

# SISTEMA INTELIGENTE DE APOIO A DECISÃO PARA A GESTÃO DA MANUTENÇÃO

**Cícero R. Cavati**

UFES/CT/DEL, Campus Universitário-Goiabeiras, Vitória ES, 29060-970 cavati@ele.ufes.br

**Evaldo Moura Junior**

UFES/CT/DEL, Campus Universitário-Goiabeiras, Vitória ES, 29060-970

**Resumo.** *Os administradores de empresas buscam utilizar programas cada vez mais eficientes na tentativa de apoiar as suas decisões e alcançar melhores resultados em seus processos. É natural que se continue buscando diariamente ferramentas cada vez mais avançadas para agregar valor ao processo e sobreviver à competitividade. Assim, apresenta-se aqui a descrição das etapas de desenvolvimento de um novo software denominada de Sistema Inteligente de Apoio à tomada de Decisão - SIAD para fins de diagnóstico. O SIAD foi implementado em VISUAL BASIC VI, usa o MS ACCESS como banco de dados e utiliza a lógica difusa para a geração dos diagnósticos. Dentre as varias vantagens que o software apresenta, destacam-se as possibilitas de gerar, armazenar e editar as características de vários equipamentos e conseqüentemente seus diagnósticos. Portanto, esta ferramenta computacional representa uma oportunidade de se apoiar a realização da gestão de manutenção quanto a avaliação de riscos de parada em equipamentos e em processos industriais, como forma de obtenção de benefícios, maximização de desempenho e minimização de perdas.*

**Palavras-chave:** *Manutenção, Software, Gestão, Sistema Inteligente.*

## 1 INTRODUÇÃO

Tem-se observado forte desenvolvimento de técnicas aplicadas ao desenvolvimento de produtos bem como de serviços na área de sistemas de energia elétrica. Em relação à manutenção de equipamentos, as técnicas tradicionais usadas são as chamadas manutenções corretiva e preventiva.

Resumidamente, podemos diferenciar estas duas formas de realizar a manutenção da seguinte maneira: a manutenção corretiva é uma atuação efetuada após a ocorrência de uma falha, destinada a colocar um item no estado no qual poderá operar de forma correta, enquanto, a preventiva é uma manutenção efetuada em intervalos pré-determinados conforme critérios prescritos, destinada a reduzir a probabilidade de falhas ou a degradação do desempenho de um item.

Outra forma de abordar o problema de manutenção é por meio da manutenção preditiva, realizando diagnósticos constantes que permitem reajustar previsões a se efetuar das operações de manutenção, estimando-se a tendência evolutiva não adequada do funcionamento do equipamento e previsão do tempo durante o qual é possível continuar a utilizá-lo antes da possível avaria. Portanto, para esse tipo de manutenção, que segundo a norma da ABNT constitui-se numa designação em particular de manutenção preventiva, é necessária a monitoração dos estados de um equipamento.

O SIAD – Sistema Inteligente de Apoio à Decisão fornece condições para os estados de um equipamento em função dos parâmetros coletados quando da sua inspeção ou quando da aquisição de dados proveniente de um banco de dados ou ainda diretamente de um sistema de monitoração on-line.

Uma análise custo/benefício dessa monitoração, determina o possível uso desse tipo de estratégia na manutenção. Entre as vantagens de se fazer o diagnóstico de equipamentos podemos citar: aumento da confiabilidade e credibilidade dos processos, redução dos custos de eventuais reparos e garantia de continuidade de fornecimento. Segundo Cavati<sup>(1,2,3)</sup> uma das ferramentas utilizada para realizar a manutenção preditiva é o SIAD – Sistema Inteligente de Apoio a Decisão, que é apoiado em lógica difusa.

Desenvolvida nos anos 60 e utilizada nas indústrias a partir dos anos 70, os sistemas que utilizam a lógica difusa conforme Mandani<sup>(4)</sup>, Zimmermann<sup>(5)</sup> e Ross<sup>(6)</sup>, mostraram-se poderosos

aliados no que tange a facilidade de configuração em relação aos métodos tradicionais de controle e com resultados expressivos tratando os problemas sob um novo paradigma.

A tradicional lógica booleana trata o mundo real como tendo apenas duas classes (verdadeiro ou falso). A lógica difusa atribui às variáveis reais (temperatura, pressão, tensão, etc) classes de conjuntos associados a termos lingüísticos (alto, baixo, quente, frio, etc) denominados pela literatura de ‘membership’. Com isso, a determinação das regras de controle, de um dado sistema, é feita, segundo Mendel<sup>(7)</sup> e Tomsovic<sup>(8)</sup>, por meio de um conjunto de regras definido pela união dessas classes, que formam a representação de todos os estados das grandezas utilizadas no sistema.

Assim “fuzzy” ganha a grande capacidade de trabalhar os problemas, utilizando uma linguagem natural, não necessitando utilizar o equacionamento matemático de complexos sistemas. Isso permite dimensionar e modelar problemas utilizando o conhecimento heurístico e o bom senso do projetista. Por essa facilidade “fuzzy” vem sendo difundido na economia, medicina, engenharia ou qualquer outro meio científico que necessite de um sistema capaz de trabalhar de forma robusta com as incertezas.

Foram desenvolvidos vários softwares capazes de desenvolver tal lógica. Dentre eles, dois se sobressaem: a Toolbox Fuzzy do Matlab® e o aplicativo FuzzyTech. Apesar de ambos serem poderosas ferramentas de desenvolvimento, estas pecam quanto à facilidade de interação com o do usuário de forma não transparente. Abaixo estão listadas algumas características que são apontadas pelos usuários como principais dificuldades desses softwares:

- ✓ Possuem pacotes fechados quanto ao número de entradas, campos, regras e saídas.
- ✓ Muitas telas simultâneas para a entrada e edição de regras, dificultando a utilização pelo operador.
- ✓ Não gera banco de dados relacionados com as entradas executadas pelo usuário.

Neste trabalho, apresenta-se um novo software, com características genéricas, capaz de desenvolver a lógica difusa para fins de atender as necessidades de realização de manutenção preditiva de forma atrativa e transparente ao usuário.

## 2 CONCEITOS DE CONJUNTOS DIFUSOS

Nesta seção não se pretende esgotar o assunto de conjuntos nebulosos ou difusos. Aqui, o intuito é apenas resumir algumas das suas principais características e implicações que foram utilizados no desenvolvimento do software.

Na literatura técnica, muitos modelos matemáticos tentam tratar dados, informações e conhecimento de uma mesma forma, constituindo-se geralmente, por um lado, numa forma muito simples, porém imprecisa ou inadequada, ou por outro lado, numa forma complexa e não prática para uso generalizado em casos reais. Assim, podemos dizer que os modelos matemáticos convencionais são: *Crisp*, isto é, ambivalentes (sim ou não, falso ou verdadeiro, maior ou menor, etc.); Determinísticos; Precisos por hipótese e Não-ambíguos. Entretanto, algumas situações no mundo real apresentam:

- Incertezas (Ex.: carga, afluências, falhas em equipamentos, etc.);
- Informações vagas (Ex.: tensão alta ou baixa, fluxo próximo ao limite, etc.);
- Ambigüidade (Ex.: reservatório cheio).

As características acima podem ser tratadas por vários métodos, alguns dos quais são resumidos a seguir:

- Modelos Probabilísticos ou Estocásticos: quando existem informações passadas que podem ser modeladas por métodos freqüenciais;
- Análise de Intervalos: avaliação de como dados representados por intervalos se propagam em operações aritméticas, cálculos diferenciais e integrais, etc;
- Teoria dos Erros: avaliação de como erros são propagados ao longo de um processo experimental;

- Modelos difusos (ou modelos baseados em Lógica Difusa): quando a incerteza deriva da imprecisão ou ambigüidade da informação ou conhecimento existente sobre o problema.

Quanto aos modelos difusos, estes são geralmente utilizados em:

- Métodos de representação de conhecimento em linguagem natural;
- Modelagem de incertezas para as quais não são disponíveis dados estatísticos;
- Modelagem de conhecimento subjetivo;
- Medida da qualidade de conhecimento subjetivo;
- Integração de métodos lógicos e numéricos;
- Modelagem de restrições e objetivos não-rígidos (*soft constraints*).

## 2.1 Funções de Pertinência

Os conjuntos convencionais são definidos pela enumeração de seus elementos ou por uma condição que defina se o elemento pertence ou não ao conjunto. Por outro lado, os conjuntos difusos possuem a seguinte representação gráfica, conforme Figura 1.

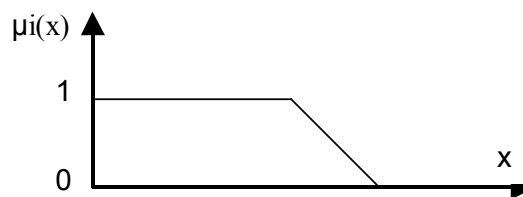


Figura 1. Representação gráfica da função de pertinência

Os conjuntos difusos podem ser vistos como uma generalização da noção de conjunto na qual a função de pertinência pode assumir valores no intervalo  $[0,1]$ . Neste caso não podemos dizer simplesmente que um elemento pertence ao conjunto e sim que o elemento pertence ao conjunto com um certo grau de pertinência.

As operações com os conjuntos difusos podem ser definidas de várias formas. A mais comum é similar ao caso dos conjuntos *crisp* conforme demonstrado abaixo:

União:

$$A \cup B = \{ (x, \mu_{A \cup B}(x)), x \in U \} \quad \mu_{A \cup B}(x) = \max [ \mu_A(x), \mu_B(x) ] \quad (1)$$

Interseção:

$$A \cap B = \{ (x, \mu_{A \cap B}(x)), x \in U \} \quad \mu_{A \cap B}(x) = \min [ \mu_A(x), \mu_B(x) ] \quad (2)$$

Complemento:

$$A^c = \{ (x, \mu_{A^c}(x)), x \in U \} \quad \mu_{A^c}(x) = 1 - \mu_A(x) \quad (3)$$

Note-se que, depois de algumas discussões teóricas, segundo<sup>(7)</sup>, ficou comprovado que o operador  $\min$  não preenche todos os requisitos da lógica clássica para a geração das regras de inferência. A definição das funções de pertinência, utilizada em uma aplicação de sistemas difusos, é uma etapa fundamental e difícil no desenvolvimento dessa aplicação. Não existem regras definitivas para a escolha dessas funções as quais representam o conhecimento de um especialista no assunto em questão ou informações extraídas de um banco de dados. Algumas características das funções de pertinência são:

- Formato: Triangular, trapezoidal, Gaussiana, sigmoideal, etc;
- Obtenção: escolhida pelo usuário, baseado em sua experiência ou através de um processo de otimização a partir de dados experimentais e/ou obtidos por simulação;
- Normalização: Geralmente as funções de pertinência são definidas no intervalo  $[0,1]$ .

## 2.2 Variáveis Lingüísticas

São variáveis cujos valores não são números, e sim, palavras ou sentenças de uma linguagem natural ou artificial. Por exemplo, consideremos a variável *Temperatura*. Suponhamos que essa

variável possa assumir os valores lingüísticos *Baixa*, *Normal* e *Alta*. O conjunto de valores assumidos pela variável lingüística é denominado Conjunto de Termos e representado por  $T(x)$ , onde  $x$  é a variável. Os valores assumidos pela variável lingüística são representados por conjuntos difusos definidos pelas correspondentes funções de pertinência. No exemplo, teríamos:

Variável lingüística: Temperatura ( $T$ )

Conjunto de Termos:  $T(T) = \{ \text{Baixa, Normal e Alta} \}$

Funções de Pertinência dadas no gráfico conforme Figura 2.

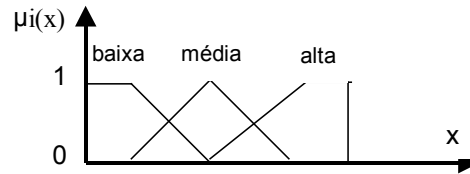


Figura 2. Função de pertinência

### 2.3 Sistemas de Inferência Difusa

Sejam  $x$  e  $y$  variáveis lingüísticas definidas nos universos de discurso  $X$  e  $Y$ , respectivamente. E sejam  $a$  e  $b$  valores assumidos por essas variáveis. Podemos formar um conjunto de proposições envolvendo essas variáveis. Para cada proposição existe um grau de pertinência, que mede a veracidade da regra. As regras mais utilizadas são:

Se  $X$  é  $a$  então  $y$  é  $b$

Se  $X$  é  $a$  e  $W$  é  $c$  então  $Y$  é  $b$

Se  $X$  é  $a$  ou  $W$  é  $c$  então  $Y$  é  $b$

A estrutura geral de um sistema de inferência difusa é mostrada na Figura 3:

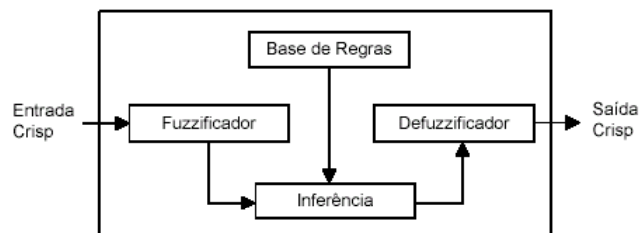


Figura 3. Sistema de Inferência Difusa

Os elementos dessa estrutura são:

- Base de Regras: coleção de regras do tipo **Se\_Então**, as quais podem ser expressas como:

Regra ( $r$ ): **SE  $u_1$  é  $F_{1r}$  e  $u_2$  é  $F_{2r}$  e ...  $u_n$  é  $F_{nr}$  ENTÃO  $v$  é  $G_r$** ,  $r=1, \dots, N$  (4)

onde:  $F_{ir}$ ,  $r=1, \dots, N$ , e  $G_r$  são conjuntos difusos definidos em  $u_i \subset R$ ,  $r = 1, 2, 3, \dots, N$  e  $v \subset R$ , respectivamente.

- Fuzzificador: determina o grau de pertinência de cada entrada no antecedente da regra. Se o antecedente tem mais de um componente (proposição), os operadores difusos E (min) e OU (max) são utilizados para combinar os efeitos respectivamente como a seguir:

$$A_r = \min_{i=1, \dots, n} \{ \mu_i(u_i) \} \quad (5)$$

$$A_r = \max_{i=1, \dots, n} \{ \mu_i(u_i) \} \quad (6)$$

onde:

$A_r$ : combinação do grau de pertinência dos antecedentes da regra;

$\mu_i(u_i)$ : grau de pertinência do  $i$ -ésimo antecedente para o valor da  $i$ -ésima entrada ( $u_i$ );

- Inferência: determina o grau de validade dos conseqüentes das regras e combina os resultado no conjunto difuso da saída. O principio utilizado assume que: regras com baixo grau de pertinência no antecedente ( $A_r$ ) devem ter pouca validade no conseqüente. Esse processo é implementado pelo operador Max-Min:

$$\mu G = \max \{ \min \{ \mu_r, \mu_i(ui) \}, \{ \mu_r(ui) \} \} \quad (7)$$

A operação acima equivale a redefinir os conjuntos difusos associados ao conseqüente das regras (modificação da forma das funções de pertinência) e a combinação dos mesmos em um conjunto difuso da saída.

- Defuzificador: produz uma saída *crisp* a partir do conjunto difuso definido pelo bloco de inferência. Alguns métodos de defuzificação utilizados são: Valor Máximo; Valor médio dos Máximos; Bissetor e Centróide ou centro de massa.

## 2.4 Tipos de Variáveis e Modelagem

Temos basicamente dois tipos de variáveis difusas: Subjetivas ou Sensitivas e Determinísticas ou Discretas. As variáveis Subjetivas são aquelas cuja função de pertinência é definida para qualquer valor contido em um intervalo contínuo, sendo que cada valor deste intervalo está associado a uma ou mais partições com um certo grau de pertinência para cada uma das partições. O nível de óleo ou a lubrificação de um equipamento são exemplos desse tipo de variável. As variáveis Determinísticas são definidas em um intervalo discreto, ou seja, existe uma função de pertinência para cada valor discreto da variável contido neste intervalo. Para os demais valores, a pertinência não é definida.

## 3 METODOLOGIA

A metodologia utilizada no desenvolvimento do software para fins de fazer a previsibilidade da manutenção preventiva de equipamentos é descrita nesta seção. Resumidamente, o desenvolvimento do projeto final passou pelas seguintes etapas como mostra a Figura 4:

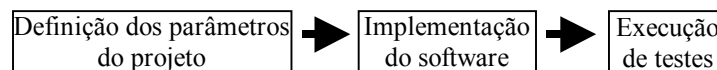


Figura 4. Etapas do desenvolvimento

### 3.1 Definição dos Parâmetros do SIAD

Nesta etapa foi realizado o estudo para conhecimento do problema, definição das condições e escopo do projeto e a decisão das ferramentas para desenvolvimento do projeto constituído por um sistema inteligente de apoio à decisão para fins de diagnóstico de equipamentos eletro/mecânicos. Neste sistema seriam incluídas as características de um determinado equipamento e as regras que ditam o comportamento do mesmo. A decisão de manutenção dos equipamentos é feita por meio de métodos heurísticos ou através de conhecimento especialista. Portanto, um caso típico de aplicação de lógica difusa.

Os softwares disponíveis no mercado são específicos demais ou muito limitados. Portanto, definiu-se que o sistema que seria desenvolvido deveria atender as seguintes condições:

- Interface amigável com o usuário, armazenamento das informações em banco de dados, capacidade de guardar as configurações de diversos equipamentos, não possuir recursos em excesso para não prejudicar a facilidade de operação do software

### 3.2 Implementação do Software

Foi utilizado como banco de dados o ACCESS e como linguagem de programação o VISUAL BASIC. A escolha dessas ferramentas seguiu basicamente dois critérios: Quantidade de recursos

suficientes para desenvolver o projeto e pré-conhecimento das ferramentas. O software possui os seguintes módulos/funcionalidades: Iniciar uma configuração para um novo equipamento; Abrir para edição uma configuração já salva de algum equipamento; Excluir uma configuração de algum equipamento; Definir variáveis de entrada; Definir partições das variáveis de entrada; Definir variáveis de saída; Definir partições das variáveis de saída subjetivas; Definir valores das variáveis de saída determinísticas; Definir as regras de inferência; Excluir as regras de inferência; Visualizar as regras de inferência; Definir os valores das variáveis de entrada; Gerar os valores das variáveis de saída. Cada um desses módulos foi implementado em uma tela diferente. Com isso foi possível separar as informações e facilitar a inclusão e a edição. Para informações mais detalhadas sobre essas funcionalidades. Foram rodadas simulações de diagnóstico de vários equipamentos. Foram incluídos valores de entrada fictícios e foi verificado que para cada grupo de valores de entrada, o software gerou a saída esperada.

### 3.3 Construção do Modelo Difuso

Para cada Equipamento cuja configuração é incluída no sistema, existe anteriormente a essa inclusão, uma construção do modelo difuso. Um diagrama desse modelo pode ser visto na Figura 5.

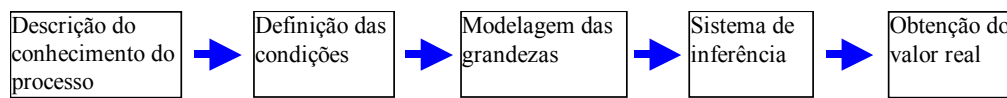


Figura 5. Modelo Difuso

### 3.4 Banco de Dados

Foi projetado um Banco de Dados em Access com a finalidade de armazenar as variáveis de entrada e saída, as partições dessas variáveis, as regras de inferência e os valores atribuídos as variáveis de entrada. Além disso, era necessário identificar qual modelagem ou configuração cada item citado pertencia.

## 4 INTERFACE DO SIAD

A interface foi desenvolvida com o intuito de ser bastante amigável, por isso todas as telas são auto-explicativas e seguem o mesmo padrão de inclusão e edição de dados, tornando mais rápida a familiarização com o programa, por parte do usuário. A Figura 6 mostra um diagrama de blocos simplificado para a geração de diagnóstico de um equipamento utilizando o software desenvolvido.

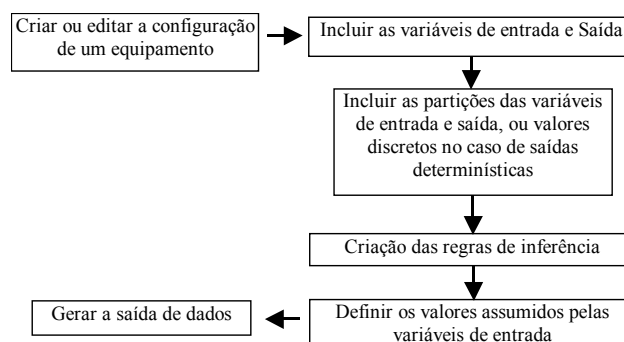


Figura 6. Fluxograma

A Figura 7 mostra os menus existentes na interface inicial do SIAD. O primeiro menu ao iniciar o software é o menu denominado Arquivo. Neste menu, podemos optar por criar uma Nova Configuração, abrir uma já existente ou excluir alguma configuração. O nome da configuração que está aberta é exibido na barra de título. Até criarmos ou abrirmos alguma configuração, os outros menus permanecerão desabilitados. A fim de garantir a integridade dos dados, o Software não permite a exclusão da configuração que está sendo usada.

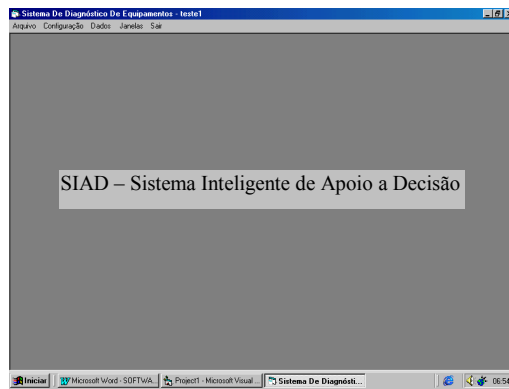


Figura 7. Tela inicial do SIAD

No menu, denominado Configuração, entra-se com as variáveis de entrada e saída, suas respectivas partições e as regras de inferência. No menu denominado Dados, são atribuídos valores para as variáveis de entrada e gerada a saída referente a esses valores. Para a definição das variáveis de entradas segue conforme a Figura 8.

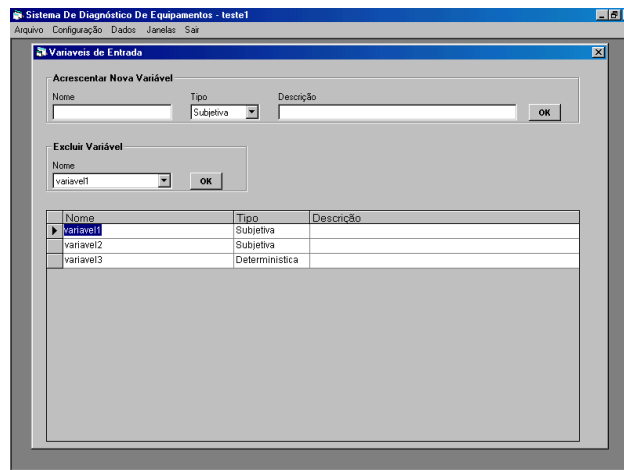


Figura 8. Variáveis de entrada

Esta tela é acessível através da opção Configuração. Aqui são definidas as variáveis de entrada do modelo difuso para o equipamento em questão.

Note-se que é necessário definir o nome da variável e seu tipo (Subjetiva ou Determinística). O campo descrição é opcional. Existe ainda a opção de excluir alguma variável. Na metade inferior da tela, existe um controle “Datagrid” que mostra as variáveis de entrada atuais do sistema. A tela de definição das variáveis de saída segue o mesmo padrão, sendo acessível também através da opção Configuração. Para a definição das partições das variáveis de entrada, veja a Figura 9.

Esta tela é também acessível através da opção Configuração. Nesta tela, inclui-se a partição de cada uma das variáveis de entrada. Antes de incluir a partição, devemos atribuir um nome para a mesma e determinar seus parâmetros. Ao adicionar uma partição para a variável de entrada em questão, o software verifica a validade dos valores. Não é permitido, por exemplo, que todos os parâmetros tenham um mesmo valor.

A tela de definição de partições para as variáveis de saída segue o mesmo padrão e as mesmas restrições, sendo acessível através também da opção Configuração.

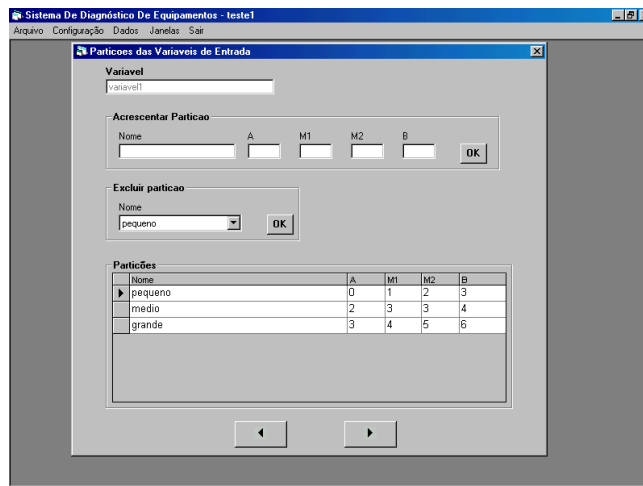


Figura 9. Partições

Para a definição dos valores determinísticos das variáveis de saída, tem-se a Figura 10.

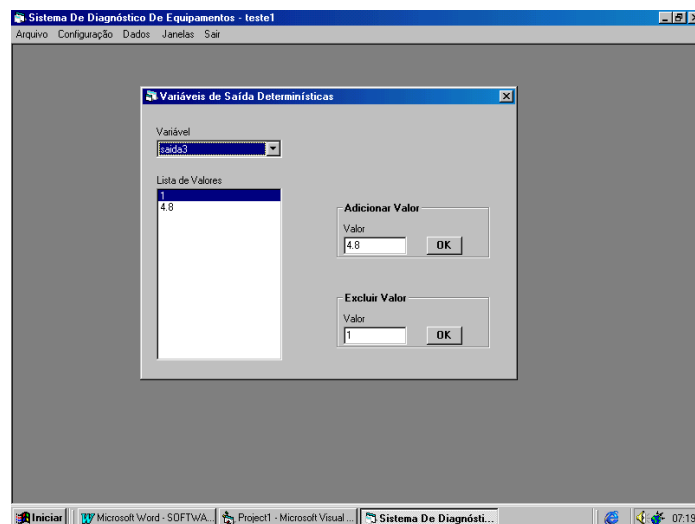


Figura 10. Valores Determinísticos

Através da opção configuração, acessa-se esta tela. As variáveis de saída podem ser subjetivas ou determinísticas. Quando são subjetivas, elas são definidas através de partições conforme demonstrado anteriormente e podem receber qualquer valor no intervalo contínuo compreendido entre dois parâmetros. Caso estas sejam determinísticas, então as variáveis só podem receber valores discretos e pré-definidos. Nesta tela definiremos os possíveis valores para cada saída determinística. Ao criar uma regra de inferência cuja saída seja uma variável determinística, só estarão disponíveis para o usuário, os valores definidos nesta tela. Para a criação de regras, segue a Figura 11.

Esta tela também é acessível através da opção Configuração. Aqui, criam-se as regras de inferência que regem o equipamento em questão.

É necessário especificar qual tipo de regra se deseja criar e adicionar as variáveis de entrada e saída. Os tipos de regras possíveis são: **Se...Então**, **Se...E... Então**, **Se...Ou...Então**. Ao adicionar as variáveis de entrada e saída, devemos também selecionar suas partições, ou valores discretos no caso de saídas determinísticas. Caso o número de variáveis de entrada não seja condizente com o tipo da regra, o sistema irá criticar através de uma mensagem de alerta e não criará a regra. Depois de criada, o texto da regra será exibido na caixa de texto na parte inferior da tela. Nesta tela exibimos apenas a última regra criada. A visualização de todas as regras bem como a opção de exclusão destas regras está acessível através da opção Configuração. Finalmente, pode-se acessar as variáveis de saída através da opção Dados. Ao carregarmos este formulário, o sistema verifica a



integridade dos dados e se todas as informações necessárias estão fornecidas. Após a verificação, caso todos os requisitos sejam atendidos, é feito todo o cálculo necessário para a geração da variável de saída. Caso contrário, uma mensagem de crítica é exibida.

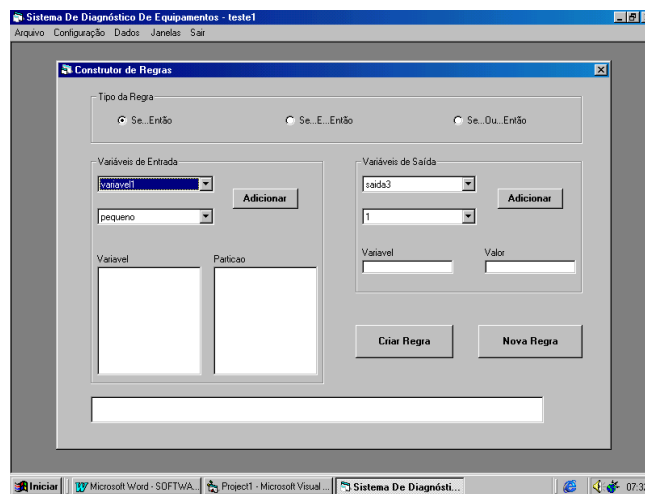


Figura 11. Regras

Foram utilizadas algumas técnicas e metodologias de engenharia de software no desenvolvimento do projeto objetivando sempre a qualidade do produto final. O ciclo básico utilizado na construção do software pode ser descrito pelas etapas: engenharia de software; análise de sistemas; projeto; codificação; teste e manutenção.

## 5 ESTUDO DE CASO E CONCLUSÃO

Desenvolveu-se um novo software denominado SIAD – Sistema Inteligente de Apoio à Decisão apoiado em lógica difusa para fins de diagnosticar estados de operação de equipamentos, visando determinação da previsibilidade da sua vida útil. Além disso, o SIAD atende as necessidades de realização de manutenção preditiva de forma atrativa e transparente ao usuário. Algumas vantagens do software SIAD diagnóstico, desenvolvido e apresentado aqui, estão relacionadas como se segue:

- ✓ Não possui restrição quanto ao número de entradas, saídas ou regras de inferência.
- ✓ Armazena a configuração (entradas, saídas e regras de inferência) de vários equipamentos, possibilitando posterior edição.
- ✓ Possui interface amigável.
- ✓ Gera a saída em tempo consideravelmente curto.
- ✓ Verifica a entrada de dados e sinaliza quando existe algum valor fora do conjunto permitido, garantindo, com isso, a integridade dos dados armazenados.

Vários estudos de casos foram desenvolvidos, tais como: Coifa de Convertedor, Segmentos de Máquina de Contínuo, Filtros de Mangas, Bombas Centrífugas e ISF. A facilidade e rapidez em obter resultados preditivos mostraram ser o SIAD uma ferramenta que deve ser incentivada o seu uso ao longo de todo o processo num chão-de-fábrica.

Até o presente momento o SIAD é o único software genérico que faz a previsibilidade de vida útil de equipamentos. Assim, o SIAD não pode ser comparado com as metodologias tradicionais de manutenção preditiva, pois as mesmas apenas apontam tendências, constituindo-se estas metodologias, desse modo, em estudos de análise de tendência.

Assim, este trabalho apresentou um original e moderno ambiente computacional na forma de um Sistema Inteligente de Apoio à Decisão – SIAD, para fins de diagnóstico de equipamentos elétrico/mecânicos. O SIAD, pela sua característica generalizada, poderá ser útil em processos de Operação/Manutenção em empresas e indústrias, permitindo dar subsídios à análise de riscos, a detecção de falhas e mais especificamente ao diagnóstico de equipamentos para suporte ao planejamento, programação e controle da manutenção.

## 6 REFERÊNCIAS

1. Cavati, C.R. A Importância da Manutenção Preditiva na Qualidade da Produção. Revista Manutenção News da Abraman, AnoIV – n. 19, p. 23-26, set./dez. 2003.
2. Cavati, C.R. A Gestão do Conhecimento na Engenharia da Manutenção Inteligente. Revista Engenharia Ciência & Tecnologia, CT-UFES, ISSN 1414-8692, Ano 6, v. 6 – n. 4, p. 18-20. jul./ag. 2003.
3. Cavati, C.R. A Decisão Inteligente e Seus Estilos. In: Anais do XXV SBPO, Natal, 2003. CD-ROM.
4. Mamdani, E.; Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. In: International Journal of Man-Machine Studies, 1998, v. 7.
5. Zimmermann H.J. Fuzzy Set Theory and Its Applications. Third Edition, Kluwer, 1996.
6. Ross T.J. Fuzzy Logic with Engineering Applications. McGraw-Hill International Editions, 1997.
7. Mendel J.M. Fuzzy Logic Systems for Engineering: A Tutorial. In: Proceedings of the IEEE, Março 1995, v. 83, n. 3, p. 345-377.
8. Tomsovic K. e Chow M.Y., Eds., Tutorial on Fuzzy Logic Applications in Power Systems. IEEE PES Winter Meeting, Singapore, jan. 2000.

## A DECISION SUPPORT INTELLIGENT SYSTEM TOOL FOR MAINTENANCE MANAGEMENT

**Cícero R. Cavati**

UFES/CT/DEL, Campus Universitário-Goiabeiras, Vitória ES, 29060-970 cavati@ele.ufes.br

**Evaldo Moura Junior**

UFES/CT/DEL, Campus Universitário-Goiabeiras, Vitória ES, 29060-970

**Abstract.** *The administrators of companies search to use more efficient programs each time in the attempt to support its decisions and to reach better results in its processes. It is natural that they daily continue searching tools each time more advanced to add value to the process and to survive to the competitiveness. Thus, the description of the stages of development of a new called software: Decision Support Intelligent System - SIAD for diagnosis ends is presented here. The SIAD was implemented in VISUAL BASIC VI, uses MS ACCESS as data base, and uses the fuzzy logic for the generation of the disgnostic. Amongst them vary advantages that software presents, are distinguished it makes possible consequently to generate, to store and to edit the characteristics of some equipment and its disgnostic. Therefore, this computational tool represents a chance of if supporting the accomplishment of the maintenance management in order to evaluate risks of stopped in equipment and industrial processes, as form of attainment of benefits, performance maximize, and loss minimize.*

**Keywords:** *Maintenance, Software, Intelligent Systems*