

**APLICAÇÃO DE TÉCNICAS DE INTELIGÊNCIA  
ARTIFICIAL NO DESENVOLVIMENTO DE UM SISTEMA  
DE MANUTENÇÃO BASEADA EM CONDIÇÃO**

**Edgar Jhonny Amaya Simeón**

**DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS  
DEPARTAMENTO DE ENGENHARIA MECANICA**

**FACULDADE DE TECNOLOGIA  
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**APLICAÇÃO DE TÉCNICAS DE INTELIGÊNCIA  
ARTIFICIAL NO DESENVOLVIMENTO DE UM SISTEMA  
DE MANUTENÇÃO BASEADA EM CONDIÇÃO**

**EDGAR JHONNY AMAYA SIMEÓN**

**ORIENTADOR: ALBERTO JOSÉ ÁLVARES**

**DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS**

**PUBLICAÇÃO: ENM.DM - 21A/08**

**BRASÍLIA/DF: JULHO – 2008**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**APLICAÇÃO DE TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL NO  
DESENVOLVIMENTO DE UM SISTEMA DE MANUTENÇÃO  
BASEADA EM CONDIÇÃO**

**EDGAR JHONNY AMAYA SIMEÓN**

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE  
ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA  
DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS  
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE  
MESTRE EM SISTEMAS MECATRÔNICOS.**

**APROVADA POR:**

---

**Prof. Alberto José Álvares, Dr. Eng. (ENM-UnB)  
(Orientador)**

---

**Prof. Carlos Humberto Llanos Quintero, Dr. Eng. (ENM-UnB)  
(Examinador Interno)**

---

**Prof. Jonny Carlos da Silva, Dr. Eng. (UFSC)  
(Examinador Externo)**

**BRASÍLIA/DF, 25 DE JULHO DE 2008**

## FICHA CATALOGRÁFICA

SIMEÓN, EDGAR JHONNY AMAYA

Aplicação de Técnicas de Inteligência Artificial no Desenvolvimento de um Sistema de Manutenção Baseada em Condição [Distrito Federal] 2008.

xx, 172p., 210 x 297 mm (ENC/FT/UnB, Mestre, Sistemas Mecatrônicos, 2008).

Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

1. Manutenção Baseada em Condição

2. Sistemas Especialistas

3. Fuzzy ARTMAP

4. Diagnóstico de Falhas

5. Foundation Fieldbus

6. OSA-CBM

I. ENM/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

AMAYA, E. J. (2008). Aplicação de Técnicas de Inteligência Artificial no Desenvolvimento de um Sistema de Manutenção Baseada em Condição. Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-21A/08, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 172p.

## CESSÃO DE DIREITOS

AUTOR: Edgar Jhonny Amaya Simeón.

TÍTULO: Aplicação de Técnicas de Inteligência Artificial no Desenvolvimento de um Sistema de Manutenção Baseada em Condição.

GRAU: Mestre

ANO: 2008

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

---

Edgar Jhonny Amaya Simeón  
SCLN 407 BLOCO C SALA 221  
70.855-530 Brasília - DF - Brasil.

## DEDICATÓRIA

Este trabalho é dedicado em primeiro lugar a Deus, aos meus pais Crisanto e Maurelia pela educação exemplar que me foi concedida, a todos meus irmãos por torcerem por meu sucesso, à Incancellabile e a todos os amigos e colegas que contribuíram para a realização desse sonho.

## **AGRADECIMENTOS**

Agradeço

A Deus, pela sua luz, paz e amor que me proporciona todos os dias, mesmo nos momentos mais difíceis, nunca me abandonou cuidando dos menores detalhes em minha existência;

À minha família, que mesmo estando longe, sempre torceu muito por mim;

Ao meu orientador, Prof. Dr. Eng. Alberto José Álvares, pelos conhecimentos transmitidos, competência, orientação e apoio;

À Fundação de Empreendimentos Científicos e Tecnológicos (FINATEC) pelo apoio financeiro concedido para o desenvolvimento do projeto;

Ao Grupo de Automação e Controle (GRACO) e ao Departamento de Engenharia Mecânica da Universidade de Brasília pelos recursos físicos fornecidos;

Ao Eng. Antonio Araujo da Eletronorte e ao pessoal da Manaus Energia pelo apoio no desenvolvimento do projeto;

Ao Prof. Dr. Eng. Ricardo Ribeiro Gudwin, pelo apoio no desenvolvimento computacional;

A todos os professores que formam o corpo docente do programa de pós-graduação em Sistemas Mecatrônicos.

Aos meus colegas do Projeto, Rosimarci, Rodrigo e Giovanni;

Aos meus colegas do Laboratório, Claudia, Ana Maria, Yesid, Rodrigo, Diego, Alvaro, Jones, Magno, Andre, Luição, Victor Celestino, Carlos Frederico, etc.;

Aos meus amigos de *juerga* Juan Carlos, Max, Eber, Faura, Martin e Dudu;

Aos companheiros de casa Jimmy, Fernand e Diego Felipe;

E a todos os amigos e colegas que me apoiaram em mais esta conquista da minha vida.

## RESUMO

### APLICAÇÃO DE TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL NO DESENVOLVIMENTO DE UM SISTEMA DE MANUTENÇÃO BASEADA EM CONDIÇÃO

**Autor:** Edgar Jhonny Amaya Simeón

**Orientador:** Alberto José Álvares

**Programa de Pós-graduação em Sistemas Mecatrônicos**

**Brasília, julho de 2008**

O objetivo principal deste trabalho é apresentar uma metodologia para o desenvolvimento de um sistema inteligente de manutenção baseada em condição usando sistemas especialistas e um sistema inteligente híbrido baseado em lógica nebulosa e redes neurais artificiais, e a implementação de sistemas especialistas. A metodologia é baseada no modelo de referência de sete camadas OSA-CBM (*Open System Architecture for Condition Based Maintenance*) adaptado a uma arquitetura computacional cliente/servidor. No lado servidor são desenvolvidas as seis primeiras camadas que executam as tarefas de aquisição de dados, processamento inteligente e tomada de decisão. A aquisição de dados *online* é via servidor OPC e dos dados históricos através de banco de dados. A sétima camada (apresentação) encontra-se no lado cliente e apresenta informações de todas as camadas prévias e permite a visualização gráfica de variáveis em tempo real e histórico, associadas à evolução dos defeitos e falhas das máquinas e equipamentos. A abordagem concebida poderá ser utilizada para o auxílio na tomada de decisão através de sugestões de ações de manutenção. As sugestões são baseadas nos diagnósticos e prognósticos do estado de funcionamento de máquinas e equipamentos. No projeto do sistema inteligente de manutenção baseada em condição utilizaram-se os métodos IDEF0 e IDEF1X da metodologia de modelagem IDEF, para documentar o modelo funcional e de informação respectivamente. A representação UML em modelo casos de uso foi utilizada para projetar o software. O sistema computacional foi desenvolvido em Java permitindo a independência do sistema operativo do usuário. O sistema inteligente instalado na usina hidrelétrica de Balbina mostra-se capaz de detectar defeitos nos seus sistemas e subsistemas antes que as mesmas atinjam um estágio de falha provocando paradas inesperadas.

## RESUMEN

### APLICACIÓN DE TÉCNICAS DE INTELIGENCIA ARTIFICIAL EN EL DESARROLLO DE UN SISTEMA DE MANTENIMIENTO BASADO EN CONDICIÓN

**Autor: Edgar Jhonny Amaya Simeón**

**Supervisor: Alberto José Álvares**

**Programa de Pós-graduação em Sistemas Mecatrônicos**

**Brasília, Julio del 2008**

El objetivo principal de este trabajo es presentar una metodología para el desarrollo de un sistema inteligente de mantenimiento basado en condición usando sistemas especialistas y un sistema inteligente híbrido basado en lógica difusa y redes neuronales artificiales, y la implementación de sistemas especialistas. La metodología es basado en el modelo de referencia de siete capas OSA-CBM (*Open System Architecture for Condition Based Maintenance*) adaptado a una arquitectura computacional cliente/servidor. En el lado servidor son desarrolladas las seis primeras capas que ejecutan tareas de adquisición de datos, procesamiento inteligente e tomada de decisión. La adquisición de datos *online* es vía servidor OPC y los datos históricos es a través de banco de datos. La séptima capa (presentación) se encuentra en el lado cliente y presenta informaciones de todas las capas previas y permite la visualización gráfica de variables en tiempo real e histórico asociadas a la evolución de los defectos y fallas de máquinas y equipos. El enfoque concebido podrá ser utilizado para el auxilio en la tomada de decisión a través de sugerencias de acciones de mantenimiento. Las sugerencias son basadas en diagnósticos y pronósticos del estado de funcionamiento de máquinas y equipos. El proyecto del sistema inteligente de mantenimiento basado en condición se utiliza los métodos IDEF0 e IDEF1X de la metodología de modelaje IDEF, para documentar el modelo funcional y de información respectivamente. La representación UML en modelo casos de uso fue utilizada para proyectar el software. El sistema computacional fue desarrollado en Java permitiendo la independencia del sistema operativo del usuario. El sistema inteligente instalado en la central hidroeléctrica de Balbina es capaz de detectar defectos en los sistemas y subsistemas antes que las mismas lleguen a un estado de falla provocando paradas inesperadas.



## **ABSTRACT**

### **ARTIFICIAL INTELLIGENCE TECHNIQUES APPLICATION IN THE DEVELOPMENT OF A CONDITION BASED MAINTENANCE SYSTEM**

**Author: Edgar Jhonny Amaya Simeón**

**Supervisor: Alberto José Álvares**

**Programa de Pós-graduação em Sistemas Mecatrônicos**

**Brasília, July of 2008**

The main objective of this work is to present a methodology for development of an intelligent system for condition based maintenance using expert systems and hybrid intelligent system based on fuzzy logic and artificial neural network, and the implementation of expert systems. The methodology is based on the seven layers OSA-CBM (Open System Architecture for Condition Based Maintenance) reference model adapted to a client/server computational architecture. In the server side the first six layers are developed, that execute data acquisition tasks, intelligent processing and decision support. The online data acquisition is performed through OPC server and the historical data is performed through database. The seventh layer (presentation - human interface) is located in the client side and presents information from all the previous layers and allows a graphic visualization of variables in real time and historical, associated to defects and failures evolution of machines and equipments. The approach conceived can be used to decision support through suggestions for maintenance practices. The suggestions are based on diagnostics and prognostics of the machines and equipments operation state. In the project of the intelligent system for condition based maintenance was used the IDEF0 and IDEF1X methods of the IDEF modeling methodology, for document the functional model and information respectively. The representation UML in use case model was utilized for project the software. The computational system was developed in Java allowing the system to be independent of operational system. The intelligent system installed in the hydroelectric power plant of Balbina shows to be capable to detect defects in its systems and subsystems before a failure stage that can yield unexpected stops.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>1</b>
<b>1.1</b>	<b>OBJETIVOS.....</b>	<b>3</b>
1.1.1	Objetivos Gerais .....	3
1.1.2	Objetivos Específicos .....	4
<b>1.2</b>	<b>FORMULAÇÃO DO PROBLEMA.....</b>	<b>4</b>
<b>1.3</b>	<b>ESTRUTURA DO DOCUMENTO.....</b>	<b>5</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA: MANUTENÇÃO BASEADA EM CONDIÇÃO..</b>	<b>6</b>
<b>2.1</b>	<b>INTRODUÇÃO.....</b>	<b>6</b>
<b>2.2</b>	<b>MANUTENÇÃO BASEADA EM CONDIÇÃO .....</b>	<b>13</b>
<b>2.3</b>	<b>SISTEMAS DE MANUTENÇÃO BASEADA EM CONDIÇÃO .....</b>	<b>14</b>
<b>2.4</b>	<b>TECNOLOGIAS USADAS NOS SISTEMAS DE MBC .....</b>	<b>15</b>
2.4.1	Fieldbus .....	15
2.4.2	Componentes COM/DCOM.....	19
2.4.3	A tecnologia OLE/COM.....	20
2.4.4	A tecnologia OPC.....	21
2.4.5	Comparação das Tecnologias .....	25
	A escolha da tecnologia OPC é porque é um protocolo aberto, transparente e independente do fabricante. A maioria dos instrumentos, controladores, CLP, etc., disponibilizam seus dados via um servidor OPC ou as armazenam em outro formato próprio do fabricante. ....	25
<b>2.5</b>	<b>ARQUITETURAS DOS SISTEMAS DE MANUTENÇÃO BASEADA EM CONDIÇÃO .....</b>	<b>25</b>
<b>2.6</b>	<b>OSA-CBM.....</b>	<b>28</b>
2.6.1	Arquitetura OSA-CBM.....	30
2.6.1.1	Aquisição de dados .....	33
2.6.1.2	Processamento de sinal .....	33
2.6.1.3	Monitoração de condição.....	33
2.6.1.4	Avaliação de saúde (diagnóstico) .....	34
2.6.1.5	Prognósticos.....	34
2.6.1.6	Tomada de decisão.....	36
2.6.1.7	Apresentação.....	36
<b>2.7</b>	<b>CONCLUSÕES DO CAPÍTULO.....</b>	<b>36</b>
<b>3</b>	<b>REVISÃO BIBLIOGRÁFICA: TÉCNICAS INTELIGENTES .....</b>	<b>38</b>
<b>3.1</b>	<b>INTRODUÇÃO.....</b>	<b>38</b>
<b>3.2</b>	<b>SISTEMAS ESPECIALISTAS.....</b>	<b>39</b>
3.2.1	Arquitetura de um sistema especialista.....	40
3.2.2	Ferramentas para construção de SE.....	41

3.2.3	Regras de produção .....	42
<b>3.3</b>	<b>REDES NEURAS ARTIFICIAIS .....</b>	<b>43</b>
<b>3.4</b>	<b>LÓGICA NEBULOSA .....</b>	<b>45</b>
<b>3.5</b>	<b>MODELO HÍBRIDO .....</b>	<b>47</b>
3.5.1	Arquitetura FAM.....	48
3.5.2	Treinamento da FAM.....	52
3.5.3	Pseudocódigo para treinamento da FAM.....	54
3.5.4	Desempenho da FAM.....	55
<b>3.6</b>	<b>INTELIGÊNCIA ARTIFICIAL APLICADA À MBC .....</b>	<b>57</b>
<b>3.7</b>	<b>CONCLUSÕES DO CAPÍTULO.....</b>	<b>58</b>
<b>4</b>	<b>METODOLOGIA .....</b>	<b>60</b>
<b>4.1</b>	<b>INTRODUÇÃO.....</b>	<b>60</b>
<b>4.2</b>	<b>MODELO DE REFERÊNCIA OSA-CBM .....</b>	<b>62</b>
4.2.1	Aquisição de dados.....	62
4.2.2	Processamento de sinal .....	62
4.2.3	Monitoração de condição.....	63
4.2.4	Avaliação de saúde (diagnóstico) .....	63
4.2.5	Prognósticos .....	63
4.2.6	Tomada de decisão .....	64
4.2.7	Apresentação .....	65
<b>4.3</b>	<b>MODELAGEM FUNCIONAL IDEF0 .....</b>	<b>65</b>
4.3.1	Atividade <i>I-kernel</i> .....	67
4.3.1.1	Atividade de prognóstico.....	70
4.3.2	Atividade clientes <i>web</i> .....	74
4.3.2.1	Atividade Cliente Applet.....	74
<b>4.4</b>	<b>MODELAGEM DE DADOS IDEF1X.....</b>	<b>75</b>
<b>5</b>	<b>SIMPREBAL: MODELAGEM UML .....</b>	<b>78</b>
<b>5.1</b>	<b>INTRODUÇÃO.....</b>	<b>78</b>
5.1.1	Arquitetura do SIMPREBAL .....	78
<b>5.2</b>	<b>REQUISITOS DE USUÁRIO.....</b>	<b>79</b>
5.2.1	Requisitos Funcionais.....	79
5.2.2	Requisitos Não-Funcionais.....	80
<b>5.3</b>	<b>REQUISITOS DO SISTEMA.....</b>	<b>81</b>
5.3.1	Casos de uso da aplicação <i>I-kernel</i> .....	82
5.3.1.1	Iniciação do <i>I-kernel</i> .....	82
5.3.1.2	Processamento inteligente.....	83
5.3.1.3	Verificação de alarmes e alertas.....	84
5.3.1.4	Shutdown do <i>I-kernel</i> .....	84

5.3.2	Casos de uso da ferramenta de C&M .....	85
5.3.2.1	Iniciação da Ferramenta C&M.....	86
5.3.2.2	Monitoramento de sinótico .....	86
5.3.2.3	Atualização de sinótico .....	86
5.3.2.4	Inspeção de variáveis .....	87
5.3.2.5	Inspeção de variáveis online .....	88
5.3.2.6	Inspeção de variáveis históricas.....	89
5.3.2.7	Shutdown da ferramenta C&M.....	89
<b>6</b>	<b>SIMPREBAL: IMPLEMENTAÇÃO COMPUTACIONAL.....</b>	<b>90</b>
<b>6.1</b>	<b>INTRODUÇÃO.....</b>	<b>90</b>
6.1.1	Requisitos Físicos.....	91
6.1.2	Extração do conhecimento dos especialistas .....	94
6.1.3	Arquivos de regras.....	95
<b>6.2</b>	<b>REGRAS DE PRODUÇÃO PARA AS CAMADAS OSA-CBM.....</b>	<b>96</b>
6.2.1	Estrutura de Regras do processamento de sinal.....	96
6.2.1.1	Processamento de sinal OPC .....	97
6.2.1.2	Processamento de sinal fieldbus .....	98
6.2.2	Estrutura de Regras da monitoração de condição .....	98
6.2.3	Estrutura de Regras da avaliação de saúde .....	99
6.2.4	Estrutura de Regras da tomada de decisão.....	100
<b>6.3</b>	<b>CLASSES IMPLEMENTADAS NO SIMPREBAL .....</b>	<b>101</b>
6.3.1	Classes do <i>I-Kernel</i> .....	101
6.3.2	Classes do <i>Confmonittool</i> .....	103
<b>7</b>	<b>ESTUDO DE CASO: GERADOR ELÉTRICO .....</b>	<b>104</b>
<b>7.1</b>	<b>INTRODUÇÃO.....</b>	<b>104</b>
7.1.1	Gerador elétrico principal.....	107
7.1.2	Resfriamento do gerador .....	107
7.1.3	Regulação de tensão .....	107
<b>7.2</b>	<b>RESULTADOS OBTIDOS .....</b>	<b>108</b>
<b>7.3</b>	<b>VALIDAÇÃO.....</b>	<b>116</b>
7.3.1	Base de conhecimento .....	116
7.3.2	Servidor .....	117
7.3.3	Cliente .....	118
<b>8</b>	<b>CONCLUSÕES, CONTRIBUIÇÕES E SUGESTÕES PARA TRABALHOS FUTUROS .....</b>	<b>119</b>
<b>8.1</b>	<b>CONCLUSÕES.....</b>	<b>119</b>
<b>8.2</b>	<b>CONTRIBUIÇÕES DO TRABALHO.....</b>	<b>119</b>
<b>8.3</b>	<b>IMPLEMENTAÇÃO COMPUTACIONAL.....</b>	<b>120</b>

8.3.1	Servidor .....	120
8.3.2	Cliente .....	121
<b>8.4</b>	<b>SUGESTÕES PARA TRABALHOS FUTUROS .....</b>	<b>121</b>
8.4.1	Implementação de algoritmos de prognóstico .....	122
8.4.2	Ampliação da base de conhecimento.....	122
8.4.3	Integração dos modelos de referência OSA-CBM e OSA-EAI.....	122
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>		<b>123</b>
<b>APÊNDICE A – ABORDAGENS DE ALGORITMOS DE PROGNÓSTICOS.....</b>		<b>136</b>
<b>A.1</b>	<b>INTRODUÇÃO.....</b>	<b>136</b>
<b>A.2</b>	<b>PROGNÓSTICO BASEADO NA EXPERIÊNCIA .....</b>	<b>138</b>
<b>A.3</b>	<b>PROGNÓSTICO BASEADO EM CARACTERÍSTICAS .....</b>	<b>138</b>
<b>A.4</b>	<b>PROGNÓSTICO BASEADO EM DADOS.....</b>	<b>139</b>
<b>A.5</b>	<b>PROGNÓSTICO BASEADO EM MODELOS FÍSICOS .....</b>	<b>141</b>
<b>A.6</b>	<b>PROGNÓSTICO ADAPTATIVO.....</b>	<b>142</b>
<b>APÊNDICE B – ARQUIVOS DE CONFIGURAÇÃO .....</b>		<b>144</b>
<b>B.1</b>	<b>ARQUIVO DE CONFIGURAÇÃO .....</b>	<b>144</b>
B.1.1	Configuração geral.....	144
B.1.2	Parâmetros da FAM.....	145
B.1.2	Servidores OPC e Tags OPC .....	145
B.1.2	Servidores de banco de dados .....	146
B.1.2	Tags Simuladas .....	147
B.1.2	Dispositivos DFI.....	148
B.1.2	Servidor de Email .....	148
<b>B.2</b>	<b>ARQUIVO CÓDIGOS DE FALHA .....</b>	<b>149</b>
<b>B.3</b>	<b>ARQUIVO CÓDIGOS DE DECISÃO.....</b>	<b>150</b>
<b>APÊNDICE C – CÁLCULO DOS CAMPOS DA CLASSE TAG.....</b>		<b>151</b>
<b>C.1</b>	<b>PROCESSAMENTO DO ITEM VALUE .....</b>	<b>151</b>
<b>C.2</b>	<b>PROCESSAMENTO DO ITEM STATUS.....</b>	<b>152</b>
<b>APÊNDICE D – MÉTODOS DE ANÁLISE FMEA E FTA .....</b>		<b>154</b>
<b>D.1</b>	<b>ANÁLISE DE MODOS E EFEITOS DE FALHAS - FMEA .....</b>	<b>154</b>
<b>D.2</b>	<b>ANÁLISE DE ÁRVORE DE FALHAS - FTA.....</b>	<b>157</b>
<b>D.3</b>	<b>COMPARAÇÃO ENTRE FTA E FMEA.....</b>	<b>158</b>
<b>APÊNDICE E – INICIAÇÃO E OPERAÇÃO DO SIMPREBAL .....</b>		<b>160</b>

<b>E.1 SIMPREBAL SERVER.....</b>	<b>160</b>
<b>E.2 SIMPREBAL CLIENT.....</b>	<b>160</b>
E.2.1 Home.....	162
E.2.2 Sistema.....	163
E.2.3 Históricos.....	163
E.2.4 KPIs.....	165
E.2.5 Produtos gerados.....	166
E.2.6 Colaboradores.....	167
E.2.7 Editar Cadastro.....	167
E.2.8 Sinótico SIMPREBAL (cliente <i>Applet</i> ).....	168
E.2.9 Inspeção De Variáveis.....	169
E.2.10 Visualizar gráfico em tempo real.....	171
E.2.11 Visualizar gráfico histórico.....	172

## LISTA DE TABELAS

TABELA 2.1- <i>SOFTWARES</i> MAIS DIFUNDIDOS (MYQ, 2000). .....	12
TABELA 2.2-TIPOS DE <i>FIELD</i> BUS DEFINIDOS NO PADRÃO IEC61158 (HÜSEMANN E PEREIRA, 2007).....	18
TABELA 3.1- FERRAMENTAS DE <i>SOFTWARE</i> PARA SE (REIS E PATI, 2000).....	40
TABELA 5.1- REQUISITOS FUNCIONAIS (RFs) (AMAYA <i>ET AL.</i> 2007A, MODIFICADO).....	80
TABELA 5.2- REQUISITOS NÃO FUNCIONAIS (RNFs) (AMAYA <i>ET AL.</i> 2007A, MODIFICADO)	81
TABELA 6.1- MODELO DE REGRAS PARA PROCESSAMENTO DE SINAL OPC.....	97
TABELA 6.2- MODELO DE REGRAS PARA PROCESSAMENTO DE SINAL <i>FIELD</i> BUS. ....	97
TABELA 6.3- MODELO DE REGRAS PARA FAIXAS DE OPERAÇÃO. ....	98
TABELA 6.4- MODELO DE REGRAS PARA MONITORAÇÃO DE CONDIÇÃO. ....	98
TABELA 6.5- MODELO DE REGRAS PARA DIAGNÓSTICO DOS CANAIS DE COMUNICAÇÃO <i>FIELD</i> BUS. ....	99
TABELA 6.6- MODELO DE REGRAS PARA DIAGNÓSTICO DA INSTRUMENTAÇÃO <i>FIELD</i> BUS. ....	99
TABELA 6.7- MODELO DE REGRAS PARA DIAGNÓSTICO DA MONITORAÇÃO DE CONDIÇÃO....	99
TABELA 6.8- MODELO DE REGRAS PARA TOMADA DE DECISÃO. ....	100
TABELA 7.1- TAGS ASSOCIADAS AO SUBSISTEMA GERADOR ELÉTRICO PRINCIPAL.....	107
TABELA 7.2- TAGS ASSOCIADAS AO SUBSISTEMA RESFRIAMENTO DO GERADOR. ....	108
TABELA 7.3- TAGS ASSOCIADAS AO SUBSISTEMA REGULAÇÃO DE TENSÃO. ....	108
TABELA A.1- MODELOS E INFORMAÇÃO PARA IMPLEMENTAÇÃO DE SISTEMAS DE PROGNÓSTICOS (ROEMER <i>ET AL.</i> 2005, MODIFICADO). ....	138
TABELA C.1- RELAÇÃO ENTRE O <i>QUALITY</i> E VALORES <i>QUALITY</i> E <i>SUBQUALITY</i> .....	152
TABELA C.2- VALORES DE <i>SUBSTATUS</i> E <i>QUALITY</i> NO ITEM <i>STATUS</i> (AMAYA <i>ET AL.</i> , 2007C). ....	153
TABELA D.1- FORMULÁRIO PADRONIZADO DE ANÁLISE FMEA. ....	154
TABELA D.2- SIGNIFICADO DOS ÍNDICES DE SEGURANÇA E/OU MEIO AMBIENTE.....	155
TABELA D.3- SIGNIFICADO DOS ÍNDICES DE PERDA DE FATURAMENTO.....	156
TABELA D.4- SIGNIFICADO DOS ÍNDICES DE CORTE DE CARGA. ....	156
TABELA D.5- SIGNIFICADO DOS ÍNDICES DE SEVERIDADE.....	156
TABELA D.6- SIGNIFICADO DOS ÍNDICES DE OCORRÊNCIA DE FALHAS.....	157
TABELA D.7- SIGNIFICADO DOS ÍNDICES DE DETECÇÃO.....	157
TABELA D.8- CAMINHOS CRÍTICOS DA ÁRVORE DE FALHAS. ....	158
TABELA D.9- COMPARAÇÃO ENTRE FTA E FMEA. ....	159

## LISTA DE FIGURAS

FIGURA 2.1- CRESCIMENTO DAS EXPECTATIVAS DE MANUTENÇÃO (MOUBRAY, 1997).....	8
FIGURA 2.2- MUDANÇAS DE VISÃO NA FALHA DO EQUIPAMENTO (MOUBRAY, 1997).....	8
FIGURA 2.3- MUDANÇA DAS TÉCNICAS DE MANUTENÇÃO (MOUBRAY, 1997).....	9
FIGURA 2.4- DIFERENTES TIPOS DE MANUTENÇÃO (BARROSO MAIA JUNIOR, 2003). ....	10
FIGURA 2.5- CAMADAS DE UM SISTEMA DE MANUTENÇÃO BASEADA EM CONDIÇÃO (LEBOLD ET AL. 2003, MODIFICADO).....	14
FIGURA 2.6- REDE <i>FIELDBUS</i> COMO UMA REDE LOCAL DE INSTRUMENTOS. ....	16
FIGURA 2.7- REDES <i>FIELDBUS</i> DE BAIXA E ALTA VELOCIDADE, H1 E HSE .....	17
FIGURA 2.8- ARQUITETURA DA INFORMAÇÃO NO CONTROLE DE PROCESSOS (OPC FOUNDATION 1998, MODIFICADO).....	21
FIGURA 2.9- APLICAÇÕES COM VÁRIOS SERVIDORES OPC. ....	22
FIGURA 2.10- RELAÇÃO ENTRE CLIENTES E SERVIDORES.....	23
FIGURA 2.11- ARQUITETURA OPC .....	24
FIGURA 2.12- ARQUITETURA GERAL DE UM SISTEMA DE PROGNÓSTICO (VACHTSEVANOS E WANG 2001, MODIFICADO). ....	26
FIGURA 2.13- ELEMENTOS BÁSICOS DE DIAGNÓSTICO E PROGNÓSTICO PARA MANUTENÇÃO BASEADA EM CONDIÇÃO, APRESENTADO POR (CHINNAM E BARUAH 2004, MODIFICADO). .....	26
FIGURA 2.14- AS SETE CAMADAS FUNCIONAIS OSA-CBM (AMAYA ET AL., 2007C). ....	27
FIGURA 2.15- FLUXO DE DADOS DAS CAMADAS OSA-CBM .....	31
FIGURA 2.16- CAMADAS FUNCIONAIS OSA-CBM (LEBOLD E THURSTON 2001, MODIFICADO).....	32
FIGURA 2.17- AS TRÊS CAMADAS MBC (JARDINE ET AL. 2006, MODIFICADO).....	32
FIGURA 2.18- CAMADAS OSA-CBM, DESTACANDO OS PADRÕES QUE PODEM SER USANDO NO DESENVOLVIMENTO DE SISTEMAS DE MBC (BENGTSSON 2004A, MODIFICADO). ....	34
FIGURA 2.19- COMPONENTES DA CAMADA DE PROGNÓSTICO GENÉRICO (LEBOLD E THURSTON 2001, MODIFICADO).....	35
FIGURA 2.20- ENTRADAS E SAÍDAS GERAIS DA CAMADA DE PROGNÓSTICO OSA-CBM (LEBOLD E THURSTON 2001, MODIFICADO).....	36
FIGURA 3.1- CONTEXTO HISTÓRICO DOS SE (CUNHA, 1995). ....	39
FIGURA 3.2- ARQUITETURA DE UM SISTEMA ESPECIALISTA (ABEL, 1998).....	41



FIGURA 3.3- ESQUEMA DE INTERAÇÃO ENTRE O EC E O ESPECIALISTA NO DOMÍNIO DO PROBLEMA (WATERMAN, 1986). .....	41
FIGURA 3.4- MODELO DE NEURÔNIO ARTIFICIAL (AMAYA <i>ET AL.</i> 2007B, MODIFICADO) .....	43
FIGURA 3.5- REDE NEURAL ARTIFICIAL MULTICAMADA .....	44
FIGURA 3.6- MODELO ART ESQUEMATIZADO .....	44
FIGURA 3.7- PROCESSO DE FUZIFICAÇÃO DA VARIÁVEL TEMPERATURA .....	46
FIGURA 3.8- MODELO FAM (CARPENTER <i>ET AL.</i> , 1992, MODIFICADO).....	47
FIGURA 3.9- ARQUITETURA FAM SIMPLIFICADA. ....	49
FIGURA 4.1- TÉCNICAS DE IA USADA NAS CAMADAS OSA-CBM.....	61
FIGURA 4.2- FAIXAS DE OPERAÇÃO DE UM EQUIPAMENTO.....	63
FIGURA 4.3- DIAGRAMA A0 DO SISTEMA INTELIGENTE DE MBC.....	66
FIGURA 4.4- DIAGRAMA FILHO DA FUNÇÃO A0. ....	68
FIGURA 4.5- DIAGRAMA FILHO DA FUNÇÃO A1. ....	69
FIGURA 4.6- DIAGRAMA FILHO DA FUNÇÃO A15. ....	71
FIGURA 4.7- DIAGRAMA FILHO DA FUNÇÃO A2. ....	72
FIGURA 4.8- DIAGRAMA FILHO DA FUNÇÃO A21. ....	73
FIGURA 4.9- MODELAGEM DA INFORMAÇÃO DO ARQUIVO DE CONFIGURAÇÃO ATRAVÉS DA METODOLOGIA.....	76
FIGURA 4.10- MODELAGEM DA INFORMAÇÃO DO BANCO DE DADOS ATRAVÉS DA METODOLOGIA.....	77
FIGURA 5.1- PRINCIPAIS PACOTES DA ARQUITETURA SIMPREBAL. ....	78
FIGURA 5.2- O <i>I-KERNEL</i> E SUA INTERAÇÃO (AMAYA <i>ET AL.</i> , 2007C). ....	79
FIGURA 5.3- CASO DE USO DO SISTEMA INTELIGENTE.....	82
FIGURA 5.4- INICIAÇÃO DA APLICAÇÃO <i>I-KERNEL</i> . ....	83
FIGURA 5.5- PROCESSAMENTO INTELIGENTE.....	83
FIGURA 5.6- VERIFICAÇÃO DE ALARMES E ALERTAS. ....	84
FIGURA 5.7- <i>SHUTDOWN</i> DA APLICAÇÃO <i>I-KERNEL</i> . ....	85
FIGURA 5.8- INICIAÇÃO DA FERRAMENTA DE C&M. ....	85
FIGURA 5.9- MONITORAMENTO DE SINÓTICO. ....	86
FIGURA 5.10- ATUALIZAÇÃO DE SINÓTICO. ....	87
FIGURA 5.11- INSPEÇÃO DE VARIÁVEIS. ....	87
FIGURA 5.12- INSPEÇÃO DE VARIÁVEIS <i>ONLINE</i> . ....	88
FIGURA 5.13- INSPEÇÃO DE VARIÁVEIS HISTÓRICAS. ....	88
FIGURA 5.14- <i>SHUTDOWN</i> DA FERRAMENTA C&M.....	89

FIGURA 6.1- MODELO HIERÁRQUICO DE AUTOMAÇÃO DE BALBINA E SAMUEL (ÁLVARES, 2008).....	90
FIGURA 6.2- INSTRUMENTOS E EQUIPAMENTOS. ....	92
FIGURA 6.3- REQUISITOS FÍSICOS DO SIMPREBAL.....	93
FIGURA 6.4- REGRAS DE PRODUÇÃO.....	96
FIGURA 6.5- CLASSES DO I-KERNEL .....	101
FIGURA 6.6- CLASSES DO <i>CONFMONITTOOL</i> .....	102
FIGURA 7.1- VISTA AÉREA DA USINA HIDRELÉTRICA DE BALBINA.....	104
FIGURA 7.2- VISTA SUPERIOR DOS 5 GERADORES. ....	105
FIGURA 7.3- ROTOR DO GERADOR ELÉTRICO. ....	106
FIGURA 7.4- ESTATOR DO GERADOR ELÉTRICO.....	106
FIGURA 7.5- HISTÓRICO DE ANOMALIAS OCORRIDAS. ....	109
FIGURA 7.6- ANOMALIAS DETECTADAS NOS SUBSISTEMAS DO GERADOR ELÉTRICO DAS UGHs.....	109
FIGURA 7.7- ANOMALIAS DETECTADAS NO SRG DAS CINCO UGHs.....	110
FIGURA 7.8- ANOMALIAS DETECTADAS NO SRG02.....	111
FIGURA 7.9- INSPEÇÃO DE VARIÁVEIS DOS SRG02 E SRG03.....	112
FIGURA 7.10- GRÁFICO DE TENDÊNCIAS DE VARIÁVEIS EM TEMPO REAL.....	112
FIGURA 7.11- GRÁFICO DE TENDÊNCIAS DE VARIÁVEIS HISTÓRICAS. ....	113
FIGURA 7.12- GRÁFICO DAS VARIÁVEIS ASSOCIADAS ÀS ANOMALIAS DETECTADAS NO SRG02. ....	113
FIGURA 7.13- ANOMALIAS DETECTADAS NO SRG03.....	114
FIGURA 7.14- GRÁFICO DAS VARIÁVEIS ASSOCIADAS ÀS ANOMALIAS DETECTADAS NO SRG03. ....	114
FIGURA 7.15- GRÁFICO DAS VARIÁVEIS ASSOCIADAS ÀS ANOMALIAS DETECTADAS NO SRG02. ....	115
FIGURA 7.16- GRÁFICO DAS VARIÁVEIS ASSOCIADAS ÀS ANOMALIAS DETECTADAS NO SRG02. ....	115
FIGURA 7.17- EMAIL RECEBIDO DO SIMPREBAL. ....	116
FIGURA 7.18- TELA PARA AVALIAÇÃO DAS TOMADAS DE DECISÃO. ....	117
FIGURA A.1-ABORDAGENS DE TÉCNICAS DE PROGNÓSTICO (BYINGTON <i>ET AL.</i> 2002, MODIFICADO).....	136
FIGURA A.2- CLASSIFICAÇÃO DOS ALGORITMOS DE PROGNÓSTICOS .....	137

FIGURA A.3- PROGNÓSTICO BASEADO EM CARACTERÍSTICAS (ROEMER <i>ET AL.</i> 2005, MODIFICADO).....	139
FIGURA A.4- PROGNÓSTICO BASEADO EM DADOS (ROEMER <i>ET AL.</i> 2005, MODIFICADO). ..	140
FIGURA A.5- PROGNÓSTICO BASEADO EM MODELOS FÍSICOS (ROEMER <i>ET AL.</i> 2005, MODIFICADO).....	141
FIGURA A.6- CONCEITO DE PROGNÓSTICO ADAPTATIVO (ROEMER <i>ET AL.</i> 2005, MODIFICADO).....	142
FIGURA B.1- PARÂMETROS DE CONFIGURAÇÃO GERAL. ....	145
FIGURA B.2- PARÂMETROS DE CONFIGURAÇÃO DA REDE FAM.....	145
FIGURA B.3- PARÂMETROS DO SERVIDOR OPC E TAGS ASSOCIADAS. ....	146
FIGURA B.4- PARÂMETROS DO SERVIDOR BANCO DE DADOS E TAGS ASSOCIADAS.....	147
FIGURA B.5- TAGS SIMULADOS. ....	147
FIGURA B.6- PARÂMETROS DOS DISPOSITIVOS DFI. ....	148
FIGURA B.7- PARÂMETROS DO SERVIDOR DE EMAIL E LISTA DE EMAILS POR GRUPOS. ....	148
FIGURA B.8- ARQUIVO CÓDIGOS DE FALHA. ....	149
FIGURA B.9- ARQUIVO CÓDIGOS DE DECISÃO.....	150
FIGURA C.1- PROCESSO DE OBTENÇÃO DA CLASSE TAG.....	151
FIGURA C.2- ESTRUTURA DA PROPRIEDADE <i>VALUE</i> DO ITEM <i>STATUS</i> (SMAR, 2005). ....	152
FIGURA E.1- SERVIDOR SIMPREBAL EM EXECUÇÃO.....	160
FIGURA E.2- TELA DE <i>LOGIN</i> .....	161
FIGURA E.3- TELA INICIAL. ....	161
FIGURA E.4- HISTÓRICOS DE ANOMALIAS, SELEÇÃO DE EQUIPAMENTOS. ....	162
FIGURA E.5- HISTÓRICOS DE ANOMALIAS. ....	163
FIGURA E.6- EDIÇÃO DA DATA DE TÉRMINO DE UMA ANOMALIA.....	164
FIGURA E.7- CÁLCULO DOS KPIs.....	167
FIGURA E.8- SINÓTICO SIMPREBAL. ....	168
FIGURA E.9- LINK PARA DETALHAMENTO DA ANOMALIA. ....	169
FIGURA E.10- INSPEÇÃO DE VARIÁVEIS. ....	170
FIGURA E.11- MENU DA TELA DE INSPEÇÃO DE VARIÁVEIS. ....	170
FIGURA E.12- GRÁFICO EM TEMPO REAL DA TEMPERATURA DE AR QUENTE DO RADIADOR DAS 5 UNIDADES GERADORAS. ....	171
FIGURA E.13- SELEÇÃO DO INTERVALO DE AQUISIÇÃO DOS DADOS HISTÓRICOS. ....	171
FIGURA E.14- GRÁFICOS HISTÓRICOS.....	172

## LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

ABNT	– Associação Brasileira de Normas Técnicas.
AI	– Analog Input.
AO	– Analog Output.
ART	– Adaptive Resonance Theory.
CLP	– Controlador Lógico Programável.
COM	– Component Object Model.
CORBA	– Common Object Request Broker Architecture.
CRIS	– Common Relational Information Schema.
DA	– Data Access.
DCOM	– Distributed Component Object Model.
DCS	– Distributed Control System.
DDE	– Dynamic Data Exchange.
DFI	– Interface <i>fieldbus</i> Distribuída.
DLL	– Dynamic Link Library.
ERP	– Enterprise Resource Planning.
FMEA	– Failure Modes and Effects Analysis.
FAM	– <i>Fuzzy</i> ARTMAP.
FB	– Function Block.
FF	– Foundation Fieldbus.
FTA	– Fault Tree Analysis.
GUI	– Gráfico User Interface.
HSE	– High Speed Ethernet.
HTML	– HyperText Markup Language.
HTTP	– Hyper Text Transfer Protocol.
HMI	– Human Machine Interface
IA	– Inteligência Artificial .
IDL	– Interface Definition Language.
IEC	– International Electrotechnical Commission.
IEEE	– Institute of Electrical and Electronics Engineers.
IP	– Internet Protocol.
ISA	– Instrumentation, System and Automation Society.
ISO	– International Standard Organisation.

ISP	– Interoperable System Project.
JDBC	– Java Database Connectivity.
JESS	– Java Expert System Shell.
JNI	– Java Native Interface.
LAN	– Local Area Network.
MBP	– Manchester Bus-Powered.
MC	– Manutenção Corretiva.
MCC	– Manutenção Centrada em Confiabilidade.
MES	– Manufacturing Execution Systems.
MIMOSA	– Machinery Information Management Open Systems Alliance.
MP	– Manutenção Preventiva.
MPd	– Manutenção Preditiva.
MTBR	– Mean Time Between Repairs.
MTTR	– Mean Time To Repairs.
NPR	– Número de Prioridade de Risco.
OLE	– Object Linking and Embedding.
OMG	– Object Management Group.
OPC	– OLE for Process Control.
OS	– Ordem de serviço.
OSA-EAI	– Open System Architecture for Enterprise Application Integration.
OSA-CBM	– Open System Architecture for Condition Based Maintenance.
PHP	– Hypertext Preprocessor
RNA	– Redes Neurais Artificiais.
RPC	– Remote Procedure Call.
SCADA	– Supervisory Control and Data Acquisition.
SE	– Sistema Especialista.
SIMPREBAL	– Sistema Inteligente de Manutenção Preditiva de Balbina.
SOAP	– Simple Object Access Protocol.
SRG	– Subsistema de Resfriamento do Gerador.
TCP/IP	– Transmission Control Protocol / Internet Protocol.
TELNET	– TELecommunication NETwork.
TPM	– Total Productive Maintenance.
UGH	– Unidade Geradora Hidráulica.
UML	– Unified Modeling Language.

# 1 INTRODUÇÃO

A atual competitividade no mercado de consumo exige das empresas a oferta de produtos e serviços de qualidade e baixo custo com o objetivo de se obter uma favorável posição neste mercado. A este cenário adiciona-se a impressionante evolução tecnológica, o movimento inexorável da globalização e o seu potencial incentivo à competitividade das organizações. Está assim instituído o quadro de mudanças que a sociedade industrial atualmente vivencia (Nunes, 2001).

As empresas têm ampliado sobremaneira o uso de novas tecnologias. Em vista disso, a modernização dos equipamentos, a automação dos sistemas e processos, a diversidade e a quantidade de componentes e acessórios utilizados crescentemente nas instalações industriais tendem a favorecer o aumento da probabilidade de ocorrência de falhas<sup>1</sup>. Nesse sentido, pode-se afirmar que, em todos os segmentos industriais, os períodos de indisponibilidade dos equipamentos afetam a capacidade produtiva das empresas, aumentando os custos operacionais e, em consequência, interferem na qualidade do produto final e no atendimento aos clientes.

As falhas em equipamentos podem representar grandes perdas econômicas e humanas, apresentando, em muitos casos, comprometimentos significativos para a imagem institucional das empresas. Essas ocorrências confirmam a relevância, nos dias de hoje, de se inovar nas estratégias de manutenção.

Na área de manutenção, a maioria das manutenções de equipamentos é ainda reativa (reparando e substituindo depois da falha) ou escassamente proativa (assumindo certo nível de degradação sem sinais do próprio equipamento, e consertando o equipamento em um horário de serviço sem saber se precisa de fato ou não). Ambos cenários são ineficientes e resultam em alto custo de produção ou tempos de paradas de máquinas (Djurdjanovica *et al.*, 2003).

---

<sup>1</sup> Falha é definida por Britto (2006) como a inabilidade de um sistema físico em realizar uma função no nível de desempenho e é identificada em geral, pela negação da função ou parte dela. Pode se dividir em dois tipos: (a) Evidente: constitui na perda da função que será percebida cedo ou tarde pelo usuário e (b) Oculta: se refere à perda da função que só será percebida se outra falha funcional ocorrer primeiro.

Dever-se-ia optar por outras estratégias de manutenção como a manutenção preventiva, que segundo Sim e Endrenyi (1988) é definida como: “uma atividade empreendida regularmente a intervalos de tempo definidos enquanto o dispositivo está operando satisfatoriamente, para reduzir ou eliminar a deterioração acumulada”. Geralmente existem dois tipos de manutenção preventiva, baseados em condição e baseados no tempo (Legat *et al.*, 1996).

Na manutenção preventiva baseada em condição, a ação tomada depois de cada inspeção é dependente do estado do sistema. Pode ser desde nenhuma ação, ou uma mínima manutenção para recuperar o sistema à fase prévia da degradação, ou uma manutenção geral para levar ao sistema para um estado tão bom quanto novo. A manutenção preventiva baseada no tempo é feita em intervalos de tempo definidos levando o sistema para um estado tão bom quanto novo (Vaurio, 1997).

Para nós, parece, freqüentemente que as máquinas falham de repente, mas na realidade as máquinas normalmente passam por um processo mensurável de degradação antes de falharem. Essa degradação é invisível para os operadores, embora muitas tecnologias tenham sido desenvolvidas para fazer tais informações visíveis.

As informações fornecidas pela intensa instrumentação dos equipamentos podem ser centralizadas por sistemas supervisórios ou bancos de dados em tempo real com taxas variáveis de amostragem, podendo ser utilizadas por sistemas baseados em conhecimento, como SE (Sistemas Especialistas), redes neurais, lógica nebulosa, raciocínio baseado em casos, sistemas inteligentes híbridos, etc.

Uma das tecnologias mais promissoras na implementação de MBC (Manutenção Baseada em Condição) é a aplicação de SE, uma das frentes da abordagem de IA para o desenvolvimento de sistemas de diagnóstico é o uso de lógica nebulosa e redes neurais para prognóstico de defeitos<sup>1</sup> e falhas. Os problemas de diagnóstico e/ou prognóstico de defeitos e falhas em plantas industriais envolvem correlações entre várias variáveis associadas ao processo.

---

<sup>1</sup> Defeito, corresponde a um evento em evolução, caracterizado por um desvio de uma condição assumida inicialmente como normal para item sob avaliação, para o instante de tempo considerado na análise (Dupont, 2003).

As variáveis devem ser acompanhadas temporalmente para análises de tendência e comparação com o desempenho esperado. Além da coleta e armazenamento de dados, exige-se um modelo amplo e hierarquizado do sistema analisado que represente os componentes e subsistemas da planta em termos de relevância e relação funcional entre si.

Na prática, os defeitos e falhas podem até serem detectados pelos próprios subsistemas computacionais ou microeletrônicos instalados nos equipamentos diretamente pelo fabricante, conectados a alguma ferramenta centralizadora de dados e informações (e.g., sistema supervísório SCADA (*Supervisory Control and Data Acquisition*), CLP (Controlador Lógico Programável)). No entanto, geralmente é necessária a intervenção de um operador experiente ou especialista para interpretar os sinais de alerta ou alarmes emitidos por essas ferramentas em relação ao equipamento como um todo, a fim de tomar as ações necessárias em resposta a uma determinada situação de falha iminente ou pane.

A forma de análise de um especialista humano que resolve problemas associados a defeitos e falhas de plantas industriais é geralmente baseada em informações aparentemente isoladas, mas conscientemente associadas entre si. Através de algum tipo de conhecimento advindo de sua experiência de campo acumulada com o enfrentamento de algumas dezenas de casos que se repetem diante de certas condições ou situações que passam a ser facilmente identificadas pelo especialista. Esse tipo de conhecimento é chamado heurístico, de natureza prática e normalmente adquirido implicitamente e processado pelos sistemas baseados em conhecimento que aplicam técnicas de IA. Estas técnicas são capazes de manipular conhecimento e gerar análises complexas similares às dos especialistas humanos.

## **1.1 OBJETIVOS**

### **1.1.1 Objetivos Gerais**

O objetivo principal deste trabalho é propor uma metodologia para concepção de um sistema de manutenção baseada em condição, usando sistemas especialistas e sistemas inteligentes híbridos<sup>1</sup> *neuro-fuzzy*.

---

<sup>1</sup> Os sistemas inteligentes híbridos são obtidos da combinação de duas ou mais técnicas de inteligência artificial, de maneira que superam as limitações das técnicas individuais.



Implementação computacional de um sistema inteligente de manutenção preditiva de Balbina (SIMPREBAL) usando sistemas especialistas com o intuito de auxiliar às equipes de operação e manutenção na tomada de decisão.

### 1.1.2 Objetivos Específicos

- a) Apresentar os conceitos de técnicas de manutenção e arquiteturas usadas, que justificam, auxiliam e contextualizam a aplicação de ferramentas inteligentes no desenvolvimento de sistemas de manutenção baseada em condição;
- b) Apresentar as principais abordagens de algoritmos de prognósticos baseada em técnicas de inteligência artificial;
- c) Estabelecer uma metodologia de trabalho baseado no modelo de referência de sete camadas OSA-CBM;
- d) Apresentar uma metodologia para prognósticos de falhas usando um sistema inteligente híbrido *fuzzy* ARTMAP;
- e) Desenvolver um protótipo de sistema especialista baseado em regras de produção para apoio à manutenção da usina hidrelétrica Balbina;
- f) Implementação computacional do SIMPREBAL (Sistema Inteligente de Manutenção Preditiva de Balbina), baseado no modelo OSA-CBM e na arquitetura cliente/servidor.
- g) Apresentar resultados qualitativos e quantitativos coerentes com o nível do conhecimento disponibilizado e implementado na base de regras do sistema especialista. As análises são feitas a partir das ocorrências de anomalias detectadas pelo SIMPREBAL e o histórico das variáveis associadas, armazenados no banco de dados.

## 1.2 FORMULAÇÃO DO PROBLEMA

A principal motivação para a realização deste trabalho surgiu no contexto do projeto de pesquisa ANEEL-Eletronorte visando à modernização da área de automação das usinas hidrelétricas de Balbina e Samuel. A necessidade de definir uma metodologia para um sistema inteligente de manutenção baseada em condição para geração de diagnósticos e prognósticos capazes de auxiliar o pessoal de operação e manutenção da usina hidrelétrica na tomada de decisão, e a implementação de um sistema computacional modelo cliente/servidor. Para atender estas demandas propõe-se uma metodologia baseada no

modelo de referência de sete camadas OSA-CBM utilizando técnicas inteligentes como sistemas especialistas e o sistema inteligente híbrido *fuzzy* ARTMAP para diagnóstico e prognóstico de anomalias. A geração de sugestões das ações de manutenção e operação, e o desenvolvimento computacional baseado em sistemas especialistas para o processamento inteligente usando uma arquitetura cliente/servidor, desenvolvendo no servidor, a aquisição, o processamento inteligente e a tomada de decisão; e no cliente, a interface com o usuário onde são apresentados os dados gerados pelo servidor.

### **1.3 ESTRUTURA DO DOCUMENTO**

O presente documento encontra-se estruturado em oito capítulos.

O capítulo um é a introdução ao trabalho. No capítulo dois é apresentada a revisão de literatura sobre Manutenção Baseada em Condição, arquiteturas para Sistemas de Manutenção Baseada em Condição e descrição das sete camadas do modelo de referência OSA-CBM. O capítulo três é dedicado a descrever as técnicas inteligentes, sistemas especialistas, lógica nebulosa, redes neurais artificiais, sistemas híbridos e a aplicação destas técnicas na Manutenção Baseada em Condição. O capítulo quatro apresenta a metodologia proposta, a aplicação do modelo de referência e modelagem IDEF0 e IDEF1X. O capítulo cinco apresenta a modelagem UML do sistema computacional a ser desenvolvido, os requisitos de usuário e os requisitos do sistema representado em casos de uso. O capítulo seis apresenta a implementação computacional do SIMPREBAL, os modelos de regras de produção implementadas nas camadas OSA-CBM e a descrição da ferramenta de configuração e monitoramento. O capítulo sete é dedicado ao estudo de caso: sistema do gerador elétrico da usina hidrelétrica de Balbina, resultados obtidos analisando as anomalias e as variáveis associadas armazenadas no banco de dados. Finalmente, no capítulo oito apresentam-se as conclusões do trabalho, contribuições e sugestões para futuras pesquisas.

O apêndice A descreve as abordagens de algoritmos aplicados em prognósticos, bem como as vantagens e desvantagens de cada um dos métodos. O apêndice B apresenta os arquivos de configuração, de códigos de falha e de códigos de decisão usados pelo SIMPREBAL; no apêndice C mostram-se o processo de cálculo para obter os campos da classe Tag; no apêndice D são apresentados os métodos de análises FMEA, FTA e a comparação entre as duas; e no apêndice E os procedimentos de iniciação e operação do SIMPREBAL.

## **2 REVISÃO BIBLIOGRÁFICA: MANUTENÇÃO BASEADA EM CONDIÇÃO**

Este capítulo apresenta uma revisão da literatura relativa à manutenção baseada em condição. São apresentados os conceitos, sistemas e as arquiteturas usadas para o desenvolvimento de um sistema de manutenção baseada em condição, o modelo de referência OSA-CBM e a descrição das suas sete camadas.

### **2.1 INTRODUÇÃO**

Para Moubray (1997), o processo de gerenciamento da manutenção, talvez mais do que qualquer outra atividade de gerenciamento, sofreu no decorrer de sua evolução, principalmente nos últimos trinta anos, importantes transformações em seus métodos. As mudanças ocorridas nesse período, seja pelo crescimento das expectativas de manutenção ou pelas mudanças de visão sobre o modo de ocorrência de defeitos e falhas ou das técnicas de manutenção, podem ser caracterizadas por três gerações distintas, todas, como sempre, fruto da necessidade de racionalização e otimização imposta por períodos de crise (Arcuri Filho, 1996).

A primeira geração, conservação, segundo Ariza (1988) começa a se destacar no século XVI, por ocasião da invenção das primeiras máquinas têxteis, movidas a vapor, perdurando até a segunda guerra mundial. A indústria, nesse período, caracterizava-se como pouco mecanizada, os equipamentos eram simples e de fácil conserto, além de o volume de produção não ser prioritário, em razão da conjuntura econômica da época (Kardec e Nascif, 1999). Nesse contexto, as condições eram propícias para a adoção da forma mais elementar de manutenção, a manutenção não-planejada, caracterizada pela atuação somente após a ocorrência da falha, ou seja, manutenção corretiva (MC).

O período pós-guerra trouxe consigo a segunda geração da manutenção. Nessa fase, a tolerância com atrasos diminuiu e a exigência de produtividade aumentou, em razão, sobretudo, das pressões originadas da guerra (Moubray, 1997). Como consequência, houve um forte aumento da mecanização das indústrias e os equipamentos, de simples e robustos, passaram a complicados, exigindo uma metodologia de manutenção mais apurada (Lucatelli, 1998). Então, começou a evidenciar-se a necessidade de maior disponibilidade, bem como de maior confiabilidade de equipamentos, a fim de se garantir maior

produtividade (Kardec e Nascif, 1999). Nesse cenário, surgiu a idéia de que as falhas poderiam e deveriam ser prevenidas, o que resultou no conceito da manutenção preventiva (MP), caracterizada pela substituição sistemática de itens com base em intervalos ou ciclos predeterminados (Lucatelli, 1998).

A terceira geração da manutenção iniciou-se segundo Lucatelli (2002) na década de 1970, motivada pelo processo de mudanças ocorrido nessa época nas indústrias. Segundo Moubray (1997), tais transformações podem ser classificadas em três áreas principais, quais sejam: (a) a expectativa de crescimento da função manutenção; (b) o melhor entendimento do modo como o equipamento falha e (c) o aumento da gama de técnicas e ferramentas de gerenciamento da manutenção (Dunn, 1998). Essa inevitável evolução deu-se sobretudo pelas novas exigências de mercado, que determinaram, em virtude da globalização e da concorrência internacional, a necessidade de redução de custos operacionais.

Ainda na década de 1970, a adoção da sistemática *just in time* tornou-se uma tendência mundial, trazendo a idéia de que mesmo pequenas pausas de produção poderiam comprometer o atendimento da demanda em razão dos baixos estoques mantidos (Kardec e Nascif, 1999). Outros fatos que contribuíram para tal evolução foram o fenômeno da automação das indústrias e o advento da informática, tornando-as extremamente complexas e, por conseqüência, transformando a confiabilidade e a disponibilidade dos equipamentos em pontos-chave em setores tão distintos como o são a saúde, as telecomunicações e o gerenciamento de edificações (Kardec e Nascif, 1999).

A Figura 2.1 ilustra graficamente a evolução das três gerações da manutenção e permite verificar o aumento na demanda pelos sistemas de manutenção com relação às exigências organizacionais. Substituiu-se o antigo conceito de substituição após avaria por um conjunto de requisitos que incluem desde a disponibilidade e confiabilidade das máquinas ao cuidado com o impacto no meio-ambiente.

As escalas crescentes de exigência também impactam em maior demanda pelo conhecimento na atividade de manutenção. A Figura 2.2 representa este fato, com o aumento no número de indicadores e análise referentes à atividade de manutenção. Mostra, também, conforme análise de Moubray (1997), na primeira geração a concepção de falha era simplesmente de que os itens mais velhos tinham mais probabilidade de falhar. Na

segunda geração as falhas ocorridas na instalação de um equipamento levaram a crença generalizada da segunda geração na curva “da banheira”. Entretanto, a pesquisa da terceira geração revelou que não apenas um ou dois, porém seis padrões de falha ocorrem realmente na prática.

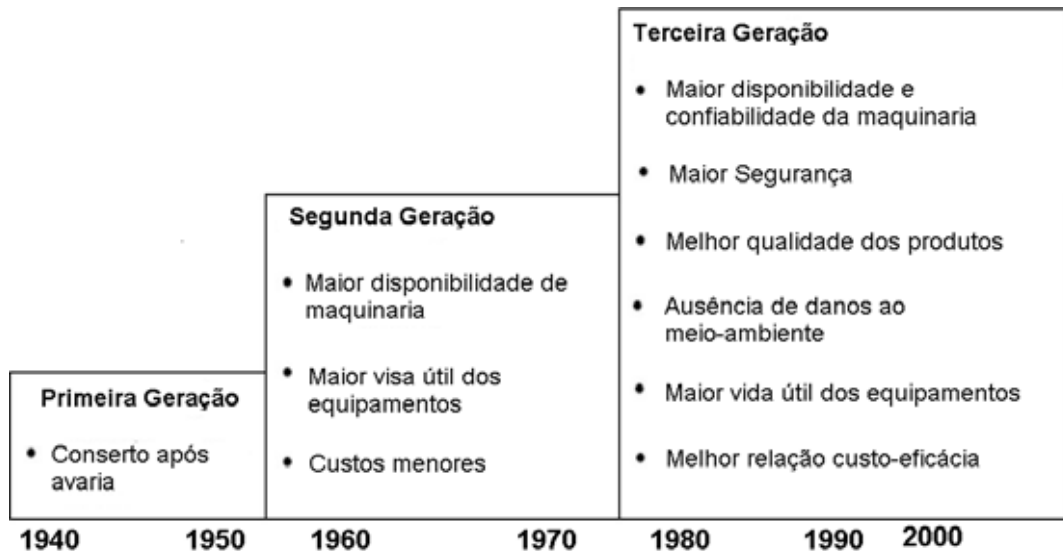


Figura 2.1- Crescimento das expectativas de manutenção (Moubray, 1997).

Na Figura 2.3 ilustra-se o impacto das demandas nas políticas de manutenção. Nota-se que na terceira geração inclui monitoração de condições, análise de risco, emprego intensivo da tecnologia da informação e de profissionais versáteis, todos os fatores de impacto ao emprego de conhecimento intensivo.

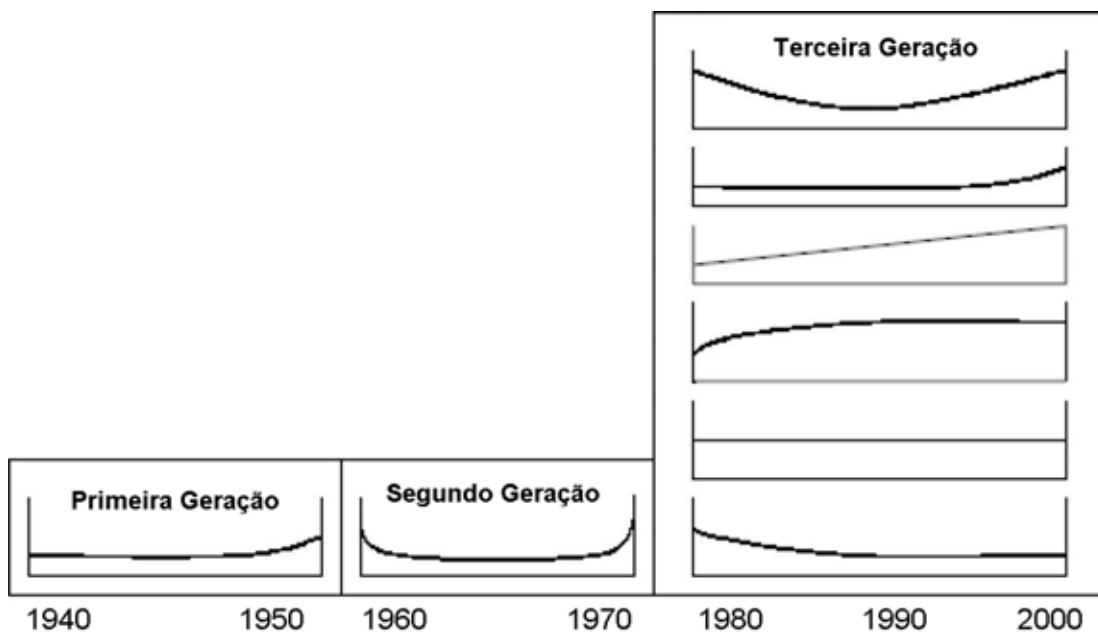


Figura 2.2- Mudanças de visão na falha do equipamento (Moubray, 1997).

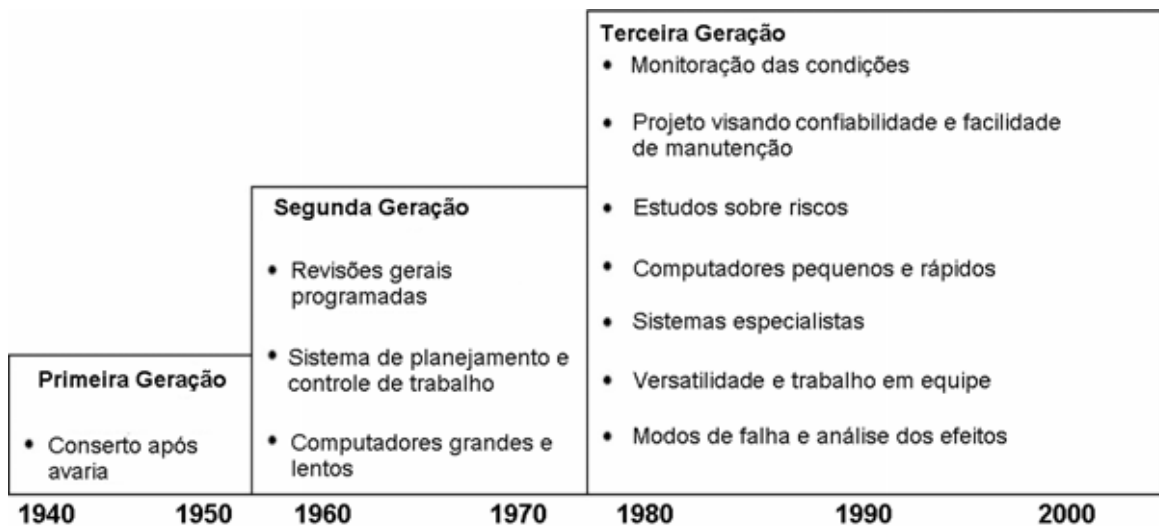


Figura 2.3- Mudança das técnicas de manutenção (Moubray, 1997).

Além dessas três gerações, Dunn (1998) considera uma quarta geração da manutenção, caracterizada por uma visão mais holística dos recursos, ainda baseada nas três gerações anteriores, a quarta geração integra todas as ferramentas de projeto e de manutenção, preconizando os seguintes aspectos (Dunn, 1998): (a) uma abordagem formal para a taxa de risco, particularmente em níveis mais altos da organização, a qual trata dos projetos de equipamentos e estratégias de manutenção; (b) princípios da MCC (Manutenção Centrada em Confiabilidade) e Manutenção Produtiva Total (TPM – *Total Productive Maintenance*) enfocando uma maior integração entre as exigências funcionais, projeto dos equipamentos e da manutenção; (c) fatores humanos, aplicados à operação e à manutenção do equipamento; (d) maior uso de tecnologias de informação para detectar, predizer e diagnosticar as falhas dos equipamentos.

Na geração atual aparecem as maiores contribuições relacionadas às metodologias de gestão da manutenção, abrangendo desde o surgimento das primeiras técnicas de monitoração de condição (MPd – manutenção preditiva), a utilização de ferramentas de auxílio à decisão e a análise de risco; o surgimento do método de análise dos modos de falha e seus efeitos (FMEA – *Failure Modes and Effects Analysis*), sistemas especialistas, redes neurais, lógica nebulosa e outras técnicas de inteligência artificial; a maior atenção na fase de projeto a aspectos de confiabilidade e manutenibilidade, até a criação de grupos de trabalho multidisciplinares, com o envolvimento de todos os níveis hierárquicos da companhia, para o estabelecimento de metodologias mais eficientes no gerenciamento de ativos, tais como a TPM e a MCC (Moubray, 1997).

O termo manutenção é definido pelo padrão SS-EN-13306 (2001) como: "combinação de toda técnica, administrativa, e ações gerenciais durante o ciclo de vida de um item cujo objetivo é manter, restaurar, o estado na qual possa executar uma função requerida" (Bengtsson, 2004b). No Brasil a ABNT no padrão NBR-5462 (1994) define manutenção como “a combinação de todas as ações técnicas e administrativas, incluindo as de supervisão, destinadas a manter ou recolocar um item em um estado no qual possa desempenhar uma função requerida”.

O termo “recolocar” tem uma conotação de “correção” a uma perda de função e o termo item como “qualquer parte, conjunto, dispositivo, subsistema, unidade funcional, equipamento ou sistema que possa ser considerado individualmente”. Com foco na definição de "manter ou recolocar" é claro que existe dois tipos (estratégias) principais na execução da manutenção mostrados na Figura 2.4. O primeiro é a abordagem preventiva, que é a manutenção efetuada em intervalos predeterminados, ou de acordo com critérios prescritos, destinada a reduzir a probabilidade de falha ou a degradação do funcionamento de um item (NBR-5462, 1994). A segunda, a abordagem corretiva, que é a manutenção efetuada após a ocorrência de uma falha de modo a recolocar um item em condições de executar uma função requerida (NBR-5462, 1994).

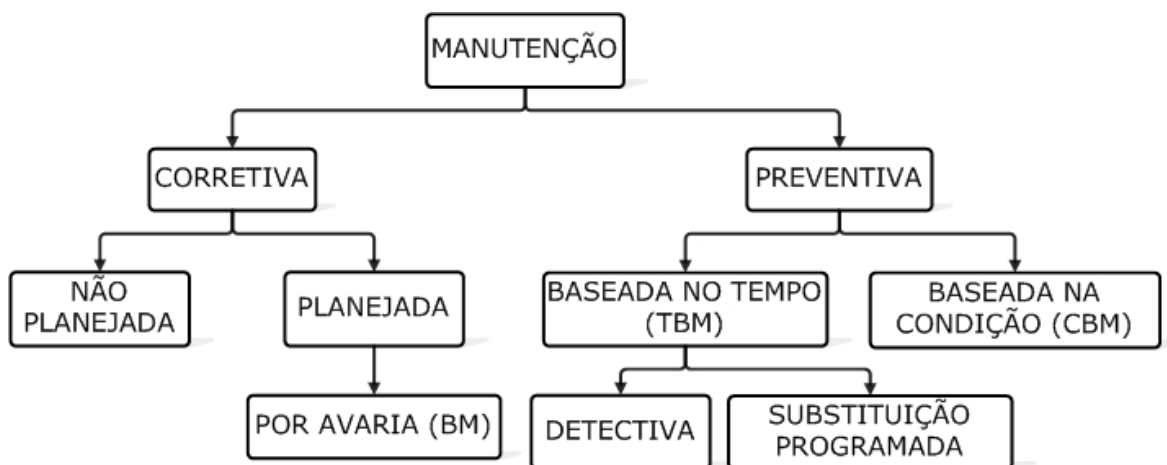


Figura 2.4- Diferentes tipos de manutenção (Barroso Maia Junior, 2003).

A globalização instalada nos mercados torna mais acirrada a concorrência, que passa a exigir das empresas um desempenho qualitativo técnico de classe mundial. O aumento substancial dos custos de manutenção e o desenvolvimento de equipamentos cada vez mais complexos têm induzido a opção por outras estratégias e tecnologias na área da manutenção. Sujeita a estas circunstâncias, da manutenção baseada em sensoriamento e

avaliação do estado atual do sistema surge uma apropriada e eficiente ferramenta para diminuir o tempo de parada devido à falha da máquina (NSF, 2008). Segundo Bengtsson (2003), com um sistema de MBC bem implementado, uma companhia na Suíça pôde economizar até 20% dos gastos em manutenção, melhoramento da qualidade, e diminuição do estoque de peças sobressalentes, etc.

Para Kothamasua e Huang (2007), a tecnologia da manutenção progrediu desde a baseada no tempo até a baseada em condição. Há bem pouco tempo a estratégia de manutenção baseada no tempo era adotada pela grande maioria de empresas. Esta estratégia consiste num plano de inspeções pré-estabelecido, seja pelo fabricante ou pelos técnicos de manutenção. A idéia da manutenção baseada em condição é monitorar um equipamento usando vários sensores e efetuar diagnósticos e prognósticos de falhas iminentes do equipamento em tempo real (Kothamasua e Huang, 2007).

Com o desenvolvimento dos microcomputadores a custos reduzidos, linguagem simples e as exigências do aumento de qualidade dos produtos e serviços pelos consumidores, os órgãos de manutenção tiveram a opção de se desenvolver e processar seus próprios programas, eliminando os inconvenientes da dependência de disponibilidade humana e de equipamentos, gerando enorme profusão de *software* e o aparecimento e desenvolvimento de empresas especializadas em *software* para manutenção. Podem ser identificadas três linhas de convergência de *softwares* de manutenção, baseado nas classificações de Lacerda e Júnior (1997):

- *Softwares* de gestão: com módulos para gerenciamento de mão-de-obra, materiais, controle de custos, emissão de relatórios gerenciais e outras facilidades de tomadas de decisão. Enquadram-se na linha de gestão empresarial de ativos (*EAM – Enterprise Asset Management*) e gestão da manutenção (*CMMS - Computerized Maintenance Management Software*).
- *Softwares* específicos ou especializados: enquadram-se neste bloco, *softwares* de manutenção específica por equipamento, por fabricante, normalmente envolvendo diagnósticos. Engloba também os sistemas especialistas em franca ascendência, bem como *softwares* específicos empregando outras técnicas de inteligência artificial.
- *Softwares* de apoio: enquadram-se neste bloco todos os outros *softwares* que não forem de gestão ou específicos.



O *software* desenvolvido neste trabalho enquadra-se dentro dos *softwares* de apoio, porque é uma aplicação que não é específica de um equipamento nem depende do fabricante, o único requisito é ter disponível os dados da instrumentação através de um servidor OPC. Este *software* não chegar ainda ser de gestão porque não integra informações das várias áreas de empresa. O *software* é uma ferramenta de apoio que sugere ações de manutenção, dependera do operador adotar ou não as sugestões.

Atualmente existe uma ampla gama de *softwares* de manutenção sendo comercializados, oferecendo soluções em função do produto, tecnologia, mercado e estratégia das diversas empresas. Segundo Tavares e Filho (2002), o mercado de *software* de manutenção representou, em 1997, mais de 900 milhões de dólares de faturamento, dos quais 56.6% na América do Norte, 27.5% na Europa, 10.3% na Ásia e Oceania e 5.7% na América Latina.

Os *softwares* CMMS mais comumente usados são: MÁXIMO (18%), SAP (13%), MP2 (13%) e o WOMANS (5,3%). O MP2 é o mais extensivamente usado em pequenas plantas, enquanto o SAP é largamente usado em grandes plantas (Alkaim, 2003). Além destes *softwares*, outros fabricantes desenvolveram sistemas baseados nas informações sobre a condição dos equipamentos. A Smar usando a tecnologia FF (*Foundation Fieldbus*) tornou possível através do sistema *AssetView*, acessar funções novas e valiosas, tais como diagnósticos e estatísticas de operação, identificação de equipamento, e histórico de calibração armazenados no próprio equipamento (Smar, 2008).

Tabela 2.1- *Softwares* mais difundidos (MyQ, 2000).

<b>FABRICANTE</b>	<b>SOFTWARE</b>
Datastream (SP)	MP5, MP2Enterprise, MP2Professional, Maintainit
Protam Eng. de Manutenção (SP)	Coswin
SAM SERVICE (SP)	MAC ACTIVE (FULL)
ASTREIN Informática (SP)	SIM
Maximiza Consultoria Sistema (SC)	Sadege
MiDS Sistemas (SP)	Máximo
SPES Eng. De Sistemas	SMI

Em 2001 a ABB trouxe o protocolo de comunicação FF, tecnologia que pode reduzir em até cinco vezes gastos de manutenção em empresas de processo, o protocolo incorpora módulo de manutenção preditiva (*Optimize-IT*) e possibilita a integração do sistema de

supervisão industrial a um sistema corporativo tipo ERP (*Enterprise Resource Planning*) (ABB, 2008). Na Tabela 2.1 apresentam-se alguns *softwares* especializados de gerência de manutenção mais difundidos.

## **2.2 MANUTENÇÃO BASEADA EM CONDIÇÃO**

A MBC (Manutenção Baseada em Condição) é uma estratégia que utiliza algumas técnicas de monitoramento das condições operativas de uma máquina, seus sistemas e componentes sem necessidades de indisponibilizar o equipamento. A análise destas condições determinará quando uma intervenção será realizada podendo ou não indisponibilizar a máquina (Pinto, 2003). Butcher (2000) define a MBC como: “ações de avaliação das condições de um equipamento baseadas em tempo real ou quase em tempo real, a qual é obtida de sensores embutidos, provas externas e medidas feitas por equipamentos portáteis”.

Para Mecabo (2007), a manutenção baseada em condição é um programa que recomenda decisões baseadas nas informações coletadas. Segundo Moya e Vera (2003) o propósito de um programa de manutenção baseada em condição é melhorar a confiabilidade e disponibilidade de um sistema, a qualidade do produto, a segurança, a programação das ações de manutenção, a redução direta dos custos de manutenção e consumo de energia, bem como propiciar facilidades na verificação dos requisitos e certificação do padrão ISO 9000 (*International Standard Organisation*).

Segundo Bengtsson (2004b) e o padrão SS-EN 13306:2001, a manutenção baseada em condição é definida como: “manutenção preventiva baseada no desempenho e/ou monitoramento de parâmetros e as ações subseqüentes”. O desempenho e monitoramento de parâmetros podem ser programados, requeridos, ou contínuos. A manutenção baseada em condição é desta forma uma tecnologia de manutenção que usa ferramentas de monitoramento de condição para analisar a condição atual de um componente e, através deste conhecimento, executar um apropriado programa de manutenção preventiva. Será preditiva desde que os intervalos de manutenção e as tarefas sejam baseados na condição do componente.

Manutenção preditiva (MPd) é definida por Bengtsson (2004b) e pelo padrão SS-EN 13306:2001, como: "Manutenção baseada em condição executada seguindo a previsão das

análises e avaliação dos parâmetros de degradação significantes do componente". Para a NBR-5462 (1994) a manutenção Preditiva é a manutenção que permite garantir uma qualidade de serviço desejada, com base na aplicação sistemática de técnicas de análise, utilizando-se de meios de supervisão centralizados ou de amostragens, para reduzir ao mínimo a manutenção preventiva e diminuir a manutenção corretiva. Manutenção desempenhada com base no acompanhamento ou monitoramento de determinados parâmetros do equipamento (vibração, temperatura, ruído). Os métodos e técnicas de como decidir a abordagem de manutenção mais apropriada e utilizar ou não manutenção baseada em condição são discutidos por Al-Najjar e Alsyouf (2003) e Starr (1997).

### 2.3 SISTEMAS DE MANUTENÇÃO BASEADA EM CONDIÇÃO

Um sistema de manutenção baseada em condição é definido por Bengtsson (2004b) como: "Um sistema que usa a manutenção baseada em condição para determinar e programar ações de manutenção preditiva automática ou em interação com outros sistemas ou operadores".

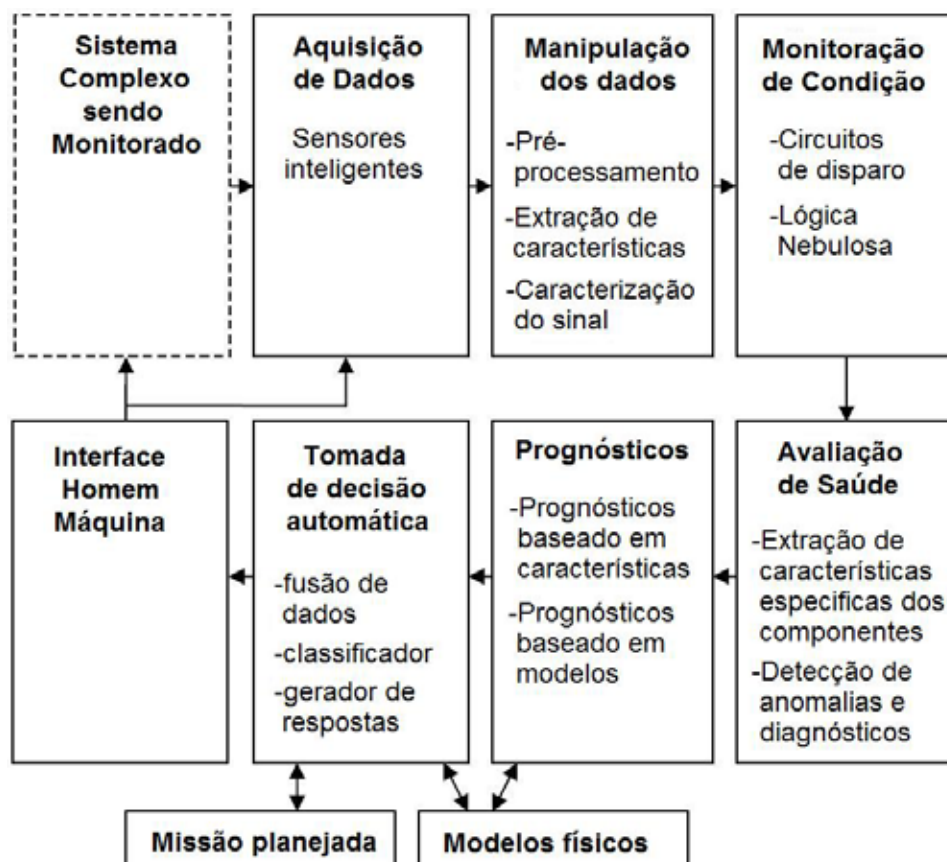


Figura 2.5- Camadas de um sistema de manutenção baseada em condição (Lebold *et al.* 2003, modificado).

Segundo Lebold *et al.* (2003) e Thurston (2001) um sistema de manutenção baseada em condição contém sete camadas ou atividades que corresponde às camadas do modelo OSA-CBM: aquisição de dados (sensor), processamento de sinal, monitoração de condição, avaliação de saúde (diagnóstico), prognóstico, tomada de decisão e apresentação, os quais podem ser visualizados na Figura 2.5.

O propósito de um sistema de manutenção baseada em condição é transformar certas grandezas das entradas bem definidas na forma de energia (vibração, temperatura, pressão, etc.) em efeitos desejados (informação da condição de um item, prognóstico da condição futura, etc.) no espaço e no tempo. O sistema de manutenção baseada em condição é assim um sistema que usa o desempenho e/ou técnicas de monitoramento de parâmetros (vibração, térmico, visual, etc.) para encontrar perturbações no desempenho das mudanças dos parâmetros característicos de um item.

Os sistemas de manutenção baseada em condição podem ter diferentes níveis de automação, desde, o executado totalmente por um usuário até o controlado totalmente por um sistema de *software e hardware* (Granell, 2007).

## **2.4 TECNOLOGIAS USADAS NOS SISTEMAS DE MBC**

Com a introdução da tecnologia de monitoramento de processos na década de 90, o operador passou a ser cada vez mais um observador do processo. Sua função atualmente é mais gerencial e neste ponto a utilização de ferramentas de apoio à decisão é crucial para o sucesso das atividades associadas à confiabilidade e a manutenibilidade de sistemas (Rigoni *et al.*, 2004), existindo porém a necessidade de automatizar os processos de manutenção com as tecnologias da informação emergentes na área de automação e sistemas de controle. A necessidade de monitorar as condições dos equipamentos através de sinais ou medidas das grandezas de processo. Isto gerou um aumento da demanda por sensores de alta precisão, alta confiabilidade, baixo custo e tamanhos compactos, nas últimas duas décadas.

### **2.4.1 Fieldbus**

Segundo Mahalik e Yen (2008), os barramentos de campo *fieldbus* constituem redes industriais e são definidos como um barramento (dois fios) digital, serial, multiponto e bi-

direcional, para a comunicação entre dispositivos de campo, como sensores, atuadores, reguladores, controladores e interfaces homem máquina (HMI – *Human Machine Interface*) em uma planta automatizada, permitem a implementação estratégias de controle distribuído, e são considerados como sofisticados, compactos e um método de comunicação digital avançada que minimiza o custo de cabeamento.

O custo é 50% menor para instalações de sistemas de controle distribuído (DCS - *Distributed Control System*), porque em muitos casos só precisa-se de um par de cabos para interconectar vários dispositivos (Armitage *et al.*, 1988). *Fieldbus* além de uma tecnologia para economizar o cabeamento, distribui os pontos de entrada/saída, possibilitando realizar o controle no local de aquisição de dados e nos pontos de ação dos processos, ou seja, nos sensores e atuadores. O *fieldbus* é uma rede local (LAN – *Local Area Network*) para automação e instrumentação de controle de processos, com capacidade de distribuir o controle no campo como mostrado na Figura 2.6.

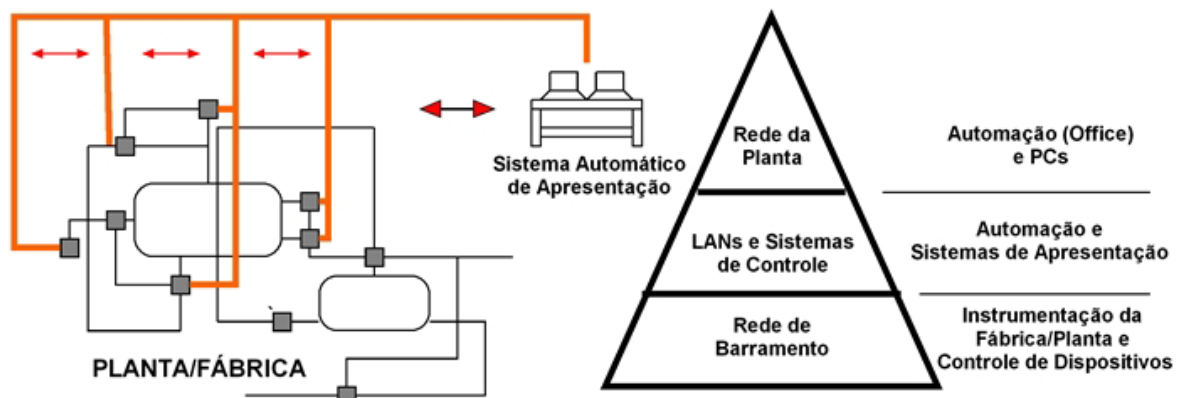


Figura 2.6- Rede *fieldbus* como uma rede local de instrumentos.

O *fieldbus* surgiu com o objetivo de interligar e operar instrumentos de campo com características diferentes e de diversos fabricantes, usufruindo de toda sua inteligência através de uma rede, proporcionando a descentralização de tarefas. Esta interligação incorpora vantagens como: maior imunidade a ruídos, pré-processamento de dados específicos, transmissão de informações adicionais de dados, capacitando o diagnóstico do dispositivo e a previsão de falhas, bem como a redução dos custos de projeto, de fiação, de instalação e de expansão. Essa filosofia objetiva grande redução no trabalho de projeto versatilidade na especificação de topologias de ligação, facilidade de instalação física do sistema, além de minimizar problemas de comunicação e falhas em equipamentos de controle (Rigoni *et al.*, 2004).

O incremento do uso dos sistemas *fieldbus* permite o intercâmbio de dados usando sistemas de comunicação moderna. As redes de área local (LAN - *Local Area Networks*), na maioria dos casos baseadas em ethernet e TCP/IP (*Transmission Control Protocol / Internet Protocol*), são usadas para interconectar diferentes sistemas *fieldbus*, além disto, o mapeamento dos componentes à LAN (Wolischlaeger *et al.*, 2004). A rede FF é uma rede digital cuja padronização levou mais de dez anos para ser concluída. A FF usa a IEC/ISA-S50.02-1992 como padrão (Fieldbuses, 2008).

Segundo Mahalik e Yen (2008), existem duas redes FF, uma de baixa velocidade concebida para interligação de instrumentos (H1 – 31.25 kbps) em dois fios chegando até 1.9km e outra de alta velocidade utilizada para integração das demais redes e para a ligação de dispositivos de alta velocidade como CLPs (HSE (*Hight Speed Ethernet*) - 100 Mbps). A interface entre estas duas redes é feita através de um dispositivo de enlace ou DFI (Interface *fieldbus* Distribuída). O DFI é o elemento chave de interface em um sistema de controle de campo, combinando recursos de comunicação, com acesso direto a entradas e saídas e controle avançado para aplicações contínuas e discretas. O DFI funciona como *bridge* H1-H1, H1-HSE ou H1-HSE-H1, e também como mestre dos barramentos H1, gerenciando a comunicação em cada canal. Estas duas redes H1 e HSE são mostradas na Figura 2.7.

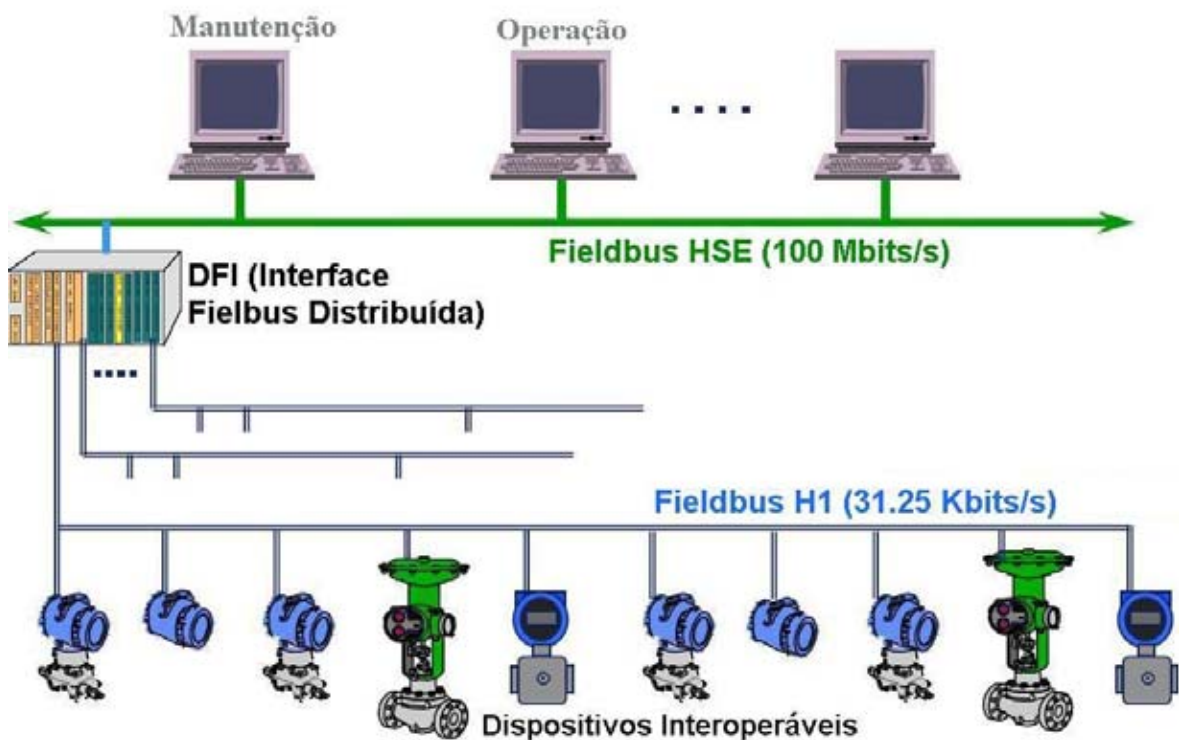


Figura 2.7- Redes *fieldbus* de baixa e alta velocidade, H1 e HSE

Ao contrário dos protocolos de rede proprietários, o *fieldbus* não pertence a nenhuma empresa, organismo ou nação. A tecnologia é controlada pela FF que é uma organização não lucrativa que consiste em mais de 140 dos principais fornecedores e usuários de controle e instrumentação do mundo (Mahalik, 2003). A FF mantém muitas das características operacionais do sistema analógico 4-20 mA, tais como uma interface física padronizada da fiação, os dispositivos alimentados por um único par de fios e as opções de segurança intrínseca, mas oferece uma série de benefícios adicionais aos usuários.

O sistema *fieldbus* permite interconectividade entre produtos de diferentes fabricantes, interoperabilidade entre estes dispositivos e permite que os mesmos possam ser trocados por dispositivos de outros fabricantes. A IEC 61158 (*International Electrotechnical Commission*) aceita os seguintes padrões: FF, *ControlNet*, *Profibus*, *P-NET*, *HSE*, *SwiftNet*, *WorldFIP* e *Interbus* mostrados na Tabela 2.2.

Tabela 2.2-Tipos de *Fieldbus* definidos no padrão IEC61158 (Hüsemann e Pereira, 2007).

Tipo	Descrição	Meio físico	Velocidade Max.	Meio de acesso
1	Foundation Fieldbus	IEC1158-2	31.25 kbps	Bus arbiter
2	ControlNet	Coaxial with transformer isolation	5 Mbps	CTDMA
3	Profibus DP/Profibus PA	RS485/IEC1158-2	12 Mbps/31.25 kbps	Token pass
4	P-Net	RS485	76.8 kbps	Virtual token
5	HSE	Ethernet	100 Mbps	CSMA/CD
6	SwiftNet	RS485	5 Mbps	TDMA
7	WorldFIP	IEC1158-2	31.25 kbps	Bus arbiter
8	Interbus-S	RS485	500 kbps	Message slot

Segundo Mahalik (2003), a FF foi estabelecida em 1994 pela fusão de *WorldFIP* e ISP (*Interoperable System Project*), caracterizando-se por ser um protocolo totalmente digital, serial, sistema de comunicação bidirecional para interconectar instrumentação de fabrica/planta e dispositivos de controle. Segundo Mahalik (2003), a especificação desta rede de campo é compatível com os padrões SP50 de instrumentação, ISA (*Instrumentation, System and Automation Society*) e a IEC. Isto permite uma distribuição confiável do controle nos dispositivos de campo através do uso de módulos *softwares* chamados blocos de funções (FBs – *Function Blocks*).

Cada bloco de função processa dados de entrada de acordo com um algoritmo específico e um conjunto interno de parâmetros de controle, produzindo dados de saída. Exemplos de blocos de funções são: entrada analógica (AI-*Analog Input*), saída analógica (AO – *Analog Output*), entrada digital (DI – *Digital Input*), controle PID (Proporcional Integral derivativo), etc. Os quais podem ser executados nos dispositivos de campo como sensores e atuadores. A especificação do meio físico é definida pelos padrões IEC1158-2, *Manchester Bus-Powered* (MBP), que adota protocolos como FF, *Profibus-PA*, e *WorldFIP*. A especificação da camada física é apropriada para aplicações de segurança intrínseca, o qual é comum nas indústrias de processo (Hüsemann e Pereira, 2007).

#### **2.4.2 Componentes COM/DCOM**

Nas comunicações entre dois programas numa rede, os programadores podem escolher a tecnologia de um *software* para conectar duas aplicações diferentes e separadas e usá-las para facilitar a interação entre essas duas aplicações através de uma rede, então o uso deste tipo de tecnologia leva a uma rápida e abstrata solução para aplicações cliente-servidor (Lebold *et al.*, 2003).

A solução para uma implementação satisfatória de uma arquitetura distribuída é o desenvolvimento de métodos padrões de intermediação e comunicação de objetos através de uma rede, essas tecnologias incluem, RPC (*Remote Procedure Call*) descrito por, *The Open Group* (2008), DCOM (*Distributed Component Object Model*) desenvolvido pela Microsoft segundo a MSDN Library (1998), o desenvolvido pela OMG (*Object Management Group*) chamado de CORBA (*Common Object Request Broker Architecture*), ilustrado pela OMG (2008), a EJB (Sun Microsystems, 2008), entre outros. Como o uso de padrões de interface IDL (*Interface Definition Language*), CORBA e DCOM faz a programação distribuída simples, permite ao desenvolvedor tratar e usar qualquer objeto remoto como se fosse local para o usuário.

A COM (*Component Object Model*) permite ao programador escrever funções e métodos que podem ser acessados ou chamados por outras aplicações, COM é a tecnologia desenvolvida pela Microsoft para substituir ao OLE (*Object Linking and Embedding*) e ao DDE (*Dynamic Data Exchange*). DCOM surgiu para lidar com as deficiências de COM no suporte de componentes remotos. DCOM é uma extensão de COM que permite interatuar



dois programas através da rede, ainda se eles estiverem escritos em diferentes linguagens de programação (Chappel, 2000).

Mesmo facilitando o trabalho do programador no desenvolvimento de aplicações cliente-servidor, o DCOM falha em duas áreas: (1) embora DCOM possa ser usado em outras plataformas (MSDN Library, 1996 e Dedo e Nelson, 1997), só logra ser executado com todas suas vantagens na plataforma Windows (Box, 2000). (2) a dificuldade de desenvolver aplicações usando DCOM em ambientes corporativos, onde a comunicação é executada através de *firewall* (Lowy, 2001). Apesar disto é possível superar estas limitações diminuindo a segurança além do aceito pela área de tecnologia da informação da corporação e pelos administradores de rede.

### **2.4.3 A tecnologia OLE/COM**

A tecnologia OLE 1.0 foi desenvolvida pela Microsoft em meados de 1990, para suprir a necessidade de se integrar diferentes aplicações dentro da plataforma Windows, de forma a solucionar os problemas de desempenho e confiabilidade do até então utilizado padrão DDE (Fonseca, 2002). Foram introduzidos dois conceitos: *Linking* cria vínculos ou referências aos objetos, armazenando no documento principal apenas os dados realmente necessários para exibir, imprimir, etc. *Embedding* incorpora os dados dos objetos ao documento principal. Neste contexto, surgiram os conceitos de objeto vinculado e de objeto incorporado: Objeto Vinculado são informações (objetos) criadas em um arquivo (arquivo origem) e inseridas em outro arquivo (arquivo destino).

Embora o objeto vinculado não se torne parte do arquivo de destino, existe um vínculo, uma conexão entre os dois arquivos de forma que o objeto vinculado no arquivo de destino seja automaticamente atualizado quando o arquivo de origem é atualizado. Objeto incorporado consiste de informações inseridas em um arquivo de destino, ao ser incorporado, o objeto se torna parte do arquivo (Duarte *et al.*,2006). Ao clicar duas vezes no objeto incorporado, ele é aberto no programa de origem em que foi criado. Qualquer alteração feita no objeto incorporado se refletirá no arquivo de destino. Outro conceito importante na tecnologia OLE é o conceito de Cliente Servidor, cliente é uma aplicação que solicita os dados e servidor é uma aplicação que disponibiliza os dados.

Segundo Duarte *et al.* (2006), houve muitas melhorias na tecnologia OLE 2.0, a mais importante é a Automação OLE, pois permite que uma aplicação seja controlada por outra aplicação. A tecnologia OLE é montada sobre a tecnologia COM que define um modo padronizado para a comunicação dos módulos cliente e servidor por meio de uma interface específica. Um módulo indica um aplicativo ou uma biblioteca (uma DLL – *Dynamic Link Libraries*). Os dois módulos podem ser executados no mesmo computador ou em máquinas diferentes conectadas através de uma rede.

#### 2.4.4 A tecnologia OPC

Segundo Anwar *et al.* (2004), a motivação para o desenvolvimento de OPC (*OLE for Process Control*) foi ter um padrão para comunicação de várias fontes de dados, dispositivos de campo, ou banco de dados na sala de controle. A arquitetura dessas fontes de dados na indústria de processo é mostrada na Figura 2.8 e envolve três níveis:

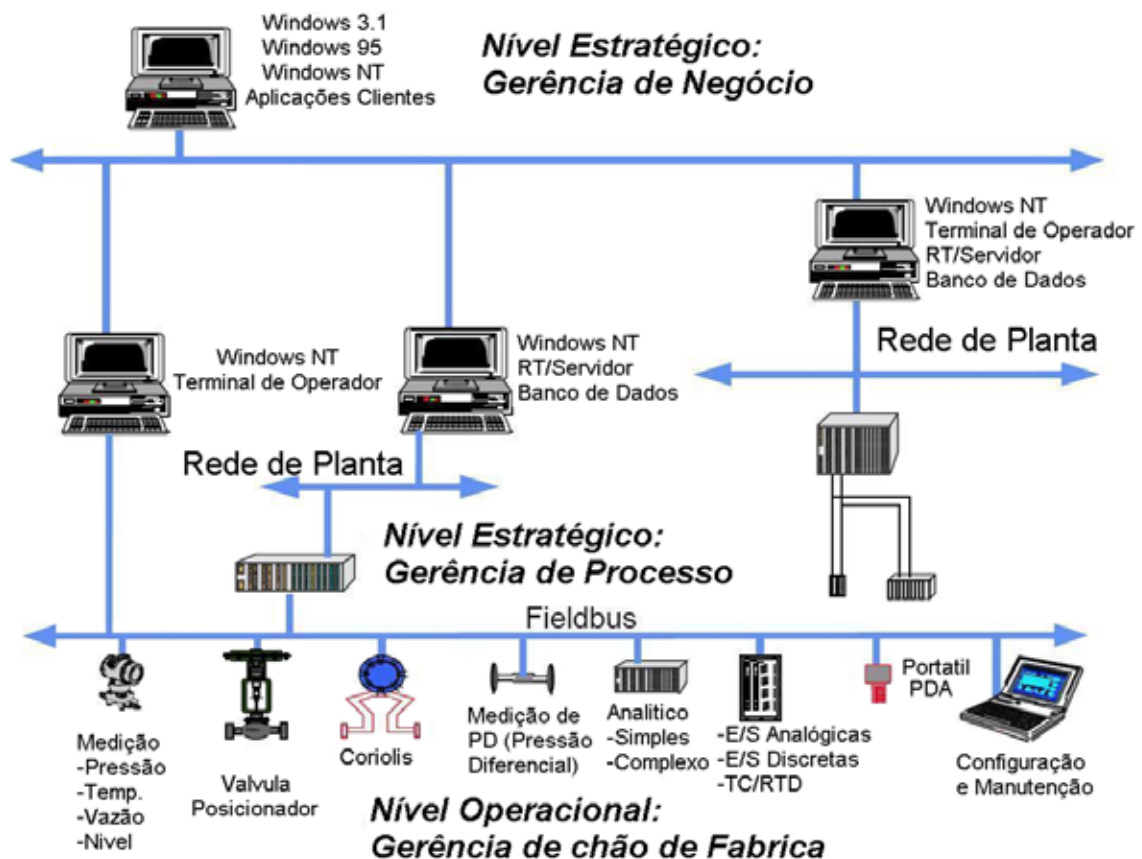


Figura 2.8- Arquitetura da informação no controle de processos (OPC Foundation 1998, modificado)

1. Gerência de chão de Fábrica: com a aparição de dispositivos de campo inteligentes, a quantidade de informação pode ser usada para avaliação de saúde dos dispositivos

de campo. Estas informações são adquiridas a partir dos dispositivos, parâmetros de configuração, materiais de construção, etc. Toda a informação tem que ser apresentada ao usuário de uma maneira consistente,

2. Gerência de Processo: com a instalação de DCS e sistemas SCADA para monitorar e controlar os processos de manufatura, disponibilizando eletronicamente os dados.
3. Gerência de Negócio: muitos benefícios podem ser ganhos com a instalação de sistemas integrados de informação dentro do sistema empresarial, administrando aspectos financeiros do processo de manufatura.

Segundo Shimanuki (1999), para acessar informações de um equipamento através de uma aplicação deve ser desenvolvida uma interface customizada ou um *driver* de comunicação. Muitas destas aplicações não conseguem acessar as informações devido à inconsistência entre fabricantes de *drivers* e *hardwares*. Na busca de uma solução para esse problema, foi desenvolvida a tecnologia OPC, que é uma tecnologia para conectar aplicações *Windows* e equipamentos de controle de processos. O OPC é um protocolo de comunicação aberto que permite um método consistente de acesso aos dados de inúmeros equipamentos dos mais diversos fabricantes (Figura 2.9). O método é o mesmo, independente da origem dos dados, o que vem oferecer ao usuário final uma maior liberdade na escolha dos equipamentos independentemente do fabricante.

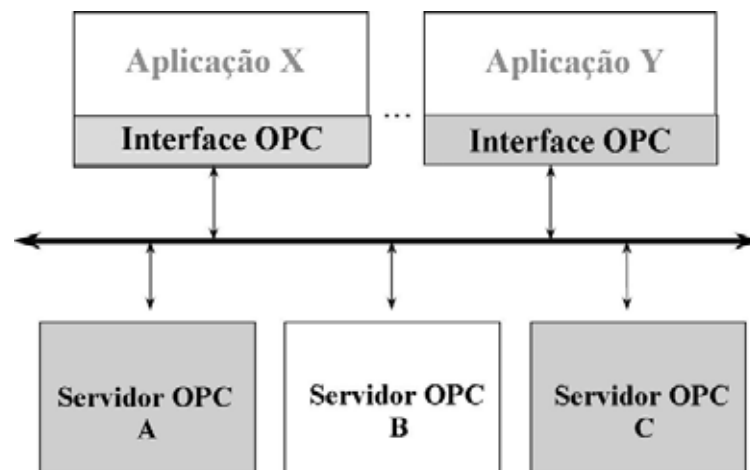


Figura 2.9- Aplicações com vários servidores OPC.

O OPC é construído usando tecnologia *Microsoft* OLE/COM, mas a especificação OPC foi desenvolvida por uma fundação aberta, a *OPC Foundation*, para atender as necessidades gerais da indústria e não as necessidades específicas de alguns fabricantes de *hardware* e *software* (OPC Foundation, 1998). A especificação ainda prevê a evolução das

funcionalidades ao longo do tempo e por isso, os componentes OPC podem se manter no topo das necessidades emergentes da indústria.

Os componentes OPC se classificam em duas categorias: clientes OPC e servidores OPC. Um cliente OPC é tipicamente um usuário dos dados tais como uma interface de operação ou um sistema SCADA. Um servidor OPC é uma fonte de dados que coleta ou gera dados a partir de um processo, disponibilizando-os aos clientes OPC. O cliente OPC interage com o servidor OPC usando uma interface definida. Qualquer cliente OPC pode se comunicar com qualquer servidor OPC, independentemente do tipo de dispositivo e do fabricante (Shimanuki, 1999), conforme está esquematicamente representado na Figura 2.10. Essa comunicação é válida somente para o OPC-DA (*Data Access*), uma vez que existem diferentes tecnologias OPC (Duarte *et al.*,2006).

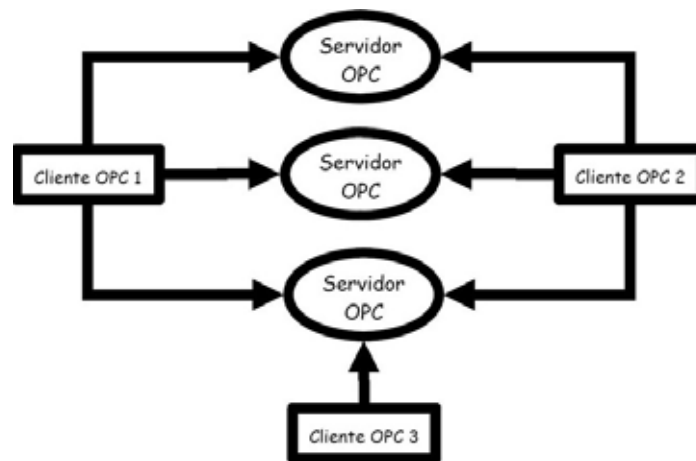


Figura 2.10- Relação entre Clientes e Servidores.

O padrão OPC estabelece as regras para que sejam desenvolvidos sistemas com interfaces padrões para comunicação dos dispositivos de campo (controladores, sensores, atuadores, etc.) com sistemas de monitoração, supervisão e gerenciamento SCADA, MES (*Manufacturing Execution Systems*), ERP, etc.

Os três componentes básicos da arquitetura OPC apresentados na Figura 2.11 são: servidor, grupo e item. Do ponto de vista do cliente, um servidor é essencialmente uma estrutura de armazenagem para grupos que, por sua vez, têm como função básica o armazenamento de itens. Esses itens, elementos mais simples na especificação, representam conexões a pontos de entrada ou saída. Assim, o item OPC não é um valor, mas apenas um meio de acesso a um valor. Desta forma, uma única variável de entrada ou saída pode ser representada por itens diferentes, com propriedades distintas e compartilhada por mais de um cliente. A

tarefa dos grupos é juntar o conjunto de itens que interessam a um determinado cliente, assumindo o papel principal na interação cliente-servidor.

Os grupos também são responsáveis por satisfazer pedidos de leitura e escrita, bem como por enviar atualizações para seus clientes, periodicamente ou por exceção. Essas transações de atualização podem ser ativadas ou desativadas no grupo ou nos itens individuais. Os grupos presentes em um servidor OPC são normalmente definidos pelos clientes, e somente o cliente criador do grupo pode acessá-lo; tal tipo de grupo é dito privado. Em alguns casos, porém, pode ser interessante que o servidor ofereça grupos passíveis de serem compartilhados por vários clientes. Quando essa capacidade é desejada, implementa-se a funcionalidade opcional dos grupos públicos.

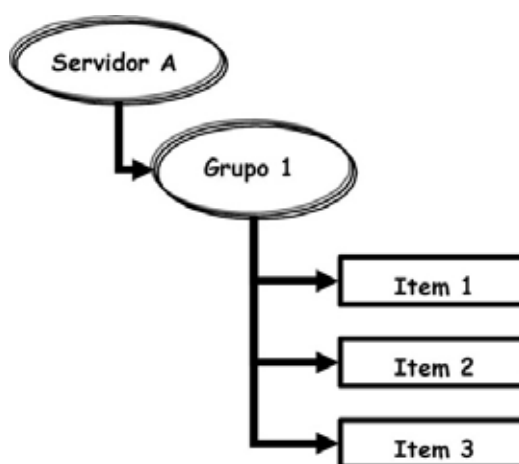


Figura 2.11- Arquitetura OPC

Do ponto de vista do cliente, a função básica do servidor é prover uma infra-estrutura de suporte aos grupos. Além disso, cabe também a ele gerenciar aspectos relacionados à conexão com uma fonte de dados, tais como parâmetros de comunicação ou taxa máxima de amostragem. Outra responsabilidade do servidor é implementar uma estrutura de endereçamento capaz de associar itens com variáveis reais (Souza *et al.*, 1998).

O item é uma estrutura a qual estão associadas três propriedades (Fonseca, 2002):

1. *Value*, último valor armazenado pelo servidor no cachê de memória do item e que é atualizado sempre que o servidor faz uma leitura no dispositivo,
2. *Quality*: informação de estado que define a qualidade do dado que pode ser: *Good*, dado válido, *Bad*, perda do *link* de comunicação com o dispositivo de campo, e *Uncertain*, no caso de existir o *link* e o dispositivo de campo estiver fora de comunicação.

### 3. *Time Stamp*, data e hora em que o item é adquirido.

Segundo Koskinen (2005), MIMOSA e OPC *Foundation* colaboraram no desenvolvimento de uma abrangente, arquitetura de informação aberta para operação e manutenção (O&M), estas informações possibilitam operações baseadas em condição (CBO - *Condition Based Operations*) e administração colaborativa do ciclo de vida de ativos (CALM - *Collaborative Asset Lifecycle Management*). O padrão OPC é utilizado para comunicação entre o dispositivo mestre de uma rede FF e o computador configurador e supervisor (Zheng e Nakagawa, 2002).

A *OpenO&M* (Operação e manutenção aberta) surgiu pela existência de padrões tecnológicos permitindo a combinação do existente OPC e MIMOSA (*Machinery Information Management Open Systems Alliance*), padrões que estão continuamente melhorando no tempo, OPC XML DA e MIMOSA OSA-EAI (Koskinen, 2005). Segundo MIMOSA e OpenO&M (2004), as contribuições dos padrões ISA SP95 (Modelo de fluxo de processo da manufatura), OPC *Foundation* (Transporte) e MIMOSA (conteúdo) permitiram o desenvolvimento de um padrão *OpenO&M* para indústrias de processo.

#### **2.4.5 Comparação das Tecnologias**

A escolha da tecnologia OPC é porque é um protocolo aberto, transparente e independente do fabricante. A maioria dos instrumentos, controladores, CLP, etc., disponibilizam seus dados via um servidor OPC ou os armazenam em outro formato próprio do fabricante.

A instrumentação *fieldbus* foi escolhida porque é aquilo que está instalada na usina hidrelétrica, mas a metodologia proposta pode ser aplicada com outro tipo de instrumentação desde que eles disponibilizaram seus dados via servidor OPC existindo para isso *softwares* que convertem os outros tipos de dados armazenados em dados OPC.

## **2.5 ARQUITETURAS DOS SISTEMAS DE MANUTENÇÃO BASEADA EM CONDIÇÃO**

Neste trabalho apresentam-se algumas arquiteturas que foram discutidas em outras pesquisas. Uma arquitetura para manutenção baseada em condição é apresentada por Vachtsevanos e Wang (2001), chamada de sistema de prognóstico o qual tem como saída o tempo de falha como o objetivo de executar ações de manutenção baseada em condição,

ver Figura 2.12. O diagnosticador avalia a condição real de um componente através da medição em tempo real de um sensor, o propósito é chegar à conclusão da existência de uma condição de falha iminente ou incipiente. O prognosticador que tem entradas do diagnosticador decide a necessidade de manter um componente, baseado no histórico de taxa de falha, modelos de falha apropriadas, e programas de manutenção.

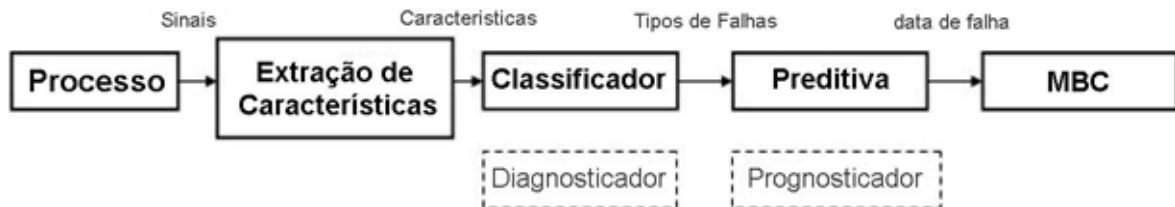


Figura 2.12- Arquitetura geral de um sistema de prognóstico (Vachtsevanos e Wang 2001, modificado).

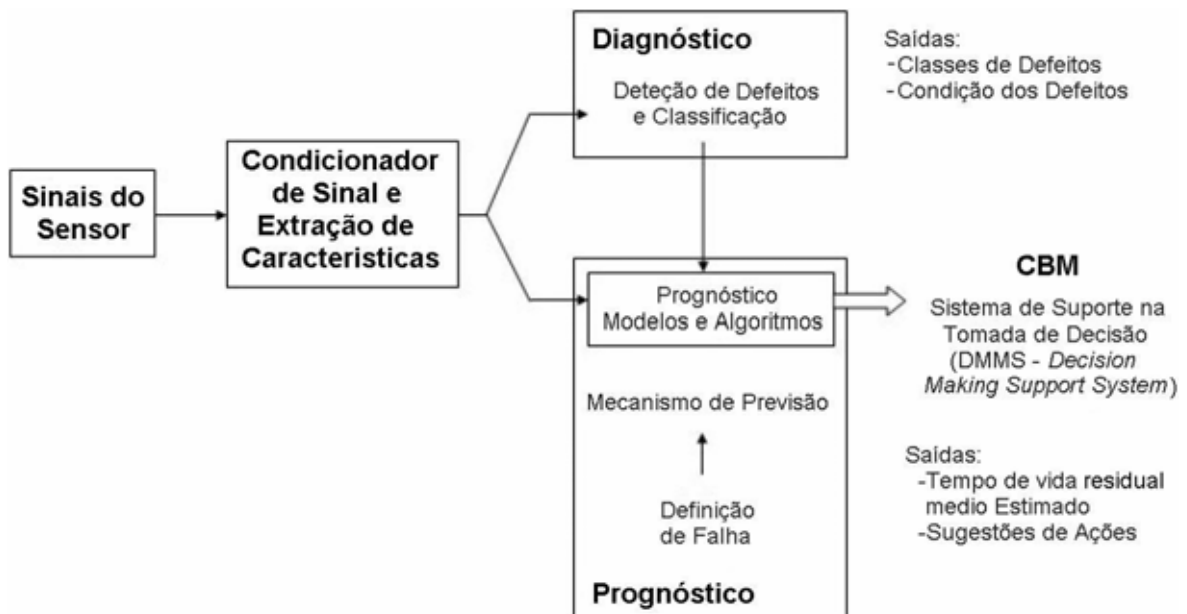


Figura 2.13- Elementos básicos de diagnóstico e prognóstico para manutenção baseada em condição, apresentado por (Chinnam e Baruah 2004, modificado).

Outra proposta é apresentada por Chinnam e Baruah (2004), focando o diagnóstico e prognóstico. Sensores não intrusivos são instalados na monitoração dos componentes com o objetivo de capturar sinais de degradação que por uma subsequente interpretação podem levar ao desenvolvimento de certas políticas de manutenção personalizadas. Isto em conjunto com os avanços da tecnologia dos sensores, *hardware* de aquisição de dados e algoritmos de processamento de sinal, redução dos custos dos computadores e redes, e a facilidade no aumento dos produtos da tecnologia da informação, fazem os diagnósticos e

prognósticos mais efetivos, bons e baratos. Na Figura 2.13 são apresentados os elementos básicos desta arquitetura para um sistema de MBC.

Thurston (2001) apresentou o trabalho de organização da Arquitetura de Sistema aberto para manutenção baseada em condição OSA-CBM. A arquitetura foi desenvolvida pela necessidade de um padrão para lidar com o fluxo de informação de componentes de *software* em um sistema. A arquitetura de sete camadas diferentes (Figura 2.14), todas representativas de capacidades diferentes, é especificada a seguir: (1) Aquisição de Dados, (2) Processamento de Sinal, (3) Monitoramento de Condição, (4) Avaliação de saúde (diagnóstico), (5) Prognóstico, (6) Tomada de Decisão, e (7) Apresentação.

As sete camadas do modelo de referência OSA-CBM são abordadas em trabalhos apresentados por Amaya *et al.* (2007c), Bengtsson (2003), Bengtsson (2004a), Bengtsson *et al.* (2004) e Thurston (2001), onde a camada de aquisição de dados fornece sinais ao sistema, os quais são processados na camada processamento de sinal. A monitoração de condição determina características anormais, as quais são classificadas na camada avaliação de saúde, a camada prognóstico prevê o tempo de vida útil, todas as informações anteriores são tomadas em conta quando são programadas ações de manutenção na camada tomada de decisão, e na apresentação podem ser mostradas todos os dados das camadas anteriores.

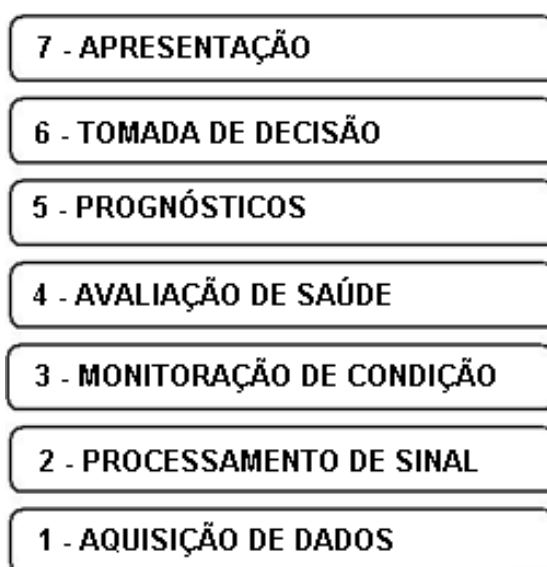


Figura 2.14- As sete camadas funcionais OSA-CBM (Amaya *et al.*, 2007c).

Outras arquiteturas são, a *Watchdog Agent tool kit* apresentada por Djurdjanovica *et al.* (2003) e Lee *et al.* (2006), que consiste em ferramentas e algoritmos para avaliação e



predição do desempenho de máquinas. Szymanski *et al.* (2003) e Bangemann *et al.* (2006) apresentaram um plataforma genérica para manutenção eletrônica (*e-maintenance*), chamada PROTEUS, para integração dos programas de manutenção existentes. Um projeto de integração é o chamado PROMISE (*PROduct lifecycle Management and Information tracking using Smart Embedded systems*)

PROMISE contribuiu ao desenvolvimento de uma plataforma abrangente para administrar o ciclo de vida de um produto, analisando-o em todas suas fases de ciclo de vida, mas não inclui MPd (Promise, 2008). TATEM é um projeto que ajuda a melhorar a operabilidade, segurança e redução dos custos de manutenção através da detecção de falhas atuais e incipientes, e inclui todos os aspectos da MPd focado na aplicação aeronáutica (Tatem, 2008).

Dentro de todas as arquiteturas de MBC apresentadas nesta seção, a arquitetura OSA-CBM foi escolhida por quatro motivos:

1. Custo: a OSA-CBM economiza tempo e dinheiro no desenvolvimento de uma arquitetura proprietária;
2. Especialização: no desenvolvimento pode-se ter varias equipes, cada um deles pode-se concentrar em camadas específicas, permitindo desenvolver e melhorar os algoritmos e tecnologias em uma determinada área;
3. Competição: a OSA-CBM permite usar as mesmas interfaces de entrada e saída nas camadas. Apresentando informações às camadas desenvolvidas por diferentes equipes é possível comparar as funcionalidades desenvolvidas. A competição pode ocorrer até o nível funcional e não necessariamente até o nível total ou sistema;
4. Cooperação: não só a competição é acrescentada, mas também a cooperação das diferentes equipes. O padrão define a interface onde cada módulo pode-se comunicar com outros módulos de forma transparente desde que esses módulos usaram a mesma tecnologia.

## **2.6 OSA-CBM**

Na implementação de sistemas de manutenção baseados em condição lida-se com a tarefa de integrar uma variedade de componentes de *software* e *hardware*, assim como de desenvolver uma estrutura para estes componentes. OSA-CBM simplifica este processo especificando uma arquitetura padrão para implementação de sistemas de manutenção

baseada em condição, descrevendo as sete camadas funcionais dos sistemas de manutenção baseada em condição, como também as interfaces entre essas camadas. O padrão provê a forma para integrar componentes de diferentes fabricantes e facilita o processo especificando as entradas e saídas entre os componentes (Walter, 2006).

A OSA-CBM é definida por MIMOSA (2008) como: “Uma arquitetura padrão para movimento de informação em um sistema de manutenção baseada em condição. Além de reduzir custos, melhora a interoperabilidade, aumenta a competência, acrescenta mudanças no desenho, e oferece cooperação adicional no domínio da manutenção baseada em condição”. Segundo Swearingen *et al.* (2007), o padrão OSA-CBM v3.1 é definido usando UML (*Unified Modeling Language*) e é desenhado como uma implementação multi-tecnológica, que separa a informação que é trocada num sistema de manutenção baseada em condição, das interfaces técnicas dos sistemas integrados, usados para transmitir informação; é assim que os vendedores e integradores podem implementar o padrão usando a tecnologia apropriada para o ambiente.

Segundo Lebold e Thurston (2001), a OSA-CBM foi desenvolvida em 2001 por um grupo líder industrial fundado parcialmente pela Marinha dos Estados Unidos através do programa DUST (*Dual Use Science and Technology*), para desenvolver e demonstrar uma arquitetura de sistema aberto para MBC. O grupo de participantes cobre uma gama extensiva de aplicações industriais, comerciais, e militares da tecnologia de MBC: *Boeing, Caterpillar, Rockwell Automation, Rockwell Science Center, Newport News Shipbuilding, e Oceana Sensor Technologies*. Outros contribuintes do grupo incluem o Laboratório de Pesquisa aplicada da *Penn State University*, e MIMOSA (Lebold e Thurston, 2001). O foco do consórcio é o desenvolvimento e demonstração de uma arquitetura de *software* que facilita interoperabilidade das camadas do *software* para MBC (Lebold e Thurston, 2001).

Segundo Walter (2006), o motivo para o desenvolvimento do padrão foi que a marinha dos Estados Unidos gastava bilhões de dólares em manutenção a cada ano. A maioria dos custos está na forma de mão-de-obra e parte dos custos vem de *software* e *hardware* proprietário. A meta foi a padronização das especificações de intercâmbio de informação dentro da comunidade de usuários de MBC podendo idealmente dirigir-se aos fornecedores base de MBC para produzir componentes de *software* intercambiável.

### 2.6.1 Arquitetura OSA-CBM

Tipicamente, um sistema de manutenção baseada em condição utiliza uma variedade de produtos de *software* e *hardware* e a combinação de interfaces padrões abertas e proprietárias. Além da dificuldade de integrar produtos de muitos fabricantes, o integrador é também limitado às capacidades do sistema. O uso de uma interface padrão poderia reduzir significativamente o tempo requerido no desenvolvimento e integração dos componentes dos sistemas especializados.

Os antigos padrões de MIMOSA serviam como um meio para o intercâmbio de informação estática entre sistemas de manutenção incompatíveis, esta foi a intenção primária para o CRIS (*Common Relational Information Schema*) e suas interfaces associadas. MIMOSA não teria, porém, um padrão de interface viável para transações entre componentes dentro de um único sistema de manutenção (Mimosa, 2008). A arquitetura desenvolvida pelo programa OSA-CBM é projetada para encaminhar a necessidade por um padrão para fluxo transacional de informação entre componentes de *software* em um sistema de MBC, sendo a última meta a interoperabilidade de componentes compatíveis com o padrão.

Depois de avaliar as opções, a decisão tomada foi criar uma arquitetura de comunicação cliente-servidor utilizando conceitos de projeto orientado a objetos. Na implementação, pela falta de um consenso na escolha de uma única tecnologia, adotou-se a decisão de desenvolver uma tecnologia neutra de uma especificação de projeto simples, que poderá ser planejada e implementada utilizando qualquer das tecnologias existentes no mercado. O primeiro passo na descrição do componente de *software* da arquitetura foi dividir o sistema de MBC em camadas, um componente de *software* compatível, onde podem ser implementadas as funções e interfaces definidas para uma ou mais camadas.

A arquitetura OSA-CBM consiste de sete camadas independentes de funcionalidades baseado no padrão ISO-13374:2007, mostrado na Figura 2.15. A ISO-13374 (2007) não especifica como implementar um sistema de monitoramento de condição, que tecnologias usar, ou que algoritmos implementar, mas provê uma estrutura geral para a área de monitoramento de condição e diagnóstico. OSA-CBM define os tipos de dados a serem usados para o processamento e apresentação dos resultados num sistema de monitoramento de condição, como também quais informações são transmitidas entre pontos de processo e

armazenamento, permitindo desenvolver as funcionalidades de cada camada independentemente.

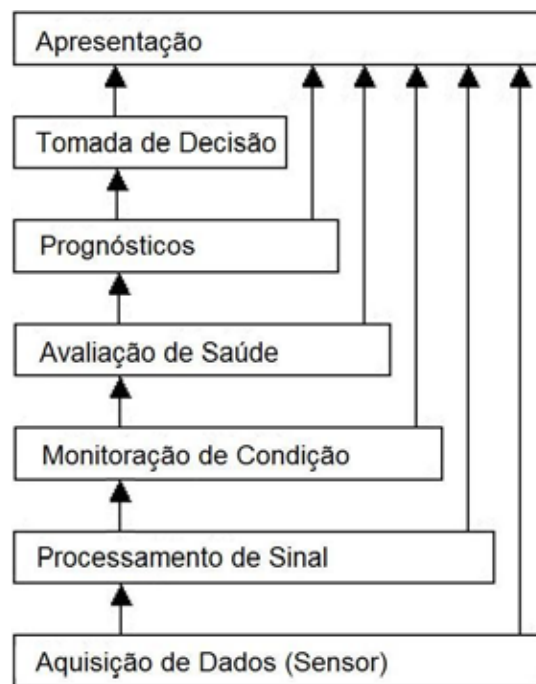


Figura 2.15- Fluxo de dados das camadas OSA-CBM

Segundo Lebold e Thurston (2001), uma vez definida a especificação do modelo OSA-CBM para cada camada, as camadas podem ser construídos no sistema. A Figura 2.16 mostra como uma camada OSA interatua com as outras para completar o sistema integrado. O centro do círculo representa o meio de comunicação entre as camadas, o que pode ser conseguido usando o protocolo da internet TCP/IP ou HTTP (*Hyper Text Transfer Protocol*), de modo que as camadas não precisam estar no mesmo computador podendo se encontrar em qualquer local da rede mundial.

Segundo Lebold *et al.* (2003), o modelo OSA-CBM consiste em sete camadas. A noção de uma arquitetura estendida em camadas é consistente com o conceito usado em Buschmann *et al.* (1996) e Álvares *et al.* (2007). As camadas hierárquicas representam uma transição lógica ou um fluxo de saída dos sensores para a camada de tomada de decisão, através das camadas intermediárias. A camada apresentação é uma exceção dentro da arquitetura, pois permite comunicação ponto-a-ponto entre esta camada e qualquer outra. Segundo Jardine *et al.* (2006), a manutenção baseada em condição consiste em três passos principais: aquisição de dados, processamento de dados e tomada de decisão de manutenção mostrados na Figura 2.17.

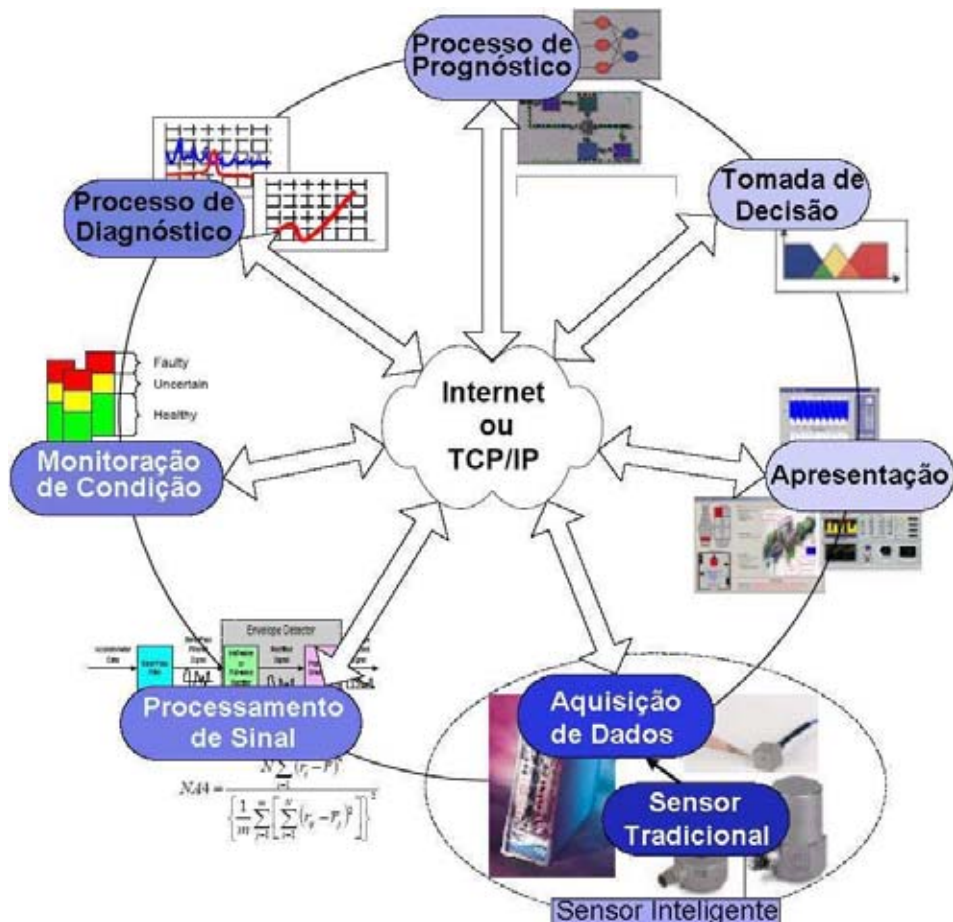


Figura 2.16- Camadas funcionais OSA-CBM (Lebold e Thurston 2001, modificado).

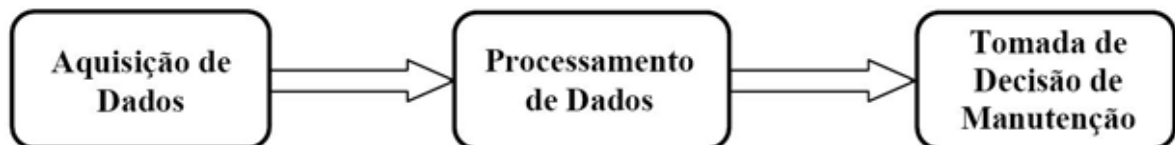


Figura 2.17- As três camadas MBC (Jardine *et al.* 2006, modificado).

Uma camada OSA-CBM fornece três tipos de informação (OSA-CBM, 2006):

1. Dados, são as informações ou eventos que uma camada gera, podem ser as leituras de sensor inteligente<sup>1</sup> na camada de aquisição de dados e o estado de saúde para a camada de diagnóstico;
2. Configuração, informação sobre os recursos de entrada das camadas, descrições dos algoritmos usados para processar dados de entrada, lista de saídas, e várias especificações de saída como unidades de engenharia e valores de alarmes;

<sup>1</sup> Sensor inteligente, dispositivo capaz de prover funções além daquelas necessárias para gerar uma correta representação da quantidade medida e/ou controlada (Especificação IEEE 1451.2).

3. Explicação, o dado ou referência a um dado usado por uma camada para produzir uma saída.

#### 2.6.1.1 Aquisição de dados

Normalmente, num contexto objetivo, o sensor é um componente de aquisição de dados e considerado parte da camada monitoração de condição. Segundo Fraden (2003) o sensor é um dispositivo que recebe e responde a um sinal ou estímulo, é assim o equipamento que captura o efeito dinâmico causado pela falha incipiente. Segundo Bengtsson (2004a) esta camada deve ser desenvolvida de acordo com o padrão IEEE (*Institute of Electrical and Electronics Engineers*) Std 1451 (Figura 2.18).

#### 2.6.1.2 Processamento de sinal

Esta camada recebe dados e sinais da camada aquisição de dados e de outros sistemas de processamento de sinal, a saída da camada processamento de sinal inclui digitalização e filtragem dos dados do sensor, espectro de frequência, sinais de sensores virtuais entre outras. Segundo Bengtsson *et al.* (2004), os propósitos da camada processamento de sinal são: (1) remover distorções e restabelecer o sinal à sua forma original, (2) remover dados irrelevantes do sensor para diagnósticos ou prognósticos, e (3) transformar o sinal para fazer as características relevantes mais explícitas.

#### 2.6.1.3 Monitoração de condição

A monitoração de condição recebe dados das camadas de aquisição de dados, processamento de sinal e de outros sistemas de monitoração de condição, onde o foco principal é comparar dados com os valores esperados (Bengtsson *et al.*, 2004). Segundo Bengtsson (2004a), esta camada deve ser desenvolvida usando o padrão ISO 13373-1 (Figura 2.18). Se os níveis normais são excedidos ou outro fenômeno anormal acontece, como aumentos súbitos ou diminuições no nível (mas ainda não excedendo níveis dos limites operacionais), os dados precisam ser diagnosticados. Os limites podem ser estáticos ou dinâmicos (Tsang, 1995).

Os avisos de limites estáticos utilizam valores limites predeterminados. De acordo com Tsang (1995), limites de advertência estáticos são mais facilmente administrados do que os limites dinâmicos. Mas eles não têm o poder de diagnóstico para predizer quando o alarme

será alcançado. Os limites dinâmicos são usados para monitorar a taxa de mudança do parâmetro medido, se um procedimento de manutenção baseada em condição usa limites de advertência dinâmicos, a taxa de mudança do parâmetro medido é considerada mais importante que o valor atual (Tsang, 1995). A monitoração de condição deve ser capaz de gerar alertas baseados nos limites operacionais estabelecidos.

#### 2.6.1.4 Avaliação de saúde (diagnóstico)

Esta camada recebe dados da monitoração de condição ou de outros sistemas de avaliação de saúde, focando-se em prescrever se a saúde do componente monitorado, sistema ou subsistema foi degradada (Bengtsson *et al.*, 2004). Segundo Bengtsson (2004a), esta camada pode ser desenvolvida a partir das normas IEEE 1232 e a ISO 13373-1 (Figura 2.18). De acordo com Yam *et al.* (2001), os diagnósticos na manutenção baseada em condição podem ser divididos em três categorias: (1) diagnósticos baseados em regras, (2) diagnósticos baseado em casos, e (3) diagnósticos baseados em modelo. A avaliação de saúde deve ter a capacidade de gerar registros de diagnósticos e indicar as possíveis falhas baseada em tendências do histórico de saúde, estado operacional e histórico de manutenção (Bengtsson *et al.*, 2004).

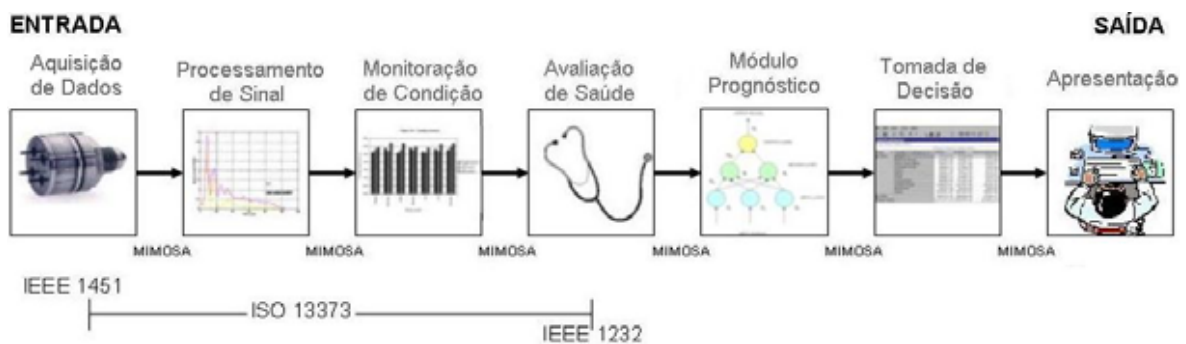


Figura 2.18- Camadas OSA-CBM, destacando os padrões que podem ser usando no desenvolvimento de sistemas de MBC (Bengtsson 2004a, modificado).

#### 2.6.1.5 Prognósticos

Continuando com o diagnóstico, o sistema terá o conhecimento se uma condição é anormal, e o que está causando essas medidas anormais, precisando ser prognosticado. Esta camada será a que poderá prever quanto tempo um equipamento pode operar antes de ser necessário executar uma ordem de manutenção prévia a uma falha. Os prognósticos podem ser executados como a camada de diagnósticos, através de diferentes técnicas de

inteligência artificial, como redes neurais recorrentes (Yam *et al.*, 2001) e redes neurais *wavelet* dinâmicas (Vachtsevanos e Wang, 2001), etc.

Djurdjanovica *et al.* (2003) apresenta o sistema *WatchDog Agent* que implementa dezenas de ferramentas e algoritmos de prognósticos, baseados em Transformada de Fourier, Modelo Auto-regressivo, lógica nebulosa (*fuzzy*), redes neurais artificiais, entre outros. Jardine *et al.* (2006) apresenta uma revisão extensa de diagnósticos e prognósticos dentro da manutenção baseada em condição. Maiores detalhes sobre abordagens de algoritmos de prognósticos podem ser obtidos no Apêndice A.

Na Figura 2.19, mostra-se um diagrama de blocos simples da camada de prognóstico genérico com o objetivo de visualizar os componentes principais. Dentro desta camada, a informação pode ser utilizada por uma abordagem baseada em modelos, abordagem baseada características ou a combinação das duas.

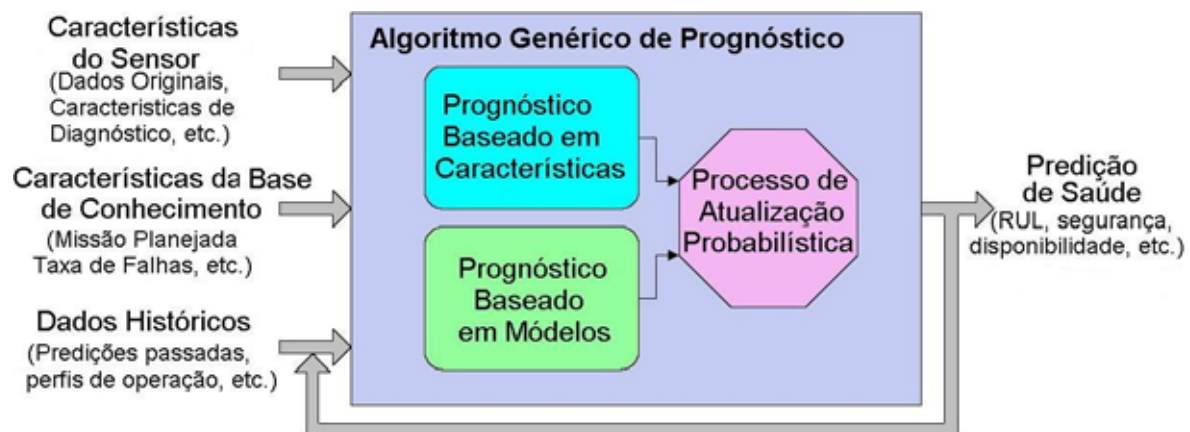


Figura 2.19- Componentes da camada de prognóstico genérico (Lebold e Thurston 2001, modificado).

A camada de prognóstico deve prover informação específica ao usuário acerca do estado de saúde dos componentes, tempo de vida remanescente (RUL - *Remaining Useful Life*), segurança e recomendações. Lebold e Thurston (2001) apresentam uma proposta para a camada de prognóstico genérico no qual os requerimentos das entradas são dados históricos, por exemplo, na forma de saúde, falhas, missão, manutenção, modelo de informação, e ativos de peças sobressalentes. Estes recursos de entradas e saídas (Figura 2.20) ajudam a definir a estrutura e aclarar a presença da camada de prognóstico no modelo OSA-CBM.



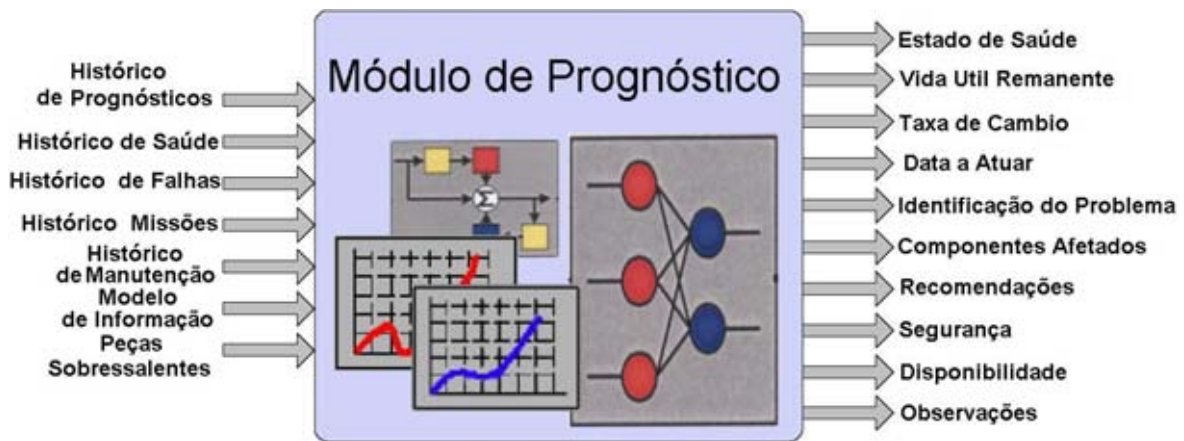


Figura 2.20- Entradas e saídas gerais da camada de prognóstico OSA-CBM (Lebold e Thurston 2001, modificado).

#### 2.6.1.6 Tomada de decisão

O último passo do sistema de manutenção baseada em condição é gerar decisões de acordo às ações de manutenção a executar. Todas as atividades prévias das camadas de avaliação de saúde e prognóstico devem ser integradas à tomada de decisão para achar a melhor solução para um evento particular. Sendo o foco principal gerar recomendações das ações de manutenção alternativas, estas ações relacionadas à manutenção ou como utilizar os recursos até completar a seqüência atual sem ocorrência de falha (Bengtsson *et al.*, 2004). Esta decisão deve de ser automática podendo ser usada por outros sistemas ou por um operador (Guo *et al.*, 2002).

#### 2.6.1.7 Apresentação

Esta camada é a interface homem/máquina, podendo se comunicar e apresentar dados das outras camadas. As camadas mais importantes para apresentar dados seriam a avaliação de saúde, prognósticos e tomada de decisão, assim como também os alertas gerados da monitoração de condição, devendo ter como possibilidade ver as camadas inferiores (Bengtsson *et al.*, 2004).

### 2.7 CONCLUSÕES DO CAPÍTULO

Neste capítulo foram estudados os conceitos de MBC, sistemas de MBC, tecnologias usadas no desenvolvimento de um sistema de MBC, as arquiteturas de sistemas de MBC, as camadas do modelo OSA-CBM, e pode-se concluir o seguinte:

A MBC é uma manutenção preventiva baseada no monitoramento das condições dos equipamentos através do uso de sensores para avaliação dos parâmetros de degradação, gerando diagnósticos e prognósticos de falhas iminentes.

Um sistema de MBC é usado para gerar diagnósticos em tempo real e através de uma interface com o usuário mostrar sugestões de ações de manutenção.

As tecnologias *Foundation Fieldbus* e OPC são protocolos abertos e independentes do fabricante.

Entre as arquiteturas de sistemas de MBC, a arquitetura OSA-CBM fornece interfaces de entrada e saída padronizadas e permite o desenvolvimento de cada uma das suas camadas independentemente.

### 3 REVISÃO BIBLIOGRÁFICA: TÉCNICAS INTELIGENTES

Este capítulo tem por objetivo a revisão e a apresentação das técnicas inteligentes de representação de conhecimento e aprendizado utilizados no desenvolvimento de um sistema de manutenção baseada em condição. Na primeira parte descreve-se a técnica de sistemas especialistas, conceitos, arquitetura. Na segunda parte são apresentados os principais conceitos sobre Redes Neurais Artificiais, lógica nebulosa e sistemas híbridos *Fuzzy ARTMAP* e seus algoritmos utilizados na fase de treinamento e desempenho.

#### 3.1 INTRODUÇÃO

Segundo Bittencourt (2001), duas linhas principais de pesquisa deram início à construção de sistemas inteligentes: a linha conexionista e a linha simbólica. A linha conexionista visa à modelagem da inteligência humana através da simulação de componentes do cérebro, isto é, de seus neurônios, e de suas interligações, dando origem à área de redes neurais artificiais. A construção de sistemas inteligentes do tipo simbólico originou-se com o sucesso dos sistemas especialistas (SEs), a partir da década de setenta, estabelecendo a manipulação simbólica de um grande número de fatos especializados sobre um domínio restrito.

Os SEs são uma classe de sistemas de Inteligência Artificial desenvolvidos para servirem como consultores na tomada de decisões que envolvam áreas restritas da Ciência, normalmente apenas dominadas por especialistas humanos. São sistemas que utilizam o conhecimento de um ou mais especialistas codificado em um programa que o aplica na resolução de problemas (Abel, 1998).

A combinação de duas ou mais técnicas de inteligência artificial é chamado de sistemas inteligentes híbridos. O objetivo principal é obter um sistema mais poderoso em termos de poder de interpretação, aprendizado, estimativa de parâmetros e generalização. Existem três formas básicas de construção de sistemas híbridos:

1. Sequencial: um subsistema com técnica A atua como entrada de outro subsistema com técnica B.
2. Auxiliar: um subsistema com técnica B é chamado pelo subsistema com técnica A.
3. Incorporado: não há separação visível entre os dois subsistemas.

Os tipos mais comuns de sistemas híbridos são o Neuro-Fuzzy, Fuzzy-Genético e Neuro-Genético.

### 3.2 SISTEMAS ESPECIALISTAS

A evolução dos SE começa nos anos 60, com o desenvolvimento dos programas de propósitos gerais GPS (*General Problem Solver*) programa criado por Newell e Simon (Durkin, 1994). Apesar de alguns progressos, esta estratégia não correspondeu à expectativa porque era muito difícil e infrutífero (Durkin, 1994). Nos anos 70, surgem os programas de propósitos especiais, especialistas em alguma área restrita. Estes sistemas possuem um corpo de conhecimentos de alto nível sobre um domínio limitado (Cunha, 1995). Nos anos 80, os SE foram desenvolvidos para aplicações nas áreas de finanças, medicina, manutenção, etc. A Figura 3.1 mostra onde se encaixam os SE dentro do contexto histórico das pesquisas em IA.

Os SE são um modo de embutir o conhecimento para imitar as decisões humanas (Jackson 1999, Lucas e Van der gaag 1991, e Russell e Norvig 2003), envolvendo como representar o conhecimento e técnicas para obter (inferências) novos conhecimentos, ou tomadas de decisões, desde uma base existente. Um dos maiores desafios nos SE é como adquirir e representar o conhecimento o mais exato possível, para tomar decisões mais perto ao especialista de domínio. Segundo Vinade (2003), os problemas geralmente apresentados no desenvolvendo de SE são a indisponibilidade dos peritos em criar conhecimento e as dificuldades no processo de extração de regra.

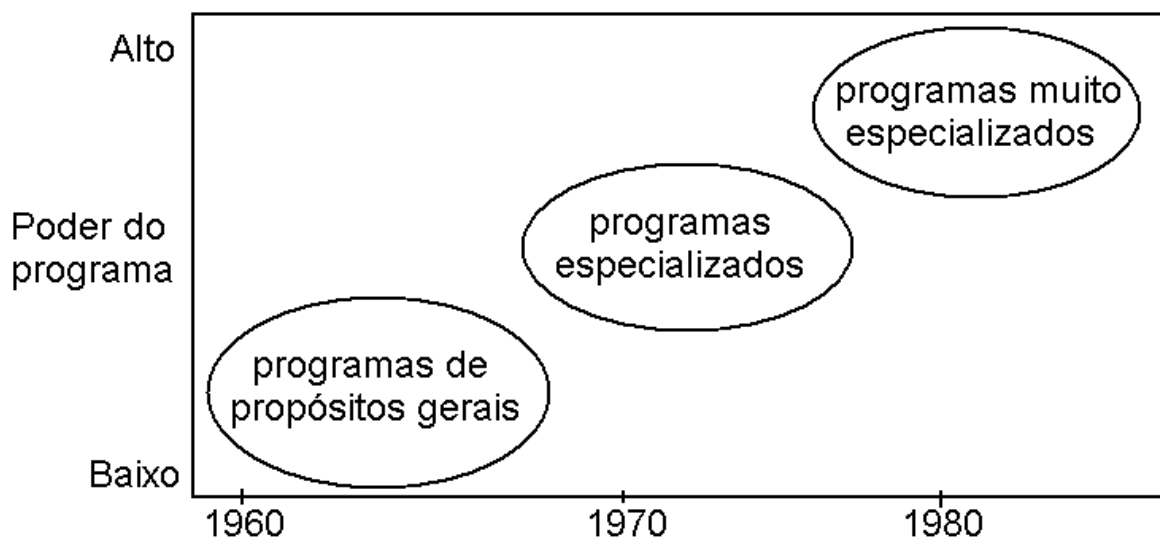


Figura 3.1- Contexto Histórico dos SE (Cunha, 1995).

Segundo Reis e Pati (2000), nas primeiras fases, os SE foram desenvolvidos em laboratórios de universidades. A maioria dos trabalhos foi feita sem nenhuma ferramenta particular, desenvolvidos por engenheiros de conhecimento. Na Tabela 3.1 listam-se alguns das ferramentas de *software* introduzidas nos Estados Unidos e reconstruídos no Japão.

Tabela 3.1- Ferramentas de *Software* para SE (Reis e Pati, 2000).

<b>Nome</b>	<b>Desenvolvedor</b>	<b><i>Hardware</i></b>
Interlisp - DJ	Xerox	Xerox 1121
Super BRAINS	Toyo Info. System	IBM 3090
ESHELL	Fujitsu	FACOM S3500, FACOM - f
VM/Prolog	IBM	IBM 3081
MYEXPERT	Toshiba	UX - 700
HPGS	Hitachi	M-200H
EUREKA	Hitachi	HIDIC V90/50
ESHELL	Fujitsu	FM.PC
ES/Kernel	Hitachi	IBM Platform

### 3.2.1 Arquitetura de um sistema especialista

Segundo Giarratano e Riley (1994), o professor Edward Feigenbaum da Universidade de Stanford, um dos pesquisadores reconhecidos em trabalhos com SE, definiu sistema especialista como: “um programa inteligente de computador que usa conhecimento e procedimentos de inferência para resolver problemas que são suficientemente difíceis para requerer significativa experiência humana para sua solução”. De acordo com esta definição, o SE é formado por dois módulos básicos mostrados na Figura 3.2 (Abel, 1998). Um banco de conhecimento contendo informações especializadas na área do problema a ser solucionado, codificado de maneira inteligível para ser facilmente modificado e/ou reutilizado (Caletti, 2003). Os mecanismos de inferência que representam os métodos inteligentes de manipulação do conhecimento para se chegar a uma solução, resposta ou conclusão, a partir de um determinado conhecimento inicial (Aulete, 1986 e Gonzalez e Dankel, 1993).

A representação do conhecimento é uma das principais preocupações dos Sistemas Especialistas e da Inteligência Artificial (Liebowitz, 1999). Segundo Durkin (1994), engenharia do conhecimento é “processo de construir um sistema especialista”. A diferença da programação convencional, desenvolver um sistema especialista é um processo altamente interativo que envolve, tipicamente, uma forma especial de interação

entre o desenvolvedor do SE, chamado Engenheiro de Conhecimento (EC), e um ou mais especialistas em um problema específico de uma área ou domínio de conhecimento. Segundo Silva (1998), em geral, é preferível que o engenheiro de conhecimento não tenha experiência no domínio no qual a aplicação de SE será desenvolvida e, do ponto de vista prático, não é necessário que a possua. A interação entre o EC e o especialista é mostrada na Figura 3.3.

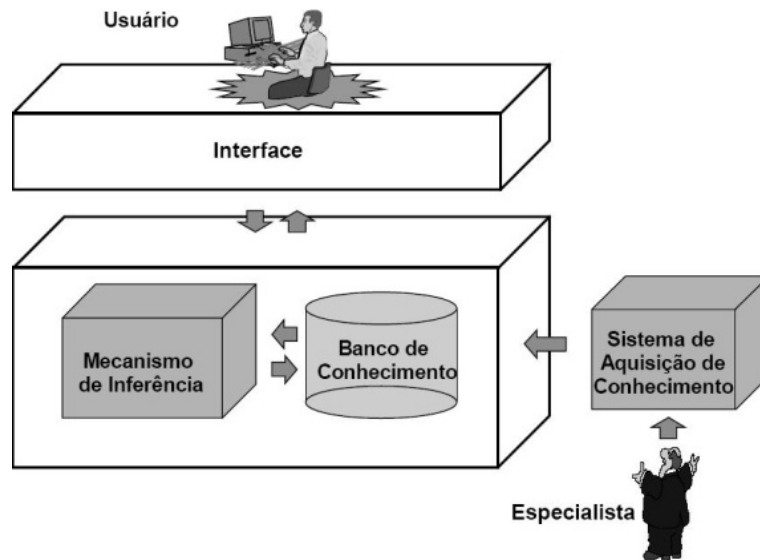


Figura 3.2- Arquitetura de um Sistema Especialista (Abel, 1998).

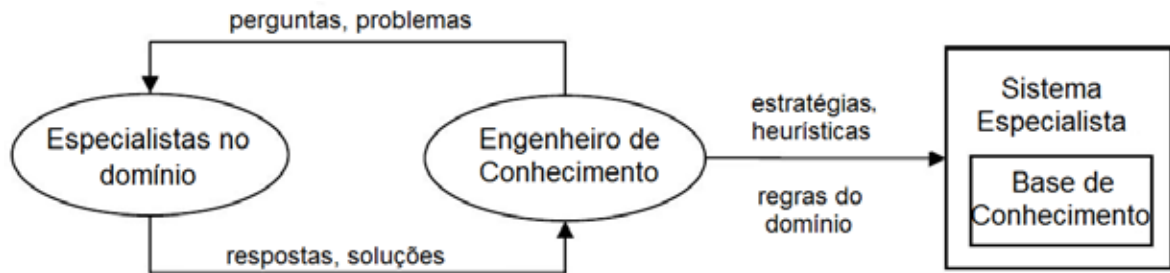


Figura 3.3- Esquema de interação entre o EC e o especialista no domínio do problema (Waterman, 1986).

### 3.2.2 Ferramentas para construção de SE

O desenvolvimento de um sistema especialista pode ser feito completamente com uma linguagem de programação como PROLOG ou LISP; porém, existem ferramentas computacionais especialmente desenvolvidas para a criação de SE, como o sistema ICAD, da KTI (*Knowledge Technologies International*), o ambiente CLIPS (*C Language Integrated Production System*) e o JESS (*Java Expert System Shell*).

O JESS é uma *shell* para SE desenvolvido completamente em Java por Ernest Friedman-Hill da *Sandia National Laboratories*. A primeira versão do JESS foi em 1995. O JESS originalmente foi o clone do CLIPS, mas atualmente tem muitas características diferenciadas do CLIPS adquirido pela influência do Java. O JESS permite chamar as funções Java, estendendo escrevendo códigos Java e embecendo o JESS em aplicações Java. Com o JESS pode-se dar a um *Applet* Java e a outras aplicações, a habilidade de raciocinar.

O JESS usa um algoritmo especial chamado RETE para o casamento das regras com os fatos, o RETE faz ao JESS mais rápido do que um simples conjunto de sentenças *if...then* (Friedman-Hill, 2003). Os métodos de inferência do JESS são de dois tipos de encadeamento (direto e reverso), porém duas estratégias de busca, de uso gratuito para instituições de ensino. A construção do SE pode ser realizada através do *prompt* do JESS ou editor de texto. O JESS versão 7.1 é usado neste trabalho para o desenvolvimento do SE.

O JESS é utilizado em diversas aplicações, mas o uso do JESS com a tecnologia dos *Applets* deixa o sistema muito pesado. Por isso quando a idéia é utilizar aplicações com JESS via navegador, devemos considerar o uso do JESS do lado do servidor, como o que ocorre no caso dos *Servlets*, dispensando o usuário de carregar grande parte do sistema para sua máquina o que torna a interação com o sistema bastante lenta e entediante do ponto de vista do usuário.

Segundo Bittencourt (2001) os SE foram desenvolvidos com base nos sistemas de produção, que é um nome genérico para todos os sistemas baseados em regras de produção. As regras consistem em pares de expressões simbólicas consistindo em uma condição e uma ação correspondente e são utilizadas para solucionar problemas de diagnóstico, predição, classificação e reconhecimento de padrões (Friedman-Hill, 2003). As regras de produção devem ser geradas em uma linguagem declarativa como o CLIPS.

### **3.2.3 Regras de produção.**

As regras de produção são do tipo: “Se CONDIÇÃO Então AÇÃO”, uma condição são antecedentes como fatos ou objetos e uma ação são conseqüentes que resultam em novos fatos ou objetos que darão seqüência ao encadeamento de regras. Em um SE baseado em regras de produção, existem basicamente dois modos de raciocínio possíveis associados ao

motor de inferência: “encadeamento direto ou progressivo” e o “encadeamento reverso ou regressivo”. No encadeamento direto, inicia-se com os dados disponíveis e usa o ciclo de inferência para extrair mais dados até o objetivo ser alcançado. No encadeamento reverso, inicia-se com a lista de objetivos ou hipóteses trabalhando em reverso desde o conseqüente até o antecedente para verificar se existem dados disponíveis que suportarem qualquer das conseqüências.

### 3.3 REDES NEURAIS ARTIFICIAIS

Com o intuito de simular a capacidade humana de aprendizagem, através de estruturas semelhantes à rede neural biológica, surgiram as Redes Neurais Artificiais (RNAs). Estas, por sua vez, são uma forma de programação não algorítmica, baseada em processamento distribuído paralelo de suas unidades, os neurônios artificiais (Haykin, 2001). Sendo possível modelar computacionalmente as conexões neurais, surge a idéia de que também será possível fazer emergir comportamentos inteligentes em máquinas (Vieira e Roisenberg, 2003).

A Figura 3.4 representa um modelo de neurônio artificial. As entradas, correspondentes aos dendritos dos neurônios biológicos, estão representadas pelas variáveis ( $X_n$ ). As ligações sinápticas entre os axônios dos outros neurônios e os dendritos deste são representadas pelos pesos ( $W_{k,n}$ ). A função de ativação ( $\varphi$ ) processa o somatório das entradas ponderadas pelos seus respectivos pesos, para produzir a saída final do neurônio ( $y_k$ ). O valor desta saída é enviado aos demais neurônios. Os pesos dos neurônios artificiais após o processo de aprendizagem armazenam o conhecimento adquirido pela rede (Braga *et al.* 2000, Haykin 2001 e Kovács 2002).

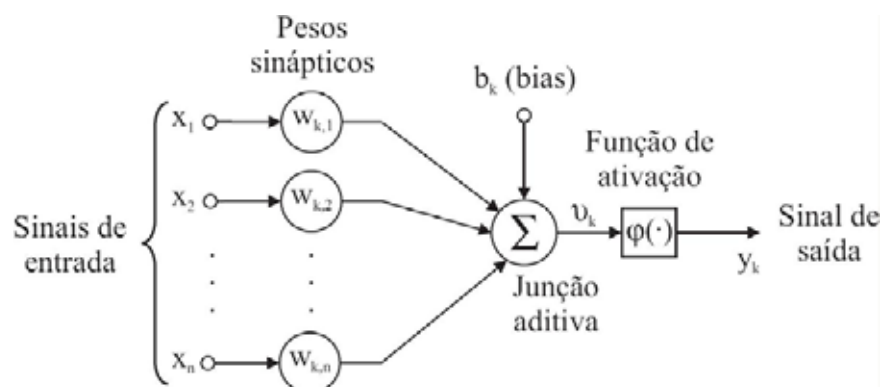


Figura 3.4- Modelo de neurônio artificial (Amaya *et al.* 2007b, modificado)



A partir do modelo de um neurônio artificial podem ser construídas as RNAs, que são neurônios artificiais interconectados simulando a estrutura neural biológica. Diversas topologias foram propostas para diferentes tipos de aplicação. Uma construção bastante popular no domínio de reconhecimento de padrões é a rede multicamada, utilizando o algoritmo de aprendizado *backpropagation*, que acabou virando sinônimo do nome da rede em si. Este esquema é apresentado na Figura 3.5.

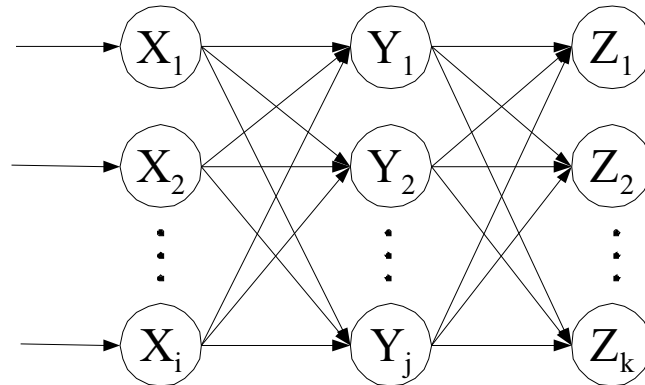


Figura 3.5- Rede neural artificial multicamada

Embora este modelo atenda razoavelmente as capacidades citadas em relação às unidades constituintes do cérebro, ele apresenta problemas quanto aos requisitos da memória como faculdade mental: o aprendizado incremental e as duas classificações, de curto e longo termo. Uma vez treinada, se apresentado um exemplo novo, sozinho, à rede, informações anteriores podem ser perdidas no processo.

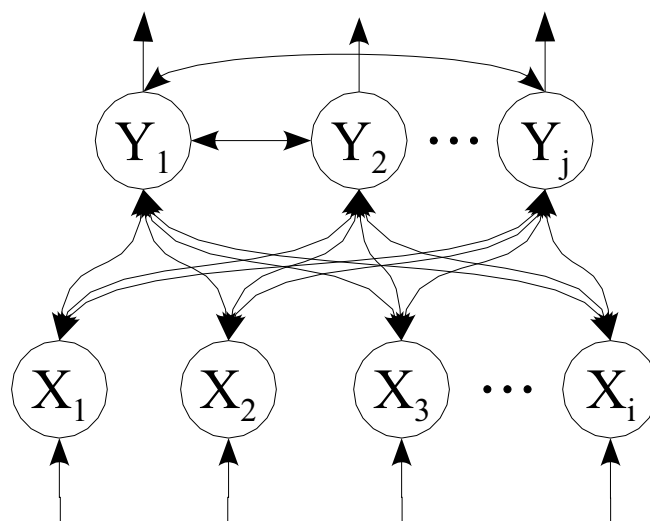


Figura 3.6- Modelo ART esquematizado

Uma outra família de topologias, as redes denominadas ART (*Adaptative Resonance Theory*) (Carpenter e Grossberg, 1987), parecem resolver satisfatoriamente estes requisitos. O modelo ART é apresentado na Figura 3.6, e consiste de um modelo de rede neural recorrente (com realimentação). A camada de reconhecimento (Y) classifica a entrada, resultando o neurônio com maior valor na função de ativação. Esta envia os resultados de volta à camada de comparação (X), capaz de avaliar se a classificação escolhida corresponde razoavelmente à entrada. Esta comparação resulta na classificação definitiva ou na criação de um novo neurônio na camada de reconhecimento, para este novo exemplo.

A teoria de Ressonância Adaptativa, ou ART, foi introduzida como uma teoria sobre o processamento cognitivo de informações no cérebro humano. Essa teoria levou ao desenvolvimento de uma série de modelos de redes neurais capazes de um aprendizado não supervisionado para classificação de padrões em tempo real. Os modelos nessa família compreendem: a rede ART1, que pode aprender a categorizar padrões de entrada binários apresentados em ordem arbitrária; a rede ART2, que pode aprender a categorizar padrões de entrada analógicos ou binários; e a rede ART3, que pode fazer uma busca paralela, ou teste de hipóteses, em códigos com reconhecimento distribuído.

### **3.4 LÓGICA NEBULOSA**

A razão humana é capaz de extrair de um amplo conjunto de informações apenas o que é relevante na solução do problema, com um mínimo grau de precisão. Para tratar de problemas que o ser humano pode resolver, é preciso uma abordagem que não utilize a precisão, o rigor e o formalismo matemático, e sim uma abordagem tolerante às falhas e verdades parciais (Zadeh, 1973).

A Lógica Nebulosa (ou Lógica *Fuzzy*), baseada nos Conjuntos Nebulosos (Zadeh, 1965), permite este tratamento baseado em valores qualitativos e não quantitativos, utilizando variáveis lingüísticas e não numéricas, para representar o problema e as regras utilizadas para resolvê-lo. Ainda que utilize estas etiquetas de linguagem, esta abordagem permite tratar de maneira categórica algumas questões imprecisas ou mal definidas.

Considerando-se uma determinada variável lingüística “temperatura”, os valores que ela poderia assumir seriam valores lingüísticos ou conjuntos nebulosos, exemplificados na

Figura 3.7. As soluções que se utilizam de Computação Nebulosa tipicamente são modeladas através de conhecimento especialista sobre o problema, e este conhecimento é representado e passado através de regras lingüísticas, do tipo: “SE x é A ENTÃO y é B”, onde “x é A” é o antecedente e “y é B” é o conseqüente. Quando o antecedente ou o conseqüente são asseverações lingüísticas, então esta regra é dita lingüística ou nebulosa.

A partir de variáveis e regras lingüísticas podem-se construir sistemas para solução de problemas de forma lingüística ou simbólica, que representam o comportamento humano de forma mais fiel que os sistemas que trabalham em termos numéricos.

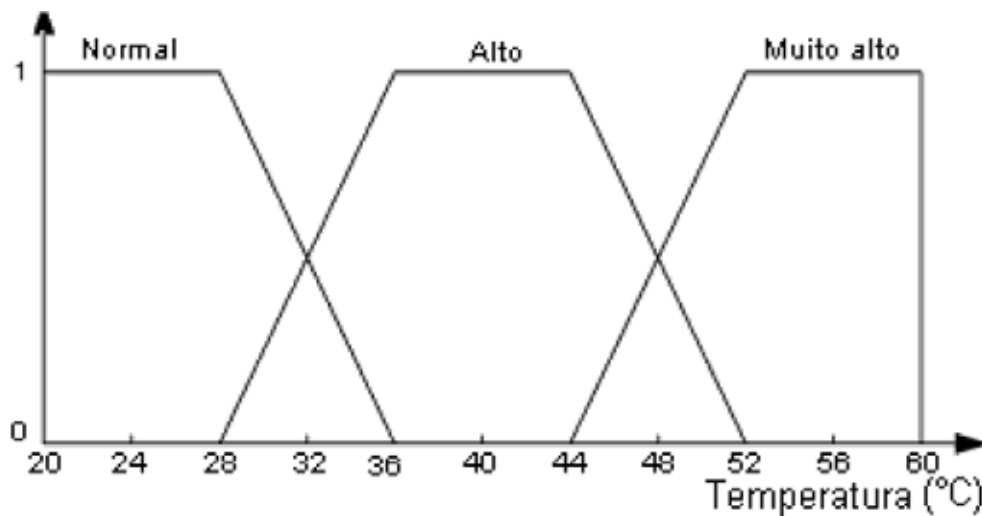


Figura 3.7- Processo de fuzificação da variável temperatura

Em suma, a lógica nebulosa permite a realização da “Computação com Palavras” (Zadeh, 1996). É possível, através do conhecimento de um especialista sobre o comportamento de um sistema, definir conjuntos e regras nebulosas, qualitativas, que podem resultar em respostas nebulosas ou categóricas (quantitativas).

Essa abordagem nos leva a um modelo computacional capaz de tratar, pelo menos em parte, do comportamento mental, que, conforme visto na seção de memória e mente, pode ser descrito em grande parte através da linguagem. Um modelo que fosse capaz de unir estas características com as questões estruturais, dos neurônios artificiais, poderia se parecer muito com o mecanismo de memória dos seres humanos.

### 3.5 MODELO HÍBRIDO

Os sistemas neurais nebulosos mostraram muitas vantagens na solução de problemas em aplicações reais. Alguns exemplos de sistemas neurais nebulosos para problemas de classificação de padrões são: *fuzzy* baseado em conhecimento e RNA multicamadas (Mitra *et al.*, 1997), sistemas neurais nebulosos (Vuorimaa *et al.*, 1995), *fuzzy* min-max RNA (Simpson, 1992), rede neural *fuzzy* ART (Carpenter *et al.*, 1991), *fuzzy* ARTMAP (Carpenter *et al.*, 1992), rede neural gaussiana ARTMAP (Williamson, 1996), e rede neural RBF *fuzzy* ARTMAP (Tontini e de Queiroz, 1996).

A incorporação de elementos de lógica nebulosa no modelo ART clássico, possibilitou o tratamento analógico de imprecisão, característicos da maneira como a linguagem representa o mundo. Os modelos conhecidos como *Fuzzy* ART (Carpenter *et al.*, 1991) possuem tais características. A proposta de construção de um ser artificial com memória consiste, portanto, na construção de um modelo desta natureza.

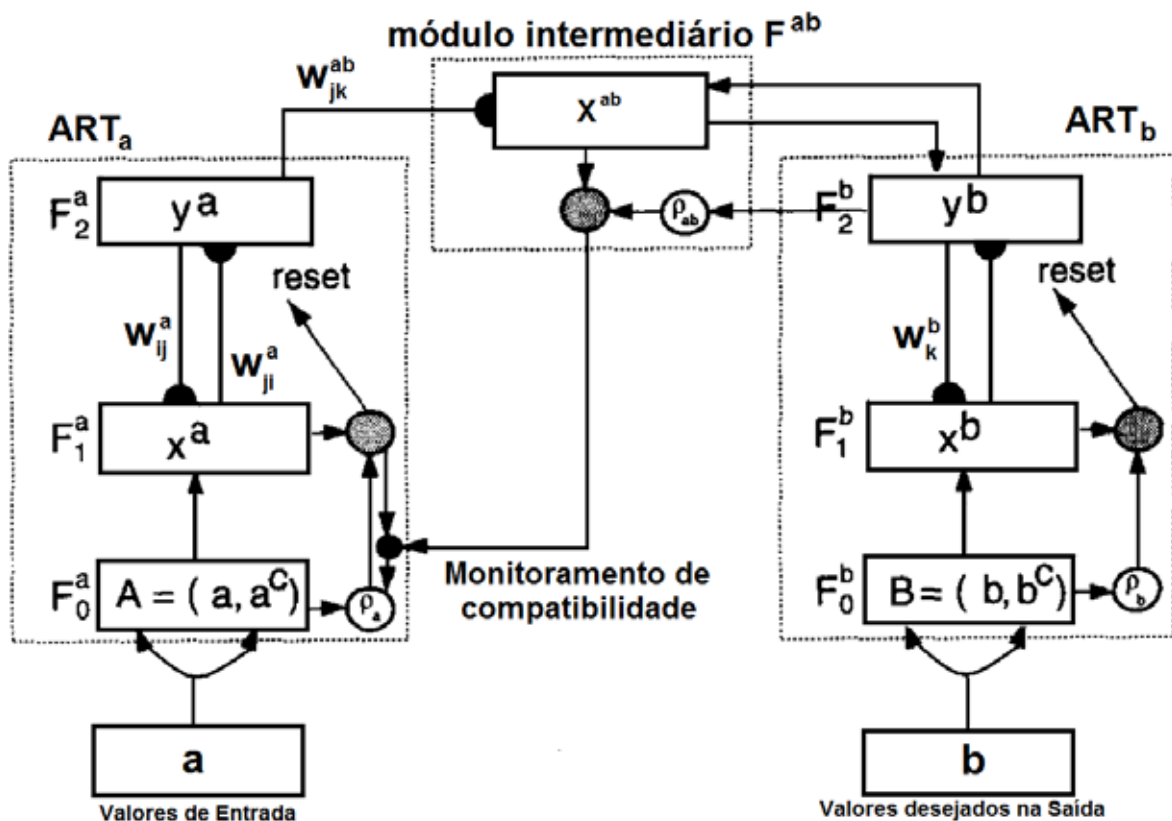


Figura 3.8- Modelo FAM (Carpenter *et al.*, 1992, modificado).

O desenvolvimento do modelo FAM (*Fuzzy* ARTMAP), ilustrada na Figura 3.8 permitiu a adaptação da rede ARTMAP para a utilização de padrões analógicos tanto na entrada como

na saída. A rede *Fuzzy*-ARTMAP é uma generalização da rede binária ARTMAP. Ela é capaz de um aprendizado supervisionado incremental, atualizando-se durante a operação sem “esquecer” o que já aprendeu anteriormente. A rede *Fuzzy*-ARTMAP pode ser empregada para classificação e/ou associação de padrões binários e/ou analógicos de entrada e saída com dimensão arbitrária. Este modelo é constituído de dois módulos *Fuzzy* ART, ART<sub>a</sub> e ART<sub>b</sub>, ligados por um módulo intermediário F<sup>ab</sup>. As operações realizadas internamente na propagação dos sinais são alteradas para as operações definidas pela lógica nebulosa, trabalhando com conjuntos e operadores nebulosos.

O módulo ART<sub>a</sub> realiza o reconhecimento dos valores de entrada e a ART<sub>b</sub>, dos valores desejados na saída. O módulo de interconexão é utilizado no treinamento para mapear entrada e saída.

Segundo Chen (1996), o modelo FAM é superior em acertos e compreensão em relação a outras técnicas com redes neurais. As características destacadas são:

- O aprendizado é rápido e ele suporta reconhecimento de eventos raros.
- Existe um dispositivo de memória para possibilitar o aprendizado incremental.
- A decisão do número de neurônios não é empírica, modelo é que as calcula.
- A tradução de um modelo FAM para um conjunto de regras lingüísticas (extração de regras dos pesos dos neurônios) pode ser realizada.

Estas características levaram à escolha neste trabalho deste modelo para prognósticos de falhas, por atender, ainda que parcialmente, diversas características de aprendizado, classificação e associação de variáveis.

### 3.5.1 Arquitetura FAM

A arquitetura FAM simplificada consiste de quatro camadas de nós como é mostrada na Figura 3.9. A primeira camada  $F_0^a$  é uma camada de pré-processamento das entradas, consiste de  $M$  nós (onde  $M$  é a dimensão do conjunto de entrada). Aceita um vetor de entrada  $a$  de dimensão  $M$  e converte esta entrada em um vetor de saída  $A$  de dimensão  $2M$ , ilustrada na Equação 3.1.

$$I = A = (a, a^c) = (a_1, \dots, a_M, a_1^c, \dots, a_1^c, \dots, a_M^c) \quad (3.1)$$

$$\text{Onde: } a_i^c = 1 - a_i; \quad 1 \leq i \leq M$$

Cada componente do vetor de entrada deve ser normalizado ao intervalo  $[0, 1]$  antes de ser aplicado a  $F_0^a$ . Este processo de converter a entrada  $a$  de dimensão  $M$  a um vetor de saída de dimensão  $2M$  é chamada de “codificação complementar”. O cálculo do complemento auxilia na maior representatividade do sistema, oferecendo o que pertence ao domínio observado e o que não pertence à rede. A codificação complementar é uma operação necessária para a operação satisfatória da FAM.

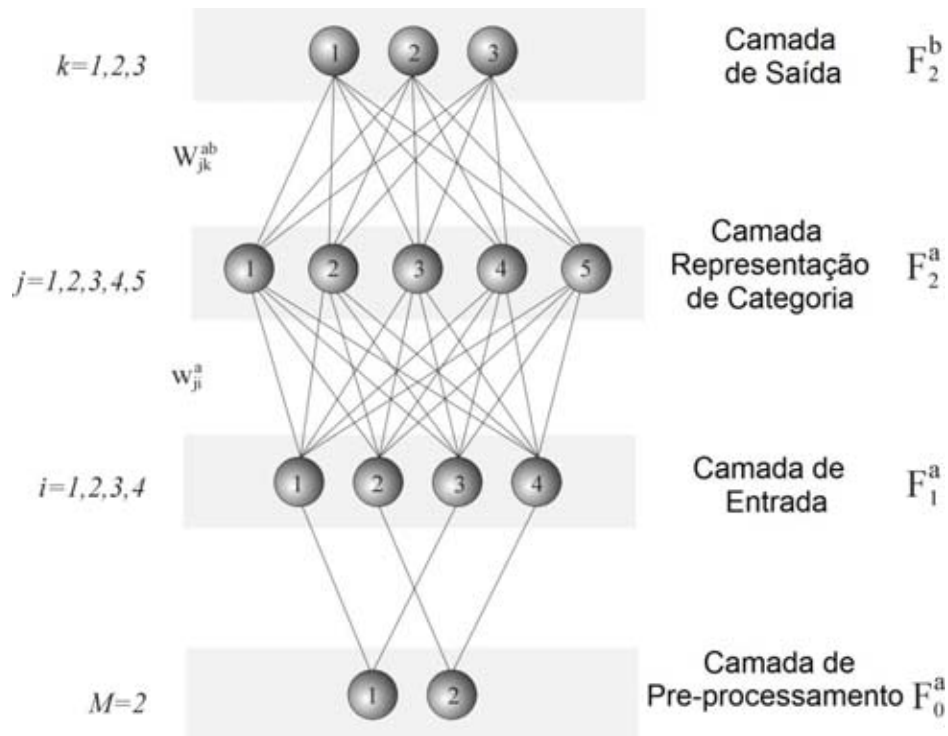


Figura 3.9- Arquitetura FAM simplificada.

A segunda camada da arquitetura FAM é a camada de entrada  $F_1^a$ , é nesta camada que as entradas  $A$  originadas desde  $F_0^a$  são aplicadas. Esta camada tem  $2M$  nós refletindo a dimensão do vetor de entrada  $A$ . Cada nó da  $F_1^a$  é indicado desde 1 até  $2M$ , onde um índice representa unicamente um nó da  $F_1^a$ .

A terceira camada da arquitetura FAM  $F_2^a$  é de representação de categoria. Esta camada tem  $N_a$  nós. O valor de  $N_a$  muda no processo de treinamento da FAM estabelecendo-se em um valor ao final do treinamento. O valor de  $N_a$  inicia-se em 1, este valor é acrescentado pela arquitetura FAM e as regras que governam o funcionamento. Cada nó nesta camada é chamado de categoria. Uma categoria é na realidade a representação comprimida de um

grupo de padrões similares. Os nós da  $F_2^a$  são indicados desde 1 até  $N_a$ , onde um índice representa unicamente um nó da  $F_2^a$ .

A quarta camada da arquitetura FAM é a camada de saída  $F_2^b$ , e tem exatamente  $N_b$  nós, onde  $N_b$  representa o número das classes de saída. Se treinarmos uma RNA para reconhecer os caracteres 'A', 'B', 'C' e 'D'; teríamos quatro classes de saída, então teríamos 4 nós na camada de saída ( $N_b = 4$ ). Como qualquer RNA, a arquitetura FAM incorpora aprendizagem através dos seus pesos de interconexão. A arquitetura FAM tem diferentes grupos de pesos.

Os pesos (conexões)  $W_{ij}^a$  que iniciam em  $F_1^a$  e convergem a  $F_2^a$ , são pesos das conexões ascendentes. A primeira letra  $i$  representa o índice do nó da  $F_1^a$ , e a segunda letra  $j$  é o nó da  $F_2^a$ .

Os pesos  $W_{ji}^a$  que iniciam em  $F_2^a$  e convergem a  $F_1^a$ , são chamados pesos descendentes. A primeira letra  $j$  representa o índice do nó da  $F_2^a$ , e a segunda letra  $i$  é o nó da  $F_1^a$ . O vetor de pesos de origem do nó  $j$  da  $F_2^a$  (vetor  $w_j^a = (w_{j1}^a, w_{j2}^a, \dots, w_{j,2M}^a)$ ) é chamado de padrão. Este tem um significado especial porque ele representa o conjunto de entradas aplicado à camada de entrada, que escolhe o nó  $j$  da camada de representação de categoria, como nó representativo. Em outras palavras  $w_j^a$  representa a forma comprimida de todas as entradas juntas.

Os pesos descendentes e ascendentes são relacionados linearmente. Porém o funcionamento da FAM pode ser modelado só um desses conjuntos de pesos. O conjunto de pesos escolhido para modelar são os pesos descendentes  $W_{ji}^a$ .

Os pesos de interconexão originados desde cada nó  $j$  da camada representação de categoria até cada nó da camada de saída  $F_2^b$ . Eles são chamados pesos *inter-ART* e são denotados por  $W_{jk}^{ab}$ . A primeira letra  $j$  representa o índice do nó da camada de origem  $F_2^a$ , e a segunda letra  $k$  representa o índice do nó da camada convergente  $F_2^b$ .

A FAM opera em duas fases, fase de treinamento e fase de desempenho (teste). Na fase de treinamento é apresentado à FAM um conjunto de pares entradas/saídas  $\{(I^1, O^1), \dots, (I^r, O^r), \dots, (I^{PT}, O^{PT})\}$ , mapeando cada entrada com sua correspondente saída. No processo de treinamento,  $I^l$  é apresentado a  $F_1^a$  e  $O^l$  é apresentado a  $F_2^b$ , e depois  $I^2$  e  $O^2$ , e assim até o ultimo padrão do conjunto de treinamento  $I^{PT}$  e  $O^{PT}$ . Este conjunto de treinamento é apresentado repetidamente à FAM, até que cada entrada seja corretamente mapeada com sua respectiva saída. O treinamento é considerado completo quando os pesos não cambiam durante a apresentação da lista. Este processo é chamado aprendizado *off-line*.

Na fase de desempenho a FAM gera uma saída para a lista de padrões  $\tilde{I}^1, \tilde{I}^2, \dots, \tilde{I}^{PS}$ . Esses padrões são apresentados na camada  $F_1^a$  e a saída é observada na camada  $F_2^b$ .

Antes de discutir detalhes das fases de treinamento e desempenho, explica-se a função e propriedades de cada parâmetro da rede. Há dois parâmetros definidos pelo usuário. Estes são o parâmetro de escolha  $\beta_a$ , e o parâmetro padrão de vigilância  $\bar{\rho}_a$ . O parâmetro de escolha  $\beta_a$  com valores no intervalo  $(0, \infty)$  influencia as entradas ascendentes aplicadas aos nós da camada  $F_2^a$ , quando um padrão é apresentado à camada de entrada. O parâmetro padrão de vigilância  $\bar{\rho}_a$  tem valores no intervalo  $[0, 1]$ . Valores pequenos de  $\bar{\rho}_a$  (perto de 0) resultam na rede agrupamento desigual dos padrões, criando grupos grossos. Valores grandes de  $\bar{\rho}_a$  (perto de 1) resulta na rede agrupamento de padrões muitos similares, criando muitos grupos pequenos. Estes grupos são nós de categoria criados na  $F_2^a$ .

Outros dois parâmetros da rede são ajustados pela própria FAM. Eles são o parâmetro de vigilância  $\rho_a$  e o número de nós  $N_a$  na camada representação de categoria. O parâmetro de vigilância  $\rho_a$  tem valores no intervalo  $[\bar{\rho}_a, 1]$ . Por default o valor é igual a  $\bar{\rho}_a$  prévio à apresentação dos pares de entrada/saída da FAM. O ultimo parâmetro a discutir é  $N_a$ , número de nós na camada de representação categórica mais 1. O aumento de 1 representa um nó não comprometido na camada  $F_2^a$ . O nó não comprometido é o nó que não tem nenhum código dos padrões de entrada. Antes do inicio do treinamento o valor deste



parâmetro é inicializado em 1 e acrescentado depois segundo as regras da fase de treinamento da FAM.

Para propósito de cambio e modificação dos pesos é introduzida a operação ( $\wedge$ ) “fuzzy-min” dos dois vetores  $P$  e  $Q$ , designado como  $P \wedge Q$ . A operação *fuzzy-min* de dois vetores  $P$  e  $Q$  é um vetor cujos componentes são iguais ao mínimo dos componentes  $P$  e  $Q$ , por exemplo, se  $P=[0.2 \ 0.7]$  e  $Q=[0.1 \ 0.8]$ , então  $P \wedge Q = [0.1 \ 0.7]$ .

### 3.5.2 Treinamento da FAM.

A implementação passo-a-passo do treinamento *off-line* da FAM proposto por Bharadwaj (2003) é apresentada a seguir:

1. Definir o parâmetro de rede  $\beta_a$  no intervalo  $(0, \infty)$ ,  $\bar{\rho}_a$  no intervalo  $[0, 1]$  e  $\varepsilon$  no intervalo  $[0, 1]$ . Os valores iniciais dos pesos descendentes ( $w_{ji}^a$ ;  $j = 1, \dots, N_a$ ,  $i = 1, \dots, 2M$ ) são escolhidos iguais a 1. Os valores iniciais dos pesos inter-ART ( $W_{jk}^{ab}$ ;  $j = 1, \dots, N_a$ ,  $k = 1, \dots, N_b$ ) iguais a 0. O número de nós da camada  $F_1^a$  é definida igual a  $2M$ . O número de nós da camada  $F_2^a$  é definida igual a 1. Antes do treinamento o único nó na camada  $F_2^a$  é o nó não comprometido. O número de nós na camada  $F_2^b$  ( $N_b$ ) é definido igual ao número de classes de saída. O índice  $r$  dos pares de entrada/saída é definido igual a 1.
2. O padrão  $r$  atual ( $I^r, O^r$ ) é apresentado à FAM. O padrão de entrada  $I^r$  é apresentado à camada de entrada  $F_1^a$  e a saída  $O^r$  é apresentada a  $F_2^b$ . O valor do parâmetro de vigilância  $\rho_a$  é definido igual ao valor de  $\bar{\rho}_a$ .
3. Calcular as entradas ascendentes em todos os nós da camada  $F_2^a$  da FAM devido à apresentação de padrões de entrada  $I^r$  na camada de entrada. Todos os nós e o nó não comprometido são incluídos quando os valores ascendentes são calculados. As entradas ascendentes para o nó  $j$  na  $F_2^a$  são calculadas de acordo à Equação 3.2.

$$T(w_j^a | I^r) = \frac{|I^r \wedge w_j^a|}{\beta_a + |w_j^a|} \quad (3.2)$$

4. Escolher o nó da  $F_2^a$  aquele que recebe a máxima entrada ascendente. Em outras palavras, escolher o nó com o maior valor  $T(w_j^a | I^r)$ . É possível assumir que o

índice do nó escolhido é  $j_{max}$ . O nó  $j_{max}$  satisfaz o critério de vigilância se:

$$\frac{|I^r \wedge w_{j_{max}}^a|}{|I^r|} \geq \rho_a$$

Escolhido o nó, temos três casos a considerar para verificar se passa o teste de vigilância.

- a. Se o nó  $j_{max}$  é um nó não comprometido, por default satisfaz o critério de vigilância e pode continuar com o passo 5.
  - b. Se o nó  $j_{max}$  é um nó comprometido e se satisfaz o critério de vigilância e pode continuar com o passo 5.
  - c. Se o nó  $j_{max}$  não satisfaz o critério de vigilância, então desqualificar o nó da competição, colocando  $T(w_{j_{max}}^a | I^r) = -1$ , e voltar ao passo 4.
5. Depois da prova de vigilância, podemos diferenciar três casos:
- a. Se o nó  $j_{max}$  é um nó não comprometido e a saída  $\mathbf{O}^r$  dos pares entrada/saída ( $I^r, \mathbf{O}^r$ ) é tal que seu componente  $k_{max}$  é um e os demais são zeros, então definir  $W_{j_{max}, k_{max}}^{ab} = 1$ . Além disto, o vetor de pesos descendentes  $w_{j_{max}}^a$  torna-se igual a  $w_{j_{max}}^a \wedge I^r$ . Depois de efetuado esses câmbios nos pesos ir ao passo 6.
  - b. Se o nó  $j_{max}$  é um nó comprometido e a saída  $\mathbf{O}^r$  dos pares entrada/saída ( $I^r, \mathbf{O}^r$ ) é tal que seu componente  $k_{max}$  é um e os demais são zeros, e ao mesmo tempo  $W_{j_{max}}^{ab}$  é tal que seu componente  $k_{max}$  é um e os demais são zeros, então a saída desejada  $\mathbf{O}^r$  é compatível com a saída atual, representado por  $W_{j_{max}}^{ab}$ . Agora o vetor de pesos descendentes  $w_{j_{max}}^a$  torna-se igual a  $w_{j_{max}}^a \wedge I^r$ . Depois de efetuado esses câmbios nos pesos ir ao passo 6.
  - c. Se o nó  $j_{max}$  é um nó comprometido e a saída  $\mathbf{O}^r$  dos pares entrada/saída ( $I^r, \mathbf{O}^r$ ) é tal que seu componente  $k_{max}$  é um e os demais são zeros, enquanto ao mesmo tempo  $W_{j_{max}}^{ab}$  é tal que um componente diferente a  $k_{max}$  é um e os demais são zeros, então a saída desejada  $\mathbf{O}^r$  não é compatível com a saída atual, representado por  $W_{j_{max}}^{ab}$ . Neste caso o nó  $j_{max}$  é substituído por  $T(w_{j_{max}}^a | I^r) = -1$ , o nível de vigilância é acrescentado ao valor de

$\frac{|I^r \wedge w_{j_{\max}}^a|}{|I^r|}$ , deve-se voltar ao passo 4 para encontrar outro nó  $j_{\max}$  que

maximize as entradas ascendentes e satisfaz a vigilância, até que ao mesmo tempo predizemos a saída correta.

6. Se todos os pares entrada/saída foram apresentados chegaremos a uma época,  $r$  é acrescentado a  $r+1$  e voltamos ao passo 2 a apresentar o par de entrada/saída  $r+1^{\text{th}}$ . Se todos os pares de entrada/saída fossem apresentados então dois casos podem ser observados.
  - a. Na apresentação da lista prévia o ultimo componente dos pesos ascendentes ou inter-ART foram mudados. Neste caso voltamos ao passo 2 e apresentamos o primeiro par de entradas/saídas, declarando  $r$  igual a 1.
  - b. Na apresentação da lista prévia não ocorre câmbios nos pesos descendentes e inter-ART. Então o treinamento é considerado completo.

Depois de completado o processo de treinamento, os pesos  $w_{ji}^a$ ;  $j = 1, \dots, N_a$ ,  $i = 1, \dots, 2M$ , e  $W_{jk}^{ab}$ ;  $j = 1, \dots, N_a$ ,  $k = 1, \dots, N_b$ , são armazenadas para ser usados na fase de desempenho. Além disto, os parâmetros de rede  $\bar{\rho}_a$  e  $\beta_a$  são declarados iguais ao valor que eles tiveram na fase de treinamento.

### 3.5.3 Pseudocódigo para treinamento da FAM

FOR (cada padrão de treinamento)

{

1.  $S = \{\text{Função de Escolha de Categoria (FEC)} \text{ valor de cada categoria na } F_2^a\}$ ;
2. Selecionar a categoria com o **máximo valor FEC** de S;
3. Executar o **teste de vigilância** na categoria selecionada;
4. IF (a categoria selecionada passa o teste de vigilância)
  - a. IF (a categoria selecionada tem a mesma classe rotulada)
 

permitir ao padrão codificação categórica; continuar com o próximo padrão;
  - b. ELSE
 

Iniciar o mecanismo de monitoramento de compatibilidade;

5. IF (nenhuma categoria passa o teste de vigilância)  
 executar o nó não comprometido;  
 }

#### FUNÇÃO DE ESCOLHA DE CATEGORIA (FEC)

$$T(w_j^a | I) = \frac{|I \wedge w_j^a|}{\beta_a + |w_j^a|} \Leftrightarrow T(w_j^a | I) = \frac{M - |R_j^{a,old}| - dis(x, w_j^{a,old})}{\beta_a + M - |R_j^{a,old}|}$$

$x = a$  = padrão de treinamento apresentado.

$$I = (x, x^c)$$

$w_j^a$  = índice de categoria  $j$ , significa que  $w_1^a$  representa a primeira categoria,  $w_2^a$  a segunda categoria e assim pela frente.

$\beta_a$  = parâmetro de escolha. Toma valores maior ou igual a 0.  $\beta_a \geq 0$

$M$  = Dimensão do espaço de entrada

$$|R_j^a| = \sum_{m=1}^M (v_{jm}^a - u_{jm}^a) = \text{Tamanho da categoria.}$$

$$dis(x, w_j^a) = \sum_{m=1}^M [(\max\{x_m, v_{jm}\} - v_{jm}) + (u_{jm} - \min\{x_m, u_{jm}\})] = \text{distancia do ponto } x \text{ da}$$

categoria  $j$ .

#### TESTE DE VIGILÂNCIA OU FUNÇÃO DE COMPATIBILIDADE DE CATEGORIA

$$\rho(w_j^a | I) = \frac{|I^r \wedge w_{j_{\max}}^a|}{|I^r|} \geq \rho_a \Leftrightarrow \rho(w_j^a | I) = \frac{M - |R_j^{a,old}| - dis(x, w_j^{a,old})}{M} \geq \rho_a$$

$\rho_a$  = Parâmetro de vigilância. Toma valores entre 0 e 1.  $0 \leq \rho_a \leq 1$

### 3.5.4 Desempenho da FAM

A implementação passo-a-passo da fase de desempenho proposto por Bharadwaj (2003) é apresentada a seguir:

1. Inicializar os pesos  $w_{ji}^a$ ;  $j = 1, \dots, N_a$ ,  $i = 1, \dots, 2M$ ,  $W_{jk}^{ab}$ ;  $j = 1, \dots, N_a$ ,  $k = 1, \dots, N_b$ , aos valores que eles tiveram ao final da fase de treinamento da FAM. Os parâmetros da rede  $\bar{\rho}_a$ ,  $\beta_a$ ,  $D$ ,  $\mu$  e  $\omega$  são escolhidos igual aos valores que eles

tiveram durante a fase de treinamento. O valor do parâmetro de vigilância  $\rho_a$  é definido igual ao valor padrão de vigilância  $\bar{\rho}_a$ .

2. Apresentar a  $r$ -th padrão de teste (i.e., padrão de teste  $\tilde{I}^r$ ) à FAM. Este padrão de teste é aplicado à camada de entrada  $F_1^a$ .
3. Calcular as entradas ascendentes desde  $F_1^a$  a todos os nós na camada  $F_2^a$  devido à apresentação dos padrões de teste  $\tilde{I}^r$ . Durante o cálculo todos os nós incluindo os nós não comprometidos devem ser considerados. As entradas ascendentes são calculadas baseadas na Equação 3.3.

$$T(w_j^a | \tilde{I}^r) = \frac{|\tilde{I}^r \wedge w_j^a|}{\beta_a + |w_j^a|} \quad (3.3)$$

4. Escolher o nó na  $F_2^a$  que recebe a máxima entrada ascendente da  $F_1^a$ . Assumir que o índice do nó escolhido  $F_2^a$  é  $j_{max}$ . Verificar se o nó satisfaz o critério de vigilância. Para fazer isso temos três casos:

- a. Se o nó  $j_{max}$  é um nó não comprometido, ele satisfaz o teste de vigilância automaticamente. Ir ao passo 5.
- b. Se o nó  $j_{max}$  é um nó comprometido e satisfaz o critério de vigilância, ir ao passo 5. O nó  $j_{max}$  satisfaz o critério de vigilância se:

$$\frac{|\tilde{I}^r \wedge w_{j_{max}}^a|}{|\tilde{I}^r|} \geq \rho_a$$

- c. Se o nó  $j_{max}$  não satisfaz o critério de vigilância, desqualifica o nó da competição, definindo  $T(w_{j_{max}}^a | \tilde{I}^r) = -1$ , e depois vai a iniciar o passo 4.

5. Depois do teste de vigilância podemos ver três casos:

- a. Se o nó  $j_{max}$  é não comprometido, então a saída do padrão de teste apresentado é designado como “desconhecido”. Ir ao passo 6.
- b. Se o nó  $j_{max}$  é um nó comprometido, e  $W_{j_{max}, k_{max}}^{ab} = 1$ , embora o resto de pesos  $W_{j_{max}, k}^{ab}$  são iguais a 0, então designamos a saída da rede  $O^r$  como o vetor  $W_{j_{max}}^{ab}$ . Ir ao passo 6.

6. Se todos os padrões de teste no conjunto teste ainda não foram aplicados à rede voltar ao passo 2 e apresentar o próximo par de entradas/saídas na seqüência. Se já

foi apresentado todos os pares de entrada saída, então o resultado pode ser analisado para encontrar o erro de classificação e outros dados estatísticos.

### 3.6 INTELIGÊNCIA ARTIFICIAL APLICADA À MBC

Conforme foi discutido no capítulo 2, a MBC possui, intrinsecamente, um componente computacional relacionado a instrumentos e redes industriais. A sua relação com sistemas especialistas, e outras técnicas de IA como redes neurais, lógica nebulosa, algoritmos genéticos etc., vem da necessidade de tratar esses dados automaticamente e, principalmente, tomar decisões relacionadas ao diagnóstico e/ou prognóstico de defeitos e falhas em máquinas e equipamentos durante a sua fase de operação.

Através do uso de sensores e redes de campo pode-se conseguir monitorar as condições de máquinas. Um sistema remoto de monitoramento distribuído, diagnóstico e integração de informação foi realizado por Guo *et al.* (2007), estabelecendo um subsistema de monitoramento de condição baseado na tecnologia *fieldbus* cobrindo as diferenças de *software*, *hardware*, bando de dados e protocolos de rede.

Na MPd, tradicionalmente foram utilizados métodos estatísticos como, regressão linear, multi-regressivo linear, *kriging* dinâmico, etc. (Bevilacqua *et al.* 2003, You 1998 e Lucifredi *et al.* 2000). Com o avanço da ciência da computação e a IA, o uso de SE na manutenção foi mais viável e difundido.

Protótipo de SE para o apoio aos diagnósticos, prognósticos de falhas e tomada de decisão de manutenção de compressores centrífugos por meio do monitoramento *online* da condição é apresentada por Mecabô (2007). Silva (1998) apresenta o desenvolvimento de um protótipo de sistema especialista para projeto de sistemas hidráulicos, disponibilizando além dos resultados calculados, diagramas dos sistemas. O protótipo resolve e apresenta opções de soluções que exigiriam um tempo considerável do especialista. Aplicações voltadas ao projeto de unidades de potência hidráulica de sistemas industriais foram desenvolvidas por Vinade (2003) e Caletti (2003). O trabalho desenvolvido por Chrissanthi (2008) mostra o uso dos SE para diagnóstico de falhas *on-line* em processos técnicos.

Redes neurais artificiais têm sido utilizadas em vários problemas de monitoração de condição, diagnóstico e prognóstico de falhas em máquinas. Lucifredi *et al.* (2000), apresentaram a comparação dos modelos: multi-regressivo linear, *kriging* dinâmico e redes

neurais, aplicados em uma usina hidrelétrica. Garcia *et al.* (2006) desenvolveram uma aplicação software para diagnóstico em tempo real de processos industriais chamado de SIMAP (*Intelligent System for Predictive Maintenance*), coleta as informações em tempo real dos sensores e de outras fontes e as processa usando técnicas de inteligência artificial como SE e RNA.

Nos últimos anos surgiram sistemas de IA híbridos aplicados à MBC. Falqueto e Telles (2007) apresentaram um SE *fuzzy* para manutenção automática de transformadores elétricos de potência da usina hidrelétrica de Itaipu. Molina *et al.* (2000), desenvolveram um sistema usando SE e RNA, chamado MAPAIS: Abreviação do espanhol para sistema avançado de manutenção preditiva incorporando áudio e vídeo. O sistema foi projetado para prever alarmes da usina hidrelétrica de Villalcampo I e incluir a um sistema integrado de monitoramento chamado de HYPERVISION. Ciarapica e Giacchetta (2006) apresentam um sistema para diagnóstico e prognóstico de falhas em uma planta de potência de ciclo combinado usando RNA recorrentes e sistemas *neuro-fuzzy*. Javadpour e Knapp (2003) implementaram um sistema preditivo baseado em RNA e lógica nebulosa para auxílio aos operadores no diagnóstico de falhas com uma alta precisão de predição.

O uso da metodologia OSA-CBM proposto neste documento foi inicialmente usado por Fu *et al.* (2004), eles apresentaram uma metodologia usando três elementos importantes: Monitoramento e Previsão, Diagnóstico e Prognóstico, e Tomada de Decisão na Manutenção. Dunsdon e Harrington (2008) propõem a aplicação da arquitetura OSA-CBM, provendo uma solução abrangente votada para o gerenciamento integrado de saúde veicular IVHM (*Integrated Vehicle Health Management*) da empresa *GE Aviation*.

Outra técnica para a MBC foi elaborado por Guo *et al.* (2002), usando o conceito realidade virtual na MBC, usando ferramentas como *OpenGL*, mostram o resultado do processamento dos alarmes numa visão 3D (três dimensões).

### **3.7 CONCLUSÕES DO CAPÍTULO**

Neste capítulo foram estudados os conceitos de SE, RNA, lógica nebulosa, sistemas inteligentes híbridos, e pode-se concluir o seguinte:

O desenvolvimento e implementação de técnicas de IA para sistemas de MBC é provavelmente a mais promissora e flexível dentre as possíveis existentes atualmente.

O uso da técnica de SE apresenta maior facilidade de implementação e comunicação com o usuário final.

A ferramenta JESS é uma *Shell* que permite implementar um SE usando a linguagem Java.

A *Fuzzy* ARTMAP é uma rede com capacidade de aprendizado incremental, reconhecer novos padrões e adiciona-o a sua base de conhecimento sem precisar re-treinamento.

A aplicação de técnicas inteligentes como SE, RNA, Lógica nebulosa e sistemas inteligentes híbridos na MBC esta tendo bons resultados quanto ao diagnostico e prognostico de falhas iminentes.



## 4 METODOLOGIA

Este capítulo apresenta a metodologia concebida para implementação do sistema inteligente de manutenção baseada em condição usando a instrumentação FF, aquisição de dados através de servidores OPC e banco de dados. Os dados são processados utilizando técnicas de IA como SE baseado em regras de produção e o modelo *fuzzy* ARTMAP para prognóstico de falhas.

A metodologia é baseada no modelo de referência OSA-CBM aplicado à manutenção baseada em condição. Esta metodologia consiste de um conjunto de especificações, técnicas e algoritmos utilizados para a definição funcional das camadas do sistema. Esta metodologia especifica as sete camadas do modelo OSA-CBM e como estas interagem entre si, sendo apresentado na forma de diagramas IDEF0 e IDEF1X, para modelagem funcional e de informação, respectivamente.

### 4.1 INTRODUÇÃO

A metodologia proposta neste trabalho é concebida a partir do modelo de referência OSA-CBM apresentado em sete camadas. A integração das sete camadas é apresentada como uma transição lógica ou um fluxo de informação da saída dos sensores para a camada de tomada de decisão, através das camadas intermediárias. A camada de apresentação é uma exceção dentro do modelo OSA-CBM, pois permite comunicação ponto-a-ponto entre esta camada e qualquer outra.

O sistema é concebido em um modelo computacional cliente/servidor. No lado servidor são executadas as seis primeiras camadas do modelo OSA-CBM. A camada de apresentação é desenvolvida no lado cliente. As informações do servidor são solicitadas pelos clientes através de conexões *sockets* diretas utilizando a *web* como meio de comunicação.

A metodologia idealizada utiliza a tecnologia Internet para a comunicação dos clientes com o servidor. Este ambiente é globalizado, centrado em rede, e distribuído. Na interface do cliente utilizam-se navegadores baseados na *Web* e as linguagens HTML, PHP e Java, permitindo a independência da plataforma computacional do usuário.

Esta metodologia considera o armazenamento de dados das variáveis de processo, diagnóstico de saúde dos equipamentos e tomadas de decisão. O processo de aprendizado concebido na camada de prognóstico da metodologia é feito através das ocorrências passadas, mapeando estados de defeitos e falhas, e predizendo futuras anomalias usando o modelo FAM. O uso da FAM é devido à existência de relações não lineares entre as variáveis associadas que geram falhas ocultas. Os códigos de saída da FAM são processados pelo SE da camada tomada de decisão.

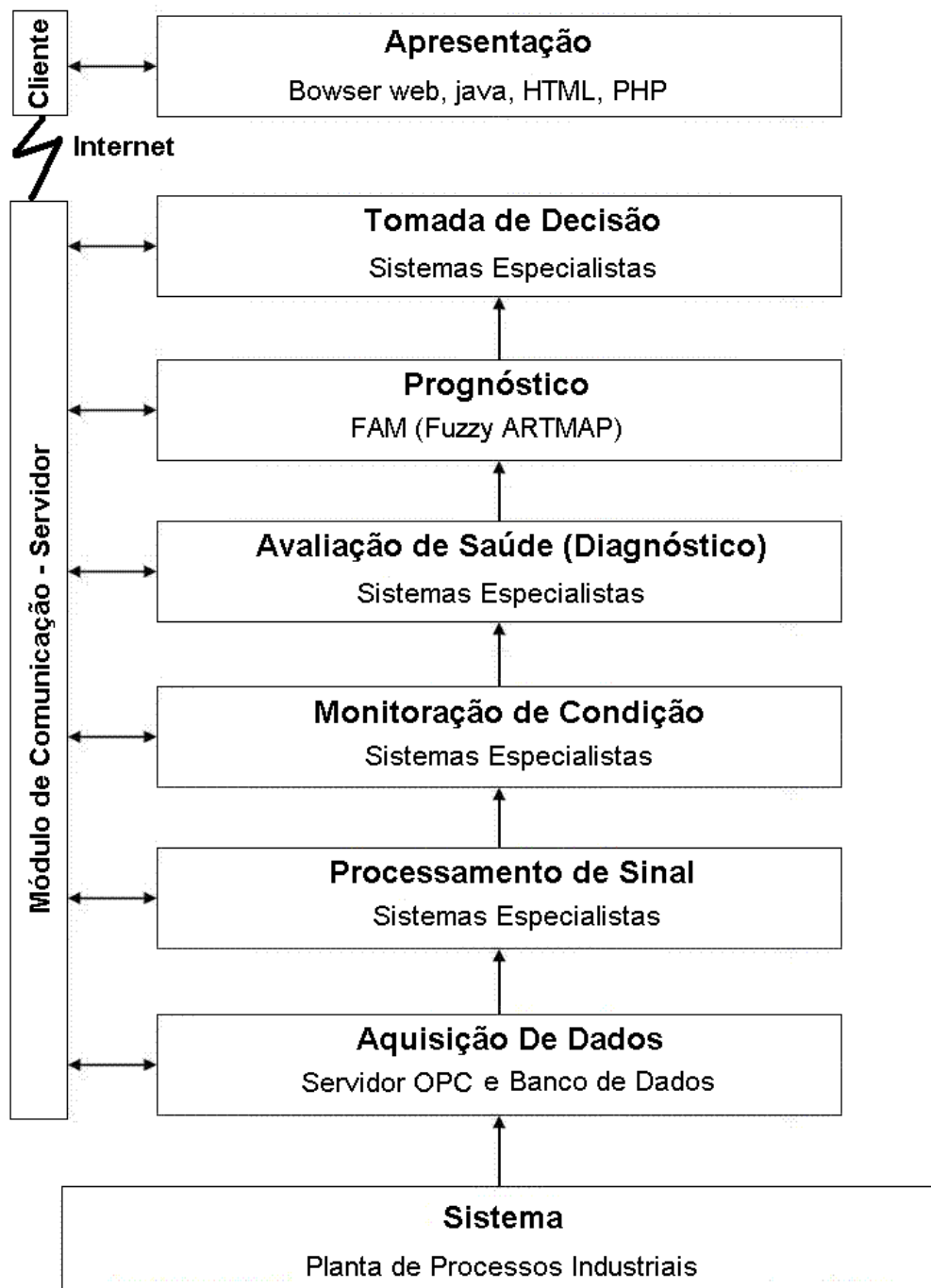


Figura 4.1- Técnicas de IA usada nas camadas OSA-CBM.

O procedimento inicia-se quando o usuário inicia o servidor, começando a aquisição de dados (itens), passando por todos os algoritmos das camadas intermediárias até chegar à camada tomada de decisão, este processo é repetido periodicamente. Uma vez iniciado o servidor, no lado cliente, o usuário inicia uma aplicação através de um navegador *browser* ou usando *telnet* através do *prompt*, começa a solicitar e visualizar as informações geradas pelas camadas mais baixas.

A metodologia pode ser aplicada para o desenvolvimento de sistemas de manutenção baseada em condição em plantas de processos industriais, desde que estes disponham de dados da instrumentação através de servidores OPC e/ou banco de dados.

## **4.2 MODELO DE REFERÊNCIA OSA-CBM**

A arquitetura utilizada para o desenvolvimento desta metodologia é apresentada na Figura 4.1, onde se ilustra a técnica de IA usada em cada uma das camadas e o fluxo de informação entre as camadas. A descrição inicia-se com o sistema a monitorar, continuando com as camadas desde o nível de aquisição de dados até chegar à camada de apresentação, sendo esta última a interface com o usuário

### **4.2.1 Aquisição de dados**

O processamento começa com a coleta de dados (itens) da instrumentação *fieldbus*, Os itens *online* são obtidos diretamente da instrumentação através de um ou mais servidores OPC. Outra forma de obter estes dados é através de bancos de dados via JDBC (*Java Database Connectivity*), desde que outros sistemas (*i.e.* sistema supervisorio SCADA) armazenem dados da instrumentação, condições de operação e avaliação de saúde, possibilitando o desenvolvimento de um sistema integrado com outros sistemas dentro de uma indústria.

### **4.2.2 Processamento de sinal**

Nesta camada, as variáveis digitais obtidas na camada aquisição de dados são processadas de modo a convertê-las numa forma específica capaz de representar a grandeza física que está sendo monitorada, caso necessário, efetuar cálculos matemáticos sobre ela. Contribuições para esta camada incluem informações da qualidade do sinal *fieldbus* que

indicam o estado de saúde da instrumentação, e qualidade de comunicação OPC que detecta o estado de conectividade com o servidor OPC, estes dados de qualidade são processadas por regras de produção do SE.

#### 4.2.3 Monitoração de condição

Esta camada determina a condição da planta, dos seus sistemas, subsistemas, ou componentes (excede limiar, ciclo de tensão, condição operacional, métrica de uso). O processamento desta camada é através de SE baseado em regras de produção, tendo como entradas as informações geradas pelas camadas de aquisição de dados e processamento de sinal. As saídas desta camada estão relacionadas às quatro faixas de operação: NORMAL, ALTO, ALARME, e TRIP, ilustrado na Figura 4.2.



Figura 4.2- Faixas de operação de um equipamento.

#### 4.2.4 Avaliação de saúde (diagnóstico)

Esta camada determina o estado da planta, sistema, subsistemas ou componentes monitorados baseados nas informações geradas pelas camadas anteriores, e de valores de referência. A avaliação ocorre através da extração das características de cada equipamento e posterior detecção de anomalias dos mesmos. As regras de produção geradas para esta camada incluem diagnóstico da instrumentação *fieldbus*, diagnóstico da comunicação OPC e diagnóstico do estado de operação dos equipamentos. A saída desta camada é um índice de estado do equipamento monitorado que é armazenado no banco de dados e mostrado na camada de apresentação através de sinais de alarme.

#### 4.2.5 Prognósticos

Nesta camada a predição de falhas é feita usando o histórico de anomalias, histórico de variáveis e às relações entre anomalias e suas variáveis associadas, uma abordagem é a aplicação de RNA, lógica nebulosa e sistemas híbridos. O uso do modelo *fuzzy* ARTMAP é a abordagem utilizada nesta metodologia devido às altas relações não lineares entre os dados e situações anormais. Enquanto um sistema especialista tenta imitar a resposta de um

operador analisando as mesmas variáveis, as redes neurais superam este limite e tentam analisar as relações não lineares entre os diferentes sinais (Molina *et al.*, 2000),

Na seleção da arquitetura de rede neural para aplicação ao prognóstico de máquinas, tomam-se em conta os seguintes critérios:

- Capacidade de adaptação incremental no tempo. Se novas informações são disponibilizadas, ou o sistema monitorado experimenta modificações. É essencial que o sistema monitorado seja capaz de se adaptar rapidamente a essas alterações.
- Treinamento rápido e estável. A RNA deve ser capaz de incorporar novas experiências com um mínimo de tempo de treinamento, sem apagar as experiências treinadas previamente.
- Capacidade de geração de hipóteses. O prognóstico, feito por um especialista, é só uma suposição. Raramente os estados internos de uma máquina são conhecidos o suficiente, para concluir desde um dado de sinal com absoluta certeza. Assim este sistema de prognóstico deve ter a capacidade de sugerir possibilidades de falha.

O desenvolvimento do modelo FAM precisa de duas fases, treinamento e desempenho. Na fase de treinamento, todas as anomalias passadas e os valores das suas variáveis associadas são requeridos. As variáveis associadas são as entradas da FAM e as saídas são os códigos de anomalias. Depois do processo de treinamento os pesos e outros parâmetros calculados do modelo FAM serão os parâmetros para a fase de desempenho, sendo que esta atua em modo *online*. Na fase de desempenho as variáveis de processo são apresentadas na camada de pré-processamento da FAM e na saída gerará um código de prognóstico associado às variáveis apresentadas.

#### **4.2.6 Tomada de decisão**

Considerando informações sobre o diagnóstico e prognóstico de saúde da máquina, sistema, subsistemas ou componentes, bem como em uma noção a respeito da severidade, urgência e importância de se tomar certa decisão. Esta camada tem o objetivo de integrar as informações necessárias visando gerar sugestões de ações de manutenção.

Segundo Bengtsson (2007), com o objetivo de ter uma decisão apropriada faz análises críticas usando a análise FMEA ou a análise por árvore de falhas FTA (*Fault Tree Analysis*), os conceitos destes métodos de análise são explicados no Apêndice D. Esta

camada irá realizar a tomada de decisão baseando-se na sua base de conhecimento gerada a partir da árvore de faltas/falhas, da árvore de sintomas, FMEA e pelas informações coletadas dos especialistas. As experiências dos operadores para tomada de decisão são implementadas em regras de produção. Estas regras de produção processam os códigos de diagnósticos e prognósticos. As decisões geradas por esta camada são enviadas à camada de apresentação através de sugestões das possíveis ações de manutenção que pode ser adotados pelos operadores e mantenedores.

#### **4.2.7 Apresentação**

Esta camada é o nível mais alto do modelo de referência OSA-CBM e o lado cliente da arquitetura cliente/servidor da metodologia proposta. Suporta a interface com o usuário apresentando informações produzidas pelas camadas mais baixas, monitoramento de variáveis em tempo real e histórico de tendências de variáveis associadas às anomalias produzidas. A camada de apresentação é desenvolvida para GUI (Interface Gráfica com o Usuário) baseada em *browser* (Netscape, Mozilla, IExplore, entre outros) usando HTML, PHP e *Applets* (Java).

#### **4.3 MODELAGEM FUNCIONAL IDEF0**

O sistema inteligente de manutenção baseada em condição foi modelado usando a metodologia IDEF0 (*Integration DEfinition language 0*). O método IDEF0 pode ser utilizado para modelar as decisões, ações e atividades de um sistema em uma forma gráfica estruturada (Kim *et al.*, 2002). Além de apresentar uma breve descrição especificando o propósito do modelo, este método também especifica o ponto de vista que ajuda a guiar e a restringir a criação do mesmo (IDEF0, 1993). A Figura 4.3 representa o nível 0 do modelo IDEF0 com todas as entradas, controles, mecanismos, saídas e funcionalidades que são propostas pela metodologia do sistema inteligente de manutenção baseada em condição.

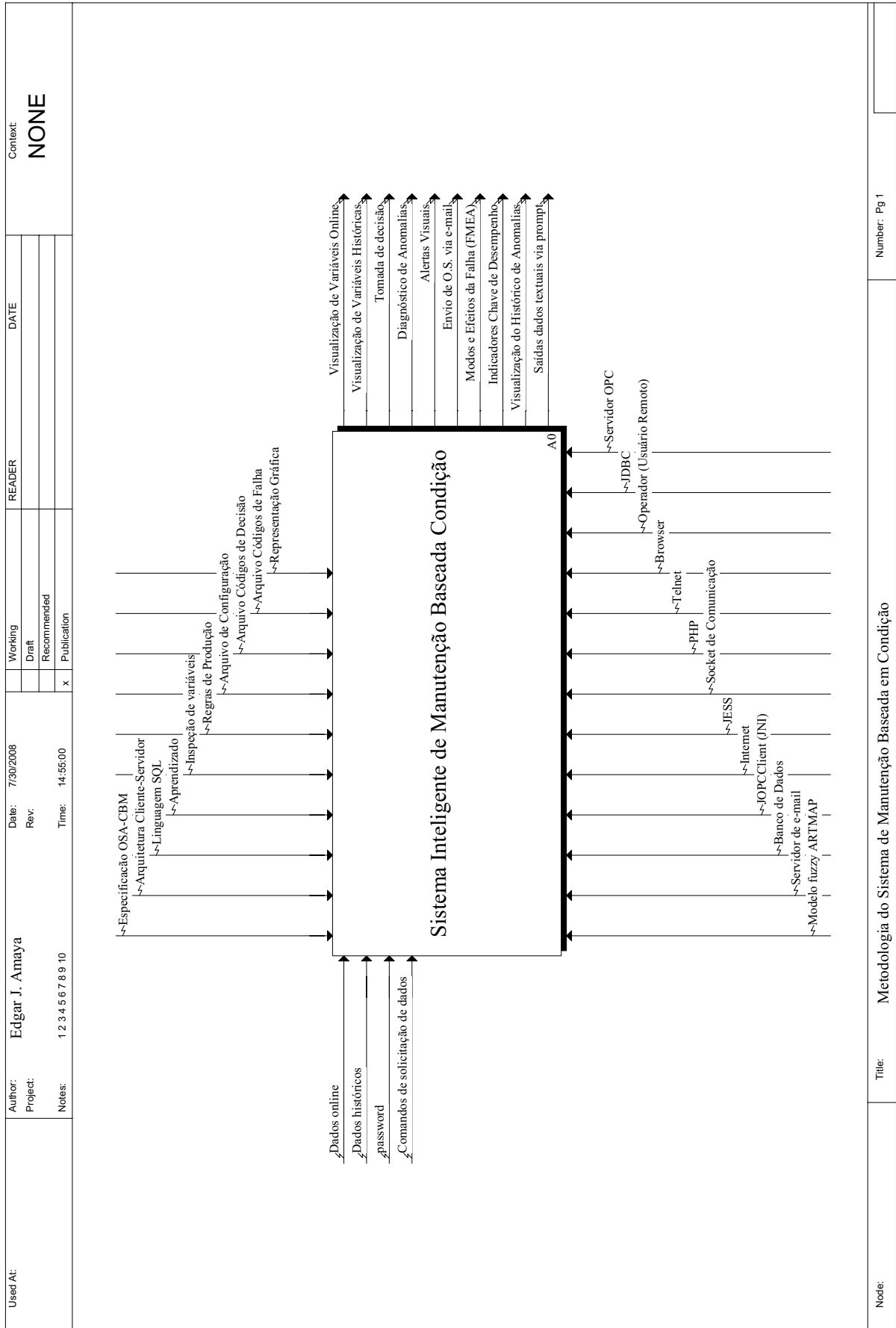


Figura 4.3- Diagrama A0 do sistema inteligente de MBC.

Para especificar com maior nível de detalhamento o modelo funcional do sistema inteligente de MBC. Na Figura 4.4 apresenta-se a estratificação do sistema em duas atividades associadas aos *softwares* interoperáveis que são o lado servidor e o lado cliente do sistema, sendo as atividades:

1. *I-kernel*;
2. Cliente *web*.

A atividade *I-kernel* tem as entrada dados *online* e históricos, e as saídas OS (ordem de serviço) via email e informações solicitadas pelo cliente *web* (variáveis, diagnósticos, prognósticos, tomadas de decisão, etc.) que servem como entrada de dados para os clientes. A atividade Cliente *web* tem como saída através de uma interface gráfica dados de diagnósticos, prognósticos, tomada de decisão, visualização gráfica de variáveis *online* e históricas, KPI, FMEA e outras informações solicitadas pelo usuário.

A seguir são apresentados os diagramas IDEF0 associados às duas atividades da metodologia do sistema concebido.

#### **4.3.1 Atividade *I-kernel***

O modelo IDEF0 da atividade *I-kernel* é apresentado na Figura 4.5, esta atividade é constituída por oito funções, seis delas representam as camadas mais baixas do modelo de referência OSA-CBM:

1. Aquisição de dados: coleta os dados da instrumentação via servidor OPC e banco de dados;
2. Processamento de sinal: verifica a qualidade dos dados adquiridos;
3. Monitoração de condição: avalia a condição de operação de um equipamento;
4. Avaliação de saúde: diagnósticos de defeitos e falhas;
5. Prognósticos: predição de possíveis falhas;
6. Tomada de decisão: gera sugestões de ações de manutenção;
7. Armazenamento de Informações: armazena as anomalias diagnosticadas, tomadas de decisão e variáveis de processo;
8. Módulo de comunicação servidor-cliente: envia informações solicitadas pelo cliente.



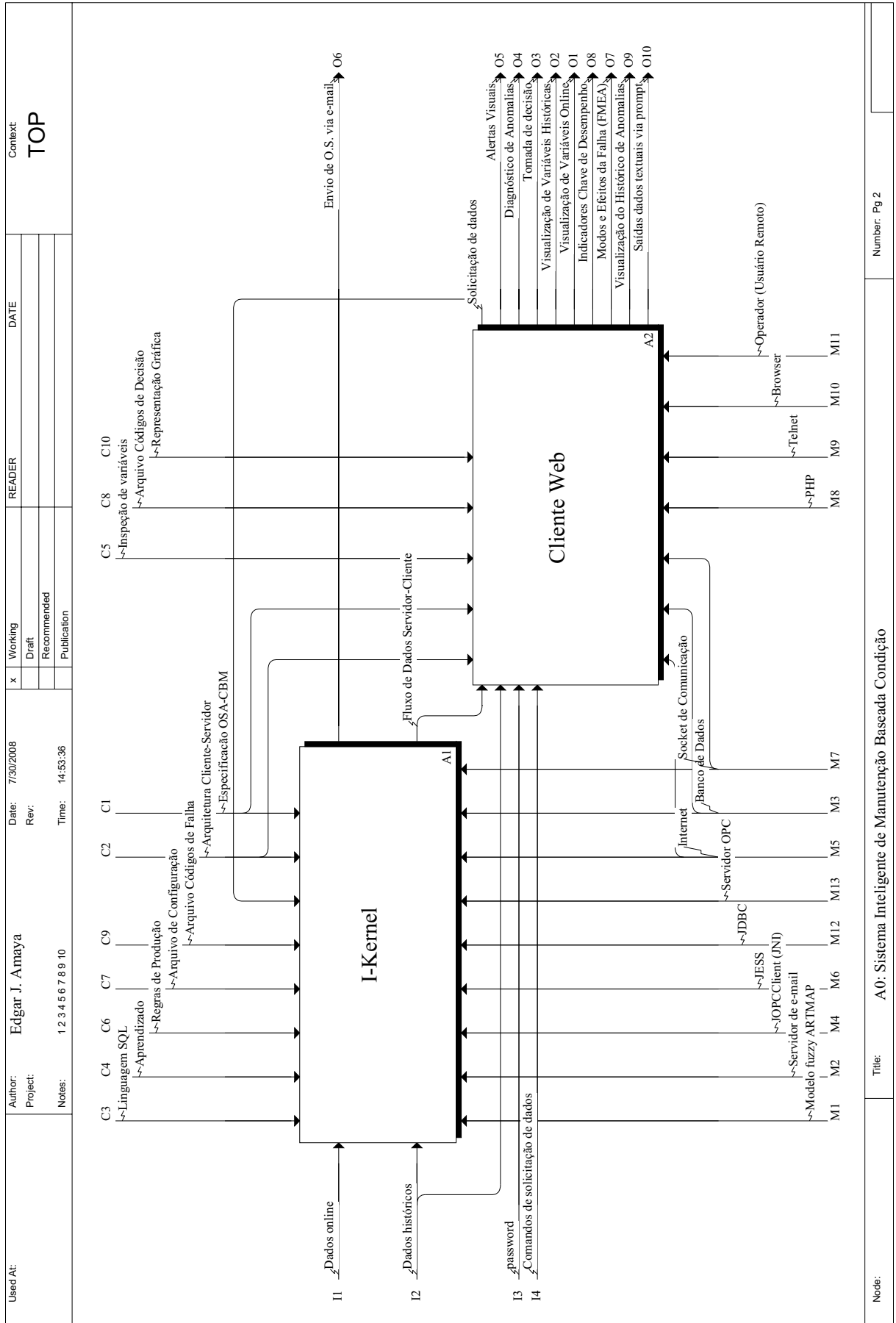


Figura 4.4- Diagrama filho da função A0.

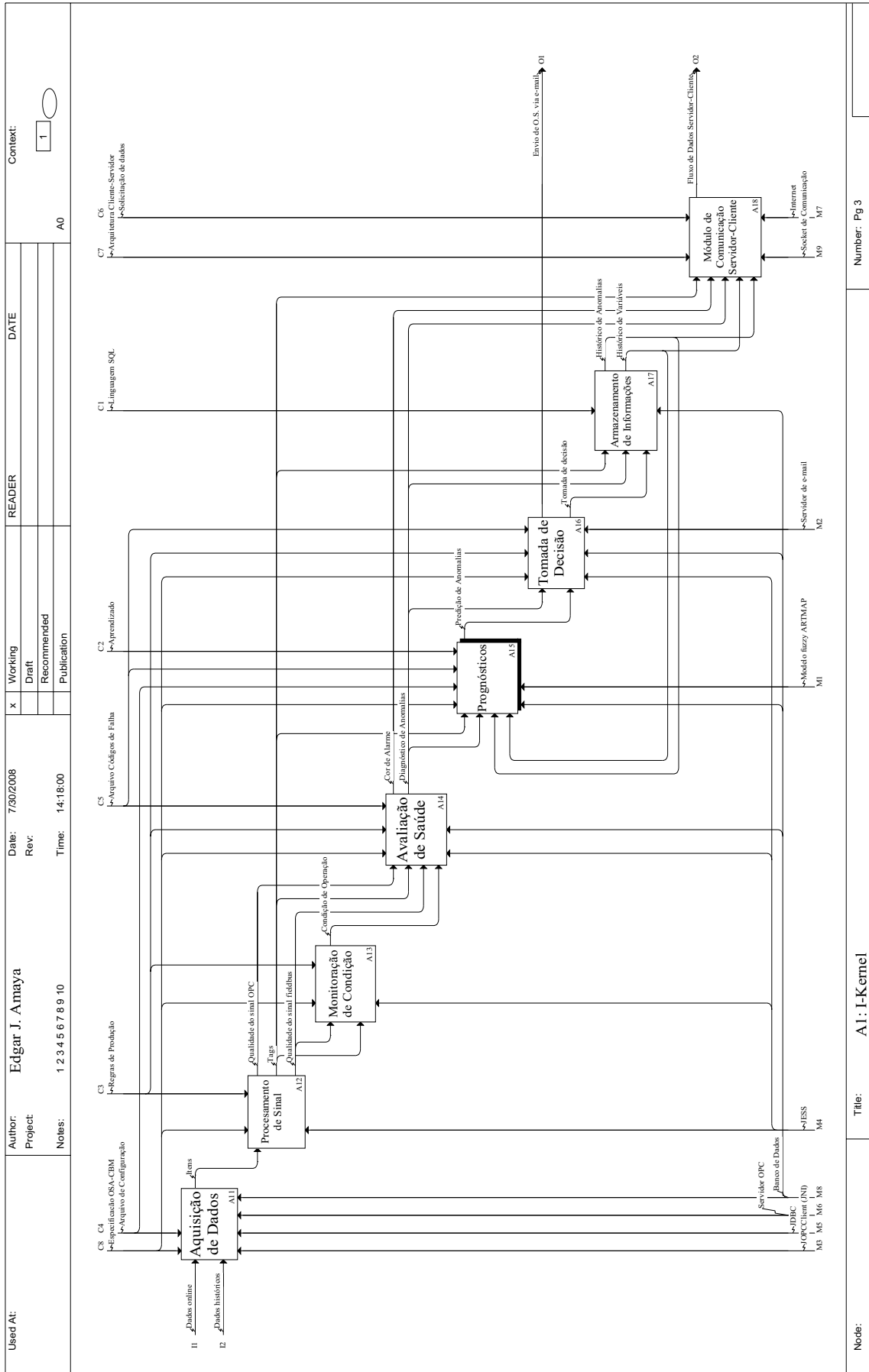


Figura 4.5- Diagrama filho da função A1.

#### 4.3.1.1 Atividade de prognóstico.

O modelo IDEF0 da atividade prognóstico é apresentado na Figura 4.6, esta atividade é constituída por sete atividades:

1. Fase de treinamento: este módulo classifica e associa as variáveis e as anomalias do histórico do banco de dados. Os padrões de entrada são enviados à camada de pré-processamento e os padrões de saída à camada de saída. Baseados nos padrões apresentados são calculados os pesos sinápticos e outros parâmetros da FAM;
2. Fase de desempenho: nesta fase usam-se os pesos e parâmetros calculados pela rede na fase de treinamento. Este módulo aceita variáveis *online*, gera e envia um vetor de tamanho  $M$  à camada de pré-processamento;
3. Camada de pré-processamento: consiste de  $M$  nós. Normaliza o vetor de entrada ao intervalo  $[0, 1]$  e converte um vetor de tamanho  $M$  em outro vetor de tamanho  $2M$  antes de ser aplicado à camada de entrada;
4. Camada de entrada: esta camada tem  $2M$  nós ou neurônios e aceita os valores de entrada normalizados.
5. Camada de representação de categoria: Esta camada tem  $N_a$  nós. O valor de  $N_a$  muda no processo de treinamento da FAM estabelecendo-se em um valor ao final do treinamento. Cada nó nesta camada é uma categoria e representa um grupo de padrões similares;
6. Camada de saída: Esta camada tem  $N_b$  nós, onde  $N_b$  representa o número das classes de saída;
7. Codificação FAM-SE: acondiciona as classes de saída geradas pela camada de saída da FAM para códigos compatíveis usados no processamento de regras de produção do SE.



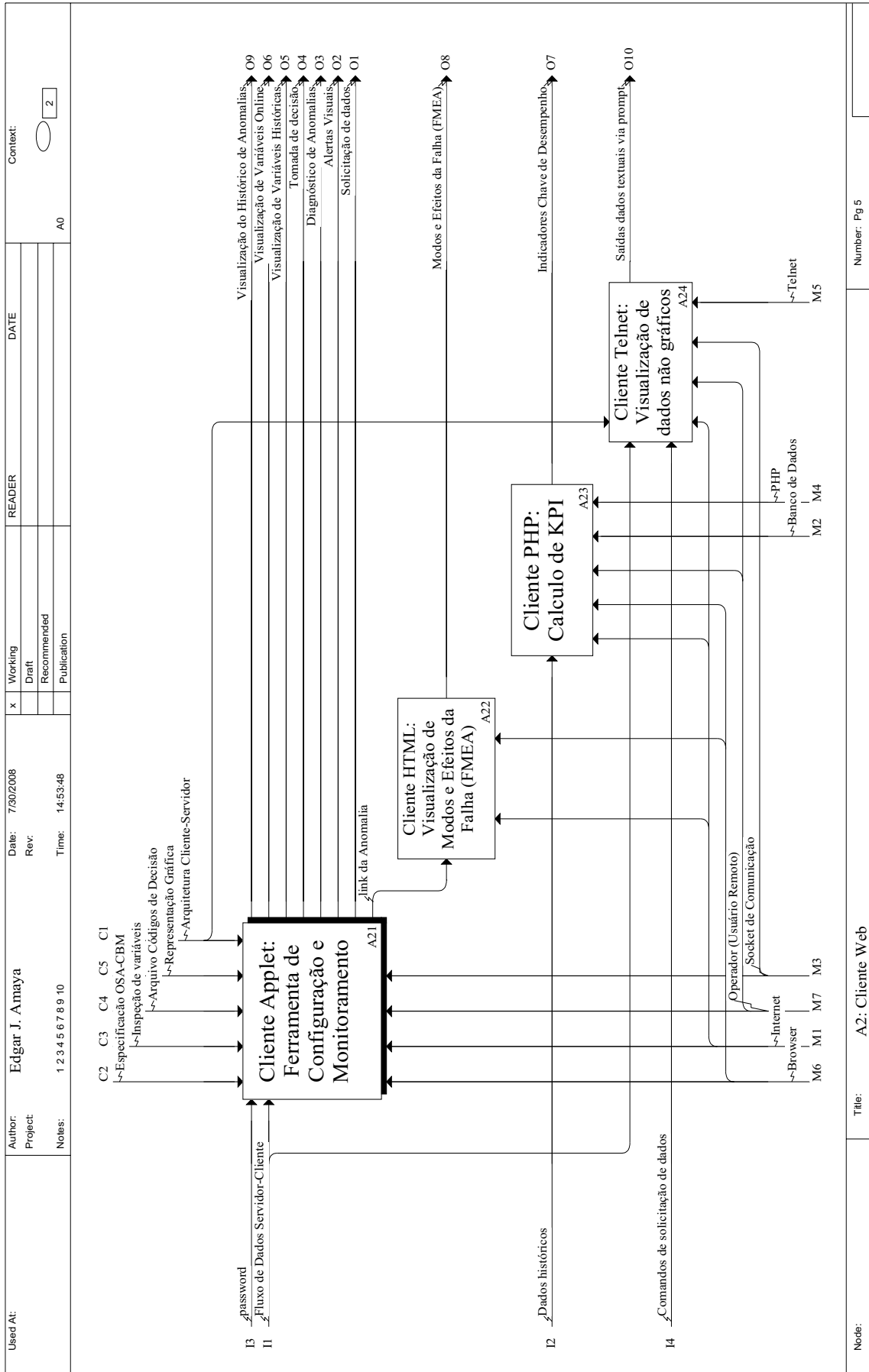


Figura 4.7- Diagrama filho da função A2.

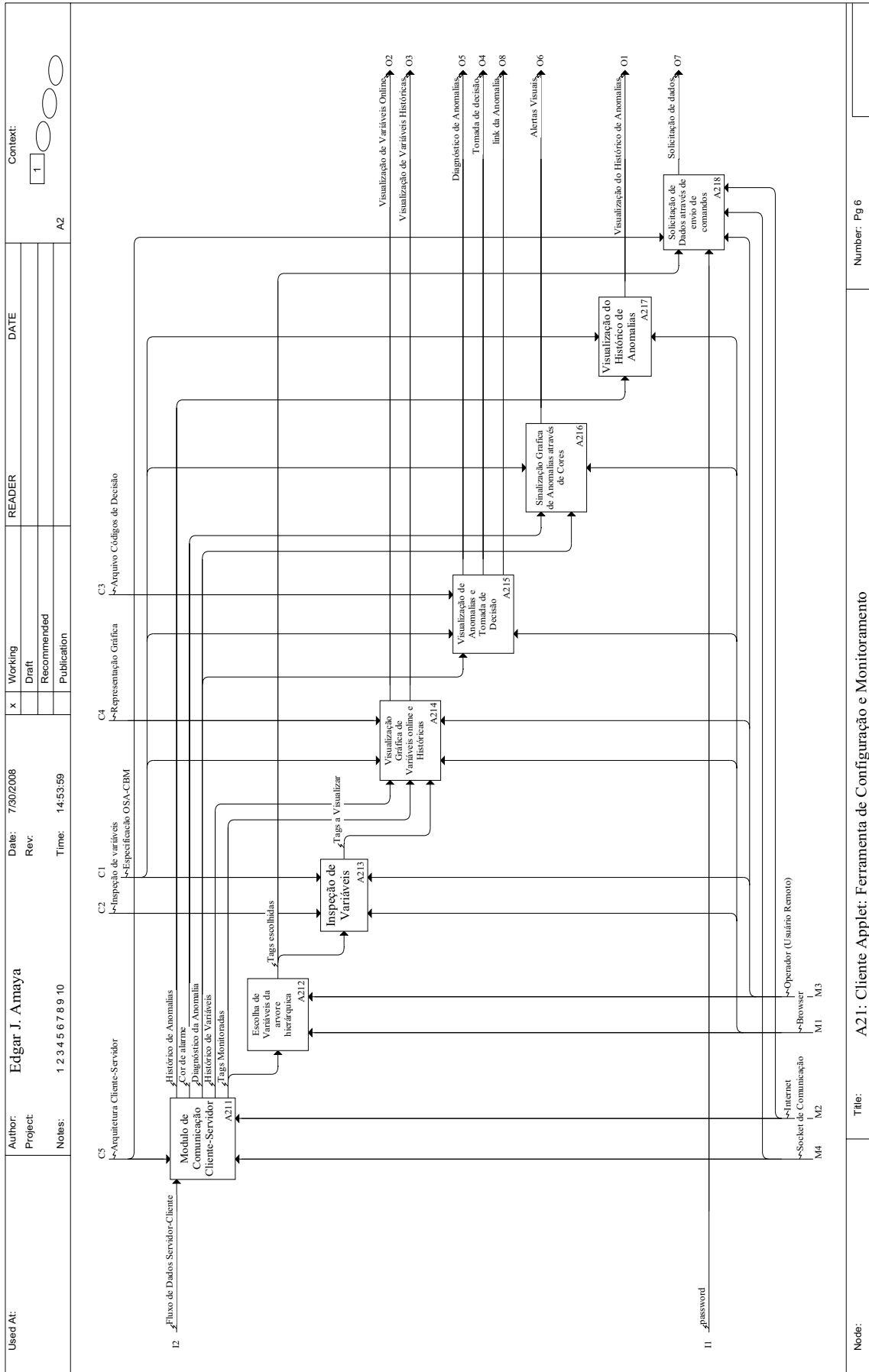


Figura 4.8- Diagrama filho da função A21.

### 4.3.2 Atividade clientes *web*

A atividade clientes *web* mostrada na Figura 4.7. Esta atividade é constituída por quatro atividades para apresentar resultados gerados pelo servidor *I-Kernel*:

1. Cliente *Applet* - Ferramenta de Configuração e Monitoramento: cliente desenvolvido em Java que recebe as informações solicitadas do servidor *I-kernel* e visualiza através de outras atividades.
2. Cliente HTML – Visualização de modos e efeitos da falha (FMEA): usa-se para explorar através de um *link* os detalhes de uma anomalia através da análise FMEA do equipamento;
3. Cliente PHP – Cálculo de KPI: calcula os indicadores-chave de desempenho de um equipamento, subsistema, sistema e de toda a planta em análise;
4. Cliente *Telnet* – Visualização de dados não gráficos: solicita dados do lado servidor através de comandos, pode ser de forma automática ou manual (via *prompt*);

A GUI disponibiliza as funcionalidades necessárias para suportar a visualização de possíveis anomalias ocorridas nos instrumentos, equipamentos, subsistemas e sistemas. Esta interface está associada a menus, opções de visualização, interações com o usuário de forma gráfica e textual (via *prompt*), seleção de variáveis a inspecionar (*online* ou histórico do banco de dados), reiniciar ou apagar (*shutdown*) o servidor.

#### 4.3.2.1 Atividade Cliente *Applet*.

Esta atividade é mostrada na Figura 4.8 e chega a ser a camada de apresentação do modelo de referência OSA-CBM. Esta atividade é constituída por oito atividades para apresentar resultados das camadas anteriores do modelo OSA-CBM através de uma interface com o usuário:

1. Módulo de comunicação cliente-servidor: recebe as informações solicitadas do servidor *I-kernel*, e as disponibiliza para as outras atividades;
2. Escolha de variáveis da árvore hierárquica: permite a seleção da árvore hierárquica de ativos e as variáveis a inspecionar;
3. Inspeção de variáveis: mostra parâmetros das variáveis como valor, qualidade e hora;

4. Visualização gráfica de variáveis *online* e históricas: permite o monitoramento gráfico de variáveis em tempo real e variáveis históricas armazenadas no banco de dados;
5. Visualização de anomalias e tomadas de decisão: mostra as anomalias ocorridas e as sugestões das ações de manutenção;
6. Sinalização gráfica de anomalias através de cores: sinal gráfico para localizar o equipamento com funcionamento anormal;
7. Visualização do histórico de anomalias: mostra as trinta últimas ocorrências de falhas ou defeitos;
8. Solicitação de dados através de envio de comandos: solicita dados do lado servidor através de comandos.

#### **4.4 MODELAGEM DE DADOS IDEF1X**

O método IDEF1X (*Integration Definition for Information Modeling*) é utilizado para desenvolver o modelo de informação do sistema inteligente de manutenção baseado em condição. As características básicas da linguagem IDEF1X são: entidades, atributos e relacionamentos (Kern, 1999). A partir do modelo da informação representado pelo modelo entidade relacionamento e pela normalização dos dados é possível gerar o modelo físico do banco de dados relacional (Oliveira, 2002). O sistema de gerenciamento de base de dados MySQL é utilizado como ferramenta de desenvolvimento.

O modelo de informação é dividido em dois domínios. As informações associados ao arquivo configuração (servidores OPC, Banco de Dados e Email; dispositivos DFI, configurações do *I-Kernel*, parâmetros da FAM, Lista de Emails, Tags OPC, Tags do Banco de Dados e Tags Simuladas), estas informações são utilizadas quando se inicia o *I-kernel*. E as informações associadas ao banco de dados Simprebal (anomalias, decisões, equipamentos, tags, cadastro de funcionários e *logs*). Este modelo foi implementado fisicamente através de um único banco de dados denominado de Simprebal, sendo constituído por trinta e três tabelas.

No modelo de informação do arquivo de configuração mostrado na Figura 4.9, os dados das variáveis a adquirir são agrupadas na entidade Tags. Existem três tipos de entidades de Tags: tags do servidor OPC (OPCTags), tags simuladas (SIMTags) e tags do banco de dados (DBTags). As tags OPC estão relacionadas a um servidor OPC (entidade



OPCServer) e a um dispositivo DFI (entidade DFIDevice). As tags do banco de dados estão relacionadas a um servidor de banco de dados (atributo DBServer). A lista de emails é organizada em grupos e armazenada na entidade Emails, o envio de emails é feito via um servidor de emails (EmailServer). As configurações básicas do *I-Kernel* estão contidas na entidade *IkernelConfiguration*. Os parâmetros do modelo FAM estão localizados na entidade parâmetros FAM.

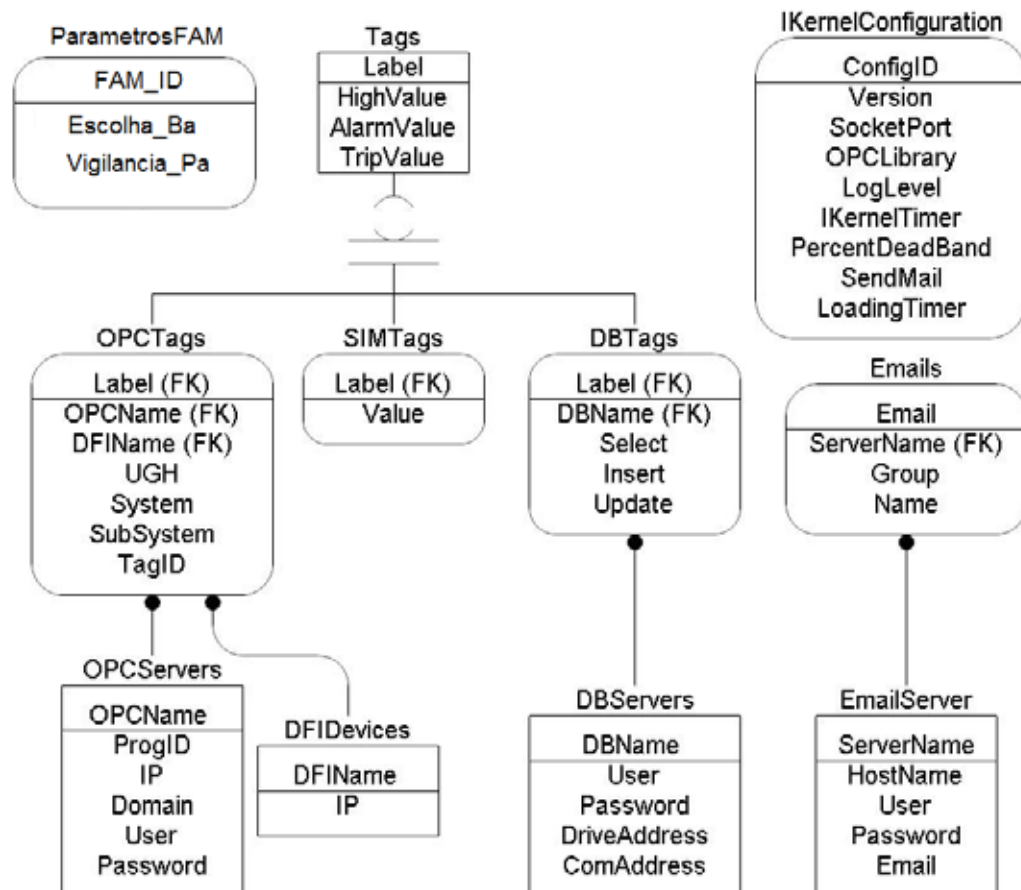


Figura 4.9- Modelagem da informação do arquivo de configuração através da metodologia IDEF1X.

Na Figura 4.10 é ilustrado o modelo de informação do banco de dados. Na entidade anomalias são registradas todas as anomalias detectadas pelo sistema inteligente de MBC. Cada anomalia possui um atributo *descrição* que define a anomalia apresentada. O atributo *modo* indica o código ou códigos com o tipo de anomalia, alarme ou alerta. A *causa* descreve a situação do equipamento que levou até a anomalia.

A *data\_inicio* registra a data que ocorreu a anomalia. A data e hora em que o equipamento volte ao seu funcionamento normal devem ser preenchidas pelo usuário no atributo *data-*

*termino*, este procedimento é ilustrado na **Erro! Fonte de referência não encontrada.** O atributo *severidade* é um índice que reflete a gravidade das conseqüências de uma falha, os valores estão apresentados no Apêndice D. O atributo *deteccão* é um índice construído com base na estimativa da probabilidade de uma falha a ser detectada, assumindo-se que ela tenha ocorrido, os valores são mostrados no Apêndice D.

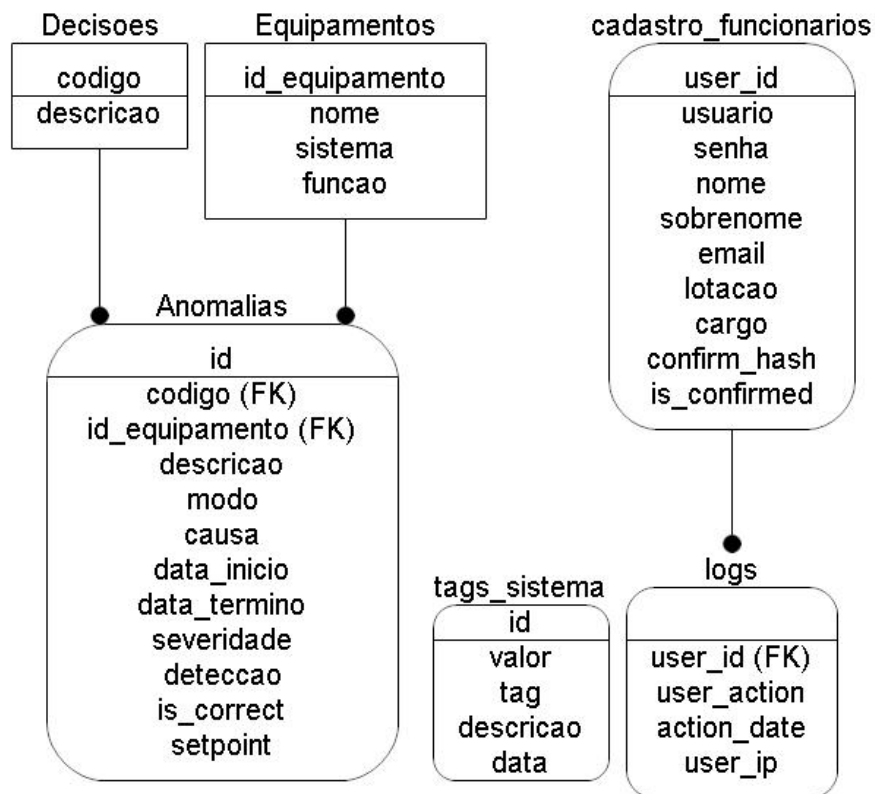


Figura 4.10- Modelagem da informação do banco de dados através da metodologia IDEF1X.

O atributo *is-correct* é usado para saber se uma falha detectada pelo sistema é correta, e ajudará a conhecer a porcentagem de acerto, este campo deve ser preenchido pelo usuário. O atributo *setpoint* é o valor da variável em operação normal. Os atributos *código* e *id Equipamentos* são declarações de chaves estrangeiras (*Foreign Key*). Isto é necessário para garantir que todas as anomalias cadastradas na base de dados estejam associadas a uma tomada de decisão e a um equipamento. Na entidade *tags\_sistema* são armazenados o histórico dos valores das tags. Quando um usuário faz o cadastro, seus dados são armazenados na entidade *cadastro\_funcionarios*. Os eventos de acesso dos usuários ao sistema são registrados na entidade *logs*.

## 5 SIMPREBAL: MODELAGEM UML

Este capítulo descreve o modelo UML do sistema computacional SIMPREBAL. São apresentados a arquitetura do sistema, os requisitos de usuário, requisitos funcionais e não funcionais. Nos requisitos do sistema são apresentados os diagramas caso de uso da aplicação *I-kernel* e da ferramenta Configuração e Monitoramento.

### 5.1 INTRODUÇÃO

A modelagem UML é desenvolvida para documentar o sistema computacional (*I-Kernel* e *confmonittool*), sistema inteligente a ser desenvolvido em Java, para dar suporte ao desenvolvimento do SIMPREBAL. O subsistema *I-Kernel* vem a ser um dos componentes fundamentais do SIMPREBAL, sendo responsável pela captura dos sinais de monitoramento dos equipamentos, seu processamento inteligente e realimentação do banco de dados com recomendações de manutenção.

#### 5.1.1 Arquitetura do SIMPREBAL

A arquitetura do sistema pode ser visualizada na Figura 5.1, na qual apresenta-se o diagrama raiz para todo o sistema, que basicamente subdivide a arquitetura em 2 pacotes, o pacote *I-Kernel* e o pacote *confmonittool*. O pacote *I-Kernel* é o lado servidor da arquitetura onde se encontra as primeiras seis camadas do modelo de referência OSA-CBM. O pacote *confmonittool* é o lado cliente, encontra-se aqui a interface com o usuário (camada de apresentação do modelo OSA-CBM).

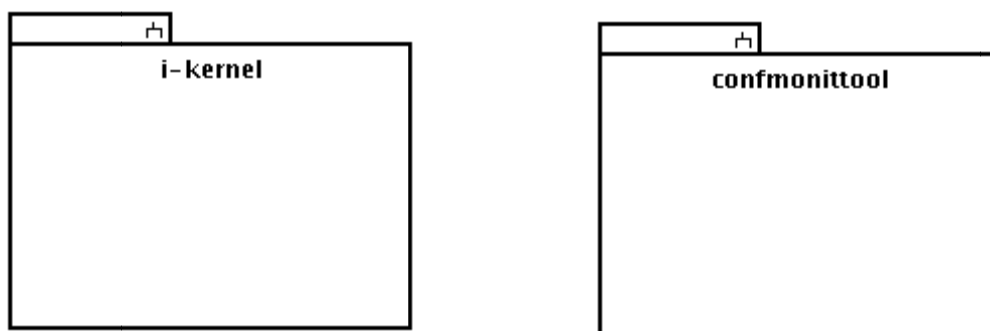


Figura 5.1- Principais pacotes da arquitetura SIMPREBAL.

A interação do pacote *I-kernel* com as fontes de dados e ferramentas é mostrada na Figura 5.2. A coleta e armazenamento de informação do banco de dados SIMPREBAL é via JDBC. A aquisição de dados dos instrumentos via JNI (*Java Native Interface*) é através do

servidor OPC. No processamento inteligente dos dados adquiridos o SIMPREBAL usa a ferramenta JESS.

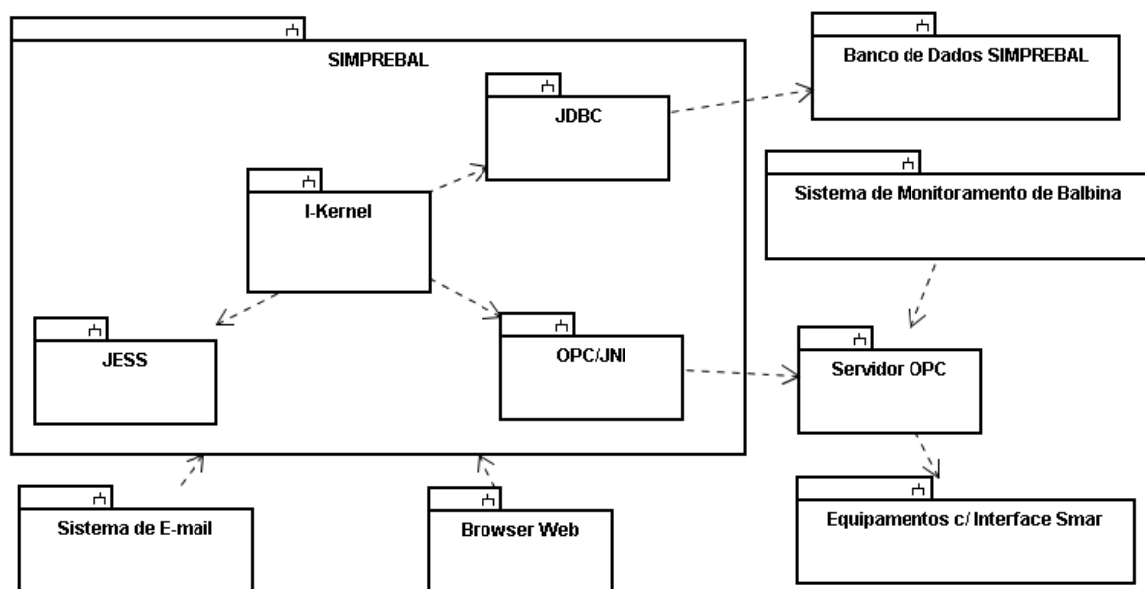


Figura 5.2- O I-Kernel e sua interação (Amaya *et al.*, 2007c).

## 5.2 REQUISITOS DE USUÁRIO

Segundo Sommerville (2007), uma maneira útil de se estruturar os requisitos é dividi-los entre requisitos de usuário e requisitos de sistema. Os requisitos de usuário devem capturar as demandas mais gerais dos usuários do sistema. Normalmente, estes requisitos são descritos em linguagem natural e não devem entrar em detalhes mais técnicos. Os requisitos de sistema são descrições mais detalhadas dos requisitos de usuário. Os requisitos de usuário serão apresentados na forma de uma lista de requisitos funcionais e requisitos não-funcionais em linguagem natural.

### 5.2.1 Requisitos Funcionais

Segundo Pressman (1995), os requisitos funcionais de um sistema descrevem as funcionalidades desejadas pelos clientes, ou seja, o que se espera que o *software* faça. Segundo Campos (2007), A norma brasileira NBR 13596 foi substituída pela ISO/IEC 9126-1, se tornando NBR ISO/IEC 9126-1, que descreve um modelo de qualidade do produto de *software*, composto de duas partes: (1) Qualidade interna e qualidade externa; e (2) Qualidade em uso. De acordo com exposto acima foram definidos os requisitos funcionais para o sistema inteligente, esta definição é detalhada na Tabela 5.1.

Tabela 5.1- Requisitos Funcionais (RFs) (Amaya *et al.* 2007a, modificado)

RF	Descrição
1	O sistema deve acessar os dados da Usina a partir dos Bancos de Dados utilizados pelo Sistema de Monitoramento da Usina, ou diretamente dos equipamentos de controle, por meio de um servidor OPC que disponibiliza as informações <i>online</i> dos equipamentos.
2	O sistema deve processar esses dados nas seguintes formas: Na forma de um sistema especialista baseado em regras de produção e na forma de RNA.
3	O sistema deve alertar o usuário por meio de mensagens de e-mail quando possíveis falhas puderem ser diagnosticadas.
4	O sistema deve alertar o usuário por meio de um alerta visual, quando possíveis falhas puderem ser diagnosticadas.
5	O sistema deve propiciar a edição de um sinótico contendo certo conjunto de variáveis sendo monitoradas, escolhidas pelo usuário e compondo uma possível tela de apresentação.
6	O sistema deve exibir o valor <i>online</i> das variáveis sendo monitoradas que foram selecionadas para compor um determinado sinótico, apresentando-as em uma tela própria previamente desenvolvida.
7	O sistema deve permitir a implementação de algum mecanismo de aprendizagem, de tal forma que os históricos de falhas e tomadas de decisão anteriores possam ser utilizados para prevenir o surgimento de novas falhas.
8	O processamento das informações se dará na forma de um ciclo operacional fechado, que seguirá a seguinte seqüência: (1) Verificação dos dados a serem adquiridos; (2) Aquisição de dados do banco de dados; (3) Aquisição de dados via OPC; (4) Armazenamento provisório de todos os dados em variáveis do JESS; (5) Para cada uma das N camadas possíveis de processamento das regras via JESS e RNA; (6) Atualização dos dados no banco de dados; (7) Atualização dos dados via OPC.

### 5.2.2 Requisitos Não-Funcionais

Segundo Pressman (1995), os requisitos não-funcionais compreendem as qualidades e restrições globais do sistema relacionados à manutenção, uso, desempenho, custo, interface, etc. Os requisitos não-funcionais são críticos para o sucesso de sistemas de *software* e estão diretamente relacionados com a satisfação dos usuários. Na Tabela 5.2 Amaya *et al.* (2007a), apresenta os requisitos não-funcionais para o sistema inteligente e seguindo uma lista detalhada de funcionalidades .

Tabela 5.2- Requisitos Não Funcionais (RNFs) (Amaya *et al.* 2007a, modificado)

RNF	Descrição
1	O sistema deve ser desenvolvido na linguagem Java.
2	As regras de produção do sistema não devem ser armazenadas diretamente em código-fonte, mas devem ser editáveis e estar disponíveis externamente em um arquivo modificável.
3	O sistema deve possuir uma interface <i>web</i> de acesso, por meio da qual seja possível ao usuário configurar o sistema, editar regras e parâmetros e monitorar as variáveis sendo processadas pelo sistema.
4	O sistema deve ser conectável a bancos de dados SQL genéricos, desde que exista um <i>driver</i> JDBC para o respectivo banco de dados.
5	O sistema deve ser conectável a servidores OPC, desde que exista um <i>driver</i> JNI.
6	O sistema deve utilizar a <i>shell</i> JESS para processar as regras na forma de sistemas especialistas.
7	Na camada de prognósticos o sistema deve usar uma rede modelo <i>Fuzzy</i> ARTMAP.
8	O sistema deve ser concebido de tal forma que as regras de produção possam ser usadas de modo intercambiável para cada uma das camadas de processamento do SIMPREBAL (2-processamento de sinal, 3-monitoração de condição; 4-avaliação de saúde; 6-tomada de decisão) e as RNA na camada (5-prognósticos)

### 5.3 REQUISITOS DO SISTEMA

Os requisitos do sistema serão detalhados de maneira mais técnica, utilizando-se diagramas UML (Gudwin, 2006). A metodologia de desenvolvimento a ser utilizada é uma adaptação do processo Unificado de modelagem (Jacobson *et al.* 1999). Os diagramas de casos de uso em UML são utilizados para descrever o uso de um sistema por atores (Schneider e Winters, 1998, e Booch *et al.*, 1999).

Um ator representa qualquer elemento externo que interage com o sistema. Um caso de uso descreve uma seqüência de passos/operações que um usuário realiza quando interage com um sistema visando realizar uma determinada tarefa/objetivo. Assim, o aspecto comportamental de um sistema a ser desenvolvido pode ser descrito. No entanto, a descrição de casos de uso não trata a questão de como esta interação será implementada. Fases posteriores à etapa de engenharia de requisitos tais como projeto e implementação focalizarão este aspecto.

Na Figura 5.3, temos um diagrama UML de casos de uso, que nos propicia uma visão geral do sistema (Gudwin, 2006). O sistema é composto de duas aplicações que são executadas de maneira independente: o *I-Kernel* e a ferramenta de C&M.

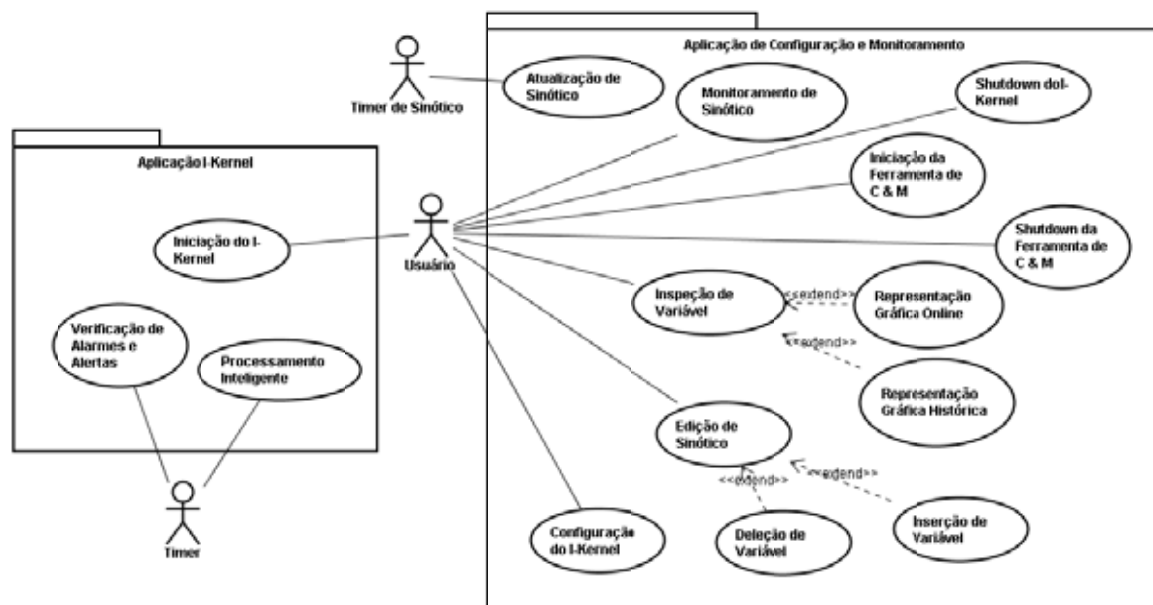


Figura 5.3- Caso de uso do sistema inteligente.

### 5.3.1 Casos de uso da aplicação *I-kernel*

A aplicação *I-Kernel* é um servidor que não possui interface com o usuário, porém, a única coisa que o usuário pode fazer é iniciar a aplicação. Além disso, um timer previamente programado envia *ticks* periódicos que iniciam um ciclo operacional de processamento inteligente e da mesma forma fazem a verificação de alarmes e alertas.

#### 5.3.1.1 Iniciação do *I-kernel*

O diagrama de caso de uso para a iniciação da aplicação *I-kernel* é apresentado na Figura 5.4. Neste caso de uso, o usuário solicita ao sistema operacional que a aplicação *I-kernel* seja carregada, e o sistema operacional carrega a aplicação. O ciclo é iniciado lendo o arquivo de configuração (Apêndice B), onde estão todos os parâmetros para o funcionamento do sistema. Na seqüência define o período para geração de alarmes (*AlarmTimer*) e o período de processamento (*IKernelTimer*). O seguinte passo é estabelecer a comunicação com o banco de dados e com o servidor OPC. Os últimos passos são a realização dos ciclos de inferência do JESS e as RNA modelo FAM que é

responsável pelo processamento inteligente do sistema. Este ciclo é repetido até que o usuário desligue (*shutdown*) a aplicação *I-kernel*.

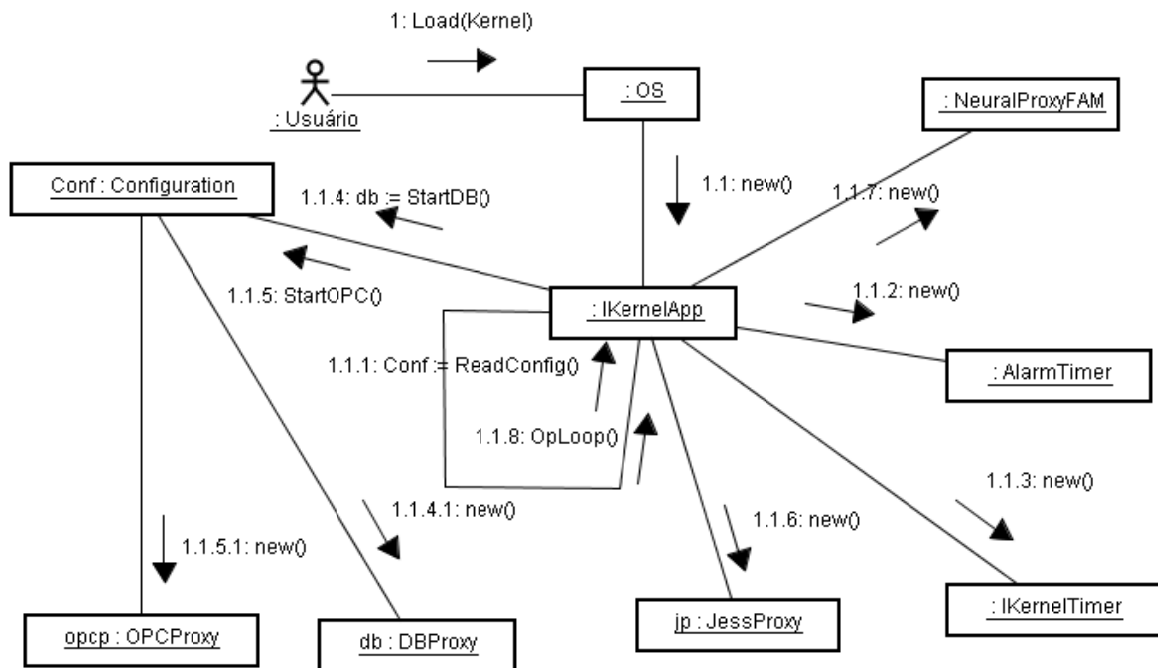


Figura 5.4- Iniciação da aplicação *I-Kernel*.

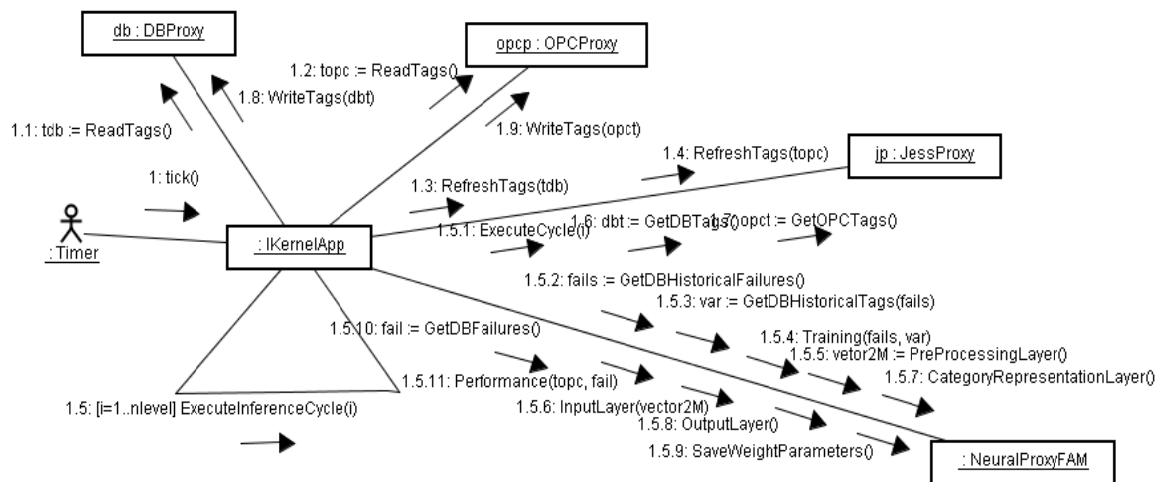


Figura 5.5- Processamento Inteligente.

### 5.3.1.2 Processamento inteligente

O processamento inteligente é representado em diagrama de caso de uso na Figura 5.5. O Timer de Processamento Inteligente iniciado no caso de uso “Iniciação do *I-kernel*”, envia *ticks* de relógio periódicos. O período depende da quantidade de variáveis, das regras e da capacidade de processamento do computador, no software desenvolvido usa-se o valor 1 minuto. O ciclo de processamento inteligente inicia coletando dados (*ReadTags*) do banco



de dados e do servidor OPC, em seguida atualiza as tags (*RefreshTags*) e executa o ciclo de inferência (*ExecuteCycle*) do sistema especialista e as camadas da rede FAM, as tags obtidas são armazenadas no banco de dados (*WriteTags*).

### 5.3.1.3 Verificação de alarmes e alertas

Neste caso de uso ilustrado na Figura 5.6, o Timer de Verificação de Alarmes e Alertas envia um *tick* de relógio, que procede a verificação de um conjunto de variáveis sendo monitoradas, e caso alguma delas esteja marcada, procede ao tipo de alarme ou alerta associado, lê os dados do arquivo de configuração (*i.e.* lista de emails), e envia OS de manutenção via email (*Mailer*). Os códigos de cores, as anomalias e o histórico de anomalias são enviados para o cliente (ferramenta de C&M) através da classe *NetworkProxy* e os erros de execução são armazenados no *Logger*.

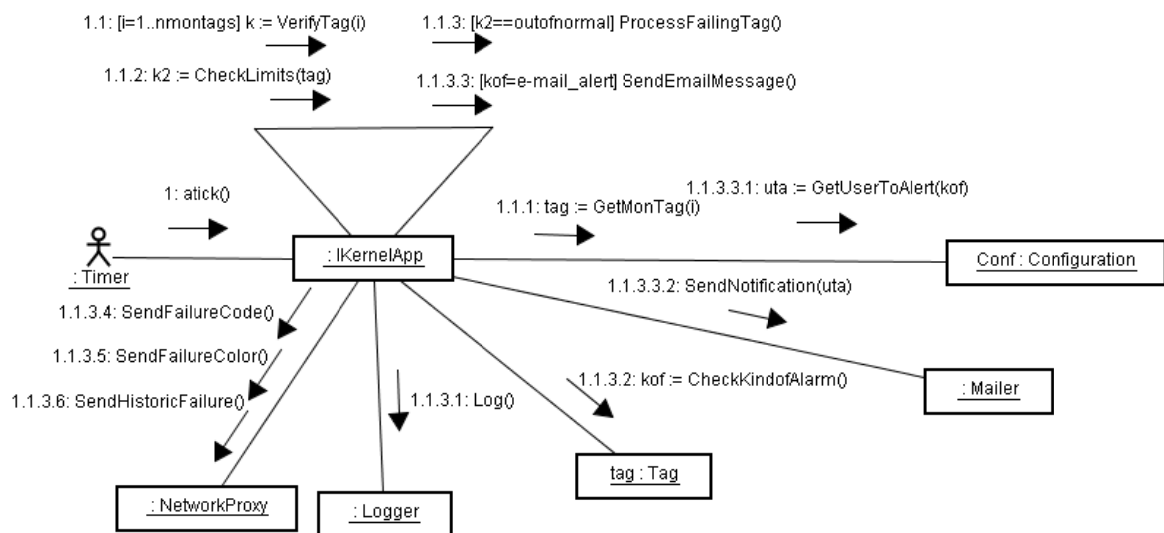


Figura 5.6- Verificação de Alarmes e Alertas.

### 5.3.1.4 Shutdown do I-kernel.

Neste caso de uso, uma vez que o sistema de C&M tenha sido iniciado, o usuário remoto (com senha de permissão) solicita o encerramento do sistema *I-Kernel*. Se a senha é válida e se o *I-kernel* está de fato operacional, o sistema encerra sua operação. Caso contrario, deve alertar o usuário que a senha é inválida ou que o *I-kernel* não está operacional. Este procedimento é mostrado na Figura 5.7.

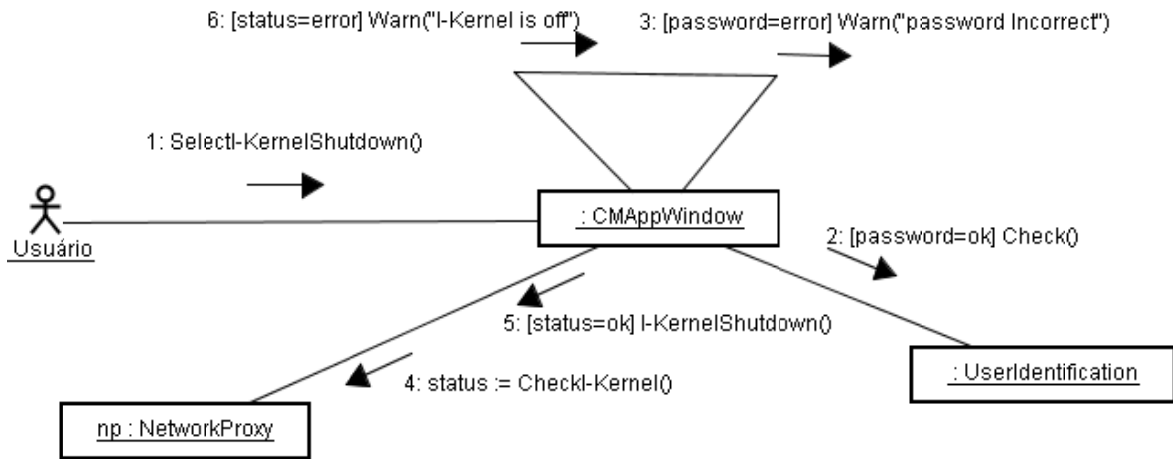


Figura 5.7- Shutdown da aplicação I-Kernel.

### 5.3.2 Casos de uso da ferramenta de C&M

A ferramenta de Configuração e Monitoramento é uma aplicação *web*, destinada ao monitoramento das variáveis sob controle do sistema e a solicitação de *shutdown* do *I-Kernel*. Este monitoramento é uma inspeção direta do estado das variáveis, selecionando-se a variável dentre todas aquelas disponíveis. O procedimento é escolher a variável a ser monitorada, e o sistema exibe seu estado diretamente. Este tipo de monitoramento visa efetuar a inspeção combinada das variáveis críticas do sistema, visualização gráfica *online* e histórica.

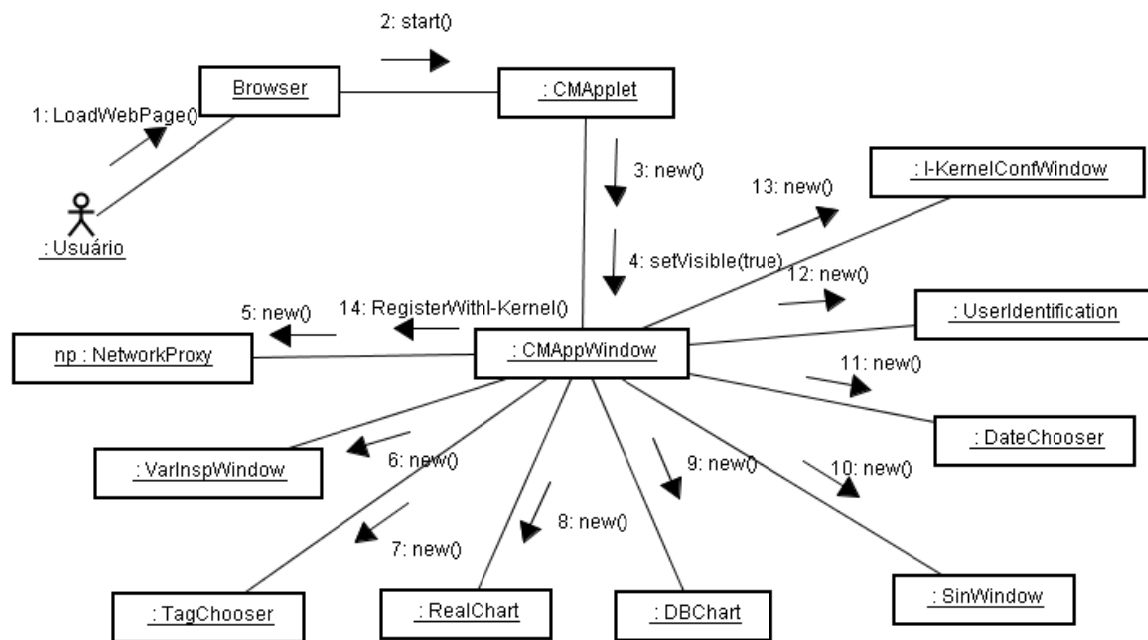


Figura 5.8- Iniciação da ferramenta de C&M.

### 5.3.2.1 Iniciação da Ferramenta C&M

Neste caso de uso, o usuário carrega em um browser Web uma URL que corresponde ao endereço da Ferramenta C&M. A partir daí, o sistema deve iniciar a ferramenta C&M e abrir sua janela principal de interação com o usuário. A Figura 5.8 apresenta o procedimento de iniciação da ferramenta C&M, e o carregamento de todas as classes e janelas a usar.

### 5.3.2.2 Monitoramento de sinótico

Este caso de uso é projetado para que o usuário obtenha um conjunto de variáveis em uma janela definida. O sistema abre a janela de monitoramento e inicia o Timer que fará a atualização da tela. Após isso, o sistema fica aguardando o usuário solicitar o fechamento da janela, e quando isso ocorre, ele apaga o Timer e fecha a janela de monitoramento. A função de monitoramento do sinótico é apresentada na Figura 5.9.

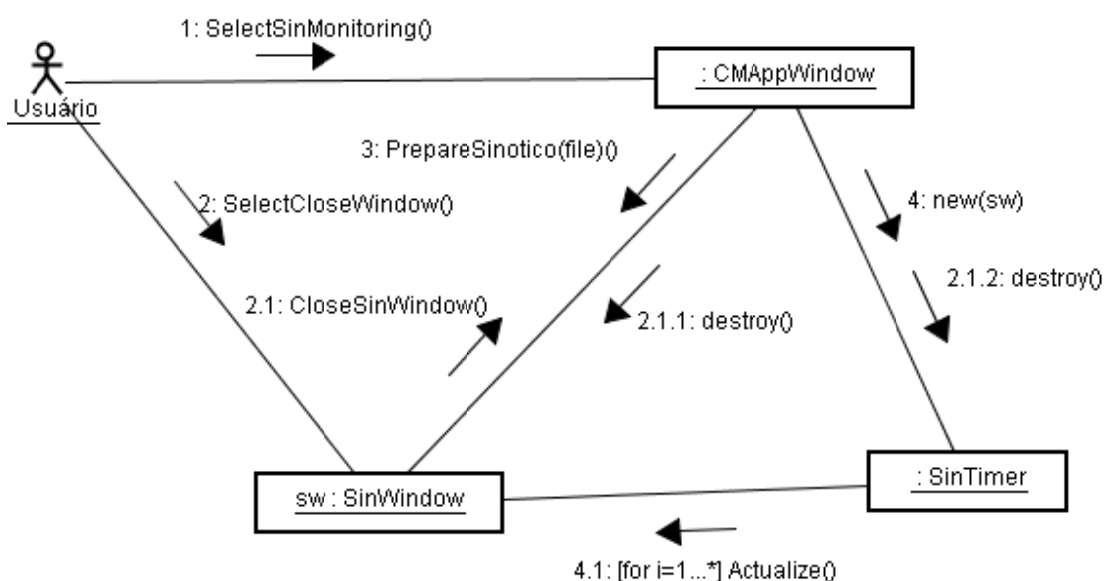


Figura 5.9- Monitoramento de Sinótico.

### 5.3.2.3 Atualização de sinótico

Quando uma janela de sinótico está sendo monitorada, ela está associada a um Timer de Sinótico que envia *ticks* periódicos de relógio. A cada *tick*, o *configmonittol* solicita ao *I-kernel* as variáveis do sinótico, atualiza seus valores e redesenha a tela para apresentar os novos valores. Este caso de uso pode ser visualizado na Figura 5.10.

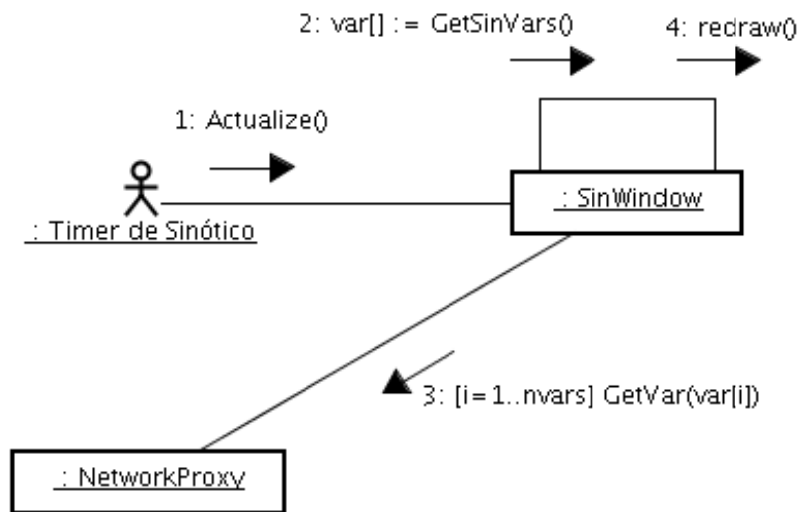


Figura 5.10- Atualização de sinótico.

#### 5.3.2.4 Inspeção de variáveis

Nesse caso de uso, o usuário tem a opção de inspecionar uma ou mais variáveis. O usuário seleciona a inspeção de variáveis, e o sistema abre uma janela de escolha de tags (*TagChooser*) em forma de árvore hierárquica. Nela, o usuário pode selecionar qual variável deseja inspecionar, e o sistema busca essa variável e exibe seu valor. O usuário pode então decidir inspecionar outra variável ou então finalizar a operação. Este caso de uso é mostrado na Figura 5.11.

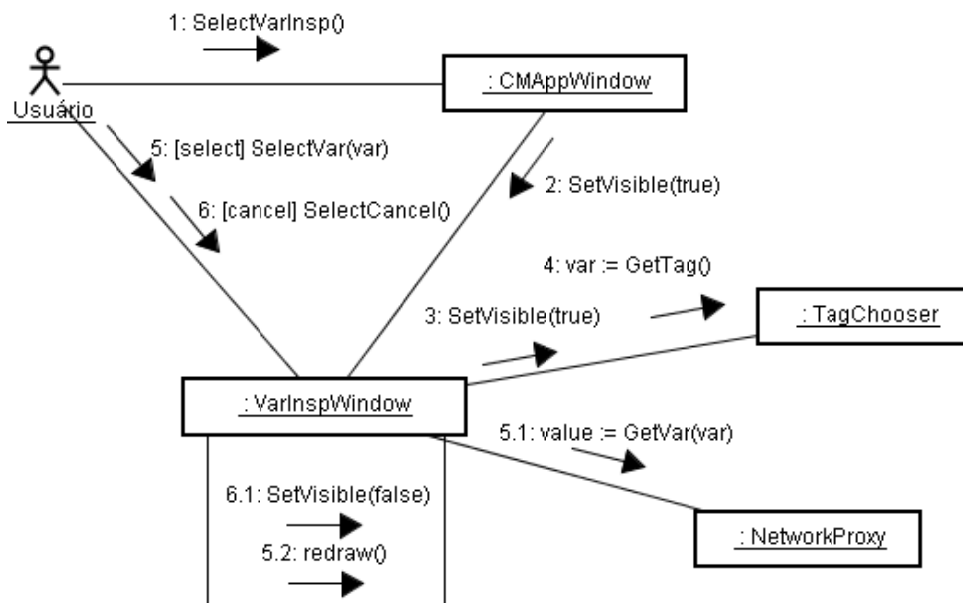


Figura 5.11- Inspeção de variáveis.

### 5.3.2.5 Inspeção de variáveis online

Este caso de uso é mostrado na Figura 5.12, o usuário tem a opção de visualizar graficamente o valor *online* de variáveis. O usuário seleciona as variáveis da tela inspeção de variáveis, e o sistema busca essas variáveis e exibe graficamente os seus valores através da classe *RealChart*.

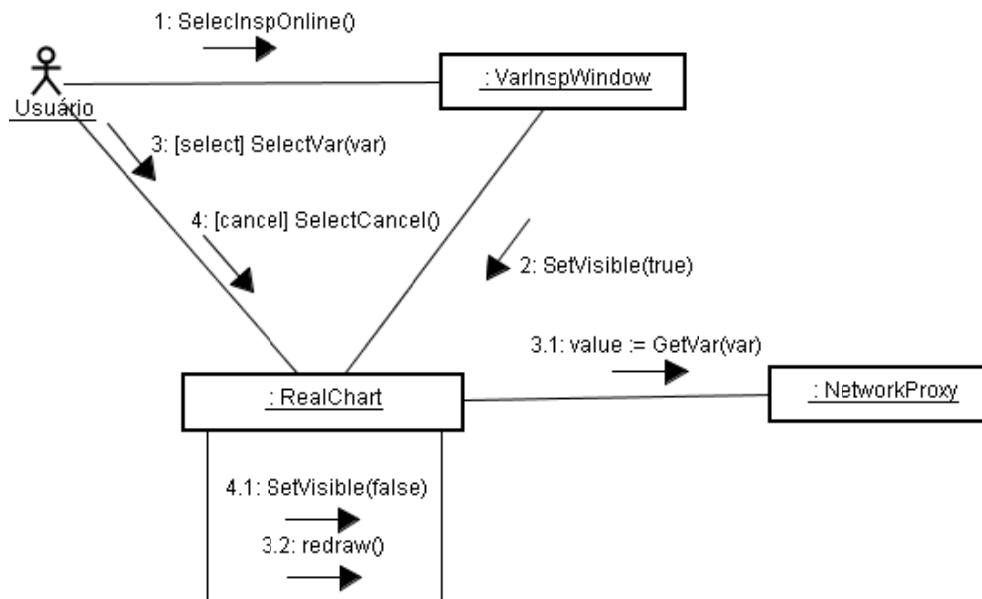


Figura 5.12- Inspeção de variáveis *online*.

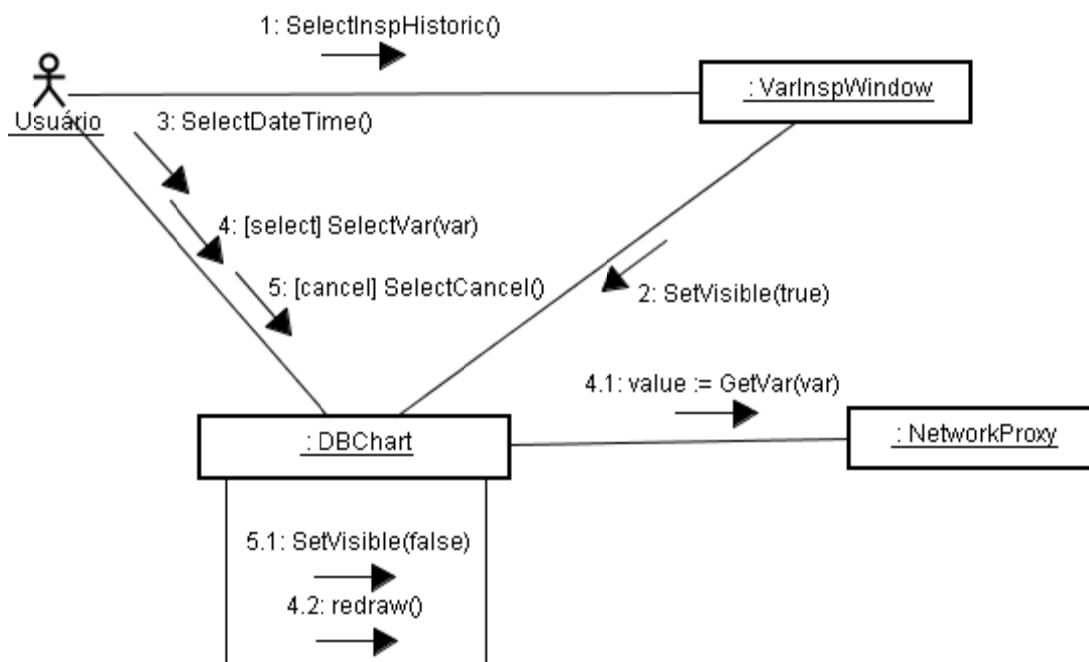


Figura 5.13- Inspeção de variáveis históricas.

### 5.3.2.6 Inspeção de variáveis históricas

Nesse caso de uso, o usuário tem a opção de visualizar graficamente o valor histórico de variáveis. O usuário seleciona as variáveis da tela inspeção de variáveis e o sistema solicita ao usuário definir a data e hora de início e término de inspeção. O sistema busca essas variáveis no banco de dados e exibe graficamente através da classe *DBChart*. Este caso de uso é mostrado na Figura 5.13.

### 5.3.2.7 Shutdown da ferramenta C&M

Uma vez que a Ferramenta C&M esteja aberta e operacional, o usuário solicita que a mesma seja encerrada. A partir daí, o sistema deve encerrar a ferramenta, a janela principal e todas as outras janelas usadas na aplicação. O *shutdown* da ferramenta C&M não implica o fechamento da aplicação *I-Kernel*, ele pode continuar em operação. O procedimento de *shutdown* da ferramenta C&M é apresentado na Figura 5.14.

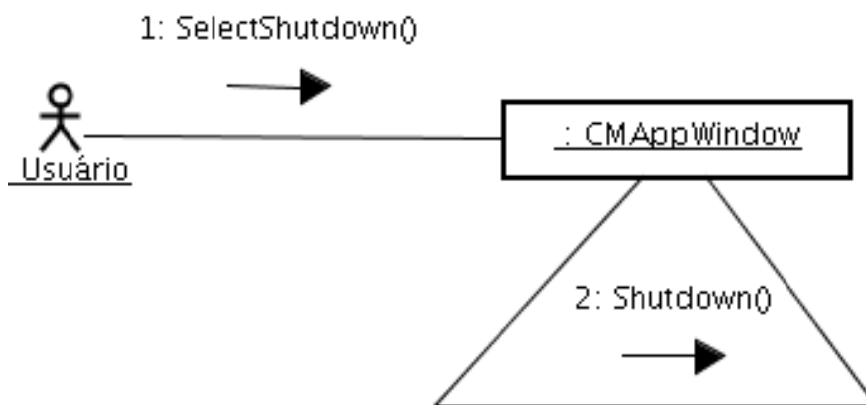


Figura 5.14- *Shutdown* da ferramenta C&M.

## 6 SIMPREBAL: IMPLEMENTAÇÃO COMPUTACIONAL

Este capítulo descreve o processo de implementação computacional do SIMPREBAL baseado na metodologia concebida. São apresentados os requisitos físicos, o processo de coleta de informação dos especialistas, os arquivos de regras, os modelos das regras de produção para as camadas OSA-CBM e finalmente as classes necessárias implementadas na implementação computacional tanto no lado servidor quanto no lado cliente.

### 6.1 INTRODUÇÃO

O SIMPREBAL é um dos sistemas que faz parte de um projeto bem maior chamado “modernização da área de automação de processos das usinas hidrelétricas de Balbina e Samuel”. A primeira fase foi executada em Balbina utilizando-se opções tecnológicas atuais para monitoração das grandezas dos ativos de geração e transmissão através do uso de instrumentação inteligente baseado em tecnologia FF, sistema de controle e supervisão (operação da usina) baseado em tecnologia Rockwell, sistema MES da Rockwell, o SIMPREBAL e o sistema de planejamento da Manutenção MAXIMO. O objetivo de utilizar estes sistemas é para contribuir na melhoria dos índices de desempenho: MTBR (*Mean Time Between Repairs*), qualidade da energia, manutenibilidade e confiabilidade.

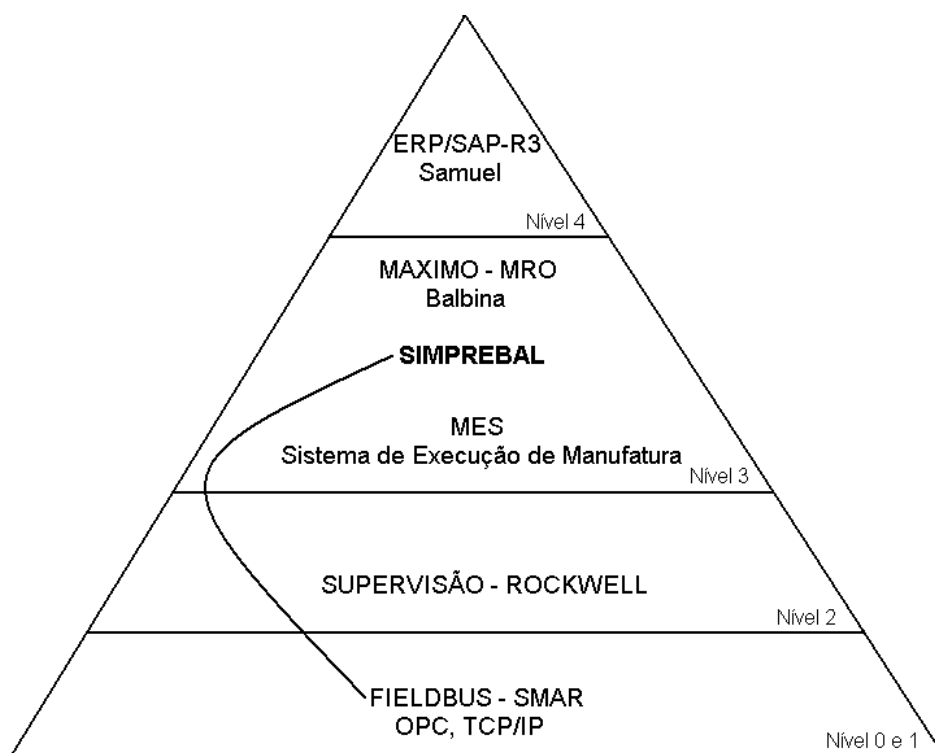


Figura 6.1- Modelo Hierárquico de Automação de Balbina e Samuel (Álvares, 2008).

A Figura 6.1 apresenta o modelo hierárquico de automação adotado em Balbina. Este modelo é baseado nas normas ISA-88 e ISA-95. ISA-95 define cinco níveis hierárquicos em companhias industriais: Nível 0, 1, 2, 3 e 4. O foco de ISA-88 é até o nível 2. ISA-95 focaliza sobre os níveis 3 e 4 (Álvares, 2008).

O objetivo dos Níveis 0, 1 e 2 é o controle dos equipamentos. O nível 3 é onde se encontra o MES que é a interface entre os níveis de processo e o nível de gestão da empresa (níveis 3 e 4). No nível 3 executa-se atividades como operação e planejamento e a manutenção baseada em condição da Usina Hidrelétrica de Balbina. No nível 4 chamado de sistema executa atividades de logística, finanças, planejamento estratégico, vendas e mercado (Álvares, 2008).

### **6.1.1 Requisitos Físicos**

Na implementação computacional da metodologia apresentada anteriormente, propõe-se o desenvolvimento de um subsistema de aquisição de dados e processamento inteligente que será chamado de *I-Kernel*. O subsistema *I-Kernel* visa o apoio à construção do sistema inteligente de MBC, e sua concepção é a de um subsistema genérico, a ser desenvolvido em linguagem Java, capaz de realizar a aquisição de dados a partir de bancos de dados e equipamentos acessíveis via servidor OPC, e seu processamento por meio de regras de produção.

Com o objetivo de tornar o desenvolvimento do *I-Kernel* uma atividade mais rápida, propõe-se integrar diversos componentes já existentes, promovendo um desenvolvimento orientado ao reuso. Da mesma forma, o *I-Kernel*, dada sua concepção na forma de um componente de software, poderá ser reutilizado em projetos futuros que demandem soluções de processamento inteligentes envolvendo o processamento de regras de produção.

Dentre os componentes que se almeja integrar encontram-se as ferramentas JESS, para promover o processamento de regras de sistemas baseados em regras, o JDBC para acesso ao banco de dados e o JNI para o acesso aos servidores OPC que dão acesso direto aos instrumentos.

Para o desenvolvimento do sistema inteligente proposto neste trabalho é necessário que exista uma rede industrial, ligando o sistema com os sensores conectados aos



equipamentos de campo. Os equipamentos e instrumentos usados para monitorar as condições das máquinas são mostrados na Figura 6.2. Na implementação usa-se a instrumentação *fieldbus*, onde os instrumentos são agrupados em uma rede por canais e conectados a uma DFI. Cada DFI tem quatro canais de comunicação, geralmente um é de reserva. O servidor OPC armazenada os itens (valores e outros parâmetros dos instrumentos) da instrumentação. Pode-se obter informações dos instrumentos através da conexão com o servidor OPC. No banco de dados é armazenado: histórico de anomalias, tomadas de decisão, variáveis de processo, usuários e dados dos equipamentos.

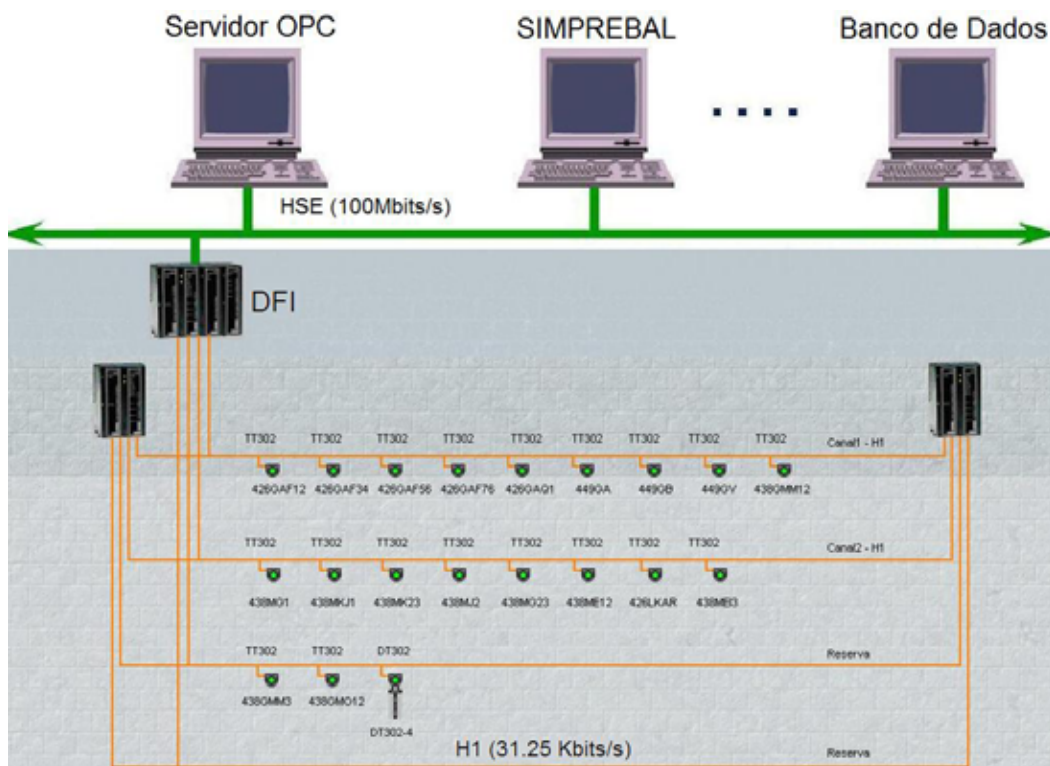


Figura 6.2- Instrumentos e equipamentos.

Na Figura 6.3 mostram-se uma série de computadores, onde diferentes serviços estão instalados. Apesar de mostrar diferentes máquinas, poderia-se assumir, sem prejuízo do conceito, que estes serviços estejam instalados todos em uma mesma máquina. A única restrição que se apresenta neste sentido é que tanto o módulo *Confmonittool* como o módulo *I-Kernel* necessariamente precisam estar instalados na mesma máquina. Essa restrição ocorre, pois o módulo *Confmonittool* será um Applet Java que se utilizará da rede para se comunicar com o módulo *I-Kernel*.

Por restrições de segurança um *Applet* Java só pode se comunicar com a mesma máquina em que foi carregado acaba sendo necessário que o módulo *I-Kernel* seja instalado na

mesma máquina que abriga o servidor *Web*, pois caso contrário o *Applet* não conseguirá se comunicar com o módulo *I-Kernel*.

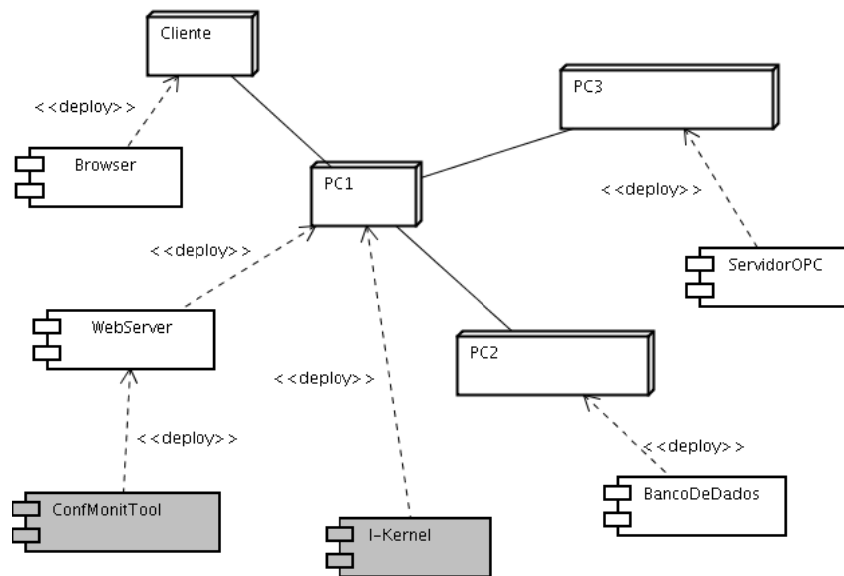


Figura 6.3- Requisitos físicos do SIMPREBAL.

Os módulos não marcados no diagrama não fazem parte do desenvolvimento, sendo considerados como reuso de software. Entretanto, o SIMPREBAL deve interagir com tais sistemas, o que justifica sua indicação na Figura 6.3. Dentre os serviços que serão incorporados por reuso, estão um servidor *web*, um Banco de Dados compatível com o JDBC e um servidor OPC que deve ser acessível por meio de JNI.

Os dois componentes de software que serão desenvolvidos no projeto são os aplicativos *I-Kernel* e *Confmonittool*. Estes aplicativos serão disponibilizados na forma de arquivos JAR (*Java Archive*) e deverão ser executados independentemente por meio de uma máquina virtual Java. O módulo *I-Kernel* é uma aplicação Java *standalone*, responsável pela aquisição de dados dos equipamentos da usina, por meio do banco de dados e dos equipamentos via OPC, seu processamento inteligente acontece visando detectar situações de manutenção preditiva, e eventualmente sugestões de tomada de decisão. O módulo *Confmonittool* é um Applet Java, carregado a partir de uma página *Web* armazenada no servidor *Web*, e é responsável pelas tarefas de configuração de parâmetros, bem como o monitoramento ativo das variáveis consideradas importantes, exibidas por meio de um sinótico.

### 6.1.2 Extração do conhecimento dos especialistas

O engenheiro de conhecimento foi quem coletou informações baseado nas experiências dos especialistas na área de manutenção e operação. Esta coleta foi feita com entrevistas, troca de idéias, comparando as informações obtidas com outros especialistas na área. As informações são organizadas em regras de produção com condições e ações, ou seja, para uma condição de um equipamento quais são as ações que os operadores ou mantenedores adotam. Nesta base de conhecimento fica armazenado todo o conhecimento adquirido dos especialistas. Durante o processamento inteligente é essa base de conhecimento que auxilia o raciocínio do sistema inteligente. Estes arquivos são gerados usando a linguagem CLIPS e executados na máquina de inferência da *shell* JESS.

A seguir apresenta-se um exemplo de uma regra de produção simples do sistema do mancal guia da turbina em pseudocódigo. Um exemplo mais complexo é apresentado em Kubiak *et al.* (2002) para um caso aplicado ao diagnóstico de falhas em turbinas a gás.

SE Pressão de óleo na cuba  $\leq$  0.25

ENTÃO

Condição: Baixo nível de óleo na cuba do mancal guia da turbina

CAUSA PROVÁVEL

- Vazamento de óleo pelas folgas nas vedações e conexões do mancal.
- Vazamento de óleo devido à corrosão na tubulação.
- Vazamento de óleo e contaminação com água pela folga existente entre as sapatas e o eixo ou por desalinhamento das sapatas.

RECOMENDAÇÕES DE MANUTENÇÃO

- Verificar estanqueidade das tubulações, válvulas e tampas de acesso.
- Verificar aperto dos parafusos de fixação da tampa e suporte da cuba.
- Realizar alinhamento e ajuste de folga das sapatas.
- Verificar estado geral quanto à corrosão.
- Restituir a carga de óleo.

As regras de produção são formadas por um conjunto de estruturas Se CONDIÇÃO Então AÇÃO. A linguagem CLIPS segue a mesma idéia, porém com sintaxe diferente.

(defrule SIMPREBAL-ON ;;Nome da regra

```
(Tag {label == "_simplerebaloff" && value == "0"}); Se teste de condição  
=> ; então  
(assert (SIMPREBAL ON)) ); ação
```

Toda regra em clips necessita ter um nome para que se possa diferenciar uma regra da outra. No exemplo acima o nome da regra é SIMPREBAL-ON. Após definido o nome da regra inicia-se o teste de condição. No exemplo acima o teste de condição verifica se o *label* de nome *simplerebaloff* recebe o valor 0. Isso significa que o *label simplerebaloff* recebe o valor 0 que é equivalente a *false*, então pode-se concluir que se *simplerebaloff* é falso, então *simplerebal on* é verdadeiro. É exatamente isso que a parte referente a ação (`assert (SIMPREBAL ON)`) quer dizer. Assim a ação está definida e a regra encerrada.

### 6.1.3 Arquivos de regras

A base de conhecimento é composta por 16 arquivos de regras de produção. Estes são divididos de acordo com o número de DFIs que existe na Usina (são três DFIs para cada UGH): 1a, 1b, 1c, 2a, 2b, 2c e assim sucessivamente até as DFIs referentes à UGH 5. Existe ainda um arquivo chamado “regras.clp” que testa a conexão do Simprebal.

O processamento de regras segue uma seqüência de execução. Um primeiro arquivo (`regras.clp`) é usada para avaliar o estado de conectividade (*connected* ou *disconnected*) do servidor OPC e das DFIs. Se o servidor OPC está conectado o sistema verifica a conectividade das DFIs. Se uma DFI está conectado o sistema processará suas regras respectivas. Cada um dos quinze arquivos de regras para as DFI serão processadas dependendo do estado de conexão da sua respectiva DFI para quem foi desenvolvida. Estes procedimentos é ilustrado na Figura 6.4.

O arquivo de regras (*i.e.* `regrasdfi1a.clp`) faz o teste para todas as tags que estão presentes na DFI 1a. O primeiro bloco de regras testa o processamento de sinal OPC da DFI, o segundo bloco de regras testa o processamento de sinal *fieldbus* da DFI. Depois da camada de processamento de sinal vem a camada de monitoração de condição que tem a finalidade de monitorar as faixas dos valores de operação das tags. Quatro valores são testados: um valor indicando a condição normal; um intervalo de valores indicando a condição alto/baixo, esse estado implica no estado de alerta do sistema e compreende uma faixa de valores próxima a faixa de alarme; o valor de alarme, que na verdade é um intervalo de valores; e o valor de *trip*. Os valores de alarme e *trip* são definidos com base nos dados do

sistema monitorado, os valores normal e alerta foram definidos para o SIMPREBAL e o valor de alerta tem a finalidade de permitir a ação proativa do sistema.

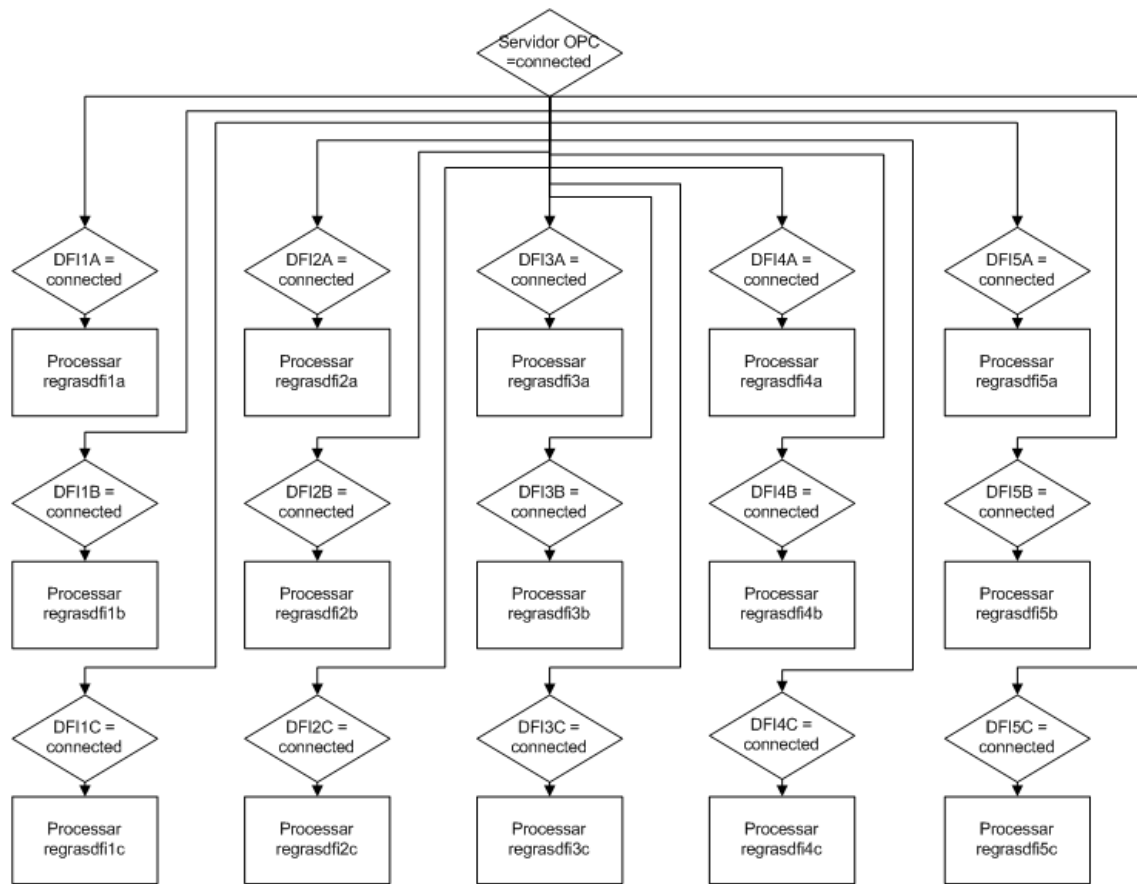


Figura 6.4- Regras de produção.

## 6.2 REGRAS DE PRODUÇÃO PARA AS CAMADAS OSA-CBM

A seguir apresenta-se a arquitetura de regras de produção para as camadas OSA-CBM usado na implementação do SIMPREBAL. Os modelos de regras desenvolvidas são para as camadas processamento de sinal, monitoração de condição, avaliação de saúde e tomada de decisão.

### 6.2.1 Estrutura de Regras do processamento de sinal

Nesta camada são implementados dois tipos de processamento. O processamento de sinal OPC verifica a qualidade do sinal do servidor OPC e o processamento de sinal *fielbus* analisa a qualidade do sinal da instrumentação. Os valores dos campos da classe Tag usado no processamento de regras nesta camada e nas seguintes é calculado segundo o Apêndice C.

### 6.2.1.1 Processamento de sinal OPC

O modelo de regras usada neste processamento é apresentado na Tabela 6.1, onde as condições destas regras são as variáveis *dfiXY*, onde X é o número da UGH (X=1, 2, 3, 4 ou 5), e Y é a DFI a que está conectada o instrumento (Y = a, b ou c). Outras condições das regras são os valores dos campos *quality* e *subquality* da classe Tag. Os subseqüentes destas regras são os 5 códigos, uma de boa comunicação e as outras para a condição de falha na comunicação. Todos os códigos dos novos fatos gerados pelas regras vão acompanhados do *label* da Tag processada.

Tabela 6.1- Modelo de regras para processamento de sinal OPC.

Se	Então
$dfi == dfiXY$ e $quality == 3$	COM-GOOD ?label
( $dfi == dfiXY$ e $quality == 1$ ) ou ( $dfi == dfiXY$ e $quality == 0$ e $subquality == 0$ ) ou ( $dfi == dfiXY$ e $quality == 0$ e $subquality == 3$ ) ou ( $dfi == dfiXY$ e $quality == 0$ e $subquality == 4$ ) ou ( $dfi == dfiXY$ e $quality == 0$ e $subquality == 5$ ) ou ( $dfi == dfiXY$ e $quality == 0$ e $subquality == 6$ ) ou ( $dfi == dfiXY$ e $quality == 0$ e $subquality == 8$ )	COM-BAD-0 ?label
$dfi == dfiXY$ e $quality == 0$ e $subquality == 1$	COM-BAD-1 ?label
$dfi == dfiXY$ e $quality == 0$ e $subquality == 2$	COM-BAD-2 ?label
$dfi == dfiXY$ e $quality == 0$ e $subquality == 7$	COM-BAD-3 ?label

Tabela 6.2- Modelo de regras para processamento de sinal *Fieldbus*.

Se	Então
(COM-GOOD ?label) e ( $(dfi == dfiXY$ e $status == 2)$ ou ( $dfi == dfiXY$ e $status == 3$ ))	signal-GOOD ?label
(COM-GOOD ?label) e ( $(dfi == dfiXY$ e $status == 1)$ ou ( $dfi == dfiXY$ e $status == 0$ e $substatus == 0$ ) ou ( $dfi == dfiXY$ e $status == 0$ e $substatus >= 5$ ))	signal-BAD-0 ?label
(COM-GOOD ?label) e ( $dfi == dfiXY$ e $status == 0$ e $substatus == 1$ )	signal-BAD-1 ?label
(COM-GOOD ?label) e ( $dfi == dfiXY$ e $status == 0$ e $substatus == 2$ )	signal-BAD-2 ?label
(COM-GOOD ?label) e ( $dfi == dfiXY$ e $status == 0$ e $substatus == 3$ )	signal-BAD-3 ?label
(COM-GOOD ?label) e ( $dfi == dfiXY$ e $status == 0$ e $substatus == 4$ )	signal-BAD-4 ?label

### 6.2.1.2 Processamento de sinal *fieldbus*

Na avaliação do sinal da instrumentação *fieldbus*, uma condição é que o sinal OPC seja de boa qualidade. Outras condições do modelo de regras, mostradas na Tabela 6.2, são os campos da classe Tag, *status* e *substatus*. Estas regras geram novos fatos para o estado do sinal *fieldbus*, acompanhado do *label* da Tag processada.

### 6.2.2 Estrutura de Regras da monitoração de condição

Na Tabela 6.3, mostra-se o modelos de regras usadas para monitoramento da condição de operação. Estas regras verificam a condição do valor medido pelo instrumento. As condições deste modelo de regras são a comparação do valor medido com seus valores limites de operação  $V_{high}$ ,  $V_{alarm}$ , e  $V_{trip}$ ; e os conseqüentes são as quatro faixas de operação: NORMAL, ALTO, ALARME, e TRIP.

Tabela 6.3- Modelo de regras para faixas de operação.

Se	Então
$value \leq V_{high}$	condition-NORMAL ?label
$V_{high} \leq value \leq V_{alarm}$	condition-HIGH ?label
$V_{alarm} \leq value \leq V_{trip}$	condition-ALARM ?label
$V_{trip} \leq value$	condition-TRIP ?label

O modelo de regras mostrado na Tabela 6.4 usa como condições os fatos gerados pelo modelo da Tabela 6.3 e da Tabela 6.2. Com as condições apresentadas, o modelo de regras gera como conseqüente um código de monitoração de condição (COD\_MC).

Tabela 6.4- Modelo de regras para monitoração de condição.

Se	Então
signal-GOOD ?label e condition-NORMAL ?label	condition COD_MC-normal
signal-GOOD ?label e condition-HIGH ?label	condition COD_MC-alto
signal-GOOD ?label e condition-ALARM ?label	condition COD_MC-alarme
signal-GOOD ?label e condition-TRIP ?label	condition COD_MC-trip

### 6.2.3 Estrutura de Regras da avaliação de saúde

O modelo de regras para diagnóstico dos canais de comunicação *fieldbus* é mostrado na Tabela 6.5. Este modelo de regras tem como condições os códigos gerados pelo modelo da Tabela 6.1 acompanhado dos *labels* dos instrumentos conectados ao canal *fieldbus*. Este modelo gera ações (*i.e.* diagnostic G1A-canalN-0, N = 1, 2 ou 3, é o número do canal).

Tabela 6.5- Modelo de regras para diagnóstico dos canais de comunicação *fieldbus*.

Se	Então
COM-BAD-0 ?labels	diagnostic G1A-canalN-0
COM-BAD-1 ?labels	diagnostic G1A-canalN-1
COM-BAD-2 ?labels	diagnostic G1A-canalN-2
COM-BAD-3 ?labels	diagnostic G1A-canalN-3

Na Tabela 6.6 mostra-se o modelo de regras para o diagnóstico dos sinais da instrumentação *fieldbus*. As condições destas regras são os fatos gerados pelo modelo mostrado na Tabela 6.2. Este modelo de regras gera ações em forma de códigos de diagnóstico de anomalias da instrumentação *fieldbus*.

Tabela 6.6- Modelo de regras para diagnóstico da instrumentação *fieldbus*.

Se	Então
signal-BAD-0 ?label	gui12 COD_DIAG#EMAILS gui14 SISTEMA-amarelo
signal-BAD-1 ?label	gui12 COD_DIAG#EMAILS gui14 SISTEMA-amarelo
signal-BAD-2 ?label	gui12 COD_DIAG#EMAILS gui14 SISTEMA-amarelo
signal-BAD-3 ?label	gui12 COD_DIAG#EMAILS gui14 SISTEMA-amarelo
signal-BAD-4 ?label	gui12 COD_DIAG#EMAILS gui14 SISTEMA-amarelo

Tabela 6.7- Modelo de regras para diagnóstico da monitoração de condição.

Se	Então
condition COD_MC-alto	gui11 COD_DIAG#EMAILS gui14 SISTEMA-amarelo
condition COD_MC-alarme	gui11 COD_DIAG#EMAILS gui14 SISTEMA-vermelho
condition COD_MC-trip	gui11 COD_DIAG#EMAILS gui14 SISTEMA-vermelho



O modelo de regras da Tabela 6.7 tem como condições os fatos gerados pelo modelo na camada de monitoração de condição (Tabela 6.4). Este modelo de regras gera ações em forma de códigos de diagnóstico das anomalias das condições de operação.

Os conseqüentes ou ações gerados pelos modelos da Tabela 6.6 e Tabela 6.7 são códigos de diagnóstico (COD\_DIAG), lista de emails (EMAILS) para o envio das ordens de serviço (OS), para a visualização na camada de apresentação e enviado o nome do sistema (SISTEMA) e a cor de alerta.

O comando *printout guiXY "texto"* envia uma mensagem para o código Java contendo o "texto". Neste caso, X é um número de 1 a 5, correspondente ao número da UGH, e Y é o número 1, 2 ou 4, que possuem os seguintes significados:

- 1: A falha em questão é relativa a um equipamento da UGH.
- 2: A falha em questão é relativa à instrumentação.
- 4: Indica uma informação interna para fazer o cliente piscar em amarelo ou vermelho, de acordo com o tipo de falha (ALERTA, ALARME, TRIP, STATUS BAD, QUALITY BAD, OFFLINE).

#### 6.2.4 Estrutura de Regras da tomada de decisão

Para integrar as informações necessárias visando gerar sugestões de ações de manutenção, o processo de tomada de decisão é feito através do modelo de regras ilustrado na Tabela 6.8. As condições destas regras são as saídas da camada do diagnóstico. Os conseqüentes ou ações gerados pelo modelo são códigos de decisão (COD\_DCS), lista de emails (EMAILS) para o envio das OS, para a visualização na camada de apresentação se enviará o nome do sistema (SISTEMA) e a cor de alerta.

Tabela 6.8- Modelo de regras para tomada de decisão.

Se	Então
diagnostic G1A-canalN-0	gui14 SISTEMA-amarelo gui12 COD_DCS#EMAILS
diagnostic G1A-canalN-1	gui14 SISTEMA-amarelo gui12 COD_DCS#EMAILS
diagnostic G1A-canalN-2	gui14 SISTEMA-amarelo gui12 COD_DCS#EMAILS
diagnostic G1A-canalN-3	gui14 SISTEMA-amarelo gui12 COD_DCS#EMAILS

### 6.3 CLASSES IMPLEMENTADAS NO SIMPREBAL

Na implementação computacional dos módulos do SIMPREBAL, a aplicação *I-Kernel* e a ferramenta *Confmonittool* usaram-se classes e métodos em Java. Cada uma destas classes e métodos para estes dois módulos são descritos a seguir.

#### 6.3.1 Classes do *I-Kernel*

As principais classes da aplicação *I-kernel* mostradas na Figura 6.5 são descritos a seguir:

- *Tag*: esta classe contém os dados das variáveis de processo e é usada no processamento inteligente pelas regras de produção;
- *Configuration*: os parâmetros de configuração são obtidos através desta classe;
- *OPCProxy*: os itens em tempo real do servidor OPC são coletados através desta classe;

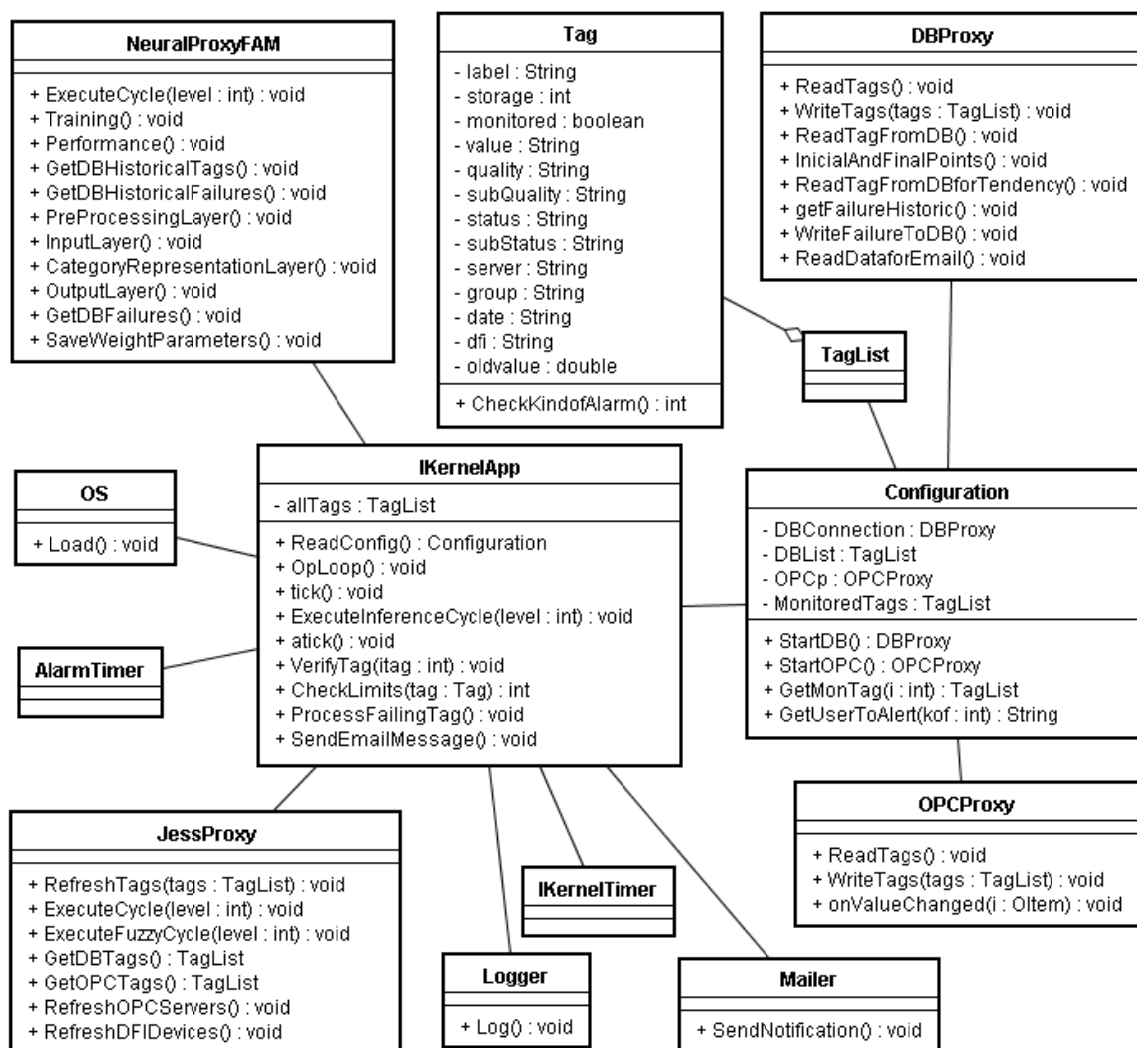


Figura 6.5- Classes do I-Kernel

- *DBProxy*: através desta classe o sistema coleta e armazena informações no banco de dados;
- *Mailer*: envia as ordens de serviço com sugestões de ações de manutenção são enviadas aos usuários via email;
- *JessProxy*: processamento de regras;
- *Logger*: registra os eventos de execução do sistema;
- *NeuralProxyFAM*: é uma classe que será implementada nos trabalhos futuros para prognóstico de falhas.

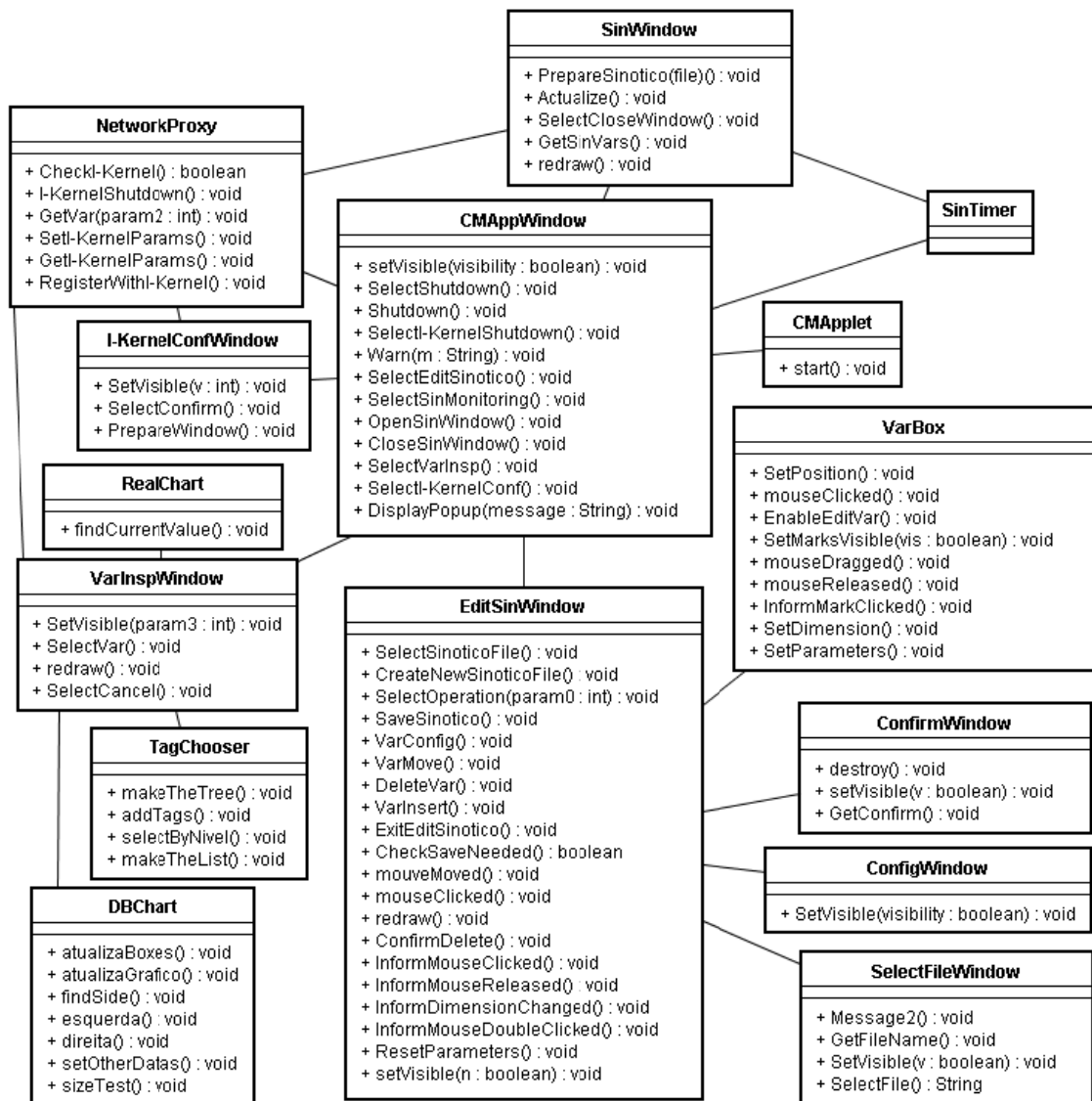


Figura 6.6- Classes do *Confmonitool*.

### 6.3.2 Classes do *Confmonittool*

As principais classes da ferramenta *Confmonittool* mostradas na Figura 6.6 são descritos a seguir:

- *NetworkProxy*: é através desta classe que o cliente solicita e recebe as informações do servidor;
- *VarInspWindo*: apresenta uma tela para inspeção de variáveis, com colunas de valor, qualidade e a data que foi coletada;
- *TagChooser*: a escolha das tags da árvore hierárquica para inspeção de variáveis é selecionada através desta classe;
- *RealChart*: suporta os gráficos de variáveis em tempo real;
- *DBChart*: o gráficos das variáveis históricas coletadas do banco de dados.

A implementação das classes tanto do *I-Kernel* quanto do *Confmonittool* deu como resultado o SIMPREBAL. A descrição de uso deste sistema é detalhada no Apêndice E.

## 7 ESTUDO DE CASO: GERADOR ELÉTRICO

Neste capítulo é apresentado um estudo de caso para demonstrar os métodos e algoritmos desenvolvidos na metodologia e implementados no SIMPREBAL. Consideraremos como sistema de estudo o sistema do gerador elétrico da usina hidrelétrica de Balbina, que é constituído por três subsistemas: gerador elétrico principal, resfriamento do gerador (radiadores), e regulação de tensão.

### 7.1 INTRODUÇÃO

Uma usina hidrelétrica é definida pela NBR-5460 (1992) como: “Instalação elétrica na qual a energia elétrica em escala industrial é obtida por conversão da energia gravitacional da água”. Esta instalação é um complexo arquitetônico e de equipamentos como mostrado na Figura 7.1, e tem por finalidade produzir energia elétrica através do aproveitamento do potencial hidráulico da água. A energia hidráulica é convertida em energia mecânica por meio de uma turbina hidráulica, que por sua vez é convertida em energia elétrica por meio de um gerador, sendo a energia elétrica transmitida para uma ou mais linhas de transmissão que é interligada à rede de distribuição.



Figura 7.1- Vista aérea da usina hidrelétrica de Balbina.

Um dos sistemas importantes em uma usina hidrelétrica é o gerador, tendo como função principal gerar eletricidade. A geração de eletricidade se dá através da conversão da energia mecânica contida na rotação do eixo que faz com que a intensidade de um campo magnético produzido por um uma série de ímãs que atravessam um conjunto de

enrolamentos varie no tempo, o que pela lei da indução de Faraday leva a indução de tensões nos terminais dos mesmos. Na Figura 7.2 mostram-se os cinco geradores da Usina Hidrelétrica de Balbina, cada um deles gera até 50 megawatts, a capacidade total da usina é 250 megawatts. Cada gerador é feito de alguns componentes básicos: eixo, excitador, rotor, e estator. Estes geradores são de baixa rotação (105.88 rpm).



Figura 7.2- Vista superior dos 5 geradores.

Quando a turbina gira, o excitador envia corrente elétrica para o rotor. O rotor (Figura 7.3) é uma série de grandes eletroímãs que rodam dentro enrolamentos de cobre chamado estator (Figura 7.4). O estator do gerador elétrico é a parte fixa da máquina, montada em volta do rotor de forma que o mesmo possa girar em seu interior. O estator é constituído de um material ferromagnético envolto em um conjunto de enrolamentos distribuídos ao longo de sua circunferência. Os enrolamentos do estator são alimentados por um sistema de tensões alternadas trifásicas. Pelo estator circula toda a energia elétrica gerada, sendo que tanto a voltagem quanto a corrente elétrica que circulam são bastante elevadas em relação ao campo, que tem como função apenas produzir um campo magnético para

"excitar" a máquina de forma que seja possível a indução de tensões nos terminais dos enrolamentos do estator.



Figura 7.3- Rotor do gerador elétrico.

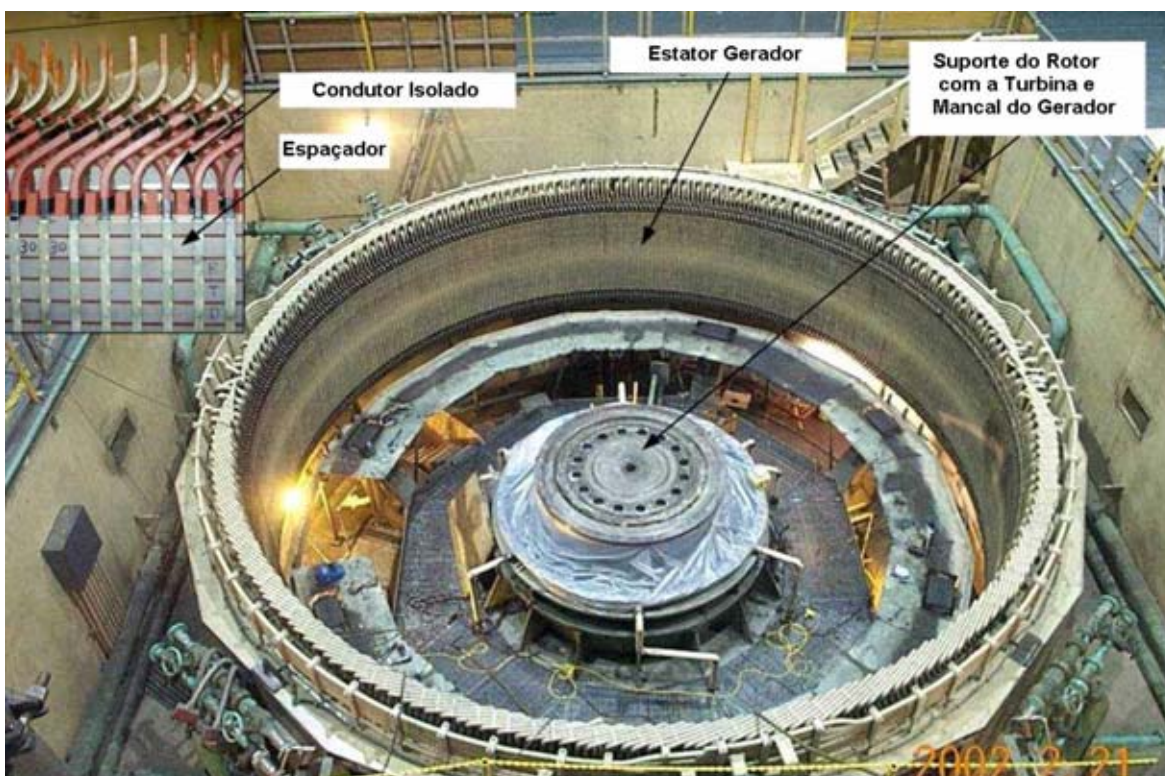


Figura 7.4- Estator do gerador elétrico.

### 7.1.1 Gerador elétrico principal

Os instrumentos instalados neste subsistema são o transmissor de temperatura dos enrolamentos e núcleos do estator nas três fases A, V e B. As Tags, valores de temperatura em funcionamento normal PV (*Process Value*), alto (*High*), alarme (*Alm*), *Trip* e a faixa de medição do instrumento associado à UGH 01 da usina de Balbina são mostradas na Tabela 7.1.

Tabela 7.1- Tags associadas ao subsistema gerador elétrico principal.

Tag	Descrição	PV	High	Alm	Trip	Faixa
149G1A	Temperatura do enrolamento do estator fase A	85	105	130	155	0-200 °C
149G1B	Temperatura do enrolamento do estator fase B	85	105	130	155	0-200 °C
149G1V	Temperatura do enrolamento do estator fase V	85	105	130	155	0-200 °C
149G2A	Temperatura do núcleo do estator fase A	80	105	130	-	0-200 °C
149G2B	Temperatura do núcleo do estator fase B	80	105	130	-	0-200 °C
149G2V	Temperatura do núcleo do estator fase V	80	105	130	-	0-200 °C

### 7.1.2 Resfriamento do gerador

O subsistema de resfriamento do gerador é constituído por oito radiadores. Em cada um dos radiadores é instalado um transmissor de temperatura de ar frio. Outro instrumento é o transmissor de temperatura de ar quente dos oito radiadores. Na Tabela 7.2, mostram-se as tags associadas a este subsistema para a UGH número 1 da usina hidrelétrica de Balbina.

### 7.1.3 Regulação de tensão

O subsistema regulação de tensão é formado por transformadores de excitação que enviam corrente elétrica para o rotor. Os transmissores para as fases A, B e V são instalados no interior do enrolamento de cada fase. A lista de transmissores é mostrada na Tabela 7.3.



Tabela 7.2- Tags associadas ao subsistema resfriamento do gerador.

<b>Tag</b>	<b>Descrição</b>	<b>PV</b>	<b>High</b>	<b>Alm</b>	<b>Trip</b>	<b>Faixa</b>
126GAF1	Temperatura de ar frio do radiador n° 1	42	44	45	-	20-60 °C
126GAF2	Temperatura de ar frio do radiador n° 2	42	44	45	-	20-60 °C
126GAF3	Temperatura de ar frio do radiador n° 3	42	44	45	-	20-60 °C
126GAF4	Temperatura de ar frio do radiador n° 4	42	44	45	-	20-60 °C
126GAF5	Temperatura de ar frio do radiador n° 5	42	44	45	-	20-60 °C
126GAF6	Temperatura de ar frio do radiador n° 6	42	44	45	-	20-60 °C
126GAF7	Temperatura de ar frio do radiador n° 7	42	44	45	-	20-60 °C
126GAF8	Temperatura de ar frio do radiador n° 8	42	44	45	-	20-60 °C
126GAQ1	Temperatura de ar quente	65	70	75	85	20-100°C

Tabela 7.3- Tags associadas ao subsistema regulação de tensão.

<b>Tag</b>	<b>Descrição</b>	<b>PV</b>	<b>High</b>	<b>Alm</b>	<b>Trip</b>	<b>Faixa</b>
G149TEA1	Temperatura do trafo de excitação fase A enrolamento n° 1	96	100	110	130	0-200 °C
G149TEA2	Temperatura do trafo de excitação fase A enrolamento n° 2	96	100	110	130	0-200 °C
G149TEB1	Temperatura do trafo de excitação fase B enrolamento n° 1	93	100	110	130	0-200 °C
G149TEB2	Temperatura do trafo de excitação fase B enrolamento n° 2	93	100	110	130	0-200°C
G149TEV1	Temperatura do trafo de excitação fase V enrolamento n° 1	99	100	110	130	0-200 °C
G149TEV2	Temperatura do trafo de excitação fase V enrolamento n° 2	99	100	110	130	0-200 °C

## 7.2 RESULTADOS OBTIDOS

Os resultados gerados pelo SIMPREBAL são armazenados no banco de dados, mostrados através da interface gráfica do usuário e enviados via email em forma de OS. Uma forma de visualizar as anomalias ocorridas é no terceiro quadrante da Figura 7.5, onde são mostradas as trinta últimas ocorrências das anomalias detectadas pelo SIMPREBAL.

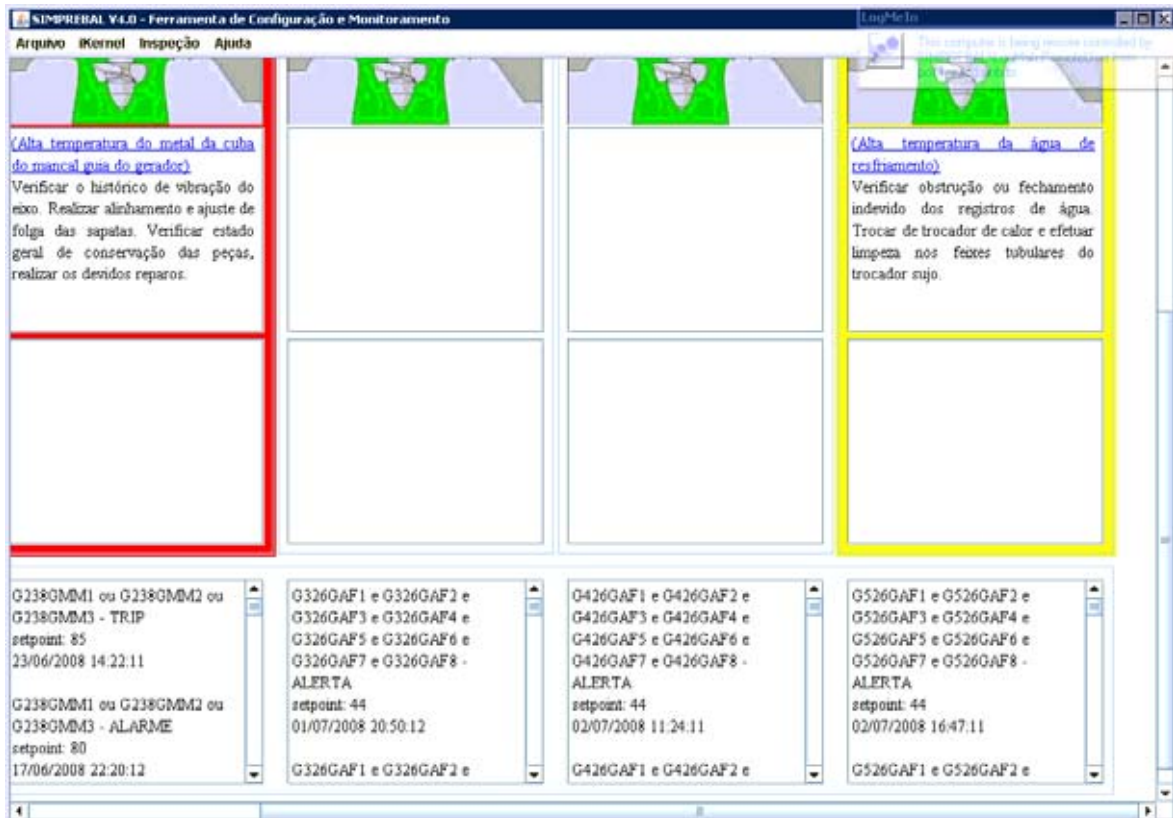


Figura 7.5- Histórico de Anomalias ocorridas.

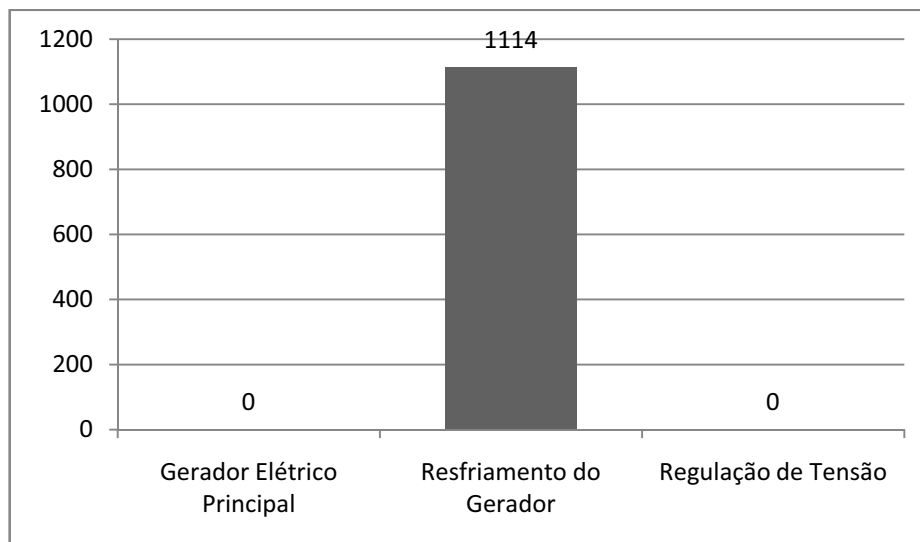


Figura 7.6- Anomalias detectadas nos subsistemas do gerador elétrico das UGHs.

Analisando os dados armazenados do banco de dados podem-se conhecer quantas anomalias apresentou cada um dos subsistemas. Na análise coletamos os dados das ocorrências dos últimos três meses. Na Figura 7.6 mostra-se o número de anomalias que o SIMPREBAL detectou nos três subsistemas dos cinco geradores elétricos. Esta análise corresponde aos meses de abril, maio e junho de 2008, pode-se ver que o único subsistema

que apresentou anomalias é o SRG (Subsistema de Resfriamento do Gerador). Sendo assim, o estudo de caso é baseado nas anomalias ocorridas no SRG em cada um das UGHs.

Na Figura 7.7 mostra-se o número de anomalias detectadas pelo SIMPREBAL no SRG das UGHs. Isto permite conhecer qual UGH apresentou mais sinais de alarme no seu SRG.

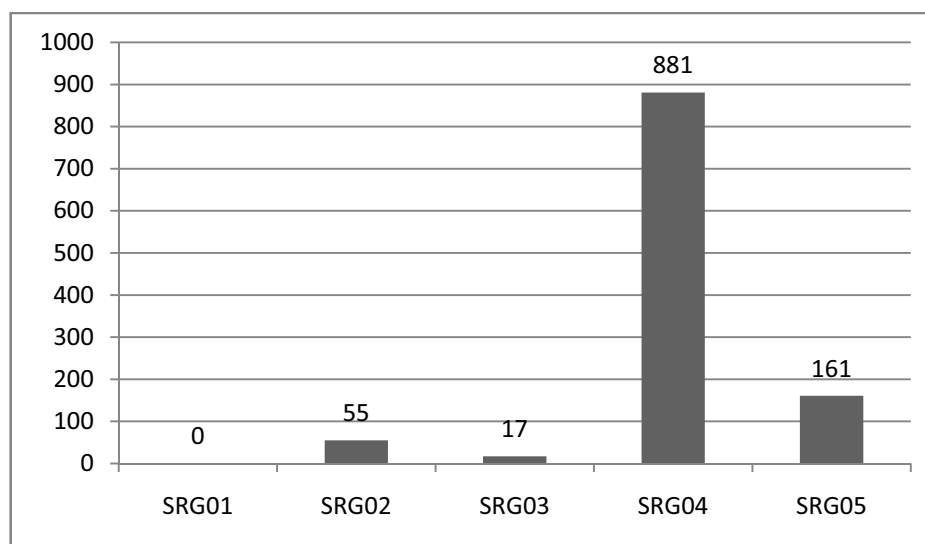


Figura 7.7- Anomalias detectadas no SRG das cinco UGHs.

A seguir analisam-se os dados das anomalias armazenadas no banco de dados dos SRG02 e SRG03, sendo que estes apresentaram um número de anomalias intermédias cujos dados podem ser de fácil análise. As anomalias apresentadas nos SRG04 e SRG05 seguiriam o mesmo procedimento de análise do que os SRG02 e SRG03. Na Figura 7.8, mostra-se o histórico de anomalias ocorridas no SRG02.

Na avaliação do estado de alerta do SRG considera-se as regras condicionadas a que o valor de temperatura de pelos menos seis dos oito transmissores de temperatura de ar frio atinja o valor limite de *setpoint*. Com o objetivo de ver os valores *online* das variáveis, a qualidade do sinal, a data e hora da ultima atualização, abrimos a tela de inspeção de variáveis mostrada na Figura 7.9, ali mostra-se as tags em análise. A tendência das variáveis *online* que permitem a visualização gráfica das variações do valor das variáveis em análise é mostrada na Figura 7.10. Conhecendo a data em que uma anomalia ocorreu, procura-se conhecer os valores que as variáveis associadas a estas anomalias tiveram no período de falha, a visualização gráfica é mostrado na Figura 7.11.

?	codigo	descricao	causa	data_inicio	severidade	setpoint
1	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 15:18:16	4	44
2	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 15:27:15	4	44
3	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 16:54:36	4	44
4	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 18:27:04	4	44
5	G226GAFD2	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 19:54:23	5	44
6	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 00:20:35	4	44
7	G226GAFD2	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 01:22:50	5	44
8	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 01:34:54	4	44
9	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 07:20:51	4	44
10	G226GAFD2	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 08:45:05	5	44
11	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 09:35:22	4	44
12	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 12:20:29	4	44
13	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 12:31:12	4	44
14	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 12:39:27	4	44
15	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 12:49:01	4	44
16	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 12:56:14	4	44
17	G226GAFD2	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 13:36:36	5	44
18	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 13:49:10	4	44
19	G226GAFD2	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 13:56:32	5	45
20	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 17:11:38	4	44
21	G226GAFD2	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 18:49:35	5	44
22	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-02 20:55:01	4	44
23	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-03 14:57:49	4	44
24	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-07 08:34:57	4	44
25	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-07 13:29:31	4	44
26	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-08 08:38:28	4	44
27	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-08 11:31:33	4	44
28	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-08 11:55:42	4	44
29	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-08 12:39:33	4	44
30	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-08 12:51:45	4	44
31	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-09 20:03:02	4	44
32	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-09 21:09:29	4	44
33	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-09 21:25:32	4	44
34	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-10 10:35:42	4	44
35	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-10 15:47:27	4	44
36	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-11 12:42:25	4	44
37	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-11 16:05:35	4	44
38	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-11 19:01:19	4	44
39	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-11 19:07:19	4	44
40	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-11 19:17:19	4	44
41	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-11 19:25:19	4	44
42	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-11 19:34:24	4	44
43	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-11 19:37:21	4	44
44	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-13 22:20:19	4	44
45	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-13 22:48:11	4	44
46	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-13 23:11:11	4	44
47	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-13 23:15:10	4	44
48	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-14 00:02:11	4	45
49	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-14 00:12:11	4	44
50	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-14 00:27:11	4	44
51	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-14 00:33:10	4	44
52	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-14 01:00:10	4	44
53	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-14 01:02:10	4	44
54	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-14 01:22:10	4	44
55	G226GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-14 03:00:10	4	44

Figura 7.8- Anomalias detectadas no SRG02.

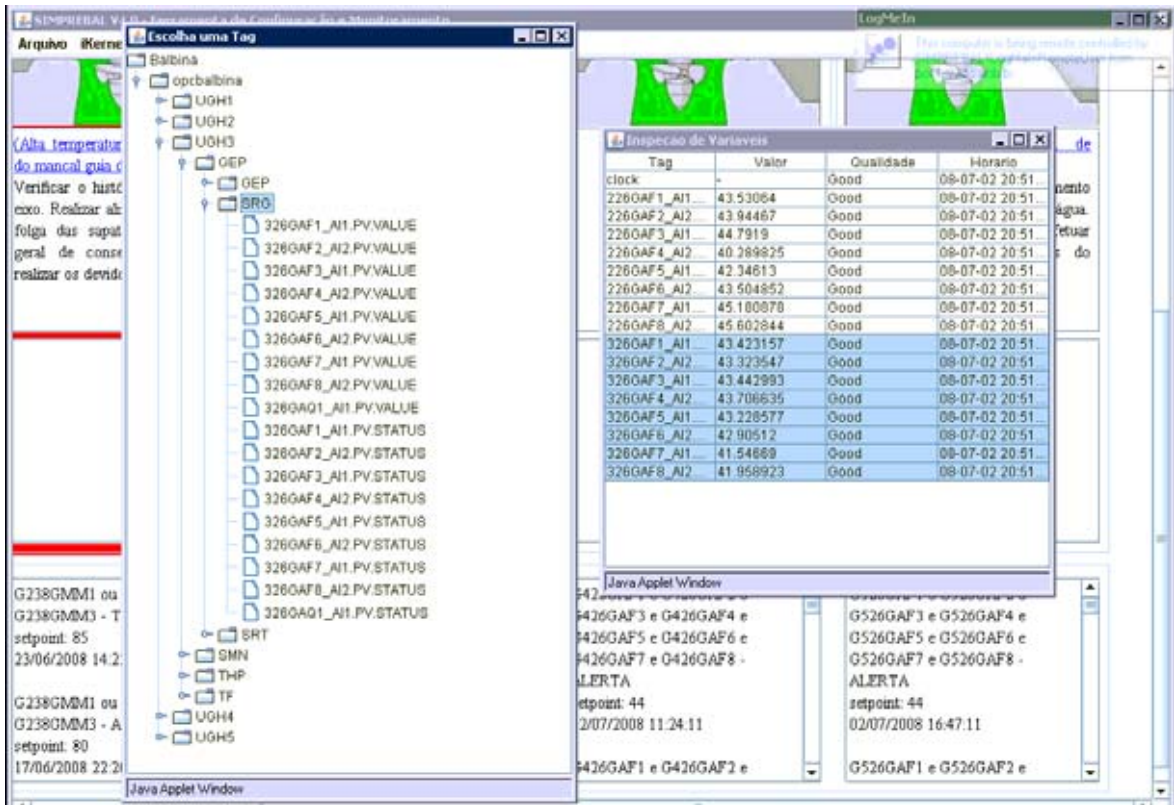


Figura 7.9- Inspeção de variáveis dos SRG02 e SRG03.

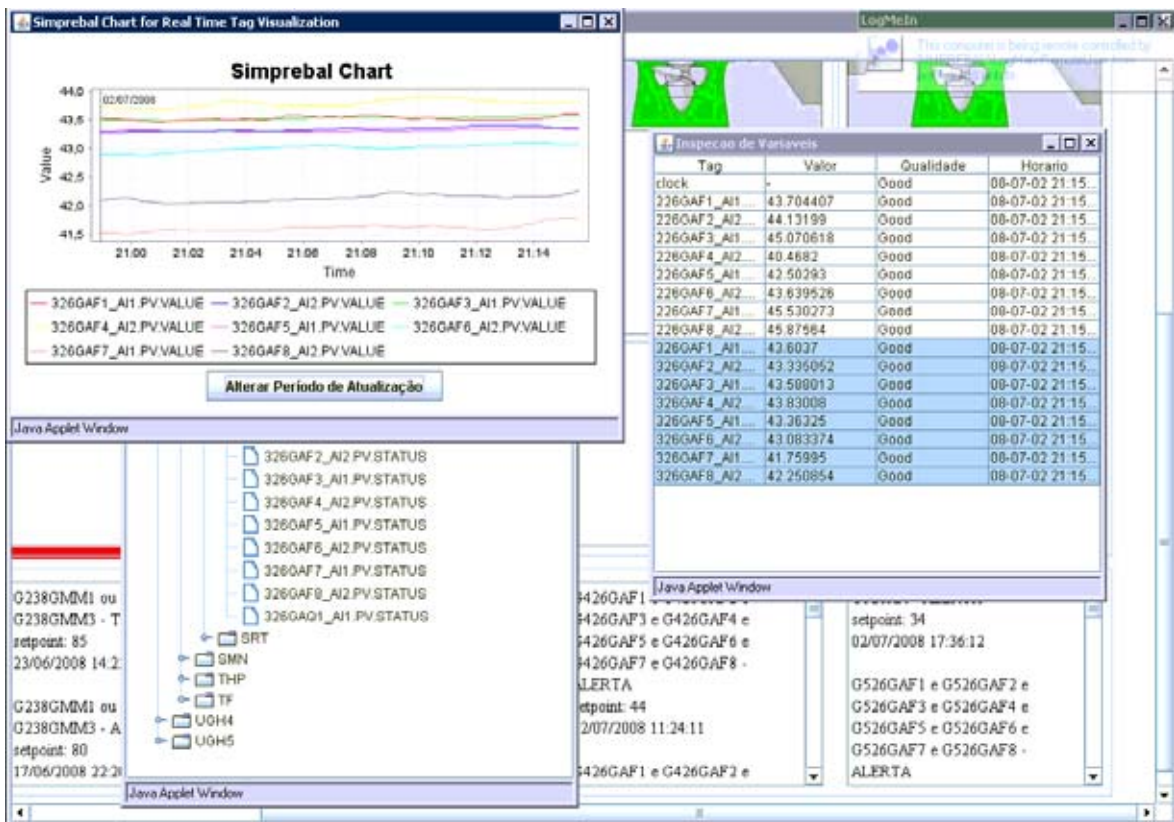


Figura 7.10- Gráfico de tendências de variáveis em tempo real.

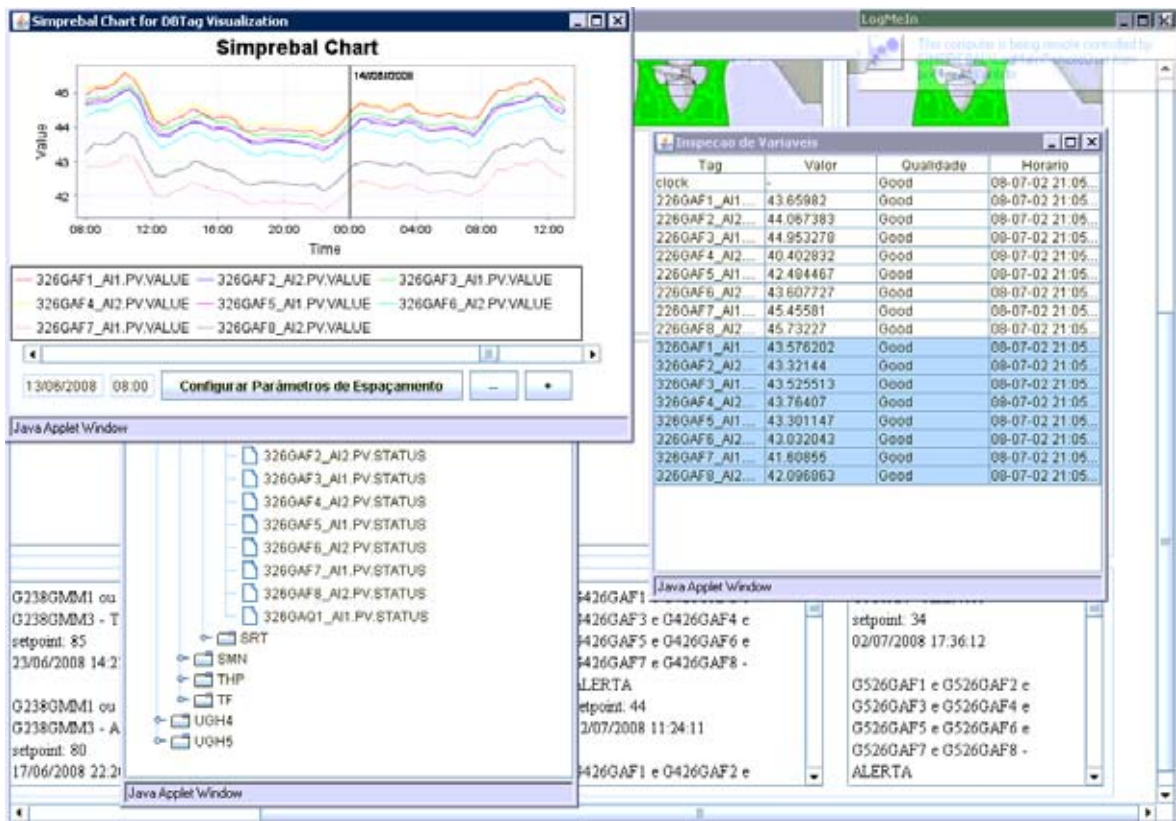


Figura 7.11- Gráfico de tendências de variáveis históricas.

Na Figura 7.12 mostra-se o gráfico histórico das variáveis associadas às anomalias detectadas pelo SIMPREBAL no SRG02. Nota-se que quando os valores de pelo menos seis das variáveis atingem o valor de *setpoint* (44 °C ) o sistema envia um alerta.

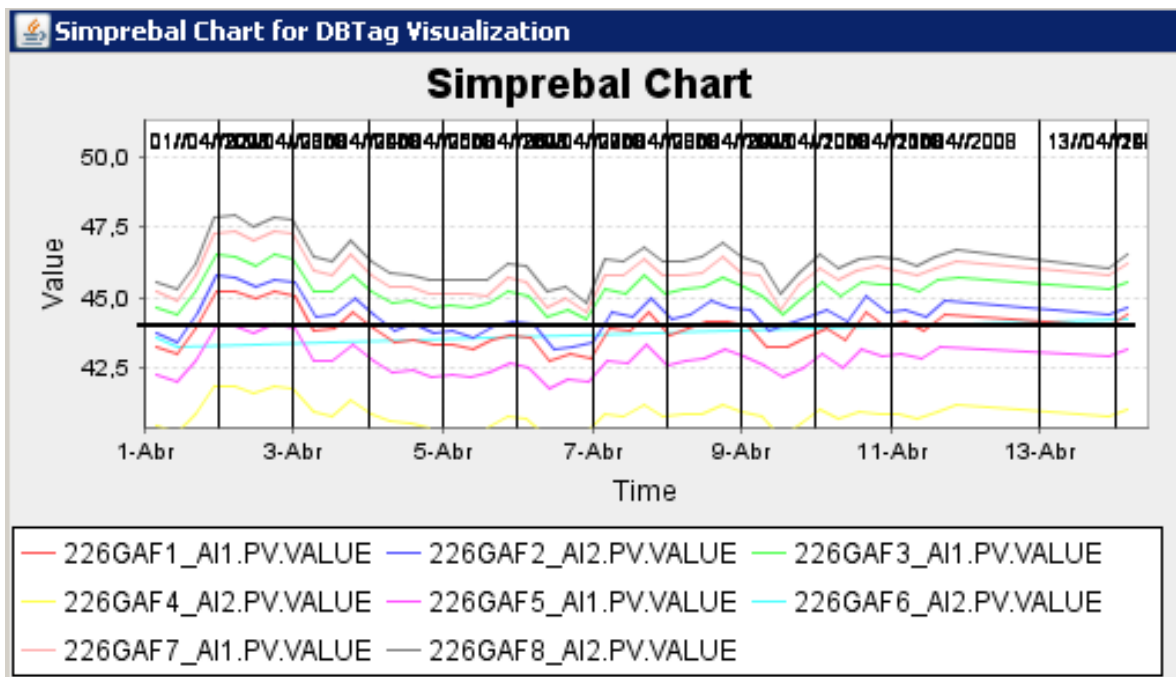


Figura 7.12- Gráfico das variáveis associadas às anomalias detectadas no SRG02.

As anomalias ocorridas no SRG03 são armazenadas no banco de dados do SIMPREBAL e mostradas na Figura 7.13. O gráfico das variáveis associadas é apresentado em três figuras. Na Figura 7.14 mostra-se o gráfico de tendências das variáveis associadas às anomalias produzidas no dia 01/04/2008. Na Figura 7.15 mostra-se o gráfico de tendências das variáveis associadas às anomalias produzidas nos dias 13/04/2008 e 14/04/2008. Na Figura 7.16 mostra-se o gráfico de tendências das variáveis associadas às anomalias produzidas no dia 25/04/2008.

?	codigo	descricao	causa	data_inicio	severidade	setpoint
1	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 21:05:18	4	44
2	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 21:12:18	4	44
3	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 21:19:24	4	44
4	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 21:23:23	4	44
5	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 21:26:25	4	44
6	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 21:48:29	4	44
7	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-04-01 22:07:30	4	44
35	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-06-13 09:29:11	4	44
36	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-06-13 09:37:12	4	44
37	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-06-13 09:42:11	4	44
38	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-06-13 10:42:11	4	44
39	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-06-14 10:52:11	4	44
40	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-06-14 11:01:11	4	44
41	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-06-14 11:06:11	4	44
42	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-06-25 14:52:11	4	44
43	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-06-25 14:58:12	4	44
44	G326GAFD1	Alta temperatura de ar frio dos radiadores	Sujeira nos radiadores	2008-06-25 15:07:11	4	44

Figura 7.13- Anomalias detectadas no SRG03.

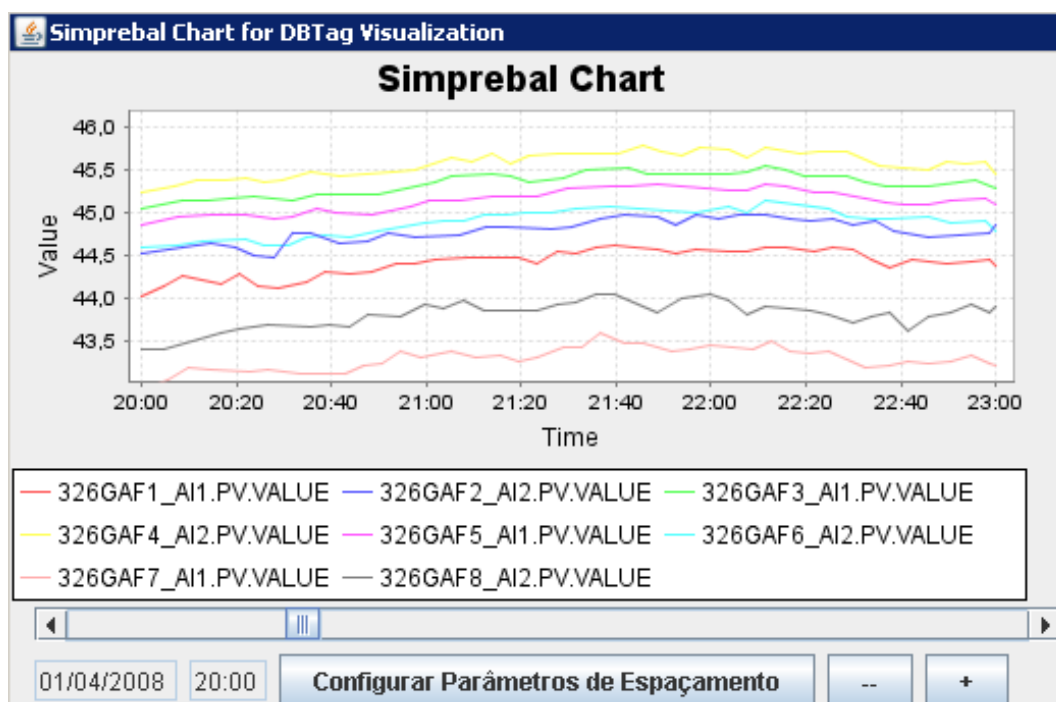


Figura 7.14- Gráfico das variáveis associadas às anomalias detectadas no SRG03.

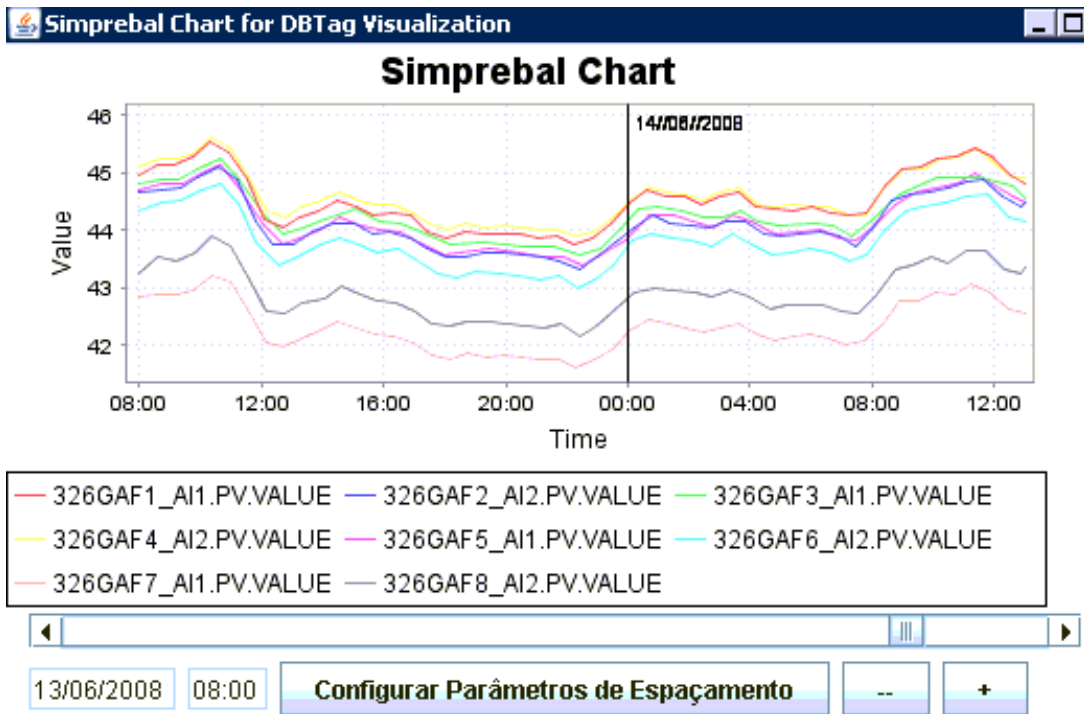


Figura 7.15- Gráfico das variáveis associadas às anomalias detectadas no SRG02.

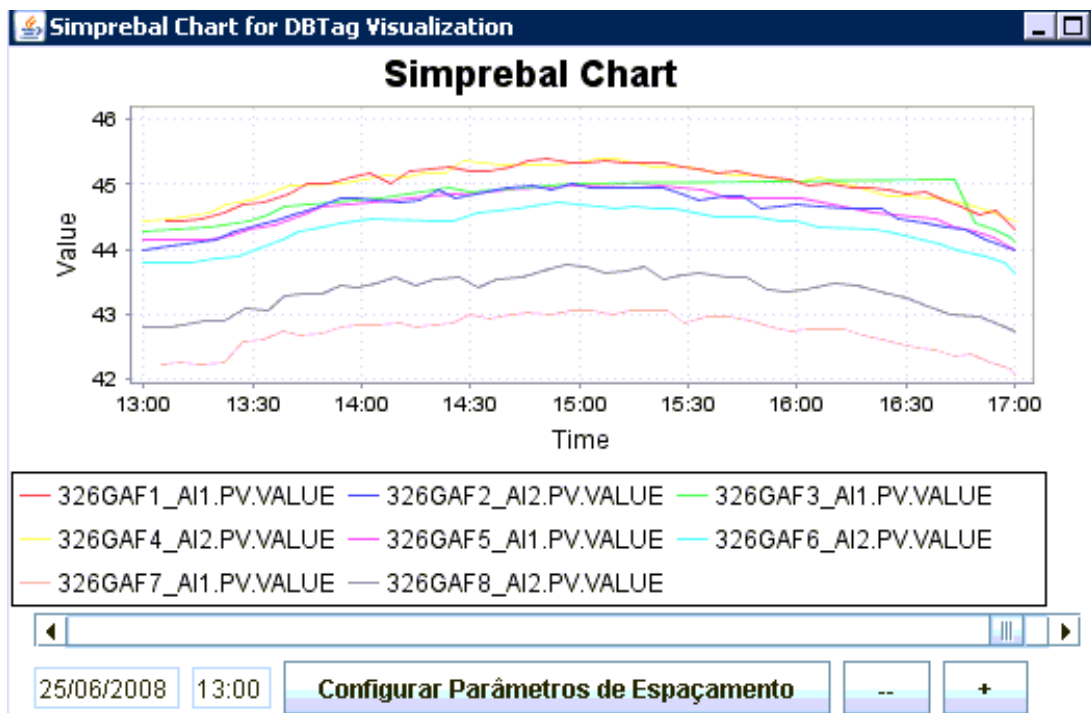


Figura 7.16- Gráfico das variáveis associadas às anomalias detectadas no SRG02.

A geração de sucessivas alertas de anomalias pelo SIMPREBAL deve-se ao fato de que o limite de alerta da temperatura de ar frio dos radiadores (44 °C) é muito próximo ao valor de temperatura em funcionamento normal (42 °C ou 43 °C). Do ponto de vista do operador, indica que o SRG está trabalhando no limite. No banco de dados estes alertas são



armazenados repetidamente, precisando redefinir a política de armazenamento de anomalias. Uma alternativa é agendar uma reunião com os operadores e ver a possibilidade mudar os valores limites de alerta.

Quando uma anomalia é detectada pelo SIMPREBAL, um email é enviado aos usuários cadastrados. Um exemplo de email recebido do SIMPREBAL é mostrado na Figura 7.17. As informações do email são: a localização da anomalia, o modo de falha, o nível de severidade, a descrição de anomalia, a causa e a sugestão de tomada de decisão.

```
Localizacao: sistema do gerador - sistema de resfriamento do gerador.  
Modo de falha: G526GAF1 e G526GAF2 e G526GAF3 e G526GAF4 e G526GAF5  
e G526GAF6 e G526GAF7 e G526GAF8 - ALERTA.  
Severidade: 4.  
Descricao: Alta temperatura de ar frio dos radiadores.  
Causa: Sujeira nos radiadores.  
Decisao: Alerta para possibilidade de efetuar limpeza interna dos radiadores..  
  
Manaus Energia  
SIMPREBAL
```

Figura 7.17- Email recebido do SIMPREBAL.

## 7.3 VALIDAÇÃO

O objetivo da validação de um sistema é determinar se o sistema realiza aquilo para o qual ele foi projetado e desenvolvido. Neste trabalho validamos as informações da base de conhecimento, o desempenho do servidor e a interface gráfica do usuário.

### 7.3.1 Base de conhecimento

A base de conhecimento foi implementada segundo as informações cedidas pelo pessoal de manutenção e operação. A primeira versão de regras tinha as informações da documentação existente de falhas cadastradas dos anos passados e as experiências das pessoas envolvidas na manutenção e operação da usina. Nas seguintes versões das regras foram se afinando em presença dos especialistas de manutenção e operação, comparando as saídas das regras produzidas usando valores de variáveis simuladas, com as ações que os especialistas adotariam se os valores dessas variáveis simuladas fossem reais.

Para o melhoramento contínuo das tomadas de decisão, foram considerados no desenvolvimento da interface com o usuário e no banco de dados, campos para realimentação de informação. Quando uma anomalia é detectada o sistema gera uma tomada de decisão, e os operadores e mantenedores devem preencher o campo *check* da Figura 7.18 para verificar se a sugestão gerada pelo sistema esta de acordo com o que é adotado pelo pessoal de operação e manutenção.

FATORES PARA AVALIAÇÃO DA FALHA				SUGESTÃO DE OS
SEVERIDADE	OCORRÊNCIA	DETECÇÃO	AVALIAÇÃO GERAL (NPR)	
7	2	1	14	Diminuir a corrente de excitação das resistências  <input checked="" type="checkbox"/> Declaro que esta sugestão de OS condiz com a realidade.

Figura 7.18- Tela para avaliação das tomadas de decisão.

### 7.3.2 Servidor

A validação do servidor foi no desempenho de processamento de informação no computador cedido pela empresa (Pentium 4 CPU 2.80GHz e 2GB RAM). Na primeira versão todas as regras estavam em um arquivo de regras. Em cada ciclo de processamento o sistema coletava as variáveis, processava todas as regras, e armazenava todas as tags e anomalias detectadas no banco de dados. Nas últimas versões o processamento é usando metaregras (processamento de regras em seqüência). Com o uso de metaregras o servidor alcançou um bom desempenho no ciclo de processamento.

No banco de dados em cada ciclo de processamento todas as tags eram armazenadas. Com esta forma de armazenamento os dados ocuparam 400MB de espaço em uma semana. A política adotada foi armazenar todas as tags cujo valor atual mude uma porcentagem com respeito ao valor anterior. Com esta nova política o espaço ocupado no disco duro foi reduzido em aproximadamente 10%, 38MB em uma semana.

O nível de conhecimento para instalação e configuração do servidor tem que ser técnico com experiência em redes. O técnico deve configurar o DCOM e instalar os *softwares* Java, Apache, MySQL e PHP. O arquivo de configuração mostrado no Apêndice B deve ser configurado com os parâmetros de rede da empresa onde é instalada. No caso do arquivo de regras tem que ser alterada, sendo necessário neste caso um engenheiro de

conhecimento. O processo de instalação e configuração do sistema foi feito pelo pessoal selecionado da usina hidrelétrica.

### **7.3.3 Cliente**

A característica da interface gráfica foi desenvolvida ao requerimento dos usuários finais do sistema (funcionários da usina hidrelétrica de Balbina). Uma interface simples, amigável e intuitiva com o objetivo que pessoas de um nível médio em conhecimentos de informática possam usar o sistema sem dificuldade. A interface gráfica foi acrescentada desde as primeiras versões até a versão final, passando por varias etapas em conversação com o pessoal da usina.

O gráfico de tendências *online* e históricas inicialmente foi projetado para uma variável, e os usuários finais queriam ter mais de uma variável em um mesmo gráfico que permitisse fazer associações entre os valores das diferentes grandezas. Na versão final o sistema suporta uma interface multi-variável e várias janelas de gráficos simultaneamente.

Nas primeiras versões da árvore hierárquica as variáveis eram mostradas com os *labels* (*i.e.* g4.srg.t.arquente), mas para os operadores era mais fácil identificar as variáveis pelo nome da tag (*i.e.* 426GAQ1\_AI1.PV.VALUE).

Depois de consideradas as apreciações e sugestões dos usuários finais quanto às melhorias da interface gráfica, a versão definitiva do sistema foi instalada na Usina Hidrelétrica de Balbina.

## **8 CONCLUSÕES, CONTRIBUIÇÕES E SUGESTÕES PARA TRABALHOS FUTUROS**

Este capítulo apresenta as contribuições e as conclusões associadas à metodologia proposta para o desenvolvimento de um sistema inteligente de MBC e à implementação computacional do SIMPREBAL. Também são apresentadas sugestões para trabalhos futuros que visam complementar a implementação computacional do SIMPREBAL.

### **8.1 CONCLUSÕES**

As principais vantagens obtidas com o uso de SE na MBC de máquinas são o aumento de confiabilidade e a rapidez no processo de diagnóstico e reparo das falhas.

A forma de representação de conhecimento utilizada favorece o reuso, a expansão e a comunicação das informações aumentando o nível de conhecimento das pessoas envolvidas nas atividades de operação e manutenção.

A Metodologia proposta permite o desenvolvimento independente das diferentes camadas, isto permite que o sistema possa ser implementada por várias equipes;

As regras de produção são fracas e precisam ser acrescentadas;

O SIMPREBAL está instalado, em funcionamento e é usado pelas equipes de operação e manutenção da Usina Hidrelétrica de Balbina.

O uso do SIMPREBAL para auxílio na tomada de decisão é mínimo, sendo usado principalmente para análise de tendência e associação das variáveis *online* e históricas;

A implementação do banco de dados permite armazenar as informações de diagnóstico de anomalias, tomadas de decisão, variáveis de processo, cadastro de usuários e eventos de acesso.

### **8.2 CONTRIBUIÇÕES DO TRABALHO**

A metodologia concebida é baseada no modelo de referência OSA-CBM e utiliza a tecnologia Internet para oferecer um ambiente de desenvolvimento de um sistema inteligente de MBC usando SE e RNA.

Os sistemas desenvolvidos a partir desta metodologia poderão ser aplicados na indústria como auxílio nas tomadas de decisão para o pessoal de operação e manutenção quanto às ações de manutenção.

A rede FAM proposta para a implementação da camada de prognósticos OSA-CBM tem a capacidade para solucionar arbitrariamente problemas complexos de classificação, convergindo rápido à solução (com poucas apresentações da lista de padrões de entrada/saída do conjunto de treinamento), tem a habilidade de reconhecer novos padrões de entrada apresentados e podem operar em modo *online* (novos padrões de entrada podem ser aprendidos pela rede FAM sem re-treinamento dos últimos padrões de entrada/saída).

Outras contribuições estão associadas à implementação do SIMPREBAL utilizando a tecnologia Java *applet* e comunicação via *socket*. Esta implementação permite ao cliente utilizar o sistema independente da plataforma computacional.

### **8.3 IMPLEMENTAÇÃO COMPUTACIONAL**

A implementação computacional foi baseada na metodologia concebida utilizando SE no processamento inteligente. O modelo computacional cliente/servidor permite ter um servidor executando várias funções (multi-tarefa) e mais de um cliente acessando simultaneamente (sistema multi-usuário).

#### **8.3.1 Servidor**

A implementação do servidor *I-Kernel* para desenvolvimento do sistema inteligente de MBC é baseada no modelo OSA-CBM, programado em linguagem Java, disponibilizado as informações geradas via *socket*. Este software, denominado Simprebalserver, é um dos módulos básicos do SIMPREBAL, que processa as informações coletadas da instrumentação via servidor OPC ou banco de dados, através de um SE gerando tomadas de decisão. Dentre as características do Simprebalserver têm-se:

1. Pode processar variáveis de sistemas convencionais 4-20mA, sendo necessário um software *bridge* para converter-lo a sinais OPC;
2. Sistema multi-tarefa, baseado em *threads*, para a aquisição de dados, processamento inteligente e comunicação com os clientes;
3. O acesso a Servidores OPC precisa de configuração DCOM;

4. Robusto, permite o acesso multi-cliente;
5. Processamento de regras de produção usando a ferramenta JESS;
6. Disponibiliza os dados gerados via *socket*;
7. Permite a visualização dos seus dados via *prompt*;
8. Envia OS via email baseados nos diagnósticos e tomadas de decisão.

### 8.3.2 Cliente

O cliente *confmonittool* corresponde à interface com o usuário e é a camada de apresentação OSA-CBM, acessa aos dados disponibilizados pelo servidor por meio de uma porta *socket* via Internet, disponibilizado por um *browser* e *applet* Java. Este software é um dos módulos básicos do SIMPREBAL que apresenta de forma gráfica as informações geradas pelo servidor. Dentre das suas características têm-se:

1. Solicita informações geradas pelo servidor através de comandos predefinidos;
2. Baseado em *applet* Java, possibilitando total compatibilidade com os browsers, bastando ativar a máquina virtual Java;
3. Monitoramento a distância dos estados de saúde dos sistemas, subsistemas e equipamentos, permitindo dispor informações dos diagnósticos e suas tomadas de decisão via internet;
4. Sistema multi-usuário baseado em *threads*;
5. Permite a visualização gráfica de variáveis *online* possibilitando aos usuários analisar as tendências das variáveis em tempo real;
6. Permite a visualização gráfica de variáveis históricas armazenados no banco de dados possibilitando realizar associações das anomalias acontecidas com suas variáveis dependentes;
7. Gera alertas visuais de anomalias dos equipamentos;
8. Permite visualizar os indicadores chave de desempenho (KPI) usando PHP e banco de dados;
9. Fornece um *link* para acesso as análise FMEA via HTML;

## 8.4 SUGESTÕES PARA TRABALHOS FUTUROS

As perspectivas de trabalhos futuros relacionados objetivam a atingir um nível funcional do SIMPREBAL para atender as expectativas de um sistema de manutenção preditiva

integrado. Melhorando a base de conhecimento, as funcionalidades atuais e acrescentando novas interfaces permitirá que o desenvolvimento alcançado seja comparado com sistemas comerciais.

#### **8.4.1 Implementação de algoritmos de prognóstico**

A implementação da camada de prognósticos do modelo de referência OSA-CBM, permitirá prever anomalias. As informações geradas por esta camada ajudarão a tomar uma decisão mais acertada e antecipada ante uma falha. Esta implementação pode ser feita codificando os algoritmos concebidos na metodologia para a rede FAM na aplicação *I-kernel*. As informações de anomalias e as variáveis associadas armazenadas no banco de dados poderão ser usadas no processo de treinamento da rede FAM.

#### **8.4.2 Ampliação da base de conhecimento.**

A base de regras de produção implementadas no SIMPREBAL requer ajuste e manutenção, acrescentar novas regras, mudar as existentes baseadas na qualidade da tomada de decisão e sugestão dos usuários finais. Com a implementação da camada de prognósticos vai ser necessário implementar novas regras na camada tomada de decisão para o processamento de diagnósticos e prognósticos.

#### **8.4.3 Integração dos modelos de referência OSA-CBM e OSA-EAI**

Existe a necessidade de integração das informações geradas pelo SIMPREBAL e por outros sistemas de informação da usina hidrelétrica de Balbina, uma alternativa é usar o modelo OSA-EAI (*Open System Architecture for Enterprise Application Integration*). Esta integração permitirá aos operadores, pessoal de manutenção, administradores da logística, fornecedores de peças e engenheiros, conhecerem as informações das condições dos equipamentos. Centralizar em um computador diferentes tipos de informação como: dados de monitoramento de condição, dados do fabricante, dados de instalação, dados de manutenção, dados operacionais, diagnóstico de saúde dos ativos e dados de confiabilidade.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ABB (2008), Web Site: <http://www.abb.com.br>, Grupo ASEA Brown Boveri.
- Abel, M. (1998), *Sistemas Especialistas*, Instituto de Informática, Universidade Federal do Rio Grande do Sul.
- Aha, D.W. (1997), Special Issue on Lazy Learning, *Artificial Intelligence Review*, Vol. 11, Nro.1-5, pp. 1-6.
- Alkaim, J. L. (2003), *Metodologia Para Incorporar Conhecimento Intensivo às Tarefas de Manutenção Centrada na Confiabilidade Aplicada em Ativos de Sistemas Elétricos*, Dissertação de Mestrado, Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Produção, Florianópolis, Brasil.
- Al-Najjar, B. e Alsayouf, I. (2003), Selecting the most efficient maintenance approach using fuzzy multiple criteria decision making. *International Journal of Production Economics*, Vol. 84, No. 1, pp. 85–100.
- Álvares, A. (2008), Plano Diretor de Automação para Balbina, Relatório Técnico de Pesquisa, UnB, Brasília.
- Amaya, E.J., Álvares, A., Tonaco, R., Souza, R. e Gudwin, R. (2007a), *An Intelligent Kernel for the Maintenance System of a Hydroelectric Power Plant*, 19th International Congress of Mechanical Engineering, COBEM2007, Universidade de Brasília, Brasília, Brasil.
- Amaya, E.J., Álvares, A., Celestino, V. e Silveira, C. (2007b), *Different Control Strategies used in Didactic Plant PD-3 of Smar Through OPC Technology*, 19th International Congress of Mechanical Engineering, COBEM2007, Universidade de Brasília, Brasília, Brasil.
- Amaya, E.J., Álvares, A., Tonaco, R. e Souza, R. (2007c), *Sistema inteligente de manutenção baseada em condição para usina hidrelétrica de Balbina*. In 8º Congresso Iberoamericano de Engenharia Mecânica, CIBIM8, Cusco, Perú.
- Anwar, M.R., Anwar, O., Shamim, S.F. e Zahid, A.A. (2004), *Human Machine Interface Using OPC (OLE for Process Control)*, Engineering, Sciences and Technology, Student Conference On, pp. 35- 40.
- Arcuri Filho, R. (1996), *O futuro conceito de manutenção*, In: XXIV Convención Panamericana de Ingenieros, San José, Costa Rica.



- Ariza, C. F. (1988), *Manutenção preventiva: objetivos, desenvolvimento e aplicação*, *Manutenção e Serviços*, Ano 1, n.5, p. 4-15.
- Armitage, B., Dunlop, G., Hutchinson, D. e Yu, S. (1988), *fieldbus: an emerging communications standard*, *Microprocess Microsyst*, vol.12, No. 10, pp. 555–562.
- Aulete, C. (1986), *Dicionário Contemporâneo da Língua Portuguesa*, 5. Ed, Editora Delta, 5 volumes, p. 1048, Rio de Janeiro, Brasil.
- Bangemann, T., Rebeuf, X., Reboul, D., Schulze, A., Szymanski, J., Thomesse, J.-P., Thron, M. e Zerhouni, N. (2006), *Proteus – creating distributed maintenance systems through an integration platform*. *Computers in Industry*, Vol. 57, No. 6, pp. 539–551.
- Barroso Maia Junior, O. (2003), *Procedimentos de manutenção baseados na técnica de confiabilidade - RCM: Um caso pratico em equipamentos de subestações*. Dissertação de Mestrado, Universidade de Brasília, Programa de Pós-Graduação em Engenharia Elétrica, Brasília.
- Bengtsson, M. (2003), *Standardization issues in condition based maintenance*, In 16th Conference of Condition Monitoring and Diagnostic Engineering Management, Sweden, Växjö: Växjö University Press.
- Bengtsson, M. (2004a), *Condition based maintenance system technology - where is development heading?*, proceedings from the 17<sup>th</sup> Conference of Euromaintenance, Barcelona, Spain, Puntex - Publicaciones.
- Bengtsson, M. (2004b), *Condition Based Maintenance Systems – An Investigation of Technical Constituents and Organizational Aspects*. Licentiate Thesis, Mälardalen University, Eskilstuna, Sweden.
- Bengtsson, M., Olsson, E., Funk, P. e Jackson, M. (2004). *Technical design of condition based maintenance systems - a case study using sound analysis and case-based reasoning*. In proceedings from the 8th Conference of Maintenance and Reliability, Knoxville, T.N., USA.
- Bengtsson, M. (2007), *On Condition Based Maintenance and its Implementation in Industrial Setting*. PhD. Thesis, Mälardalen University, Eskilstuna, Sweden.
- Bevilacqua, M., Braglia, M., e Montanari, R. (2003), *The classification and regression tree approach to pump failure rate analysis*. *Reliability Engineering and System Safety*, Vol. 79, No. 1, pp. 59–67.
- Bittencourt, G. (2001), *Inteligência artificial: ferramentas e teorias*, 2ed. 362 p, UFSC Florianópolis, Brasil.

- Bharadwaj, M. (2003), *Semi-Supervised Learning in Exemplar Based Neural Networks*, master thesis, department of Electrical and Computer Engineering in the College of Engineering at the University of Central Florida, Orlando, Florida.
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad P e Stal, M. (1996), *Pattern-Oriented Software Architecture - A System of Patterns*, volume 1, John Wiley and Sons, 605 Third Avenue, New York, NY 10158-0012, USA.
- Booch, G., Jacobson, I. e Rumbaugh, J. (1999), *The Unified Modeling Language User Guide*, Addison-Wesley.
- Box, D. (2000), *A Young Person's guide to the simple object access protocol: SOAP increases interoperability across platforms and languages*. MSDN Magazine (março).
- Braga, A., Ludemir, T. e Carvalho, A. (2000), *Redes neurais artificiais: Teoria e aplicações*, Livros Técnicos e Científicos, Rio de Janeiro, Brasil.
- Britto, T. (2006), *Metodologia da Manutenção Centrada em Confiabilidade Aplicada a Para-Raios de Alta Tensão*, Dissertação de Mestrado, Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Elétrica, Florianópolis, Brasil.
- Butcher, S. W. (2000), *Assessment of condition-based maintenance in the department of defense*. USA, McLean, VA, Logistics Management Institute, [http://www.acq.osd.mil/log/mrmp/senior\\_steering/condition/LMI%20CBM%20Report.pdf](http://www.acq.osd.mil/log/mrmp/senior_steering/condition/LMI%20CBM%20Report.pdf).
- Byington, C.S., Roemer, M.J. e Galie, T. (2002), *Prognostic Enhancements to Diagnostic Systems for Improved Condition-Based Maintenance*, IEEE Aerospace Conference Proceedings.
- Caletti, L. (2003), *Desenvolvimento de um Protótipo de Sistema Especialista para Projeto de Unidades de Potência Hidráulica*, Dissertação de Mestrado, Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica, Florianópolis, Brasil.
- Campos, F. M. (2007), *Quais são as Reais Características da Qualidade da NBR ISO/IEC 9126-1?*, <http://www.testexpert.com.br/>.
- Carpenter, G.A. e Grossberg, S. (1987), *A massively parallel architecture for a self-organizing neural pattern recognition machine*, Computer Vision, Graphics, and Image Processing, Vol. 37, pp. 54-115.

- Carpenter, G.A., Grossberg, S., e Rosen, D.B. (1991), Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, *Neural Networks*, Vol. 4, pp. 759-771.
- Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., and Rosen, D.B. (1992), Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, pp. 698-713.
- Chappel, D. (2000), *Understanding Microsoft Windows 2000 Distributed Services*, Microsoft Press.
- Chen, C. H. (1996), *Fuzzy logic and neural network handbook*, McGraw-Hill. New York.
- Chinnam, R. B. e Baruah, P. (2004), *A neuro-fuzzy approach for estimating mean residual life in condition-based maintenance systems*. *International Journal of Materials and Product Technology*, Vol. 20, No. 1–3, pp.166–179.
- Chrissanthi, A. (2008), Online expert systems for fault diagnosis in technical processes, *The Journal of Knowledge Engineering*, Vol. 25, No. 2, pp. 115-132.
- Ciarapica, F.E. e Giacchetta, G. (2006), Managing the condition-based maintenance of a combined-cycle power plant: An approach using soft computing techniques, *Journal of Loss Prevention in the Process Industries*, Vol. 19, No. 4, pp. 316-325.
- Cunha, F. (1995), *Um sistema especialista para previdência privada*, Dissertação de Mestrado, Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Produção, Florianópolis, Brasil.
- Dedo, D. e Nelson, G. (1997), *Integrate the Enterprise*, MSDN Library, Microsoft Corporation.
- Djurdjanovica, D., Lee, J. e Ni, J. (2003), *Watchdog agent - an infotronics-based prognostics approach for product performance degradation assessment and prediction*. *Advanced Engineering Informatics*, Vol. 17, No. 3–4, pp. 109–125.
- Duarte, C., Figueiredo, L. e Corrêa, M. (2006), *Utilização do Matlab no Ensino da Tecnologia OPC Aplicada a Controle De Processos*, XVI Congresso Brasileiro de Automática, CBA2006, Salvador, Bahia, Brasil.
- Dunn, S. (1997), *Maintenance terminology – some key terms*, <http://www.maintenanceresources.com/ReferencesLibrary/MaintenanceManagement/KeyTerms.htm>.

- Dupont, C. J. (2003), *Integração de Análises de Defeitos e Definição de um grau de Risco Global para Transformadores de Potência*, Tese de Doutorado em Engenharia Elétrica – COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.
- Durkin, J. (1994), *Expert Systems Design and Development*. Ed. Prentice Hall.
- Dunsdon, J., e Harrington, M. (2008), The Application of Open System Architecture for Condition Based Maintenance to Complete IVHM, Aerospace Conference IEEE , pp.1-9.
- Engel, S.J., Gilmartin, B.J., Bongort, K. e Hess, A. (2000), Prognostics, the real issues involved with predicting Life Remaining”, IEEE Aerospace Conference Proceedings, Vol. 6, No. 2000, pp. 457–469.
- Falqueto, J. e Telles, M. S. (2007), Automation of diagnosis of electric power transformers in Itaipu Hydroelectric Plant with a Fuzzy Expert System, Emerging Technologies and Factory Automation IEEE Conference on, pp. 577-584.
- Fieldbuses (2008), Web Site <http://www.interfacebus.com/>.
- Fonseca, M. O. (2002), *Comunicação OPC, uma abordagem prática*. In VI Seminário de Automação de Processos, Associação Brasileira de Metalurgia e Materiais, Vitória, ES.
- Fraden, J. (2003), *Handbook of Modern Sensors: physics, designs, and applications*, Spring Science Business Media Inc., 233 Spring st, New York, NY 10013, USA.
- Frelicot, C. (1996), A fuzzy-based prognostic adaptive system, RAIRO-APII-JESA, Journal Europeen des Systemes Automatises, Vol. 30, No. 2-3, pp. 281-299.
- Friedman-Hill, E (2003), *Jess in Action: Rule-Based Systems in Java*. 1ra. Ed, Manning Editor, Greenwich, CT.
- Fu, C., Ye, L., Liu, Y., Yu, R., Iung, B., Cheng, Y. e Zeng, Y. (2004), Predictive Maintenance in Intelligent-Control-Maintenance-Management System for Hydroelectric Generating Unit, *IEEE Transactions on Energy Conversion*, Vol. 19, No. 1, pp. 179-186.
- Garcia, M. C., Sanz-Bobia, M. A. e Picob, J. (2006), SIMAP: Intelligent System for Predictive Maintenance Application to the health condition monitoring of a windturbine gearbox, Journal Computers in Industry. E-maintenance Special Issue, Vol. 57, No. 6, pp. 552-568.
- Giarratano, J. e Riley, G. (1994), *Expert systems: principles and programming*, 2ed., PWS Publishing Company, Boston.

- Gonzalez, A. e Dankel, D. (1993), *The Engineering of knowledge-based systems*, Prentice Hall. New Jersey.
- Granell, V. (2007), *Levels of automation in manufacturing systems: Aligning strategic and tactical decisions by means of operational measurements*, Licentiate Thesis, Chalmers University of Technology, Göteborg, Sweden.
- Groer, P.G. (2000), Analysis of Time-to-Failure with a Weibull Model, *Proceedings of the Maintenance and Reliability Conference*, Knoxville, TN.
- Gudwin, R. (2006), *Especificação do Sistema - Sistema I-Kernel: Um Kernel Inteligente para o SIMPREBAL - Sistema de Manutenção Preditiva de Balbina*, Relatório Técnico.
- Guo, J., Li, Z., Chen, Y., Wang, Y. e Cheng, S. (2002), *Virtual environment conceptions for CBM of hydro-electric generating units*, International Conference on Power System Technology, pages 1957–1961, Kunming, China.
- Guo, J., Liu, Y., Xiao, Z., Chen, Q. e Xu, X. (2007). B/C/S Based Remote Condition Monitoring and Diagnostic Support System of Generating Unit, *Wireless Communications, Networking and Mobile Computing*, International Conference on, pp.6262-6265.
- Haykin, S. (2001), *Redes Neurais: Princípios e prática*, 2.ed., Bookman, Porto Alegre, RS, Brasil.
- Hüsemann, R. e Pereira, C. E. (2007), *A multi-protocol real-time monitoring and validation system for distributed fieldbus - based automation applications*. Control Engineering Practice, Vol. 15, No. 8, pp. 955–968.
- IDEF0, (1993), *Integration Definition for Functional Modeling (IDEF0)*, In: Federal Information Processing Standards Publication 18.
- ISO13374-2 (2007), *Condition Monitoring and Diagnostics of Machines - Data Processing Communication and Presentation - Part 2: Data processing*, International Organization for Standardization.
- Jackson, P. (1999), *Introduction to expert systems*, Addison-Wesley, Harlow, England.
- Jacobson, I., Booch, G. e Rumbaugh, J. (1999), *The Unified Software Development Process*, Addison Wesley.
- Jardine, A.K.S., Lin, D. e Banjevic, D. (2006), *A review of machinery diagnostics and prognostics implementing condition-based maintenance*. Mechanical Systems and Signal Processing, Vol. 20, No. 7, pp.1483–1510.

- Javadpour, R. e Knapp, G. M. (2003), A fuzzy neural network approach to machine condition monitoring, *Computers and Industrial Engineering*, Vol. 45, No. 2, pp. 323–330.
- Kardec, P. e Nascif, J. (1999), *Manutenção – Função estratégica*. Editora Qualitymark, Rio de Janeiro, RJ, Brasil.
- Kern, V. M. (1999), Modelagem da Informação com IDEF1X: Linguagem, Método, Princípio do Consenso. *Revista Alcance*, ano VI, No. 3, pp. 99-107. Itajaí: Editora da UNIVALI.
- Kim, C. and Weston, R. H. and Hodgson, A. and Lee, K. (2002), *The complementary use of IDEF and UML modelling approaches*, In: *Computers in Industry* 50, pp.35 – 56.
- Koskinen, K (2005), *Manufacturing Systems – vertical integration and use of information in operations and maintenance*, Tik-86.5141 Enterprise Systems Architecture.
- Kothamasua, R. e Huang, S. H. (2007), Adaptive Mamdani fuzzy model for condition-based maintenance, *Fuzzy Sets and Systems*, Vol. 158, No. 24, pp. 2715–2733.
- Kovács, Z. L. (2002), *Redes neurais artificiais: Fundamentos e aplicações*, 3.ed., Livraria da Física, São Paulo, Brasil.
- Kubiak, J., García-Gutiérrez, A. e Urquiza, G. (2002), The diagnosis of turbine component degradation: case histories, *Applied Thermal Engineering*, Vol. 22, No. 17, pp. 1955-1963.
- Lacerda, J. e Júnior, P. (1997), *A Informatização Integrada da Manutenção. Seus Desdobramentos e os Sistemas Especialistas*. ABRAMAN - 12º, Congresso Brasileiro de Manutenção, São Paulo, SP, Brasil.
- Lebold, M. e Thurston, M. (2001), *Open standards for condition-based maintenance and prognostics systems*. In *Proceedings from the 5<sup>th</sup> Conference of Maintenance and Reliability*, Knoxville, T.N., USA.
- Lebold, M., Reichard, K. e Boylan D. (2003), *Utilizing DCOM in an Open System Architecture Framework for Machinery Monitoring and Diagnostics*, In *Proceedings from Aerospace Conference*, pages 1227–1236, Irvine, USA.
- Lee, J., Ni, J., Djurdjanovic, D., Qiu, H. e Liao, H. (2006), Intelligent prognostics tools and e-maintenance, *Computers in Industry*, Vol. 57, No. 6, pp. 476–489.
- Legat, V., Zaludova, A. H., Cervenka, V. e Jurca, V. (1996), *Contribution to optimization of preventive replacement*. *Reliability Engineering and System Safety*, Vol. 51, No. 3, pp. 259–266.

- Lewis, F.L. (1986), *Optimal Estimation: With an Introduction to Stochastic Control Theory*, John Wiley and Sons, New York.
- Liebowitz, J. (1999), *The Handbook of Applied Expert Systems*, CRC Press, 868p.
- Lucas, P. J. F. e Van der Gaag, L. C. (1991), *Principles of expert systems*, Addison-Wesley, Workingham, England.
- Lucatelli, M. V. (1998), *Estudo de Procedimentos de Manutenção Preventiva de Equipamentos Eletromédicos*, Dissertação de Mestrado em Engenharia Elétrica, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.
- Lucatelli, M. V. (2002), *Proposta de Aplicação da Manutenção Centrada em Confiabilidade em Equipamentos Médico-Hospitalares*, Tese de Doutorado em Engenharia Elétrica, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.
- Mahalik, N.P. (2003), *Fieldbus Technology: Industrial Network Standards for Real-time Distributed Control*, Springer publication, ISBN:3540401830.
- Mahalik, N.P. e Yen, M.(2008), *Extending fieldbus standards to food processing and packaging industry: A review*, Computer Standards & Interfaces.
- Ljung, L. (1999), *System Identification: Theory for the User*, Prentice-Hall, New Jersey, 2nd ed.
- Lowy, J. (2001), *Web Services- Hurdle the Firewall*, Net Magazine.
- Lucifredi, A., Mazzieri, C. e Rossi, M. (2000), application of multiregressive linear models, dynamic kriging models and neural network models to predictive maintenance of hydroelectric power systems, *Mechanical Systems and Signal Processing*, Vol. 14, No. 3, pp. 471–494.
- Mecabô, L. (2007), *Desenvolvimento de um protótipo de sistema especialista para apoio à manutenção de turbocompressores centrífugos de gás natural*. Dissertação de Mestrado, Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica, Florianópolis.
- MIMOSA e OpenO&M (2004), *Executive Overview*, August.
- MIMOSA (2008), *Web Site: <http://www.mimosa.org/>*, Machinery Information Management Open Systems Alliance.
- Mitra, S., De, R.K. e Pal, S.K. (1997), Knowledge-based fuzzy MLP for classification and rule generation, *IEEE Trans. Neural Networks*, Vol. 8, pp. 1338–1350.

- Molina, J. M., Isasi, P., Berlanga, A. e Sanchis, A. (2000), Hydroelectric power plant management relying on neural networks and expert system integration, *Engineering Applications of Artificial Intelligence*, Vol. 13, No. 3, pp. 357-369.
- Moubray, J. (1997), *Reliability-Centered Maintenance*. 2ed., Industrial Press Inc., Woodbine, NJ.
- Moya, C. C. e Vera, J. C. H. (2003), *Evaluation of condition based maintenance through activity based cost*. Maintenance Journal, Vol. 16, No. 3, pp. 54–61.
- MSDN Library (1996), *DCOM Technical Overview*, Microsoft Corporation.
- MSDN Library (1998), *Distributed Component Object Model Protocol - DCOM/1.0*,
- MyQ (2000), *Software de Manutenção : Comprar ou Desenvolver ?*, Revista: Nova Manutenção y Qualidade, ano 7, No. 28, pp. 34–35, ISSN1413-4659, Rio de Janeiro, RJ, Brasil.
- NBR-5460 (1992), *Eletrotécnica e eletrônica, Sistemas elétricos de potência, Terminologia*, Associação Brasileira de Normas Técnicas, Rio de Janeiro, Brasil.
- NBR-5462 (1994), *Confiabilidade e Mantabilidade, Terminologia*, Associação Brasileira de Normas Técnicas, Rio de Janeiro, Brasil.
- Nunes, E. L. (2001), *Manutenção centrada em confiabilidade (MCC): análise da implantação em uma sistemática de manutenção preventiva consolidada*. Dissertação de Mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção, Florianópolis.
- NSF (2008), *Web Site: <http://www.imscenter.net>*, Industry/University Cooperative Research Center for Intelligent Maintenance Systems.
- Oliveira, C. H. P. (2002), *SQL Curso prático*, ed. Novatec, 272p, São Paulo, Brasil.
- OMG (2008), *CORBA/IIOP Specification*, <[http://www.omg.org/technology/documents/formal/corba\\_iiop.htm](http://www.omg.org/technology/documents/formal/corba_iiop.htm)>, Object Management Group.
- OPC Foundation (1998), *OPC Overview, Version 1.0*, outubro 1.
- OSA-CBM (2006), *Standard v3.1*, Open System Architecture for Condition Based Monitoring.
- Pinto, A. M. (2003), *Análise da manutenção de unidades geradoras de hidrelétricas no atual cenário do setor elétrico brasileiro*, Dissertação de Mestrado, Universidade de Brasília, Programa de Pós-Graduação em Engenharia Elétrica, Brasília.
- Pressman, R. S. (1995), *Engenharia de Software*, Pearson Education, São Paulo, Brasil.
- Promise (2008), *PROduct lifecycle Management and Information tracking using Smart Embedded systems*, web site: <http://www.promise.no/>.



- Reis, D. e Pati, N. (2000), Applications of Artificial Intelligence to Condition-Based Maintenance, RAE - Revista de Administração de Empresas, Vol. 40, No. 2, pp. 102-107, São Paulo, Brasil.
- Rigoni, E., Pepplow, L. A. e Silveira, P. R. (2004), *Sistema especialista de apoio à confiabilidade e a manutenção de sistemas técnicos automatizados*, In EMC 6610 Pós-Mec, Florianópolis.
- Roemer, M., Byington, C., Kacprzyński, G. e Vachtsevanos, G. (2005). An Overview of Selected Prognostic Technologies with Reference to an Integrated PHM Architecture. *Proceedings of the First International Forum on Integrated System Health Engineering and Management in Aerospace*.
- Russell, S. e Norvig, P. (2003), *Artificial intelligence: A modern approach*, Prentice Hall.
- Schneider, G. e Winters, J. P. (1998), *Applying Use Cases: a practical guide*, Addison Wesley.
- Sharda, R. (1994), Neural network for the MS/OR analyst: An application bibliography, *Interfaces*, Vol. 24, No. 2, pp. 116-130.
- Shimanuki, Y. (1999), OLE for process control (OPC) for new industrial automation systems, In: *Systems, Man, and Cybernetics*, Vol. 6, pp. 1048-1050 vol.6, Tokyo, Japan.
- Silva, J. C. (1998), *Expert System Prototype for Hydraulic System Design Focusing on Concurrent Engineering Aspects*, Tese de Doutorado em Engenharia Mecânica, UFSC, Florianópolis, Brasil.
- Sim, S. H. e Endrenyi, J. (1988), *Optimal preventive maintenance with repair*. IEEE Transactions on Reliability, Vol. 37, No. 1, pp. 92–96.
- Simpson, P.K. (1992), Fuzzy min–max neural networks—Part 1: Classification, IEEE Trans. Neural Networks, Vol. 3, pp. 776–786.
- Smar (2008), *Web Site: <http://www.smar.com.br>*.
- Smar (2005), *Foundation Fieldbus - Function Blocks*, instruction Manual.
- Sommerville, I. (2007), "Engenharia de Software", 8a. edição, Addison-Wesley/Pearson
- Souza, L. C. A., Filho, C. S. e Pena, R. T. (1998), *Padrão de Acesso a Dados OPC e sua Implementação em um Driver OPC-MODBUS*, In: V Simpósio Regional de Instrumentação/ II Congresso Mineiro de Automação, ISA /GRINST – IBP, Belo Horizonte, p. 157-164.
- SS-EN-13306 (2001), *Maintenance terminology*, Swedish Standard SS-EN 13306, European Standard EN 13306.

- Starr, A. G. (1997), *A structured approach to the selection of condition based maintenance*. In proceedings from the 5th International Conference on FACTORY 2000, UK, Cambridge.
- Studer, L. e Masulli, F. (1996), *On the structure of a neuro-fuzzy system to forecast chaotic time series*, International Symposium on Neuro-Fuzzy Systems, pp. 103 – 110.
- Sun Microsystems (2008), *Enterprise JavaBeans™ Specification*, <<http://java.sun.com/products/ejb/docs.html>>.
- Swearingen, K., Majkowski, W., Bruggeman, B., Gilbertson, D., Dunsdon, J. e Sykes, B. (2007), *An Open System Architecture for Condition Based Maintenance Overview*, Aerospace Conference Proceedings, pp. 1–8.
- Szymanski, J., Bangemann, T., Thron, M., Thomesse, J.-P., Reboeuf, X., Lang, C. e Garcia, E. (2003), *PROTEUS - a European initiative for e-maintenance platform development*, Emerging Technologies and Factory Automation Proceedings, IEEE Conference, Vol. 2, pp. 415-420.
- Tavares, L. e Filho, A. (2002), *A Manutenção Como Uma Atividade Corporativa*, Disponível na Internet <<http://www.abraman.org.br/publicações>>.
- Tatem (2008), *Technologies And Techniques for nEw Maintenance concepts*, web site: <http://www.tatemproject.com/>.
- The Open Group (2008), *DCE 1.1: Remote Procedure Call: CAE Specification*, <http://www.opengroup.org/onlinepubs/9629399>.
- Thurston, M. G. (2001). *An Open Standard for Web-Based Condition-Based Maintenance Systems*. In proceedings from the IEEE System Readiness Technology Conference, Autotestcon Proceedings, pages 401–415, Valley Forge, P.A., USA.
- Tontini, G. e de Queiroz, A.A. (1996), *RBF fuzzy-ARTMAP: A new fuzzy neural network for robust on-line learning and identification of patterns*, In Proc. IEEE Int. Conf. Syst., Man, Cybern., Vol. 2, pp. 1364–1369.
- Tsang, A. (1995), *Condition-based maintenance: Tools and decision making*. Journal of Quality in Maintenance Engineering, Vol. 1, No. 3, pp.3–17.
- Vachtsevanos, G. e Wang, P. (2001), *Fault prognosis using dynamic wavelet neural networks*. In proceedings from the IEEE System Readiness Technology Conference, Autotestcon Proceedings, pages 857–870, Valley Forge, PA, USA.
- Vaurio, J. K. (1997), *On time-dependent availability and maintenance optimization of standby units under various maintenance policies*, Reliability Engineering and System Safety, Vol. 56, No. 1, pp. 79–89.

- Vieira, R. C. e Roisenberg, M. (2003), *Redes Neurais Artificiais: um breve tutorial*, Laboratório de Conexionismo e Ciências Cognitivas, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil.
- Vinade, C. (2003), *Sistematização Do Processo De Projeto Para Confiabilidade E Manutenibilidade Aplicado A Sistemas Hidráulicos E Implementação De Um Sistema Especialista*, Tese de Doutorado, Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica, Florianópolis.
- Vuorimaa, P., Jukarainen, T. e Karpanoja, E. (1995), A neuro-fuzzy system for chemical agent detection, *IEEE Trans. Fuzzy System*, Vol. 3, pp. 415–424.
- Walter, R. (2006), *Open Systems Architecture for Condition-based Maintenance (OSA-CBM) Primer*, August.
- Waterman, Donald A. (1986), *A guide to expert systems*, Addison-Wesley, p. 419, ISBN 0201083132.
- Williamson, J.R. (1996), Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps, *Neural Networks*, Vol. 9, No. 5, pp. 881–897.
- Yam, R., Tse, P., Li, L. e Tu, P. (2001), *Intelligent predictive decision support system for condition-based maintenance*, *Journal of Advanced Manufacturing Technology*, Vol. 17, No. 5, pp. 383–391.
- You, H. S. (1998), Real-time monitoring and detecting of after-burning hazards of continuous catalyst regenerators, *Journal of Loss Prevention in the Process Industries*, Vol. 11, No. 1, pp. 25–41.
- Zadeh, L.A. (1965), Fuzzy Sets. *Information and Control*, Vol. 8, pp. 338-353.
- Zadeh, L. A. (1973), Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, *IEEE Transactions on System, Man and Cybernetics*, Vol. 3, pp. 28-44.
- Zadeh, L.A. (1996), Fuzzy Logic = Computing with Words, *IEEE Transactions on Fuzzy Systems*, Vol. 4, No. 2, pp. 103-111.
- Zheng, L. e Nakagawa, H. (2002), *OPC (Ole for Process Control) specification and its developments*, In Technical report, 41st SICE Annual Conference.

## **APÊNDICES**

# APÊNDICE A – ABORDAGENS DE ALGORITMOS DE PROGNÓSTICOS

Neste apêndice são apresentados os métodos e algoritmos usados na camada de prognóstico do modelo de referência OSA-CBM. Em cada um dos algoritmos descrevem-se as técnicas de previsão usadas e os dados requeridos.

## A.1 INTRODUÇÃO

O Prognóstico de falhas é abordado por várias técnicas, desde estimacões Bayesianas e outros métodos estatísticos e probabilísticos até ferramentas de inteligência artificial. Estas tecnologias incluem o filtro *Kalman* adaptativo multi-etapas (Lewis, 1986), modelos auto-regressivos de media móvel (Lewis, 1986), modelos *Weibull* (Groer, 2000), previsão por modelos e *cluster* de busca (Frelicot, 1996), e métodos de estimacão de parâmetros (Ljung 1999). No domínio da inteligência artificial, raciocínio baseado em casos (Aha, 1997), modelos inteligentes baseados em decisão e grafos *min-max*, foram considerados como métodos potenciais para algoritmos de prognósticos. Outros métodos como redes de *Petri*, redes neurais, sistemas de lógica nebulosa (*fuzzy*) e sistemas híbridos como *neuro-fuzzy* (Studer e Masulli, 1996) encontram ampla utilidade como ferramentas de prognóstico.

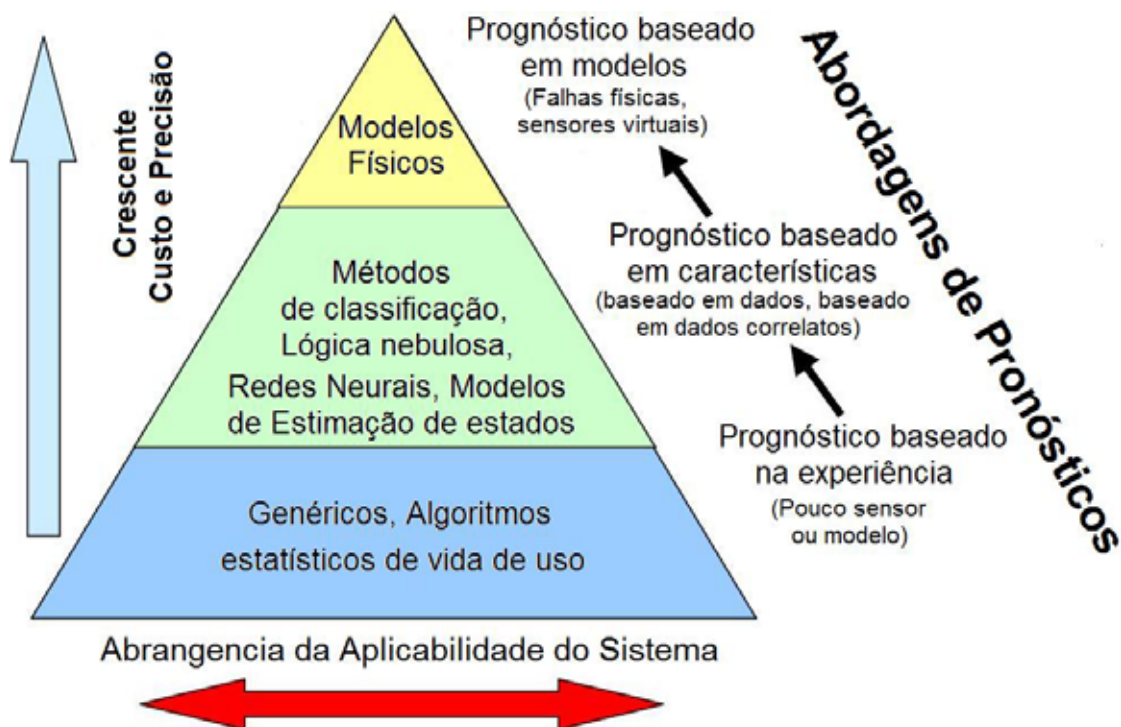


Figura A.1-Abordagens de técnicas de prognóstico (Byington *et al.* 2002, modificado).

Na Figura A.1 apresenta-se o resumo das possíveis abordagens de prognósticos em função à aplicabilidade a vários sistemas e o custo relativo de implementação. As tecnologias de prognóstico normalmente usam características medidas ou inferidas, também os modelos baseado em dados e/ou baseado na física, para prever a condição de um sistema em algum tempo futuro. Com incertezas associadas, os prognósticos podem ser aplicados para administrar modos de falha por condição do material ou por perda de funcionalidade. Os algoritmos de prognóstico podem ser de desenho genérico, mas específico em termos de aplicação.

Segundo Roemer *et al.* (2005), os desenvolvedores de sistemas de prognóstico podem implementar várias abordagens e associar bibliotecas de algoritmos para aplicações personalizadas desde os modelos simples de uso de histórico até as abordagens avançadas que usam análise de características ou modelos físicos de falha. Dependendo do nível crítico do sistema monitorado, vários níveis de dados, modelos e informações históricas serão necessários para o desenvolvimento e implementação da abordagem de prognóstico desejada. Os possíveis algoritmos para a implementação de um sistema de prognósticos podem ser visualizados na Figura A.2.

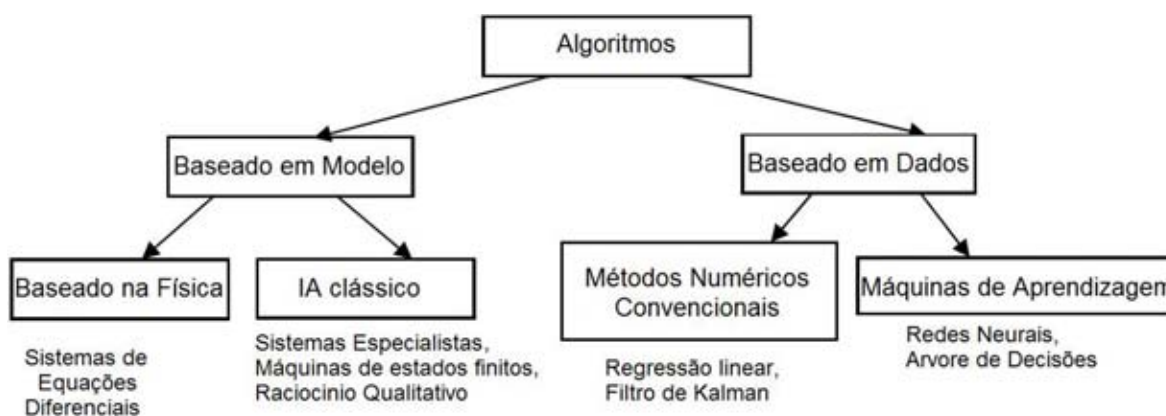


Figura A.2- Classificação dos algoritmos de prognósticos

Na Tabela A.1, Roemer *et al.* (2005), mostram uma visão dos modelos recomendados e informações necessárias para a implementação das abordagens específicas. Os três níveis de algoritmos vão desde os métodos baseado na experiência (confiabilidade) até as abordagens de falhas baseado na física que usam dados dos sensores.

Tabela A.1- Modelos e informação para implementação de sistemas de prognósticos  
(Roemer *et al.* 2005, modificado).

	Precisão de prognóstico →		
	Baseado na experiência	Baseado em características	Baseado na física
Engenharia do modelo	Não Requerido	Útil	Requerido
Histórico de falhas	Requerido	Não Requerido	Útil
Condições de operação passadas	Útil	Não Requerido	Requerido
Condições atuais	Útil	Requerido	Requerido
Identificação de padrões de falha	Não Requerido	Requerido	Requerido
Histórico de manutenção	Útil	Não Requerido	Útil
Em geral	Não Sensor/Não Modelo	Sensor/Não Modelo	Sensor e Modelos

## A.2 PROGNÓSTICO BASEADO NA EXPERIÊNCIA

Neste caso precisa-se de um modelo físico de um sistema ou componente e não existem sensores para monitorar a condição, por tanto o prognóstico baseado na experiência é a única alternativa. Este modelo de prognóstico só precisa de experiências de falhas ou recomendações de componentes em operações similares (Byington *et al.*, 2002).

## A.3 PROGNÓSTICO BASEADO EM CARACTERÍSTICAS

A abordagem do prognóstico baseado em características apóia-se na habilidade de seguir os sinais e desvios de tendências, e associar taxas de cambio dessas variações de características específicas e medidas, da sua condição normal de operação. Esta técnica mostrada na Figura A.3, pode ser implementada em sistemas com experiência de falhas condicionais ou baixa degradação de algum tipo de falha, como perda de eficiência numa unidade geradora hidráulica. Geralmente os prognósticos baseados em tendências trabalham bem em sistemas com algum nível de degradação, porque a perda de condição é tipicamente o resultado da interação de múltiplos componentes. Esta abordagem requer suficiente informação dos sensores disponíveis para estimar a condição atual do sistema e o nível relativo de incerteza nas medições. Os modelos físicos ou estatísticos são úteis e ajudam a classificar uma falha específica, mas não é um requerimento para esta

abordagem. Uma alternativa ao modelo físico é implementar um SE de condições de falha (Byington *et al.*, 2002). A base de conhecimento do SE deverá ter informações das características do sistema, processará os dados dos sensores e na saída teremos uma classificação das possíveis falhas.

Segundo Roemer *et al.* (2005), o prognóstico baseado em características pode ser implementado em sistemas de geração elétrica, baseado nas mudanças de várias características mensuráveis como temperatura, corrente, e voltagem de vários pontos do sistema. As características de uma unidade geradora hidráulica como: potência gerada, temperatura dos trocadores de calor, temperatura de óleo dos mancais, etc., combinado com falhas conhecidas podem ser extraídas dos dados dos sensores. Obtidas estas características podem ser comparadas com a vida útil restante estimada para prover evidências que confirmem as condições degradadas de falha.

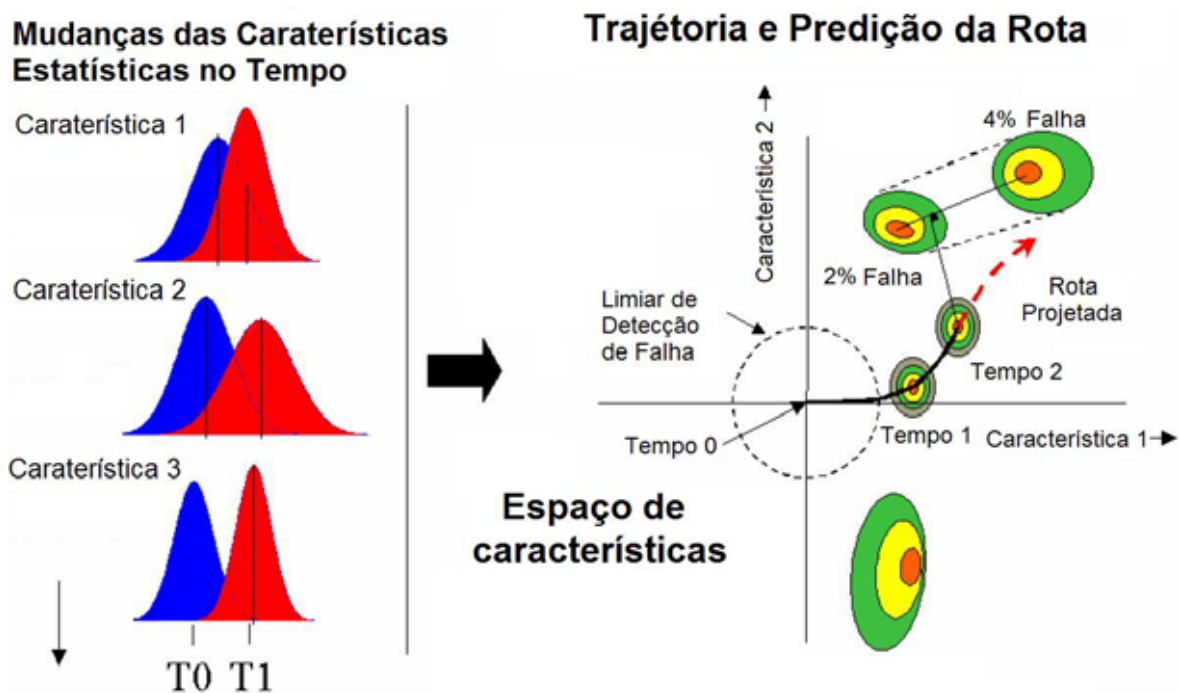


Figura A.3- Prognóstico Baseado em Características (Roemer *et al.* 2005, modificado).

#### A.4 PROGNÓSTICO BASEADO EM DADOS

Em muitos casos é difícil ou impraticável determinar o modelo baseado na física para propósitos de prognósticos. Se temos os dados históricos de defeitos e falhas em termos de gráficos no domínio do tempo de vários sinais que levaram à falha, ou conjunto de dados estatísticos. Nestes casos, uma forma é usar um aproximador não linear que pode ser



ajustado usando algoritmos formais estabelecidos e gerar uma saída desejada diretamente em termos dos dados. Esta técnica inclui as redes neurais, os quais são baseados em técnicas de processamento de sinal em sistemas nervosos biológicos, e sistema de lógica nebulosa, os quais são baseados em lingüística e habilidades de raciocínio humano (Roemer *et al.*, 2005).

Em prognósticos, as RNA, os sistemas de lógica nebulosa e outros métodos de inteligência computacional são ferramentas alternativas para pesquisadores na área de prognósticos (Sharda, 1994). A diferença dos métodos tradicionais baseados em modelo, as RNA são baseados em dados, auto-adaptativos, podem fazer suposições acerca do modelo para o problema em estudo, aprendem de exemplos e fazem relações entre os dados, sendo apropriadas para problemas onde é fácil ter os dados de domínio do conhecimento do sistema em estudo.

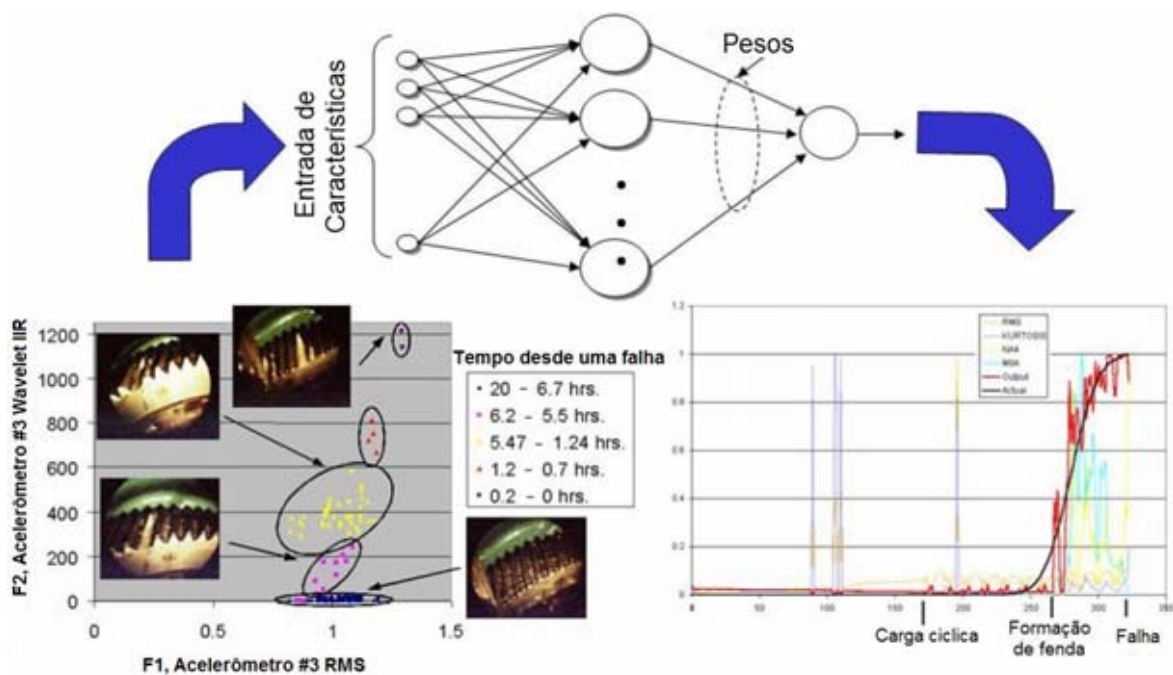


Figura A.4- Prognóstico Baseado em Dados (Roemer *et al.* 2005, modificado).

O problema principal das RNA é que o raciocínio das suas decisões não é evidente. Apesar disso eles provêm ferramentas viáveis para problemas práticos de predição. Entendendo como um determinado defeito ou falha é relacionado com uma característica mensurável ou inferido do sistema monitorado, a abordagem do modelo baseado em dados é comumente utilizada. Baseado nas características de entrada selecionadas associado com a progressão de falhas, a saída desejada de predição do tempo de falha é obtido baseado em

um processo de treinamento onde a rede automaticamente ajustara os pesos e limiar baseados nas relações entre o tempo de falha e as magnitudes das características correlatas. Na Figura A.4 mostra-se um exemplo de uma rede neural treinada com os dados do processo para prever o tempo de falha de um equipamento. Depois de treinada a rede neural, esta arquitetura pode ser usadas para prever progressões das mesmas características para diferentes ensaios em condições de operação similar.

### A.5 PROGNÓSTICO BASEADO EM MODELOS FÍSICOS

Os modelos baseado na física provêm os meios para calcular o dano dos componentes críticos como função das condições operacionais e avaliar os efeitos acumulados em termos de tempo de uso do componente. Integrando técnicas físicas e estocásticas o modelo pode ser usado para avaliar a distribuição do tempo de uso restante do componente como função de incertezas nas propriedades dos componentes. As representações históricas dos perfis de operação servem como base para o cálculo da acumulação de danos futuros. Este modelo pode ser usado para prognóstico de falhas em tempo real com um limite específico de confiabilidade (Byington *et al.*, 2002). O diagrama de blocos desta abordagem é mostrado na Figura A.5.

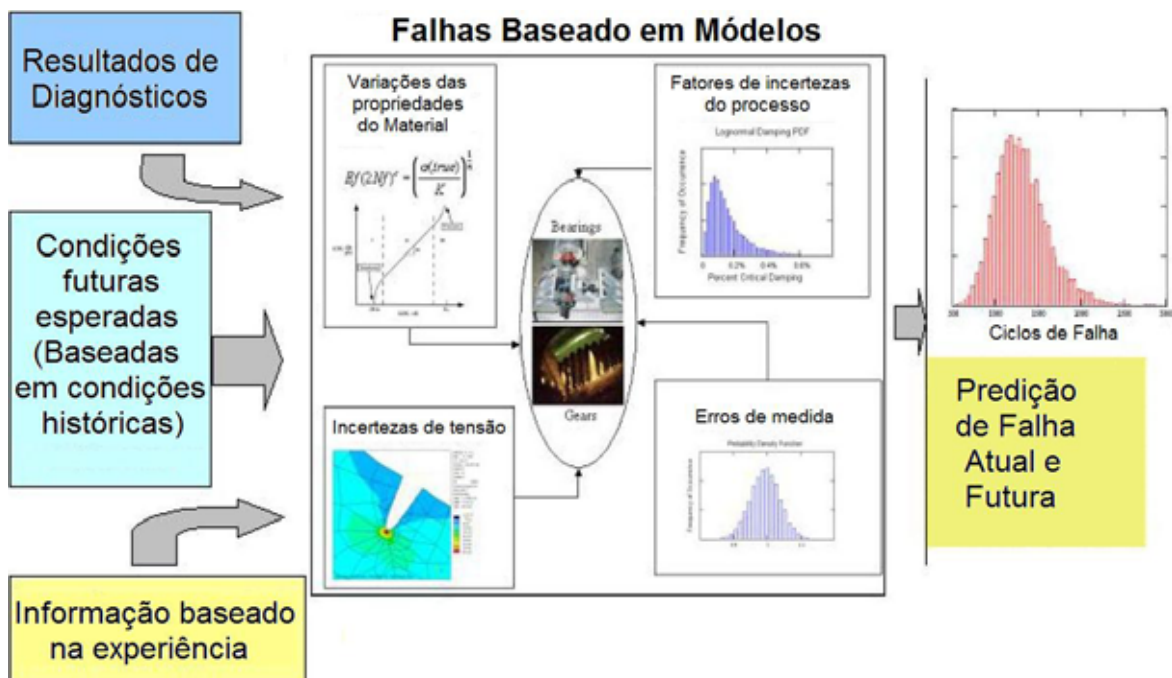


Figura A.5- Prognóstico Baseado em Modelos Físicos (Roemer *et al.* 2005, modificado).

Segundo Roemer *et al.* (2005), a abordagem do modelo baseado em modelos físicos difere da abordagem baseada em características em que eles podem estimar o tempo útil restante em ausência de qualquer instrumento real, mas os sinais do instrumento podem ser simulados através de modelos ou instrumentos virtuais. A combinação ou fusão das abordagens baseadas em modelos e baseadas em características provêm uma completa habilidade de prognóstico da vida inteira do componente.

## A.6 PROGNÓSTICO ADAPTATIVO

Segundo Roemer *et al.* (2005), a atualização do prognóstico, baseado em informações adicionais de estados de percepção (detecção de falhas e diagnósticos) que podem estar disponíveis é desejável. O conceito de prognóstico adaptativo exige que a informação disponível no tempo atual possa ser usada para modificar previsões futuras, atualizando o prognóstico. Esta técnica é ilustrada na Figura A.6 por Engel *et al.* (2000), descrito a seguir: considerar o ponto  $d_0$ , como a condição inicial do dano no modelo de prognóstico. O prognóstico de vida, desde o tempo  $k$  até um pré-determinado nível de dano é representado por  $RUL_0$  ou tempo de vida restante. Supor que algumas medidas imperfeitas  $z(k)$  avaliando o estado de danos está disponível no tempo  $k = k + p\Delta T$ . O desafio é encontrar um estado de dano atual ótimo para reinicializar o modelo e/ou ajustar os parâmetros para calibrar e estabelecer prognósticos com uma maior precisão.

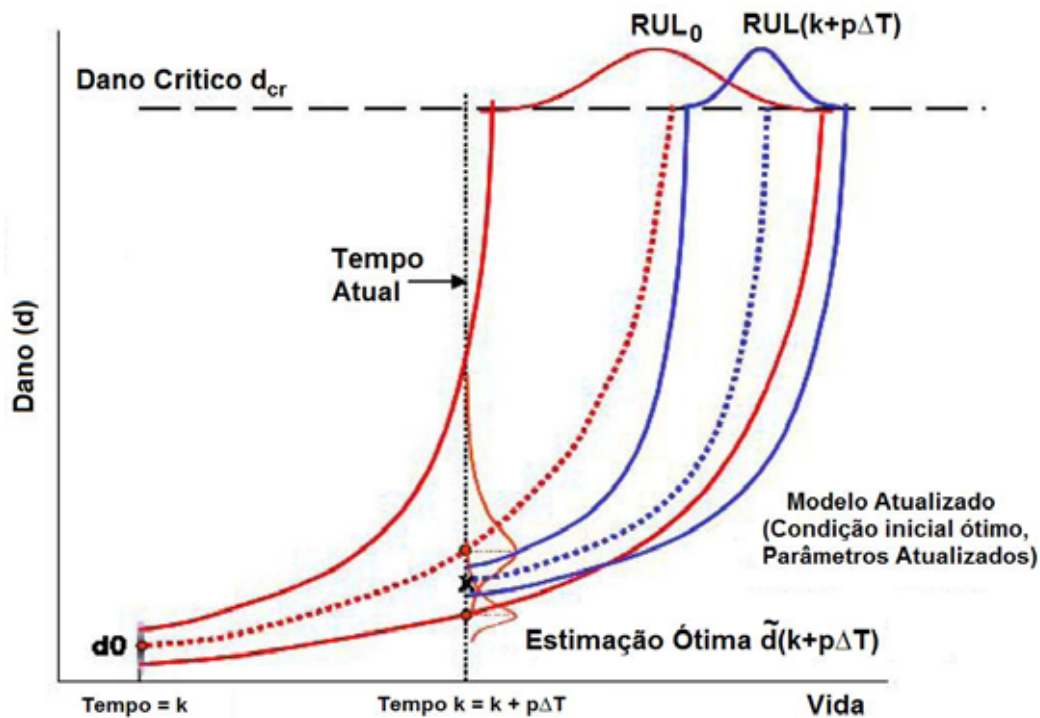


Figura A.6- Conceito de Prognóstico Adaptativo (Roemer *et al.* 2005, modificado).

Apesar da utilização de uma condição inicial, no tempo  $k=k+p\Delta T$ , como mostrado na Figura A.6, é aparente que o significado de predição tem se deslocado e o limite de confiabilidade nos resultados da RUL tem uma menor variância do que a original. O aperfeiçoamento da precisão de predição geralmente pode significar a decisão a tomar baseado em falhas prováveis que reduzirão perdas operacionais (Roemer *et al.*, 2005).

## APÊNDICE B – ARQUIVOS DE CONFIGURAÇÃO

Neste apêndice descrevem-se os três arquivos de configuração utilizados pelo SIMPREBAL. No arquivo de configuração estão os parâmetros de configuração para inicialização do servidor *I-kernel*. Os arquivos de código de falha e de decisão armazenam os códigos de todas as falhas possíveis e suas respectivas tomadas decisão.

### B.1 ARQUIVO DE CONFIGURAÇÃO

O arquivo de configuração possui os parâmetros para a inicialização da aplicação *I-kernel*. Quando o *I-kernel* é iniciado, lê os parâmetros contidos no arquivo. Estes parâmetros são para configurar os parâmetros gerais, da rede FAM, dos servidores OPC, Banco de dados, Emails, dispositivos DFI, tags OPC, tags do banco de dados e tags simuladas. A seguir apresentam-se os campos do arquivo de configuração.

#### B.1.1 Configuração geral

A seguir descrevem-se todos os parâmetros gerais para inicialização do *I-kernel*, estes parâmetros com seus valores são mostrados na Figura B.1.

- VERSION, versão do SIMPREBAL;
- SOCKETPORT, porta *socket* para comunicação do SimprebalServer com o SimprebalClient;
- OPCLIBRARY, biblioteca usada para obter os dados via OPC. Se o valor é um usa-se o JOPCClient, e se é zero o Openscada;
- LOGLEVEL, forma de mostrar o eventos do SimprebalServer . Se o valor é um armazena os eventos num arquivo de texto, e se é zero é mostrado na tela do *prompt*;
- IKERNELTIMER, período de execução do processamento inteligente do SimprebalServer, expressado em milissegundos;
- PERCENTDEADBAND , porcentagem que o valor de uma variável tem que mudar para ser armazenado no banco de dados;
- SENDMAIL, permissão pra envio de emails, se é um o simprebalserver enviar emails, e se é zero não enviara emails;
- LOADINGTIMER, tempo expressado em milissegundos para carregar os valores de todas as variáveis.

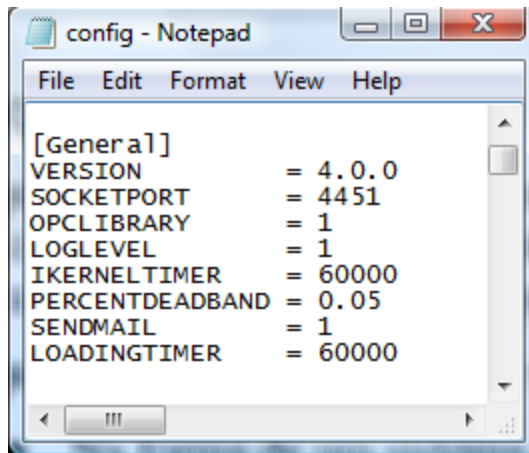


Figura B.1- Parâmetros de configuração geral.

### B.1.2 Parâmetros da FAM

Os parâmetros para o funcionamento da rede FAM apresentada na metodologia são dois: o parâmetro de escolha (Escolha\_Ba) com valores no intervalo  $(0, \infty)$  e o parâmetro padrão de vigilância (Vigilancia\_Pa) tem valores no intervalo  $[0, 1]$ . Estes parâmetros são mostrados na Figura B.2.

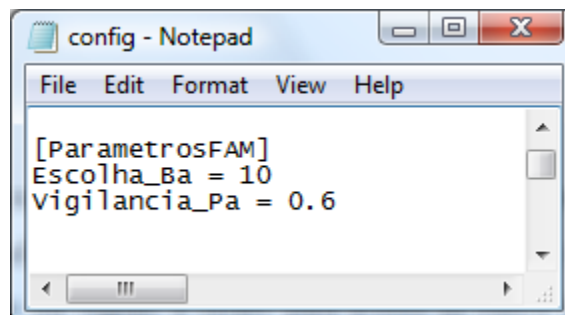


Figura B.2- Parâmetros de configuração da rede FAM.

### B.1.2 Servidores OPC e Tags OPC

Na Figura B.3 mostra-se os parâmetros do servidor OPC. No campo *OPCServers* estão os nomes dos servidores OPC disponíveis, pode ser mais de um. Cada servidor OPC tem sua configuração descrita a seguir:

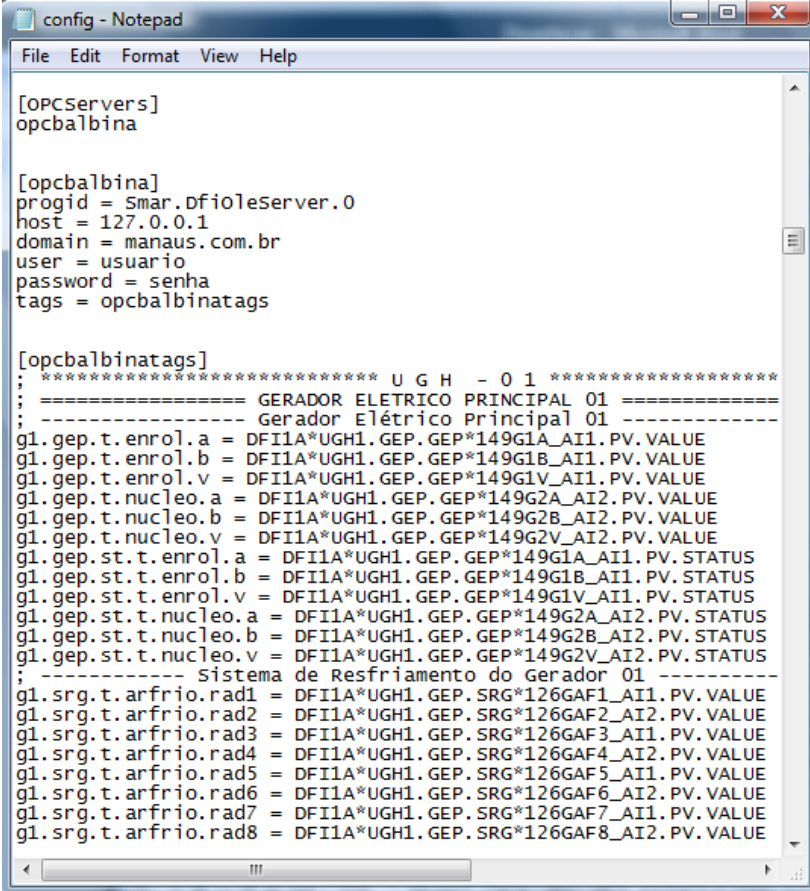
- progid, nome do servidor OPC, depende do fabricante;
- host, número IP do computador onde está o servidor OPC;
- domain, domínio do computador onde está o servidor OPC;
- user, nome do usuário que faz *login* do computador servidor OPC;
- password, senha do usuário;

- tags, lista de tags no servidor OPC.

Para adicionar uma nova tag, tem que ter informação, a que DFI pertence esta tag, a que unidade geradora, sistema e subsistema, e é necessário que haja VALUE e STATUS, além disso, o arquivo de configuração tem que seguir o formato seguinte:

- labelvalue = DFI\*UGH.SISTEMA. SUBSISTEMA\*TAGVALUE
- labelstatus = DFI\*UGH.SISTEMA. SUBSISTEMA\*TAGSTATUS

Onde *labelvalue* é um mnemônico associado a uma determinada tag, que descreve o que essa tag significa, e o *labelstatus* é o mesmo do que o *labelvalue* mais uma letra st depois do segundo ponto (mnemônico para o status de uma tag).



```

[OPCServers]
opcbalbina

[opcbalbina]
progid = Smar.DfiOleServer.0
host = 127.0.0.1
domain = manaus.com.br
user = usuario
password = senha
tags = opcbalbinatags

[opcbalbinatags]
; ***** U G H - 0 1 *****
; ----- GERADOR ELETRICO PRINCIPAL 01 -----
; ----- Gerador Elétrico Principal 01 -----
g1.gep.t.enrol.a = DFIIA*UGH1.GEP.GEP*149G1A_AI1.PV.VALUE
g1.gep.t.enrol.b = DFIIA*UGH1.GEP.GEP*149G1B_AI1.PV.VALUE
g1.gep.t.enrol.v = DFIIA*UGH1.GEP.GEP*149G1V_AI1.PV.VALUE
g1.gep.t.nucleo.a = DFIIA*UGH1.GEP.GEP*149G2A_AI2.PV.VALUE
g1.gep.t.nucleo.b = DFIIA*UGH1.GEP.GEP*149G2B_AI2.PV.VALUE
g1.gep.t.nucleo.v = DFIIA*UGH1.GEP.GEP*149G2V_AI2.PV.VALUE
g1.gep.st.t.enrol.a = DFIIA*UGH1.GEP.GEP*149G1A_AI1.PV.STATUS
g1.gep.st.t.enrol.b = DFIIA*UGH1.GEP.GEP*149G1B_AI1.PV.STATUS
g1.gep.st.t.enrol.v = DFIIA*UGH1.GEP.GEP*149G1V_AI1.PV.STATUS
g1.gep.st.t.nucleo.a = DFIIA*UGH1.GEP.GEP*149G2A_AI2.PV.STATUS
g1.gep.st.t.nucleo.b = DFIIA*UGH1.GEP.GEP*149G2B_AI2.PV.STATUS
g1.gep.st.t.nucleo.v = DFIIA*UGH1.GEP.GEP*149G2V_AI2.PV.STATUS
; ----- Sistema de Resfriamento do Gerador 01 -----
g1.srg.t.arfrio.rad1 = DFIIA*UGH1.GEP.SRG*126GAF1_AI1.PV.VALUE
g1.srg.t.arfrio.rad2 = DFIIA*UGH1.GEP.SRG*126GAF2_AI2.PV.VALUE
g1.srg.t.arfrio.rad3 = DFIIA*UGH1.GEP.SRG*126GAF3_AI1.PV.VALUE
g1.srg.t.arfrio.rad4 = DFIIA*UGH1.GEP.SRG*126GAF4_AI2.PV.VALUE
g1.srg.t.arfrio.rad5 = DFIIA*UGH1.GEP.SRG*126GAF5_AI1.PV.VALUE
g1.srg.t.arfrio.rad6 = DFIIA*UGH1.GEP.SRG*126GAF6_AI2.PV.VALUE
g1.srg.t.arfrio.rad7 = DFIIA*UGH1.GEP.SRG*126GAF7_AI1.PV.VALUE
g1.srg.t.arfrio.rad8 = DFIIA*UGH1.GEP.SRG*126GAF8_AI2.PV.VALUE

```

Figura B.3- Parâmetros do Servidor OPC e tags associadas.

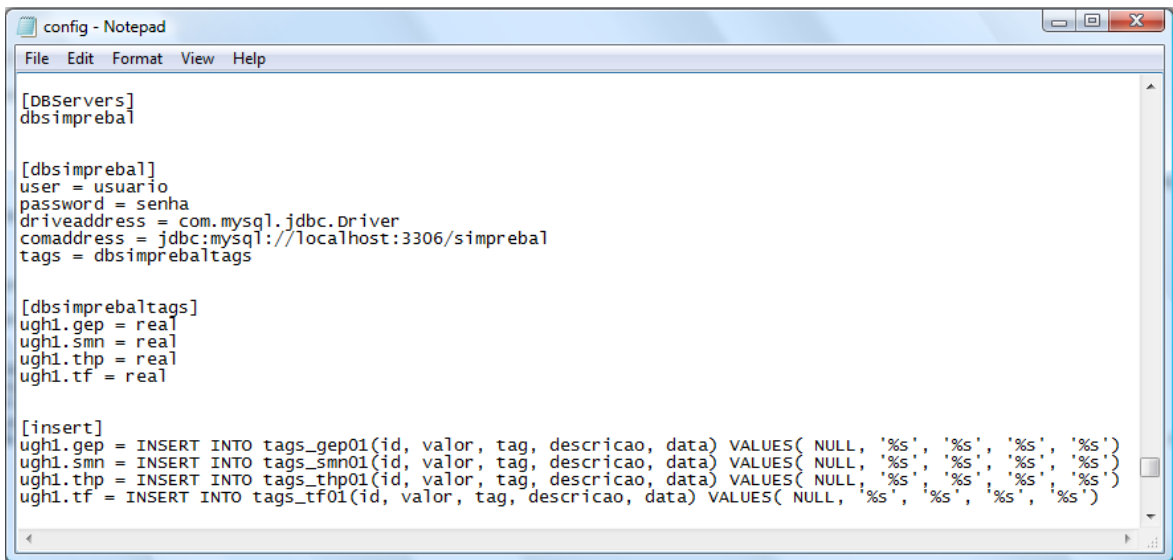
### B.1.2 Servidores de banco de dados

Os parâmetros do servidor de banco de dados são mostrados na Figura B.4. No campo *DBServers* estão os nomes dos servidores de banco de dados disponíveis, pode ser mais de um. Cada servidor de banco de dados tem sua configuração descrita a seguir:

- user, usuário do bando de dados;

- password, senha do usuário do banco de dados;
- driveaddress, driver, depende do tipo de banco de dados;
- comaddress, endereço eletrônico do bando de dados;
- tags, lista de tags no servidor do Banco de Dados.

As tags do banco de dados podem ser reais ou simuladas. As tags reais estão ligadas a comandos *insert*, *select* ou *update*. Para armazenamento de tags usa-se o comando *insert*.



```

[DBServers]
dbsimprebal

[dbsimprebal]
user = usuario
password = senha
driveaddress = com.mysql.jdbc.Driver
comaddress = jdbc:mysql://localhost:3306/simprebal
tags = dbsimprebaltags

[dbsimprebaltags]
ugh1.gep = real
ugh1.smn = real
ugh1.thp = real
ugh1.tf = real

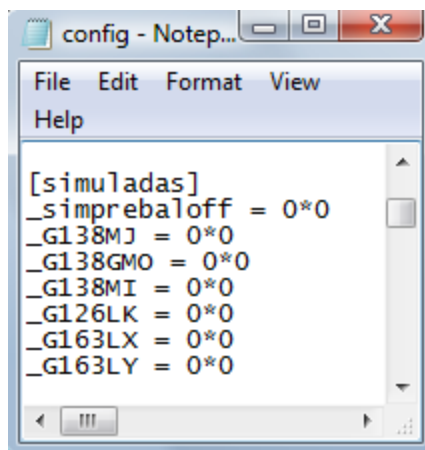
[insert]
ugh1.gep = INSERT INTO tags_gep01(id, valor, tag, descricao, data) VALUES( NULL, '%s', '%s', '%s', '%s')
ugh1.smn = INSERT INTO tags_smn01(id, valor, tag, descricao, data) VALUES( NULL, '%s', '%s', '%s', '%s')
ugh1.thp = INSERT INTO tags_thp01(id, valor, tag, descricao, data) VALUES( NULL, '%s', '%s', '%s', '%s')
ugh1.tf = INSERT INTO tags_tf01(id, valor, tag, descricao, data) VALUES( NULL, '%s', '%s', '%s', '%s')

```

Figura B.4- Parâmetros do servidor banco de dados e tags associadas.

### B.1.2 Tags Simuladas

As tags simuladas são mostradas na Figura B.5. Estas tags são usadas em modo *offline* para testas variáveis simuladas.



```

[simuladas]
_simprebaloff = 0*0
_G138MJ = 0*0
_G138GMO = 0*0
_G138MI = 0*0
_G126LK = 0*0
_G163LX = 0*0
_G163LY = 0*0

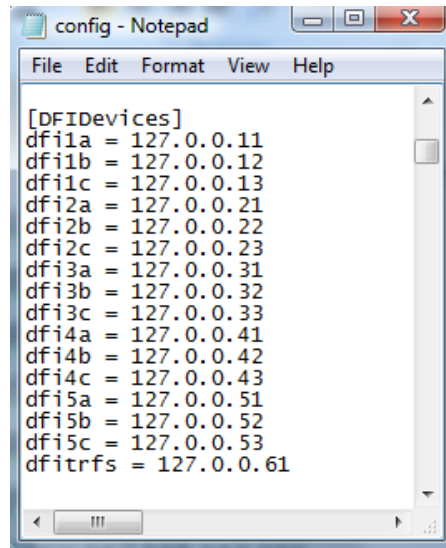
```

Figura B.5- Tags simulados.



### B.1.2 Dispositivos DFI

Sendo que todos os instrumentos estão conectados a um dispositivo DFI. Os parâmetros das DFI são mostrados na Figura B.6. Estes parâmetros incluem o nome do dispositivo DFI e o numero de IP.



```
config - Notepad
File Edit Format View Help

[DFIDevices]
dfi1a = 127.0.0.11
dfi1b = 127.0.0.12
dfi1c = 127.0.0.13
dfi2a = 127.0.0.21
dfi2b = 127.0.0.22
dfi2c = 127.0.0.23
dfi3a = 127.0.0.31
dfi3b = 127.0.0.32
dfi3c = 127.0.0.33
dfi4a = 127.0.0.41
dfi4b = 127.0.0.42
dfi4c = 127.0.0.43
dfi5a = 127.0.0.51
dfi5b = 127.0.0.52
dfi5c = 127.0.0.53
dfitrfs = 127.0.0.61
```

Figura B.6- Parâmetros dos dispositivos DFI.

### B.1.2 Servidor de Email

Os parâmetros do servidor de Emails, grupos de usuario e os emails dos membros dos grupos são mostrados na Figura B.7.



```
config - Notepad
File Edit Format View Help

[EmailServer]
hostname = 127.0.0.1
user = usuario
password = senha
serverName = nome
email = simprebal@simprebal.com

[emailsgroups]
administrators
electricians
electronics
mechanics
operators
programmers

[administrators]
admin1 = admin1@mail.com
admin2 = admin2@mail.com
admin3 = admin3@mail.com

[electricians]
electric1 = electric1@mail.com
electric2 = electric2@mail.com
```

Figura B.7- Parâmetros do servidor de email e lista de emails por grupos.

Estes parâmetros são descritos a seguir:

- hostname, número IP do computador onde está o servidor de email;
- user, usuário da conta do email;
- password, senha do usuário de email;
- servername, nome do servidor de email;
- email, endereço de email do usuário.

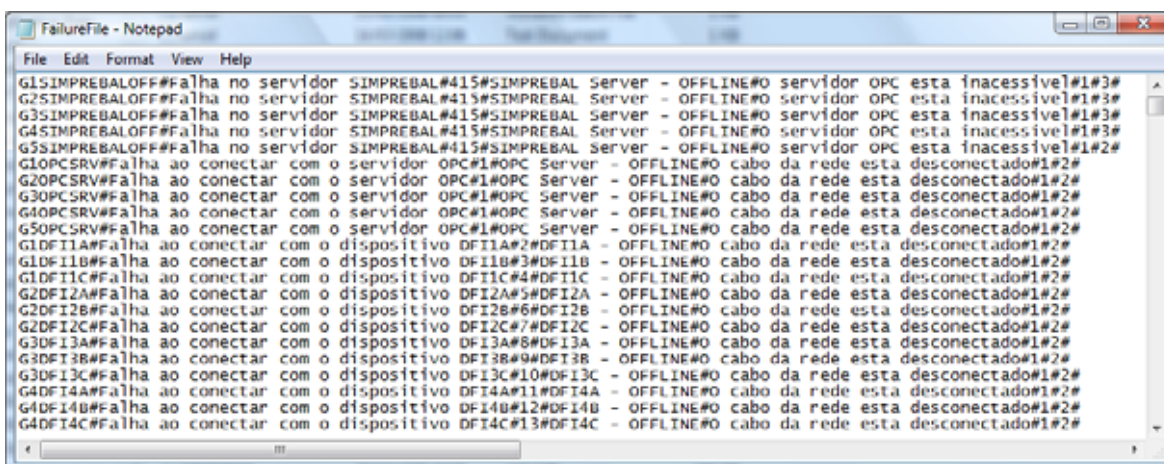
## B.2 ARQUIVO CÓDIGOS DE FALHA

O arquivo códigos de falha mostrado na Figura B.8 é usado para dispor de dados de falha de uma maneira codificada para o *I-kernel*. Este arquivo é serializado e usado como um objeto pela aplicação Java. Os dados deste arquivo têm a seguinte estrutura:

Código#descrição#id do equipamento#modo#causa#detecção#severidade#setpoint

G126GAQ1D1#Alta temperatura de ar quente dos radiadores#409#G126GAQ1 - ALERTA#Operação prolongada do gerador com carga máxima. Sujeira nos radiadores#1#7#70.

Note que o trecho de código acima contém códigos separados pelo caractere #. O primeiro se refere ao código da falha, o segundo é a descrição da falha, o terceiro é o ID do equipamento conforme cadastrado no banco de dados. O quarto é modo de falha (tag - condição), o quinto é a causa da falha, o sexto o fator de detecção da falha, o sétimo é a severidade da falha (estes dois últimos campos são valores referentes ao FMEA) e finalmente o valor de *setpoint*.



```
FailureFile - Notepad
File Edit Format View Help
G1SIMPREGALOFF#Falha no servidor SIMPREBAL#415#SIMPREGAL Server - OFFLINE#0 servidor OPC esta inacessivel#1#3#
G2SIMPREGALOFF#Falha no servidor SIMPREBAL#415#SIMPREGAL Server - OFFLINE#0 servidor OPC esta inacessivel#1#3#
G3SIMPREGALOFF#Falha no servidor SIMPREBAL#415#SIMPREGAL Server - OFFLINE#0 servidor OPC esta inacessivel#1#3#
G4SIMPREGALOFF#Falha no servidor SIMPREBAL#415#SIMPREGAL Server - OFFLINE#0 servidor OPC esta inacessivel#1#3#
G5SIMPREGALOFF#Falha no servidor SIMPREBAL#415#SIMPREGAL Server - OFFLINE#0 servidor OPC esta inacessivel#1#2#
G1OPCSRVR#Falha ao conectar com o servidor OPC#1#OPC Server - OFFLINE#0 cabo da rede esta desconectado#1#2#
G2OPCSRVR#Falha ao conectar com o servidor OPC#1#OPC Server - OFFLINE#0 cabo da rede esta desconectado#1#2#
G3OPCSRVR#Falha ao conectar com o servidor OPC#1#OPC Server - OFFLINE#0 cabo da rede esta desconectado#1#2#
G4OPCSRVR#Falha ao conectar com o servidor OPC#1#OPC Server - OFFLINE#0 cabo da rede esta desconectado#1#2#
G5OPCSRVR#Falha ao conectar com o servidor OPC#1#OPC Server - OFFLINE#0 cabo da rede esta desconectado#1#2#
G1DFI1A#Falha ao conectar com o dispositivo DFI1A#2#DFI1A - OFFLINE#0 cabo da rede esta desconectado#1#2#
G1DFI1B#Falha ao conectar com o dispositivo DFI1B#3#DFI1B - OFFLINE#0 cabo da rede esta desconectado#1#2#
G1DFI1C#Falha ao conectar com o dispositivo DFI1C#4#DFI1C - OFFLINE#0 cabo da rede esta desconectado#1#2#
G2DFI2A#Falha ao conectar com o dispositivo DFI2A#5#DFI2A - OFFLINE#0 cabo da rede esta desconectado#1#2#
G2DFI2B#Falha ao conectar com o dispositivo DFI2B#6#DFI2B - OFFLINE#0 cabo da rede esta desconectado#1#2#
G2DFI2C#Falha ao conectar com o dispositivo DFI2C#7#DFI2C - OFFLINE#0 cabo da rede esta desconectado#1#2#
G3DFI3A#Falha ao conectar com o dispositivo DFI3A#8#DFI3A - OFFLINE#0 cabo da rede esta desconectado#1#2#
G3DFI3B#Falha ao conectar com o dispositivo DFI3B#9#DFI3B - OFFLINE#0 cabo da rede esta desconectado#1#2#
G3DFI3C#Falha ao conectar com o dispositivo DFI3C#10#DFI3C - OFFLINE#0 cabo da rede esta desconectado#1#2#
G4DFI4A#Falha ao conectar com o dispositivo DFI4A#11#DFI4A - OFFLINE#0 cabo da rede esta desconectado#1#2#
G4DFI4B#Falha ao conectar com o dispositivo DFI4B#12#DFI4B - OFFLINE#0 cabo da rede esta desconectado#1#2#
G4DFI4C#Falha ao conectar com o dispositivo DFI4C#13#DFI4C - OFFLINE#0 cabo da rede esta desconectado#1#2#
```

Figura B.8- Arquivo códigos de falha.

### B.3 ARQUIVO CÓDIGOS DE DECISÃO

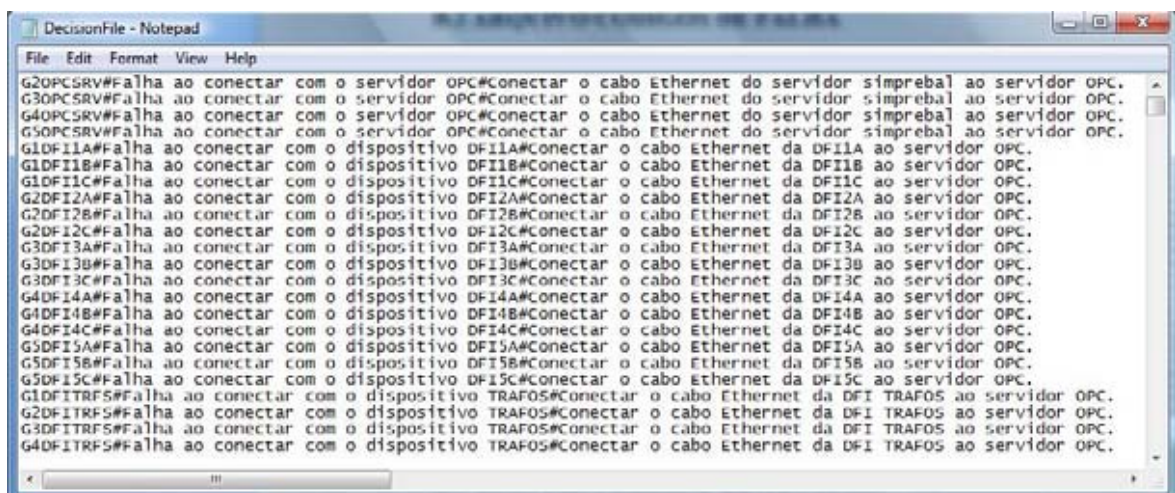
O arquivo códigos de decisão mostrado na Figura B.9 é usado para dispor dos dados de tomada decisão de uma maneira fácil para a ferramenta C&M. Este arquivo é serializado e usado como um objeto pela aplicação Java. Para acrescentar uma nova linha neste arquivo é necessário que o usuário crie um código de falha, no arquivo de falha, e então utilize este código no arquivo de decisão, referente à nova falha inserida. As mensagens gravadas no arquivo de decisão seguem a forma abaixo:

Código#falha#tomada de decisão.

G126GAQ1D1#Alta temperatura de ar quente dos radiadores#Reduzir a carga gerada.

Efetuar limpeza interna dos radiadores.

Note que o trecho de código acima contém códigos separados pelo caractere #. O primeiro se refere ao código da decisão, o segundo é a descrição da falha, o terceiro é a descrição da decisão.



```
DecisionFile - Notepad
File Edit Format View Help
G20PC5RV#Falha ao conectar com o servidor OPC#Conectar o cabo Ethernet do servidor simprebal ao servidor OPC.
G30PC5RV#Falha ao conectar com o servidor OPC#Conectar o cabo Ethernet do servidor simprebal ao servidor OPC.
G40PC5RV#Falha ao conectar com o servidor OPC#Conectar o cabo Ethernet do servidor simprebal ao servidor OPC.
G50PC5RV#Falha ao conectar com o servidor OPC#Conectar o cabo Ethernet do servidor simprebal ao servidor OPC.
G1DFI1A#Falha ao conectar com o dispositivo DFI1A#Conectar o cabo Ethernet da DFI1A ao servidor OPC.
G1DFI1B#Falha ao conectar com o dispositivo DFI1B#Conectar o cabo Ethernet da DFI1B ao servidor OPC.
G1DFI1C#Falha ao conectar com o dispositivo DFI1C#Conectar o cabo Ethernet da DFI1C ao servidor OPC.
G2DFI2A#Falha ao conectar com o dispositivo DFI2A#Conectar o cabo Ethernet da DFI2A ao servidor OPC.
G2DFI2B#Falha ao conectar com o dispositivo DFI2B#Conectar o cabo Ethernet da DFI2B ao servidor OPC.
G2DFI2C#Falha ao conectar com o dispositivo DFI2C#Conectar o cabo Ethernet da DFI2C ao servidor OPC.
G3DFI3A#Falha ao conectar com o dispositivo DFI3A#Conectar o cabo Ethernet da DFI3A ao servidor OPC.
G3DFI3B#Falha ao conectar com o dispositivo DFI3B#Conectar o cabo Ethernet da DFI3B ao servidor OPC.
G3DFI3C#Falha ao conectar com o dispositivo DFI3C#Conectar o cabo Ethernet da DFI3C ao servidor OPC.
G4DFI4A#Falha ao conectar com o dispositivo DFI4A#Conectar o cabo Ethernet da DFI4A ao servidor OPC.
G4DFI4B#Falha ao conectar com o dispositivo DFI4B#Conectar o cabo Ethernet da DFI4B ao servidor OPC.
G4DFI4C#Falha ao conectar com o dispositivo DFI4C#Conectar o cabo Ethernet da DFI4C ao servidor OPC.
G5DFI5A#Falha ao conectar com o dispositivo DFI5A#Conectar o cabo Ethernet da DFI5A ao servidor OPC.
G5DFI5B#Falha ao conectar com o dispositivo DFI5B#Conectar o cabo Ethernet da DFI5B ao servidor OPC.
G5DFI5C#Falha ao conectar com o dispositivo DFI5C#Conectar o cabo Ethernet da DFI5C ao servidor OPC.
G1DFITRF5#Falha ao conectar com o dispositivo TRAF05#Conectar o cabo Ethernet da DFI TRAF05 ao servidor OPC.
G2DFITRF5#Falha ao conectar com o dispositivo TRAF05#Conectar o cabo Ethernet da DFI TRAF05 ao servidor OPC.
G3DFITRF5#Falha ao conectar com o dispositivo TRAF05#Conectar o cabo Ethernet da DFI TRAF05 ao servidor OPC.
G4DFITRF5#Falha ao conectar com o dispositivo TRAF05#Conectar o cabo Ethernet da DFI TRAF05 ao servidor OPC.
```

Figura B.9- Arquivo códigos de decisão.

## APÊNDICE C – CÁLCULO DOS CAMPOS DA CLASSE TAG

Neste apêndice são mostrados os procedimentos para o cálculo dos campos da classe Tag. A classe Tag é usada pelos arquivos de regras no processamento inteligente. Este processo de cálculo é baseado na instrumentação *fieldbus* da Smar.

### C.1 PROCESSAMENTO DO ITEM VALUE

O processo de cálculo dos campos da classe Tag é mostrado na Figura C.1. Um transmissor *fieldbus* (i.e. 149G1A) disponibiliza valores dos seus parâmetros através de itens, para o processamento selecionamos dois itens: VALUE (i.e. 149G1A\_AI1.PV.VALUE) e STATUS (i.e. 149G1A\_AI1.PV.STATUS). No tipo VALUE estão contidas informações do valor da grandeza, e a qualidade do sinal OPC. O tipo STATUS contém informação da qualidade do sinal da instrumentação *fieldbus*.

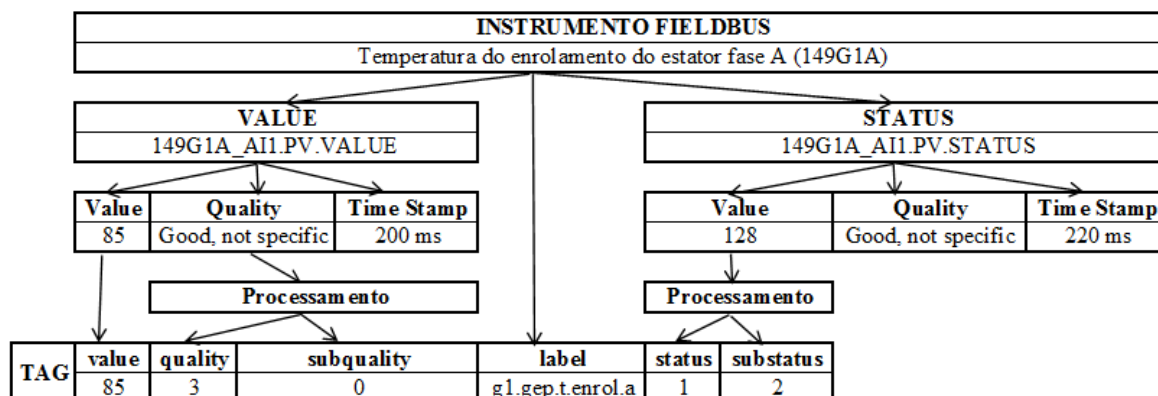


Figura C.1- processo de obtenção da classe Tag.

No item VALUE (i.e. 149G1A\_AI1.PV.VALUE) é processada a propriedade *Quality*. Esta propriedade é do tipo string (i.e. *Good, non-specific*). Para maior facilidade no processamento das regras vamos converter a propriedade *Quality* de string para números inteiros *quality* e *subquality*. Os valores de *quality* e *subquality* são obtidos a partir da Tabela C.1. Depois deste processamento teremos os valores dos campos *value*, *quality* e *subquality* da classe Tag.

Tabela C.1- Relação entre o *Quality* e valores *quality* e *subquality*.

<b>Quality</b>	<b>quality</b>	<b>subquality</b>
Good, non-specific	3	0
Good, Local Override	3	6
Uncertain, non-specific	1	0
Uncertain, Last Usable Value	1	1
Uncertain, Sensor Not Accurate	1	4
Uncertain, Engineering Units Exceeded	1	5
Uncertain, Sub-Normal	1	6
Bad, non-specific	0	0
Bad, Configuration Error	0	1
Bad, Not Connected	0	2
Bad, Device Failure	0	3
Bad, Sensor Failure	0	4
Bad, Last Known Value	0	5
Bad, Comm Failure	0	6
Bad, Out of Service	0	7

## C.2 PROCESSAMENTO DO ITEM STATUS

Uma característica muito útil disponibilizada pela instrumentação *fieldbus* é o item *STATUS* (i.e. 149G1A\_AI1.PV.STATUS). A grande vantagem deste item é que permite qualificar o valor do sinal *fieldbus*, em caso de falha leva ao instrumento para uma condição de segurança (*Fail-Safe*) (Smar, 2005). A propriedade *value* do item *STATUS* é apresentada na Figura C.2, pode-se observar que é composta de três elementos: *Quality*, *SubStatus* e *Limits*. Estes três elementos são relacionados matematicamente na Equação C.1.

$$status = 64 * quality + 4 * substatus + Limit \quad (C.1)$$

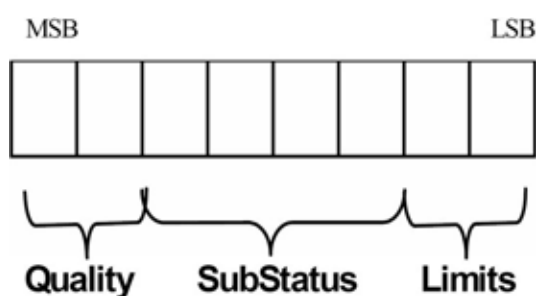


Figura C.2- Estrutura da propriedade *value* do item *STATUS* (Smar, 2005).

A seguir apresenta-se um exemplo para o cálculo dos valores numéricos do *Quality*, *SubStatus* e *Limit*.

Valor do item *STATUS* = 78

Dividido o número por 64. O quociente será o *Quality* e armazenado o resto:

$Quality = 78 / 64 = 1$  (*Uncertain*, ver Tabela C.2)

Resto = 14

Dividido o resto por 4. O quociente será o *SubStatus* e o resto será o limite:

$SubStatus = 14 / 4 = 3$  (Valor inicial, ver Tabela C.2)

Limit = 2

Depois de efetuar estas operações têm-se todos os valores dos campos da classe Tag: *label*, *value*, *quality*, *subquality*, *status*, e *substatus*.

Tabela C.2- Valores de *SubStatus* e *Quality* no item *STATUS* (Amaya *et al.*, 2007c).

		Quality			
		0 = Bad	1 = Uncertain	2 = Good Non Cascade	3 = Good Cascade
<b>SubStatus</b>	0	Não específico	Não específico	Não específica (baixa prioridade)	Não específica
	1	Erro na configuração	Último valor usável	Alarme ativo de bloco	Inicialização aprovada (IA)
	2	Não conectado	Substituto	Alarme ativo de consulta	Requisição de inicialização (IR)
	3	Falha no Equipamento	Valor inicial	Alarme ativo crítico	Não solicitado (NI)
	4	Falha no Sensor	Conversão do sensor não exato	Alarme de bloco desconhecido	Não selecionado (NS)
	5	Sem Comunicação, com último valor usavel	Violação de limite de unidade técnica	Alarme de consulta desconhecido	Cancelamento local (LO)
	6	Sem Comunicação, com valor não usável	Sub-normal	Alarme crítico desconhecido	Estado de falha ativo (FSA)
	7	Fora de serviço (alta prioridade)			Estado de falha iniciado (IFS)

## APÊNDICE D – MÉTODOS DE ANÁLISE FMEA E FTA

Neste apêndice são descritos os conceitos de Análise de Modos e Efeitos de Falhas (FMEA), Análise de Árvore de Falhas (FTA) e uma comparação entres estes dois métodos de analise.

### D.1 ANÁLISE DE MODOS E EFEITOS DE FALHAS - FMEA

Inicialmente faz-se o levantamento das funções e dos modos e efeitos de falhas, FMEA, a partir da descrição textual do sistema desenvolvida, dos registros de cartões de anormalidades (ordens de serviços de manutenção), dos planos de manutenção atuais e dos descritivos funcionais e de instrumentação dos equipamentos e componentes. A documentação da análise FMEA foi desenvolvida segundo o formulário padronizado mostrado na Tabela D.1.

Tabela D.1- Formulário padronizado de análise FMEA.

IDENTIFICAÇÃO DO EQUIPAMENTO					FATORES PARA AVALIAÇÃO DO COMPONENTE							TAREFA PROPOSTA PARA MANUTENÇÃO
IDENTIFICAÇÃO DO SUBSISTEMA					SEGURANÇA E MEIO AMBIENTE	PERDA DE FATURAMENTO	CORTE DE CARGA	DETECÇÃO	SEVERIDADE	OCORRÊNCIA	AVALIAÇÃO GERAL (NPR)	
FUNÇÃO		FALHA FUNCIONAL	MODO DE FALHA	EFEITO DA FALHA								
COMPONENTE	FUNÇÃO DO COMPONENTE											

A seguir há uma explicação de cada coluna do formulário apresentado.

- Função: Descrição da função do subsistema ou equipamento.
- Componente: Identificação do componente.
- Função do componente: Descrição sucinta e exata da tarefa que o componente deve desempenhar.
- Falha funcional: Descrição de todas as possíveis falhas pertinentes a cada componente.
- Modo de falha: Descrição simples e concisa das ocorrências (causas) que podem dar origem ao tipo de falha considerado.

- Efeito da falha: Conseqüência da ocorrência da falha, percebida ou não pelo usuário final. Pode ser local (não afeta os outros componentes) ou global (pode afetar outras funções ou componentes).
- Fatores para avaliação do componente: Consiste numa série de critérios utilizados para avaliar a criticidade ou prioridade de risco de um componente. Nesta avaliação é considerada a influência de três parâmetros: severidade, ocorrência e detecção das falhas. Para padronizar e tornar menos subjetiva a avaliação da severidade de cada falha funcional, foram categorizadas três classes de efeitos de falhas: os que afetam a segurança e o meio ambiente, os que provocam perda de faturamento, e os que provocam corte de carga. A seguir há uma descrição mais detalhada das classes de efeitos das falhas.
- Segurança e/ou o meio ambiente: Caracteriza a severidade de uma falha de acordo com a intensidade com que ela pode afetar a segurança dos funcionários ou o meio ambiente, conforme estabelecido pelas normas ISO de segurança e proteção ao meio ambiente, ISO 18001 (segurança) ISO 14001 (meio ambiente). A Tabela D.2 mostra o significado dos valores atribuídos aos índices de classificação das falhas quanto à intensidade com que afetam a segurança e/ou o meio ambiente.
- Perda de faturamento: Indica o grau com que uma determinada falha afeta a economia da usina, na medida em que afeta a geração de energia a capacidade total. A Tabela D.3 apresenta o significado dos valores atribuídos aos índices de perda de faturamento.
- Corte de carga: Indica a probabilidade de a falha provocar parada da unidade geradora. Os critérios de classificação dos índices de corte de carga são mostrados na Tabela D.4.

Tabela D.2- Significado dos índices de segurança e/ou meio ambiente.

<b>Segurança e/ou Meio Ambiente</b>	
1	Não afeta a segurança e/ou o meio ambiente
2-3	Remota possibilidade de afetar a segurança e/ ou o meio ambiente
4-6	Possibilidade moderada de afetar a segurança e/ou o meio ambiente
7-8	Grande possibilidade de afetar a segurança e/ ou o meio ambiente
9	Afeta a segurança e/ ou o meio ambiente
10	Afeta gravemente a segurança e/ ou o meio ambiente



Tabela D.3- Significado dos índices de Perda de faturamento.

<b>Perda de Faturamento</b>	
1	Não provoca perda de faturamento
3-5	Pode provocar perda de faturamento menor que 2,5% da receita mensal
7-10	Pode provocar perda de faturamento maior ou igual a 2,5% da receita mensal

Tabela D.4- Significado dos índices de corte de carga.

<b>Corte de Carga</b>	
1	Não provoca corte de carga
3	Risco remoto de provocar corte de carga
5	Risco moderado de provocar corte de carga
7	Provoca corte de carga de até 5% da carga da instalação
10	Provoca corte de carga maior ou igual a 5% da capacidade máxima da instalação

Tabela D.5- Significado dos índices de severidade.

<b>Severidade</b>	
1	Falha de menor importância
2 – 3	Provoca redução do desempenho do componente
4 – 6	O componente sofrerá uma degradação progressiva
7 – 8	O componente não desempenha sua função
9	Colapso do processo
10	Os problemas são catastróficos e podem ocasionar danos a bens ou pessoas

Os parâmetros de avaliação da criticidade das falhas são descritos a seguir:

- Severidade: Trata-se de um índice que reflete a gravidade das conseqüências de uma falha. Quanto maior o índice, maior a gravidade. O índice de severidade foi assumido como sendo igual ao maior índice dentre os índices das três classes de efeitos de falhas descritos anteriormente (segurança e meio ambiente, perda de faturamento ou corte de carga). A Tabela D.5 apresenta o padrão utilizado para quantificação da gravidade das falhas em índices de severidade.

- Ocorrência: É um índice definido em função do número de ocorrências de falhas registrados em um período considerado. A Tabela D.6 relaciona os valores e conceitos dos índices de ocorrência.
- Detecção: É um índice construído com base na estimativa da probabilidade de uma falha ser detectada, assumindo-se que ela tenha ocorrido. A
- Tabela D.7 relaciona os valores e conceitos dos índices de detecção.

Tabela D.6- Significado dos índices de ocorrência de falhas.

<b>Ocorrência</b>	
1	Menor ou igual a 1 em 8 anos
2	1 falha no período analisado
3	2 falhas
5	3 falhas
7	4 falhas
10	5 ou mais falhas

Tabela D.7- Significado dos índices de detecção.

<b>Detecção</b>	
1	Probabilidade muito alta de detecção
2 – 3	Probabilidade alta de detecção
4 – 6	Probabilidade moderada de detecção
7 – 8	Probabilidade pequena
9	Probabilidade muito pequena
10	Probabilidade remota

## **D.2 ANÁLISE DE ÁRVORE DE FALHAS - FTA**

A análise de árvore de falhas consiste na construção de um diagrama lógico, através de um processo dedutivo que partindo de um evento indesejado pré-definido, busca as possíveis causas de tal evento. O processo segue investigando as sucessivas combinações de falhas dos componentes até atingir as chamadas falhas básicas (ou eventos básicos), as quais constituem o limite de resolução da análise. O evento indesejado é comumente chamado de evento topo da árvore. Portanto, o conceito fundamental da FTA consiste na tradução de um sistema físico em um diagrama lógico estruturado, FT (*Fault Tree*), em que certas causas específicas conduzem a um evento topo de interesse.

Estruturada a árvore, procede-se a definição dos cortes mínimos, isto é, das combinações mínimas de eventos que quando ocorridas levam falha do sistema. Um exemplo dos caminhos críticos da árvore, isto é, os cortes mínimos com maior probabilidade de ocorrência são mostrados na Tabela D.8.

Tabela D.8- Caminhos críticos da árvore de falhas.

<b>Equipamento</b>	<b>Evento Básico</b>	<b>Probabilidade do corte mínimo</b>
Mancal Guia	Corrosão na tubulação da serpentina do sistema de resfriamento de óleo	8,57E-05
Mancal de escora	Desgaste na sede da válvula do sistema de circulação de óleo	5,72E-05
Mancal de escora	Corrosão dos patins da cuba	2,86E-05
Mancal guia	Desgaste do metal patente da cuba	2,86E-05
Sistema de vedação do eixo	Desgaste dos anéis de vedação	2,86E-05
Sistema de vedação do eixo	Vazamento na tubulação de drenagem	2,86E-05
Sistema de vedação do eixo	Desgaste na sede da válvula de alívio do sistema de drenagem	2,86E-05
Sistema de vedação do eixo	Quebra do mecanismo interno da válvula de alívio do sistema de drenagem	2,86E-05
Sistema do distribuidor	Deterioração das buchas de fixação das palhetas	2,86E-05

### **D.3 COMPARAÇÃO ENTRE FTA E FMEA**

Apesar da semelhança entre as duas técnicas no que se refere a finalidade, existem várias diferenças entre elas quanto a aplicação e ao procedimento de análise. A Tabela D.9 compara as duas técnicas apresentando suas principais diferenças.

Tabela D.9- Comparação entre FTA e FMEA.

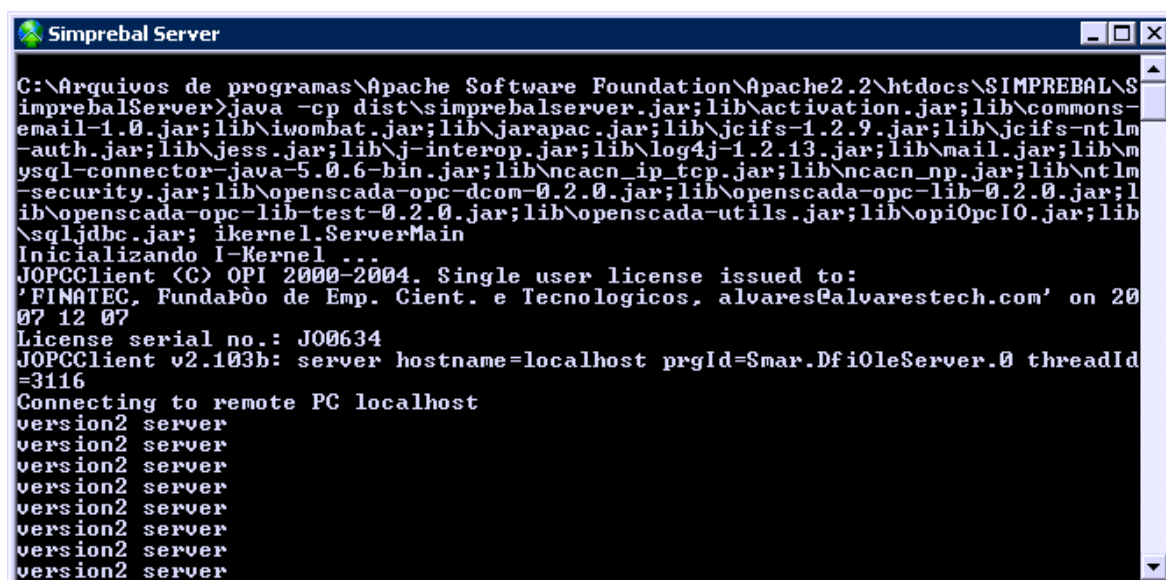
	<b>FTA</b>	<b>FMEA</b>
Objetivo	Identificação das causas primárias das falhas	Identificação das falhas críticas em cada componente, suas causas e conseqüências
	Elaboração de uma relação lógica entre falhas primárias e falha final do produto	Hierarquizar as falhas
Procedimento	Identificação da falha que é detectada pelo usuário do produto	Análise das falhas em potencial de todos os elementos do sistema, e previsão das conseqüências
	Relacionar essa falha com falhas intermediárias e eventos mais básicos por meio de símbolos lógicos	Relação de ações corretivas (ou preventivas) a serem tomadas
Aplicação	Melhor método para análise individual de uma falha específica	Pode ser utilizada na análise de falhas simultâneas ou correlacionadas
	O enfoque é dado à falha final do sistema	Todos os componentes do sistema são passíveis de análise

## APÊNDICE E – INICIAÇÃO E OPERAÇÃO DO SIMPREBAL

Neste apêndice são descritos os passos a seguir para iniciar o servidor e cliente SIMPREBAL. Também é apresentada a descrição detalhada dos componentes da ferramenta C&M, os procedimentos para a inicialização e operação e as atividades suportadas pelo cliente tais como: cálculo de KPI, gráficos de variáveis *online* e históricos, mensagens de anomalias e tomadas de decisão.

### E.1 SIMPREBAL SERVER

A inicialização do *simprebalserver* é executando o arquivo “server.bat” localizado na pasta principal do SIMPREBAL “...\simprebal\simprebalserver”. Uma vez inicializada aparece uma janela do *simprebalserver* em execução mostrada na Figura E.1.



```
C:\Arquivos de programas\Apache Software Foundation\Apache2.2\htdocs\SIMPREBAL\simprebalServer>java -cp dist\simprebalserver.jar;lib\activation.jar;lib\commons-email-1.0.jar;lib\iwombat.jar;lib\jarapac.jar;lib\jcifs-1.2.9.jar;lib\jcifs-ntlm-auth.jar;lib\jess.jar;lib\j-interop.jar;lib\log4j-1.2.13.jar;lib\mail.jar;lib\mysql-connector-java-5.0.6-bin.jar;lib\ncacn_ip_tcp.jar;lib\ncacn_np.jar;lib\ntlm-security.jar;lib\openscada-opc-dcom-0.2.0.jar;lib\openscada-opc-lib-0.2.0.jar;lib\openscada-opc-lib-test-0.2.0.jar;lib\openscada-utils.jar;lib\opiopci0.jar;lib\sqljdbc.jar; ikernel.ServerMain
Inicializando I-Kernel ...
JOPCCClient (C) OPI 2000-2004. Single user license issued to:
'FINATEC, Funda o de Emp. Cient. e Tecnologicos, alvares@alvarestech.com' on 20
07 12 07
License serial no.: J00634
JOPCCClient v2.103b: server hostname=localhost prgId=Smar.Dfi0leServer.0 threadId
=3116
Connecting to remote PC localhost
version2 server
version2 server
version2 server
version2 server
version2 server
version2 server
version2 server
version2 server
```

Figura E.1- Servidor Simprebal em execu o.

### E.2 SIMPREBAL CLIENT

Depois de iniciado o *simprebalServer*, o lado cliente pode ser inicializado atrav s de um *browser*, pode ser o Internet Explorer ou o Mozilla Firefox, etc. Escrever o endere o URL do SIMPREBAL e aparecer  no browser a tela de *login* do SIMPREBAL, conforme mostrado na Figura E.2, na qual o usu rio deve digitar a senha cadastrada. Caso o cadastro ainda n o tenha sido realizado, clique em “Cadastrar novo usu rio” e preencha todos os campos solicitados. Efetuado o *login*, o usu rio entrar  na tela inicial do SIMPREBAL mostrado na Figura E.3.

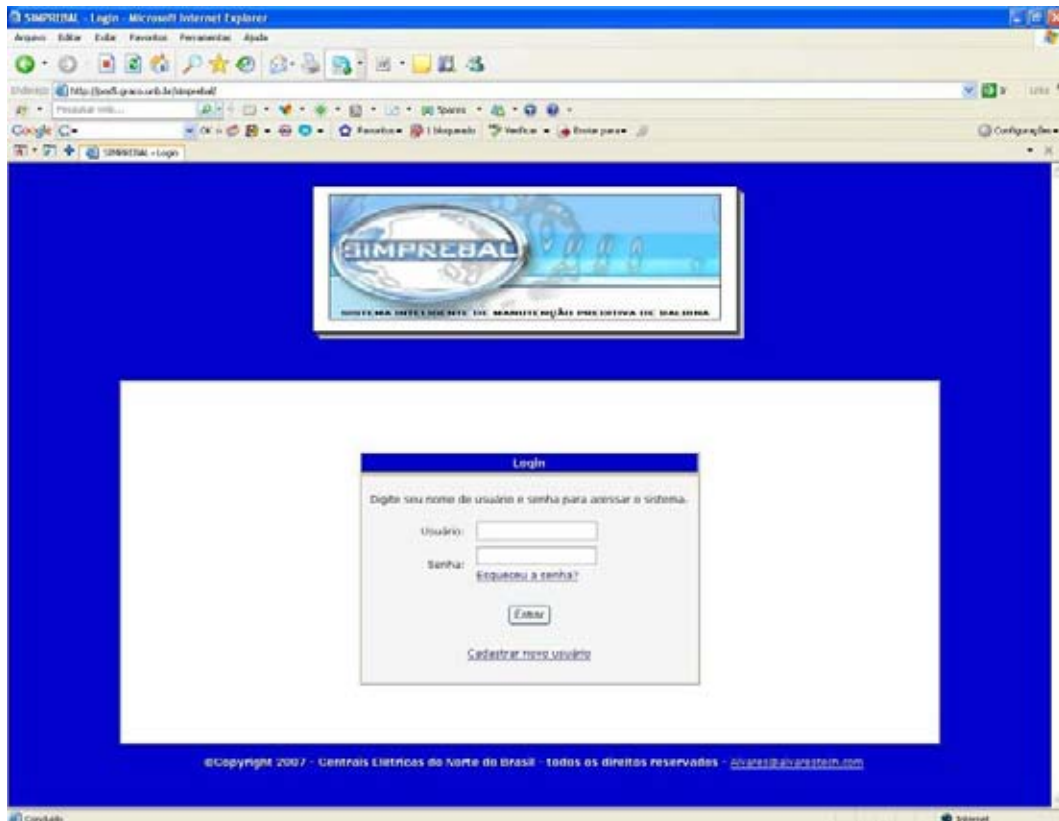


Figura E.2- Tela de *login*.



Figura E.3- Tela inicial.

A seguir descreve-se cada um dos itens do menu superior da tela apresentada na Figura E.3.

### E.2.1 Home

Este menu contém os seguintes submenus, à esquerda:

- Tags Monitoradas: Permite visualizar quais são as tags que atualmente estão sendo monitoradas pelo SIMPREBAL, e quais sistemas e equipamentos pertencem.
- Acessar Sinótico (Sistema de mancal, Sistema da turbina, Sistema do gerador): Permite acessar o *SimprenalClient*, se trata de um supervisor para mostrar as ocorrências de falhas no sistema, a variação das tags, em tempo real e através de históricos, a análise dos modos e efeitos de cada falha ocorrida e sugerir ordens de serviço.
- Acessar Sinótico (Trafos): Este link encontra-se atualmente vazio, mas corresponde uma expansão futura do sistema de transformadores.

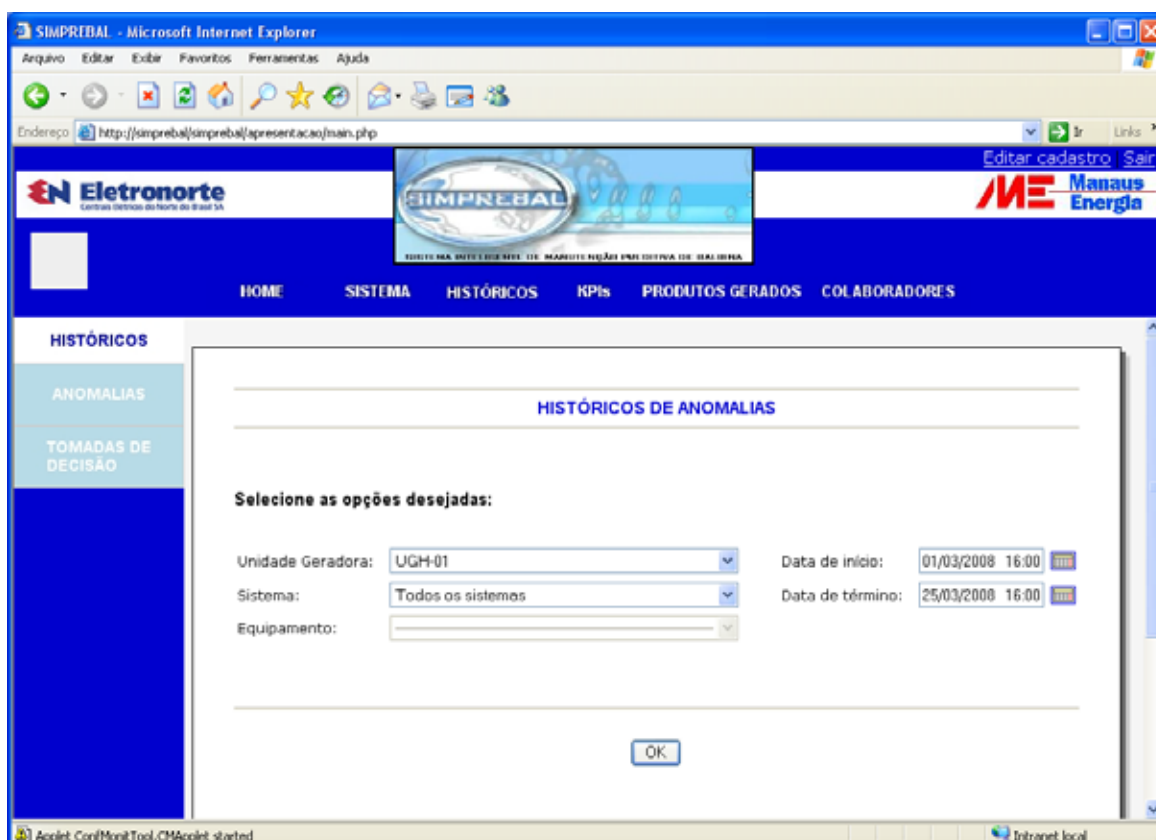


Figura E.4- Históricos de anomalias, seleção de equipamentos.

## E.2.2 Sistema

Descreve a metodologia usada no desenvolvimento do sistema. Contém o submenu arquitetura que detalha a arquitetura de concepção do SIMPREBAL (arquitetura de sete camadas do modelo OSA-CBM).

The screenshot shows the SIMPREBAL web application interface. The main content area is titled 'HISTÓRICOS DE ANOMALIAS'. It displays the following information:

Unidade Geradora: UGH-01  
Sistema: Todos os sistemas  
Equipamento: Todos os equipamentos

Sistema	Equipamento	ID anomalia	Descrição	Causa	Data de início	Data de término
Sistema de medicao	Servidor OPC	1283	Falha ao conectar com o servidor OPC	O cabo da rede esta desconectado	2008-03-25 08:34:38	

Below the table, there is a modal window titled 'EDITAR DATA DE TÉRMINO' with the following fields:

- Digite o número da UGH:
- Digite o ID Anomalia:
- Data de término:
- Enviar informações

Figura E.5- Históricos de anomalias.

## E.2.3 Históricos

Mostra os históricos de anomalias e de tomadas de decisão, tanto dos equipamentos da usina (mancais, geradores, turbinas) quanto do sistema de medição (dispositivos, instrumentação *fieldbus*, servidor OP e servidor SIMPREBAL). Acessando os submenus de anomalias ou decisões surge uma tela (Figura E.4) para seleção do equipamento, do sistema, da unidade geradora e do período de tempo desejado, sendo que é possível visualizar tanto as anomalias ou decisões de um equipamento específico quanto de todos os equipamentos de um determinado sistema ou de todos os sistemas de uma determinada



unidade geradora, de um único equipamento de todas as unidades geradoras ou de todos os equipamentos de todas as unidades geradoras da usina.

Selecionada a opção desejada, os históricos poderão ser visualizados clicando-se no botão OK. A tela da Figura E.5 permite a visualização de um exemplo de históricos de anomalias registrados no banco de dados. As informações de históricos são fornecidas em formato de tabela contendo os campos ID Anomalia, descrição da anomalia, causa, data de início e data de término. O ID Anomalia é a chave primária que identifica uma anomalia específica no banco de dados.

Além de mostrar os históricos de anomalias, a tela fornece também a opção de editar uma data de término de falha. É importante que as falhas estejam com as datas de término preenchidas para que seja possível calcular indicadores de desempenho tais como tempo médio entre falhas, tempo médio de reparo e taxa de falha.

The screenshot displays a software interface titled "EDITAR DATA DE TÉRMINO". On the left, there is a calendar for "Marco, 2008" with the date "25" highlighted. The main area contains the following information:

- o número da UGH:
- Digite o ID Anomalia:
- Descrição: Alta densidade do óleo no mancal combinado
- DT302-1 - ALARME
- Data de início: 2008-03-18 18:22:22
- Data de término:
- Hour:  :
- Day:
- Button:

Figura E.6- Edição da data de término de uma anomalia.

Para editar a data de término de uma determinada falha é necessário digitar o número da unidade geradora e o ID da anomalia nos respectivos campos de edição. Digitando um número de UGH e um ID válido, aparecerão informações sobre a falha cuja data de término se deseja editar no quadro logo abaixo.

A data de término então poderá ser editada selecionando-se a data em que a referida falha parou de acontecer, conforme mostrado na Figura E.6. Após selecionar a data de término,

deve-se clicar no botão Enviar informações para atualizar a nova data de término da falha no banco de dados.

#### **E.2.4 KPIs**

KPIs (*Key Performance Indicators*) são indicadores-chave de desempenho, ou seja, indicadores que refletem o progresso da usina em direção às suas metas organizacionais. A Figura E.7 mostra uma tela de cálculo dos KPIs para o sistema de resfriamento e lubrificação do mancal combinado. O menu KPIs contém o submenu Calcular KPIs, que calcula os KPIs para os equipamentos selecionados durante o intervalo de tempo selecionado. Os nove KPIs são os seguintes:

Para cada uma das tags de um referido equipamento:

1. Número de ocorrências de ALERTA;
2. Número de ocorrências de ALARME;
3. Número de ocorrências de TRIP;

Para cada equipamento em si:

4. Número de ocorrências de falhas;
5. Taxa de falhas;
6. Tempo médio entre falhas;
7. Tempo médio para reparo;
8. Número de prioridade de risco (fator de criticidade do equipamento);

E para o SIMPREBAL:

9. Porcentagem de decisões acertadas com relação às falhas do equipamento escolhido.

Uma vez que uma ALERTA não sinaliza uma falha, mas sim um defeito, o número de ocorrências de falhas consiste no somatório da quantidade de verificações de ocorrências de ALARMES ou TRIPS em cada uma das tags ou relações entre tags que permitem reconhecer uma falha, ou seja, é o somatório dos modos de ALARMES ou TRIPS para um determinado equipamento.

A taxa média de falhas é calculada usando a Equação E.1. O tempo médio entre falhas é definido como o inverso da taxa média de falhas.

$$\lambda = \frac{\text{n}^\circ \text{ de falhas do componente no período considerado}}{\text{tempo total que o componente ficou operando}} \quad (\text{E.1})$$

O tempo médio para reparo (MTTR) é calculado usando a Equação E.2.

$$\text{MTTR} = \frac{\text{tempo total que o componente ficou em reparo}}{\text{n}^\circ \text{ de reparos do componente no período considerado}} \quad (\text{E.2})$$

O número de prioridade de risco (NPR) de uma falha é calculado usando a Equação E.3. O NPR é utilizado para a priorização da tomada de ação. É uma maneira prática de priorizar certas falhas e avaliar quais providências devem ser tomadas primeiramente. Vale ressaltar que os índices de ocorrência, severidade e detecção mostrados no Apêndice D foram utilizados para avaliar a importância das falhas funcionais e, portanto, atribuídos a cada uma das falhas funcionais de cada componente, enquanto o NPR foi utilizado para avaliar os próprios componentes. Conseqüentemente, nos componentes que apresentam mais de uma falha funcional, o NPR do componente foi definido como sendo igual ao maior NPR de suas falhas funcionais.

$$\text{NPR} = \text{Severidade} \times \text{Ocorrência} \times \text{Detecção} \quad (\text{E.3})$$

O fator de criticidade de um equipamento corresponde ao maior NPR encontrado nas falhas deste mesmo equipamento dentro do intervalo de tempo especificado.

A confiabilidade ou porcentagem de acertos do SIMPREBAL corresponde à porcentagem de falhas que foram verificadas por um operador e assinaladas como corretas, isto é, que correspondem à realidade e é calculada usando a Equação E.4.

$$\% \text{Acertos} = \frac{\text{n}^\circ \text{ de falhas acertadas}}{\text{n}^\circ \text{ total de falhas detectadas}} \quad (\text{E.4})$$

### **E.2.5 Produtos gerados**

Este menu relata todos os documentos gerados em função do projeto de P&D Modernização da área de automação de processos da usina hidrelétrica de Balbina, que resultou no desenvolvimento do SIMPREBAL. Os submenus à esquerda contêm relatórios de pesquisa, artigos publicados, manuais de manutenção e operação do sistema, e os cursos ministrados.

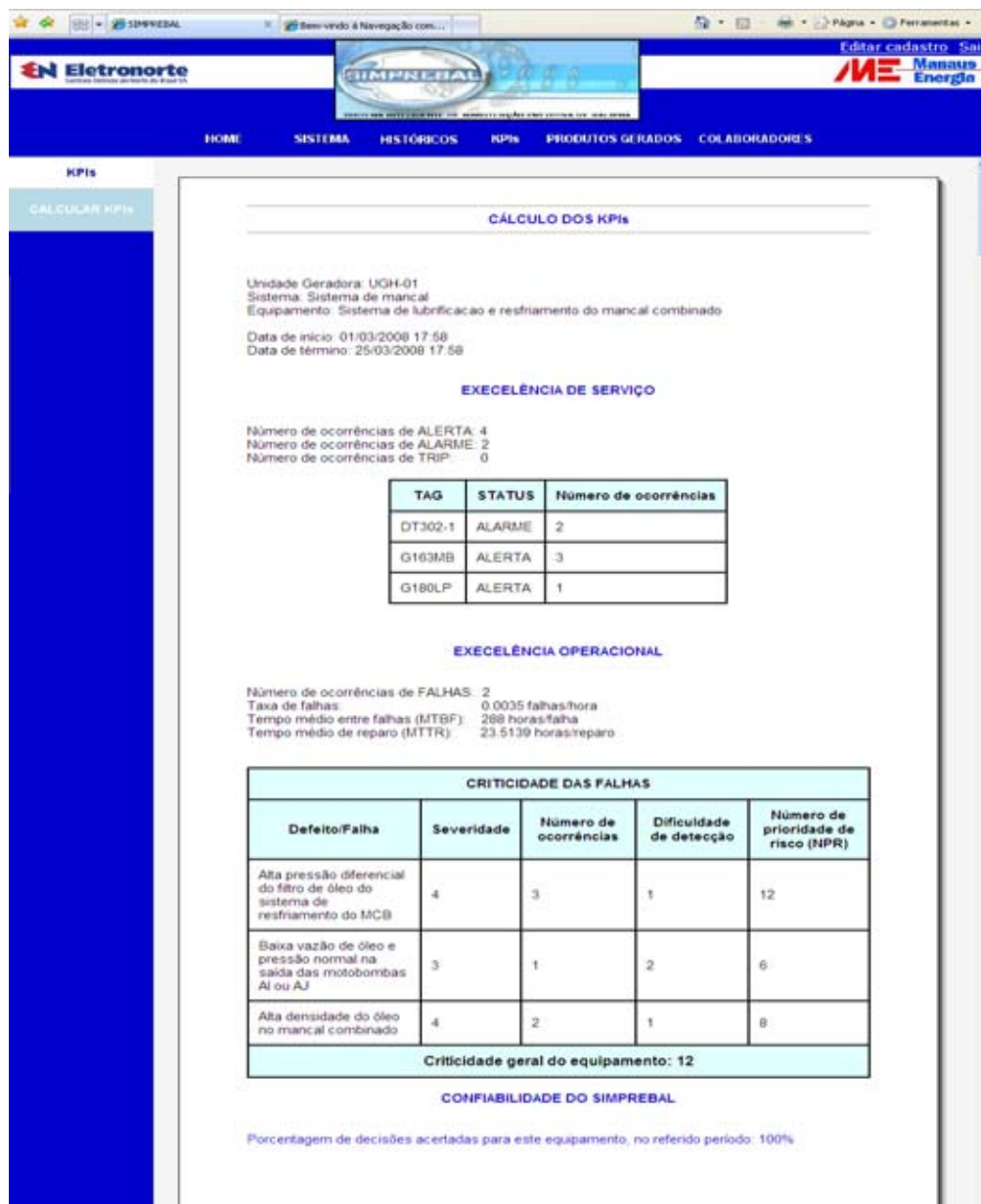


Figura E.7- Cálculo dos KPIs.

### E.2.6 Colaboradores

Este menu contém os contatos da equipe de desenvolvedores do SIMPREBAL, bem como a identificação dos colaboradores *online*, além de registros dos colaboradores que acessaram o sistema nos últimos 30 (trinta) dias.

### E.2.7 Editar Cadastro

No canto superior esquerdo da tela há uma opção para o usuário editar seu cadastro, podendo, portanto, alterar nome de usuário, nome, sobrenome, email, cargo e lotação.

### E.2.8 Sinótico SIMPREBAL (cliente *Applet*)

Ao clicar em acessar sinótico de mancal, turbina e gerador o usuário encontra uma tela semelhante à mostrada na Figura E.8. Nesta tela podem-se observar as figuras referentes às 5 (cinco) UGHs da usina. Em cada figura estão representados os equipamentos monitorados pelo SIMPREBAL. Os equipamentos que aparecem com a cor verde estão em funcionamento normal, os equipamentos com a cor amarela estão com defeito, ou seja, possuem valores das tags próximos aos valores de alarme (diz-se que estão em estado de alerta) e os equipamentos na cor vermelha (como é o caso do mancal guia da turbina da unidade geradora 1 da Figura E.8) estão em estado de alarme ou trip.

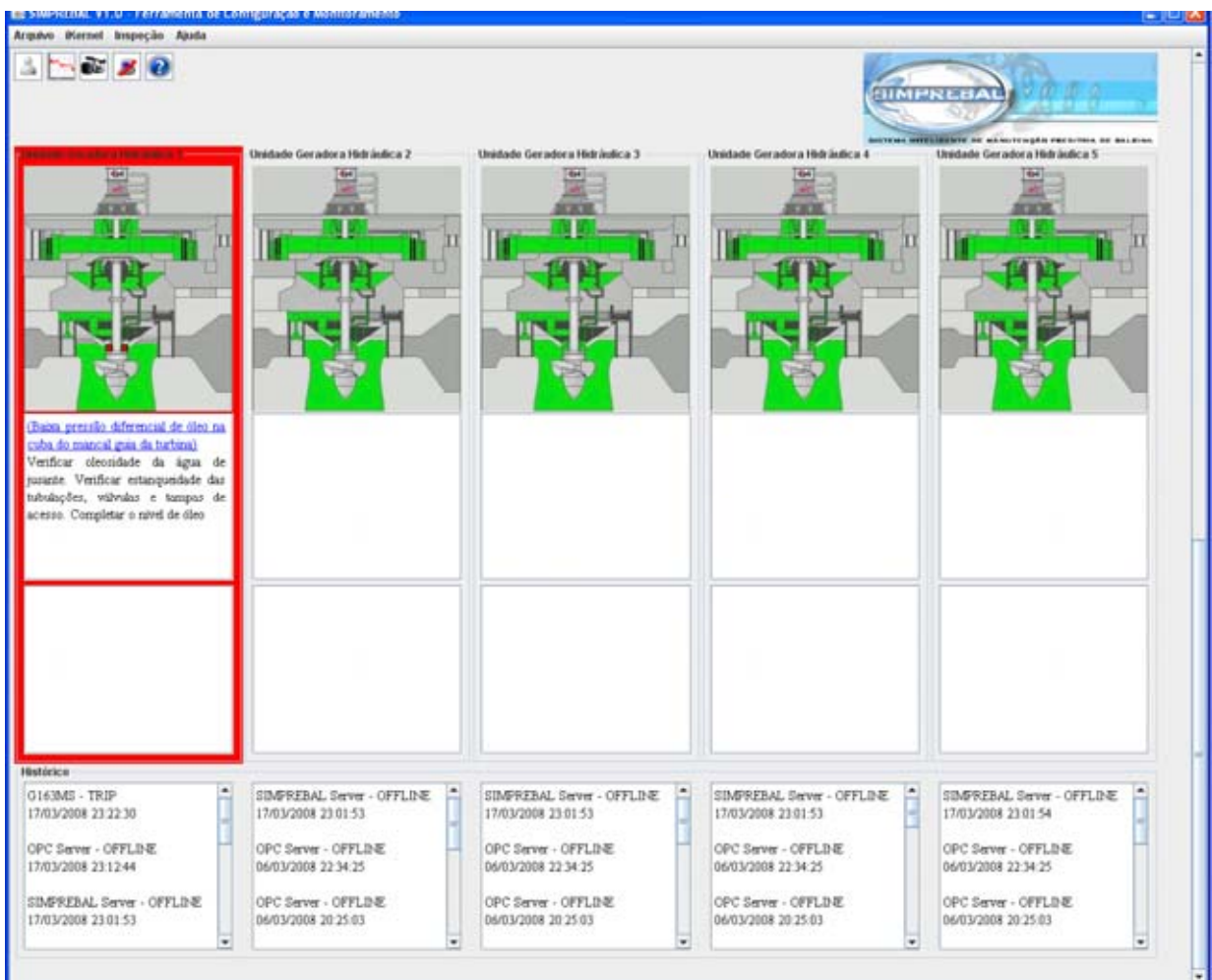


Figura E.8- Sinótico SIMPREBAL.

No primeiro quadrante abaixo da figura de cada UGH serão apresentadas as anomalias e as tomadas de decisão referentes aos equipamentos do sistema de mancal, sistema da turbina ou sistema do gerador. No segundo quadrante serão apresentadas as anomalias e as tomadas de decisão referentes ao sistema de medição, são, portanto, falhas de

processamento de sinal. E o terceiro e último quadrante disponibiliza um histórico das trinta últimas ocorrências de falhas ou defeitos em cada UGH. O primeiro e o segundo quadrante mostram a descrição da anomalia entre parênteses e sob a forma de um link, conforme observado na Figura E.9.

Clicando no link o usuário é direcionado para uma página HTML contendo a análise dos modos e efeitos da referida falha (FMEA). Os botões de atalho, situados na parte superior do SIMPREBAL, são, respectivamente, da esquerda para a direita:

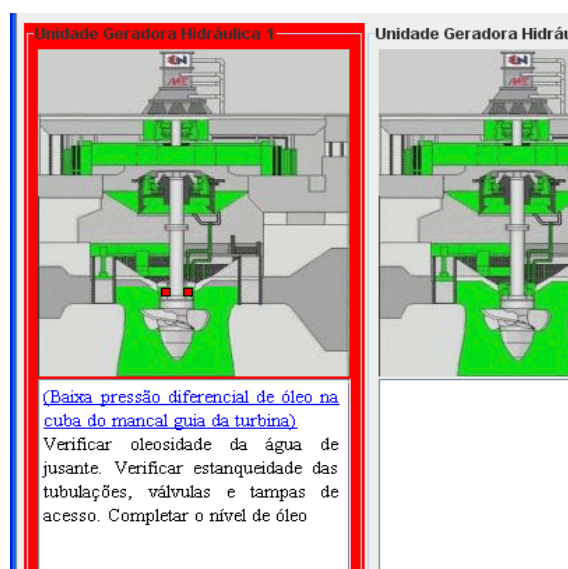


Figura E.9- Link para detalhamento da anomalia.

1. Botão Sair: Sair do sinótico.
2. Botão Inspeção de Variáveis: Inspeccionar as tags monitoradas através de visualização gráfica ou por acompanhamento de mudança de valor.
3. Botão Câmeras de segurança: Acessar as câmeras de segurança que eventualmente sejam instaladas.
4. Botão Shutdown: Desconectar o servidor efetuando a operação de shutdown (este procedimento só é possível mediante a digitação de uma senha).
5. Botão Ajuda: Acesso ao manual de operação.

### E.2.9 Inspeção De Variáveis

Esta função permite que o usuário visualize o valor das tags monitoradas pelo SIMPREBAL. Conforme mostrado na Figura E.10 ao selecionar a opção Inspeção de Variáveis o SIMPREBAL abrirá uma janela, com o título “Escolha uma Tag”, na qual será

apresentada uma árvore com todas as 5 UGHs, os sistemas monitorados, seus equipamentos e por fim as tags de cada um deles. Para acessar os valores das variáveis escolha uma tag e dê um duplo clique nela. Aparecerá uma janela com o título Inspeção de variáveis. Nesta janela o usuário poderá acompanhar a variação dos valores das tags escolhidas, bem como verificar a qualidade do sinal monitorado.

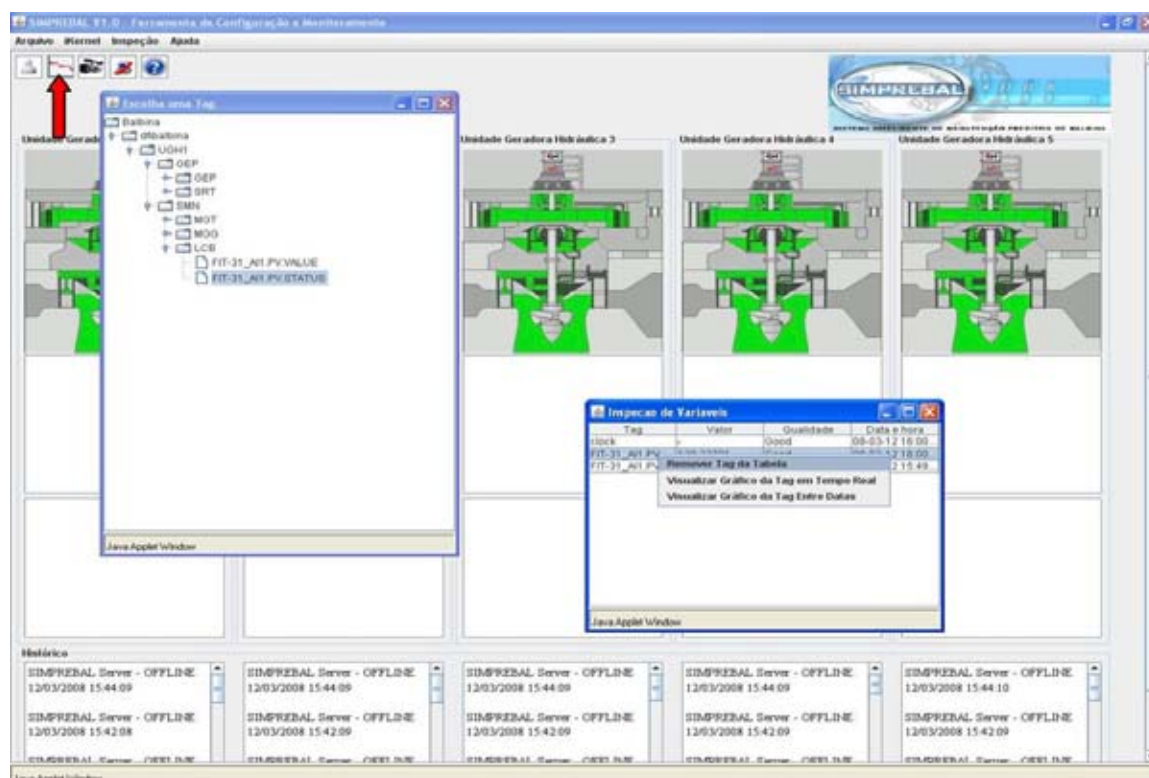


Figura E.10- Inspeção de variáveis.

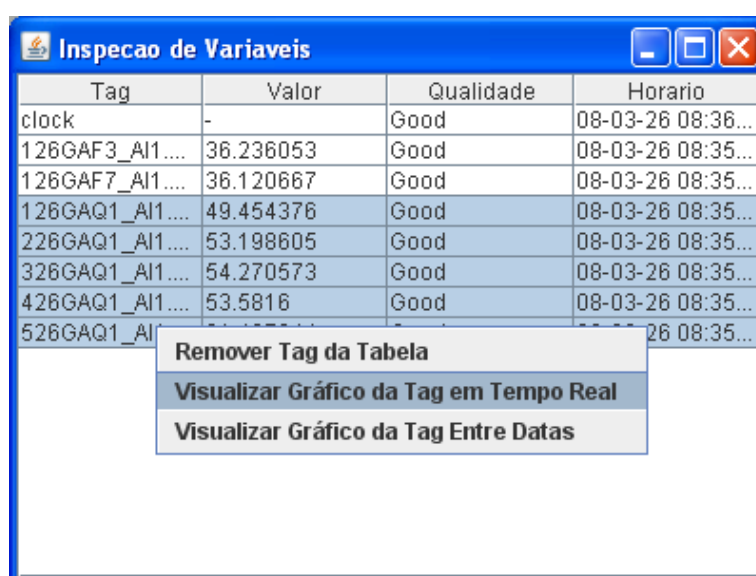


Figura E.11- Menu da tela de inspeção de variáveis.

Se o usuário clicar com o botão direito do mouse em uma ou mais destas variáveis, aparecerá um menu com as opções remover tag da tabela, visualizar gráfico da tag em tempo real e visualizar gráfico da tag entre datas. Para selecionar o gráfico de mais de uma variável deve-se selecionar as variáveis desejadas mantendo a tecla Ctrl pressionada, conforme mostrado na Figura E.11.

### E.2.10 Visualizar gráfico em tempo real

A opção Visualizar Gráfico da Tag em Tempo Real permite monitorar a variação de uma ou mais tags a partir do instante em que se clica nesta opção. Conforme mostrado na Figura E.12.

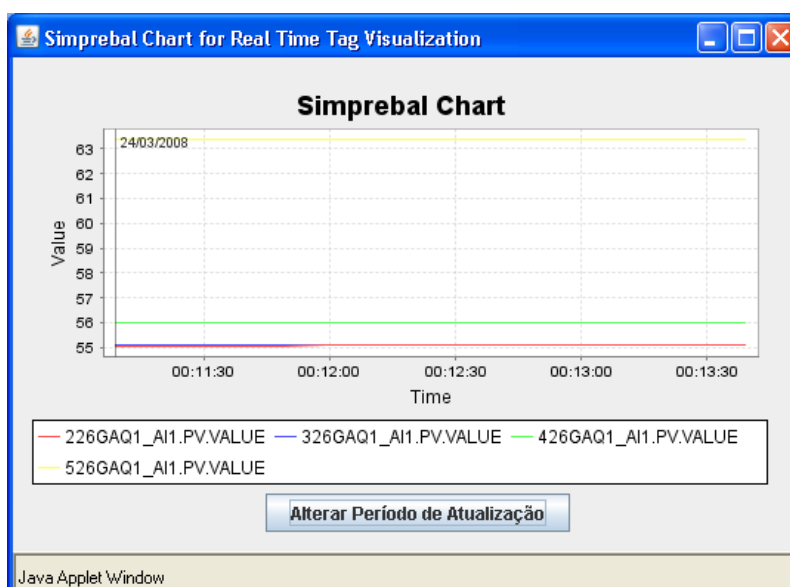


Figura E.12- Gráfico em tempo real da temperatura de ar quente do radiador das 5 unidades geradoras.

Figura E.13- Seleção do intervalo de aquisição dos dados históricos.



### E.2.11 Visualizar gráfico histórico

Esta opção permite visualizar o gráfico de valores de uma ou mais tags entre dois instantes escolhidos, datas de início e de término, conforme observado na Figura E.13.

Clicando no botão OK o SIMPREBAL apresentará um gráfico com a variação dos valores do(s) tag(s) selecionado(s) no intervalo de tempo determinado pelo usuário, conforme ilustrado na Figura E.14.

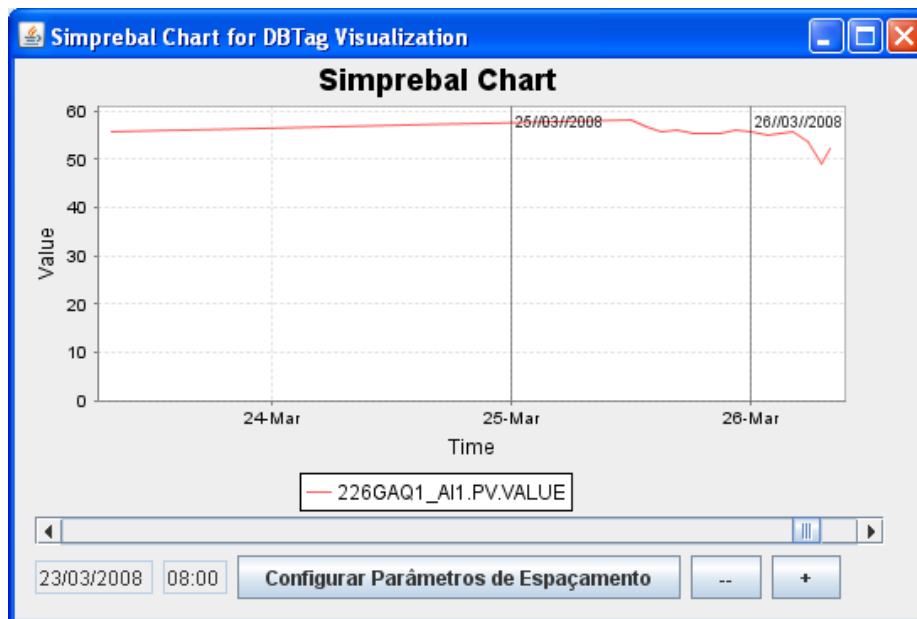


Figura E.14- Gráficos históricos