEXANDRE BIAZIN**

**UM MODELO DE INTEGRAÇÃO DE LÓGICA  
FUZZY A BANCO DE DADOS CONVENCIONAIS**

**Florianópolis - SC  
2002**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**

**ALEXANDRE BIAZIN**

**UM MODELO DE INTEGRAÇÃO DE LÓGICA FUZZY A  
BANCO DE DADOS CONVENCIONAIS**

Dissertação submetida à Universidade Federal de Santa Catarina  
como parte dos requisitos para obtenção do grau de Mestre em  
Ciência da Computação.

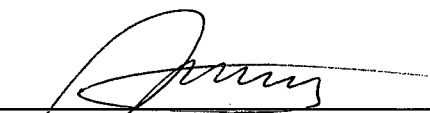
**Orientador: Prof. Dr. João Bosco da Mota Alves**

**Florianópolis, Janeiro de 2002**

# UM MODELO DE INTEGRAÇÃO DE LÓGICA FUZZY A BANCO DE DADOS CONVENCIONAIS

ALEXANDRE BIAZIN


Esta dissertação foi julgada adequada para a obtenção do título de **Mestre em Ciência da Computação** Área de Concentração **Sistemas de Computação**, e aprovada em sua forma final pelo **Programa de Pós-Graduação em Ciência da Computação**.



---


Prof. Dr. Fernando O. Gauthier  
Coordenador do Curso

BANCA EXAMINADORA



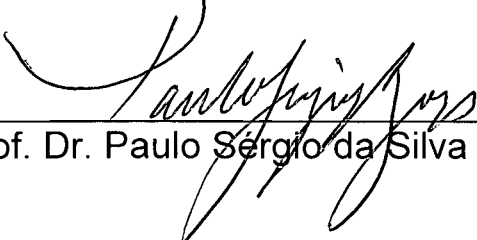
---

Prof. Dr. João Bosco da Mota Alves – **Orientador**



---

Prof. Dr. Luiz F. Jacintho Maia



---

Prof. Dr. Paulo Sérgio da Silva Borges

## AGRADECIMENTOS

Ao Professor Orientador João Bosco da Mota Alves pela sua valiosa contribuição para o desenvolvimento dessa pesquisa.

A minha esposa Clair T. Gomes de Góis Biazin pelo incentivo e colaboração para que este trabalho fosse realizado.

Aos meus pais pelo eterno apoio e incentivo em todos os momentos.

A todos os meus amigos e colegas pela paciência e alegria proporcionada nos momentos de intenso trabalho e estudo.

Aos professores Dr. Luiz F. Jacintho Maia e Dr. Paulo Sérgio da Silva Borges pelo apoio e conhecimento transmitidos, indispensáveis à realização deste trabalho.

A todos contribuíram para o desenvolvimento dessa pesquisa.

## RESUMO

Este trabalho apresenta as várias pesquisas realizadas em Lógica Fuzzy integrada ao SGBD (Sistemas Gerenciadores de Banco de Dados), expondo os métodos e práticas utilizadas para integração das duas tecnologias. Através da inclusão da uma máquina de inferência fuzzy interna ao SGBD, permite consultar dados em forma nebulosa. Este procedimento utiliza entradas e saídas convencionais, e regras definidas no modelo, possibilitando que uma consulta seja feita sem alterar os padrões estabelecidos na linguagem SQL.

## **ABSTRACT**

This paper presents the several research works conducted in Fuzzy Logic integrated DBMS (Database Management System), exposing the methods and practices used for integration of the two technologies. Through the inclusion of a fuzzy inference machine inside the DBMS. It allows to consult data in fuzzy form. This procedure uses conventional inputs and outputs, and defined rules in the model, facilitating for a search to be made without altering the patterns established in the language SQL.

## SUMÁRIO

<b>AGRADECIMENTOS</b> .....	<b>III</b>
<b>RESUMO</b> .....	<b>IV</b>
<b>ABSTRACT</b> .....	<b>V</b>
<b>SUMÁRIO</b> .....	<b>VI</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>VII</b>
<b>ÍNDICE DE TABELAS</b> .....	<b>VIII</b>
<b>1 INTRODUÇÃO</b> .....	<b>1</b>
1.1 Motivação .....	3
1.2 Objetivos.....	3
1.3 Organização do Trabalho.....	4
<b>2 LÓGICA FUZZY</b> .....	<b>6</b>
2.1 História da Lógica fuzzy .....	7
2.3 Conjuntos Fuzzy .....	8
2.4 Variáveis Lingüísticas .....	10
2.5 Operações com Conjuntos Fuzzy .....	13
2.5.1 União .....	13
2.5.2 Interseção .....	14
2.5.3 Complemento .....	14
<b>3 SISTEMAS FUZZY</b> .....	<b>16</b>
3.1 Funções de Pertinência .....	17
3.1.1 Função Triangular.....	17
3.1.2 Função Trapezoidal .....	18
3.1.3 Função LR (Left-Right).....	19
3.2 Construção de Funções de Pertinência .....	20
3.2.1 Intuição .....	20
3.2.2 Inferência .....	21
3.3 Processamento Fuzzy .....	22
3.3.1 Fuzificação.....	22
3.3.2 Avaliação de Regras .....	23
3.3.3 Defuzzificação .....	25

3.4 Sistema Fuzzy Baseados em Regras.....	29
3.4.1 Vantagens de Controladores Fuzzy Baseados em Regras .....	32
<b>4 TECNOLOGIAS DE BANCOS DE DADOS.....</b>	<b>33</b>
4.1 Banco de Dados Distribuídos .....	34
4.2 Banco de Dados Relacional.....	35
4.3 Banco de Dados Orientado a Objetos.....	36
4.4 Linguagem SQL .....	38
4.4.1 A Linguagem SQL 3.....	38
<b>5 BANCO DE DADOS FUZZY .....</b>	<b>41</b>
5.1 Uma Proposta para Bancos de Dados Fuzzy .....	42
5.2 Processamento de Consultas Fuzzy .....	48
5.3 Linguagens de Consulta Fuzzy.....	51
5.4 Álgebra Relacional Nebulosa .....	53
5.5 Cálculo Relacional.....	55
5.6 Modelo Físico do Banco de Dados Fuzzy .....	58
5.6.1 Performance em um Banco de Dados Fuzzy .....	59
5.6.2 Estrutura de Grid File .....	61
5.6.3 Indexação Fuzzy .....	62
5.6.4 Multi Level Grid File.....	65
<b>6 FUNÇÕES FUZZY INTEGRADAS AO SISTEMA GERENCIADOR DE BANCO DE DADOS .....</b>	<b>68</b>
6.1 Máquina de Inferência Fuzzy .....	69
6.2 O Modelo de Consulta .....	69
6.3 Problema Proposto.....	70
6.3.1 Especificação da Linguagem .....	71
6.3.2 SQL e Funções Fuzzy .....	71
6.4 Construção DA máquina de inferência fuzzy.....	72
6.4.1 Fuzzificação.....	72
6.4.2 Definição das Regras .....	73
6.4.3 Defuzzificação .....	74
6.5 Integração da Máquina de inferência Fuzzy ao SGBD .....	74
6.5.1 Referenciando a Biblioteca no SGBD .....	76
6.5.2 Triggers .....	78
<b>7 CONCLUSÃO.....</b>	<b>79</b>
<b>8 REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>81</b>
<b>ANEXO I – REGRAS DE INFERÊNCIA FUZZY .....</b>	<b>86</b>



## ÍNDICE DE FIGURAS

FIGURA 2.1 - CONJUNTOS CRISP.....	10
FIGURA 2.2 - CONJUNTO FUZZY .....	10
FIGURA 2.3 - FUNÇÃO DE PERTINÊNCIA.....	11
FIGURA 2.4 - UNIÃO DE CONJUNTOS FUZZY .....	13
FIGURA 2.5 - INTERSEÇÃO DE CONJUNTOS FUZZY .....	14
FIGURA 2.6 - COMPLEMENTO DE CONJUNTOS FUZZY .....	15
FIGURA 3.1 - FUNÇÃO TRIANGULAR.....	18
FIGURA 3.2 - FUNÇÃO TRAPEZOIDAL.....	19
FIGURA 3.3 - FUNÇÃO LR.....	20
FIGURA 3.4 - FUNÇÃO DE PERTINÊNCIA TEMPERATURA .....	21
FIGURA 3.5 - FUZIFICAÇÃO DO VALOR +40 .....	23
FIGURA 3.6 - GRÁFICO PARA A VARIÁVEL TEMPERATURA .....	24
FIGURA 3.7 - GRÁFICO PARA A VARIÁVEL WATT .....	24
FIGURA 3.8 - GRÁFICO DE DEFUZZIFICAÇÃO.....	25
FIGURA 3.9 - ERROS QUE PODEM OCORRER DEVIDO A DEFUZZIFICAÇÃO PELO CENTRO-DA-ÁREA.....	26
FIGURA 3.10 - GRÁFICO DE DEFUZZIFICAÇÃO C-O-M .....	27
FIGURA 4.1 - EVOLUÇÃO DO BANCO DE DADOS (RAO, 1994).....	37
FIGURA 5.1 – NÍVEIS DO BANCO DE DADOS FUZZY .....	50
FIGURA 5.2 - ABORDAGENS DE ARQUITETURA ENVOLVENDO INCERTEZAS ..	51
FIGURA 5.3 – EXEMPLO DA ESTRUTURA DE GRID FILE.....	62
FIGURA 5.4 – ESQUEMA CONCEITUAL DO MÉTODO DE INDEXAÇÃO PROPOSTO POR BOSCH .....	63
FIGURA 5.5 – ESTRUTURA DO DIRETÓRIO EM MLGF .....	66
FIGURA 5.5 – O ESPAÇO DE PARTIÇÃO CORRESPONDENTE A ESTRUTURA DO MLGF .....	67

<b>FIGURA 6.1 – MÁQUINA DE INFERÊNCIA FUZZY E O SGBD.....</b>	<b>69</b>
<b>FIGURA 6.2 – ESQUEMA DA CONSULTA .....</b>	<b>70</b>
<b>FIGURA 6.3 – FUNÇÃO DE PERTINÊNCIA PARA VARIÁVEL VALOR EMPRÉSTIMO.....</b>	<b>72</b>
<b>FIGURA 6.4 – FUNÇÃO DE PERTINÊNCIA PARA VARIÁVEL SALDO MÉDIO .....</b>	<b>73</b>
<b>FIGURA 6.5 – FUNÇÃO DE PERTINÊNCIA PARA VARIÁVEL VALOR PATRIMÔNIO .....</b>	<b>73</b>
<b>FIGURA 6.6 – FUNÇÃO DE PERTINÊNCIA PARA VARIÁVEL MONTANTE DIVIDAS .....</b>	<b>73</b>
<b>FIGURA 6.7 INTERVALO DE VALORES PARA DEFUZZIFICAÇÃO.....</b>	<b>74</b>
<b>FIGURA 6.8 – PROCESSO DE INTEGRAÇÃO DA MÁQUINA DE INFERÊNCIA FUZZY AO SGBD.....</b>	<b>75</b>

## ÍNDICE DE TABELAS

TABELA 5.1 - RELAÇÃO MERCADOS .....	43
TABELA 5.2 - RELAÇÃO DE COMPATIBILIDADE .....	45
TABELA 5.3 - RELAÇÃO ESPECIALISTA, E RELAÇÃO AVALIAÇÃO .....	46
TABELA 5.4 - RELAÇÃO TEMPORÁRIA R1 .....	46
TABELA 5.5 - RELAÇÃO TEMPORÁRIA R2 .....	47
TABELA 5.6 - RELAÇÃO TEMPORÁRIA R3 .....	47
TABELA 5.7 - RESULTADO FINAL .....	48
TABELA 5.8 – PROJEÇÃO DE UMA RELAÇÃO NEBULOSA.....	55
TABELA 5.9 – TUPLAS SELECIONADAS UTILIZANDO CÁLCULO DE PREDICADOS.....	57
TABELA 5.10 – TUPLAS SELECIONADAS UTILIZANDO TODOS OS OPERANDOS DO PREDICADO DA CONSULTA .....	58
TABELA 6.1 – RELAÇÃO CLIENTE .....	71

## 1 INTRODUÇÃO

Uma das grandes mudanças ocorridas na ciência e matemática foi, à introdução dos conceitos de incerteza. Esta mudança se manifestou por uma transição gradual da visão tradicional, na qual insiste que essa incerteza é indesejável na ciência e deveria ser evitada por todos os meios possíveis. Porém existe uma visão alternativa, na qual é tolerante da incerteza e insiste que a ciência não pode evitá-la.

Nas teorias clássicas, um conceito pode ser verdadeiro ou falso, como por exemplo, um elemento pode pertencer ou não a um conjunto, uma proposição pode ser verdadeira ou falsa, etc. Na teoria nebulosa, um conceito pode assumir valores intermediários entre o verdadeiro e o falso. Os “graus de verdade” podem ser vistos, por exemplo, como *possibilidade* de um evento ocorrer, a *pertinência* de um elemento a um conjunto ou a *relevância* de uma parte para o todo. Esta se divide em duas áreas principais de estudo: a teoria dos conjuntos nebulosos, como extensão da teoria clássica dos conjuntos, e a lógica nebulosa, também como extensão da lógica clássica.

A teoria dos conjuntos nebulosos baseia-se na possibilidade de um elemento poder possuir graus de pertinência a um determinado conjunto, resultando, em consequência, em uma extensão das operações clássicas entre conjuntos.

Na lógica nebulosa uma proposição do tipo “José é alto” possui graus de verdade ou relevância, dessa forma as operações e regras de inferência devem ser igualmente estendidas de forma a produzir deduções lógicas também com níveis de relevância.

Muitas áreas de aplicação têm sido alvo da teoria nebulosa, entre elas os sistemas de bancos de dados. Existem vários trabalhos teóricos a respeito, apesar de não haver um grande número de implementações. No modelo de dados nebuloso, os atributos das entidades de dados possuem valores conceituais associados a graus de relevância, como por exemplo, o atributo ALTURA poderia assumir o valor “Baixo” com 40% de relevância, ou “Alto” com 70% de relevância.

Segundo (BRAGA, 1998), a maioria dos trabalhos implementados ou sugeridos baseiam-se em bancos dados que já possuem valores nebulosos, contudo a maioria dos bancos de dados são constituídos de dados convencionais ou exatos. A teoria nebulosa permite a utilização de vários recursos interessantes como o estabelecimento de regras nebulosas entre variáveis.

Os sistemas de banco de dados atuais, não contemplam a teoria dos conjuntos nebulosos, porém constantemente é necessária sua utilização. Por ser um sistema largamente utilizado em diversas áreas, essa dificuldade é maior percebida nos sistemas gerenciadores de banco de dados relacionais. Portanto nem sempre os valores a serem manipulados são valores exatos, como por exemplo:

*“Selecione todos os clientes que possuam uma credibilidade **alta**”*

A credibilidade do cliente depende de vários fatores que devem ser levados em consideração, e não somente se o cliente está com dívidas ou não.

## 1.1 MOTIVAÇÃO

Tradicionalmente, os Sistemas de Gerenciamento de Base de Dados (SGBD), baseiam-se em dados que não possuem valores nebulosos, sendo constituídos de dados convencionais ou exatos. Dados exatos nem sempre conseguem modelar o mundo real, pois valores subjetivos fazem parte da natureza humana e precisão ser considerados.

Os sistemas gerenciadores de banco de dados existentes não contemplam a utilização de dados nebulosos em suas consultas a informações. Grande parte dessa limitação deve-se a linguagem SQL, pois a mesma fundamenta-se na teoria relacional incorporando algumas extensões de orientação a objetos.

Existem poucos estudos na área da integração de lógica fuzzy e banco de dados. Estes estudos são direcionados em sua maioria para Data Mining, o qual não representa por inteiro os mecanismos com que a lógica fuzzy pode ser incorporada a um banco de dados convencional.

## 1.2 OBJETIVOS

Este trabalho pretende demonstrar formas de integração de sistemas nebulosos a sistemas gerenciadores de banco de dados convencionais. Que possibilite a utilização da linguagem SQL para consultar e fazer análises em bancos de dados utilizando variáveis nebulosas.

Desenvolver uma metodologia para utilização de valores não exatos em um banco de dados convencional, permitindo uma integração entre as variáveis nebulosas e valores exatos e aplicando os métodos dos algoritmos básicos de inferência fuzzy na resolução de valores conceituais como, por exemplo, uma variável “Velocidade”, poderia ser modelada como “Rápida”, “Moderada” e “Lenta”.

Investigar formas de integrar ao banco de dados, funções e metodologias que permitam criar uma “visão” nebulosa dos dados, desenvolvendo um modelo de integração de um SGBD e funções fuzzy.

### 1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em 8 capítulos, incluindo esta introdução.

O capítulo 2 faz uma revisão dos conceitos de lógica fuzzy e sua aplicação nas diversas áreas. Expõe a teoria dos conjuntos nebulosos, sua funcionalidade, construção e operação.

O capítulo 3 descreve o funcionamento dos sistemas fuzzy, incluindo os tipos de funções de pertinência, fuzificação, regras e defuzzificação de valores.

O capítulo 4 descreve as diversas tecnologias dos Sistemas Gerenciadores de Banco de Dados e as linguagens de consulta atualmente utilizadas.

O capítulo 5 se destina à descrição das várias propostas desenvolvidas para Banco de Dados Fuzzy, considerando sua construção e problemas de performance. Também expõem os conceitos de álgebra relacional nebulosa e as definições das linguagens de consulta fuzzy.

O capítulo 6 expõe a criação e integração de uma máquina de inferência fuzzy a um Sistema Gerenciador de Banco. E utilizando um modelo concreto procura demonstrar a funcionalidade do modelo.

O Capítulo 7 apresenta as conclusões e considerações finais do trabalho e os resultados obtidos do desenvolvimento e testes do modelo proposto.



## 2 LÓGICA FUZZY

A lógica fuzzy vem rapidamente se tornando a mais bem sucedida das tecnologias para desenvolver sistemas de controle sofisticados. A razão para isso é que a lógica fuzzy dirige-se perfeitamente a tais aplicações, e como se assemelha a decisão humana, possui uma habilidade de gerar soluções precisas de informação certa ou aproximada. Preenche uma abertura importante nos métodos de projeto da engenharia os quais não podem ser resolvidos por aproximações puramente matemáticas (como por exemplo, um projeto linear de controle), e aproximações puramente baseados em lógica (por exemplo sistemas especialistas) no projeto do sistema.

Enquanto outras equações requerem aproximações exatas para modelar o comportamento do mundo real, o projeto fuzzy pode acomodar as ambigüidades da linguagem e da lógica humanas do mundo real. Fornece um método intuitivo, descrevendo sistemas em termos humanos e automatiza a conversão daquelas especificações de sistema em modelos eficazes.

Quando se fala que o tempo está “ensolarado” não se quer dizer que não haja nuvens. Da mesma forma, em um tempo nublado pode haver sol. Geralmente aceita-se como ensolarado um céu com até 20% ou 30% de nuvens (KLIR,1995). Como se pode então definir os limites de nossas proposições? Pode-se dizer que, quando o ser humano utiliza tais termos, ele infere um grau de verdade. Dentro de uma região tropical, por exemplo, pode-se aceitar nuvens em maior quantidade, visto que uma pouca quantidade de

sol poderia ser suficiente para as atividades que dele dependem. Os valores que definem o grau de verdade de uma proposição podem não ser conscientes, mas de alguma forma definidos pelo contexto social, lingüístico, ou uma base de referência experimental. A definição e a modelagem destes termos e valores constituem a base da teoria e do estudo dos sistemas nebulosos.

## 2.1 HISTÓRIA DA LÓGICA FUZZY

Nos anos 60, um professor de engenharia elétrica e ciência da computação denominado Lotti Zadeh desenvolveu uma variação da tradicional teoria de conjunto e lógica multivalorada, essa tornava a análise e controle sistemas reais complexos mais tratáveis. Zadeh tinha observado a grande quantidade de regras que pessoas utilizavam para criação de deduções, que nem elas podiam explica-las. Por exemplo, pode-se olhar uma pessoa e dizer, "ele deve ter uns 35 anos", mas não se sabe explicar como se consegue saber isso. Zadeh percebeu que havia desenvolvido o que é agora conhecido como lógica do fuzzy. Zadeh conceitualizou muitos dos desenvolvimentos importantes no campo, embora a lógica fuzzy foi recebida primeiramente com ceticismo, desde então tem tornado-se amplamente aceita por engenheiros e cientistas da computação, e está tornando-se mais e mais comum em aplicações nos mais diversos campos.

O primeiro estágio da transição da visão tradicional, à visão moderna de incerteza, começou no século XIX, quando a física ficou interessada com os processos a nível molecular. Embora precisas, as leis da Mecânica Newtoniana são relevantes para o estudo processos, sua aplicação atual ao número de entidades envolvidas poderiam ter resultados na demanda computacional, apesar desta estar além das capacidades existentes e, como nós compreendemos agora, superamos limites computacionais fundamentais. Isto

é, leis precisas são negadas e aplicadas com domínio não unicamente em prática (baseada em tecnologia de computador existente), mas em princípio.

O estudo e a pesquisa nessa área possuem vários enfoques. Um dos mais explorados inicialmente em aplicações práticas foi o controle de processos (COX,1994). Os fundamentos teóricos se dividem em teoria dos conjuntos nebulosos e em lógica nebulosa. A teoria dos conjuntos nebulosos é uma extensão da teoria dos conjuntos, representando um enfoque mais amplo. Aqui estão incluídos os estudos de aritmética, programação, topologia, teoria de grafos e análise de dados. A lógica nebulosa tem por princípio estender a lógica clássica de forma a modelar e incluir a incerteza e a imprecisão do raciocínio humano (ZADEH,1965,1988).

### 2.3 CONJUNTOS FUZZY

Os conjuntos fuzzy possibilitam a representação de incerteza não estatística, e o *raciocínio aproximado*, que inclui as operações utilizadas para fazer inferências em lógica fuzzy. A tradicional lógica Aristotélica é bivalente em ambos fatos e operações. Assim, em lógica bivalente uma declaração é verdadeira ou falsa. Um programa tradicional de lógica é um objeto se a declaração X é verdadeira, e outra coisa se ela é falsa. Essas espécies de regras, tecnicamente chamadas regras de produção, são freqüentemente referidas para regras "se-então", porque elas são expressas no formato "se A então B."

- Se **A** e **B** então **C**
- Se **A** e não-**B** então **C**
- Se **A** ou **B** então não-**C**
- Se (((não-**A**) ou **B**) e não-**C**) ou **D** então **E**

Uma das visões do Zadeh foi que no mundo real freqüentemente encontramos graus de verdade, fenômenos que são "tipo de A", ou "geralmente

B", ou "muito C". A, B, e C não são inteiramente falso ou verdadeiro, ou membros perfeitos de um conjunto ou categoria. Conseqüentemente uma regra tal como, "Se está nublado, então choverá", simplesmente não funciona no mundo real. Às vezes, na maior parte do tempo, está parcialmente nublado, ou está bom, ou talvez está inteiramente nublado, mas as nuvens poderão não ser adequadas para produzir chuva. Coisas reais ocorrem por graus, por exemplo, escuridão pode variar de "não nublado" para "completamente nublado". Na teoria de conjuntos fuzzy, o céu é um membro do conjunto "céus nublados" por graus, a declaração "O céu é nublado" pode variar em seu grau de verdade, em uma escala de zero para um. Isto introduz o conceito de nebulosidade, que Bellman e Zadeh em 1970 definiram como um tipo de imprecisão que é associado com classes, onde não há uma transição repentina do membro para o não membro. Porém não existem somente graus na escuridão, mas a chuva também ocorre em graus, por exemplo, pode chover um pouco, ou pôde chover muito. Assim a aplicação no mundo real não poderia fazer muito sentido designar um valor de 0 ou 1 a escuridão, ou então tentar estimar o 1 ou 0, sim ou não, verdadeiro ou falso para responder se choverá. Portanto, como lógica tradicional poderia tentar responder essa pergunta.

O exemplo demonstrado na figura 2.1 exhibe-se uma linha delimitando a idade adulta da não-adulta de acordo com a lógica tradicional e nebulosa. Na lógica tradicional a idade adulta se inicia no primeiro segundo após o 18º aniversário.

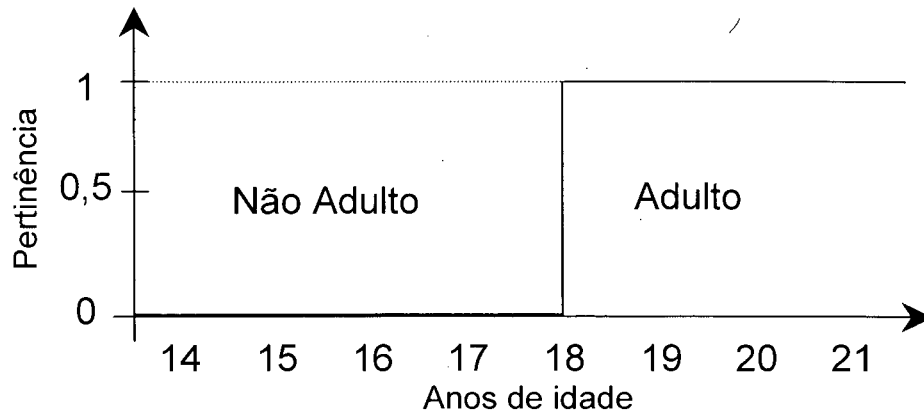


Figura 2.1 - Conjuntos Crisp

Entretanto na lógica Fuzzy (figura 2.2) essa transição é mais vaga, e se dá através de uma curva de função de pertinência (nesse caso no conjunto de pessoas adultas).

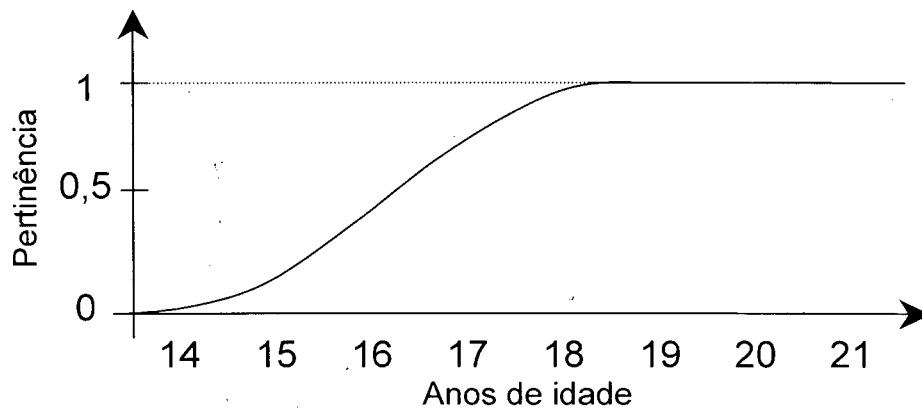


Figura 2.2 - Conjunto Fuzzy

## 2.4 VARIÁVEIS LINGÜÍSTICAS

O conceito de variável lingüística é amplamente exposto por (COX,1994, ZADEH,1988) e mesmo formalmente definido por (KLIR,1995, BELCHIOR,1997), e por quase todos os autores. A variável lingüística é, em termos mais simples, uma variável cujo domínio, ou valores, são termos da linguagem referentes a um certo contexto (ZADEH, 1988). Pode-se dizer, por

exemplo, que a variável lingüística “Idade” pode assumir os valores “Jovem”, “Adulto” ou “Velho”.

Aplicando estes conceitos à teoria nebulosa, os valores (Jovem, Adulto, Velho) seriam nomes de conjuntos nebulosos, denominados termos lingüísticos, aos quais estão associadas a um universo de discurso IDADES (Figura 2.3). A variável nítida Idade, chamada também de variável base (KLIR,1995, BELCHIOR,1997), tem seus valores classificados pelos conjuntos nebulosos. Eles permitem uma melhor adaptação dos conceitos genéricos às variáveis lingüísticas.

Este processo de classificação é similar ao trabalho com a variável através de intervalos (KLIR,1995). Contudo, a extensão fornecida pela manipulação nebulosa nos dá maior nível de informação. Uma das razões se deve à relevância em relação aos valores da variável base. A flexibilidade fornecida pela função de pertinência nos permite fazer operações com os intervalos mantendo uma maior adequação com os conceitos do mundo real.

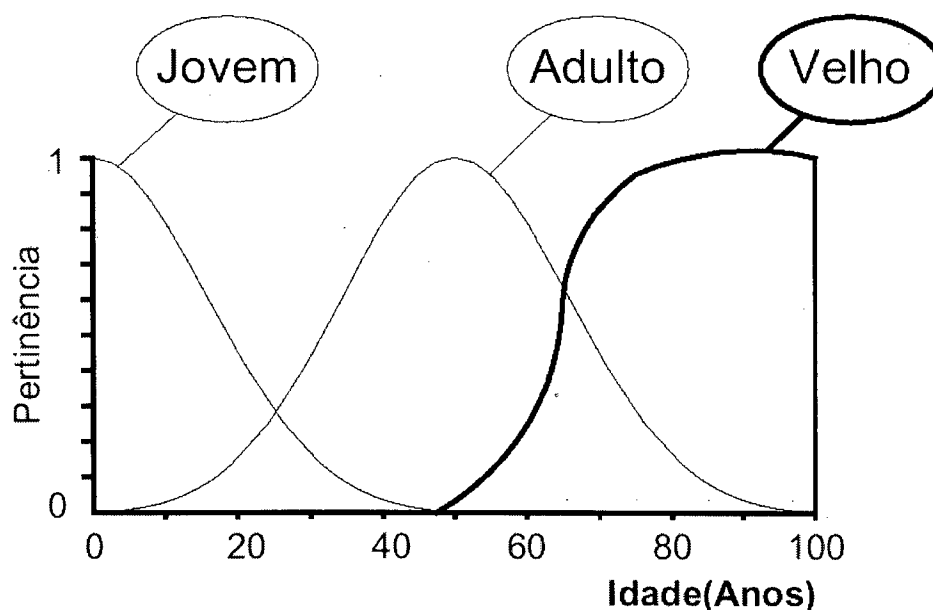


Figura 2.3 - Função de Pertinência

Para a construção de uma variável lingüística (KLIR,1995, BELCHIOR,1997) de nome  $V$ , definida sobre um universo de discurso  $X$ , devemos ter um conjunto de termos lingüísticos  $T$  com gramática de geração  $g$  e uma regra semântica  $m$ . A gramática  $g$  define a sintaxe dos termos lingüísticos. Ela pode ser constituída, simplesmente de uma lista de termos, como por exemplo:

***termo* := Jovem | Adulto | Velho**

Ou então regras de formação mais complexas que envolvam modificadores ou operadores, do tipo,

***Termo* := Modificador termo |**

**Não termo |**

**Jovem | Adulto | Velho**

***Modificador* := Muito | Pouco | Razoavelmente**

Estas regras permitem a formação sintática de termos lingüísticos mais complexos como “Muito Velho”, “Não Jovem”, etc.

A regra semântica  $m$  define como os termos lingüísticos estarão associados a seus conjuntos nebulosos ou funções de pertinência. Ela assinala o grupo de termos  $t \in T$  a um grupo de conjuntos nebulosos  $f(X)$ , definido sobre um universo de discurso  $X$ , da forma:

$$m: T \rightarrow f(X)$$

Para um termo qualquer  $t$  associado a uma variável base  $u$ , tem-se,

$$m(t) = \{u, mt(u) \mid u \in X\}$$

No exemplo da figura 2.3 tem-se a regra semântica representadas pelas funções de pertinência JOVEM, ADULTO e VELHO, as quais

caracterizam os conjuntos nebulosos  $f(X)$ . Usa-se definir o conjunto de elementos definidos acima como uma quintupla  $(V, T, X, g, m)$ , a qual caracteriza a definição formal de uma variável lingüística (KLIR,1995).

## 2.5 OPERAÇÕES COM CONJUNTOS FUZZY

A teoria dos conjuntos fuzzy contém 3 operações básicas: operações de complemento, interseção e união. Elas são baseadas no conceito e pertinência, ou não, de um elemento aos conjuntos. A extensão da teoria nebulosa destes operadores é disposta de forma a manter a sua consistência semântica (BELCHIOR,1997, LI;LIU,1990, COX,1994, KLIR,1995).

### 2.5.1 UNIÃO

A função membro da união de dois conjuntos fuzzy A e B com funções  $\mu_A$  e  $\mu_B$  membro (figura 2.4), são definidas respectivamente como o máximo das duas funções membro individuais. Isto é chamado o *critério máximo*.

$$\mu_{A \cup B} = \max(\mu_A, \mu_B)$$

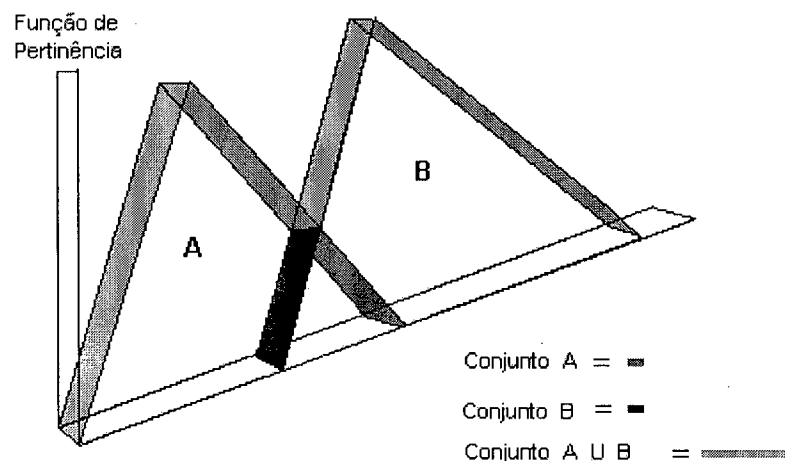


Figura 2.4 - União de Conjuntos Fuzzy



A operação da união na teoria de conjuntos fuzzy é o equivalente a operação **OU** da álgebra booleana.

### 2.5.2 INTERSEÇÃO

A função membro da interseção de dois conjuntos fuzzy A e B com funções membro  $\mu_A$  e  $\mu_B$  (figura 2.5) é definida respectivamente como o mínimo das duas funções individuais dos membros. Isto é chamado o *critério* mínimo.

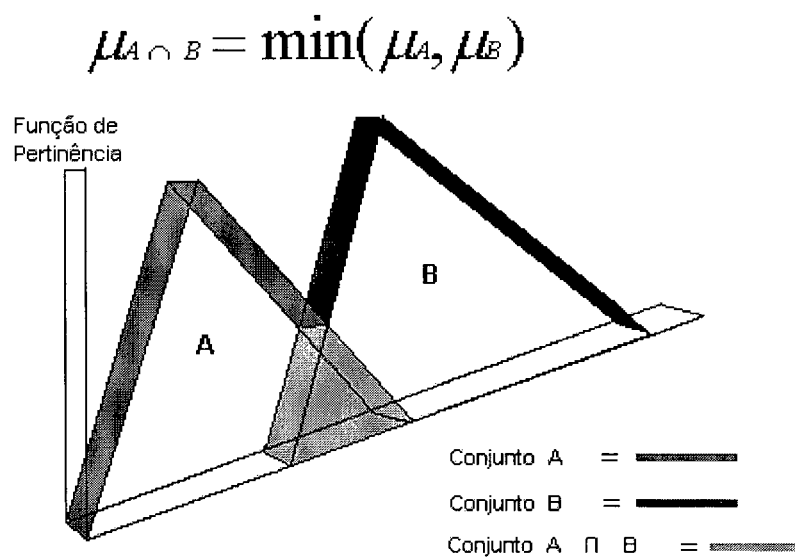


Figura 2.5 - Interseção de Conjuntos Fuzzy

A operação da interseção na teoria de conjuntos fuzzy é o equivalente a operação **E** da álgebra booleana.

### 2.5.3 COMPLEMENTO

A função membro do complemento de um conjunto fuzzy A com função membro  $\mu_A$  (figura 2.6) é definida como a negação da função membro especificada. Isto é chamado o *critério* da negação.

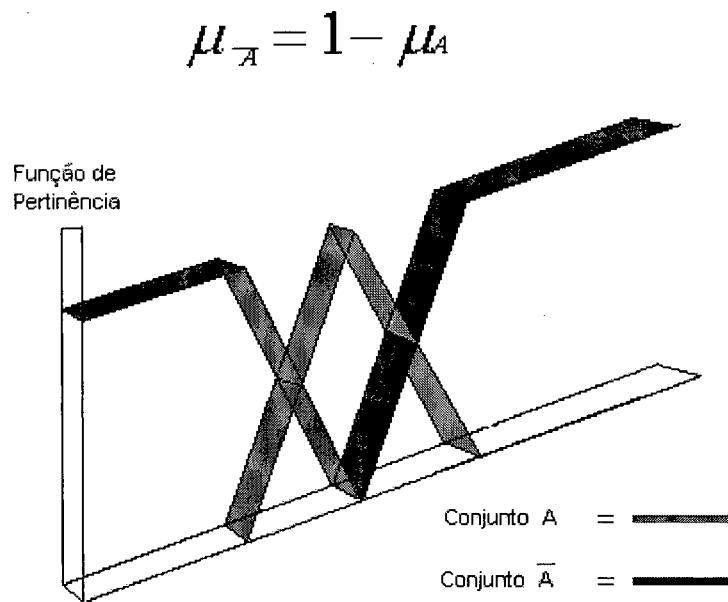


Figura 2.6 - Complemento de Conjuntos Fuzzy

A operação do complemento na teoria de conjuntos fuzzy é o equivalente a operação NÃO da álgebra booleana. As seguintes regras, as quais que são comuns na teoria de conjuntos clássicos, também aplicam-se à teoria de conjuntos fuzzy.

### 3 SISTEMAS FUZZY

Um sistema Fuzzy pode ser montado para o mesmo propósito de qualquer outro sistema de computação, como por exemplo, mapear entradas e saídas. A maior vantagem dos sistemas Fuzzy é a tolerância a erros e capacidade de dar respostas exatas diante de dados ambíguos.

Para (SHAW, 1999) ao se criar um sistema Fuzzy, deve-se avaliar e definir o problema, criando relações lógicas e de conjuntos, convertendo a informação nos chamados conjuntos nebulosos e interpretando o modelo. Para usar um modelo de lógica Fuzzy, seria preciso:

- Analisar o problema, certificando-se de que ele está totalmente compreendido;
- Definir o problema a partir da identificação das funções de pertinência (isto é, o grau de inclusão nos conjuntos pré-definidos);
- Converter a linguagem do problema em uma regra;
- Estabelecer um procedimento apropriado para “Fuzificar” e processar o problema;
- Aprender a interpretar o modelo e avaliar seus resultados;
- Ajustar os resultados se eles forem inapropriados ou inconvenientes.

Sistemas Fuzzy podem ser usados como um método simples de codificação de dados não lineares, tais como prognósticos de mercado ou estabilidade financeira de uma organização, possibilitando assim a construção

de um sistema Fuzzy usando um Sistema Especialista. Esses sistemas geralmente funcionam em velocidades altas e em computadores convencionais, produzindo resultados similares ao modo como os seres humanos lidam intuitivamente com a maioria dos problemas da vida real.

### 3.1 FUNÇÕES DE PERTINÊNCIA

Segundo (SHAW, 1999) as funções de pertinência fuzzy representam os aspectos fundamentais de todas as ações teóricas e práticas de sistemas fuzzy. Uma função de pertinência é uma função numérica gráfica ou tabulada que atribui valores de pertinência fuzzy para valores discretos de uma variável, em seu universo de discurso. É preciso lembrar que o universo de discurso de uma variável representa o intervalo numérico de todos os possíveis valores reais que uma variável específica pode assumir.

Entre os vários formatos existentes para serem usados como funções de pertinência, existem como principais:

#### 3.1.1 FUNÇÃO TRIANGULAR

Este tipo de função serve para indicar apenas um único ponto de máximo no conjunto. A definição é do tipo:

$$f(x) \begin{cases} y_{min}, & x \leq i \text{ ou } x \geq f \\ y_{min} + (y_{max} - y_{min}) x (x - i) / (m - i), & i < x \leq m \\ y_{min} + (y_{max} - y_{min}) x (f - x) / (f - m), & m < x < f \end{cases}$$

Onde  $y_{min}$  é o menor valor possível para a função,  $y_{max}$  o maior valor,  $i$  o início da ascendente da função,  $m$  o valor de máximo, e  $f$  o final da descendente, como é demonstrado na figura 3.1.

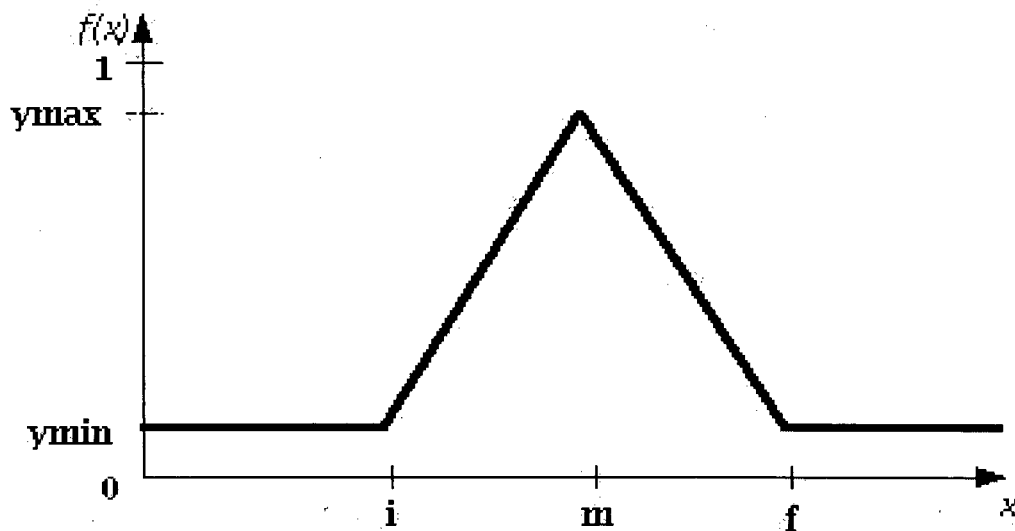


Figura 3.1 - Função Triangular

### 3.1.2 FUNÇÃO TRAPEZOIDAL

Nesta função é possível representar todo um intervalo de pontos de máximo, conforme a fórmula:

$$f(x) \begin{cases} y_{\min}, x \leq i \text{ ou } x \geq f \\ y_{\min} + (y_{\max} - y_{\min}) x (x - i) / (m1 - i), i < x \leq m1 \\ y_{\max}, m1 \leq x \leq m2 \\ y_{\min} + (y_{\max} - y_{\min}) x (f - x) / (f - m2), m2 < x < f \end{cases}$$

Onde  $y_{\min}$  é o menor valor possível para a função,  $y_{\max}$  o maior valor,  $i$  o início do trapézio,  $m1$  o início do intervalo de máximo,  $m2$  o fim do intervalo de máximo, e  $f$  o final do trapézio, como é demonstrado no gráfico abaixo (figura 3.2).

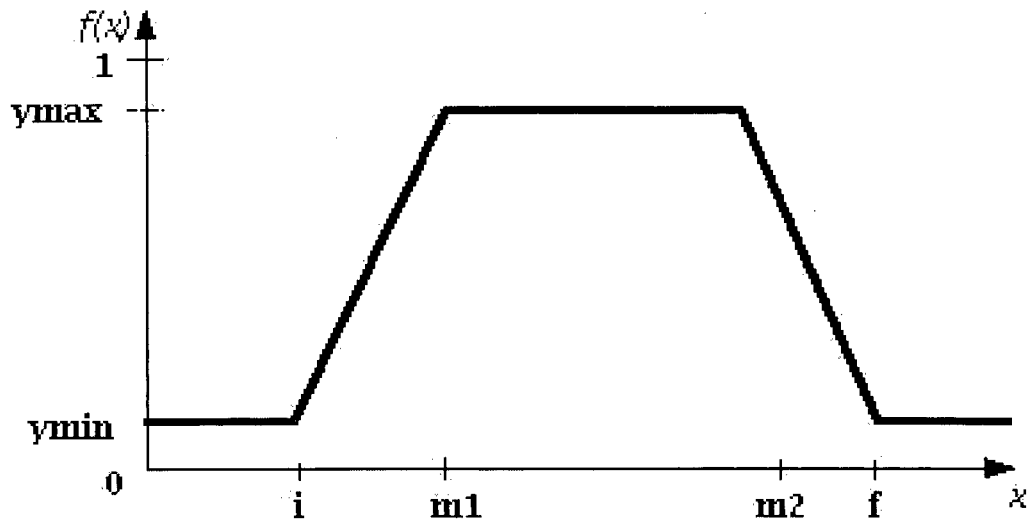


Figura 3.2 - Função Trapezoidal

### 3.1.3 FUNÇÃO LR (LEFT-RIGHT)

Este tipo de conjunto é mais genérico, pois permite a utilização de vários tipos de função para definir o intervalo de representação. As funções são chamadas de *L* (left, pois define a função à esquerda) e de *R* (right, define a da direita). A formulação padrão é do tipo:

$$f(x) \begin{cases} L((m1 - x) / \alpha), x < m1 \\ 1, m1 \leq x \leq m2 \\ R((m2 - x) / \beta), x > m2 \end{cases}$$

Onde *L* e *R* podem ser funções do tipo  $e^{-x}$ ,  $1/(1+x^2)$ , ou outra qualquer que tenha  $f(0) = f(1) = 0$  e mantenha-se no intervalo  $[0;1]$ ;  $m1$  é onde inicia o intervalo de máximo, terminando em  $m2$ ; alfa é o espalhamento lateral da esquerda e beta o da direita.

Para exemplificar, coloca-se  $L = R = 1/(1+x^2)$ , espalhamento esquerdo 5 e direito 2, então obtém-se um gráfico semelhante ao apresentado a seguir (figura 3.3).

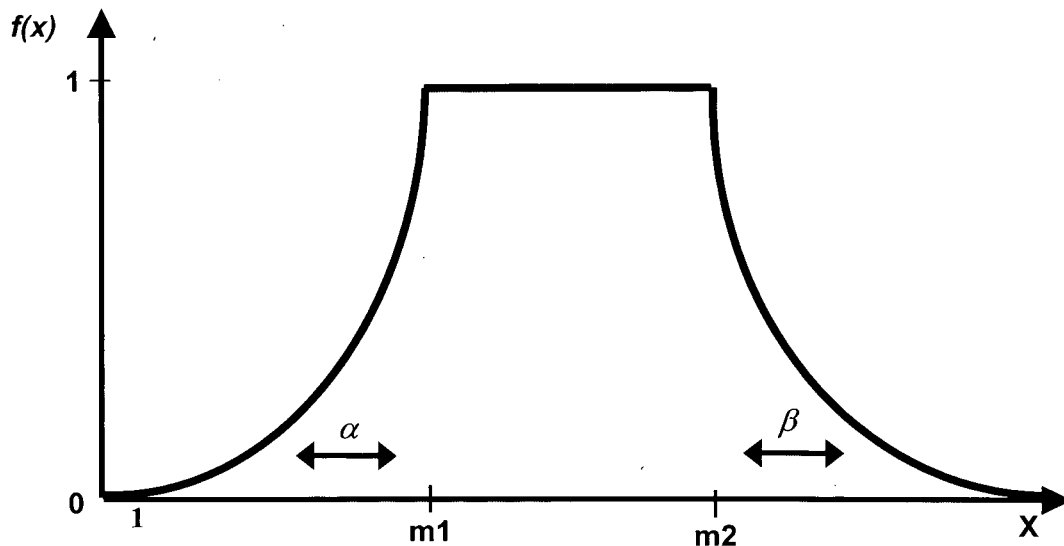


Figura 3.3 - Função LR

### 3.2 CONSTRUÇÃO DE FUNÇÕES DE PERTINÊNCIA

Embora as funções de pertinência mostradas sejam triangulares e trapezoidais, isto não é obrigatório. A quantidade de funções em um universo de discurso e seu formato é escolhida com base na experiência, na natureza do processo a ser controlado.

Um número prático de conjuntos fuzzy lingüísticos (funções de pertinência) é algo entre 2 e 7. Quanto maior o número de conjuntos, maior a precisão, mas a demanda computacional também é mais significativa. Por exemplo, experiências mostraram que uma mudança de 5 conjuntos triangulares para 7 aumenta a precisão em torno de uns 15%, a partir de valores maiores não há melhorias extremamente significativas (SHAW, 1999). Existem vários métodos para criação de uma função de pertinência.

#### 3.2.1 INTUIÇÃO

Simplesmente derivado da capacidade dos homens para desenvolver funções de pertinência através de sua própria inteligência inata e entendimento. Intuição envolve conhecimento semântico e contextual sobre um

problema e pode também envolver valores verdades lingüísticos sobre este conhecimento, como por exemplo, funções de pertinência para a variável fuzzy temperatura figura 3.4.

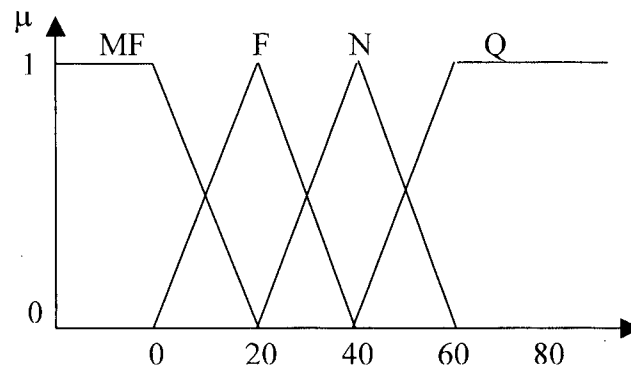


Figura 3.4 - Função de Pertinência Temperatura

A característica importante destas curvas para propor o uso em operações fuzzy é sua sobreposição.

### 3.2.2 INFERÊNCIA

Usar conhecimento para a performance do raciocínio dedutivo. Isto é, deseja-se deduzir ou inferir uma conclusão, dado um corpo de fatos e conhecimentos, como por exemplo, na identificação de um triângulo, sejam A, B e C os ângulos internos de um triângulo, na ordem ABC0, e seja U o universo dos triângulos; isto é:

- $U = \{(A, B, C) / ABC0 ; A + B + C = 180\}$
- Define-se um número de formas, por exemplo:
- I - Triângulo aproximadamente isósceles:  $I(A,B,C) = 1 - (\min(A-B, B-C))/60^\circ$
- R - Triângulo aproximadamente reto:  $R(A,B,C) = 1 - |A-90^\circ|/90^\circ \dots$



### 3.3 PROCESSAMENTO FUZZY

Deseja-se um sistema fuzzy para controle de temperatura através de condicionador de ar, ou seja, se ela está alta, então o condicionador de ar terá que baixar a potência para esfriar o ambiente, se a temperatura está boa deve assim permanecer, e se está baixa deve subir.

#### 3.3.1 FUZIFICAÇÃO

A fuzificação é processo de converter conjuntos de dados reais em graus de pertinência de funções e relacioná-los entre si, criando valores Fuzzy. O tipo de função de pertinência e seu grau são determinados através da experiência ou intuição.

Para (SHAW, 1999), a fuzificação é uma espécie de *pré-processamento* de categorias ou classes dos sinais de entrada, reduzindo grandemente o número de valores a serem processados. Uma menor quantidade de valores processados significa que há uma computação mais veloz. As funções de pertinência também podem ser um conjunto tabulado de valores numéricos, e um procedimento de consulta a tabelas.

A Figura 3.5 demonstra um conjunto de sete funções de pertinência trapezoidal e eqüidistante, com rótulos lingüísticos, cujo universo de discurso foi normalizado para  $-100 \dots +100$ . Para fuzificar o valor discreto  $+40$ , deve-se observar que uma linha vertical no ponto  $+40$  corta as funções de pertinência PM e PS, respectivamente, sem cruzar nenhuma outra função de pertinência. Isso deve ser interpretado como o equivalente fuzificado do valor discreto  $+40$ , neste caso, pertence:

- à função de pertinência fuzzy PM com grau 0,4;
- à função de pertinência fuzzy PS com grau 0,6;

- à todas as outras funções de pertinência fuzzy com grau 0,0;

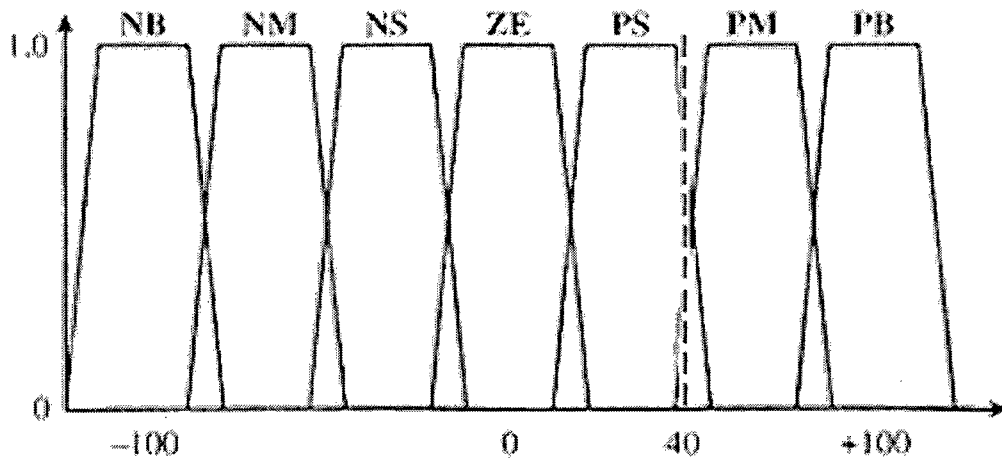


Figura 3.5 - Fuzificação do valor +40

Assim, o vetor fuzzy equivalente ao número discreto +40 é  $\{0;0;0;0;0,6;0,4;0\}$ . A soma dos componentes não-nulos é 1 devido à superposição de 50%.

### 3.3.2 AVALIAÇÃO DE REGRAS

Avaliação de regras é o processo de inferência, onde são obtidas respostas numéricas a partir de um conjunto de regras baseadas nos valores de entrada. Por exemplo, estabelecem-se as variáveis T para temperatura e P para potência do motor do condicionador de ar, de acordo com as regras a seguir:

- se T = quente então P = esfriar
- se T = normal então P = zero
- se T = frio então P = esquentar

Para que se atenda adequadamente as regras desejadas, são definidos os conjuntos para cada variável. Cada conjunto possível para cada variável será definido como uma função trapezoidal, conforme apresentado a

seguir. O gráfico (Figura 3.6) para a variável de temperatura (medida em graus Celsius) é descrito conforme as possibilidades definidas nas regras.

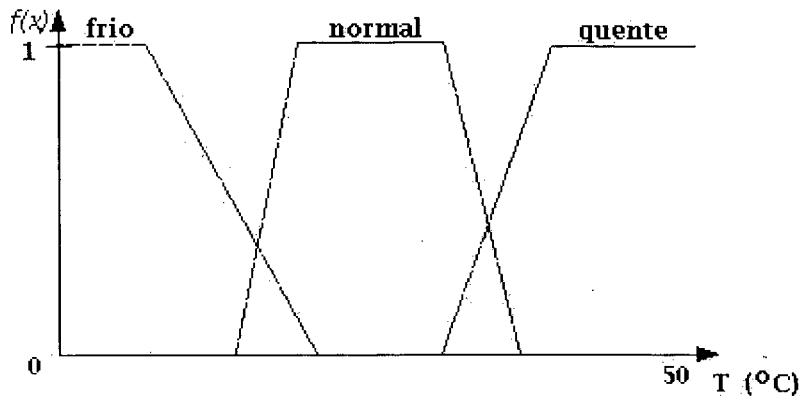


Figura 3.6 - Gráfico para a Variável Temperatura

Para a potência (em Watt), o gráfico é semelhante (figura 3.7):

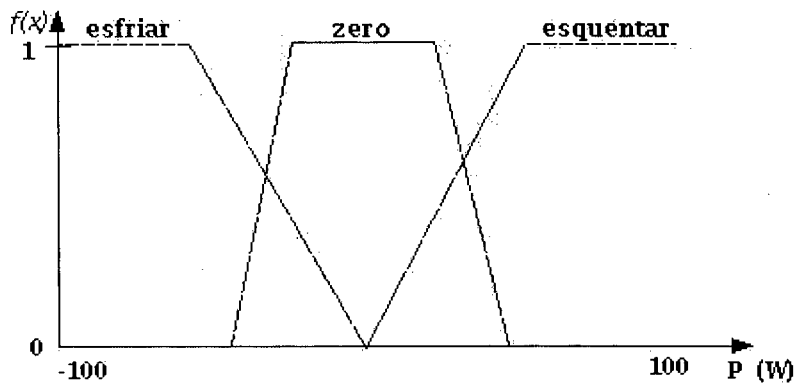


Figura 3.7 - Gráfico para a Variável Watt

Uma vez definidas as variáveis e seus conjuntos, para dada temperatura X verifica-se o ponto máximo e mínimo de intersecção. Ponto máximo é o maior valor obtido com resultado de uma das funções, e o mínimo é o menor resultado. Por exemplo, para uma temperatura de 20 graus, o ponto máximo seria dado pela função trapezoidal do conjunto normal (um valor, digamos, de 0,6). Já o ponto mínimo seria o resultado dado pela função do conjunto frio (suponhamos um valor de 0,25). Então, para uma temperatura de 20, teremos 0,6 como máximo e 0,25 como mínimo, que serão utilizados na próxima etapa.

### 3.3.3 DEFUZZIFICAÇÃO

Na defuzzificação, o valor da variável lingüística de saída inferida pelas regras fuzzy será traduzido num valor discreto. O objetivo é obter-se um único valor numérico discreto que melhor represente os valores fuzzy inferidos da variável lingüística de saída, ou seja, a distribuição de possibilidades. Assim, a defuzzificação é uma transformação inversa que traduz a saída do domínio fuzzy para o domínio discreto. Para selecionar o método apropriado de defuzzificação, pode-se utilizar um enfoque baseado no centróide ou nos valores máximos que ocorrem da função de pertinência resultante. Os seguintes métodos são muito utilizados: (1) Centro-da-Área (C-o-A), (2) Centro-do-Máximo (C-o-M), e (3) Média-do-Máximo (M-o-M) (SHAW 1999).

Como o ponto máximo foi obtido no conjunto normal da temperatura, então será definida uma área abaixo de 0,6 no conjunto zero da potência; como o ponto mínimo veio do conjunto frio, então a área abaixo de 0,25 no conjunto esquentar da potência. O somatório das áreas dará a quantidade de potência a ser somada ao motor para tender a potência a zero e, portanto, colocar a temperatura dentro do normal. A figura 3.8 ilustra a área a ser obtida.

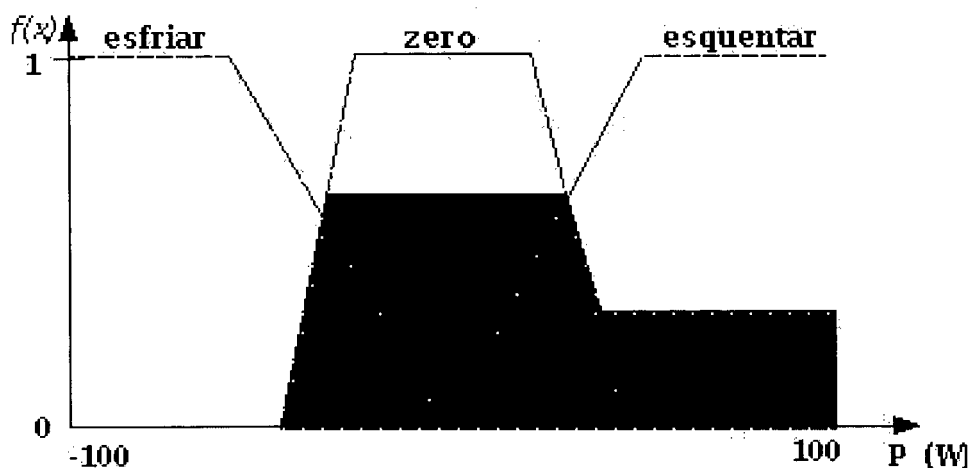


Figura 3.8 - Gráfico de Defuzzificação

- Defuzzificação Centro-da-Área (C-o-A)

O método Centro-da-Área é freqüentemente chamado de método do Centro-de-Gravidade, pois ele calcula o centróide da área composta que representa o termo de saída fuzzy ( $\mu_{OUT}$ ), esse termo de saída fuzzy é composto pela união de todas as contribuições de regras. O centróide é um ponto que divide a área de  $\mu_{OUT}$  em duas partes iguais.

O método de defuzzificação C-o-A calcula o centróide desta área. O cálculo do centróide da área se dá da seguinte forma:

$$u^* = \frac{\sum_{i=1}^N u_i \mu_{OUT}(u_i)}{\sum_{i=1}^N \mu_{OUT}(u_i)}$$

O método de defuzzificação C-o-A apresenta pequenos problemas, um deles ocorre quando as funções de pertinência não possuem sobreposição, onde o centro geométrico da figura na realidade não deveria ter significado físico, outro fator é que se mais de uma regra tiver a mesma saída fuzzy há uma sobreposição de áreas que não é devidamente contabilizada. Conforme demonstrado na figura 3.9.

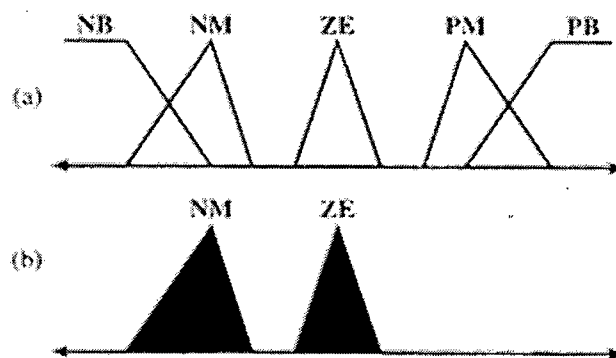


Figura 3.9 - Erros que podem ocorrer devido a defuzzificação pelo Centro-da-Área

### ■ Defuzzificação Centro-do-Máximo (C-o-M)

Neste método os picos das funções de pertinência representados no universo de discurso da variável de saída são usados, enquanto ignora-se as áreas das funções de pertinência. As contribuições múltiplas de regras são consideradas por esse método. Os valores não-nulos do vetor de possibilidades de saída são posicionados nos picos correspondentes (figura 3.10). Assumindo que representam pesos, o valor de saída defuzzificado, discreto, é determinado achando-se o ponto de

apoio onde os pesos ficam equilibrados. Assim, as áreas das funções de pertinência não desempenham nenhum papel e apenas os máximos (pertinências singleton) são usados.

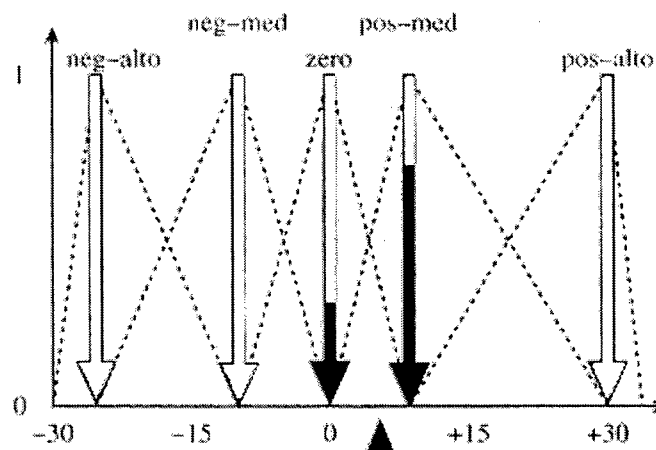


Figura 3.10 - Gráfico de Defuzzificação C-o-M

O cálculo do valor defuzzificado é realizado através da seguinte equação:

$$u^* = \frac{\sum_{i=1}^N u_i \cdot \sum_{k=1}^n \mu_{O,k}(u_i)}{\sum_{i=1}^N \sum_{k=1}^n \mu_{O,k}(u_i)}$$

Onde  $\mu_{O,k}(u_i)$  indicam os pontos em que ocorrem os máximos (alturas) das funções de pertinência de saída.

#### ■ Defuzzificação Média-do-Máximo (M-o-M)

Uma abordagem para defuzzificação poderia ser a de se utilizar a saída cujo valor tenha o maior valor de pertinência  $\mu_{OUT}(u_i)$ . Em casos onde a função de pertinência tenha mais de um máximo essa idéia não poderia ser utilizada. O a formula para cálculo desta defuzzificação é a seguinte:

$$u^* = \sum_{m=1}^M \frac{u_m}{M}$$

Onde  $u_m$  é o m-ésimo elemento no universo de discurso, onde a função PERTINÊNCIA tenha um máximo e M é o número total desses elementos (figura 3.11). A abordagem M-o-M é também chamada de solução mais plausível, por desconsiderar o formato das funções de pertinência de saída.

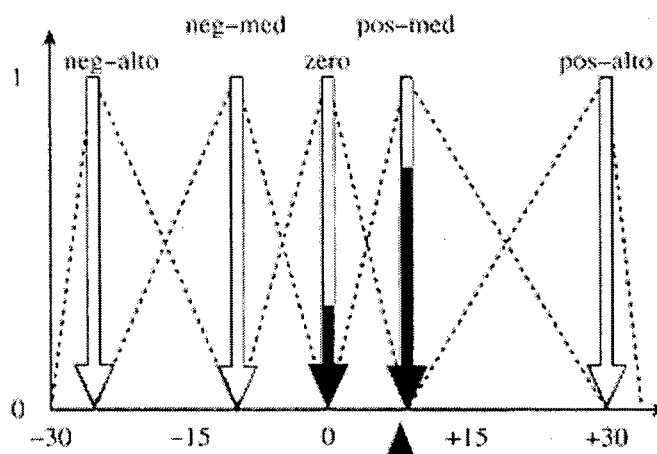


Figura 3.11 - Gráfico de Defuzzificação M-o-M

### ■ Qual Método de Defuzzificação deve-se Usar?

**Aplicações em malha fechada:** a propriedade de continuidade, é importante, pois se a saída de um controlador fuzzy controla uma variável do processo, saltos na saída do controlador podem causar instabilidade e oscilações, logo é prudente optar pela defuzzificação *C-o-M*.

**Reconhecimento de padrões:** pode-se usar o método *M-o-M*, porque se deseja identificar objetos pela classificação do sinal de um sensor, o resultado mais plausível é interessante. O vetor de possibilidades de saída é o resultado da classificação, pois ele contém informações sobre a similaridade entre o sinal e os objetos-padrão.

**Suporte à decisão:** a escolha do método de defuzzificação depende do contexto da decisão. Decisões quantitativas, como alocação de recursos, ou priorização de projetos, pode-se usar o *C-o-M*, enquanto o *M-o-M* é recomendado para decisões qualitativas, como detecção de fraude em cartões de crédito e avaliação de crédito.

## 3.4 SISTEMA FUZZY BASEADOS EM REGRAS

Há dois tipos básicos de implicações fuzzy, também chamados por regras de inferência ou associações: o *modus ponens* (modo afirmativo) e o *modus tollens* (modo negativo), ambos operando de forma a se basear em premissas ou condições, as quais geram uma determinada consequência.

<i>modus ponens:</i>	Premissa 1:	$u = A$
	Premissa 2:	<i>se</i> $u = A$ <i>então</i> $y = B$
	Consequência:	$y = B$

Isso se relaciona com o mecanismo de inferência progressiva usada em controladores fuzzy e sistemas especialistas.



<i>modus tollens</i> :	Premissa 1:	$y = \text{n\~{a}o-B}$
	Premissa 2:	$\text{se } a = A \text{ ent\~{a}o } y = B$
	Conseq\u00eancia:	$a = \text{n\~{a}o-A}$

Isso se relaciona com o mecanismo de infer\u00eancia regressiva usada em sistemas especialistas, assunto de diversos textos e publica\u00e7\u00f5es relativos \u00e0 \u00e1rea de sistemas especialistas.

Sistemas especialistas que usam infer\u00eancia progressiva, geralmente empregam vari\u00e1veis discretas ou vari\u00e1veis simb\u00f3licas convertidas em n\u00fameros discretos. Como resultado, acabam lidando com um enorme n\u00famero de regras, centenas e muitas vezes milhares s\u00e3o utilizadas na base de conhecimento. Essa ordem de grandeza \u00e9 muito maior que a quantidade de regras geralmente utilizada em um sistema fuzzy (tipicamente entre 20 e 100). Adicionalmente, nos sistemas especialistas as regras s\u00e3o disparadas em s\u00e9rie, n\u00e3o em paralelo. Seu verdadeiro prop\u00f3sito \u00e9 conduzir a algum tipo de diagn\u00f3stico, agindo como conselheiro ou dando sugest\u00f5es (SHAW, 1999).

Uma regra fuzzy ou infer\u00eancia fuzzy relaciona conjuntos fuzzy usando o *modus ponens* do seguinte modo:

### **SE X = A ENT\u00c3O Y = B**

Onde A PERTENCE X e B PERTENCE Y. Um controlador fuzzy cont\u00e9m muitas destas infer\u00eancias fuzzy, e todas s\u00e3o ativadas em paralelo, ou seja, ao mesmo tempo. Assim, um controlador fuzzy "raciocina" com infer\u00eancia associativa paralela. Quando uma entrada \u00e9 fornecida, um controlador fuzzy dispara cada regra em paralelo com graus diferentes, dependendo de um peso chamado grau de suporte, que \u00e9 um n\u00famero no intervalo [0,1] associado a cada regra, para inferir um resultado ou sa\u00edda. Essa opera\u00e7\u00e3o paralela \u00e9 o que garante aos controladores fuzzy sua alta velocidade de processamento (por exemplo, num controlador fuzzy industrial t\u00edpico, de 3 entradas e 1 sa\u00edda, com 80 regras, o tempo de ciclo para varrer a estrutura de regras pode levar menos de um milissegundo). Sistemas fuzzy raciocinam com conjuntos ling\u00fc\u00edsticos em

vez de proposições lógicas bivalentes, a forma geral de uma inferência lógica fuzzy, associação fuzzy ou regra fuzzy é:

**SE <condições> ENTÃO <conclusão>**

ou

**SE <antecedente> ENTÃO <conseqüente>**

As <condições> relacionam-se a valores fuzzy lingüísticos de uma ou mais variáveis, como por exemplo:

**SE pressão = < muito baixa > E temperatura = < média > ENTÃO válvula = < abrir um pouco >**

Modelos de sistemas físicos estimam sua saída como função da entrada. Uma função  $f:U \Rightarrow Y$  mapeia o universo de discurso de entrada  $U$  ao universo de discurso de saída  $Y$ . Isto é, para cada elemento  $u \in U$  a função atribui unicamente um elemento  $y \in Y$ , denota-se tal atribuição única como  $y = f(u)$ . Qualquer sistema físico, químico ou biológico estima funções no sentido em que respondem e associam suas respostas com estímulos. Em outras palavras, mapeiam tais estímulos em respostas ou transformam entradas em saídas, e esta transformação define a função de entrada/saída  $f:U \Rightarrow Y$ , caracterizando o sistema em questão. Assim, qualquer sistema pode ser identificado pelo modo como opera e transforma grandezas de entrada em saída, portanto sistemas *fuzzy* também podem estimar funções de entrada e saída, por técnicas *heurísticas*:

**variável de entrada  $\Rightarrow$  inferência fuzzy (regra)  $\Rightarrow$  variável de saída**

Um especialista humano, entrevistado para ajudar a formular o conjunto de regras fuzzy, pode articular *associações de entrada/saída lingüísticas*. Assim, sistemas fuzzy podem produzir estimativas de um sistema não-linear complexo sem recorrer a modelos matemáticos. Nesse escopo, a

metodologia fuzzy é um método de estimação de entrada e saída livre de modelos matemáticos.

### **3.4.1 VANTAGENS DE CONTROLADORES FUZZY BASEADOS EM REGRAS**

Segundo (SHAW, 1999), controladores fuzzy baseados em regras têm um grande número de vantagens práticas, o que os tornaram a mais popular configuração usada pelos softwares de sistemas de desenvolvimento de controladores fuzzy.

- Regras de controle fuzzy são de fácil compreensão. Na medida em que são baseadas no senso comum, e o efeito ou resultado de cada regras pode ser facilmente interpretado.
- Todas as funções de controle associadas com uma regra podem ser testadas individualmente.
- Regras individuais combinam-se para formar uma estrutura de onde se executa um controle complexo. Essa cooperação de regras, oriunda do processamento paralelo, permite á lógica fuzzy controlar sistemas complexos usando expressões simples.
- Tanto funções de controle lineares quanto não-lineares podem ser implementadas por um sistema baseado em regras, usando o conhecimento de um especialista formulado em termos lingüísticos.
- Processamento paralelo é veloz. O controlador fuzzy completa a tarefa de processamento sem envolver muitos cálculos, e com isso a velocidade de processamento é aumentada.
- Num controlador fuzzy cada regra é processada independentemente, e com isso seu efeito no resultado final é mínimo. Isso significa que uma falha parcial no sistema não pode deteriorar significativamente a performance do controlador.

## **4 TECNOLOGIAS DE BANCOS DE DADOS**

Entende-se por banco de dados qualquer sistema que reúna e mantenha organizada uma série de informações relacionadas a um determinado assunto em uma determinada ordem, sendo assim, o sistema de banco de dados é basicamente um sistema de manutenção de registros por computador.

Os sistemas de banco de dados são projetados para armazenar e administrar grande volume de dados. O gerenciamento de dados engloba definições de estruturas para armazenamento da informação e a elaboração de mecanismos para manipulação destas informações, mantendo também a segurança e integridade das informações armazenadas.

Um dos grandes recursos oferecidos pelo banco de dados é fornecer uma visão abstrata das informações, isto é, o sistema abstrai certos detalhes de como são armazenados e mantidos os dados. As primeiras aplicações de bancos de dados eram simples e manipulavam um pequeno número de informações, exigindo para a sua construção apenas alguns dados estatísticos, tais como volumes de dados ou frequência de uso, informações que as linguagens de programação da época gerenciavam sem muita dificuldade.

Segundo (CERICOLA, 1991) uma definição de banco de dados bem aceita por vários autores seria:

“Um banco de dados é uma coleção de dados organizados e integrados, armazenados em forma de tabelas interligadas através de chaves primárias e estrangeiras, que constituem uma representação natural dos dados, sem imposição de restrições ou modificações, de forma a ser adequada a qualquer computador, podendo ser utilizado por todas as aplicações relevantes sem a necessidade de serem definidos em programas, pois utiliza as definições existentes nas bases de dados, através do dicionário de dados ativo e dinâmico.”

Com o passar do tempo, a dependência pela informática aumentou muito e conseqüentemente o número de informações a serem processadas cresceu, surgindo assim a necessidade de aplicações maiores e mais complexas. O Sistema Gerenciador Banco de Dados (SGBD) possibilita o uso de um sistema gerenciador mais poderoso onde se podem operar diversas aplicações sobre uma mesma base de dados, garantindo a consistência, integridade e rapidez no processo das informações.

Mesmo com o grande desenvolvimento dos Bancos de Dados e de suas ferramentas, a realização de algumas tarefas ainda são difíceis e algumas até muito primitivas. Assim, cada vez mais as aplicações de Banco de Dados tornam-se extremamente complexas, exigindo uma completa e rigorosa formulação, análise das necessidades de informação antes de iniciar o processo de desenvolvimento do projeto e um constante aperfeiçoamento de aplicativos e ferramentas de desenvolvimento.

#### **4.1 BANCO DE DADOS DISTRIBUÍDOS**

Um sistema de Banco de Dados Distribuído consiste em uma coleção de locais, conectados através de um sistema de rede de comunicação, no qual cada local é um sistema de banco e dados em seu próprio direito, mas os locais cooperam de forma que um usuário em qualquer local pode ter acesso a

qualquer dado da rede, exatamente como se o dados estivesse armazenado no próprio local do usuário (CERICOLA,1991).

Os sistemas distribuídos compreendem a atualização das informações cooperativas em aplicações nos computadores micro, mini e grande porte, dentro de um processamento misto. A conexão entre computadores é fundamental para implementar o uso de sistemas distribuídos.

Um sistema de Banco de Dados Distribuído é composto de uma rede de bancos de dados locais, armazenados em diversas máquinas, devendo “aparecer” para o usuário com um único banco de dados lógico, instalado em uma só máquina.

## 4.2 BANCO DE DADOS RELACIONAL

Um banco de dados relacional consiste em uma coleção de tabelas, cada qual designada por um nome único. Uma linha em uma tabela representa um relacionamento entre um conjunto de valores. Uma vez que numa tabela é uma coleção de tais relacionamentos, existe uma correspondência íntima entre o conceito de tabela e o conceito matemático de relação.

O Sistema Gerenciador de Banco de Dados Relacional (SGBDR) é o resultado de pesquisas efetuadas por um grupo de pesquisadores da IBM no final da década de 1960, e que só se tornou uma realidade estabelecida no mercado computacional em princípios da década de 1980. Historicamente, a Oracle Corporation foi a primeira que lançou no mercado um Sistema Gerenciador de Banco de Dados Relacional, instalando em 1979 esse projeto na NASA (*National Aeronautics Space Agency*) nos Estados Unidos. (MORAIS,1995)

A evolução do processamento cooperativo e a evolução dos microcomputadores pessoais como culturas separadas deram origem a um

lapso tecnológico que os sistemas distribuídos tratam de minimizar e integrar. O papel de cada cultura deve ser levado em consideração no tratamento das informações cooperativas quando surgir à polêmica entre as aplicações centralizadas e as descentralizadas, considerando a revolução que o mundo do processamento cooperativo está enfrentando.

### 4.3 BANCO DE DADOS ORIENTADO A OBJETOS

Os banco de dados orientados a objetos integram a orientação a objetos com aptidões de banco de dados. Através de construções orientadas a objetos, os usuários podem esconder detalhes de implementação de seus módulos, compartilhar a referência a objetos e expandir seus sistemas através de módulos existentes. Através dos bancos de dados, os usuários podem obter o estado em que os objetos se encontram, e estar atualizados entre as várias solicitações de programa, podendo ao mesmo tempo compartilhar a mesma informação. Os bancos de dados orientados a objeto combinam os benefícios e conceitos de orientação a objeto com a funcionalidade dos bancos de dados:

**Bancos de dados orientados a objeto = orientação a objeto + aptidões de banco de dados.**

Os bancos de dados orientados a objetos são o resultado da evolução de várias tecnologias desde a criação dos primeiros sistemas gerenciadores de banco de dados como demonstrado na figura 4.1.

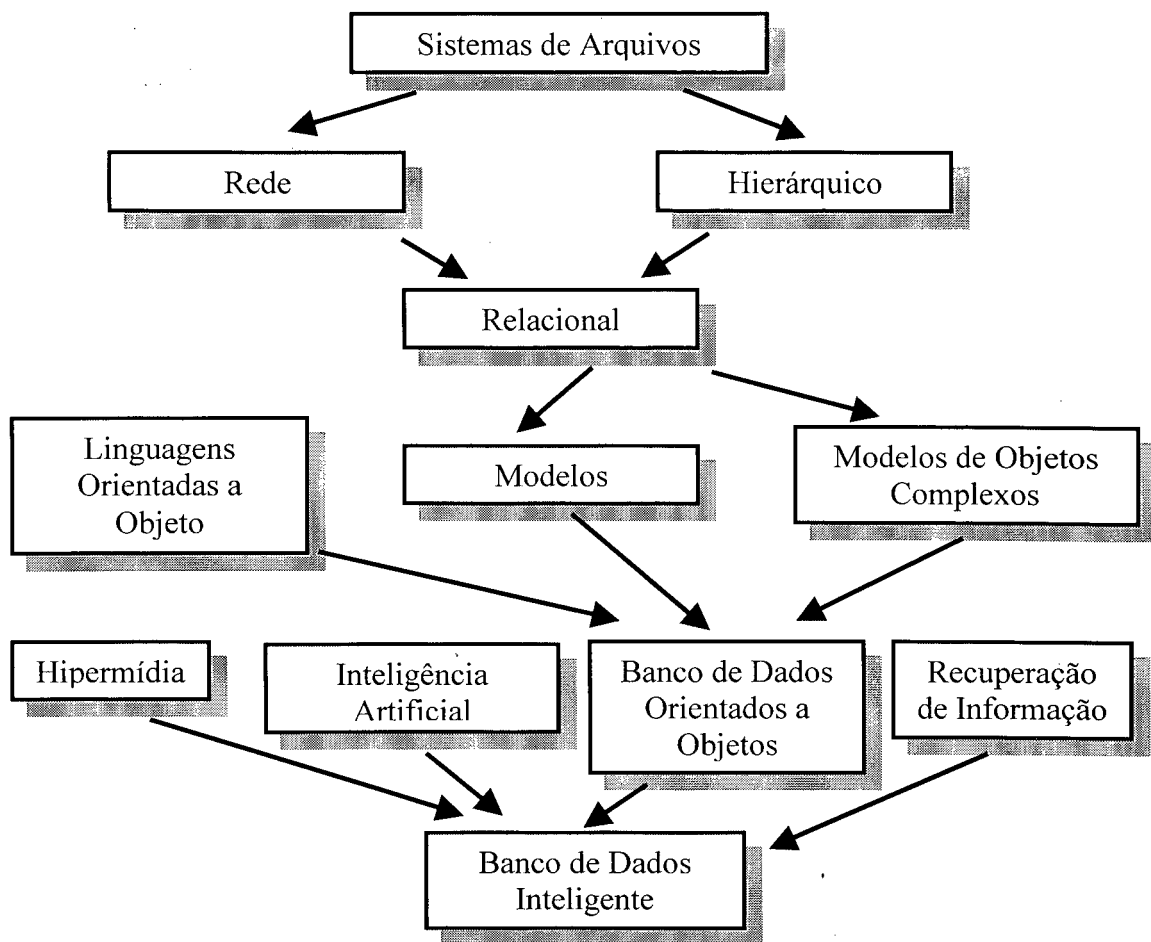


Figura 4.1 - Evolução do Banco de Dados (RAO, 1994)

A tendência dos SGDB é que se tornem cada vez mais inteligentes, no sentido de auxiliar o ser humano na manipulação e consulta de dados. E uma das tecnologias que auxiliarão para o que os Banco de Dados alcancem este estágio, é a Inteligência Artificial.



## 4.4 LINGUAGEM SQL

SQL, que é a sigla para *Structured Query Language*. Foi desenvolvida pela IBM em meados dos anos 70 como uma linguagem de manipulação de dados (DML - *Data Manipulation Language*) para suas primeiras tentativas de desenvolvimento de bancos de dados relacionais. A grande vantagem do SQL sobre modelos de dados anteriores é que as operações realizadas sobre os dados são especificadas numa linguagem não procedural e conjuntos de dados são manipulados com um único comando. Isto faz com que os desenvolvedores não tenham de navegar por uma estrutura complexa de banco de dados, reduzindo a quantidade de código necessário para acessar os dados.

O SQL tornou-se de fato o padrão depois de 1986, quando o *American National Standards Institute (ANSI)*, a organização responsável pelos padrões industriais nos Estados Unidos, endossou o SQL como linguagem padrão para os bancos de dados relacionais. Desde então, o SQL já sofreu três atualizações oficiais, em 1989, em 1992, e outra em 1999 a qual é utilizada como padrão atualmente. Na mais nova versão, SQL3, a nova característica e a mais importante é a adição de recursos de orientação a objetos na linguagem.

### 4.4.1 A LINGUAGEM SQL 3

Recentemente houve o aumento da popularidade dos bancos de dados orientados a objetos (SGBDOO – Sistemas Gerenciadores de Banco de Dados Orientado). Os bancos de dados orientados a objetos tratam os dados de modo diferente dos bancos de dados relacionais. Estes implementam acesso aos dados através de linhas e colunas, enquanto os orientados a objetos têm uma estrutura mais complexa, que não é suportada pelo padrão SQL92.

Como alguns fornecedores de SGBD começaram a acrescentar a seus produtos características além do escopo do padrão SQL92, houve a necessidade de se estudar um novo padrão, de modo a garantir a portabilidade entre os vários SGBD. Daí a necessidade do SQL3. Algumas das características implementadas incluem tipos de dados novos e estendidas, TAD, múltiplos tipos estados nulos, suporte para objetos e identidade de objetos, encapsulamento, herança e *triggers*.

O SQL92 tem um conjunto de tipos de dados atômicos. No SQL3, estes tipos de dados são mantidos como tipos de dados pré-definidos, embora alguns tenham sido modificados ou estendidos. Por exemplo, os tipos de dados *character* e *bit* foram estendidos para incluir o conceito de um "objeto maior". Estes "*large objects*" facilitam o uso de objetos multimídia dentro do banco de dados.

Dois tipos adicionais de dados, antes não suportados, foram incluídos: *boolean* e *enumerated*. O tipo *boolean* somente pode ter os valores "*true*", "*false*" e "*unknown*". O tipo *enumerated* permite que se defina domínios cujos valores são restritos a um pequeno conjunto de valores. Como por exemplo, o comando CREATE DOMAIN cores (azul, vermelho, amarelo) que define um domínio chamado cores e restringe os valores inseridos na coluna de tipo *enumerated* a um dos três valores especificados para o domínio.

O segundo grupo de objeto de dados suportado pela nova versão, e não suportado pela anterior, é o tipo abstrato de dados (TAD). O TAD é utilizado pelo SQL3 para possibilitar o uso de objetos na estrutura SQL. O TAD permite aos usuários criar novos tipos de dados definindo tipos abstratos de dados a partir de tipos de dados pré-definidos ou a partir de outros TAD já existentes. Os TAD suportam os conceitos de encapsulamento e subtipos da orientação a objetos.

Os objetos são itens de dados que combinam complexas estruturas de dados com processos ou métodos para manipular estes dados. A noção de

combinar dados e processos é chamada de encapsulamento. Enquanto os tipos de dados que no SQL92 são identificados através de seus valores, no SQL3, um dado definido como *object* será identificado por um identificador único de objeto, gerado quando o objeto é criado. Os conceitos de subtipo e supertipo são baseados no conceito de herança da orientação a objetos, que é a capacidade de se criar novos objetos a partir de outros já existentes, herdando suas características. Atributos e métodos podem ser herdados, o que diminui a redundância de informações. No SQL3, um TAD pode ser um subtipo de outro TAD. Por exemplo, para definir um TAD como um objeto PESSOA. Um subtipo de PESSOA poderia ser ESTUDANTE, que herdaria as propriedades e comportamento de PESSOA. Um subtipo do subtipo ESTUDANTE poderia ser GRADUANDO, que herdaria as características de PESSOA e ESTUDANTE.

## 5 BANCO DE DADOS FUZZY

Segundo (KLIR,1995), as aplicações da teoria de conjuntos fuzzy e lógica fuzzy dentro do campo de ciência da computação tem sido muito extensa, particularmente naquelas tentativas interessadas no armazenamento e manipulação de conhecimento em uma maneira compatível com pensamento humano. Isto inclui bancos de dados fuzzy, sistemas de recuperação de informação fuzzy, e sistemas especialistas fuzzy. Essas três áreas não são independentes umas das outras. Por Exemplo, cada sistema especialista possui um banco de dados como um sub-sistema. Por outro lado, a recuperação de informação, poderá ser tratada por um sistema de banco de dados específico. Portanto a principal diferença entre sistemas especialistas e sistemas de banco de dados é a capacidade de sistemas especialistas de criarem inferências.

A motivação para aplicação da teoria de conjunto fuzzy ao projeto de bancos de dados, é devido ao armazenamento de informação e sistemas de recuperação ignorarem a necessidade para tratar informação imprecisa. O banco de dados que pode acomodar informação imprecisa, pode armazenar e manipular não somente fatos precisos, mas também opiniões especialistas subjetivas, julgamentos, e valores que podem ser especificados em termos lingüísticos. Este tipo de informação pode ser completamente útil quando o banco de dados for utilizado como auxílio a decisões em áreas tal como diagnostico médico, emprego, investimento, e exploração geológica, onde

dados imprecisos e subjetivos não são unicamente comuns, mas totalmente valiosos.

Além disso, também auxilia o usuário em sua restrição de formular perguntas ao banco de dados em termos precisos. Perguntas vagas tal como "Que candidatas ao emprego são altamente educadas e moderadamente experientes?", ou "Que hotéis com preços razoáveis estão localizados próximos ao centro de cidade?" Frequentemente são assuntos relacionados aos usuários de banco de dados, portanto estes tipos de perguntas são mais fáceis de serem formuladas e pesquisadas, do que perguntas precisas.

O sistema de banco de dados que associa imprecisão é capaz de propagar apropriadamente o nível de incerteza associada com os dados, com nível de incerteza associada às respostas ou conclusões baseadas nos dados.

## 5.1 UMA PROPOSTA PARA BANCOS DE DADOS FUZZY

Vários modelos para representar informação em bancos de dados têm sido propostos. Um destes modelos, referidos como **modelo do relacional**, tem tornado-se predominante. Virtualmente todos bancos de dados fuzzy descritos na literatura são concebidos em termos do modelo relacional.

Em 1982 Buckles e Petry desenvolveram um modelo para um banco de dados relacional fuzzy que contem, como um caso especial, o modelo crisp clássico de um banco de dados relacional. O modelo de um banco de dados relacional clássico consiste de um conjunto de relações multidimensionais conceitualizadas como tabelas. As colunas destas tabelas correspondem aos campos ou atributos e são usualmente chamados domínios. Cada domínio é definido sobre uma base de domínio apropriada, ou conjunto universal. As linhas são elementos da relação, elas correspondem aos registros ou entradas, e são chamadas tuplas. O acesso ao banco de dados é efetuado através de uma álgebra relacional, consistindo da aplicação procedural de operações

contendo quatro elementos básicos como: um nome da operação; os nomes das relações e os nomes dos domínios para serem operados, e uma expressão condicional opcional. Por Exemplo, se determinado banco de dados contem uma relação ternária ESTUDANTE, com os domínios NOME, ENDEREÇO e CURSO, pode-se obter os nomes e endereço de todos estudantes cujo curso seja ciência da computação por construir uma nova relação com domínios NOME e ENDEREÇO como uma projeção da relação original. A operação algébrica que desempenha esta tarefa poderia ser:

**Projetar (ESTUDANTE: NOME, ENDEREÇO) onde  
CURSO = "ciência da computação"**

A álgebra também contem outras operações relacionais, tal como, **Complemento, União, Interseção, e Junção**, que desempenha tarefas correspondentes na relação e domínios especificados para produzir a informação desejada.

Porém o banco de dados do relacional Fuzzy proposto por Buckles e Petry difere deste modelo crisp em dois aspectos. Os primeiros elementos das tuplas contidos nas relações podem ser subconjuntos crisp do domínio conjunto universal, e no segundo, uma relação de semelhança é definida em cada domínio conjunto universal. A primeira qualificação permite aos elementos das tuplas consistência única do domínio dos conjuntos universais (como no modelo de banco de dados do relacional convencional) ou subconjuntos crisp do domínio conjuntos universais. Por exemplo, na relação MERCADO com os domínios ÁREA, TAMANHO, e POTENCIAL representado pela tabela 5.1.

<b>RELAÇÃO: MERCADOS</b>		
<b>ÁREA</b>	<b>TAMANHO</b>	<b>POTENCIAL</b>
Oeste	Grande	bom
Meio-oeste	(grande, médio)	(moderado, bom)
Sul	pequeno	(bom, excelente)

Tabela 5.1 - Relação Mercados

Valores de domínio que não são únicos podem indicar, por exemplo, a união das opiniões ou julgamentos de vários especialistas.

A segunda qualificação é baseada na suposição que no modelo de banco de dados clássico, onde uma relação crisp equivalente é definida em cada conjunto do domínio universal, o qual agrupa juntamente elementos que são estritamente equivalentes. Esta equivalência é utilizada, por exemplo, quando tuplas redundantes são eliminadas ou ignoradas. Mais frequentemente, as classes de equivalência geradas por esta relação são simplesmente as únicas do conjunto universal. No modelo de banco de dados fuzzy, esta relação de equivalência é generalizada para uma relação de equivalência fuzzy (ou uma relação de semelhança). Esta introdução de nebulosidade fornece um elemento interessante de flexibilidade, visto que o valor ou significado das estruturas de diferentes usuários de banco de dados individuais, podem ser refletidos por modificar o domínio das relações equivalentes de forma correta.

A álgebra do relacionamento fuzzy utiliza para acessar o banco de dados fuzzy os mesmos quatro componentes da álgebra relacional convencional e, em adição, leva em conta a especificação de um nível de entrada, definindo o **mínimo** grau aceitável de semelhança entre os elementos em algum domínio especificado. No caso especial do banco de dados convencional, todos os níveis de entrada são implicitamente assumidos por ser **igual a 1**, desta forma requer equivalência total para unir ou eliminar as tuplas. No banco de dados fuzzy, tuplas podem ser unidas se elas são consideradas suficientemente parecidas.

Como exemplo da utilização do modelo de banco de dados fuzzy e da associação da álgebra relacional fuzzy, supõe-se que determinado banco de dados contenha as opiniões de um grupo de especialistas em três opções de política, **X**, **Y**, e **Z**.

Duas relações são armazenadas no banco de dados:

- ESPECIALISTAS, que possuem os domínios: NOME e CAMPO, associando o nome e campo de cada especialista;
- AVALIAÇÃO, que tem os domínios: OPÇÃO, NOME e OPINIAO, associando o nome de cada especialista as suas opiniões expressas das opções de política.

Estas duas relações são especificadas na tabela 5.2. Em adição, a seguinte relação de compatibilidade fuzzy onde é definida à OPINIÃO no conjunto universal de domínio (altamente favorável (**AF**), favorável (**F**), ligeiramente favorável (**LF**), ligeiramente negativa (**LN**), negativa (**N**), e altamente negativa (**AN**), (tabela 5.2):

	<b>AF</b>	<b>F</b>	<b>LF</b>	<b>LN</b>	<b>N</b>	<b>AN</b>
<b>AF</b>	1	.8	.6	.2	0	0
<b>F</b>	.8	1	.8	.6	.2	0
<b>LF</b>	.6	.8	1	.8	.6	.2
<b>LN</b>	.2	.6	.8	1	.8	.6
<b>N</b>	0	.2	.6	.8	1	.8
<b>AN</b>	0	0	.2	.6	.8	1

Tabela 5.2 - Relação de Compatibilidade

As relações de equivalência crisps em que classes de equivalências são únicas, são assumidas para ser definidas nos domínios NOME, CAMPO e OPÇÃO.

Supõem-se a seguinte pergunta para tal banco de dados fuzzy:

"Que sociólogo está de considerável acordo com o Kass em relação a opção de política Y?" O primeiro passo será recuperar a opinião de Kass em relação a opção Y (tabela 5.3). Isto é efetuado com a operação de álgebra relacional:



RELAÇÃO: ESPECIALISTA		RELAÇÃO: AVALIAÇÃO		
NOME	CAMPO	OPÇÃO	NOME	OPINIÃO
Cohen	Sociologo	X	Osborn	Favorável
Fadem	Economista	X	Fee	Negativa
Fee	Advogado	X	Fadem	Ligeiramente Favorável
Feldman	Economista	X	Feldman	Altamente Favorável
Kass	Médico	Y	Cohen	Ligeiramente Favorável
Osborn	Sociologo	Y	Osborn	Ligeiramente Favorável
Schreiber	Sociólogo	Y	Fee	Altamente Favorável
Spccterman	Sociólogo	Y	Schreiber	Favorable
		Y	Kass	Favorável
		Y	Fadem	Negativa
		Y	Specterman	Altamente Favorável
		Y	Feldman	Ligeiramente Negativa
		Z	Osborn	Negativa
		Z	Kass	Ligeiramente Negativa
		Z	Fee	Ligeiramente Favorável

Tabela 5.3 - Relação Especialista, e Relação Avaliação

**(Projeção ( Seleção AVALIAÇÃO onde NOME = "Kass" e OPÇÃO = Y)  
sobre OPINIÃO) resultando R1)**

O resultado é uma relação temporário R1(tabela 5.4) no domínio opinião é dada por:

RELAÇÃO: R1 OPINIÃO
Favorável

Tabela 5.4 - Relação temporária R1

O próximo passo envolve a seleção de todos os sociólogos da tabela de especialistas. Isto é efetuado pela operação:

**(Projeção(Seleção ESPECIALISTA onde CAMPO = Sociólogos) sobre NOME) resultando R2)**

Neste caso, R2 (tabela 5.5) é uma relação temporária no domínio NOME, que contém a listagem dos sociólogos:

RELAÇÃO: R2 NOME
Cohen
Osborn
Schneiber
Specterman

Tabela 5.5 - Relação temporária R2

Uma relação temporária R3 deve ser construída nos domínios NOME e OPINIÃO com as opiniões dos sociólogos em R2 sobre a opção Y. A expressão algébrica para esta operação é:

**(Projeção (Seleção(Junção R2 e AVALIAÇÃO sobre o NOME) onde OPÇÃO = Y) sobre NOME, OPINIÃO) resultado R3**

A relação R3 (tabela 5.6) que é produzida é dada por:

RELAÇÃO: R3	
NOME	OPINIÃO
Cohen	ligeiramente negativa
Osborn	ligeiramente favorável
Schneiber	favorável
Specterman	altamente favorável

Tabela 5.6 - Relação temporária R3

Finalmente, desempenha-se uma junção na relação de R1 (resultando a opinião de Kass) e R3 (resultando a opinião dos sociólogos) que especifica um nível de entrada similar a **.75** no domínio OPINIÃO, que foi escolhido como exemplo para representar a condição de acordo "considerável". A expressão algébrica para esta consulta é:

**(Junção R3 e R1 sobre a OPINIÃO) com  
ENTRADA(OPINIÃO) ≥ .75 e ENTRADA(NOME) ≥ 0**

A especificação de um nível de entrada semelhante a zero para NOME, é necessária para permitir a união de nomes dentro dos conjuntos, como mostrados (tabela 5.7) no resultado dado por:

<b>NOME</b>	<b>OPINIÃO</b>
Osborn Schreiber Specterman	ligeiramente favorável favorável altamente favorável

Tabela 5.7 - Resultado Final

Desta forma, a incerteza contida na especificação de "acordo considerável" e na semelhança definida sobre as opiniões possíveis é propagada à resposta dada.

O modelo de banco de dados fuzzy ilustrado foi desenvolvido por (BUCKLES; PETRY, 1982), e introduz nebulosidade unicamente por meios de relações de equivalência fuzzy ou, mais geralmente, relações de compatibilidade fuzzy em conjuntos individuais de domínios universais. Nebulosidade em dados não é demonstrador por este modelo. Entretanto, outros modelos de banco de dados fuzzy têm sido propostos em que valores de atributos podem ser expressos em termos lingüísticos fuzzy.

## 5.2 PROCESSAMENTO DE CONSULTAS FUZZY

As arquiteturas propostas para o processamento de consultas envolvendo incertezas possuem 3 abordagens básicas (BOSC;PIVET,1992).

A primeira abordagem se constitui de uma divisão da consulta em duas partes.

- A primeira é constituída de sentenças booleanas convencionais e é submetida diretamente ao SGBD.

- A segunda parte, contendo sentenças imprecisas ou nebulosas é utilizada posteriormente para executar um “ranking” nas tuplas recuperadas na primeira parte da consulta.

As duas partes são na verdade conectadas por uma conjunção E. A parte imprecisa pode ser constituída por proposições nebulosas do tipo “Em torno de 30” e combinadas pelos operadores E e OU nebulosos usualmente definidos pelos operadores min e max. Este sistema (BOSC;PIVERT,1992) é o PREFERENCES (LACROIX;LAVENCY,1987), que possui como problema principal o fato de que a consulta imprecisa não faz parte efetivamente da consulta, apenas é usada para “ranking”.

A segunda abordagem permite a utilização direta de sentenças imprecisas. As sentenças são traduzidas transformando os valores imprecisos em intervalos. Em alguns sistemas é utilizado o modelo de similaridade onde o usuário deve definir os limites de confiança ou através de funções normalizadas de similaridade dentro do domínio dos atributos. A vantagem deste modelo, bem como o anterior é a sua simplicidade de implementação, (BOSC;PIVERT, 1997) inclusive sugerem a derivação da parte booleana da consulta para um SGBD convencional, resultando assim num aproveitamento das vantagens já existentes do sistema.

O problema básico desta abordagem é que ao traduzir os termos imprecisos em intervalos os limites dos mesmos apresentam a mesma descontinuidade dos valores nítidos.

A terceira abordagem, inicialmente pesquisada por (TAHANI,1976), (TAHANI,1977) e outros, é baseada na teoria dos conjuntos nebulosos(ZADEH,1965) e utiliza o conceito de relação nebulosa. As operações entre as relações são as extensões das operações básicas (união, interseção, etc.), para as relações nebulosas. As sentenças envolvem diretamente consultas precisas e imprecisas, utilizando conjunções e disjunções entre ambos os tipos de consultas. Os atributos dos predicados

envolvem variáveis lingüísticas, termos nebulosos e modificadores. Esta abordagem ainda propõe a introdução de quantificadores nebulosos às consultas. Ela é a mais complexa para implementação porque sugere a inclusão de algoritmos especializados dentro do SGBD, permitindo assim extensão do uso de suas capacidades. No modelo criado por (LI;LIU,1990) para um banco de dados nebuloso baseado em PROLOG, a arquitetura do sistema (figura 5.1) proposto se divide nos níveis Interno, Conceitual e Externo, correspondendo respectivamente às visões física, lógica e conceitual do banco de dados.

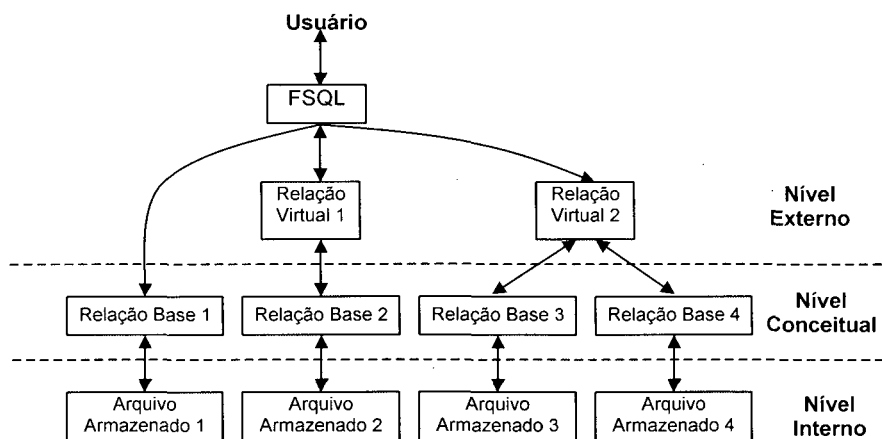


Figura 5.1 – Níveis do Banco de Dados Fuzzy

No nível do usuário é estabelecida a linguagem de definição de dados nebulosa e de consulta. A característica especial da linguagem é a definição de atributos nulos ou incompletos.

Um comparativo entre as 3 abordagens sugeridas pode ser observado na figura 5.2.

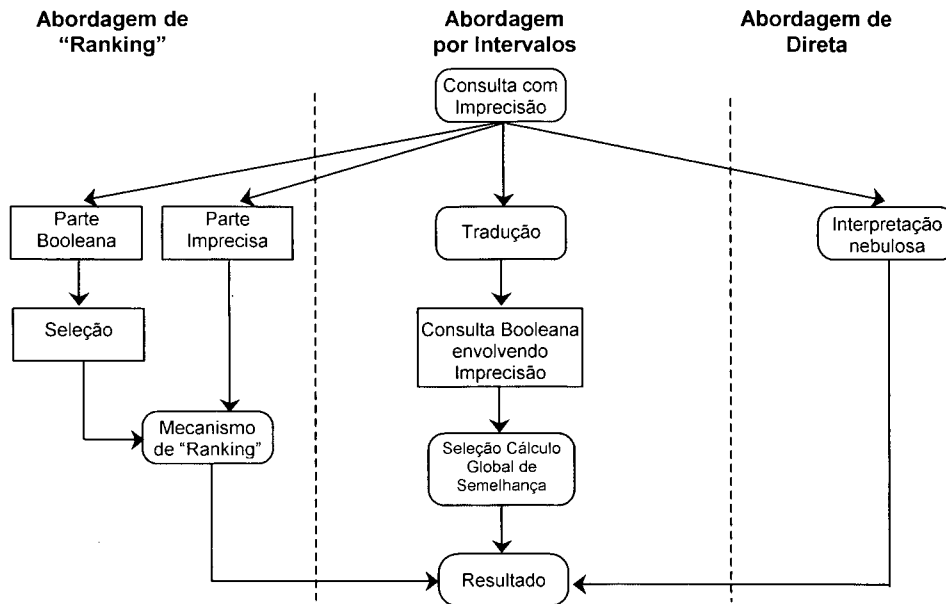


Figura 5.2 - Abordagens de Arquitetura Envolvendo Incertezas

### 5.3 LINGUAGENS DE CONSULTA FUZZY

Todas as linguagens de consulta que incorporam a teoria nebulosa, utilizam o SQL como base. De um modo geral as linguagens tem o mesmo formato proposto para o SQL, entretanto, a diferença está nas sentenças de criação das tabelas e nas proposições utilizadas nas condições da consulta.

Os comandos de criação das tabelas podem possuir palavras chave especiais para designar os atributos que serão nebulosos (PETRY,1996), ou simplesmente assumir que qualquer variável pode admitir valores nebulosos(LI;LIU,1990).

Em alguns casos a linguagem dispõe de um termo extra no SQL padrão para designar o grau de confiança da nebulosidade das tuplas do resultado(LI,LIU,1990):

**SELECT** <Atributos>  
**FROM** <Relações>  
**WHERE** <Condições>  
**WITH** <Grau de pertinência ou termo nebuloso de segunda ordem>

Supõem-se, por exemplo, a necessidade de recuperar todos os funcionários jovens de uma empresa com certeza maior que 80%. A sentença nebulosa ficaria:

```
SELECT  IDADE, FUNCIONARIO
FROM    EMPRESA
WHERE   IDADE = 'Jovem'
WITH    0.8
```

O termo seguinte à cláusula **WITH** pode também ser um termo nebuloso mapeando o grau encontrado na tupla numa outra possibilidade, como por exemplo, "Possível":

```
SELECT  IDADE, FUNCIONARIO
FROM    EMPRESA
WHERE   IDADE = 'Jovem'
WITH    Possível
```

O resultado desta consulta seria uma lista de funcionários, e a pertinência de sua tupla em relação ao termo "Possível".

Os operadores de seleção, projeção e união, têm o mesmo formato do SQL padrão, contudo, seu método de cálculo difere, desta forma uma álgebra nebulosa é em geral proposta sobre múltiplas maneiras.

## 5.4 ÁLGEBRA RELACIONAL NEBULOSA

As propostas de extensão nebulosa da álgebra relacional são feitas sobre a extensão das operações entre as relações nebulosas (LI;LIU,1990), (PETRY,1996), (SHENOI,1990). As operações da álgebra estão divididas em dois grupos, o primeiro se refere às operações básicas entre conjuntos, o segundo contém as operações de seleção, projeção e junção.

Dadas duas relações nebulosas compatíveis  $R$  e  $S$  com funções de pertinência  $\mu_R(t)$  e  $\mu_S(t)$ , onde  $t$  indexa as tuplas de  $R$  e  $S$ , as operações básicas de conjunto são(LI;LIU,1990):

- **União** ( $R \cup S$ ): As tuplas resultantes da união de  $R$  e  $S$  podem não aparecer simultaneamente em ambas as relações, desta forma, cada tupla  $t$  que aparece nas duas relações terá seu valor de pertinência calculado pelo máximo das duas tuplas:

$$\mu_{R \cup S}(t) = \max\{\mu_R(t), \mu_S(t)\}$$

As tuplas de uma relação que não aparecem na outra levam o seu próprio valor de pertinência. As duas relações devem ser compatíveis, ou seja, o número de atributos e os domínios de cada atributo devem ser os mesmos. Esta restrição se aplica as operações de interseção e diferença.

- **Interseção** ( $R \cap S$ ): Cada tupla  $t$  que está em  $R$  e  $S$ , tem sua pertinência calculada por:

$$\mu_{R \cap S}(t) = \min\{\mu_R(t), \mu_S(t)\}$$



• **Diferença (R - S):** A diferença entre duas relações pode também ser vista como a interseção de R com a negação de S, cuja função de pertinência é dada por:

$$\mu_{R - S}(t) = \min\{\mu_R(t), (1 - \mu_S(t))\}$$

• **Produto Cartesiano (R1 × R2 × ... × Rn):** O produto cartesiano pode ser feito por relações não compatíveis. O conjunto de tuplas final  $t = (t_1, t_2, \dots, t_n)$  é composto pelo produto cartesiano das tuplas  $t_i$  de cada relação  $R_i$ . A pertinência do produto cartesiano é dada por:

$$\mu_{R_1 \times R_2 \times \dots \times R_n}(t) = \min\{\mu_{R_1}(t_1), \mu_{R_2}(t_2), \dots, \mu_{R_n}(t_n)\}$$

• **Seleção** - A avaliação de um predicado  $\phi$  sobre uma relação R envolvendo um conjunto de atributos  $A_1, A_2, \dots, A_n$ , um conjunto de elementos do domínio de cada atributo  $D_f(A_1), D_f(A_2), \dots, D_f(A_n)$  e um conjunto de operadores nebulosos  $op_1, op_2, \dots, op_n$  é feita substituindo-se cada elemento  $d_i \in D_f(A_i)$  de cada tupla nos atributos  $A_i$  envolvidos. A pertinência  $\mu_{\sigma_\phi}(R)$  de cada tupla da seleção  $\sigma_\phi(R)$  é feita através do cálculo entre os operadores nebulosos, que podem ser lógicos (AND, OR, NOT) aritméticos (+, -, \*, /, =, <, >, ≤, ≥) ou operadores nebulosos especiais.

• **Projeção** - A projeção de uma relação nebulosa (tabela 5.8)  $R = (D_f(A_1) \times D_f(A_2) \times \dots \times D_f(A_n))$  sobre um grupo de atributos  $A_1, A_2, \dots, A_k, K \leq n$  pode gerar um conjunto de tuplas redundantes. As tuplas  $t$  de menor pertinência são eliminadas:

$$\mu_R(d_{i_1}, d_{i_2}, \dots, d_{i_k}) = \max_{d_j \in D_f(A_j), j \in \{i_1, i_2, \dots, i_k\}} \{\mu_R(d_1, d_2, \dots, d_n)\}$$

NOME	NOME	$\mu$
José	Jóvem	0.5
Pedro	Adulto	0.7

Tabela 5.8 – Projeção de uma relação nebulosa

• **Junção** - A operação de junção consiste de um produto cartesiano seguido de uma seleção. Em (LI;LIU,1990) a junção é calculada sobre as extensões cilíndricas (ZADEH,1965) de cada relação em relação a todos os atributos do universo, desta forma, a comparação entre duas tuplas da junção pode ser sempre feita entre tuplas compatíveis. A função de pertinência final da junção é dada por:

$$\mu_{\triangleright \triangleleft_{i=1}^n R_i}(a_1, a_2, \dots, a_n) = \min\{\mu_{R_1}(a_1, a_2, \dots, a_n), \dots, \mu_{R_n}(a_1, a_2, \dots, a_n)\}$$

Neste processo leva-se em conta somente o grau de relevância das extensões cilíndricas.

Outra alternativa é inserir dentro do cálculo a pertinência do predicado de junção (PETRY,1996):

$$\mu_{R \triangleright \triangleleft_{A \theta B} S}(x, y) = \min\{\mu_R(x), \mu_S(y), \mu_{\theta}(x.A, y.B)\}$$

De todo modo, não fica claro como dois valores nebulosos de duas tuplas podem ser comparados, visto que, no caso de conjuntos nebulosos não possui igualdade.

## 5.5 CÁLCULO RELACIONAL

O cálculo nebuloso de predicados tem a mesma forma do cálculo de predicados de primeira ordem (LI;LIU,1990), (PETRY,1996). A diferença básica

está em que a interpretação das fórmulas não assume somente verdadeiro ou falso, mas graus de pertinência.

Dada uma relação  $R_{(\lambda)} = \{t \mid \mu_R(t) \geq \lambda\}$ , onde  $\lambda$  é o nível mínimo de pertinência da tupla  $t$  variando em  $R$ , uma expressão é definida recursivamente pelas suas fórmulas atômicas  $\Psi$  que somente podem ser:

- $\mu_R(t)$
- $t[i] \theta C$  ou  $C \theta t[i]$ , sendo que  $t[i]$  é o  $i$ -ésimo atributo da tupla  $t$  e  $\theta$  pode ser qualquer operador de comparação nebuloso ou não.  $C$  é uma constante que pode ser um valor nebuloso ou não.
- $t[i] \theta u[j]$ , onde  $t$  e  $u$  são atributos de duas tuplas como definido acima.
- $\neg\Psi$ , a operação nebulosa NOT é aplicada a fórmula.
- $\Psi_1 \wedge \Psi_2$ , a conjunção de duas fórmulas através do operador nebuloso AND.
- $\Psi_1 \vee \Psi_2$ , a disjunção de duas fórmulas através do operador nebuloso OR.

O cálculo de predicados é basicamente utilizado para as operações de seleção de tuplas. A avaliação das fórmulas resulta num grau de pertinência que classifica ou rejeita as tuplas selecionadas em uma consulta. Isto pode ocasionar em perda de informação, visto que algumas implementações das operações de AND e OR nebulosas podem retornar resultados iguais para vários valores de pertinência diferentes, como por exemplo, uma consulta do tipo:

```

SELECT    PACIENTE, PRESSÃO, TEMPERATURA
FROM      PACIENTES
WHERE     PRESSÃO = Alta AND TEMPERATURA = Alta

```

Na tabela 5.9 pode-se observar que as duas tuplas recuperadas apresentariam o mesmo grau de pertinência, entretanto, a ordem de importância do paciente P1, com maior TEMPERATURA parecer mais visível.

PACIENTE	PRESSÃO	$\mu$	TEMPERATURA	$\mu$	AÇÚCAR	$\mu$
P1	Alta	0.6	Alta	0.9	Baixo	0.2
P2	Alta	0.6	Alta	0.7	Baixo	0.7

Tabela 5.9 – Tuplas selecionadas utilizando cálculo de predicados

Segundo proposta de (BERTHIER;MUNTZ,1995), as tuplas deveriam ser comparadas levando-se em consideração todos os operandos do predicado da consulta. A comparação não seria na mesma ordem dos atributos da relação, mas teria a ordem alterada através de operadores sincronizados com as operações de conjunção e disjunção, como:

$\Psi = (\text{PRESSÃO} = \text{Alta} \text{ AND } \text{TEMPERATURA} = \text{Alta}) \text{ OR } (\text{AÇÚCAR} = \text{Baixo})$

seria escrito convencionalmente por:

$$\mu_{\Psi}(t) = \max(\min(\mu_{\text{Pressão}}(t), \mu_{\text{Temperatura}}(t)), \mu_{\text{Açúcar}}(t)).$$

onde  $t$  é uma tupla qualquer pertencente à  $R$ . A proposta de (BERTHIER;MUNTZ,1995), é de dois operadores, chamados de  $F$  e  $G$ , os quais substituem as operações de mínimo e máximo. O operador  $F$  ordena os seus operandos de forma crescente e  $G$  de forma decrescente. A lista de graus de pertinência devidamente ordenada de cada tupla é então comparada e classificada. A consulta  $\Psi$  seria escrita como:

$$\Psi = G (F(\text{Pressão}, \text{Temperatura}), \text{Açúcar}),$$

A classificação obtida é exibida na tabela 5.10, onde os graus de pertinência são ordenados da esquerda para a direita.

PACIENTE	$\mu$	$\mu$	$\mu$
P1	0.6	0.9	0.2
P2	0.7	0.6	0.7

Tabela 5.10 – Tuplas selecionadas utilizando todos os operandos do predicado da consulta

As arquiteturas propostas visam estender as operações lógicas e a teoria dos conjuntos para adaptar o modelo e o cálculo relacional respectivamente, entretanto, existem outras propostas, como, por exemplo, utilizando o banco de dados orientado por objetos ou o modelo rede (PETRY, 1996). A maioria destas propostas utiliza um banco de dados nebuloso que deve ser construído ou convertido de um banco de dados convencional. A filosofia de bancos de dados nebulosos tem sido proposta em vários tipos de sistemas onde se requer tratamento de imprecisão.

## 5.6 MODELO FÍSICO DO BANCO DE DADOS FUZZY

Grandes esforços de pesquisas foram gastos em métodos por representar informação imprecisa em vários modelos de banco de dados utilizando a teoria dos Conjuntos Fuzzy. Porém, a pesquisa dirigida a estruturas de acesso para manipular consultas Fuzzy, está efetivamente ainda em uma fase imatura. Um tempo de resposta rápido é muito importante quando os sistemas de banco de dados tratam com dados imprecisos (fuzzy). As atuais estruturas de índice crisp são inapropriadas para representar eficientemente estes dados, para uma melhor eficiência torna-se necessário indexar ambos os atributos crisp e fuzzy juntos (YAZICI; GEORGE, 1999).

A manipulação de dados é simples onde o armazenamento e recuperação de dados são precisos, porém quando as informações precisas estão indisponíveis, é freqüente o caso que alguma informação relevante esteja em forma imprecisa.

### **5.6.1 PERFORMANCE EM UM BANCO DE DADOS FUZZY**

Alguns modelos de banco de dados, assim como banco de dados de relacionais (BOSC; GALIBOURG; HAMON, 1988; BUCKLES; PETRY, 1982), modelos de banco de dados de segunda forma normal (NF<sup>2</sup>) (YAZICI; BUCKLES; PETRY, 1995) e modelos de banco de dados orientados a objetos (BORDOGNA, 1994; YAZICI; KOYUNCU, 1997), já foram estendidos para lidar com imprecisão utilizando a teoria dos conjuntos Fuzzy (ZADEH, 1965; ZADEH, 1992). Porém alguns aspectos de implementação sobre estes trabalhos não receberam atenção suficiente como, por exemplo, para um item essencial em sistemas de banco de dados, o tempo de resposta. A eficiência fica mais crucial quando se trabalha com nebulosidade, devido as causas adicionais de degradação de desempenho, como por exemplo, a grande quantidade de dados armazenados, a natureza altamente numérica dos métodos propostos e, o mais importante, a insuficiência das estruturas de acesso existentes. Desta forma, pesquisas em estruturas de acesso fuzzy são indispensáveis para examinar bancos de dados fuzzy eficazmente.

A crucial diferença entre consultas fuzzy e consultas crisp, é o número de registros alocados na memória. Um número maior de tuplas será selecionado através de condições fuzzy. Se um registro consiste de um atributo fuzzy "altura", e realiza-se uma consulta "encontre todas as pessoas altas", uma porção considerável do banco de dados será é retornada para memória (YAZICI; GEORGE, 1999).

Consultas fuzzy permitem expressar predicados vagos representados pelos conjuntos fuzzy. Neste caso, o caminho de acessos das estruturas de índice existente não pode ser usado diretamente, visto que predicados fuzzy

podem não se referir valores de entrada do índice. A consulta fuzzy também tem outras diferenças sobre as consultas crisp. Por exemplo, até mesmo quando a consulta crisp produz uma resposta vazia, a consulta fuzzy correspondente pode prover uma resposta apropriada. Isto é devido ao fato que consultas crisp são muito rígidas, enquanto que consultas fuzzy são flexíveis. Além disso, consultas fuzzy também possibilitam um ranking das possíveis respostas, enquanto que as consultas crisp clássicas só provem uma lista de respostas não diferenciadas.

A estrutura de acesso proposta por (BOSC;PIVERT, 1992;PETRY, 1996), é usar um índice por predicados fuzzy amarrados em um atributo. O princípio é associar cada grau de um predicado fuzzy com uma lista de tuplas que satisfaz aquele predicado. Este método só trata de domínios homogêneos, e assume que as relações subjacentes são crisp. Porém, em muitas aplicações complexas, é necessário lidar com dados crisp e fuzzy de alguns atributos fuzzy-valorados em bancos de dados, causando complicações adicionais para o eficiente acesso aos dados de tais bancos de dados. Uma possível estrutura de acesso eficiente para recuperação de dados fuzzy deveria ser multidimensional. A razão para isto é aqueles atributos não-fuzzy e fuzzy precisam ser usados como atributos organizadores para a eficiência de ambas as consulta, fuzzy e crisp. Um grupo de atributos que são escolhidos do conjunto de todos os atributos para organizar o arquivo são chamados de atributos organizadores. A seleção destes atributos normalmente está baseada em tipos de consultas comuns no banco de dados, e em outras exigências físicas de projeto.

Utilizando uma estrutura de multidimensional a efetividade de um de banco de dados fuzzy aumenta. O custo de armazenamento incorrido por esta estrutura de índice (em comparação a aproximações alternativas) é mais baixo. Para uma eficiente estrutura multidimensional de acesso a dados, vários modelos poderão ser utilizados: K-d Trees (BANTLEY, 1975; BANTLEY, 1979; BANTLEY;MAURER, 1980), grid files (FREESTON,1987; NIEVERGELT, 1984), Multi Level Grid File (WHANG, 1991), e os outros (SALZBERG, 1988).

### 5.6.2 ESTRUTURA DE GRID FILE

No modelo de banco de dados de relacional convencional, uma relação  $R$  de grau  $N$  e cardinalidade  $M$  está composto de  $N$  colunas e  $M$  linhas onde as colunas denotam os atributos e as linhas denotam as tuplas. Um atributo  $A_i$  de tupla  $T$  leva valores de domínio  $D_i$  onde os domínios de  $R$  são  $D_1, D_2, \dots, D_n$ . Um atributo crisp pode levar valores de um único tipo de dados em um modelo de banco de dados de relacional. Porém, um atributo fuzzy pode representar valores crisp e fuzzy, por exemplo, o atributo que representa a renda mensal pode expressar um valor preciso como 1000 ou um valor fuzzy, como baixo.

As estruturas de arquivos tradicionais são projetadas para lidar com chave única para acelerar o processo de consulta. Porém, as exigências de aplicações mais complexas fizeram as estruturas de acesso tradicionais (SALZBERG, 1988) ineficazes. O Grid File Structure foi desenvolvida para superar as desvantagens de estruturas de dados unidimensionais. Grid File é uma estrutura de arquivos multidimensional que divide o espaço de domínio em um espaço  $K$ -dimensional, com  $K$  atributos organizados em um registro. A estrutura consiste em um diretório de grid (a raiz) e em blocos de dados (as folhas). O diretório do grid que é uma estrutura de matriz  $k$ -dimensional em uma seqüência contígua de blocos de disco, onde cada entrada de diretório tem um ponteiro para uma região do grid.

Cada região do grid é armazenada em um recipiente de dados, porém, mais de uma região do grid pode ser armazenado em um recipiente de dados, visto que a utilização de dados poderá ser muito baixa se cada recipiente contiver uma única região de grid. As regiões de grid designadas a um recipiente de dados precisam formar um convexo (retangular), dividindo o espaço do atributo, onde o diretório e os arquivos de dados são mantidos em um armazenamento secundário. Para cada dimensão são armazenadas escalas lineares na memória principal. Eles são usados para calcular a posição da entrada de diretório.



Então, uma consulta exata pode ser respondida em dois acessos de disco. Um para o diretório de grid e o outro para os blocos de dados. Quando um recipiente de dados encher, é dividido em dois, neste caso a entrada de diretório do grid que aponta para a região do grid dividida, também deve ser dividida. Conseqüentemente, se o Grid File for k-dimensional, a divisão é executada junto com k-i dimensões. Embora a estrutura de Grid File possa ser utilizada efetivamente enquanto que a distribuição de dados seja uniforme. O crescimento do diretório, o custo de divisão de diretório e as operações de união podem alcançar uma taxa exponencial quando há uma correlação entre os atributos.

Por exemplo, supõe-se que a necessidade de um arquivo das cidades na Turquia onde se necessita representar cada cidade pelos atributos, latitude e longitude. A estrutura do Grid File pode ser usada para armazenar as cidades juntas em um arquivo com os dados organizando pelos valores da latitude e longitude. O diretório do grid e registros de dados do Grid File são demonstrados na figura 5.3.

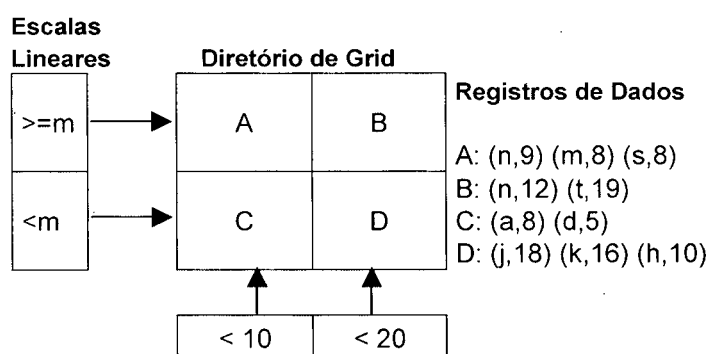


Figura 5.3 – Exemplo da Estrutura de Grid File

### 5.6.3 INDEXAÇÃO FUZZY

Estruturas de acesso convencionais não podem ser usadas diretamente, visto que predicados fuzzy não se referem a valores de entrada do

índice. Bosc et al. (BOSC;PIVERT, 1992), propôs uma estrutura de índice por predicados fuzzy amarrados a um atributo.

O princípio do índice é associar cada grau de um predicado fuzzy com a lista de tuplas que satisfaça o predicado. Este esquema de indexação resulta em muitos índices por predicado fuzzy amarrados a um atributo, porém aqueles predicados emitidos do mesmo termo primário fuzzy que usa modificadores (muito, mais ou menos,...), podem usar o mesmo índice. Nesta proposta, um acesso em graus de pertinência é provido para determinado termo primário que só se refere a valores existentes do atributo interessado. Entretanto, um nível adicional é adicionado no índice para cada termo primário, este nível contém os valores de pertinência de cada predicado fuzzy. Por exemplo, considera-se o predicado fuzzy (Salário = 'Alto') endereçando a relação crisp Empregado (Nome, Salário) como mostrado na Figura 5.4. Usando esta estrutura de índice, também é possível acessar os empregados que satisfazem mais ou menos o predicado (Salário = 'Alto').

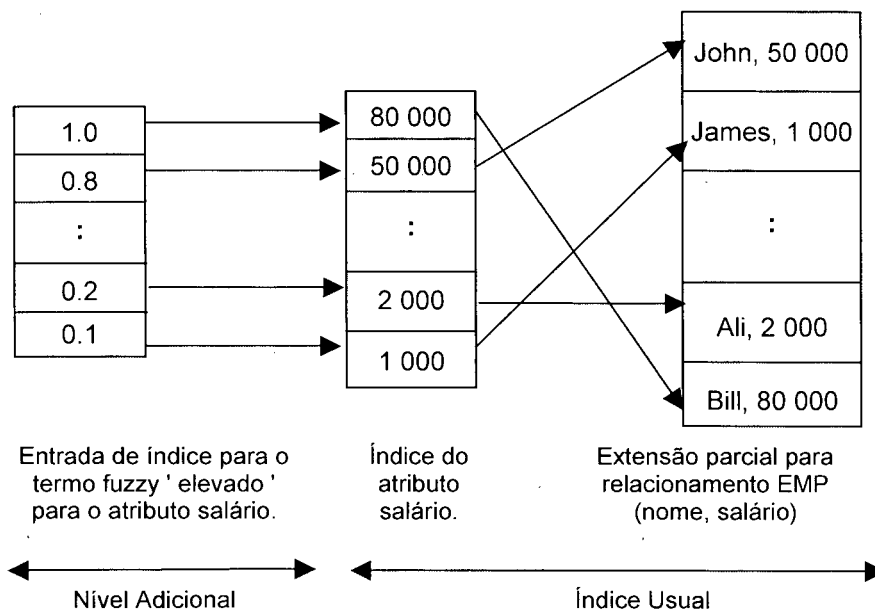


Figura 5.4 – Esquema conceitual do método de Indexação proposto por Bosc

Yazicy e George (YAZICI;GEORGE, 1999) fizeram algumas observações sobre a estrutura de índice fuzzy (Figura 5.4) proposta por Bosc et al. (BOSC;PIVERT, 1992; PETRY, 1996):

1. Esta estrutura de índice só trata com domínios homogêneos;
2. Cada entrada do índice aponta para um registro relacionado ao principal, criando assim uma grande estrutura de índice. Isto causa problemas em operações de inserção e deleção de registros;
3. Cada valor crisp pode ter um índice para cada termo fuzzy, aumentando o tamanho da estrutura de índice. Se valores fuzzy mudam freqüentemente, os custos de atualização tornam-se elevados.
4. A duplicação de endereços de blocos de dados na estrutura de indexação fuzzy causa degradação de performance na recuperação de resultados de consulta fuzzy.
5. A estrutura de indexação fuzzy não impõe qualquer ordenação dos dados no atributo fuzzy, desta forma conduzindo a um baixo desempenho em consultas fuzzy. O agrupamento dos registros é feito ao nível de índice fuzzy e ignorado no nível físico do banco de dados.
6. Esta estrutura de índice é desenvolvida para indexar relações crisp, embora seja discutida a possibilidade de ser estendido ao caso onde os atributos possuem valores com conhecimento deficiente e representam uma possibilidade de distribuição.

Embora este esquema de índice aumente um pouco a velocidade do processo de consulta fuzzy no que se refere à pesquisa seqüencial, o custo de armazenamento aumenta significativamente. Em muitos casos, a suposição de domínios fuzzy homogêneos é restritiva. Por exemplo, considera-se o atributo salário, algumas vezes podemos não discernir o salário exato de um empregado e podemos atribuir um valor fuzzy como, por exemplo, “alto”. Por outro lado, freqüentemente sabemos o salário do empregado e assim armazenamos um valor crisp como “\$20.000”. É necessário tratar dados crisp e fuzzy, como por exemplo, em um domínio heterogêneo, porém, como

esperado, ao tratar com atributos que têm um domínio heterogêneo existem complexidades adicionais quando estes atributos forem indexados.

Para o caso de atributos fuzzy com domínios homogêneos que assumem somente valores fuzzy (baixo, alto), primeiramente determina-se os predicados fuzzy que satisfazem as condições declaradas e a entrada, usando relações de semelhança em ordem para executar uma consulta fuzzy. Neste caso uma procura pelo índice é feita, e as tuplas com esses valores fuzzy pré-detectadas são devolvidas. Por outro lado, quando os atributos possuem somente dados crisp, são usadas funções de pertinência para determinar a abrangência requerida para realiza uma consulta fuzzy, e uma procura por abrangência é executada no banco de dados. Se um atributo permite dados fuzzy e crisp, isto significa que existe um domínio heterogêneo necessitando de uma aproximação que combine estes métodos de uma maneira consistente.

#### **5.6.4 MULTI LEVEL GRID FILE**

A estrutura do Multi Level Grid File (WHANG;KRISHNAMURTY, 1991) é uma estrutura de acesso multidimensional que diminui as desvantagens do Grid File. A estrutura do Grid File é melhorada através do tratamento das operações de união e divisão localmente em diretórios, sem sacrificar as capacidades de procura multi-atributo do Grid File. A estrutura de MLGF(Multi Level Grid File) é uma estrutura de acesso flexível para as operações inserção e deleção de registros, reduzindo razoavelmente o número de acessos exigidos ao disco em comparação ao Grid File. Entradas de diretório vazias não ocorrem em MLGF, sendo assim, o tamanho de diretório é mais compacto em comparação ao Grid File. Os atributos organizadores são transformados em padrões de bit, onde cada organização dos atributos é unida para formar uma chave (freqüentemente fragmentada) de padrões de bit.

No MLGF uma entrada de diretório é formada de um vetor de região e um ponteiro (WHANG;KRISHNAMURTY, 1991). O vetor da região é uma composição de padrão de bits da organização dos atributos.

A estrutura de MLGF demonstrado na figura 5.5, representa um arquivo de atributos organizados e um grid bi-dimensional (lógico), onde os blocos físicos são mostrados através de retângulos internos pontilhados (Figura 5.6). A estrutura de MLGF demonstrada é composta de dois níveis, S1 e S2, onde S2 é o diretório para o arquivo S1 e S1 é o diretório para o arquivo de dados. Existem três entradas no diretório S1 (Figura 5.5). Uma entrada (00,0) em S1 representa todos os registros cujos valores do primeiro atributo começa com 00, e os valores do segundo atributo começa com 0. Nota-se também que os blocos vazios (por exemplo (0,01), (1,1)) não são representados em qualquer lugar nesta estrutura, e cada bloco físico pode representar uma região de tamanho variado, embora só uma entrada no diretório corresponde aquele bloco. A estrutura de MLGF emprega uma representação consistente a todos os níveis de diretório. Isto armazena string de bits em pedaços, assim, resultando em um diretório compacto. O tamanho de diretório do MLGF cresce de uma maneira linear, e o domínio de uma dimensão particular é completamente representado utilizando o símbolo '- '.

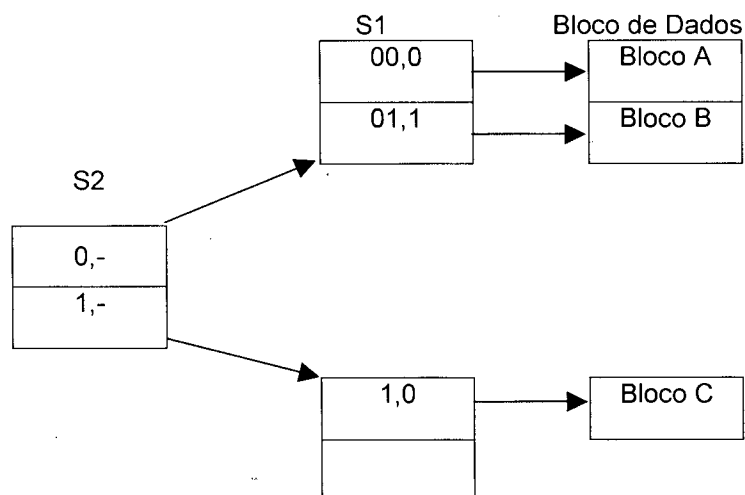


Figura 5.5 – Estrutura do diretório em MLGF

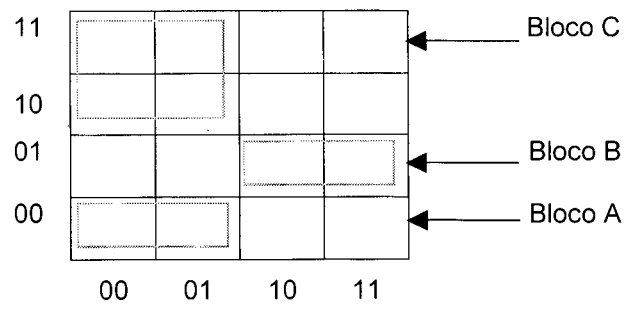


Figura 5.5 – O espaço de partição correspondente a estrutura do MLGF

## **6 FUNÇÕES FUZZY INTEGRADAS AO SISTEMA GERENCIADOR DE BANCO DE DADOS**

Existem várias propostas de linguagens para manipulação e consulta de dados nebulosos (LI;LIU,1990), (BUCLES;PETRY,1982), (PETRY;COBB,1995), (SHENOI,1990), (BERTHIER;UNTZ,1995), todas propõem a utilização de um modelo relacional fuzzy, utilizam graus de pertinência dos conjuntos para simular valores de resultados nebulosos.

Porém os atuais sistemas de banco de dados disponibilizam vários recursos para integração de funções complexas. Recursos os quais se integram a própria linguagem de consulta SQL e sem que os padrões desta sejam alterados. A linguagem SQL, que é padrão para consulta e manipulação de dados em um SGBD utiliza como base a álgebra relacional e não prever o tratamento de variáveis nebulosas. Portanto poucos modelos propostos tentam manter maior fidelidade ao padrão do SQL, no que se refere à manipulação nebulosa dos dados.

Em geral, é preferível armazenar informação fuzzy de forma lingüística e gerar as funções de relacionamento implícitas requeridas ou, alternativamente, tratar as informações fuzzy completamente em sua forma lingüística. Porém esta última torna-se muito difícil de ser aplicada em um SGBD convencional, devido a não compatibilidade dos padrões de pesquisa.

## 6.1 MÁQUINA DE INFERÊNCIA FUZZY

Uma máquina de inferência fuzzy é um sistema baseado em regras. Ela deve possuir entradas e saídas, realizando um mapeamento não-linear entre os espaços destas. Sua função-objetivo pode ser diversificada: controle, classificação, reconhecimento de padrões, filtragem, previsão, diagnóstico, modelagem, etc. Em suma, uma máquina de inferência fuzzy pode ser vista como uma ferramenta geral de tomada de decisões (BORGES, 2000).

O modelo proposto neste trabalho, para integração de um SGBD convencional e valores nebulosos, consiste na manipulação de consultas que utilizem uma máquina de inferência fuzzy integradas ao SGDB, como demonstrado na figura 6.1.



Figura 6.1 – Máquina de Inferência Fuzzy e o SGBD

## 6.2 O MODELO DE CONSULTA

Os valores de entrada crisp, armazenados em domínios de determinadas relações, são fuzificados em variáveis lingüísticas e submetidos às regras previamente definidas internamente na função de pertinência fuzzy do SGDB (figura 6.2).



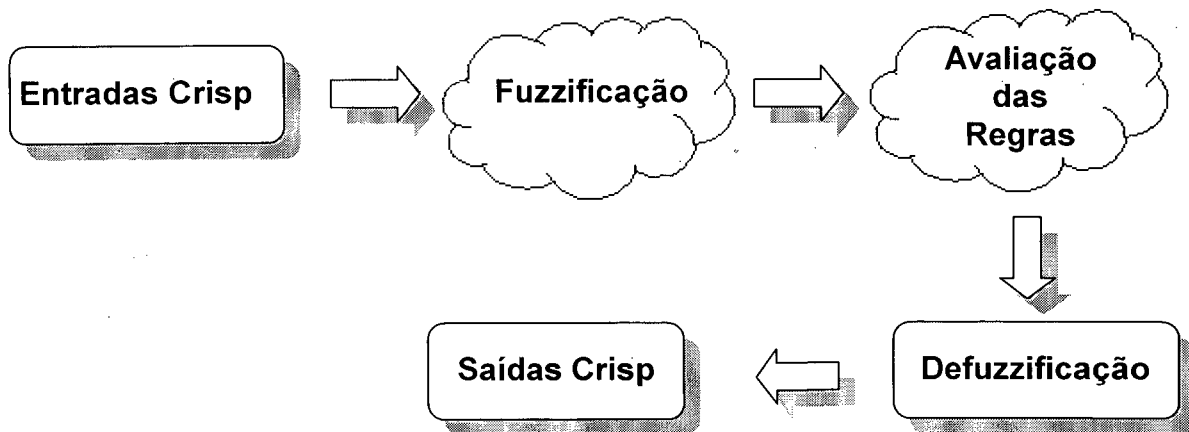


Figura 6.2 – Esquema da Consulta

Como as entradas e saídas são valores crisp, tal operação torna-se transparente para o SGDB. As funções e regras fuzzy estão encapsulada em uma função integrada ao SGDB.

### 6.3 PROBLEMA PROPOSTO

Determinada instituição financeira deseja realizar uma análise de crédito de seus clientes utilizando Lógica Fuzzy, para determinar sua confiabilidade no momento de conceder empréstimos. Esta análise deve levar em consideração vários fatores: O valor a ser emprestado, o saldo médio mensal do cliente, o capital que o cliente possui, e o montante de dívidas deste cliente.

Os dados dos clientes estão armazenados em relações dentro de um SGBD. Em especial utiliza-se a relação CLIENTE, a qual armazena dos dados relevantes para a consulta, a qual pode ser definida com conforme tabela 6.1.

CLIENTE
NOME_CLIENTE
VALOR_EMPRESTIMO
VALOR_CAPITAL
MONTANTE_DIVIDAS
SALDO_MEDIO_MENSAL

Tabela 6.1 – Relação Cliente

### 6.3.1 ESPECIFICAÇÃO DA LINGUAGEM

Na utilização da função fuzzy, pode-se manipular consultas utilizando valores nebulosos da seguinte maneira:

$$\Pi_{\text{Atr}(R)} (\sigma_{f(a,b,c) = x} R)$$

Neste caso, projeta-se um determinado atributo onde se procura o um valor no banco de dados equivalente ao retornando pela função fuzzy. O valor pela função  $f_{(a,b,c)}$ , onde  $(a,b,c)$  poderão ser atributos previamente armazenados no Banco de Dados.

### 6.3.2 SQL E FUNÇÕES FUZZY

Os SGDB atuais possuem vários recursos que podem ser utilizados para se desenvolver funções especiais. Estes recursos integram-se perfeitamente a linguagem padrão de consulta SQL. No caso da análise de crédito, a consulta seria feita da seguinte maneira:

```

SELECT    NOME_CLIENTE
FROM      CLIENTE
WHERE     CONFIABILIDADE(VALOR_EMPRESTIMO,VALOR_CAPITAL,
VALOR_DIVIDAS, SALDO_MEDIO) ='ALTA'
```

Onde CONFIABILIDADE() é a função fuzzy baseada em regras, e 'ALTA' é a variável lingüística nebulosa a qual será comparada com o valor retornado pela função.

## 6.4 CONSTRUÇÃO DA MÁQUINA DE INFERÊNCIA FUZZY

A função fuzzy é construída em C++, onde define-se o método de fuzzificação, as regras fuzzy e a defuzzificação. E finalmente a função é integrada ao SGBD.

### 6.4.1 FUZZIFICAÇÃO

O primeiro passo é o mapeamento do domínio de números reais para o domínio fuzzy. A fuzzificação representa que há atribuição de valores lingüísticos, descrições vagas ou qualitativas, definidas por funções de pertinência às variáveis de entrada.

As funções de pertinência para as variáveis nebulosas são definidas, como é demonstrado nas figuras 6.3, 6.4, 6.5 e 6.6.

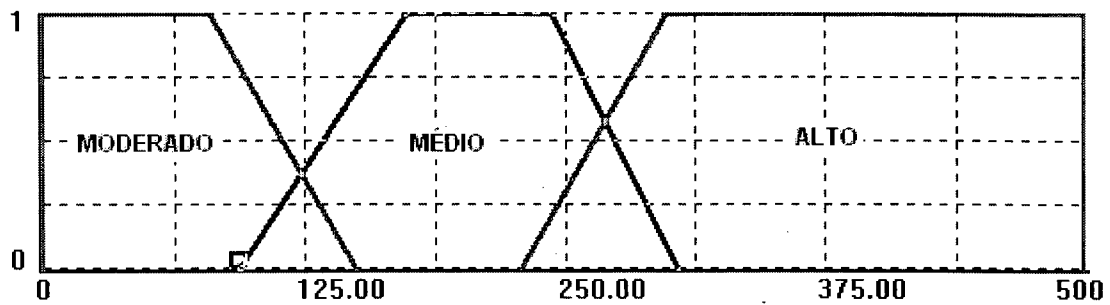


Figura 6.3 – Função de pertinência para variável Valor Empréstimo

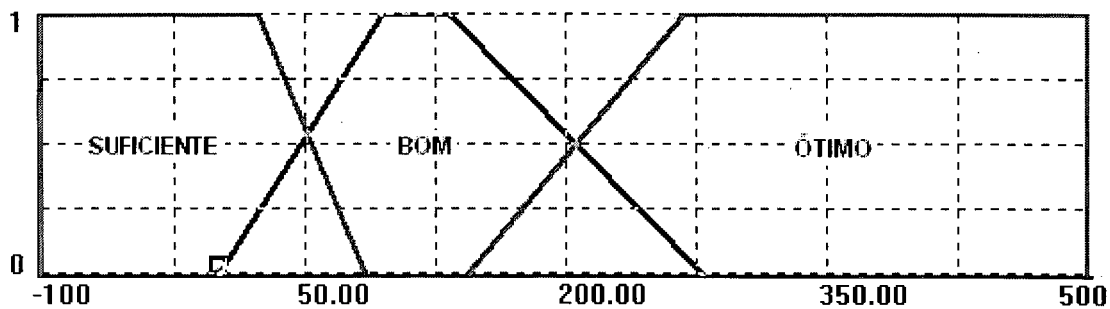


Figura 6.4 – Função de pertinência para variável Saldo Médio

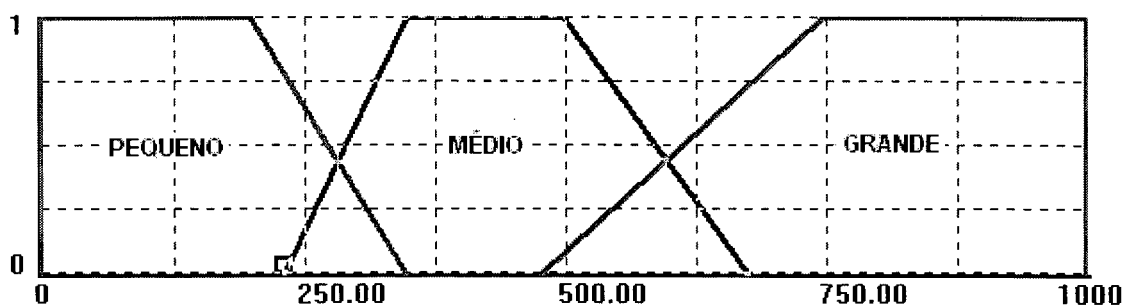


Figura 6.5 – Função de pertinência para variável Valor Patrimônio

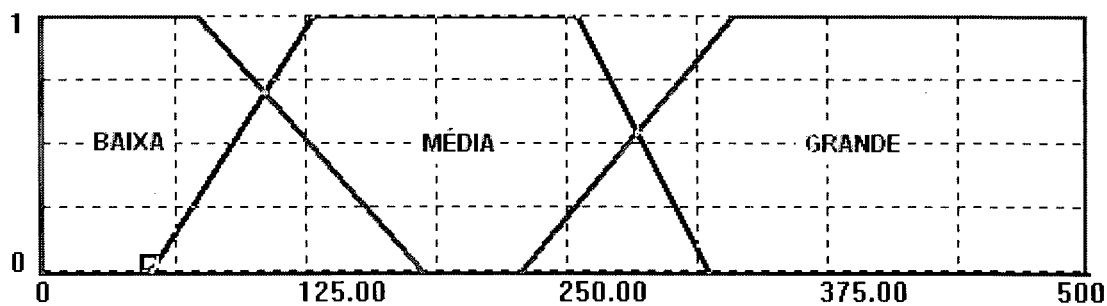


Figura 6.6 – Função de pertinência para variável Montante Dividas

#### 6.4.2 DEFINIÇÃO DAS REGRAS

As variáveis fuzzy são submetidas as regras de inferência definidas e baseadas em um conhecimento heurístico do especialista. A definição da e combinação das regras é descrita em detalhes no anexo I.

### 6.4.3 DEFUZZIFICAÇÃO

Finalmente após a avaliação das regras, os valores fuzzy são defuzzificados e comparados com os valores críps armazenados no banco de dados. O método de defuzzificação sugerido para este modelo é o *M-o-M*, recomendado para o suporte a decisões qualitativas e conforme citado no capítulo 5 item 5.3.3.

A figura 6.7 demonstra os possíveis intervalos de valores para defuzzificação.

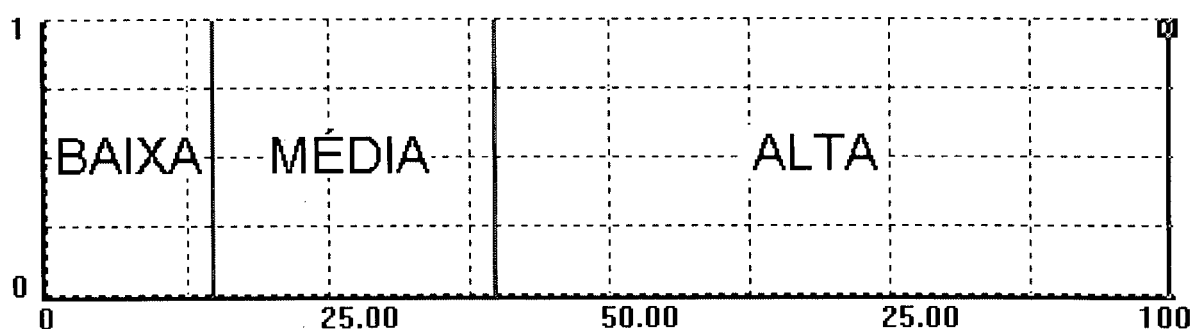


FIGURA 6.7 Intervalo de valores para defuzzificação.

### 6.5 INTEGRAÇÃO DA MÁQUINA DE INFERÊNCIA FUZZY AO SGBD

Após a função ser definida e implementada, ela deve ser integrada ao SGBD, para que possa ser utilizada pela aplicação. Um protótipo desta integração foi desenvolvido utilizando o Sistema Gerenciador de Banco de Dados Oracle<sup>1</sup>, como demonstrado na figura 6.7.

<sup>1</sup> Oracle é uma marca registrada a Oracle Corporation

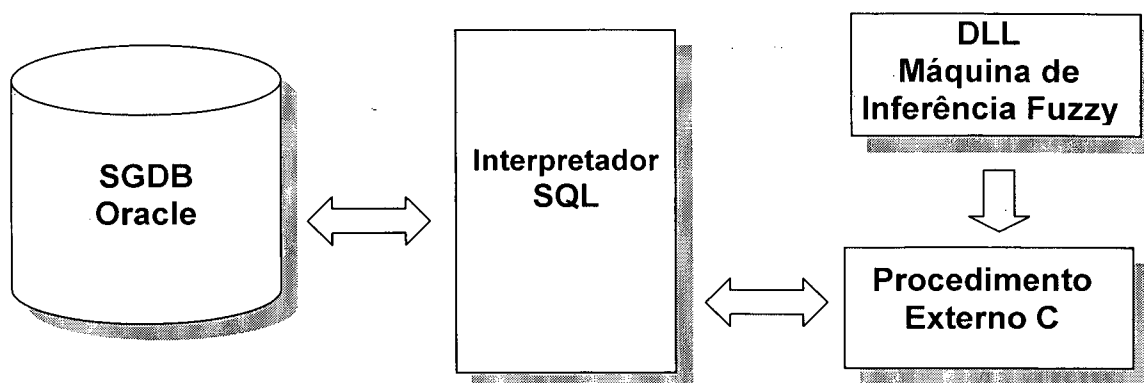


Figura 6.8 – Processo de Integração da Máquina de Inferência Fuzzy ao SGBD

Para o banco de dados um procedimento externo, é uma rotina armazenada em uma DLL (Dynamic Link Library), onde cria-se uma rotina utilizando uma determinada linguagem, a qual posteriormente é chamada para executar funções especiais.

As chamadas a rotinas externas permitem:

- Executar todos os procedimentos no servidor, portanto executando mais rapidamente e evitando o tráfego de rede;
- Integração do servidor de banco de dados com sistemas externos e fontes de dados;
- Ampliação da funcionalidade do servidor de banco de dados.

Para ser integrada ao SGBD, a máquina de inferência é transformada em uma DLL. Esta DLL contém todo o processo de fuzzificação, avaliação de regras e defuzzificação do modelo proposto. Neste caso a Máquina de Inferência Fuzzy, desenvolvida na linguagem "C++".

Com a utilização desta máquina de inferência fuzzy, o processo de consulta ao banco de dados é equivalente a uma consulta crisp convencional, além de permitir a utilização de variáveis nebulosas. A velocidade de consulta também é equivalente a uma consulta crisp normal, permitindo a utilização de índices da forma convencional. Estas facilidades eliminam umas das grandes

desvantagens das estruturas para banco de dados fuzzy, a velocidade de resposta a consultas, conforme descreve (YAZICI;GEORGE, 1999).

Como o modelo fuzzy é integrado ao banco de dados, existe a vantagem do estabelecimento de critérios definidos para os valores nas variáveis lingüísticas, eliminando assim a interpretação de cada indivíduo. Por exemplo, um saldo bancário de \$10.000 pode ser interpretado como grande para alguns e pequeno para outros.

Por ser genérico, o modelo proposto pode ser implementado em diversas situações e para diversas plataformas. A maior dificuldade provavelmente com a utilização de outras plataformas de banco de dados.

#### **6.5.1 REFERENCIANDO A BIBLIOTECA NO SGBD**

O primeiro passo é criar uma referência do protótipo fuzzy com o SGBD. Portanto para referenciar a DLL (confiab.dll) no SGDB Oracle é necessário a criação de algumas funções ou procedimentos internos.

Procedimentos e funções, são um grupo de comandos SQL e PL/SQL agrupados como uma unidade lógica usada para resolver um problema ou uma tarefa específica relacionada ao um banco de dados. Um procedimento ou função é criada e armazenada numa forma compilada no banco de dados, e pode ser executada por um usuário ou uma aplicação. As funções possuem a característica de retornarem um valor associado e intrínseco à própria função, enquanto um procedimento não retorna nenhum valor, a não ser através de seus parâmetros.

O comando para referenciar a função especial no Banco de Dados Oracle é:

```
create library confiab is 'c:\oracle\ora81\lib\confiab.dll'
```

O segundo passo é definir uma função que contenha a interface entre a biblioteca e o SGDB, definindo os parâmetros e o nome da função de chamadas.

```
CREATE OR REPLACE FUNCTION fuzzy_confiab
(crisp1_inputs REAL,
crisp2_inputs REAL,
crisp3_inputs REAL,
crisp4_inputs REAL)
RETURN REAL AS LANGUAGE C
LIBRARY confiab
NAME "confiab";
```

O terceiro passo é fazer o tratamento dos dados defuzzificados, mapeando a resposta para uma variável lingüística.



```
CREATE OR REPLACE FUNCTION confiabilidade
(crisp1_inputs REAL,crisp2_inputs REAL,crisp3_inputs REAL,crisp4_inputs
REAL)
return varchar is
fuzzy_var char(10);
begin
if fuzzy_confiab(crisp1_inputs,crisp2_inputs,crisp3_inputs,crisp4_inputs) >= 0
and fuzzy_confiab(crisp1_inputs,crisp2_inputs,crisp3_inputs,crisp4_inputs) <=
15 then
fuzzy_var := 'BAIXA';
else
if fuzzy_confiab(crisp1_inputs,crisp2_inputs,crisp3_inputs,crisp4_inputs) > 15
and fuzzy_confiab(crisp1_inputs,crisp2_inputs,crisp3_inputs,crisp4_inputs) <=
40 then
fuzzy_var := 'MEDIA';
else
fuzzy_var := 'ALTA';
end if;
end if;
return fuzzy_var;
end;
```

Desta forma a função “confiabilidade()” torna-se disponível para utilização em qualquer aplicação desenvolvida para o SGBD Oracle.

### **6.5.2 TRIGGERS**

Para facilitar a operação, existe a possibilidade de implantar triggers no banco de dados, com o intuito de manter os dados permanentemente atualizados, como por exemplo, o saldo médio em conta.

Os triggers no Oracle permitem que se escrevam procedures que são automaticamente executadas no momento de um comando de inserção, eliminação ou atualização de linhas de tabelas de um banco de dados. Os triggers em banco de dados podem ser usados para uma variedade de tarefas de gerenciamento de um banco de dados. Podem ser usados para automatizar a geração de dados nas tabelas, fazer auditoria das modificações feitas em tabelas, forçar complexas regras de integridade e implantar autorizações de segurança dos dados.

## 7 CONCLUSÃO

Pode-se observar no decorrer deste trabalho, que os sistemas gerenciadores de banco de dados convencionais não dispõem de mecanismos para tratamento de dados nebulosos. Entretanto é possível desenvolver funções específicas para manipulação destes dados nebulosos.

A Lógica Fuzzy, juntamente com a teoria dos conjuntos Fuzzy, demonstrou-se um ótimo instrumento para o tratamento de informações imprecisas. Através de máquinas de inferência fuzzy é possível tratar várias variáveis de entrada e saída com grande rapidez e eficiência, portanto uma ótima técnica para ser integrada a um Sistema Gerenciador de Banco de Dados.

Apesar de existirem várias propostas para tratamento de variáveis nebulosas, a maioria sustenta-se sobre modelos matemáticos teóricos, e sua utilização não é muito observada nas corporações. Nota-se que há pouca interação entre a área acadêmica e de pesquisa com quem utiliza a tecnologia.

A utilização de um Sistema Gerenciador de Banco de Dados Fuzzy portanto, pode-se trabalhar dados nebulosos com banco de dados convencionais utilizando-se as técnicas de fuzzificação e defuzzificação, onde os dados manipulados pelo SGDB convencional são valores comuns. Para esta tarefa, o encapsulamento das funções fuzzy demonstrou-se uma técnica muito interessante.

No protótipo desenvolvido, pode-se observar que a tarefa de integrar funções especiais ao SGDB não é uma tarefa trivial, exige um grande

conhecimento do SGBD em questão. Já o desenvolvimento da máquina de inferência fuzzy mostrou-se mais fácil, visto que existem vários procedimentos pré-definidos.

Concluiu-se que existem grandes vantagens na utilização da máquina de inferência fuzzy como uma função especial do banco de dados. Dentre elas podemos citar a velocidade de busca e processamento, e a padronização na interpretação das variáveis lingüísticas nebulosas. Este modelo pode ser implantado para solução da na maioria dos problemas que envolvem processamento fuzzy e banco de dados, principalmente em sistemas de apoio a decisão, análise de dados e consultas em linguagem natural.

Um das propostas para trabalhos futuros seria a integração de funções especiais fuzzy em Sistema Gerenciadores de Banco de Dados Orientado a Objetos. Torna-se interessante também a avaliação do modelo fuzzy proposto em consultas distribuídas, e sua utilização em sistemas de Data Mining, SAD (Sistemas de Apoio a Decisão), SIG (Sistemas de Informações gerenciais) e SE (Sistemas Especialistas).

## 8 REFERÊNCIAS BIBLIOGRÁFICAS

AZIZ , Shahriz Abdul.; PARTHIBAN, "Jeyakody. *Fuzzy Logic*". Disponível em:<[http://www-dse.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/sbaa/report.intro.html](http://www-dse.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sbaa/report.intro.html)>. Acesso em: 20 de Março de 2000.

BARKAKATI, Nabajyoti.;HIPSON Peter D. "*Visual C++ Guia de Desenvolvimento Avançado*". São Paulo: Berkeley Brasil Editora, 1994.

BECKMANN, N.; KRIEGEL, H.; SCHNEIDER, R.; SEEGER, B. "*The R-Tree: An Efficient and Robust Method for Points and Rectangles*". In: Proceedings of the 1990 ACM.

BELCHIOR, A.D. "*Um Modelo Fuzzy para Avaliação da Qualidade de Software*". D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1997.

BENTLEY, J.L.; "*Multidimensional Binary Search Trees Used for Associative Searching*". CACM, 509-517, 1975.

\_\_\_\_\_; "*Multidimensional Binary Search Trees in Database Applications*". IEEE Trans. on Software Engineering, V.SE-5, No. 4, 333-340, 1979.

BENTLEY, J.L., MAURER H.A., "*Efficient Worst-Case Data Structures for Range Searching*". Acta Informatica 13, 155-168, 1980.

BERNARDES, MC. ; D'OTTAVIANO, IML. "*Fuzzy sets on drawing fair plane curves*". In: Anais do VIII SIBGRAPI. I. pp.87-94, 1995.

BERTHIER, ANR.; MUNTZ, R. "*F-G: A Fuzzy Algebra for Approximate in Databases*". In: X Brazilian Symposium on Databases. pp.365-376. Recife, Brazil, 1995.

BORGES Paulo Sergio S. "*Fuzzy Logic*". Material distribuído no curso de Mestrado em Ciência da Computação na disciplina Introdução a Inteligência Computacional. Unoesc/UFSC, 2000.

BORDOGNA, G.; LUCARELLA D.; PASI G., "A Fuzzy Object-Oriented Data Model". Proceedings of the IEEE 3rd International Conference on Fuzzy Systems, 313-318, 1994.

BOSC, P.; Pivert O., "Fuzzy Querying in Conventional Databases". Fuzzy Logic for Management of Uncertainty, Edited by L.A.Zadeh and J. Kacprzyk, John Wiley and Sons Inc, 1992.

\_\_\_\_\_; "Extending SQL Retrieval Features for the Handling of Flexible Queries". In: Dubois D, Prade H, and Yager. Fuzzy Information Engineering: A Guided Tour of Applications. 15. New York. Wiley Computer Publishing, John Wiley & Sons, INC. 1997;

BOSC, P.; M. GALIBOURG; H. HAMON, "Fuzzy Querying With SQL: Extensions and Implementation Aspects". Fuzzy Sets & Systems, 28, 333-349, (1988).

BRAGA, André Luiz. "Ferramentas de Manipulação Nebulosa de Dados com Aplicação em Sistemas de Informação Geográfica". COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1998.

BUCKLES, BP.; PETRY, FE. "A fuzzy representation of data for relational databases". Fuzzy Sets and Systems 7:213-26. 1982.

BURROUGH, PA. "Fuzzy mathematical methods for soil survey and land evaluation". Journal of Soil Science 40:477-92. 1989.

BURROUGH, PA; MACMILLAN, RA; VAN Deursen W. "Fuzzy classification methods for determining land suitability from soil profile observations and topography". Journal of Soil Science (43):477-92. 1992.

CERICOLA, Osvaldo Vicente. "Banco de Dados Relacional e Distribuído". Rio de Janeiro: Livros Técnicos e Científicos, 1991.

CHANG, L.; BURROUGH, PA. "Fuzzy reasoning: A new quantitative aid for land evaluation". Soil Survey and Land Evaluation. 7(2). 69-80. 1987.

CODD, EF. "A Relational Model of Data for Large Shared Data Banks". Communications of ACM 6:377-87. 1970.

COX, E. "The Fuzzy Systems Handbook". Boston: AP Professional; 1994.

\_\_\_\_\_. - "Fuzzy Logic for Business and Industry". 1994.

\_\_\_\_\_. - "Fuzzy Fundamentals" IEEE Spectrum". 18 1992.

DIJKMEIJER, J.; HOOP, S. "Topologic Relations Between Fuzzy Areas Objects". In: Proceedings of the 7th International Symposium on Spatial Data Handling. pp.377-393. 1996.

DUBOIS, D.; PRADE, H. "Mesuring properties of fuzzy sets: A general technique and its use in fuzzy query evaluation". Fuzzy Sets and Systems 38:137-52. 1990.

DUTTA, S. "Qualitative Spatial Reasoning: A Semi-quantitative Approach Using Fuzzy Logic". In: Buchmann A, Günther O, Smith TR, and Wang F. Design and Implementation of Large Spatial Databases. Springer-Verlag. 1989.

\_\_\_\_\_ - "Approximate Spatial Reasoning: Integrating Qualitative and Quantitative Constraints". International Journal of Approximate Reasoning 5:307-30. 1991.

EBERHART, Russ; SIMPSON, Pat.; DOBBINS, Roy. "Computational Intelligence PC Tools". AP Professional, United Kingdom, 1996.

FISHER, PF.; PATHIRANA, D. "Evaluation of fuzzy membership of land cover classes in suburban areas of North-East Ohio". In: ASPRS-ACSM Fall Convention. pp.125-132. 1989.

FREESTON, M., "The BANG File: A New Kind of Grid File". ACM, 260-269, 1987.

KHOSHAFIAN, Setrag. "Banco de Dados Orientado a Objetos". São Paulo: IBPI Press, 1994.

KLIR, George J.; YUAN, Bo. "Fuzzy Sets and Fuzzy Logic: Theory and Applications". Upper Saddle River, New Jersey: Prentice Hall; 1995.

\_\_\_\_\_. "Fuzzy databases and information retrieval systems". 1 ed. New Jersey: Prentice-Hall; 1995.

LACROIX, M.; LAVENCY, P. "Preferences: Putting more knowledge into queries". In: Proceedings of the 13 VLDB Conference. Brighton. 1987.

LEUNG, Y. "Towards a flexible framework for recognition". Environment and Planning. A(16). 203-215. 1984.

LI, D.; LIU, D. "A Fuzzy PROLOG Database System". Lauton, Somerset, England: Foulk PWD, editor. Research Studies Press Ltd.; 1990.

MARTIN, James.; ODELL, J. James. "Análise e Projeto Orientados a Objetos." São Paulo: Makron Books, 1995.

MORAIS, Rinaldo de Oliveira, "Oracle 7 Server: Conceitos Básicos". São Paulo: Érica, 1995.

- MÜLLER, Daniel Nehme. "Notas Sobre Lógica Fuzzy". Disponível em: <<http://www.ulbra.tche.br/~danielnm/ia/fuzzy/fuzzy.html>>. Acesso em: 18 de Março de 2000.
- NIEVERGELT, J.; HINTERBERGER H.; SEVCIK K.C., "The Grid File: An Adaptable, Symmetric Multikey File Structure". ACM Transactions on Database Systems, 9(1), 38-71, 1984.
- NOLA, AD.; PEDRYCZ, W.; SESSA, S. "Fuzzy relational structures: The state-of-art". Fuzzy Sets and Systems 75:241-62. 1995.
- PETRY, F.; COBB, M. "Fuzzy Querying Binary Relationships in Spatial Databases". In:IEEE International Conference on Cybernetics and Society. pp.378-385. 1995.
- PETRY, F. "Fuzzy Databases". Boston/Dordrecht/London: Zimmermann HC, editora Kluwer Academic Publishers; 1996.
- RAO, Bindu R. "Object-Oriented Databases Technology, Applications, and Products". s.l. McGraw-Hill, s.d., 1994.
- RAO, Valluru; RAO, Hayagriva. "C++ Neural Networks and Fuzzy Logic". Second Edition. New York: MIS:Press, 1995.
- ROSS, TJ. "Fuzzy Logic with Engineering Applications". New York: Cox L and Morriss JM, editors.McGraw-Hill, Inc.; 1995.
- SAATY, RW. "The Analytic Hierarchy Process - What is it and how it is used." Mathematical Modeling. 9 161-176. 1987.
- SALZBERG, B., "File Structures: An Analytical Approach". Prentice-Hall International Editions, 1988.
- SANT'ANNA, Vanderley. "Banco de Dados Relacional Difuso". Disponível em: <[http://www.eps.ufsc.br/~martins/fuzzy/fuz\\_ap/dtb1/dtb1.htm](http://www.eps.ufsc.br/~martins/fuzzy/fuz_ap/dtb1/dtb1.htm)>. Acesso em: 25 de Março de 2000.
- SHAW, Ian S.; SIMÕES, Marcelo Godoy. "Controle e Modelagem Fuzzy". São Paulo: Edgard Blücher Ltda, 1999.
- SHENOI, S.; MELTON, A.; FAN, LT. "An equivalence classes model of fuzzy relational databases". Fuzzy Sets and Systems 38:153-70. 1990.
- SIGMOD "International Conference on Management of Data". pp.322-331. Atlantic City,NJ,1990.
- SOUSA, Carlos Pimentel de. "Lógica Fuzzy Aplicada". <http://www.dee.ufc.br/~pimentel/fuzzy/fuzzy.html>, 17 de Março de 2000.

- SOUZA, João A. de.; DANDOLINI, Gertrudes Aparecida. "Construção de Conjuntos Difusos". Disponível em: <[http://www.eps.ufsc.br/~martins/fuzzy/fuz\\_ap/cont\\_di/cont\\_di.htm](http://www.eps.ufsc.br/~martins/fuzzy/fuz_ap/cont_di/cont_di.htm)>. Acesso em: 23 de Março de 2000.
- TAHANI, V. "A fuzzy model of document retrieval systems". In: Inf.Proc. & Manag. 12. pp.177-187. 1976.
- \_\_\_\_\_, "A conceptual framework for fuzzy query processing: A step toward very intelligent database systems". In: Inf.Proc. & Manag. 13. pp.289-303. 1977.
- WHANG, K.Y.; KRISHNAMURTY R., "The Multilevel Grid File- A Dynamic Hierarchical Multidimensional File Structure". Database Systems for Advanced Applications, 449-456, 1991.
- WILCOX, Jobathan. "Object Methods in Distributed Computing". Dr. Dobb's Journal, November 1994.
- WINBLAD, L. Ann.; EDWARDS, D. Samuel. "Software Orientado a Objeto". São Paulo: Makron Books, 1993.
- ZADEH LA. "Fuzzy Sets". Information and Control 8(8):338-53. 1965.
- \_\_\_\_\_, "Fuzzy Logic". IEEE Computer 1988.
- ZADEH, L.A. and KACPRZYK J., "Fuzzy Logic for Management of Uncertainty". Eds. John Wiley and Sons Inc., 607-644, New York, 1992.
- YAZICI, Adam.;GEORGE, Roy. "Fuzzy Database Modeling". New York: Physica Verlag, 1999.
- YAZICI, Adam.; KOYUNCU M., "Fuzzy Object-Oriented Database Modeling Coupled with Fuzzy Logic". Fuzzy Sets and Systems 89, 1-26, 1997.
- YAZICI, A.; BUCKLES B.P.;PETRY F.E., "A Semantic Data Model Approach to Knowledge-Intensive Applications". International Journal of Expert Systems: Research and Applications, Vol.8 (1), 77-91, 1995.



## **ANEXO I – REGRAS DE INFERÊNCIA FUZZY**

Regra 1

SE Emprestimo = ALTO  
E SaldoMedio = OTIMO  
E Capital = GRANDE  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 2

SE Emprestimo = ALTO  
E SaldoMedio = OTIMO  
E Capital = GRANDE  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 3

SE Emprestimo = ALTO  
E SaldoMedio = OTIMO  
E Capital = GRANDE  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 4

SE Emprestimo = MODERADO  
E SaldoMedio = SUFICIENTE  
E Capital = PEQUENO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = BAIXA

Regra 5

SE Emprestimo = ALTO  
E SaldoMedio = OTIMO  
E Capital = MEDIO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = MEDIA

Regra 6

SE Emprestimo = ALTO  
E SaldoMedio = OTIMO  
E Capital = MEDIO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 7

SE Emprestimo = ALTO  
E SaldoMedio = OTIMO  
E Capital = MEDIO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 8

SE Emprestimo = ALTO  
E SaldoMedio = OTIMO  
E Capital = PEQUENO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = BAIXA

Regra 9

SE Emprestimo = ALTO  
E SaldoMedio = OTIMO

E Capital = PEQUENO

E Dividas = BAIXA

ENTÃO CONFIABILIDADE = BAIXA

Regra 10

SE Emprestimo = ALTO  
E SaldoMedio = BOM  
E Capital = GRANDE  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = MEDIA

Regra 11

SE Emprestimo = ALTO  
E SaldoMedio = BOM  
E Capital = GRANDE  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 12

SE Emprestimo = ALTO  
E SaldoMedio = BOM  
E Capital = GRANDE  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 13

SE Emprestimo = ALTO  
E SaldoMedio = BOM  
E Capital = MEDIO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 14

SE Emprestimo = ALTO  
E SaldoMedio = BOM  
E Capital = MEDIO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 15

SE Emprestimo = ALTO  
E SaldoMedio = BOM  
E Capital = MEDIO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 16

SE Emprestimo = ALTO  
E SaldoMedio = BOM  
E Capital = PEQUENO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 17

SE Emprestimo = ALTO  
E SaldoMedio = BOM  
E Capital = PEQUENO  
E Dividas = MEDIA

ENTÃO CONFIABILIDADE = BAIXA

Regra 18

SE Emprestimo = ALTO

E SaldoMedio = BOM

E Capital = PEQUENO

E Dividas = BAIXA

ENTÃO CONFIABILIDADE = BAIXA

Regra 19

SE Emprestimo = ALTO

E SaldoMedio = SUFICIENTE

E Capital = GRANDE

E Dividas = GRANDE

ENTÃO CONFIABILIDADE = BAIXA

Regra 20

SE Emprestimo = ALTO

E SaldoMedio = SUFICIENTE

E Capital = GRANDE

E Dividas = MEDIA

ENTÃO CONFIABILIDADE = MEDIA

Regra 21

SE Emprestimo = ALTO

E SaldoMedio = SUFICIENTE

E Capital = GRANDE

E Dividas = BAIXA

ENTÃO CONFIABILIDADE = ALTA

Regra 22

SE Emprestimo = ALTO

E SaldoMedio = SUFICIENTE

E Capital = MEDIO

E Dividas = GRANDE

ENTÃO CONFIABILIDADE = BAIXA

Regra 23

SE Emprestimo = ALTO

E SaldoMedio = OTIMO

E Capital = PEQUENO

E Dividas = GRANDE

ENTÃO CONFIABILIDADE = BAIXA

Regra 24

SE Emprestimo = ALTO

E SaldoMedio = SUFICIENTE

E Capital = MEDIO

E Dividas = BAIXA

ENTÃO CONFIABILIDADE = BAIXA

Regra 25

SE Emprestimo = ALTO

E SaldoMedio = SUFICIENTE

E Capital = PEQUENO

E Dividas = GRANDE

ENTÃO CONFIABILIDADE = BAIXA

Regra 26

SE Emprestimo = ALTO

E SaldoMedio = SUFICIENTE

E Capital = PEQUENO

E Dividas = MEDIA

ENTÃO CONFIABILIDADE = BAIXA

Regra 27

SE Emprestimo = ALTO

E SaldoMedio = SUFICIENTE

E Capital = PEQUENO

E Dividas = BAIXA

ENTÃO CONFIABILIDADE = BAIXA

Regra 28

SE Emprestimo = MEDIO

E SaldoMedio = OTIMO

E Capital = GRANDE

E Dividas = GRANDE

ENTÃO CONFIABILIDADE = BAIXA

Regra 29

SE Emprestimo = MEDIO

E SaldoMedio = OTIMO

E Capital = GRANDE

E Dividas = MEDIA

ENTÃO CONFIABILIDADE = MEDIA

Regra 30

SE Emprestimo = MEDIO

E SaldoMedio = OTIMO

E Capital = GRANDE

E Dividas = BAIXA

ENTÃO CONFIABILIDADE = ALTA

Regra 31

SE Emprestimo = MEDIO

E SaldoMedio = OTIMO

E Capital = MEDIO

E Dividas = GRANDE

ENTÃO CONFIABILIDADE = MEDIA

Regra 32

SE Emprestimo = MEDIO

E SaldoMedio = OTIMO

E Capital = MEDIO

E Dividas = BAIXA

ENTÃO CONFIABILIDADE = MEDIA

Regra 33

SE Emprestimo = ALTO

E SaldoMedio = SUFICIENTE

E Capital = MEDIO

E Dividas = BAIXA

ENTÃO CONFIABILIDADE = BAIXA

Regra 34

SE Emprestimo = MEDIO

E SaldoMedio = OTIMO  
E Capital = PEQUENO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 35  
SE Emprestimo = MEDIO  
E SaldoMedio = OTIMO  
E Capital = PEQUENO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = BAIXA

Regra 36  
SE Emprestimo = MEDIO  
E SaldoMedio = OTIMO  
E Capital = PEQUENO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 37  
SE Emprestimo = MEDIO  
E SaldoMedio = BOM  
E Capital = GRANDE  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = ALTA

Regra 38  
SE Emprestimo = MEDIO  
E SaldoMedio = BOM  
E Capital = GRANDE  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = ALTA

Regra 39  
SE Emprestimo = MEDIO  
E SaldoMedio = BOM  
E Capital = GRANDE  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 40  
SE Emprestimo = MEDIO  
E SaldoMedio = BOM  
E Capital = MEDIO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 41  
SE Emprestimo = MEDIO  
E SaldoMedio = BOM  
E Capital = MEDIO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 42  
SE Emprestimo = MEDIO  
E SaldoMedio = OTIMO  
E Capital = MEDIO

E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 43  
SE Emprestimo = MEDIO  
E SaldoMedio = BOM  
E Capital = PEQUENO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 44  
SE Emprestimo = MEDIO  
E SaldoMedio = BOM  
E Capital = PEQUENO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = BAIXA

Regra 45  
SE Emprestimo = MEDIO  
E SaldoMedio = BOM  
E Capital = PEQUENO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = BAIXA

Regra 46  
SE Emprestimo = MEDIO  
E SaldoMedio = SUFICIENTE  
E Capital = GRANDE  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = MEDIA

Regra 47  
SE Emprestimo = MEDIO  
E SaldoMedio = SUFICIENTE  
E Capital = GRANDE  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 48  
SE Emprestimo = MEDIO  
E SaldoMedio = SUFICIENTE  
E Capital = GRANDE  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 49  
SE Emprestimo = MEDIO  
E SaldoMedio = SUFICIENTE  
E Capital = MEDIO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 50  
SE Emprestimo = MEDIO  
E SaldoMedio = SUFICIENTE  
E Capital = MEDIO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = BAIXA

Regra 51  
SE Empréstimo = MEDIO  
E SaldoMedio = SUFICIENTE  
E Capital = MEDIO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 52  
SE Empréstimo = MEDIO  
E SaldoMedio = SUFICIENTE  
E Capital = PEQUENO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 53  
SE Empréstimo = MEDIO  
E SaldoMedio = SUFICIENTE  
E Capital = PEQUENO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = BAIXA

Regra 54  
SE Empréstimo = MEDIO  
E SaldoMedio = SUFICIENTE  
E Capital = PEQUENO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = BAIXA

Regra 55  
SE Empréstimo = MODERADO  
E SaldoMedio = OTIMO  
E Capital = GRANDE  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = MEDIA

Regra 56  
SE Empréstimo = MODERADO  
E SaldoMedio = OTIMO  
E Capital = GRANDE  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = ALTA

Regra 57  
SE Empréstimo = MODERADO  
E SaldoMedio = OTIMO  
E Capital = GRANDE  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 58  
SE Empréstimo = MODERADO  
E SaldoMedio = OTIMO  
E Capital = MEDIO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = MEDIA

Regra 59

SE Empréstimo = MODERADO  
E SaldoMedio = OTIMO  
E Capital = MEDIO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 60  
SE Empréstimo = MODERADO  
E SaldoMedio = OTIMO  
E Capital = MEDIO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 61  
SE Empréstimo = MODERADO  
E SaldoMedio = OTIMO  
E Capital = PEQUENO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = MEDIA

Regra 62  
SE Empréstimo = MODERADO  
E SaldoMedio = OTIMO  
E Capital = PEQUENO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 63  
SE Empréstimo = MEDIO  
E SaldoMedio = BOM  
E Capital = MEDIO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 64  
SE Empréstimo = MODERADO  
E SaldoMedio = BOM  
E Capital = GRANDE  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = MEDIA

Regra 65  
SE Empréstimo = MODERADO  
E SaldoMedio = BOM  
E Capital = GRANDE  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = ALTA

Regra 66  
SE Empréstimo = MODERADO  
E SaldoMedio = BOM  
E Capital = GRANDE  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = ALTA

Regra 67  
SE Empréstimo = MODERADO  
E SaldoMedio = BOM

E Capital = MEDIO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 68  
SE Emprestimo = MODERADO  
E SaldoMedio = BOM  
E Capital = MEDIO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 69  
SE Emprestimo = MODERADO  
E SaldoMedio = BOM  
E Capital = MEDIO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 70  
SE Emprestimo = MODERADO  
E SaldoMedio = BOM  
E Capital = PEQUENO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 71  
SE Emprestimo = MODERADO  
E SaldoMedio = BOM  
E Capital = PEQUENO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = BAIXA

Regra 72  
SE Emprestimo = MODERADO  
E SaldoMedio = BOM  
E Capital = PEQUENO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 73  
SE Emprestimo = MODERADO  
E SaldoMedio = SUFICIENTE  
E Capital = GRANDE  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = MEDIA

Regra 74  
SE Emprestimo = MODERADO  
E SaldoMedio = SUFICIENTE  
E Capital = GRANDE  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 75  
SE Emprestimo = MODERADO  
E SaldoMedio = SUFICIENTE  
E Capital = GRANDE  
E Dividas = BAIXA

ENTÃO CONFIABILIDADE = ALTA

Regra 76  
SE Emprestimo = MODERADO  
E SaldoMedio = SUFICIENTE  
E Capital = MEDIO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 77  
SE Emprestimo = MODERADO  
E SaldoMedio = OTIMO  
E Capital = PEQUENO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 78  
SE Emprestimo = MODERADO  
E SaldoMedio = SUFICIENTE  
E Capital = MEDIO  
E Dividas = BAIXA  
ENTÃO CONFIABILIDADE = MEDIA

Regra 79  
SE Emprestimo = MODERADO  
E SaldoMedio = SUFICIENTE  
E Capital = PEQUENO  
E Dividas = GRANDE  
ENTÃO CONFIABILIDADE = BAIXA

Regra 80  
SE Emprestimo = MODERADO  
E SaldoMedio = SUFICIENTE  
E Capital = PEQUENO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = BAIXA

Regra 81  
SE Emprestimo = MODERADO  
E SaldoMedio = SUFICIENTE  
E Capital = MEDIO  
E Dividas = MEDIA  
ENTÃO CONFIABILIDADE = MEDIA