

**UNIVERSIDAD NACIONAL DE INGENIERÍA**  
**FACULTAD DE INGENIERÍA MECÁNICA**



**APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL EN LA IMPLEMENTACIÓN  
DE UN SISTEMA DE MANTENIMIENTO BASADO EN CONDICION PARA  
UNA CENTRAL HIDROELECTRICA**

**TESIS**

**PARA OPTAR EL TITULO PROFESIONAL PROFESIONAL DE**

**INGENIERO MECATRONICO**

**EDGAR JHONNY AMAYA SIMEON**

**PROMOCION 2002 – II**

**LIMA – PERU**

**2009**

Este trabajo es dedicado en primer lugar a Dios, a mis padres Crisanto y Maurelia por la educación ejemplar que me fue concedida, a mis hermanos por el apoyo y a todos mis amigos y colegas que contribuyeron para la realización de este sueño.

## **TABLA DE CONTENIDOS**

|  |           |
|--|-----------|
| <b>PROLOGO.....</b>                                    | <b>1</b>  |
| <b>CAPÍTULO 1: INTRODUCCIÓN.....</b>                   | <b>3</b>  |
| <b>1.1 OBJETIVOS.....</b>                              | <b>5</b>  |
| <b>1.2 FORMULACIÓN DEL PROBLEMA.....</b>               | <b>5</b>  |
| <b>1.3 ANTECEDENTES.....</b>                           | <b>6</b>  |
| <b>CAPÍTULO 2: MARCO CONCEPTUAL.....</b>               | <b>8</b>  |
| <b>2.1 DEFINICIÓN Y TIPOS DE MANTENIMIENTO.....</b>    | <b>8</b>  |
| <b>2.2 MANTENIMIENTO BASADO EN CONDICIÓN.....</b>      | <b>11</b> |
| 2.2.1 Sistemas MBC.....                                | 12        |
| <b>2.3 TECNOLOGÍAS DE COMUNICACIÓN INDUSTRIAL.....</b> | <b>12</b> |
| 2.3.1 Fieldbus.....                                    | 13        |
| 2.3.2 Tecnología OPC.....                              | 14        |
| <b>2.4 ARQUITECTURA OSA-CBM.....</b>                   | <b>19</b> |
| 2.4.1 Adquisición de datos.....                        | 22        |
| 2.4.2 Procesamiento de señal.....                      | 22        |
| 2.4.3 Monitoreo de condición.....                      | 23        |
| 2.4.4 Evaluación de salud.....                         | 24        |
| 2.4.5 Pronósticos.....                                 | 24        |
| 2.4.6 Toma de decisión.....                            | 26        |
| 2.4.7 Presentación.....                                | 27        |
| <b>2.5 SISTEMAS ESPECIALISTAS.....</b>                 | <b>27</b> |
| 2.5.1 Arquitectura.....                                | 28        |
| 2.5.2 Herramientas.....                                | 30        |
| 2.5.3 Reglas de producción.....                        | 31        |
| <b>CAPÍTULO 3: METODOLOGÍA.....</b>                    | <b>33</b> |
| <b>3.1 CONSIDERACIONES DEL MODELO OSA-CBM.....</b>     | <b>33</b> |
| 3.1.1 Adquisición de datos.....                        | 35        |
| 3.1.2 Procesamiento de señal.....                      | 36        |
| 3.1.3 Monitoreo de condición.....                      | 36        |
| 3.1.4 Evaluación de salud.....                         | 37        |
| 3.1.5 Pronóstico.....                                  | 37        |

|  |   |           |
|--|---|-----------|
| 3.1.6  | Toma de decisión.....                                     | 38        |
| 3.1.7  | Presentación.....   | 39        |
| <b>3.2</b>   | <b>MODELO FUNCIONAL IDEF0.....</b>                        | <b>39</b> |
| 3.2.1  | Actividad <i>I-kernel</i> .....                           | 41        |
| 3.2.2  | Actividad cliente <i>web</i> .....                        | 47        |
| 3.2.3  | Actividad Cliente Applet. ....                            | 48        |
| <b>3.3</b>   | <b>MODELO DE DATOS IDEF1X.....</b>                        | <b>49</b> |
| <b>CAPÍTULO 4: MODELO UML.....</b>                   |   | <b>53</b> |
| <b>4.1</b>   | <b>ARQUITECTURA DEL SISTEMA .....</b>                     | <b>53</b> |
| <b>4.2</b>   | <b>REQUISITOS DE USUARIO.....</b>                         | <b>54</b> |
| 4.2.1  | Requisitos funcionales.....                               | 55        |
| 4.2.2  | Requisitos no funcionales.....                            | 56        |
| <b>4.3</b>   | <b>REQUISITOS DEL SISTEMA .....</b>                       | <b>57</b> |
| <b>4.4</b>   | <b>CASOS DE USO DE LA APLICACIÓN <i>I-KERNEL</i>.....</b> | <b>58</b> |
| 4.4.1  | Iniciación del <i>I-kernel</i> .....                      | 58        |
| 4.4.2  | Procesamiento inteligente.....                            | 59        |
| 4.4.3  | Verificación de alarmas y alertas.....                    | 60        |
| 4.4.4  | Shutdown del <i>I-kernel</i> . ....                       | 61        |
| <b>4.5</b>   | <b>CASOS DE USO DE LA HERRAMIENTA DE C&amp;M.....</b>     | <b>62</b> |
| 4.5.1  | Iniciación de la herramienta C&M .....                    | 62        |
| 4.5.2  | Monitoreo de sinóptico .....                              | 63        |
| 4.5.3  | Actualización de sinóptico.....                           | 63        |
| 4.5.4  | Inspección de variables.....                              | 64        |
| 4.5.5  | Inspección de variables <i>online</i> .....               | 65        |
| 4.5.6  | Inspección de variables históricas.....                   | 66        |
| 4.5.7  | Shutdown de la herramienta C&M .....                      | 66        |
| <b>CAPÍTULO 5: IMPLEMENTACIÓN COMPUTACIONAL.....</b> |   | <b>68</b> |
| <b>5.1</b>   | <b>REQUISITOS FÍSICOS.....</b>                            | <b>68</b> |
| <b>5.2</b>   | <b>ESTRUCTURA DE REGLAS DE PRODUCCIÓN.....</b>            | <b>71</b> |
| 5.2.1  | Estructura de reglas de procesamiento de señal.....       | 71        |
| 5.2.2  | Estructura de reglas de monitoreo de condición .....      | 73        |
| 5.2.3  | Estructura de reglas de evaluación de salud .....         | 74        |
| 5.2.4  | Estructura de reglas de la toma de decisión .....         | 76        |
| <b>5.3</b>   | <b>CLASES IMPLEMENTADAS EN EL SIMPREBAL .....</b>         | <b>77</b> |
| 5.3.1  | Clases del <i>I-Kernel</i> .....                          | 77        |
| 5.3.2  | Clases del <i>Corfmonitool</i> .....                      | 79        |

**CAPÍTULO 6: APLICACIÓN EN UNA CENTRAL HIDROELÉCTRICA...81**

**6.1 CENTRAL HIDROELÉCTRICA Y SUS SISTEMAS ..... 81**

6.1.1 Sistema mancal..... 83

6.1.2 Sistema de la turbina..... 84

6.1.3 Sistema generador..... 86

**6.2 MÉTODO DE ANÁLISIS..... 87**

**6.3 RESULTADOS Y DISCUSIONES ..... 88**

6.3.1 Análisis de la fallas..... 88

6.3.2 Ejemplo: Sistema de Enfriamiento del Generador..... 92

6.3.3 Análisis Comparativo ..... 94

**CAPÍTULO 7: APORTES PARA APLICACIONES ACADÉMICAS.....96**

**7.1 ADQUISICION DE DATOS..... 96**

**7.2 BANCO DE DATOS..... 97**

**7.3 AMBIENTE DE DESARROLLO JAVA..... 97**

**7.4 SISTEMAS ESPECIALISTAS..... 98**

**CONCLUSIONES Y RECOMENDACIONES.....99**

**CONCLUSIONES..... 99**

**RECOMENDACIONES..... 100**

**BIBLIOGRAFÍA.....102**

**APÉNDICE A: ARCHIVOS DE CONFIGURACIÓN.....111**

**A.1 ARCHIVO DE CONFIGURACION ..... 111**

A.1.1 Configuración general..... 111

A.1.2 Servidores OPC y tags OPC..... 113

A.1.3 Servidores de banco de datos ..... 114

A.1.4 Tags simuladas ..... 115

A.1.5 Dispositivos DFI ..... 115

A.1.6 Servidor de email ..... 116

**A.2 ARCHIVO CÓDIGOS DE FALLA..... 116**

**A.3 ARCHIVO CÓDIGOS DE DECISIÓN..... 117**

**APÉNDICE B: CÁLCULO DE LA CLASE TAG.....119**

**B.1 PROCESAMIENTO DEL ÍTEM *VALUE*..... 119**

**B.2 PROCESAMIENTO DEL ÍTEM STATUS ..... 121**

|  |            |
|--|------------|
| <b>APÉNDICE C: MÉTODOS DE ANÁLISIS FMEA Y FTA.....</b>           | <b>123</b> |
| <b>C.1 ANÁLISIS DE MODOS DE FALLAS Y SUS EFECTOS - FMEA ....</b> | <b>123</b> |
| <b>C.2 ANÁLISIS DE ÁRBOL DE FALLAS - FTA.....</b>                | <b>127</b> |
| <br>   |            |
| <b>APÉNDICE D: INICIACIÓN Y OPERACIÓN DEL SIMPREBAL.....</b>     | <b>129</b> |
| <b>D.1 SERVIDOR SIMPREBAL .....</b>                              | <b>129</b> |
| <b>D.2 CLIENTE SIMPREBAL .....</b>                               | <b>130</b> |
| D.2.1 Home.....  | 131        |
| D.2.2 Sistema.....   | 132        |
| D.2.3 Históricos .....   | 133        |
| D.2.4 KPIs.....  | 135        |
| D.2.5 Productos generados.....                                   | 138        |
| D.2.6 Colaboradores .....  | 139        |
| D.2.7 Editar Registro .....                                      | 139        |
| D.2.8 Sinóptico SIMPREBAL .....                                  | 139        |
| D.2.9 Inspección de variables .....                              | 142        |
| D.2.10 Visualizar gráfico en tiempo real.....                    | 143        |
| D.2.11 Visualizar gráfico histórico .....                        | 145        |

## LISTA DE TABLAS

|   |     |
|---|-----|
| TABLA 2.1- PARADIGMAS DE MANTENIMIENTO (MOUBRAY 1999, MODIFICADO). .....                        | 10  |
| TABLA 2.2- DIFERENCIAS ENTRE UN SE Y UN SISTEMA CONVENCIONAL (BAUCHSPIESS<br>2004). .....       | 28  |
| TABLA 4.1- REQUISITOS FUNCIONALES (RFS) (AMAYA ET AL. 2007A). .....                             | 55  |
| TABLA 4.2- REQUISITOS NO FUNCIONALES (RNFs) (AMAYA ET AL. 2007A). .....                         | 56  |
| TABLA 5.1- MODELO DE REGLAS PARA PROCESAMIENTO DE SEÑAL OPC. ....                               | 71  |
| TABLA 5.2- MODELO DE REGLAS PARA PROCESAMIENTO DE SEÑAL <i>FIELD</i> BUS. ....                  | 72  |
| TABLA 5.3- MODELO DE REGLAS PARA LOS RANGOS DE OPERACIÓN.....                                   | 73  |
| TABLA 5.4- MODELO DE REGLAS PARA MONITOREO DE CONDICIÓN.....                                    | 73  |
| TABLA 5.5- MODELO DE REGLAS PARA DIAGNÓSTICO DE CANALES <i>FIELD</i> BUS.....                   | 74  |
| TABLA 5.6- MODELO DE REGLAS PARA DIAGNÓSTICO DE LA INSTRUMENTACIÓN<br><i>FIELD</i> BUS. ....    | 75  |
| TABLA 5.7- MODELO DE REGLAS PARA DIAGNÓSTICO DE MONITOREO DE CONDICIÓN. 75                      |     |
| TABLA 5.8- MODELO DE REGLAS PARA TOMA DE DECISIÓN. ....   | 76  |
| TABLA 6.1- INSTRUMENTACIÓN DEL SISTEMA DE MANCAL. ....  | 84  |
| TABLA 6.2- INSTRUMENTACIÓN DEL SISTEMA DE LA TURBINA. ....                                      | 85  |
| TABLA 6.3- INSTRUMENTOS DEL SISTEMA GENERADOR. ....   | 86  |
| TABLA 6.4- EJEMPLOS DE REGISTROS DE OTM. ....   | 87  |
| TABLA 6.5- CUADRO COMPARATIVO ENTRE UN OPERADOR Y EL SE. ....                                   | 95  |
| TABLA B.1- RELACIÓN ENTRE EL <i>QUALITY</i> Y VALORES <i>QUALITY</i> Y <i>SUBQUALITY</i> . .... | 120 |
| TABLA B.2- VALORES DE <i>SUBSTATUS</i> Y <i>QUALITY</i> (AMAYA ET AL., 2007C). ....             | 122 |
| TABLA C.1- FORMULARIO ESTANDARIZADO DE ANÁLISIS FMEA. ....                                      | 123 |
| TABLA C.2- SIGNIFICADO DE LOS ÍNDICES DE SEGURIDAD O MEDIO AMBIENTE.....                        | 125 |
| TABLA C.3- SIGNIFICADO DE LOS ÍNDICES DE PÉRDIDA DE FACTURAMIENTO. ....                         | 125 |
| TABLA C.4- SIGNIFICADO DE LOS ÍNDICES DE CORTE DE CARGA. ....                                   | 126 |
| TABLA C.5- SIGNIFICADO DE LOS ÍNDICES DE SEVERIDAD.....   | 126 |
| TABLA C.6- SIGNIFICADO DE LOS ÍNDICES DE OCURRENCIA DE FALLAS. ....                             | 127 |
| TABLA C.7- SIGNIFICADO DOS ÍNDICES DE DETECCIÓN. ....   | 127 |

## LISTA DE FIGURAS

|   |    |
|---|----|
| FIGURA 1.1- ARQUITECTURA OSA-CBM .....  | 3  |
| FIGURA 2.1- DIFERENTES TIPOS DE MANTENIMIENTO (BARROSO MAIA JUNIOR, 2003). 8        |    |
| FIGURA 2.2- REDES <i>FIELDBUS</i> H1 Y HSE. ....                                    | 14 |
| FIGURA 2.3- ARQUITECTURA DE INFORMACIÓN EN CONTROL DE PROCESOS.....                 | 16 |
| FIGURA 2.4- APLICACIONES CON VARIOS SERVIDORES OPC. ....                            | 16 |
| FIGURA 2.5- RELACIÓN ENTRE CLIENTES Y SERVIDORES.....                               | 17 |
| FIGURA 2.6- ARQUITECTURA OPC.....   | 18 |
| FIGURA 2.7- CAMADAS FUNCIONALES OSA-CBM (LEBOLD Y THURSTON 2001).....               | 20 |
| FIGURA 2.8- PADRONES A USAR EN LAS CAMADAS OSA-CBM (BENGTSSON 2004A). 24            |    |
| FIGURA 2.9- COMPONENTES DE LA CAMADA PRONÓSTICO (LEBOLD Y THURSTON<br>2001). ....   | 25 |
| FIGURA 2.10- ENTRADAS Y SALIDAS DE LA CAMADA PRONÓSTICO. ....                       | 26 |
| FIGURA 2.11- ESTRUCTURA DE UN SE BASADO EN REGLAS DE PRODUCCIÓN. ....               | 29 |
| FIGURA 2.12- INTERACCIÓN ENTRE EL IC Y EL ESPECIALISTA (WATERMAN, 1986). ...        | 29 |
| FIGURA 3.1- ARQUITECTURA CONCEPTUAL DEL SISTEMA MBC.....                            | 34 |
| FIGURA 3.2- RANGO DE OPERACIÓN DE UN EQUIPO. ....                                   | 36 |
| FIGURA 3.3- ESTRUCTURA DE UNA FUNCIÓN Y SUS DETALLES .....                          | 40 |
| FIGURA 3.4- DIAGRAMA A0 DEL SISTEMA INTELIGENTE MBC.....                            | 42 |
| FIGURA 3.5- DIAGRAMA DETALLADO DE LA FUNCIÓN A0. ....                               | 43 |
| FIGURA 3.6- DIAGRAMA DETALLADO DE LA FUNCIÓN A1. ....                               | 44 |
| FIGURA 3.7- DIAGRAMA DETALLADO DE LA FUNCIÓN A2. ....                               | 45 |
| FIGURA 3.8- DIAGRAMA DERIVADO DE LA FUNCIÓN A21.....                                | 46 |
| FIGURA 3.9- MODELO DE INFORMACIÓN IDEF1X DE CONFIGURACIÓN.....                      | 50 |
| FIGURA 3.10- MODELO DE INFORMACIÓN IDEF1X DEL BANCO DE DATOS. ....                  | 51 |
| FIGURA 4.1- PAQUETES DE LA ARQUITECTURA SIMPREBAL. ....                             | 53 |
| FIGURA 4.2- EL <i>I-KERNEL</i> Y SU INTERACCIÓN (AMAYA <i>ET AL.</i> , 2007C). .... | 54 |
| FIGURA 4.3- CASO DE USO DEL SISTEMA INTELIGENTE.....                                | 57 |
| FIGURA 4.4- INICIACIÓN DE LA APLICACIÓN <i>I-KERNEL</i> .....                       | 59 |
| FIGURA 4.5- PROCESAMIENTO INTELIGENTE. ....   | 60 |
| FIGURA 4.6- VERIFICACIÓN DE ALARMAS Y ALERTAS. ....                                 | 61 |

|  |     |
|--|-----|
| FIGURA 4.7- <i>SHUTDOWN</i> DE LA APLICACIÓN <i>I-KERNEL</i> .                 | 61  |
| FIGURA 4.8- INICIACIÓN DE LA HERRAMIENTA DE C&M.                               | 62  |
| FIGURA 4.9- MONITOREO DE SINÓPTICO.  | 63  |
| FIGURA 4.10- ACTUALIZACIÓN DE SINÓPTICO.                                       | 64  |
| FIGURA 4.11- INSPECCIÓN DE VARIABLES.  | 64  |
| FIGURA 4.12- INSPECCIÓN DE VARIABLES <i>ONLINE</i> .                           | 65  |
| FIGURA 4.13- INSPECCIÓN DE VARIABLES HISTÓRICAS.                               | 66  |
| FIGURA 4.14- <i>SHUTDOWN</i> DE LA HERRAMIENTA C&M.                            | 67  |
| FIGURA 5.1- INSTRUMENTACIÓN Y EQUIPOS.   | 69  |
| FIGURA 5.2- REQUISITOS FÍSICOS DEL SIMPREBAL.                                  | 70  |
| FIGURA 5.3- CLASES DEL <i>I-KERNEL</i> .                                       | 78  |
| FIGURA 5.4- CLASES DEL <i>CONFMONITTOOL</i> .                                  | 79  |
| FIGURA 6.1- VISTA SUPERIOR DE LOS 5 GENERADORES.                               | 82  |
| FIGURA 6.2- ILUSTRACIÓN DEL GRUPO TURBINA GENERADOR.                           | 82  |
| FIGURA 6.3- EJE Y MANCAL GUÍA DE LA TURBINA.                                   | 83  |
| FIGURA 6.4- SISTEMA DE REGULACIÓN DE VELOCIDAD.                                | 85  |
| FIGURA 6.5- ANOMALÍAS DETECTAS.  | 89  |
| FIGURA 6.6- OSCILACIONES DE UNA VARIABLE MONITOREADA ENTRE ALERTA Y<br>ALARMA. | 90  |
| FIGURA 6.7- CURVA DE HISTÉRESIS PARA MANIPULACIÓN DE VARIABLES.                | 91  |
| FIGURA 6.8- ZONA DE HISTÉRESIS DE MANIPULACIÓN DE VARIABLES.                   | 91  |
| FIGURA 6.9- INTERFACE MOSTRANDO LAS ANOMALÍAS EN TIEMPO REAL.                  | 92  |
| FIGURA 6.10- ANOMALÍAS ALMACENADAS EN EL BANCO DE DATOS.                       | 93  |
| FIGURA 6.11- EMAIL DE ANOMALÍAS ENVIADO POR EL SISTEMA.                        | 93  |
| FIGURA 6.12- SELECCIÓN E INSPECCIÓN DE VARIABLES.                              | 94  |
| FIGURA 6.13- TENDENCIA DE VARIABLES ASOCIADAS A ANOMALÍAS DEL SEG.             | 94  |
| FIGURA 6.14- FALLAS OCURRIDAS Y LAS DETECTADAS POR EL SISTEMA.                 | 95  |
| FIGURA A.1- PARÁMETROS DE CONFIGURACIÓN GENERAL.                               | 112 |
| FIGURA A.2- PARÁMETROS DEL SERVIDOR OPC Y TAGS ASOCIADAS.                      | 113 |
| FIGURA A.3- PARÁMETROS DEL SERVIDOR BANCO DE DATOS Y TAGS ASOCIADAS. ...       | 114 |
| FIGURA A.4- TAGS SIMULADAS.  | 115 |
| FIGURA A.5- PARÁMETROS DE LOS DISPOSITIVOS DFI.                                | 115 |

|   |     |
|---|-----|
| FIGURA A.6- PARÁMETROS DEL SERVIDOR DE E-MAIL. ....   | 116 |
| FIGURA A.7- ARCHIVO CÓDIGOS DE FALLA.....   | 117 |
| FIGURA A.8- ARCHIVO CÓDIGOS DE DECISIÓN. ....   | 118 |
| FIGURA B.1- PROCESO DE CÁLCULO DE LA CLASE TAG.....   | 120 |
| FIGURA B.2- ESTRUCTURA DE LA PROPIEDAD <i>VALUE</i> DEL ÍTEM <i>STATUS</i> (SMAR, 2005).<br>..... | 121 |
| FIGURA D.1- SERVIDOR EN EJECUCIÓN. ....   | 129 |
| FIGURA D.2- INTERFACE DE <i>LOGIN</i> .....   | 130 |
| FIGURA D.3- INTERFACE INICIAL. ....   | 131 |
| FIGURA D.4- HISTÓRICOS DE ANOMALÍAS Y SELECCIÓN DE EQUIPOS. ....                                  | 132 |
| FIGURA D.5- HISTÓRICOS DE ANOMALÍAS.....  | 133 |
| FIGURA D.6- EDICIÓN DE LA FECHA DE TÉRMINO DE UNA ANOMALÍA. ....                                  | 134 |
| FIGURA D.7- CÁLCULO DE LOS KPIs. ....   | 136 |
| FIGURA D.8- SINÓPTICO SIMPREBAL. ....   | 140 |
| FIGURA D.9- LINK PARA DETALLAMIENTO DE LA ANOMALÍA.....   | 141 |
| FIGURA D.10- INSPECCIÓN DE VARIABLES. ....  | 142 |
| FIGURA D.11- MENÚ DE LA INTERFACE INSPECCIÓN DE VARIABLES. ....                                   | 143 |
| FIGURA D.12- GRÁFICO EN TIEMPO REAL DE TEMPERATURAS.....  | 144 |
| FIGURA D.13- SELECCIÓN DEL INTERVALO DE ADQUISICIÓN DE LOS DATOS HISTÓRICOS.<br>.....             | 144 |
| FIGURA D.14- GRÁFICOS HISTÓRICOS .....  | 145 |

## LISTA DE SÍMBOLOS Y ABREVIACIONES

|      |  |
|------|--|
| DA   | – Data Access.                                       |
| DCOM | – Distributed Component Object Model.                |
| DCS  | – Distributed Control System.                        |
| DFI  | – Interface <i>fieldbus</i> Distribuída.             |
| FMEA | – Failure Modes and Effects Analysis.                |
| FF   | – Foundation Fieldbus.                               |
| FTA  | – Fault Tree Analysis.                               |
| GUI  | – Grafic User Interface.                             |
| HSE  | – High Speed Ethernet.                               |
| HTML | – HyperText Markup Language.                         |
| HTTP | – Hyper Text Transfer Protocol.                      |
| HMI  | – Human Machine Interface.                           |
| IA   | – Inteligencia Artificial.                           |
| IC   | – Ingeniero de Conocimiento.                         |
| IEEE | – Institute of Electrical and Electronics Engineers. |
| IP   | – Internet Protocol.                                 |
| ISO  | – International Standard Organization.               |
| JDBC | – Java Database Connectivity.                        |
| JESS | – Java Expert System Shell.                          |
| JNI  | – Java Native Interface.                             |
| LAN  | – Local Area Network.                                |
| MES  | – Manufacturing Execution Systems.                   |
| MGT  | –Mancal Guía de la Turbina.                          |
| MTBR | – Mean Time Between Repairs.                         |
| MTTR | – Mean Time To Repairs.                              |
| NPR  | – Número de Prioridad de Riesgo.                     |
| OLE  | – Object Linking and Embedding.                      |
| OPC  | – OLE for Process Control.                           |

|                  |  |
|------------------|--|
| <b>OSA-EAI</b>   | – Open System Architecture for Enterprise Application Integration. |
| <b>OSA-CBM</b>   | – Open System Architecture for Condition Based Maintenance.        |
| <b>OTM</b>       | – Orden de Trabajo de Mantenimiento.                               |
| <b>PHP</b>       | – Hypertext Preprocessor.  |
| <b>PLC</b>       | – Controlador Lógico Programable.                                  |
| <b>SCADA</b>     | – Supervisory Control and Data Acquisition.                        |
| <b>SE</b>        | – Sistema Especialista.  |
| <b>SEG</b>       | – Sistema de Enfriamiento del Generador.                           |
| <b>SG</b>        | – Sistema Generador.   |
| <b>SM</b>        | – Sistema Mancal.  |
| <b>ST</b>        | – Sistema Turbina.   |
| <b>SIMPREBAL</b> | – Sistema Inteligente de Mantenimiento Predictivo de Balbina.      |
| <b>TCP/IP</b>    | – Transmission Control Protocol / Internet Protocol.               |
| <b>TELNET</b>    | – TELEcommunication NETwork.                                       |
| <b>UGH</b>       | – Unidad Generadora Hidráulico.                                    |
| <b>UML</b>       | – Unified Modeling Language.                                       |

## PRÓLOGO

La presente tesis desarrolla la implementación de un sistema inteligente MBC (Mantenimiento Basado en Condición) usando Sistemas Especialistas. Para éste estudio, se ha organizado el trabajo en siete capítulos que serán expuestos de la siguiente forma:

En el **capítulo I**, se realiza una introducción a esta tesis, los objetivos, formulación del problema y los trabajos desarrollados por otros autores relacionados al tema.

En el **capítulo II** es presentada una revisión de literatura sobre Mantenimiento Basado en Condición, Sistemas de Mantenimiento Basado en Condición, el modelo de referencia OSA-CBM y su arquitectura. Además se presentan los conceptos, arquitecturas y herramientas para el desarrollo de Sistemas Especialistas.

El **capítulo III** presenta la metodología propuesta, la aplicación del modelo de referencia OSA-CBM y los modelos IDEF0 e IDEF1X.

En el **capítulo IV** se describe el modelo UML del sistema computacional, los requisitos de usuario y los requisitos del sistema representado en casos de uso.

El **capítulo V** presenta la implementación computacional del Sistema, los modelos de reglas de producción implementadas en cada una de las capas y las clases implementadas en el SIMPREBAL.

El **capítulo VI** es dedicado a mostrar los resultados obtenidos de la implementación en la central hidroeléctrica de Balbina en el Brasil.

En el **capítulo VII** se presentan los principales aportes de esta tesis para el desarrollo de aplicaciones académicas relacionados a la adquisición de datos, almacenamiento en banco de datos y procesamiento inteligente usando Sistemas Especialistas.

Finalmente, se presentan las **conclusiones y recomendaciones** para futuras investigaciones.

El **apéndice A** presenta la descripción de los archivos de configuración, de códigos de fallas y toma de decisión. En el **apéndice B** se muestra los cálculos matemáticos para hallar los valores de los campos de la clase *tag*. El **apéndice C** trata los conceptos de los métodos de análisis FMEA (*Failure Modes and Effects Analysis*) y FTA (*Fault Tree Analysis*) y finalmente el **apéndice D** describe la iniciación y operación del software implementado.

# CAPÍTULO 1

## INTRODUCCIÓN

La competitividad actual exige de las empresas la oferta de productos y servicios de calidad y bajo costo (Nunes, 2001). La diversidad y la cantidad de componentes utilizados en las instalaciones industriales tienden a favorecer la probabilidad de ocurrencia de fallas<sup>1</sup>. En vista de eso, las empresas vienen ampliando el uso de nuevas tecnologías y la automatización de máquinas y procesos.

Las empresas están optando por instalar sistemas de mantenimiento, tales como MBC (Mantenimiento Basado en Condición), donde las acciones son efectuadas después de cada inspección dependiendo del estado de las máquinas.

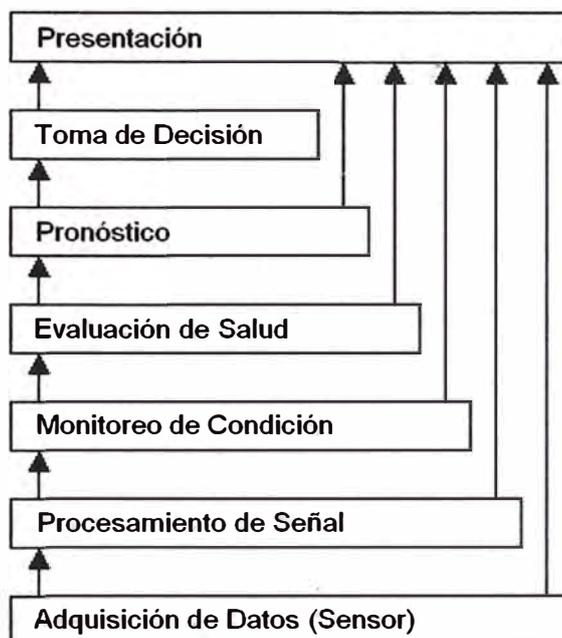


Figura 1.1- Arquitectura OSA-CBM

En esta tesis se propone una metodología para el desarrollo de un sistema MBC usando el modelo de referencia de siete capas OSA-CBM (*Open System Architecture for Condition Based Maintenance*) mostrada en la Figura 1.1.

La adquisición de datos del sistema es de dos formas: *online* vía servidor OPC e histórica a través de banco de datos. Estos datos son procesados a través de SE (Sistemas Especialistas) en las capas de procesamiento de señal, monitoreo de condición, evaluación de salud, pronóstico y toma de decisión. La capa presentación muestra las informaciones de todas las capas previas y permite visualizar gráficamente variables en tiempo real e histórico asociadas a la evolución de defectos<sup>2</sup> y fallas.

Los problemas de diagnóstico o pronóstico de defectos y fallas en plantas industriales envuelven correlaciones entre varias variables asociadas al proceso. Las variables son acompañadas temporalmente para análisis de tendencia y comparación con el desempeño esperado.

Los conceptos, algoritmos y herramientas usados en el desarrollo de esta tesis podrán ser usados en otras aplicaciones en el ámbito académico e industrial, que involucren adquisición y almacenamiento de datos, procesamiento inteligente, y diagnóstico de fallas en máquinas e instrumentos.

## **1.1 OBJETIVOS**

- El objetivo de esta tesis es proponer una metodología para el desarrollo de un sistema inteligente MBC usando SE, y su implementación computacional en java siguiendo un modelo cliente/servidor para el diagnóstico de fallas de máquinas e instrumentos a través del internet.
- Presentar los conceptos de técnicas de mantenimiento, la arquitectura OPC y OSA-CBM, que justifican, auxilian y contextualizan la aplicación de SE en el desarrollo de un sistema MBC.
- Desarrollar un prototipo de SE basado en reglas de producción para auxiliar las decisiones del personal de operación y mantenimiento.
- Presentar los resultados obtenidos de la implementación computacional denominada SIMPREBAL (Sistema Inteligente de Mantenimiento Predictivo de Balbina) instalada en la central hidroeléctrica descrita en el capítulo 6.
- Presentar los aportes de generados en esta tesis para el desarrollo de aplicaciones académicas en otras áreas de investigación que involucren adquisición y almacenamiento de datos, procesamiento inteligente y aplicaciones cliente/servidor.

## **1.2 FORMULACIÓN DEL PROBLEMA**

La principal motivación para la realización de este trabajo surgió en el contexto del proyecto de investigación ANEEL-Electronorte: “Modernización del área de automatización de las centrales hidroeléctricas de Balbina y Samuel”, localizadas en estado de Amazonas-Brasil. La necesidad de definir una metodología para un sistema inteligente MBC para generación de diagnósticos y pronósticos de

anomalías que auxilien al personal de operación y mantenimiento en la toma de decisión. Para atender estas demandas se propone una metodología basado en el modelo de referencia de siete capas OSA-CBM. La implementación computacional para generación de sugerencias de acciones de mantenimiento.

### 1.3 ANTECEDENTES

El MBC posee, intrínsecamente, un componente computacional relacionado a instrumentos, redes industriales y el tratamiento de estos datos durante la fase de operación a través de Sistemas Especialistas u otras técnicas de IA (Inteligencia Artificial) como redes neuronales, lógica difusa, algoritmos genéticos, etc.

A través del uso de sensores y redes de campo se puede monitorear las condiciones de las máquinas. Un sistema remoto de monitoreo distribuido, diagnóstico e integración de información fue realizado por Guo *et al.* (2007), estableciendo un subsistema de monitoreo de condición basado en la tecnología *fieldbus* cubriendo las diferencias de *software*, *hardware*, datos y protocolos de red.

Un prototipo de SE para el apoyo al diagnóstico, pronóstico de fallas y toma de decisión de mantenimiento de compresores centrífugos por medio del monitoreo de condición *online* fue presentada por Mecabô (2007). Aplicaciones relacionadas al proyecto de unidades de potencia hidráulica de sistemas industriales fueron desarrolladas por Vinade (2003) y Caletti (2003). El trabajo desarrollado por Chrissanthi (2008) muestra el uso de los SE para diagnóstico de fallas *online* en procesos industriales.

Garcia *et al.* (2006) desarrollaron una aplicación *software* para diagnóstico en tiempo real de procesos industriales, el cual colecta informaciones en tiempo real de los sensores y de otras fuentes y las procesan usando SE y RNA. Falqueto y Telles (2007) presentaron un SE *fuzzy* para mantenimiento automático de transformadores eléctricos de potencia de la central hidroeléctrica de Itaipu. Molina *et al.* (2000), desarrollaron un sistema usando SE y RNA, llamado MAPAIS (Mantenimiento Predictivo Avanzado usando Imagen y Sonido). Ciarapica y Giacchetta (2006) presentaron un sistema para diagnóstico y pronóstico de fallas en una planta de potencia de ciclo combinado usando RNA recurrentes y sistemas *neuro-fuzzy*. Javadpour y Knapp (2003) implementaron un sistema predictivo basado en RNA y lógica difusa para auxilio a los operadores en el diagnóstico de fallas.

El uso del modelo de referencia OSA-CBM propuesto en esta tesis fue inicialmente usado por Fu *et al.* (2004), ellos presentaron una metodología usando tres elementos: monitoreo y previsión, diagnóstico y pronóstico, y toma de decisión de mantenimiento. Dunsdon y Harrington (2008) propusieron la aplicación de la arquitectura OSA-CBM, proponiendo una solución destinado al gerenciamiento integrado de salud vehicular IVHM (*Integrated Vehicle Health Management*).

---

<sup>1</sup> Falla es definida por Britto (2006) como la inhabilidad de un sistema físico para realizar una función en el nivel de desempeño y es identificada en general, por la negación de la función o parte de ella. Se puede dividir en dos tipos: (a) Evidente: constituye en la pérdida de función que será percibida tarde o temprano por el usuario y (b) Oculta: se refiere a la pérdida de la función que sólo será percibida si otra falla funcional ocurre primero.

<sup>2</sup> Defecto, corresponde a un evento en evolución, caracterizado por un desvío de una condición asumida inicialmente como normal para ítem sobre evaluación, para el instante de tiempo considerado en el análisis (Dupont, 2003).

## CAPÍTULO 2

### MARCO CONCEPTUAL

En este capítulo son presentados conceptos MBC, sistemas MBC, el modelo de referencia OSA-CBM y la descripción de sus capas. También son presentados los conceptos relacionados a Sistemas Especialistas, arquitectura, herramientas para la construcción de un SE y reglas de producción.

#### 2.1 DEFINICIÓN Y TIPOS DE MANTENIMIENTO

Moubray (2000) define el término mantenimiento como el conjunto de acciones que permiten mantener ó restablecer un bien a un estado operacional específico o asegurar un determinado servicio.

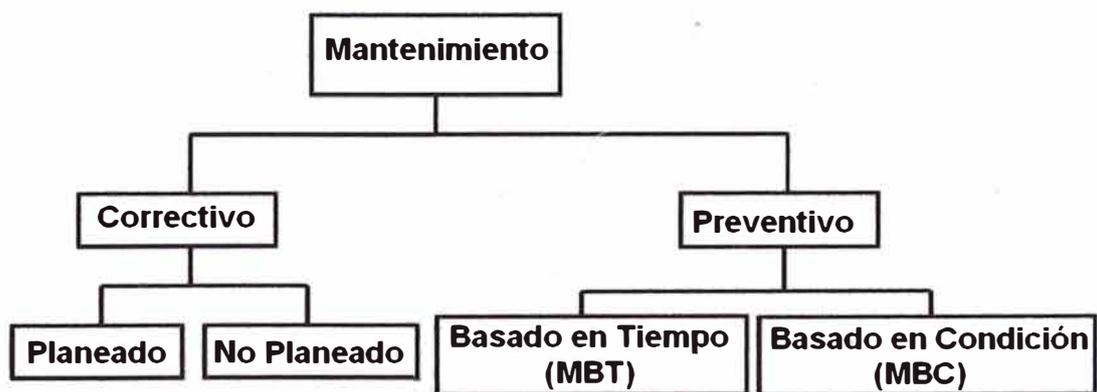


Figura 2.1- Diferentes tipos de mantenimiento (Barroso Maia Junior, 2003).

Según la definición de "mantener o restablecer" es claro que existen dos estrategias principales en la ejecución del mantenimiento mostrados en la Figura 2.1.

El primero es un enfoque preventivo, que según Sim y Endrenyi (1988) es definido como: “una actividad emprendida regularmente a intervalos de tiempo definidos en cuanto el dispositivo está operando satisfactoriamente, para reducir o eliminar la deterioración acumulada”. La segunda, es un enfoque correctivo, que es el mantenimiento efectuado después de la ocurrencia de una falla de modo a recolocar un ítem en condiciones de ejecutar una función requerida (NBR-5462, 1994).

A lo largo de los años, el proceso de mantenimiento industrial pasó por una serie de innovaciones que condujeron a un nuevo paradigma. Moubray (1999) resume en su artículo algunos puntos considerados importantes para la comprensión del nuevo paradigma respecto al gerenciamiento del mantenimiento (Tabla 2.1).

Para Kothamasua y Huang (2007), la tecnología de mantenimiento progresó desde la basada en tiempo hasta la basada en condición. Hace poco tiempo la estrategia de mantenimiento basado en tiempo era adoptada por la gran mayoría de empresas. Esta estrategia consiste en un plano de inspecciones preestablecido, sea por el fabricante o por los técnicos de mantenimiento. La idea de MBC es monitorear un equipo usando sensores y efectuar diagnósticos y pronósticos de fallas inminentes del equipo en tiempo real (Kothamasua y Huang, 2007).

Con el surgimiento de computadoras a bajos precios y las exigencias del aumento en la calidad de productos y servicios. Se tuvo la necesidad de eliminar los inconvenientes de la disponibilidad humana y de equipos, surgimiento de empresas especializadas en *software* para mantenimiento (Lacerda y Júnior, 1997).

Tabla 2.1- Paradigmas de mantenimiento (Moubray 1999, modificado).

| <b>Antiguo</b>  | <b>Nuevo</b>   |
|---|--|
| Mantenimiento trata de la preservación física de los equipos  | Mantenimiento trata de la preservación de las funciones de los equipos   |
| Mantenimiento de rutina trata de la preservación de fallas  | Mantenimiento de rutina trata de evitar, reducir o eliminar la consecuencias de las fallas   |
| El objetivo primario de la función de mantenimiento es optimizar la disponibilidad de la instalación a menor costo.   | Mantenimiento afecta todos los aspectos de eficacia y riesgo del negocio, seguridad, integridad del medio ambiente, calidad del producto y servicio al cliente, no solamente la disponibilidad de la instalación y los costos                    |
| La mayoría de los equipos es mas susceptible a fallas cuando envejece   | La mayoría de las fallas no es mas susceptible de suceder cuando un equipo envejece  |
| Datos de tasa de falla deben estar disponibles antes, para realmente posibilitar el desarrollo de programas de mantenimiento de suceso.   | Decisiones sobre el gerenciamiento de fallas del equipo casi siempre deben ser efectuadas con poca información sobre la tasa de falla.   |
| La frecuencia de las actividades de mantenimiento predictivo debe ser basada en la frecuencia de las tasas o en la criticidad de la falla del ítem  | La frecuencia de las actividades de mantenimiento predictivo debe ser basada en el periodo de fallas (también conocido como " <i>Lead time to failure</i> " – tiempo de conducción para falla o " <i>P-F interval</i> " – intervalo P-F).        |
| Política de mantenimiento genéricos pueden ser desarrollados para la mayoría de equipos semejantes  | Política de mantenimiento genéricos solamente deben ser aplicados para equipos idénticos, cuyo contexto de operación, función y deseos de estandarización de desempeño son también idénticos.  |
| Políticas de mantenimiento deben ser formuladas por gerentes y programas de mantenimiento regidos por especialistas calificados o contratados externos (un enfoque de arriba para abajo). | Políticas de mantenimiento deben ser formuladas por personas próximas a los equipos. El papel de la gerencia es proveer las herramientas para ayudarlos a tomar decisiones correctas y asegurar que las decisiones sean sensibles y defendibles. |

Según Tavares y Filho (2002), el mercado de *software* de mantenimiento representó en 1997 más de 900 millones de dólares de facturamiento, de los cuales 56.6% en América del Norte, 27.5% en Europa, 10.3% en Asia y Oceanía y 5.7% en América Latina. Los *softwares* CMMS más comúnmente usados son: MÁXIMO (18%), SAP (13%), MP2 (13%) y WOMANS (5,3%). El MP2 es mas extensivamente usado en pequeñas plantas, en cuanto SAP es ampliamente usado en grandes plantas (Alkaim, 2003).

## **2.2 MANTENIMIENTO BASADO EN CONDICIÓN**

El MBC es una estrategia que utiliza algunas técnicas de monitoreo de las condiciones operativas de una máquina, sus sistemas y componentes sin necesidad de parar un equipo. El análisis de estas condiciones determinará cuando una intervención será realizada pudiendo o no ocasionar la parada de una máquina (Pinto, 2003). Butcher (2000) define al MBC como: “acciones de evaluación de las condiciones de un equipo basados en tiempo real o casi en tiempo real, la cual es obtenida de sensores embutidos, pruebas externas y medidas realizadas por equipamientos portátiles”.

Para Mecabo (2007), el MBC es un programa que recomienda decisiones basadas en informaciones colectadas. Según Bengtsson (2004b) el MBC es definido como: “mantenimiento preventivo basado en el desempeño o monitoreo de parámetros y las acciones subsecuentes”.

El desempeño y monitoreo de parámetros pueden ser programados, requeridos, o continuos. El MBC es de esta forma una tecnología de mantenimiento que usa herramientas de monitoreo de condición para analizar la condición actual de un componente y, a través de este conocimiento, ejecutar un apropiado programa de mantenimiento preventivo. Será predictiva desde que los intervalos de mantenimiento y las tareas sean basados en la condición del componente.

### **2.2.1 Sistemas MBC**

Bengtsson (2004b) lo define como: "Un sistema que usa el MBC para determinar y programar acciones de mantenimiento predictivo automático o en interacción con otros sistemas u operadores".

El propósito de un sistema MBC es transformar medidas definidas en forma de energía (vibración, temperatura, presión, etc.) en efectos deseados (información de la condición de un ítem, pronóstico de condición futura, etc.).

## **2.3 TECNOLOGÍAS DE COMUNICACIÓN INDUSTRIAL**

Con la introducción de la tecnología de monitoreo de procesos en la década de los 90, actualmente la función del operador es mas gerencial a través del uso de herramientas de apoyo en actividades asociadas a la confiabilidad y al mantenimiento de sistemas (Rigoni *et al.*, 2004), surgió la necesidad de monitorear las condiciones de los equipos a través de señales o medidas de las variables de proceso. En este sentido presentamos las tecnologías usadas en el desarrollo de la metodología como son la instrumentación *foundation fieldbus* y la tecnología OPC.

### 2.3.1 Fieldbus

Según Mahalik y Yen (2008), los buses de campo *fieldbus* constituyen redes industriales y son definidos como un bus (dos hilos) digital, serial, multipunto y bidireccional, para la comunicación entre dispositivos de campo, como sensores, actuadores, reguladores, controladores e interfaces hombre máquina HMI (*Human Machine Interface*).

Según Armitage *et al.* (1988) *fieldbus* distribuye puntos de entrada/salida, posibilitando realizar el control en el local de adquisición de datos y en los puntos de acción de procesos, o sea, en los sensores y actuadores. *Fieldbus* es una red local LAN (*Local Area Network*) para automatización de procesos, con capacidad de distribuir el control en el campo como mostrado en la Figura 2.2.

*Fieldbus* surgió con el objetivo de interconectar y operar instrumentos de campo con características diferentes y de diversos fabricantes, proporcionando la descentralización de tareas. Esta interconexión tiene ventajas como: mayor inmunidad a ruidos, pre-procesamiento de datos específicos, transmisión de informaciones adicionales de datos, capacitando el diagnóstico del dispositivo y la previsión de fallas, así como la reducción de costos de proyecto, de cableado, de instalación y expansión (Rigoni *et al.*, 2004).

Según Mahalik y Yen (2008), existen dos redes FF (*Foundation Fieldbus*) mostradas en la Figura 2.2, una de baja velocidad concebida para interconexión de instrumentos (H1 – 31.25 kbps) en dos hilos llegando hasta 1.9km y otra de alta velocidad utilizada para integración de las demás redes y para la conexión de dispositivos de alta velocidad como PLCs conocido como HSE (*High Speed Ethernet*) a una velocidad de 100 Mbps. La interface entre estas dos redes es a través de un dispositivo de enlace o DFI (*Interface fieldbus Distribuida*).

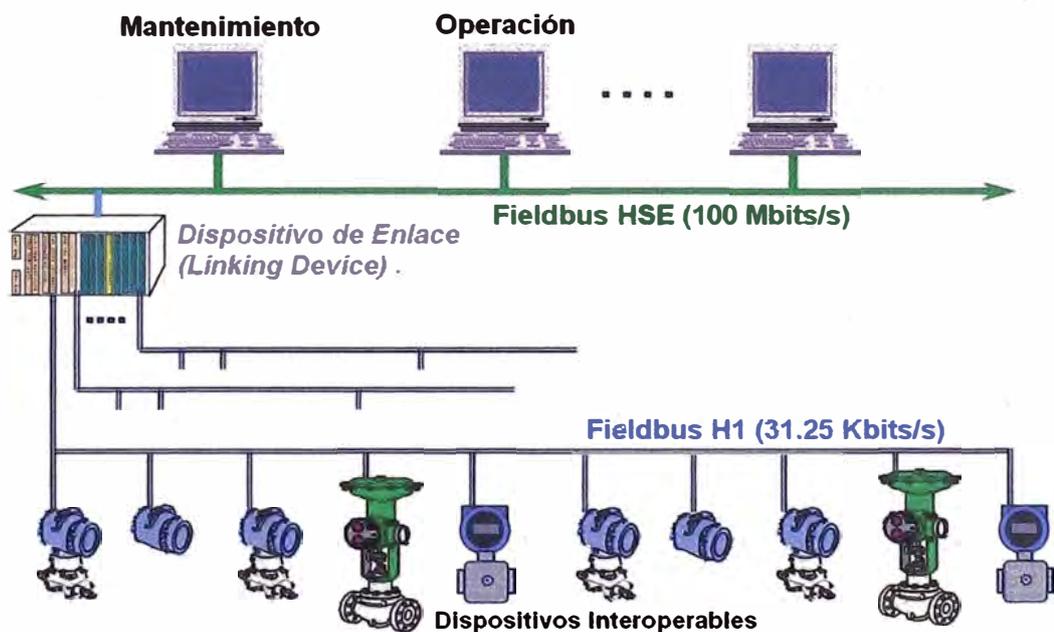


Figura 2.2- Redes *fieldbus* H1 y HSE.

### 2.3.2 Tecnología OPC

Según Anwar *et al.* (2004), la motivación para el desarrollo de OPC (*OLE for Process Control*) fue tener un padrón para la comunicación de

varias fuentes de datos, dispositivos de campo o banco de datos en la sala de control. La arquitectura de esas fuentes de datos en la industria de procesos es mostrada en la Figura 2.3 y consiste de tres niveles:

1. **Gerencia de Campo:** con la aparición de dispositivos de campo inteligentes, la cantidad de información puede ser usada para evaluación de salud de los dispositivos de campo. Estas informaciones son adquiridas a partir de los dispositivos, parámetros de configuración, materiales de construcción, etc.
2. **Gerencia de Proceso:** con la instalación de DCS (*Distributed Control System*) y sistemas SCADA para monitorear y controlar los procesos de manufactura, disponibilizando electrónicamente los datos.
3. **Gerencia de Negocio:** muchos beneficios pueden obtenerse con la instalación de sistemas integrados de información dentro de un sistema empresarial, administrando aspectos financieros del proceso de manufactura.

Según Shimanuki (1999), OPC es un protocolo de comunicación abierto que permite un método consistente para acceder a datos de inúmeros instrumentos de los más diversos fabricantes (Figura 2.4). El método es independiente del origen de datos, ofreciendo al usuario final una mayor libertad en la selección de los instrumentos independientemente del fabricante.

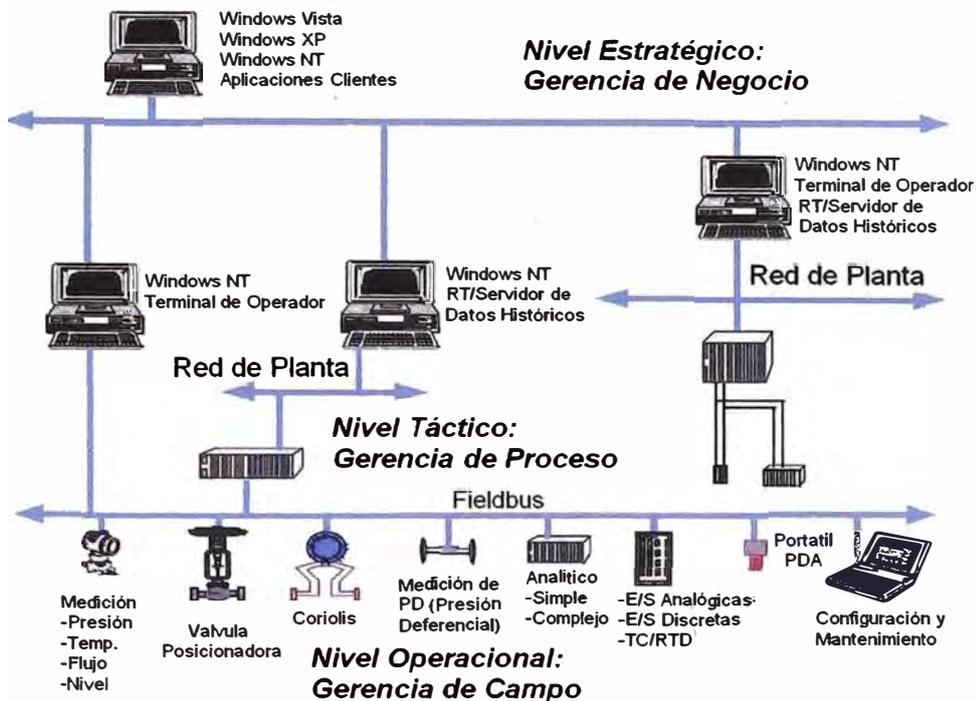


Figura 2.3- Arquitectura de información en control de procesos.

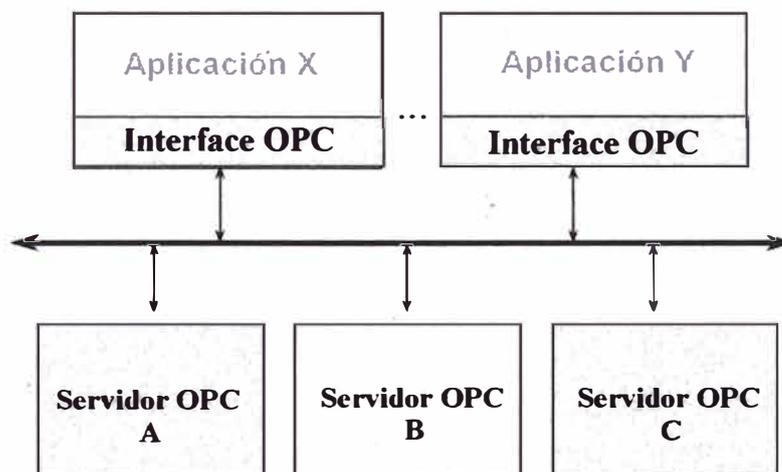


Figura 2.4- Aplicaciones con varios servidores OPC.

OPC fue desarrollado usando la tecnología *Microsoft OLE/COM*, sin embargo, la especificación OPC fue desarrollada por una fundación abierta

para atender las necesidades principales de la industria y no las necesidades de algunos fabricantes de *hardware* y *software* (OPC Foundation, 1998).

Los componentes OPC se clasifican en dos categorías: clientes y servidores OPC. Un cliente OPC es típicamente un usuario de los datos a través de una interface de operación o un sistema SCADA. Un servidor OPC es una fuente de datos que colecta o genera datos a partir de un proceso. El cliente OPC interactúa con el servidor OPC usando una interface definida. Cualquier cliente OPC se puede comunicar con cualquier servidor OPC, independientemente del tipo de dispositivo y del fabricante (Shimanuki, 1999), conforme está esquemáticamente representado en la Figura 2.5. Esa comunicación es válida solamente para el OPC-DA (*Data Access*), una vez que existen diferentes tecnologías OPC (Duarte *et al.*,2006).

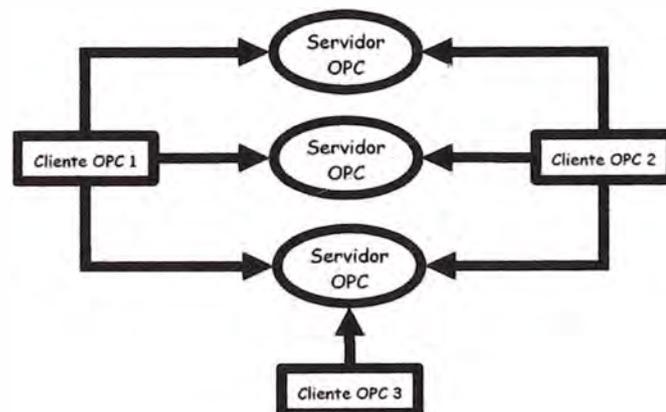


Figura 2.5- Relación entre Clientes y Servidores.

Los tres componentes básicos de la arquitectura OPC presentados en la Figura 2.6 son: servidor, grupo e ítem. Del punto de vista del cliente, un

servidor es esencialmente una estructura de almacenamiento de grupos que, por su vez, tiene como función básica el almacenamiento de ítems. Esos ítems, elementos más simples en la especificación, representan conexiones a puntos de entrada o salida.

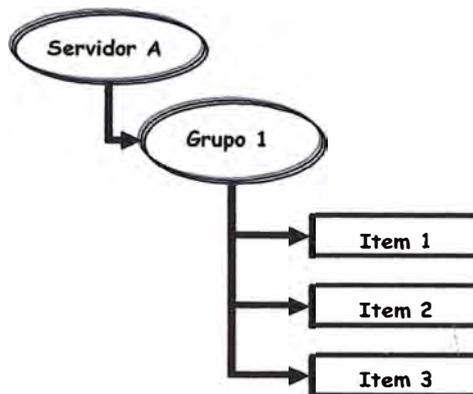


Figura 2.6- Arquitectura OPC.

Un ítem OPC no es un valor, sino apenas un medio de acceso a un valor. De esta forma, una única variable de entrada o salida puede ser representada por ítems diferentes, con propiedades distintas y compartida por mas de un cliente. La tarea de los grupos es unir un conjunto de ítems que interesan a un determinado cliente, asumiendo el papel principal en la interacción cliente-servidor.

Del punto de vista del cliente, la función básica del servidor es proveer una infraestructura de soporte a los grupos. Además de eso, cabe también a él administrar aspectos relacionados a la conexión con la fuente de datos, tales como los parámetros de comunicación o tasa máxima de muestreo. Otra responsabilidad del servidor es implementar una estructura de

direccionamiento capaz de asociar ítems con variables reales (Souza *et al.*, 1998).

Un ítem es una estructura al cual están asociados tres propiedades (Fonseca, 2002):

1. **Value:** último valor almacenado por el servidor en la memoria temporal del ítem y que es actualizado siempre que el servidor hace una lectura en el dispositivo,
2. **Quality:** información de estado que define la calidad del dato, puede ser: *Good*, dato válido, *Bad*, pérdida del enlace de comunicación con el dispositivo de campo, y *Uncertain*, en caso de existir enlace y el dispositivo de campo estuviera fuera de comunicación.
3. **Time Stamp:** fecha y hora en que un ítem es adquirido.

## 2.4 ARQUITECTURA OSA-CBM

En la implementación de sistemas MBC se pugna con la tarea de integrar una variedad de componentes de *software* y *hardware*, así como de desarrollar una estructura para estos componentes. OSA-CBM simplifica este proceso especificando una arquitectura padrón para la implementación de sistemas MBC. El padrón provee la forma para integrar componentes de diferentes fabricantes y facilita el proceso especificando las entradas y salidas (Walter, 2006).

La arquitectura OSA-CBM consiste de siete capas con funcionalidades independientes basado en el padrón ISO-13374:2007. La ISO-13374 (2007) no

especifica como implementar un sistema de monitoreo de condición, que tecnologías usar, o que algoritmos implementar, mas provee una estructura general para el área de monitoreo de condición y diagnóstico. OSA-CBM define los tipos de datos a ser usados para el procesamiento y presentación de los resultados en un sistema de monitoreo de condición, como también cuales informaciones son transmitidas entre los puntos del proceso y almacenamiento, permitiendo desarrollar las funcionalidades de cada camada independientemente.

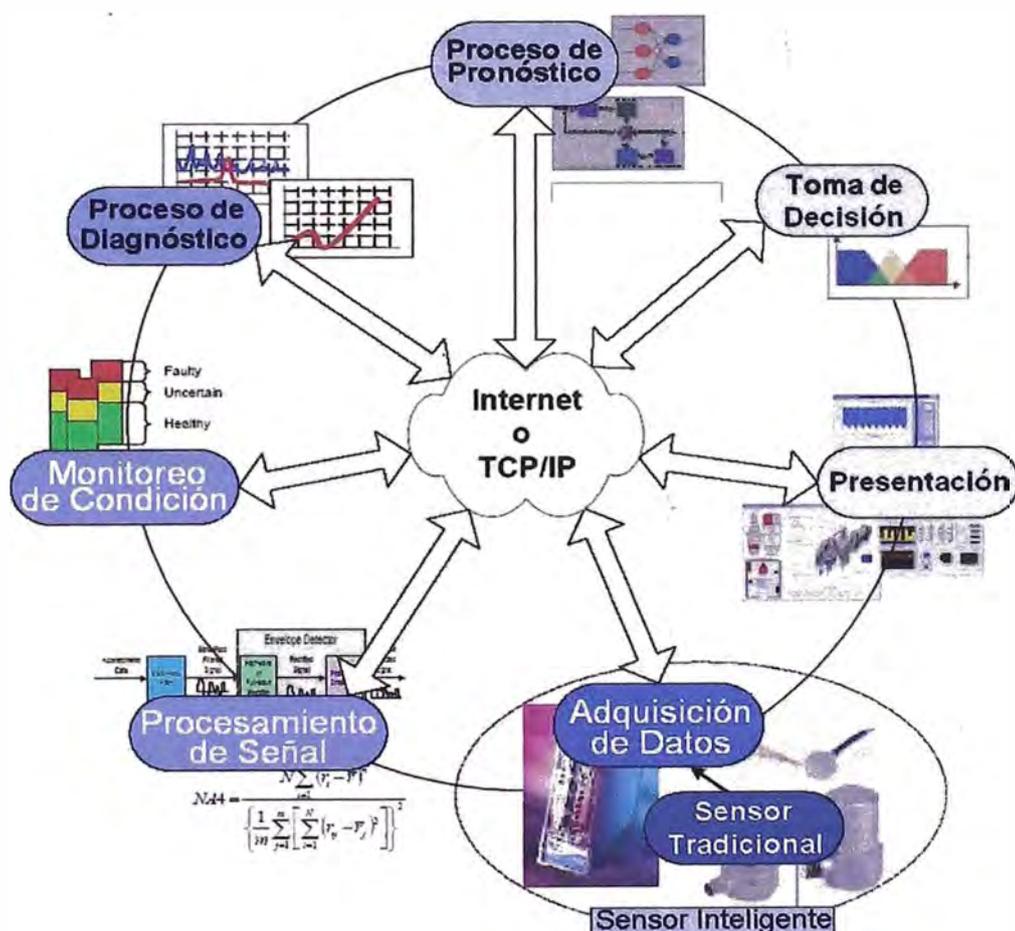


Figura 2.7- Camadas funcionales OSA-CBM (Lebold y Thurston 2001).

En la Figura 2.7 muestra como una camada OSA-CBM interactúa con las otras para completar el sistema integrado. El centro del círculo representa el medio de comunicación entre las capas, lo que puede ser conseguido usando el protocolo de internet TCP/IP o HTTP (*Hyper Text Transfer Protocol*), de modo que las capas no necesitan estar en el mismo computador pudiendo encontrarse en cualquier local de la red mundial.

La noción de una arquitectura extendida en capas es consistente con el concepto usado en Buschmann *et al.* (1996), Álvares *et al.* (2007), Amaya *et al.* (2007c), Bengtsson (2003), Bengtsson (2004a), Bengtsson *et al.* (2004) y Thurston (2001). Una camada OSA-CBM ofrece tres tipos de información (OSA-CBM, 2006):

1. **Datos:** son las informaciones o eventos que una camada genera, pueden ser las lecturas de un sensor inteligente<sup>1</sup> en la camada de adquisición de datos y el estado de salud en la camada de diagnóstico;
2. **Configuración:** información sobre los recursos de entrada de las capas, descripciones de los algoritmos usados para procesar datos de entrada, lista de salidas, y varias especificaciones de salida como unidades de ingeniería y valores de alarmas;
3. **Explicación:** el dato o referencia a un dato usado por una camada para producir una salida.

Las capas jerárquicas representan una transición lógica o flujo de información desde la salida de los sensores hasta la camada de toma de decisión, a través de las capas intermediarias. La camada de presentación es una excepción

dentro de la arquitectura, pues ella permite una comunicación punto a punto entre esta camada y cualquier otra. A seguir se detallan las funcionalidades de cada una de las camadas.

#### **2.4.1 Adquisición de datos**

Normalmente, en un contexto objetivo, el sensor es un componente de adquisición de datos y considerado parte de la camada monitoreo de condición. Según Fraden (2003) el sensor es un dispositivo que recibe y responde a una señal o estímulo, es así que el equipo captura el efecto dinámico causado por una falla incipiente. Según Bengtsson (2004a) esta camada debe ser desarrollada de acuerdo con el padrón IEEE (*Institute of Electrical and Electronics Engineers*) Std 1451 (Figura 2.8).

#### **2.4.2 Procesamiento de señal**

Esta camada recibe señales de la camada adquisición de datos y de otros sistemas de procesamiento de señal, la salida de esta camada incluye digitalización y filtrado de los datos del sensor, espectro de frecuencia, señales de sensores virtuales entre otras. Según Bengtsson *et al.* (2004), los propósitos de la camada procesamiento de señal son: (1) remover distorsiones y restablecer una señal a su forma original, (2) remover datos irrelevantes del sensor para diagnósticos o pronósticos, y (3) transformar la señal para hacer las características relevantes mas explícitas.

### 2.4.3 Monitoreo de condición

Esta camada recibe informaciones de las camadas de adquisición de datos, procesamiento de señal y de otros sistemas de monitoreo de condición, donde el objetivo principal es comparar datos con los valores esperados (Bengtsson *et al.*, 2004). Según Bengtsson (2004a), esta camada debe ser desarrollada usando el padrón ISO 13373-1 (Figura 2.8). Si los niveles normales son excedidos u otro fenómeno anormal sucede, como aumentos súbitos o disminuciones en el nivel (sin exceder niveles de los límites operacionales), los datos necesitan ser diagnosticados. Los límites pueden ser estáticos o dinámicos (Tsang, 1995).

Los avisos de límites estáticos utilizan valores límites predeterminados. De acuerdo con Tsang (1995), límites de advertencia estáticos son más fácilmente administrados que los límites dinámicos. Mas estos no tienen la capacidad de diagnóstico para predecir cuando una alarma será alcanzado. Los límites dinámicos son usados para monitorear la tasa de cambio del parámetro medido, si un procedimiento de MBC usa límites de advertencia dinámicos, la tasa de cambio del parámetro medido es considerada mas importante que el valor actual (Tsang, 1995). La monitoreo de condición debe ser capaz de generar alertas basados en los límites operacionales establecidos.

#### 2.4.4 Evaluación de salud

Esta camada recibe datos de la camada monitoreo de condición o de otros sistemas de diagnóstico, objetivándose en prescribir si la salud del componente monitoreado, sistema o subsistema fue degradado (Bengtsson *et al.*, 2004). Según Bengtsson (2004a), esta camada puede ser desarrollada a partir de las normas IEEE 1232 y la ISO 13373-1 (Figura 2.8). De acuerdo con Yam *et al.* (2001), los diagnósticos en el MBC pueden ser divididos en tres categorías: (1) diagnósticos basados en reglas, (2) diagnósticos basados en casos, y (3) diagnósticos basados en modelos. La evaluación de salud debe tener la capacidad de generar registros e indicar las posibles fallas basada en tendencias del histórico de salud, estado operacional e histórico de mantenimiento (Bengtsson *et al.*, 2004).

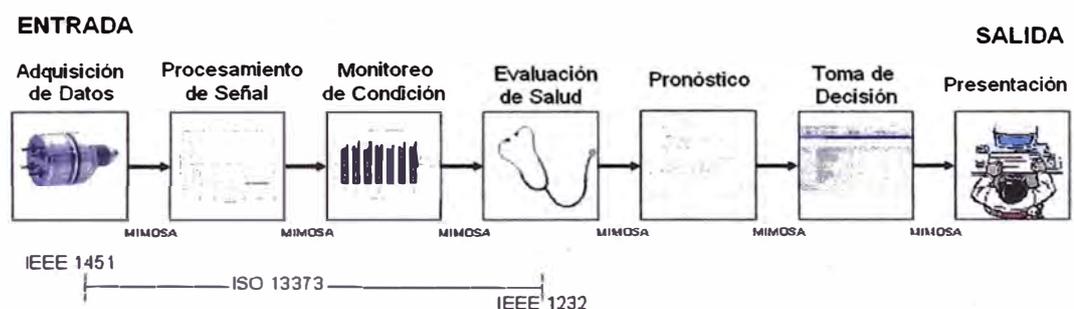


Figura 2.8- Padrones a usar en las camadas OSA-CBM (Bengtsson 2004a).

#### 2.4.5 Pronósticos

Continuando con el diagnóstico, el sistema tendrá el conocimiento si una condición es anormal, y lo que está causando esas medidas anormales, necesitando ser pronosticado. Esta camada será la que podrá predecir cuanto

tiempo un equipo puede operar antes de ser necesario ejecutar una orden de mantenimiento previa a una falla. Los pronósticos pueden ser ejecutados como la camada de diagnósticos, a través de diferentes técnicas de inteligencia artificial, como redes neuronales recurrentes (Yam *et al.*, 2001) y redes neuronales *wavelet* dinámicas (Vachtsevanos y Wang, 2001), etc.

En la Figura 2.9, se muestra un diagrama de bloques genérico de la camada de pronóstico con el objetivo de visualizar los componentes principales. Dentro de esta camada, la información puede ser utilizada por un enfoque basado en modelos, enfoque basado en características o la combinación de ambas.

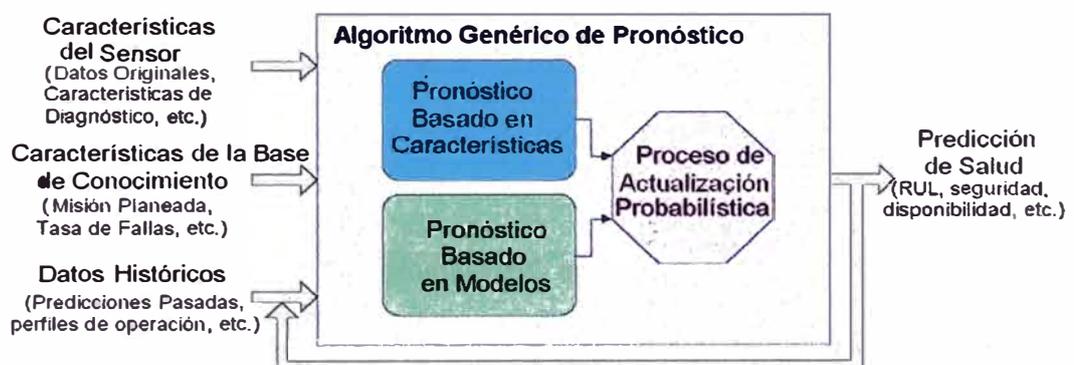


Figura 2.9- Componentes de la camada pronóstico (Lebold y Thurston 2001).

La camada de pronóstico debe proveer información específica al usuario acerca del estado de salud de los componentes, tiempo de vida remanente (RUL - *Remaining Useful Life*), seguridad y recomendaciones. Lebold y Thurston (2001) presentan una propuesta para la camada de

pronóstico genérico en el cual los requerimientos de las entradas son datos históricos, por ejemplo, en la forma de salud, fallas, misión, mantenimiento, modelo de información, y piezas de repuestos. Estos recursos de entradas y salidas (Figura 2.10) ayudan a definir la estructura y aclarar la presencia de la camada de pronóstico en el modelo OSA-CBM.

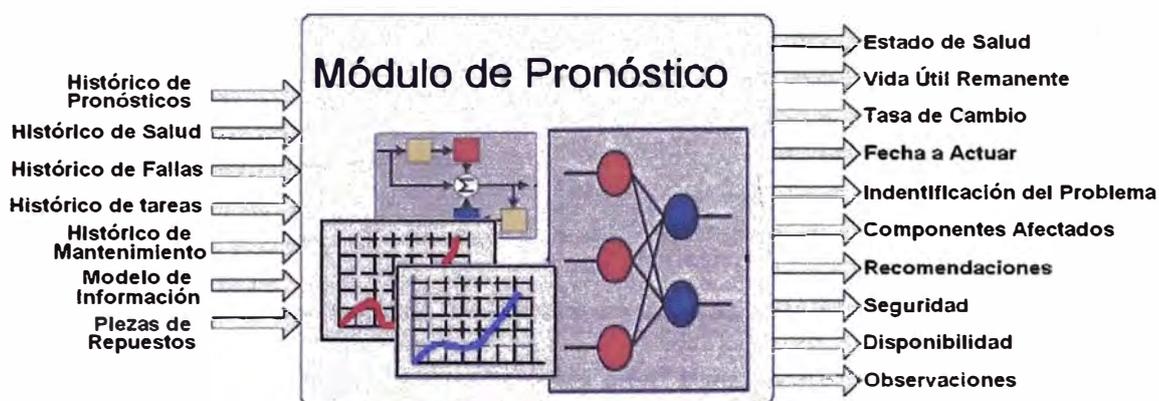


Figura 2.10- Entradas y salidas de la camada pronóstico.

#### 2.4.6 Toma de decisión

El último paso del sistema MBC es generar decisiones de acuerdo con las acciones de mantenimiento a ejecutar. Todas las actividades previas de las camadas de evaluación de salud y pronóstico deben ser integradas a la toma de decisión para hallar la mejor solución para un evento particular. Siendo el objetivo principal generar recomendaciones de las acciones de mantenimiento alternativas, estas acciones relacionadas al mantenimiento o como utilizar los recursos hasta completar la secuencia actual sin ocurrencia de falla (Bengtsson *et al.*, 2004). Esta decisión debe ser automática pudiendo ser usada por otros sistemas o por un operador (Guo *et al.*, 2002).

#### **2.4.7 Presentación**

Esta camada es la interface hombre/máquina, pudiéndose comunicar y presentar datos de las otras camadas. Las camadas mas importantes para presentar informaciones serían, la de diagnóstico, pronósticos y toma de decisión, así como también las alertas generadas de la camada monitoreo de condición, debiendo tener como posibilidad ver las camadas inferiores (Bengtsson *et al.*, 2004).

### **2.5 SISTEMAS ESPECIALISTAS**

Vinade (2003) define, de forma general, SE como un programa de computador desarrollado para resolver problemas de un área específica, tal como el raciocinio de un especialista humano de la misma área. Para emular la habilidad de un especialista, este tipo de programa utiliza la capacidad de relacionar las informaciones del problema a ser solucionado con el conocimiento almacenado.

Un SE no imita necesariamente la estructura de la mente humana, ni los mecanismos de la inteligencia. Son programas prácticos que usan estrategias heurísticas desarrolladas por humanos en la solución de clases específicas de problemas. Son una clase de programas en el área de IA, y han contribuido para el suceso de esta área a través de varios productos comerciales desarrollados.

Los SE son recomendados para resolver problemas que pueden tener diferentes soluciones delante de un conjunto de datos disponibles. Los datos del problema y los casos almacenados en los SE son combinados a través de heurísticas,

que generan informaciones hasta encontrar una solución. La Tabla 2.2 aborda estas características mencionadas de un SE presentando las principales diferencias entre un SE y un sistema convencional.

Tabla 2.2- Diferencias entre un SE y un sistema convencional (Bauchspiess 2004).

|                                       | <b>Sistema Convencional</b>                | <b>Sistema Especialista</b>  |
|---------------------------------------|--|--|
| <b>Representación de conocimiento</b> | Implícita, inserida en estructura de datos | Explícita. El conocimiento es representado separadamente del código  |
| <b>Conocimiento Traducido en</b>      | Algoritmos determinísticos                 | Estructuras jerárquicas, redes semánticas, reglas, arboles de decisión, tablas de decisión o redes de inferencia |
| <b>El computador Ejecuta</b>          | Procesamiento numérico (cálculos)          | Procesamiento simbólico (inferencias)  |
| <b>Explicación de raciocinio</b>      | Generalmente inexistente                   | Existente y recomendable   |

### 2.5.1 Arquitectura

Segundo Giarratano y Riley (1994), el profesor Edward Feigenbaum de la Universidad de *Stanford*, uno de los investigadores reconocidos en trabajos con SE, definió SE como: “un programa inteligente de computador que usa conocimiento y procedimientos de inferencia para resolver problemas que son suficientemente difíciles para requerir significativa experiencia humana para su solución”. De acuerdo con esta definición, un SE es formado por dos módulos básicos mostrados en la Figura 2.11 (Abel, 1998).

Una base de conocimiento conteniendo informaciones especializadas en el área del problema a ser solucionado, codificado de manera inteligible

para ser fácilmente modificado o reutilizado (Caletti, 2003). El motor de inferencia que representa los métodos inteligentes de manipulación de conocimiento para llegar a una solución, respuesta o conclusión, a partir de un determinado conocimiento inicial (Aulete, 1986 y Gonzalez y Dankel, 1993).

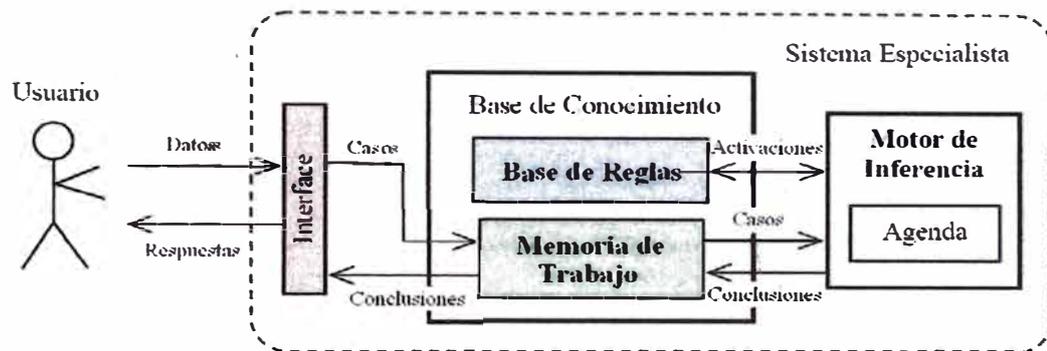


Figura 2.11- Estructura de un SE basado en reglas de producción.

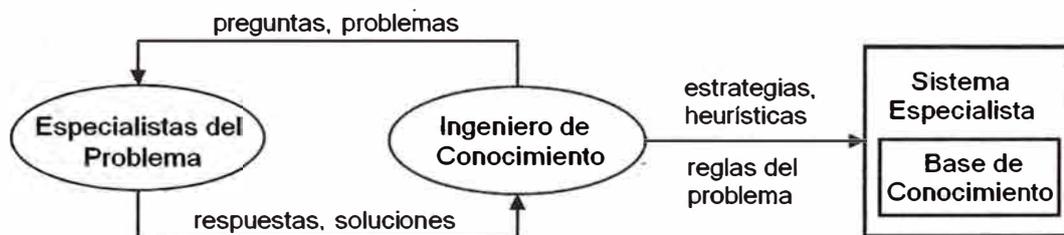


Figura 2.12- Interacción entre el IC y el especialista (Waterman, 1986).

Siendo la representación de conocimiento una de las principales preocupaciones de los SE y de la Inteligencia Artificial (Liebowitz, 1999), surge la ingeniería de conocimiento como un “proceso de construir un SE” (Durkin, 1994). A diferencia de la programación convencional, desarrollar un SE es un proceso altamente interactivo que envuelve, típicamente, una forma

especial de interacción entre el desarrollador del SE, llamado Ingeniero de Conocimiento (IC), y un o mas especialistas en un problema específico de un área o dominio de conocimiento (Figura 2.12).

### 2.5.2 Herramientas

El desarrollo de un SE puede ser realizado completamente en un lenguaje de programación como PROLOG o LISP. Mas existen herramientas computacionales especialmente desarrolladas para la creación de SE, como el sistema ICAD de la KTI (*Knowledge Technologies International*), el ambiente CLIPS (*C Language Integrated Production System*) y JESS (*Java Expert System Shell*).

JESS es una *shell* para SE desarrollado completamente en Java por Ernest Friedman-Hill de la *Sandia National Laboratories*, cuya primera versión fue en 1995 de uso gratuito para instituciones educativas. JESS originalmente fue clon de CLIPS, actualmente por la influencia de Java tiene muchas características que lo diferencian de CLIPS. JESS permite su extensión llamando a funciones Java, digitando códigos Java y embebiendo a JESS en aplicaciones Java. Con JESS se puede dar a un *Applet* Java y a otras aplicaciones, la habilidad de razonar.

JESS usa un algoritmo especial llamado RETE para el relacionamiento de las reglas con los casos, el RETE torna al JESS mas rápido que un simple conjunto de sentencias *if...then* (Friedman-Hill, 2003).

Los métodos de inferencia de JESS son de dos tipos de concatenación (directo y reverso), por tanto dos estrategias de busca. La construcción de SE puede ser realizada a través del *prompt* de JESS o desde un editor de texto. JESS versión 7.1 es usado en esta tesis para el desarrollo del SE.

JESS es utilizado en diversas aplicaciones, mas el uso de JESS con la tecnología de *Applets* deja al sistema muy pesado. Por eso cuando la idea es utilizar aplicaciones con JESS vía navegador, debemos considerar su uso en el lado servidor, como lo que ocurre en el caso de los *Servlets*, prescindiendo al usuario cargar gran parte del sistema para su máquina lo que torna la interacción con el sistema bastante lenta.

### **2.5.3 Reglas de producción**

Las reglas de producción son del tipo: “Si CONDICIÓN Entonces ACCIÓN”, una condición son antecedentes (casos u objetos) y una acción son consecuentes (nuevos casos u objetos) que darán secuencia a la concatenación de reglas.

En un SE basado en reglas de producción, existen básicamente dos modos de raciocinio posibles asociados al motor de inferencia: “concatenación directa o progresiva” y “concatenación reversa o regresiva”. En la concatenación directa, se inicia con los datos disponibles y se usa el ciclo de inferencia para extraer más datos hasta alcanzar el objetivo. En la concatenación reversa, se inicia con la lista de objetivos o hipótesis

trabajando en reverso desde el consecuente hasta el antecedente para verificar si existen datos disponibles que soporten cualquiera de las consecuencias.

---

<sup>1</sup> Sensor inteligente, dispositivo capaz de proveer funciones más de las necesarias, para generar una correcta representación de la cantidad medida o controlada (Especificación IEEE 1451.2).

## **CAPÍTULO 3**

### **METODOLOGÍA**

En este capítulo se presentan las especificaciones, técnicas y algoritmos utilizados en la metodología para la implementación de un sistema inteligente MBC usando informaciones provenientes de servidores OPC y banco de datos, y procesadas a través de SE. Esta metodología es presentada en la forma de diagramas IDEF0 e IDEF1X, para el modelo funcional y de información, respectivamente.

#### **3.1 CONSIDERACIONES DEL MODELO OSA-CBM**

Basado en el estudio de metodologías de sistemas se sugirió y adoptó el modelo de referencia OSA-CBM. Con relación a este modelo, se pueden listar algunas ventajas:

- Se trata de una metodología que busca ser genérica para satisfacer el desarrollo de soluciones tanto de *software* como de *hardware*;
- Por ser definida en capas, permite el desarrollo modular, o sea, es posible modificar funciones lógicas o elementos de *hardware* independientemente de los otros elementos pertenecientes a los demás módulos, facilitando, por tanto, el mantenimiento futuro y evolución del sistema.
- Permite construir un sistema abierto que sigue el concepto propuesto por la definición de la IEEE (1996). Según la referida institución, un sistema abierto

ofrece las potencialidades que permiten aplicaciones correctamente desarrolladas, funcionar en una variedad de vendedores, interoperar con otras aplicaciones del sistema y de interacción con el usuario.

La arquitectura utilizada en el desarrollo de esta metodología es presentada en la Figura 3.1, donde se ilustra la técnica de IA usada y el flujo de información entre las capas. La descripción se inicia con el sistema a monitorear, continuando con las capas desde la adquisición de datos hasta llegar a la camada de presentación, siendo esta última la interface con el usuario.

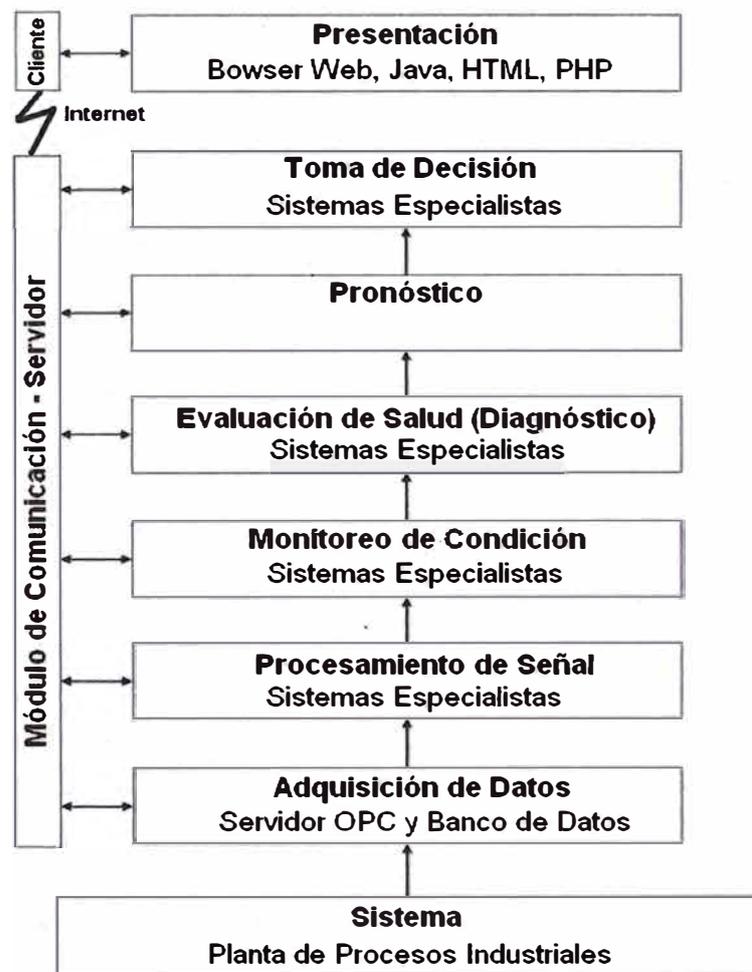


Figura 3.1- Arquitectura conceptual del sistema MBC.

El sistema MBC es concebido en un modelo computacional cliente/servidor. En el lado servidor son ejecutadas las seis primeras capas y la capa de presentación es implementada en el lado cliente. Las informaciones del servidor son solicitadas por los clientes a través de conexiones *sockets* directas utilizando la *web* como medio de comunicación.

La metodología idealizada utiliza la tecnología de internet para comunicación de los clientes con el servidor. Este ambiente es globalizado, centrado en red y distribuido. En la interface del cliente se utilizan navegadores basados en *Web* y lenguajes HTML (*HyperText Markup Language*), PHP y Java, permitiendo la independencia del sistema operativo del usuario. Además se considera el almacenamiento de las variables de proceso, diagnóstico de salud de los equipos y tomas de decisión.

### **3.1.1 Adquisición de datos**

El procesamiento se inicia con la colecta *online* de datos (ítems) provenientes de la instrumentación a través de servidores OPC vía JNI (*Java Native Interface*). Otra forma de obtener datos es a través de bancos de datos vía JDBC (*Java Database Connectivity*), desde que otros sistemas (*i.e.* sistema SCADA) almacenen datos de instrumentación, condiciones de operación y evaluación de salud, posibilitando el desarrollo de un sistema integrado con otros sistemas dentro de una planta industrial.

### 3.1.2 Procesamiento de señal

En esta camada, las variables digitales obtenidas en la camada de adquisición de datos son procesadas de modo de convertirlas en una forma capaz de representar la medida física que está siendo monitoreada, caso necesario, efectuar cálculos matemáticos sobre ella. Contribuciones para esta camada incluyen informaciones de calidad de señal *fieldbus* que indican el estado de salud de la instrumentación, y calidad de comunicación OPC que detecta el estado de conectividad con el servidor OPC, estos datos de calidad son procesadas por las reglas de producción del SE.

### 3.1.3 Monitoreo de condición

Esta camada determina la condición de la planta, sus sistemas, subsistemas o componentes (límite de operación, ciclo de tensión y condición operacional). El procesamiento de esta camada es a través de SE basado en reglas de producción, teniendo como entradas las informaciones generadas por las camadas de adquisición de datos y procesamiento de señal. Las salidas de esta camada están relacionadas a los cuatro rangos de operación: NORMAL, ALTO, ALARMA, y TRIP, ilustrado en la Figura 3.2.

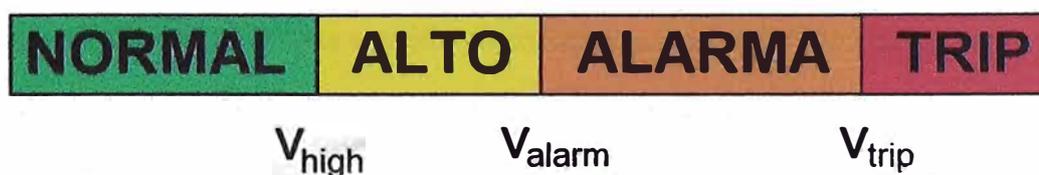


Figura 3.2- Rango de operación de un equipo.

### **3.1.4 Evaluación de salud**

Esta camada determina el estado de la planta, sistema, subsistemas o componentes monitoreados basados en las informaciones generadas por las camadas anteriores, y de los valores de referencia. La evaluación ocurre a través de la extracción de características de cada equipo y posterior detección de anomalías de los mismos. Las reglas de producción generadas para esta camada incluyen diagnóstico de la instrumentación *fieldbus*, diagnóstico de comunicación OPC y diagnóstico del estado de operación de los equipos. La salida de esta camada es un índice de estado del equipo monitoreado que es almacenado en banco de datos y mostrado en la camada de presentación a través de señales de alarma.

### **3.1.5 Pronóstico**

En esta camada la predicción de fallas puede ser realizada usando el histórico de anomalías, histórico de variables y las relaciones entre las anomalías y sus variables asociadas. Según Molina *et al.* (2000), mientras un SE intenta imitar la respuesta de un operador analizando las mismas variables, las redes neuronales superan este límite e intentan analizar las relaciones no lineales entre las diferentes señales. Amaya (2008) propone para el desarrollo de esta camada un enfoque aplicando RNA, lógica difusa y sistemas inteligentes híbridos como la red *Fuzzy* ARTMAP.

Souza (2008) propone la aplicación de técnicas de pronóstico basadas en la teoría de decisión aplicado al Mantenimiento Centrado en Confiabilidad

(a través de medidas de tasas de falla, tiempo medio entre fallas, confiabilidad, disponibilidad, tiempos de reparo, etc.) para estimar cuando una falla probablemente irá ocurrir según ciertas condiciones y decidir el instante de realización de mantenimiento.

### **3.1.6 Toma de decisión**

Considerando las informaciones sobre el diagnóstico y pronóstico de salud de la máquina, sistema, subsistemas o componentes, así como una noción respecto de la severidad, urgencia e importancia de tomar cierta decisión. Esta camada tiene el objetivo de integrar informaciones necesarias con fines de generar sugerencias de acciones de mantenimiento.

Según Bengtsson (2007), con el objetivo de tener una decisión apropiada es sugerido realizar análisis críticas usando el método de análisis FMEA o el análisis por árbol de fallas FTA, los conceptos de estos métodos de análisis son explicados en el Apéndice C. Esta camada irá realizar la toma de decisión basándose en su base de conocimiento generada a partir del árbol de faltas/fallas, del árbol de síntomas, FMEA y de las informaciones colectadas de los especialistas.

Las experiencias de los operadores para toma de decisión son implementadas en reglas de producción donde se procesan códigos de diagnósticos. Las decisiones generadas por esta camada son enviadas a la

camada de presentación a través de sugerencias de las posibles acciones de mantenimiento que puede ser adoptado por los operadores y mantenedores.

### **3.1.7 Presentación**

Esta camada es el nivel mas alto del modelo de referencia OSA-CBM y el lado cliente de la arquitectura cliente/servidor de la metodología propuesta. Soporta una interface de usuario presentando las informaciones producidas por las camadas mas bajas, monitoreo de variables en tiempo real e histórico de tendencias de variables asociadas a las anomalías producidas. La camada de presentación es desarrollada para una GUI (*Graphic User Interface*) basada en *browser* (Netscape, Mozilla, IExplore, entre otros) usando HTML, PHP y *Applets* (Java).

## **3.2 MODELO FUNCIONAL IDEF0**

Los diagramas IDEF0 (*Integration DEFINition language 0*) puede ser utilizado para modelar las decisiones, acciones y actividades de un sistema en forma gráfica estructurada (Kim *et al.*, 2002).

Estos diagramas poseen funciones que reciben entrada, control, salida y mecanismos, la entrada recibe el dato a ser convertido por la actividad, el control agrega responsabilidad de como y cuando una entrada debe ser procesada y ejecutada, la salida presenta el resultado de como la entrada fue procesada y el mecanismo representa quien debe ejecutar esta actividad (puede ser una persona, equipo, máquina u otras organizaciones).

La Figura 3.3 muestra que una función representada en el nivel superior puede ser descompuesta en otras sub-funciones que muestren detalles adicionales.

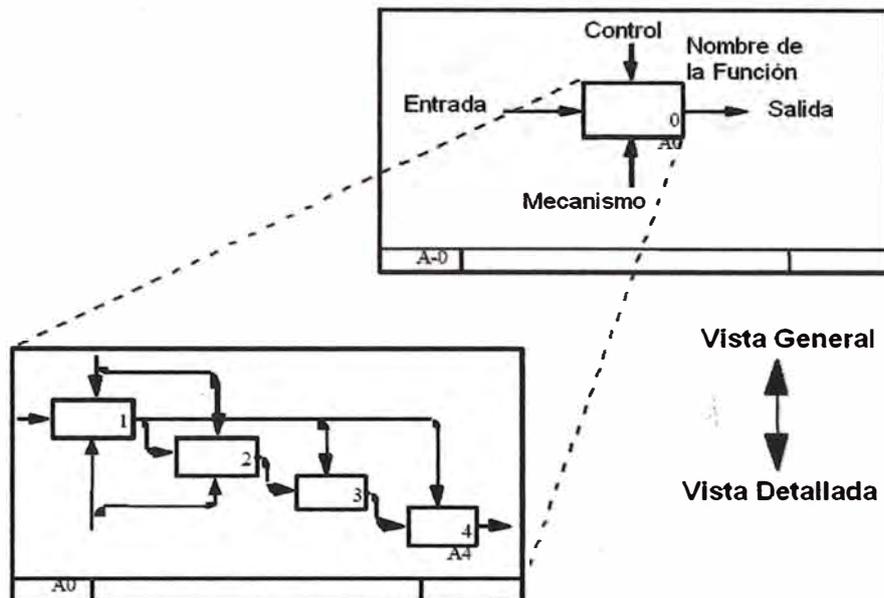


Figura 3.3- Estructura de una función y sus detalles

La Figura 3.4 representa el nivel 0 del modelo IDEF0 del sistema inteligente MBC con todas sus entradas, controles, mecanismos, salidas y funcionalidades. En la Figura 3.5 se presenta el modelo funcional del sistema inteligente MBC detallado en dos actividades, el servidor (*I-kernel*) y el lado cliente (*Cliente web*).

La actividad *I-kernel* tiene como entrada datos *online* e históricos, y salidas OTM (Orden de Trabajo de Mantenimiento) vía email e informaciones solicitadas por el cliente *web* (variables, diagnósticos, pronósticos, toma de decisión, etc.) que sirven como entrada de datos para los clientes.

La actividad Cliente *web* tiene como salida a través de una interface gráfica, datos de diagnósticos, pronósticos, toma de decisión, visualización gráfica de variables *online* e históricas, KPI y FMEA.

### 3.2.1 Actividad *I-kernel*

El modelo IDEF0 de la actividad *I-kernel* es presentado en la Figura 3.6, constituida por ocho funciones:

1. Adquisición de datos: coleta datos *online* e históricas provenientes de instrumentos vía servidor OPC y banco de datos respectivamente;
2. Procesamiento de señal: verifica la calidad de los datos adquiridos;
3. Monitoreo de condición: evalúa la condición de operación de un equipo;
4. Evaluación de salud: diagnósticos de defectos y fallas;
5. Pronósticos: predicción de posibles fallas;
6. Toma de decisión: genera sugerencias de acciones de mantenimiento;
7. Almacenamiento de Informaciones: almacena las anomalías diagnosticadas, toma de decisión y variables de proceso;
8. Módulo de comunicación servidor-cliente: envía informaciones solicitadas por el cliente.

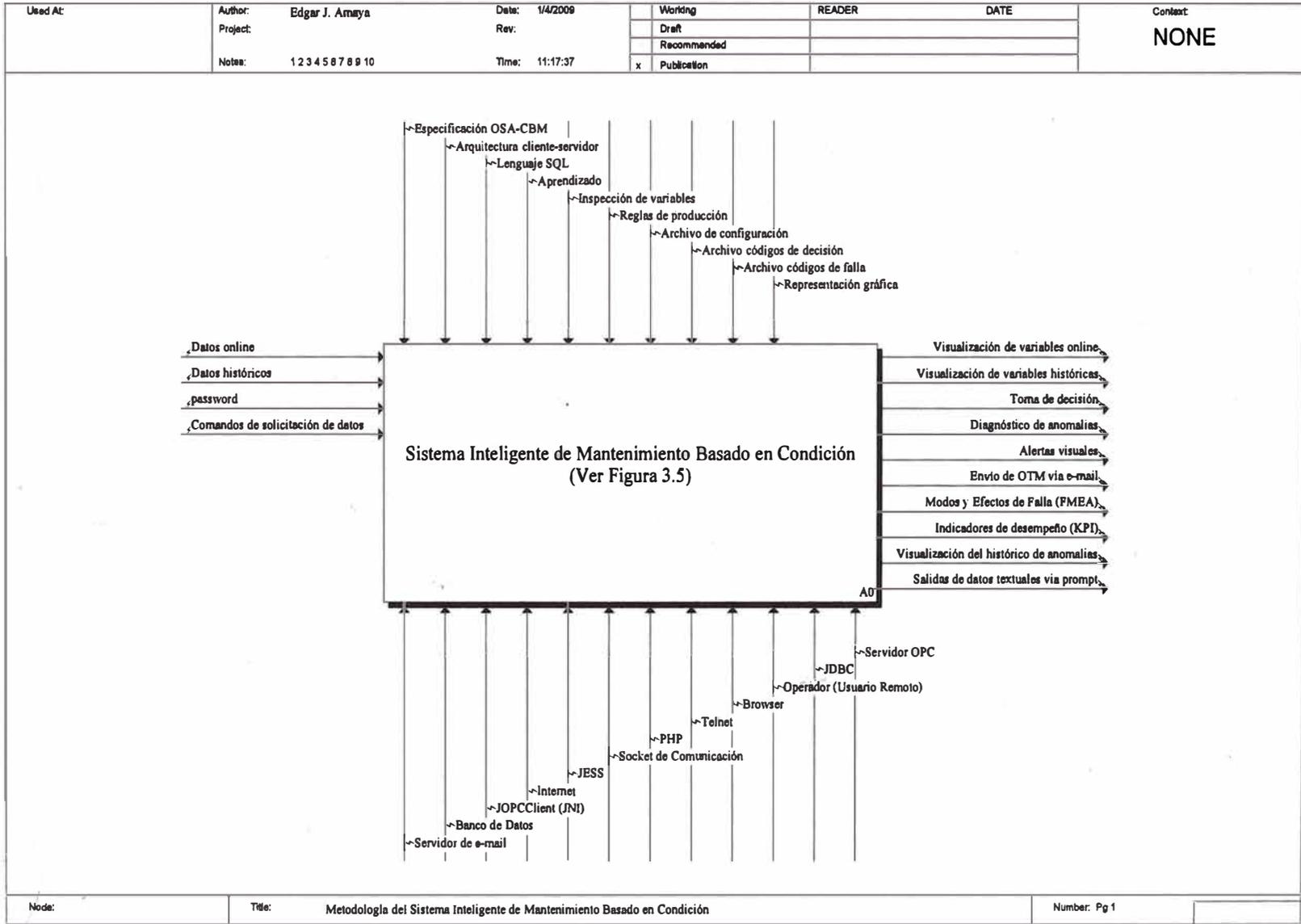


Figura 3.4- Diagrama A0 del sistema inteligente de MBC.

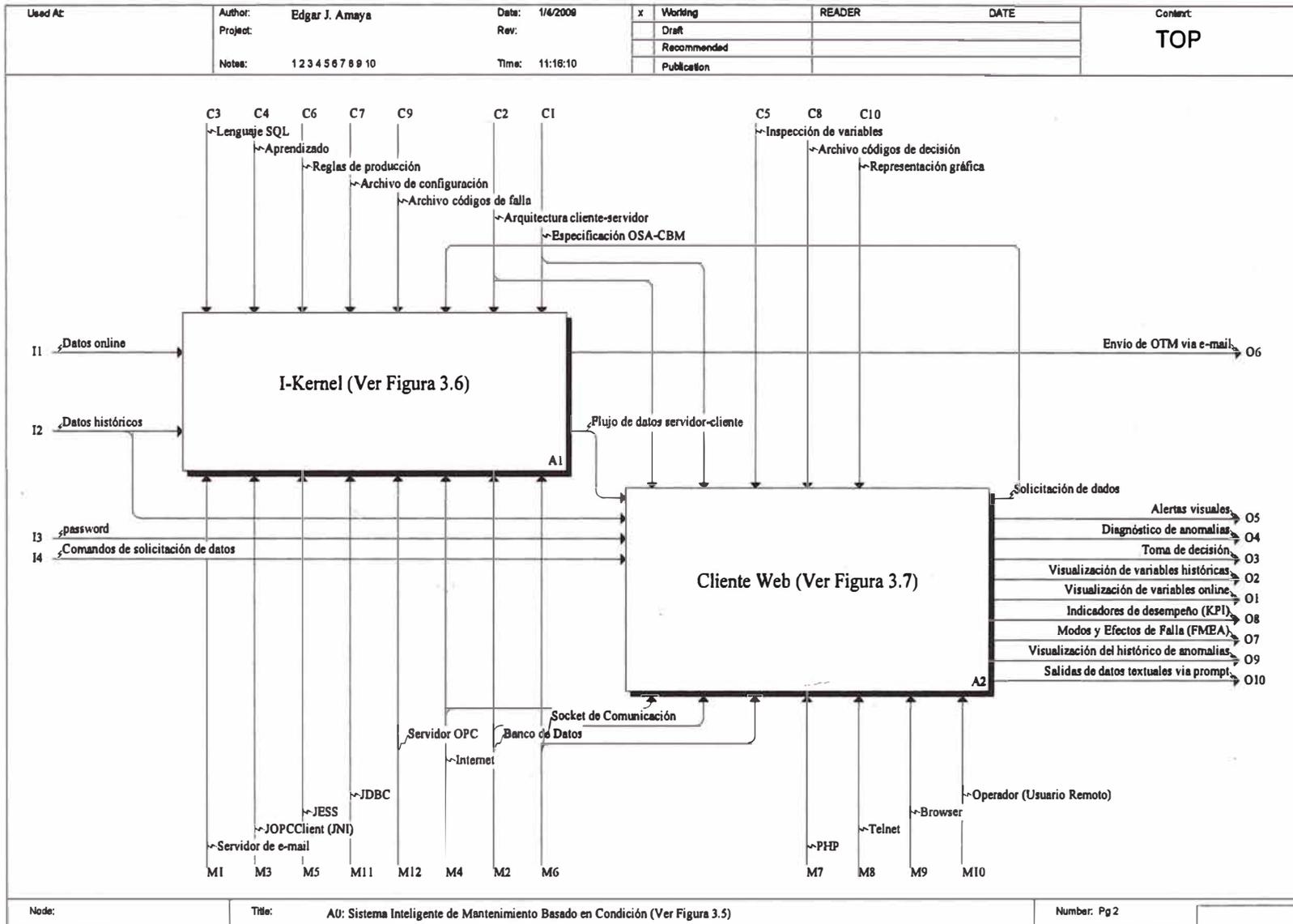


Figura 3.5- Diagrama derivado de la función A0.

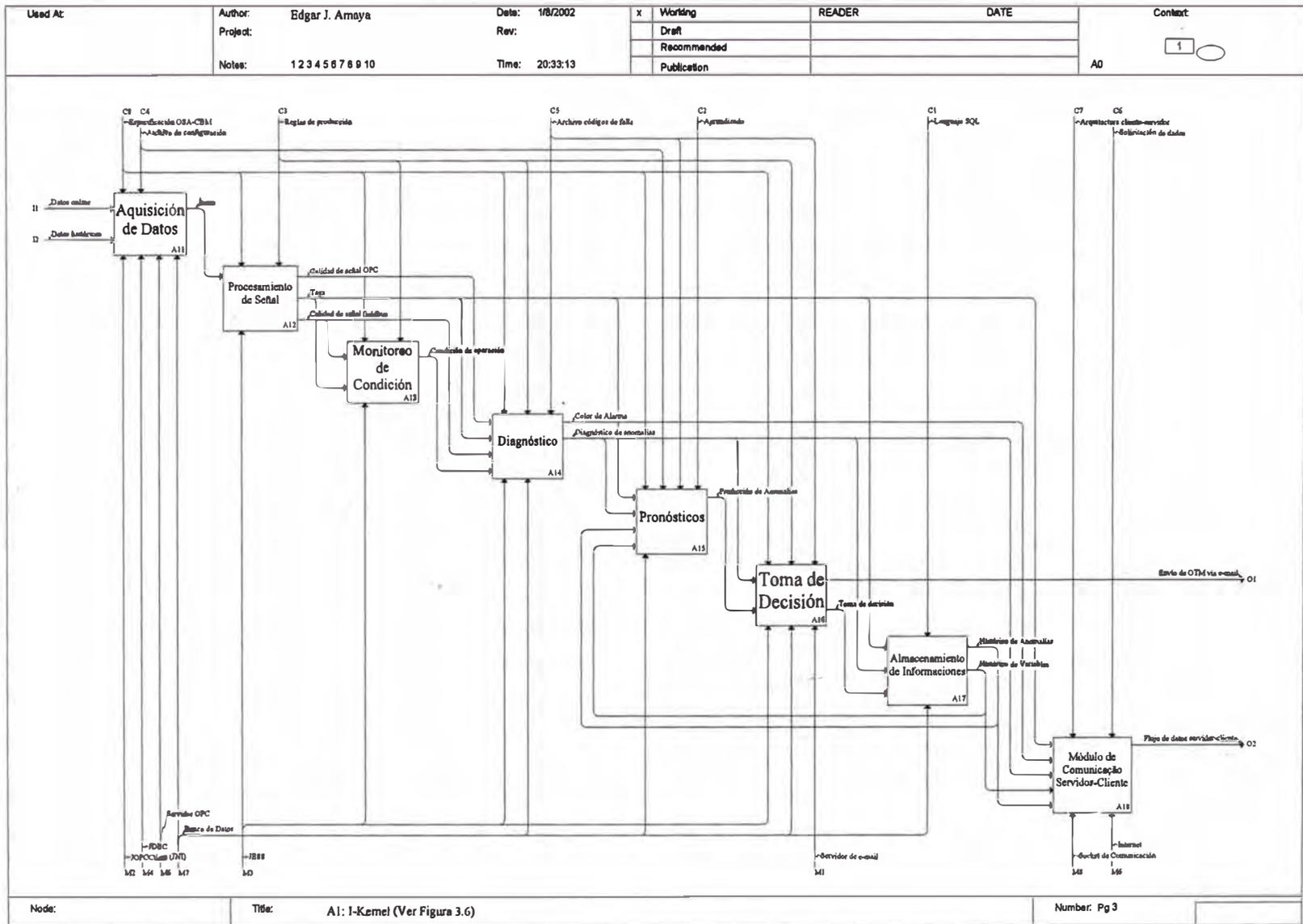
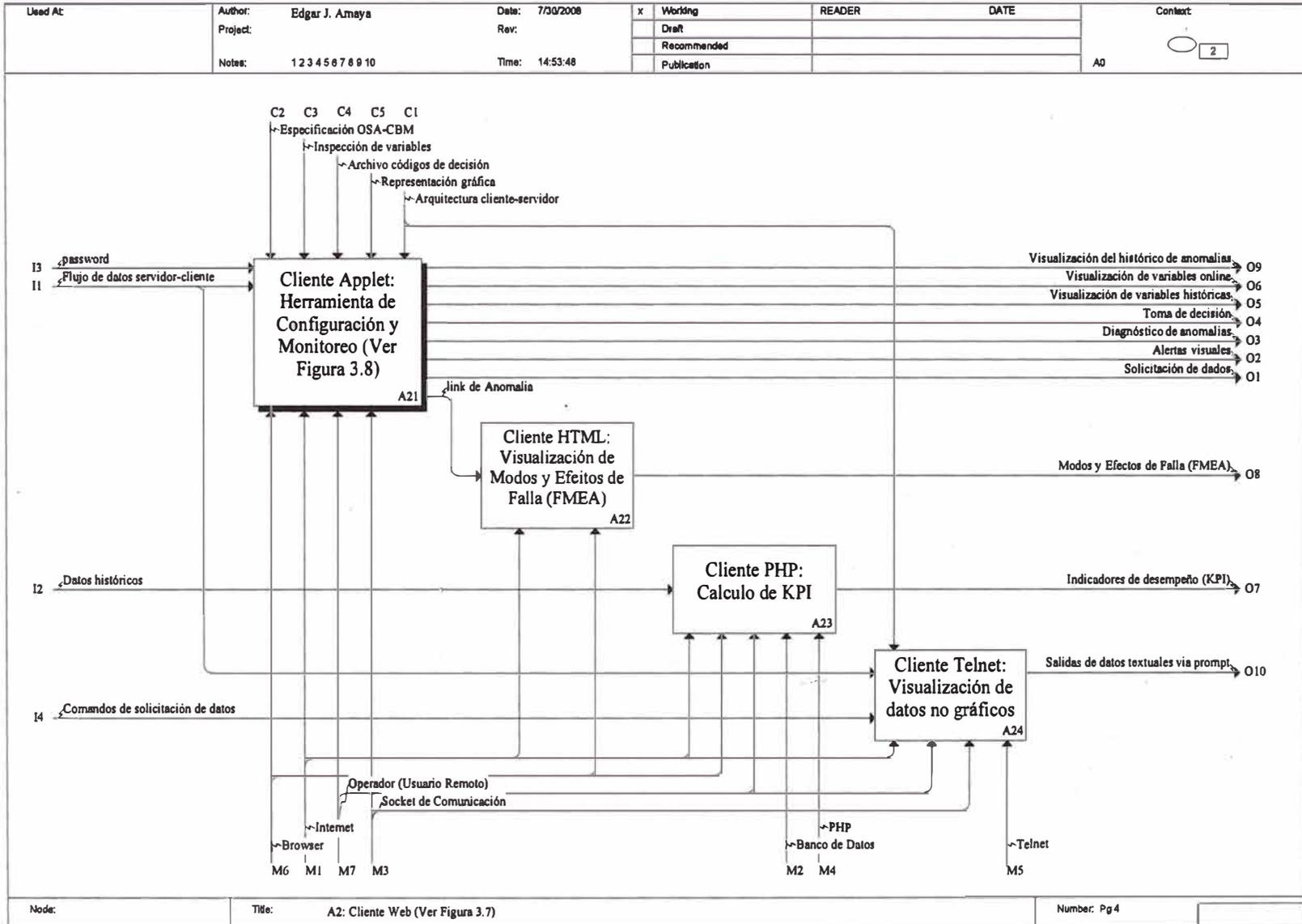


Figura 3.6- Diagrama derivado de la función A1.



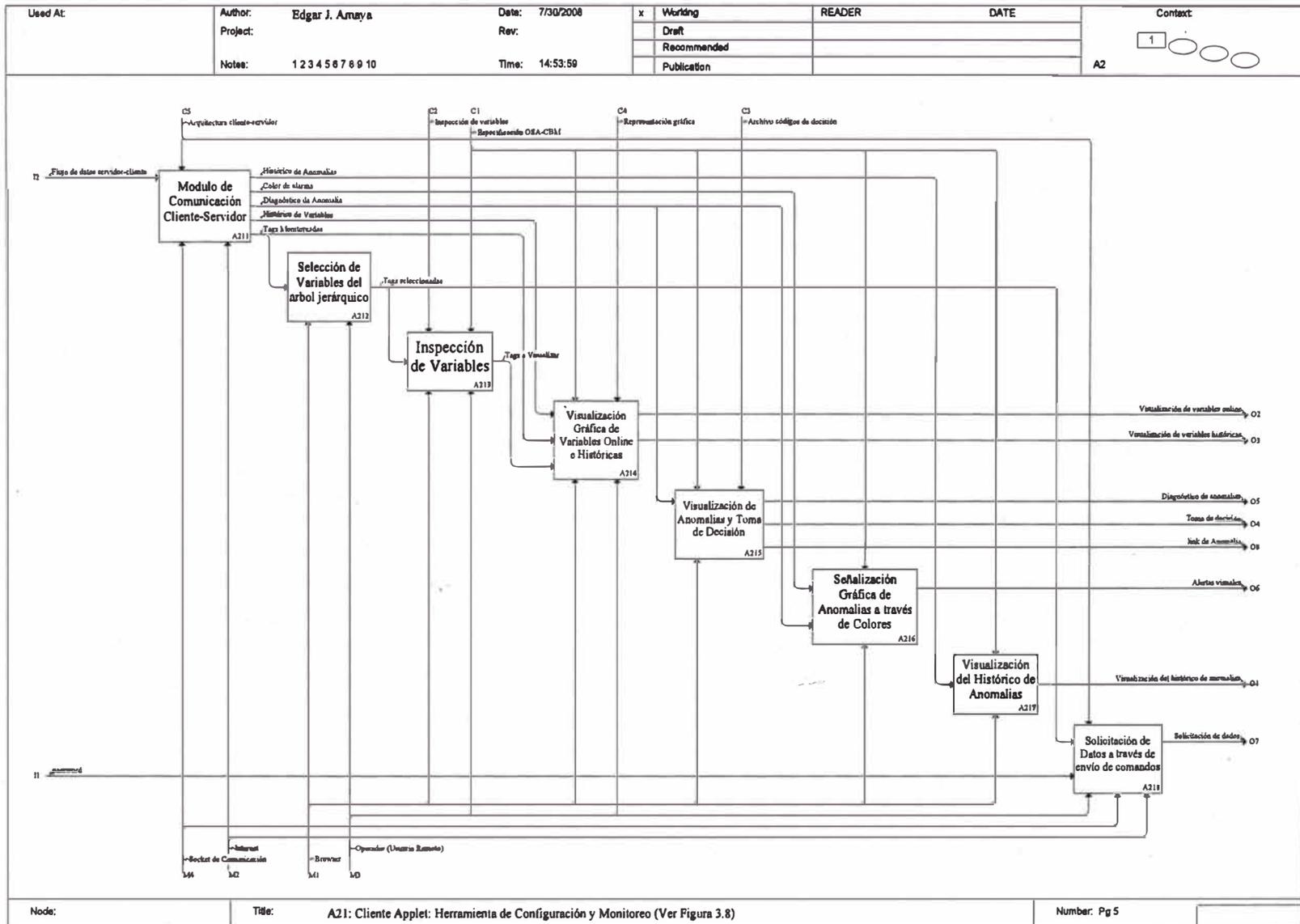


Figura 3.8- Diagrama derivado de la función A21.

### 3.2.2 Actividad cliente web

Esta actividad es mostrada en la Figura 3.7 y es constituida por cuatro actividades para presentar resultados generados por el servidor *I-Kernel*:

1. Cliente *Applet* - Herramienta de Configuración y Monitoreo: cliente desarrollado en Java que recibe informaciones del servidor *I-kernel* y las muestra al usuario a través de otras actividades.
2. Cliente HTML – Visualización de modos de falla y sus efectos (FMEA): se usa para explorar mediante un *link* los detalles de una anomalía a través del análisis FMEA del equipo;
3. Cliente PHP – Cálculo de KPI: calcula los indicadores claves de desempeño de un equipo, subsistema, sistema y de toda la planta en análisis;
4. Cliente *Telnet* – Visualización de datos no gráficos: solicita datos del servidor a través de comandos de forma manual (via *prompt*).

La GUI proporciona las funcionalidades necesarias para soportar la visualización de posibles anomalías ocurridas en los instrumentos, equipos, subsistemas y sistemas. Esta interface está asociada a menús, opciones de visualización, interfaces de usuario gráfica y textual (vía *prompt*), selección de variables a inspeccionar (*online* o histórico), reiniciar o apagar (*shutdown*) el servidor.

### **3.2.3 Actividad Cliente Applet.**

Esta actividad es mostrada en la Figura 3.8 y llega a ser la camada de presentación de la modelo de referencia OSA-CBM y esta constituida por ocho actividades para presentar los resultados de las camadas anteriores OSA-CBM a través de una interface de usuario:

1. Módulo de comunicación cliente-servidor: recibe informaciones del servidor *I-kernel*, y las disponibiliza a las otras actividades;
2. Selección de variables del árbol jerárquico: permite la selección del árbol jerárquico de activos las variables a inspeccionar;
3. Inspección de variables: muestra parámetros de las variables como valor, calidad y hora;
4. Visualización gráfica de variables *online* e históricas: permite el monitoreo gráfico de variables en tiempo real e históricas;
5. Visualización de anomalías y toma de decisión: muestra las anomalías ocurridas y las sugerencias de acciones de mantenimiento;
6. Señalización gráfica de anomalías a través de colores: señal gráfico para localizar el equipo con funcionamiento anormal;
7. Visualización del histórico de anomalías: muestra las treinta últimas ocurrencias de defectos y fallas;
8. Solicitación de datos a través del envío de comandos: solicita datos del servidor a través de comandos.

### 3.3 MODELO DE DATOS IDEF1X

El método IDEF1X (*Integration Definition for Information Modeling*) es utilizado para desarrollar el modelo de información del sistema inteligente MBC. Las características básicas del lenguaje IDEF1X son: entidades, atributos y relacionamientos (Kern, 1999). A partir del modelo de información representado por el modelo entidad relacionamiento y por la normalización de los datos es posible generar el modelo físico del banco de datos relacional (Oliveira, 2002). El sistema de gerenciamiento de base de datos MySQL es utilizado como herramienta de desarrollo.

El modelo de información es dividido en dos dominios. Las informaciones asociadas a la configuración del sistema (servidores OPC, Banco de Datos y e-mail; dispositivos DFI, configuraciones del *I-Kernel*, Lista de e-mails, Tags OPC, Tags del Banco de Datos y Tags Simuladas), estas informaciones son utilizadas cuando se inicia el *I-kernel*. Informaciones asociadas al banco de datos (anomalías, decisiones, equipos, tags, registro de funcionarios y *logs*) son implementadas físicamente a través de un banco de datos constituido por treinta y tres tablas.

En el modelo de configuración mostrado en la Figura 3.9, los datos de las variables a adquirir son agrupados en la entidad Tags. Existen tres tipos de entidades Tags: tags OPC (OPCTags), tags simuladas (SIMTags) y tags de banco de datos (DBTags). Las tags OPC están relacionadas a un servidor OPC (entidad OPCServer) y a un dispositivo DFI (entidad DFIDevice). Las tags de banco de datos están relacionadas a un servidor de banco de datos (atributo DBServer). La lista de emails

es organizada en grupos y almacenada en la entidad Emails, el envío de emails es efectuado vía un servidor de emails (EmailServer). Las configuraciones básicas del *I-Kernel* están contenidas en la entidad *IKernelConfiguration*.

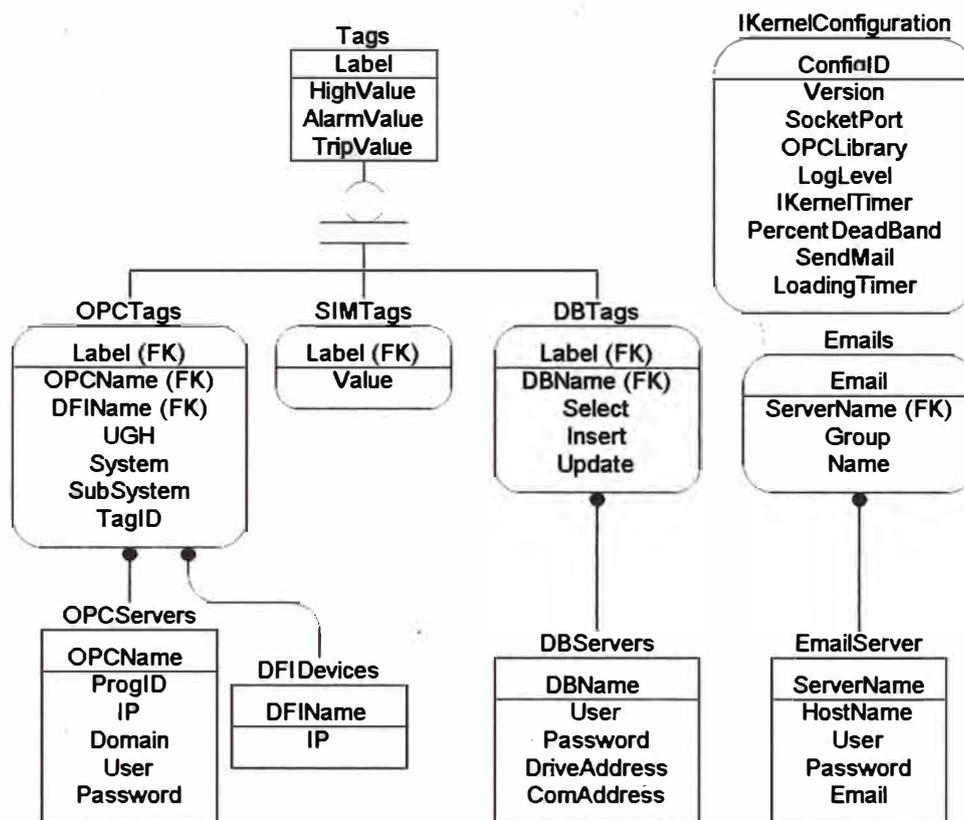


Figura 3.9- Modelo de información IDEF1X de configuración.

En el modelo de configuración mostrado en la Figura 3.9, los datos de las variables a adquirir son agrupados en la entidad Tags. Existen tres tipos de entidades Tags: tags OPC (OPCTags), tags simuladas (SIMTags) y tags de banco de datos (DBTags). Las tags OPC están relacionadas a un servidor OPC (entidad OPCServer) y a un dispositivo DFI (entidad DFIDevice). Las tags de banco de datos están relacionadas a un servidor de banco de datos (atributo DBServer). La lista de emails

es organizada en grupos y almacenada en la entidad Emails, el envío de emails es efectuado vía un servidor de emails (EmailServer). Las configuraciones básicas del *I-Kernel* están contenidas en la entidad *IkernelConfiguration*.

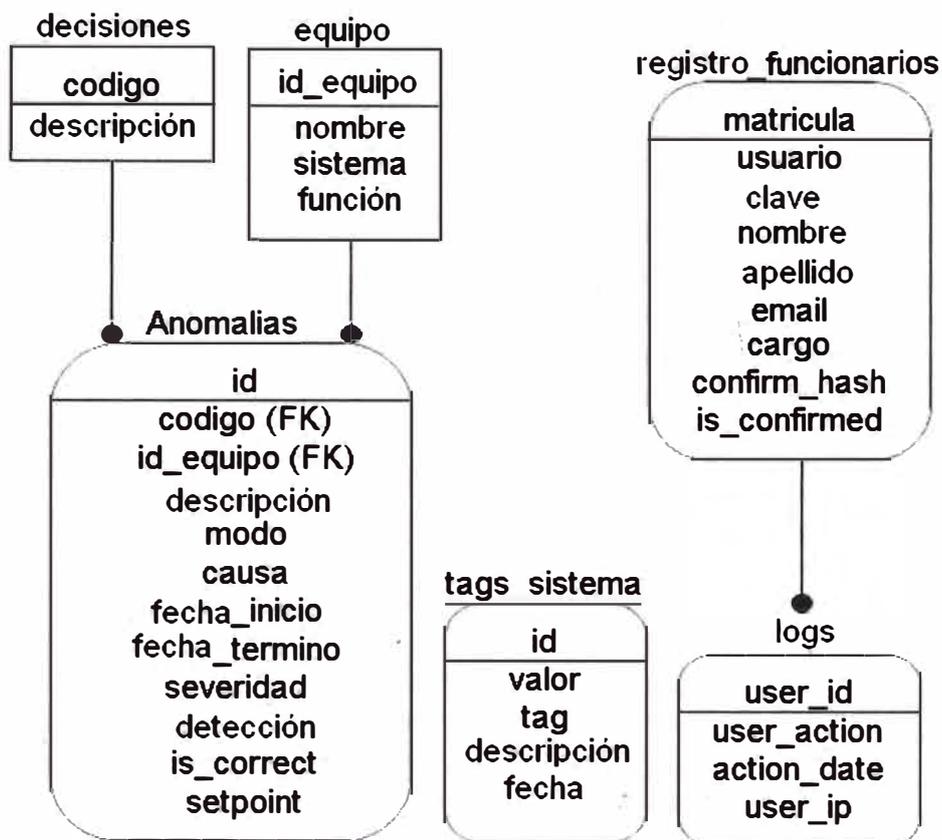


Figura 3.10- Modelo de información IDEF1X del banco de datos.

En la Figura 3.10 es ilustrado el modelo de información de banco de datos. En la entidad anomalías son registradas todas las anomalías detectadas por el sistema inteligente MBC. Cada anomalía posee un atributo *descripción* el cual define la anomalía presentada. El atributo *modo* indica el código de la anomalía. La *causa* describe la situación que llevo al equipo hasta la anomalía.

*fecha\_inicio* registra el día y la hora en que ocurrió una anomalía. La fecha y hora en que el equipo regrese a su funcionamiento normal debe ser registrado por el usuario en el atributo *fecha-termino*. El atributo *severidad* es un índice que refleja la gravedad de las consecuencias de una falla, los valores están representados en el Apéndice C. El atributo *detección* es un índice basado en la estimación de la probabilidad de una falla a ser detectada, asumiendo que ella ocurrió, los valores son mostrados en el Apéndice C.

La función del atributo *is-correct* es dar a conocer si una falla detectada por el sistema es correcta, y ayudará a conocer el porcentaje de acierto, este campo debe ser registrado por el usuario. El atributo *setpoint* es el valor de la variable en operación normal. Los atributos *código* e *id\_equipamento* son declaraciones de llaves extranjeras (*Foreign Key*). Esto es necesario para garantizar que todas las anomalías registradas en la base de datos estén asociadas a una toma de decisión y a un equipo. En la entidad *tags\_sistema* es almacenada el histórico de los valores de las tags. Cuando un usuario se registra, sus datos son almacenados en la entidad *registro\_funcionarios*. Los eventos de acceso de los usuarios al sistema son registrados en la entidad *logs*.

## CAPÍTULO 4

### MODELO UML

En este capítulo se describe el modelo UML del sistema computacional. Se presenta la arquitectura del sistema, los requisitos de usuario, requisitos funcionales y no funcionales. En los requisitos del sistema son presentados los diagramas caso de uso de la aplicación *I-kernel* y la herramienta Configuración y Monitoreo.

#### 4.1 ARQUITECTURA DEL SISTEMA

La arquitectura del sistema es mostrada en la Figura 4.1, donde se presenta el diagrama raíz de todo el sistema, básicamente subdivide la arquitectura en 2 paquetes, *I-Kernel* y *confmonittool*. El paquete *I-Kernel* es el lado servidor de la arquitectura donde se encuentran las primeras seis capas del modelo OSA-CBM. El paquete *confmonittool* es el lado cliente, aquí se encuentra la interface de usuario (camada de presentación del modelo OSA-CBM).

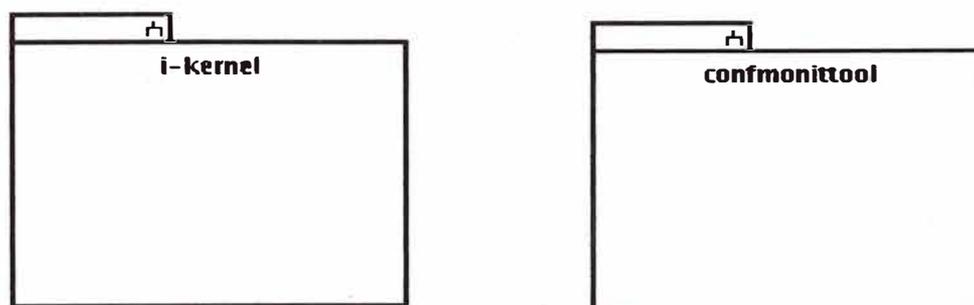


Figura 4.1- Paquetes de la arquitectura SIMPREBAL.

La interacción del paquete *I-kernel* con las fuentes de datos y herramientas es mostrada en la Figura 4.2. La colecta y almacenamiento de información en el banco de datos es vía JDBC. La adquisición de datos de la instrumentación es a través de servidores OPC vía JNI (*Java Native Interface*). En el procesamiento inteligente de los datos adquiridos se usa la herramienta JESS.

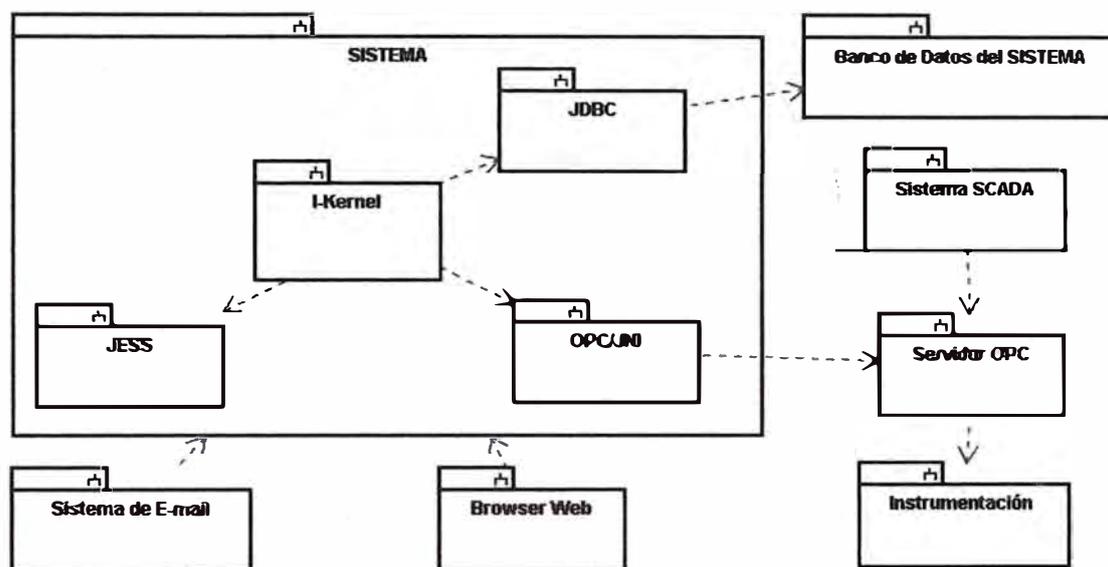


Figura 4.2- El *I-Kernel* y su interacción (Amaya *et al.*, 2007c).

## 4.2 REQUISITOS DE USUARIO

Según Sommerville (2007), una forma de estructurar los requisitos es dividirlos en requisitos de usuario y requisitos de sistema. Los requisitos de usuario capturan las demandas más generales de los usuarios del sistema. Normalmente, estos requisitos son descritos en lenguaje natural y no entran en detalles más técnicos. Los requisitos de usuario son presentados en la forma de una lista de

requisitos funcionales y requisitos no funcionales en lenguaje natural. Los requisitos de sistema son descripciones más detalladas de los requisitos de usuario.

Tabla 4.1- Requisitos Funcionales (RFs) (Amaya *et al.* 2007a).

| RF | Descripción  |
|----|--|
| 1  | El sistema debe tener acceso a los datos de la Planta a partir de los Bancos de Datos utilizados por el Sistema de Monitoreo de la Planta o directamente de la instrumentación, por medio de servidores OPC que disponibiliza informaciones en tiempo real.  |
| 2  | El sistema debe procesar esos datos en forma de un SE basado en reglas de producción.  |
| 3  | El sistema debe alertar al usuario por medio de mensajes de e-mail cuando posibles fallas pudieran ser diagnosticadas.   |
| 4  | El sistema debe alertar al usuario por medio de una alerta visual, cuando posibles fallas pudieran ser diagnosticadas.   |
| 5  | El sistema debe propiciar la edición de un sinóptico conteniendo cierto conjunto de variables monitoreadas, escogidas por el usuario y componiendo una interface de presentación.  |
| 6  | El sistema debe exhibir el valor <i>online</i> de las variables monitoreadas que fueron seleccionadas para componer un determinado sinóptico, presentándolas en una interface propia previamente desarrollada.   |
| 7  | El sistema debe permitir la implementación de algún mecanismo de aprendizaje, de tal forma que los históricos de fallas y tomas de decisiones anteriores puedan ser utilizados para prevenir el surgimiento de nuevas fallas.  |
| 8  | El procesamiento de las informaciones se dará en la forma de un ciclo operativo cerrado, que seguirá la siguiente secuencia: (1) Verificación de los datos a ser adquiridos; (2) Adquisición de datos del banco de datos; (3) Adquisición de datos vía OPC; (4) Almacenamiento provisorio de todos los datos en variables JESS; (5) Para cada una de las N camadas posibles de procesamiento de reglas vía JESS; (6) Actualización de los datos en el banco de datos; (7) Actualización dos datos vía OPC. |

#### 4.2.1 Requisitos funcionales

Según Pressman (1995), los requisitos funcionales de un sistema describen las funcionalidades deseadas por los clientes, lo que se espera que

el *software* realice. Según Campos (2007), el padrón NBR 13596 fue substituida por la ISO/IEC 9126-1, tornándose NBR ISO/IEC 9126-1, que describe un modelo de calidad del producto de *software*, compuesto de dos partes: (1) Calidad interna y calidad externa; y (2) Calidad en uso. De acuerdo con lo expuesto son definidos los requisitos funcionales para el sistema inteligente, esta definición es detallada en la Tabla 4.1.

Tabla 4.2- Requisitos No Funcionales (RNFs) (Amaya *et al.* 2007a).

| RNF | Descripción   |
|-----|---|
| 1   | El sistema debe ser desarrollado en lenguaje de programación Java.  |
| 2   | Las reglas de producción del sistema no deben ser almacenadas directamente en el código fuente, mas deben ser editables y estar disponibles externamente en un archivo modificable.   |
| 3   | El sistema debe poseer una interface <i>web</i> de acceso, por medio de la cual sea posible al usuario configurar el sistema, editar reglas, parámetros y monitorear las variables procesadas por el sistema.   |
| 4   | El sistema debe ser conectable a bancos de datos SQL genéricos, desde que exista un <i>driver</i> JDBC para el respectivo banco de datos.   |
| 5   | El sistema debe ser conectable a servidores OPC, desde que exista un <i>driver</i> JNI.   |
| 6   | El sistema debe utilizar la <i>shell</i> JESS para procesar las reglas en la forma de SE.   |
| 7   | En la camada de pronósticos el sistema debe usar algún método de aprendizaje.   |
| 8   | El sistema debe ser concebido de tal forma que las reglas de producción puedan ser usadas de modo intercambiable para cada una de las camadas de procesamiento (2-procesamiento de señal, 3-monitoreo de condición; 4-evaluación de salud; 6-toma de decisión). |

#### 4.2.2 Requisitos no funcionales

Según Pressman (1995), los requisitos no funcionales comprenden la calidad y restricciones globales del sistema relacionados al mantenimiento,

uso, desempeño, costo, interface, etc. Los requisitos no funcionales son críticos para el suceso de sistemas de *software* y están directamente relacionados con la satisfacción de los usuarios. En la Tabla 4.2 Amaya *et al.* (2007a), presenta los requisitos no funcionales para el sistema inteligente siguiendo una lista detallada de funcionalidades.

### 4.3 REQUISITOS DEL SISTEMA

Los requisitos del sistema son detallados de manera más técnica, utilizándose diagramas UML (Gudwin, 2006). La metodología de desarrollo utilizada es una adaptación del proceso unificado de modelamiento (Jacobson *et al.* 1999). Los diagramas de casos de uso en UML son utilizados para describir el uso de un sistema por actores (Schneider y Winters, 1998, y Booch *et al.*, 1999).

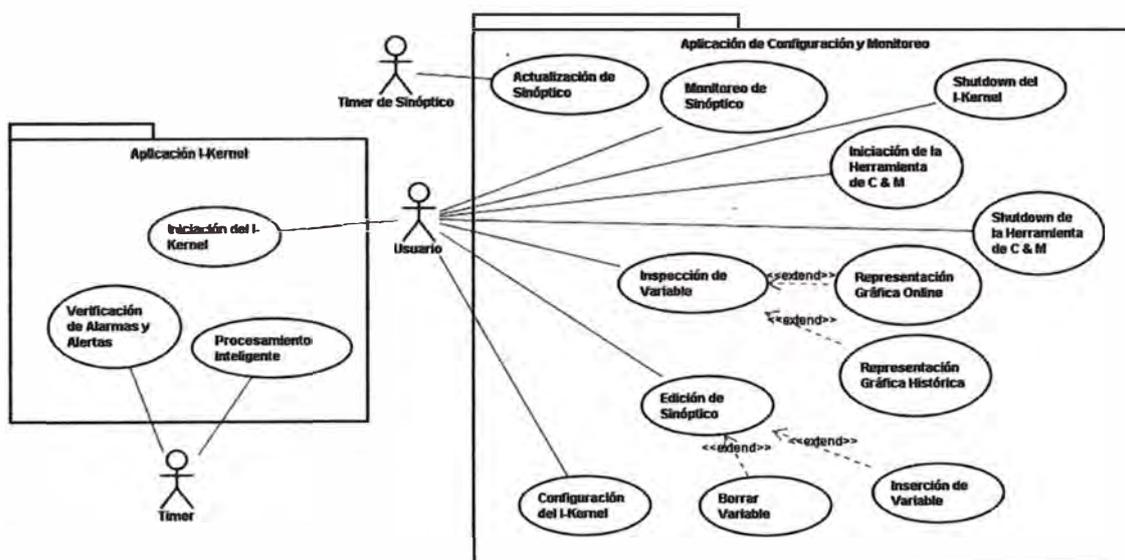


Figura 4.3- Caso de uso del sistema inteligente.

Un actor representa cualquier elemento externo que interactúa con el sistema.

Un caso de uso describe una secuencia de pasos/operaciones que un usuario realiza

cuando interactúa con un sistema tratando de realizar una determinada tarea. No en tanto, la descripción de casos de uso no trata la cuestión de como esta interacción será implementada. Fases posteriores a la etapa de ingeniería de requisitos tales como proyecto e implementación desarrollaran este aspecto.

En la Figura 4.3, tenemos un diagrama UML de casos de uso, que nos propicia una visión general del sistema (Gudwin, 2006). El sistema es compuesto de dos aplicaciones que son ejecutadas de manera independiente: el *I-Kernel* y la herramienta de C&M.

#### **4.4 CASOS DE USO DE LA APLICACIÓN *I-KERNEL***

La aplicación *I-Kernel* es un servidor que no posee interface de usuario, por ello, lo único que el usuario puede hacer es iniciar la aplicación. Además de eso, un temporizador previamente programado que inicia el ciclo de procesamiento inteligente y de la misma forma realiza la verificación de alarmas y alertas.

##### **4.4.1 Iniciación del I-kernel**

El diagrama de caso de uso para iniciación de la aplicación *I-kernel* es presentado en la Figura 4.4. En este caso de uso, el usuario solicita al sistema operacional que la aplicación *I-kernel* sea cargada, y el sistema operacional carga la aplicación. El ciclo se inicia leyendo el archivo de configuración (Apéndice A), donde están todos los parámetros para el funcionamiento del sistema. En la secuencia define el período para generación de alarmes (*AlarmTimer*) y el período de procesamiento (*IKernelTimer*). El siguiente

paso es establecer la comunicación con el banco de datos y con el servidor OPC. Los últimos pasos son la realización del ciclo de inferencia del JESS que es responsable por el procesamiento inteligente. Este ciclo es repetido hasta que el usuario apague (*shutdown*) la aplicación *I-kernel*.

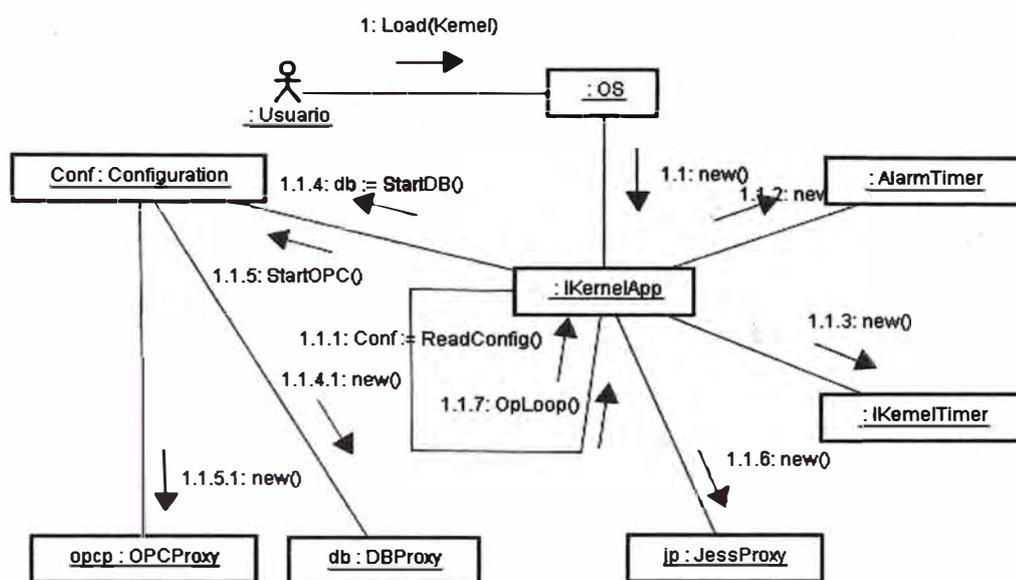


Figura 4.4- Iniciación de la aplicación *I-Kernel*.

#### 4.4.2 Procesamiento inteligente

El procesamiento inteligente es representado en diagrama de caso de uso en la Figura 4.5. El temporizador de procesamiento inteligente iniciado en el caso de uso “Iniciación del *I-kernel*”, envía *ticks* de reloj periódicos. El período depende de la cantidad de variables, de las reglas y de la capacidad de procesamiento de la computadora, en el software desarrollado se usa el valor de 1 minuto. El ciclo de procesamiento inteligente inicia colectando datos (*ReadTags*) del banco de datos y del servidor OPC, en seguida actualiza las

tags (*RefreshTags*) y ejecuta el ciclo de inferencia (*ExecuteCycle*) del SE, las tags obtenidas son almacenadas en el banco de datos (*WriteTags*).

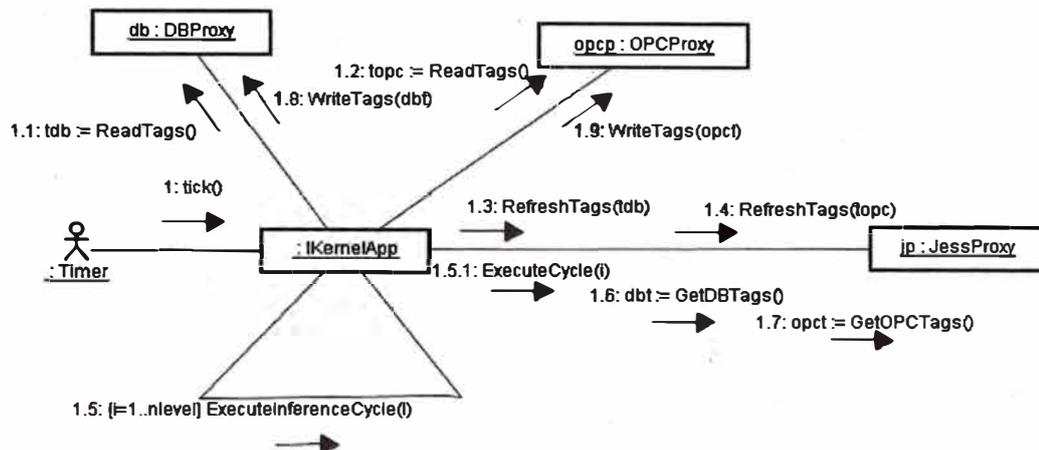


Figura 4.5- Procesamiento inteligente.

#### 4.4.3 Verificación de alarmas y alertas

Este caso de uso es ilustrado en la Figura 4.6, el temporizador de verificación de alarmas y alertas envía un *tick* de reloj, que procede la verificación de un conjunto de variables monitoreadas, y caso alguna de ellas esté marcada, procede al tipo de alarma o alerta asociado, lee los datos del archivo de configuración (*i.e.* lista de e-mails), y envía OTM vía e-mail (*Mailer*). Los códigos de colores, las anomalías y el histórico de anomalías son enviados para el cliente (herramienta de C&M) a través de la clase *NetworkProxy* y los errores de ejecución son almacenados en el *Logger*.

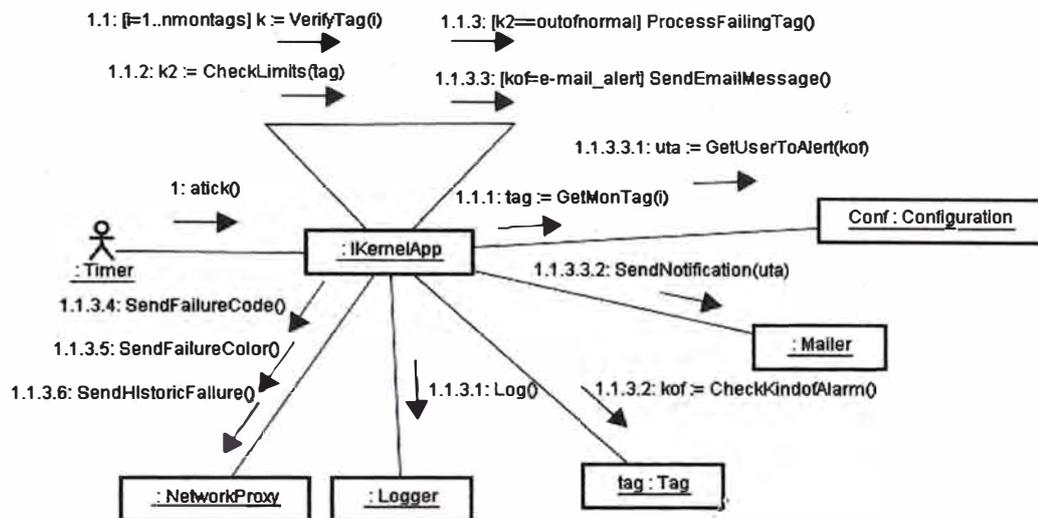


Figura 4.6- Verificación de Alarmas y Alertas.

#### 4.4.4 Shutdown del I-kernel.

Una vez que el sistema de C&M ha sido iniciado, el usuario remoto (con clave de acceso) solicita el cerramiento del sistema *I-Kernel*. Si la clave es válida y si el *I-kernel* está operativo, el sistema cierra su operación. Caso contrario, debe alertar al usuario que la clave es inválida o que el *I-kernel* no está operativo. Este procedimiento se muestra en la Figura 4.7.

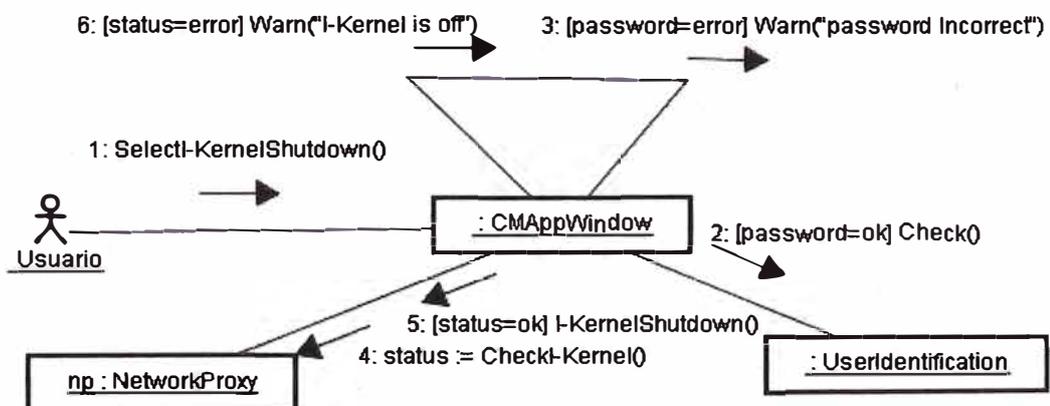


Figura 4.7- *Shutdown* de la aplicación *I-Kernel*.

## 4.5 CASOS DE USO DE LA HERRAMIENTA DE C&M

La herramienta de Configuración y Monitoreo es una aplicación *web*, destinada al monitoreo de las variables de control del sistema y la solicitud de *shutdown* del *I-Kernel*. El monitoreo es una inspección directa del estado de las variables, seleccionando una variable dentro de todas aquellas disponibles. El procedimiento es escoger la variable a ser monitoreada, y el sistema exhibe su estado directamente. Este tipo de monitoreo efectúa la inspección combinada de las variables críticas del sistema, visualización gráfica *online* e histórica.

### 4.5.1 Iniciación de la herramienta C&M

En este caso de uso, el usuario carga en un *browser Web* una URL que corresponde a la dirección de la herramienta C&M. A partir de allí, el sistema debe iniciar la herramienta C&M y abrir su ventana principal de interacción con el usuario. La Figura 4.8 presenta el procedimiento de iniciación de la herramienta C&M, y el cargamento de todas las clases y ventanas a usar.

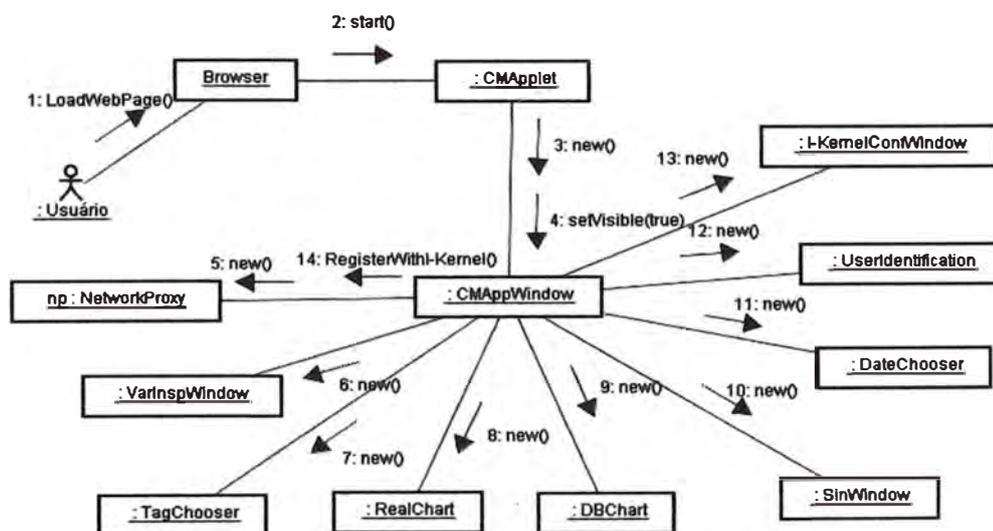


Figura 4.8- Iniciación de la herramienta de C&M.

#### 4.5.2 Monitoreo de sinóptico

Este caso de uso es proyectado para que el usuario obtenga un conjunto de variables en una ventana definida. El sistema abre la ventana de monitoreo e inicia el temporizador que hará la actualización de la interface. Después de eso, el sistema queda esperando que el usuario solicite el cerramiento de la ventana, y cuando eso ocurre, el borra el temporizador y cierra la ventana de monitoreo. La función de monitoreo de sinóptico es presentada en la Figura 4.9.

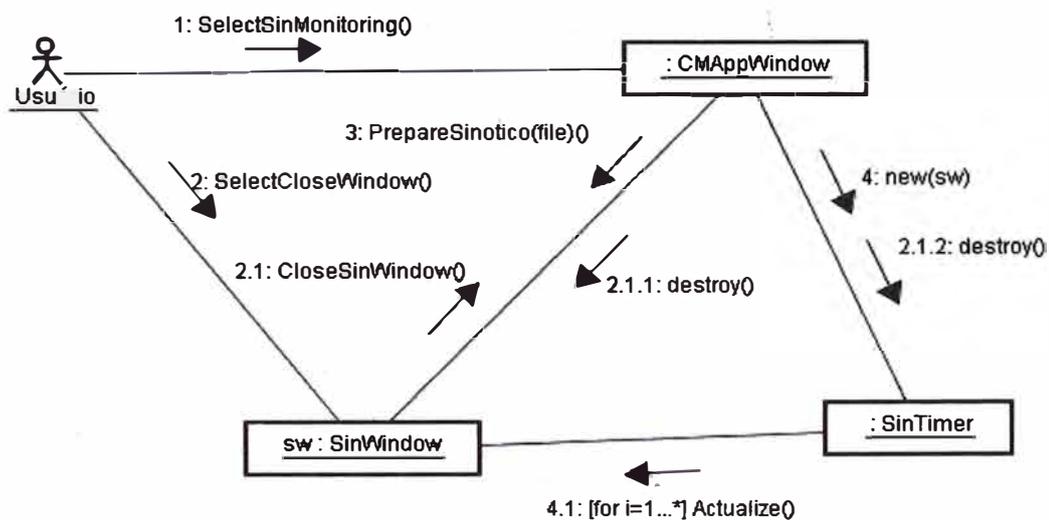


Figura 4.9- Monitoreo de Sinóptico.

#### 4.5.3 Actualización de sinóptico

Cuando una ventana de sinóptico está siendo monitoreada, ella está asociada a un temporizador de Sinóptico que envía *ticks* periódicos de reloj. A cada *tick*, el *configmonittol* solicita al *I-kernel* las variables del sinóptico, actualiza sus valores y rediseña la interface para presentar los nuevos valores. Este caso de uso puede ser visualizado en la Figura 4.10.

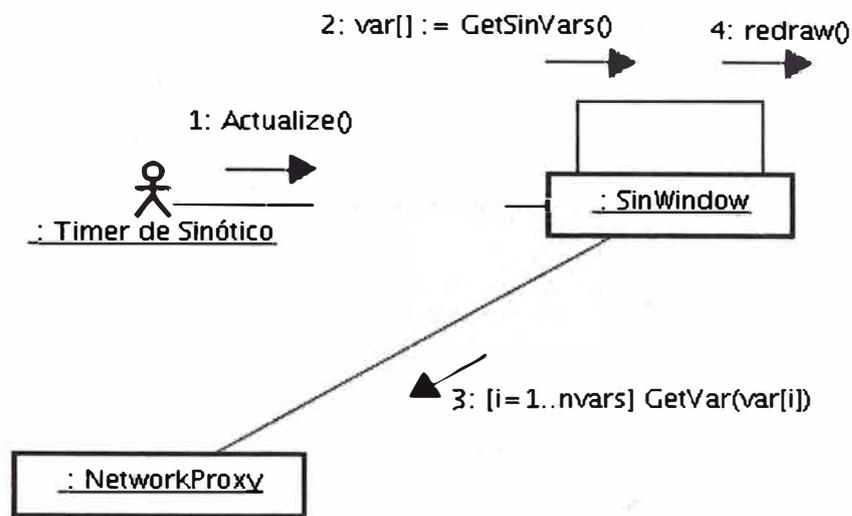


Figura 4.10- Actualización de sinóptico.

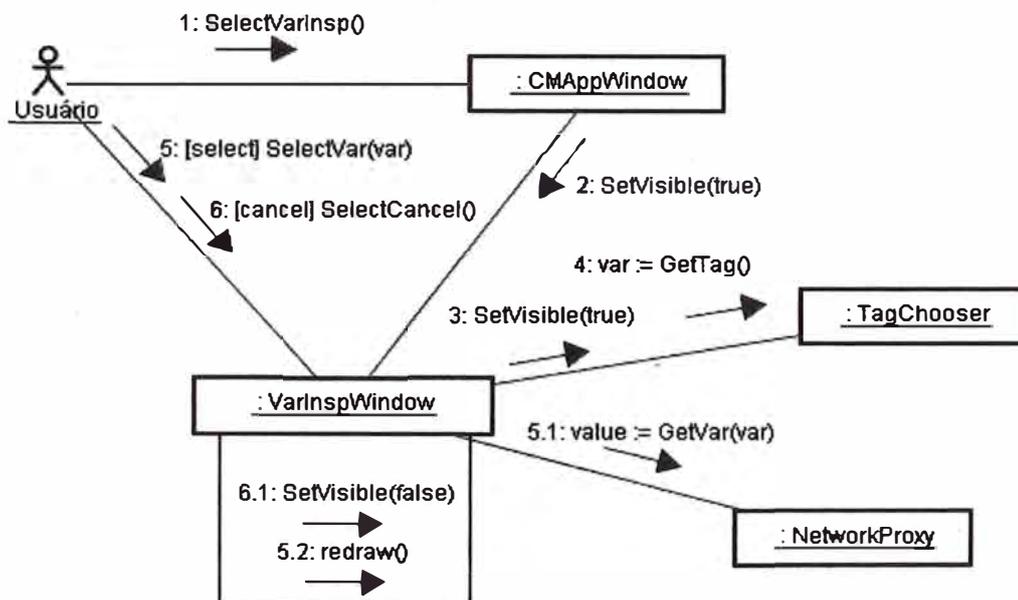


Figura 4.11- Inspección de variables.

#### 4.5.4 Inspección de variables

En este caso de uso, el usuario tiene la opción de inspeccionar una o más variables. El usuario selecciona la inspección de variables, y el sistema

abre una ventana de selección de tags (*TagChooser*) en forma de árbol jerárquico. En ella, el usuario puede seleccionar cual variable desea inspeccionar, y el sistema busca esa variable y exhibe su valor. El usuario puede decidir inspeccionar otra variable o finalizar la operación. Este caso de uso es mostrado en la Figura 4.11.

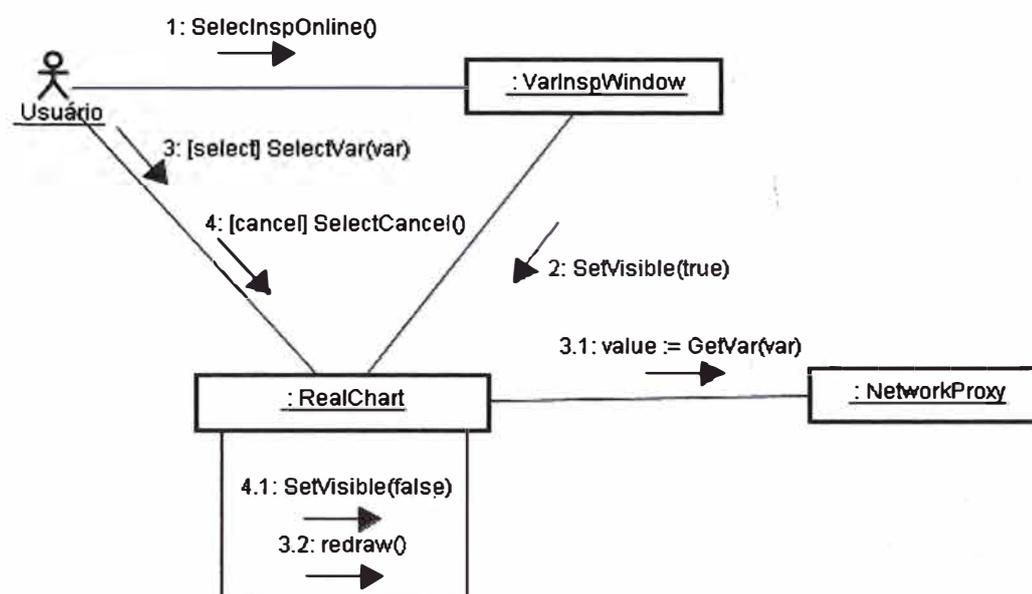


Figura 4.12- Inspección de variables *online*.

#### 4.5.5 Inspección de variables *online*

Este caso de uso es mostrado en la Figura 4.12, el usuario tiene la opción de visualizar gráficamente el valor *online* de las variables. El usuario selecciona las variables de la interface inspección de variables, y el sistema busca esas variables y exhibe gráficamente sus valores a través de la clase *RealChart*.

#### 4.5.6 Inspección de variables históricas

En este caso de uso, el usuario tiene la opción de visualizar gráficamente el valor histórico de las variables. El usuario selecciona las variables de la interface inspección de variables y el sistema solicita al usuario definir la fecha y hora de inicio y termino de inspección. El sistema busca esas variables en el banco de datos y exhibe gráficamente a través de la clase *DBChart*. Este caso de uso es mostrado en la Figura 4.13.

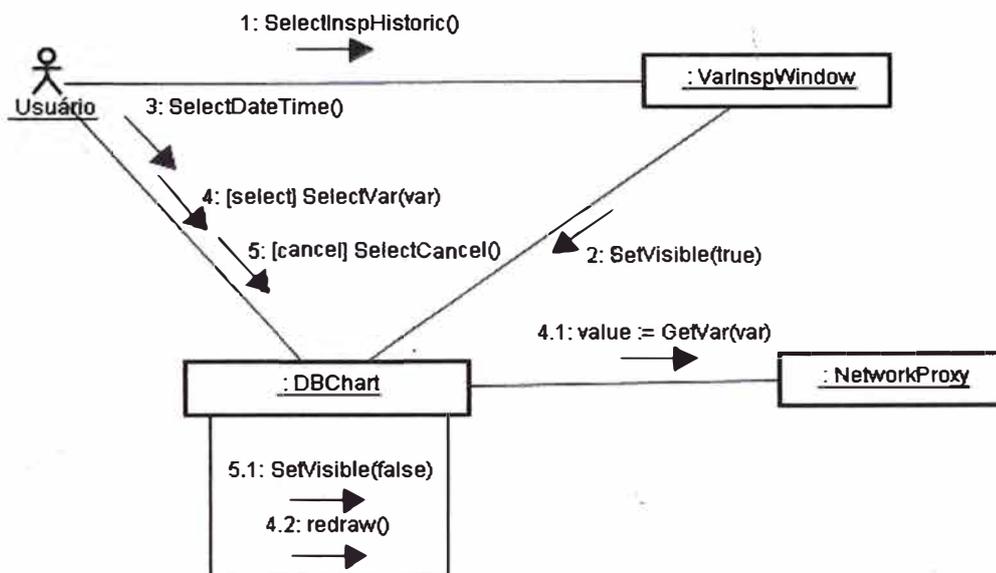


Figura 4.13- Inspección de variables históricas.

#### 4.5.7 Shutdown de la herramienta C&M

Una vez que la herramienta C&M esté abierta y operativa, el usuario solicita que la misma sea cerrada. A partir de allí, el sistema debe cerrar la herramienta, la ventana principal y todas las otras ventanas usadas en la aplicación. El *shutdown* de la herramienta C&M no implica el cerramiento de

la aplicación *I-Kernel*, él puede continuar en operación. El procedimiento de *shutdown* de la herramienta C&M es presentado en la Figura 4.14.

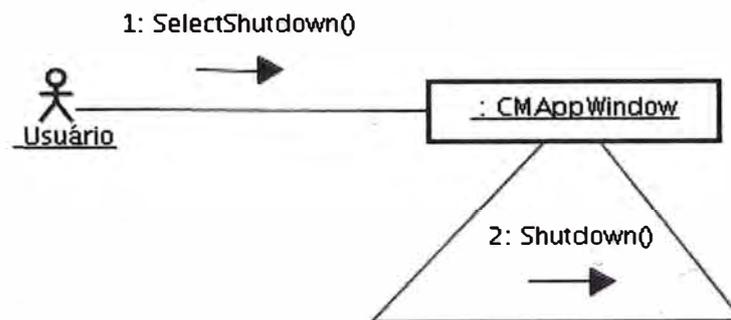


Figura 4.14- *Shutdown* de la herramienta C&M.

## **CAPÍTULO 5**

### **IMPLEMENTACIÓN COMPUTACIONAL**

En este capítulo se describe el proceso de implementación computacional del SIMPREBAL basado en la metodología desarrollada. Son presentados los requisitos físicos, los modelos de las reglas de producción y las clases implementadas tanto en el lado servidor cuanto en el lado cliente.

#### **5.1 REQUISITOS FÍSICOS**

El SIMPREBAL es uno de los sistemas que hace parte de un proyecto mayor llamado “modernización del área de automatización de procesos de las centrales hidroeléctricas de Balbina y Samuel”. La primera fase fue ejecutada en Balbina utilizándose opciones tecnológicas actuales para monitoreo de las medidas de los activos de generación y transmisión a través del uso de instrumentación inteligente basadas en tecnología FF, sistema de control y supervisión (operación) basado en tecnología Rockwell, sistema MES de la Rockwell, el SIMPREBAL y el sistema de planeamiento de Mantenimiento MAXIMO. El objetivo de utilizar estos sistemas es para contribuir en la mejoría de los índices de desempeño: MTBR (*Mean Time Between Repairs*), calidad de energía, mantenibilidad y confiabilidad.

En la implementación computacional de la metodología presentada anteriormente, se propone el desarrollo de un subsistema de adquisición de datos y procesamiento inteligente que será llamado *I-Kernel*. El subsistema *I-Kernel* apoya la construcción del sistema inteligente MBC, y su concepción es la de un subsistema genérico, a ser desarrollado en lenguaje Java, capaz de realizar adquisición de datos a partir de bancos de datos y de instrumentos vía servidor OPC, su procesamiento por medio de reglas de producción.

Para el desarrollo del sistema inteligente propuesto en este trabajo es necesaria una red industrial, interconectando el sistema con los instrumentos de campo. Los instrumentos usados para monitorear las condiciones de las máquinas son mostrados en la Figura 5.1. En la implementación se usa la instrumentación *fieldbus*, donde los instrumentos son agrupados en una red por canales y conectados a una DFI. Cada DFI tiene cuatro canales de comunicación, generalmente uno es de reserva.

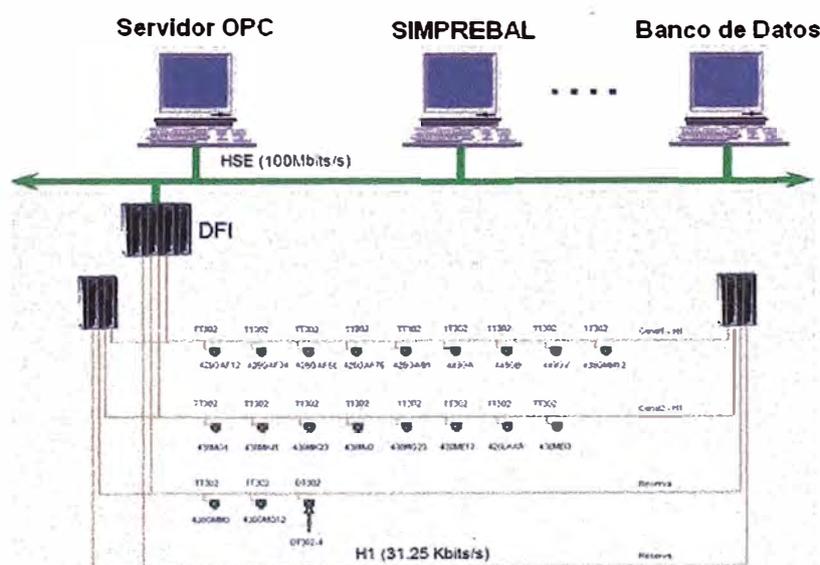


Figura 5.1- Instrumentación y equipos.

En la Figura 5.2 se muestran una serie de computadores, donde diferentes servicios están instalados. A pesar de mostrar diferentes máquinas, se podría asumir, sin perjuicio de concepto, que estos servicios estén instalados todos en una misma máquina. La única restricción que se presenta en este sentido es que tanto el módulo *Confmonittool* como el módulo *I-Kernel* necesariamente necesitan estar instalados en la misma máquina. Esa restricción ocurre, pues el módulo *Confmonittool* será un *Applet* Java que utilizará la red para comunicarse con el módulo *I-Kernel*.

Por restricciones de seguridad un *Applet* Java sólo puede comunicarse con la misma máquina en que fue cargado, acaba siendo necesario que el módulo *I-Kernel* sea instalado en la misma máquina donde se encuentra el servidor *web*, pues caso contrario el *Applet* no conseguirá comunicarse con el módulo *I-Kernel*.

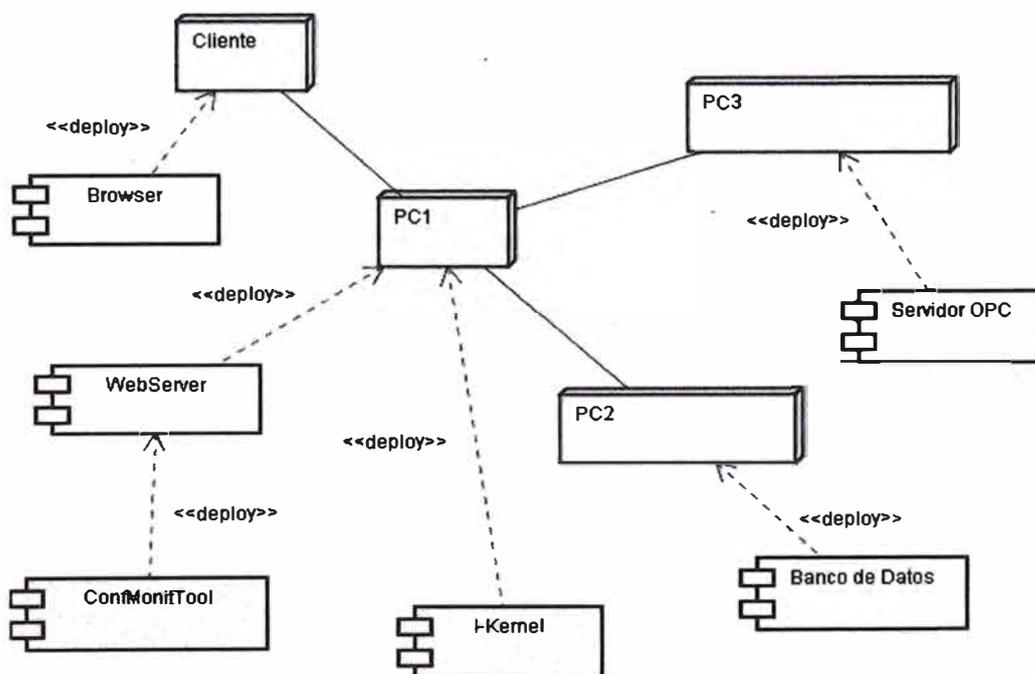


Figura 5.2- Requisitos físicos del SIMPREBAL.

## 5.2 ESTRUCTURA DE REGLAS DE PRODUCCIÓN

A seguir se presentan las estructuras de reglas de producción para las capas OSA-CBM usados en la implementación del SIMPREBAL. Los modelos de reglas desarrolladas son para las capas procesamiento de señal, monitoreo de condición, evaluación de salud y toma de decisión.

### 5.2.1 Estructura de reglas de procesamiento de señal

En esta capa es implementado dos tipos de procesamiento. El procesamiento de señal OPC y el procesamiento de señal *fielbus*. Los valores de los campos de la clase Tag usado en el procesamiento de reglas en esta capa y en las siguientes son calculados según el Apéndice B.

Tabla 5.1- Modelo de reglas para procesamiento de señal OPC.

| Si  | Entonces         |
|---|------------------|
| $dfi = dfiXY$ y $quality = 3$   | COM-GOOD ?label  |
| ( $dfi = dfiXY$ y $quality = 1$ ) o<br>( $dfi = dfiXY$ y $quality = 0$ y $subquality = 0$ ) o<br>( $dfi = dfiXY$ y $quality = 0$ y $subquality = 3$ ) o<br>( $dfi = dfiXY$ y $quality = 0$ y $subquality = 4$ ) o<br>( $dfi = dfiXY$ y $quality = 0$ y $subquality = 5$ ) o<br>( $dfi = dfiXY$ y $quality = 0$ y $subquality = 6$ ) o<br>( $dfi = dfiXY$ y $quality = 0$ y $subquality = 8$ ) | COM-BAD-0 ?label |
| $dfi = dfiXY$ y $quality = 0$ y $subquality = 1$  | COM-BAD-1 ?label |
| $dfi = dfiXY$ y $quality = 0$ y $subquality = 2$  | COM-BAD-2 ?label |
| $dfi = dfiXY$ y $quality = 0$ y $subquality = 7$  | COM-BAD-3 ?label |

El modelo de reglas para el procesamiento de señal OPC es presentado en la Tabla 5.1, las condiciones de estas reglas son las variables  $dfiXY$ , donde

X es el número de la UGH (X=1, 2, 3, 4 o 5), e Y es la DFI en la cual está conectada el instrumento (Y = a, b o c). Otras condiciones de las reglas son los valores de los campos *quality* y *subquality* de la clase Tag. Los subsecuentes de estas reglas son los 5 códigos acompañados del *label* de la Tag procesada, una de buena comunicación y las otras para la condición de falla en la comunicación.

En el procesamiento de señal *fieldbus*, una condición es que la señal OPC sea de buena calidad. Otras condiciones del modelo de reglas, mostradas en la Tabla 5.2, son los campos de la clase Tag, *status* y *substatus*. Estas reglas generan nuevos casos para el estado de la señal *fieldbus*, acompañado del *label* de la Tag procesada.

Tabla 5.2- Modelo de reglas para procesamiento de señal *fieldbus*.

| Si   | Entonces            |
|--|---------------------|
| (COM-GOOD ?label) y<br>((dfi = dfiXY y status = 2) o<br>(dfi = dfiXY y status = 3))  | signal-GOOD ?label  |
| (COM-GOOD ?label) y<br>((dfi = dfiXY y status = 1) o<br>(dfi = dfiXY y status = 0 y substatus = 0) o<br>(dfi = dfiXY y status = 0 y substatus >= 5)) | signal-BAD-0 ?label |
| (COM-GOOD ?label) y<br>(dfi = dfiXY y status = 0 y substatus = 1)  | signal-BAD-1 ?label |
| (COM-GOOD ?label) y<br>(dfi = dfiXY y status = 0 y substatus = 2)  | signal-BAD-2 ?label |
| (COM-GOOD ?label) y<br>(dfi = dfiXY y status = 0 y substatus = 3)  | signal-BAD-3 ?label |
| (COM-GOOD ?label) y<br>(dfi = dfiXY y status = 0 y substatus = 4)  | signal-BAD-4 ?label |

### 5.2.2 Estructura de reglas de monitoreo de condición

En la Tabla 5.3 se muestran los modelos de reglas usados para el monitoreo de condición de operación. Estas reglas verifican la condición del valor medido por el instrumento. Las condiciones de este modelo de reglas son la comparación del valor medido con sus valores límites de operación  $V_{high}$ ,  $V_{alarm}$ , y  $V_{trip}$ ; y los consecuentes son los cuatro rangos de operación: NORMAL, ALTO, ALARMA y TRIP.

Tabla 5.3- Modelo de reglas para los rangos de operación.

| Si                                   | Entonces                |
|--------------------------------------|-------------------------|
| $value \leq V_{high}$                | 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   |

El modelo de reglas mostrado en la Tabla 5.4 usa como condiciones los casos generados por la Tabla 5.3 y la Tabla 5.2. Con las condiciones presentadas, el modelo de reglas genera como consecuente un código de monitoreo de condición (COD\_MC).

Tabla 5.4- Modelo de reglas para monitoreo de condición.

| Si  | Entonces                |
|---|-------------------------|
| signal-GOOD ?label y<br>condition-NORMAL ?label | condition COD_MC-normal |
| signal-GOOD ?label y<br>condition-HIGH ?label   | condition COD_MC-alto   |
| signal-GOOD ?label y<br>condition-ALARM ?label  | condition COD_MC-alarma |
| signal-GOOD ?label y<br>condition-TRIP ?label   | condition COD_MC-trip   |

### 5.2.3 Estructura de reglas de evaluación de salud

El modelo de reglas para diagnóstico de los canales de comunicación *fieldbus* es mostrado en la Tabla 5.5. Este modelo de reglas tiene como condiciones los códigos generados por el modelo de la Tabla 5.1 acompañado de los *labels* de los instrumentos conectados al canal *fieldbus*. Este modelo genera acciones (*i.e.* diagnostic G1A-canal N-0, N = 1, 2 o 3).

Tabla 5.5- Modelo de reglas para diagnóstico de canales *fieldbus*.

| Si                | Entonces                |
|-------------------|-------------------------|
| 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 |

En la Tabla 5.6 se muestra el modelo de reglas para el diagnóstico de las señales de la instrumentación *fieldbus*. Las condiciones de estas reglas son los casos generados por la Tabla 5.2. Este modelo de reglas genera acciones en forma de códigos de diagnóstico de anomalías de la instrumentación *fieldbus*.

El modelo de reglas de la Tabla 5.7 tiene como condiciones los casos generados por el modelo de reglas de la camada de monitoreo de condición (Tabla 5.4). Este modelo de reglas genera acciones en forma de códigos de diagnóstico de anomalías de las condiciones de operación.

Tabla 5.6- Modelo de reglas para diagnóstico de la instrumentación *fieldbus*.

| Si                  | Entonces   |
|---------------------|--|
| signal-BAD-0 ?label | gui12 COD_DIAG#EMAILS<br>gui14 SISTEMA-amarillo  |
| signal-BAD-1 ?label | gui12 COD_DIAG#EMAILS<br>gui14 SISTEMA- amarillo |
| signal-BAD-2 ?label | gui12 COD_DIAG#EMAILS<br>gui14 SISTEMA- amarillo |
| signal-BAD-3 ?label | gui12 COD_DIAG#EMAILS<br>gui14 SISTEMA- amarillo |
| signal-BAD-4 ?label | gui12 COD_DIAG#EMAILS<br>gui14 SISTEMA- amarillo |

Tabla 5.7- Modelo de reglas para diagnóstico de monitoreo de condición.

| Si                      | Entonces   |
|-------------------------|--|
| condition COD_MC-alto   | gui11 COD_DIAG#EMAILS<br>gui14 SISTEMA- amarillo |
| condition COD_MC-alarma | gui11 COD_DIAG#EMAILS<br>gui14 SISTEMA-rojo      |
| condition COD_MC-trip   | gui11 COD_DIAG#EMAILS<br>gui14 SISTEMA-rojo      |

Los consecuentes o acciones generados por los modelos de la Tabla 5.6 y la Tabla 5.7 son códigos de diagnóstico (COD\_DIAG), lista de e-mails (EMAILS) para el envío de OTM, para la visualización en la camada de presentación se envía el nombre del sistema (SISTEMA) y el color de alerta.

El comando *printout guiXY "texto"* envía un mensaje para el código Java conteniendo el "texto". En este caso, X es un número de 1 a 5, correspondiente al número de la UGH, e Y es el número 1, 2 o 4, que poseen los siguientes significados:

- 1: La falla es relativa a un equipo de la UGH.
- 2: La falla es relativa a la instrumentación.
- 4: Indica una información interna para hacer al cliente parpadear en amarillo o rojo, de acuerdo con el tipo de falla (ALERTA, ALARMA, TRIP, STATUS BAD, QUALITY BAD, OFFLINE).

#### **5.2.4 Estructura de reglas de la toma de decisión**

Para integrar informaciones necesarias que generen sugerencias de acciones de mantenimiento, el proceso de toma de decisión es hecho a través del modelo de reglas ilustrado en la Tabla 5.8. Las condiciones de estas reglas son las salidas de la camada de diagnóstico. Los consecuentes o acciones generados por el modelo son códigos de decisión (COD\_DCS), lista de e-mails (EMAILS) para el envío de las OTM, para la visualización en la camada de presentación se enviará el nombre del sistema (SISTEMA) y el color de alerta.

Tabla 5.8- Modelo de reglas para toma de decisión.

| <b>Si</b>               | <b>Entonces</b>                                |
|-------------------------|--|
| diagnostic G1A-canalN-0 | gui14 SISTEMA-amarillo<br>gui12 COD_DCS#EMAILS |
| diagnostic G1A-canalN-1 | gui14 SISTEMA-amarillo<br>gui12 COD_DCS#EMAILS |
| diagnostic G1A-canalN-2 | gui14 SISTEMA-amarillo<br>gui12 COD_DCS#EMAILS |
| diagnostic G1A-canalN-3 | gui14 SISTEMA-amarillo<br>gui12 COD_DCS#EMAILS |

### 5.3 CLASES IMPLEMENTADAS EN EL SIMPREBAL

Los dos componentes de software que son desarrollados son los aplicativos *I-Kernel* y *Confmonittool*. Estos aplicativos son disponibilizados en forma de archivos JAR (*Java Archive*) y deberán ser ejecutados independientemente por medio de una máquina virtual Java. El módulo *I-Kernel* es una aplicación Java *standalone*, responsable por la adquisición de datos por medio de banco de datos y de servidores OPC, su procesamiento inteligente objetiva detectar situaciones anormales, y eventualmente sugerencias de toma de decisión. El módulo *Confmonittool* es un *Applet* Java, cargado a partir de una página *Web* almacenada en el servidor *Web*, y es responsable por las tareas de configuración de parámetros, así como el monitoreo activo de las variables, exhibidas a través de una interface de usuario.

La implementación de las clases tanto del *I-Kernel* cuanto del *Confmonittool* que originaron como resultado el SIMPREBAL. La descripción de uso de este sistema es detallada en el Apéndice D.

#### 5.3.1 Clases del *I-Kernel*

Las principales clases de la aplicación *I-kernel* mostradas en la Figura 5.3 son descritas a seguir:

- **Tag:** esta clase contiene los datos de las variables de proceso y es usada en el procesamiento inteligente por las reglas de producción;
- **Configuration:** Obtiene los parámetros de configuración;
- **OPCProxy:** Colecta los ítems en tiempo real del servidor OPC;

- *DBProxy*: a través de esta clase o sistema coleta y almacena informaciones en el banco de datos;
- *Mailer*: envía las OTM con sugerencias de las acciones de mantenimiento a los usuarios vía e-mail;
- *JessProxy*: procesamiento de reglas;
- *Logger*: registra los eventos de ejecución del sistema.

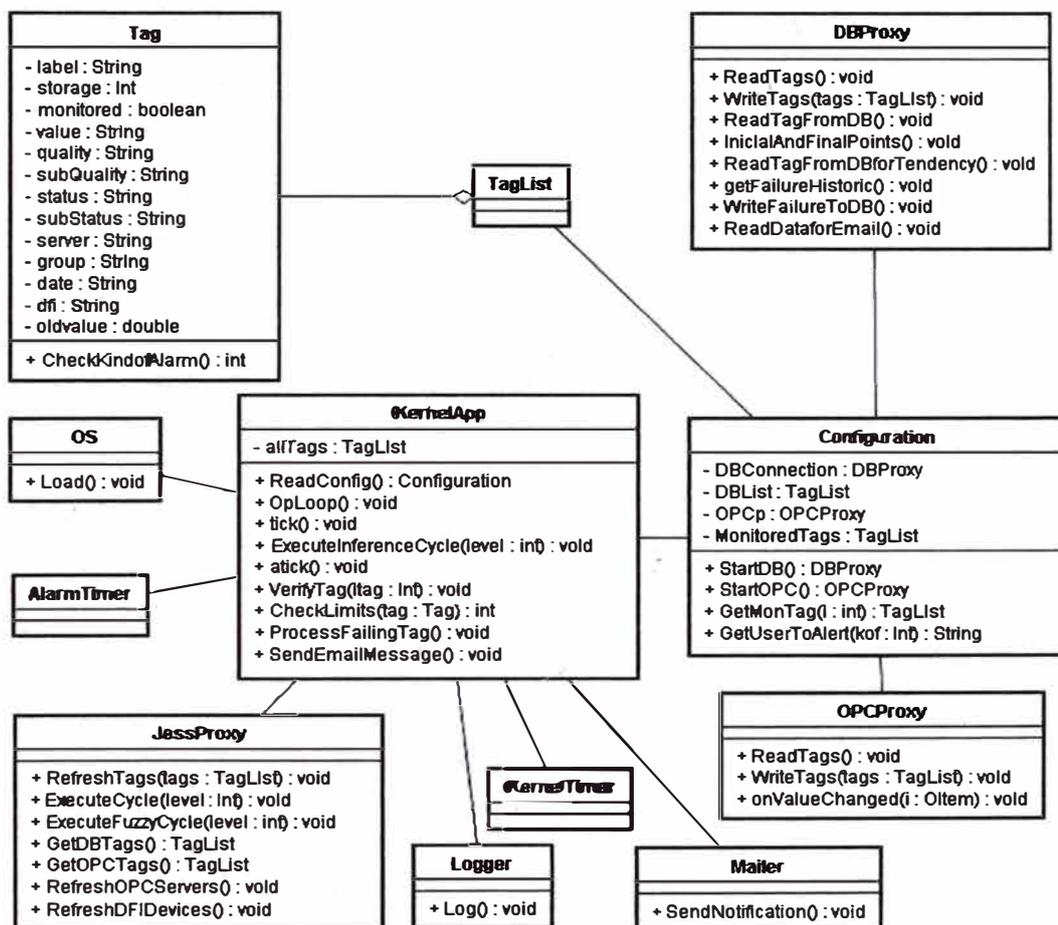


Figura 5.3- Clases del I-Kernel.

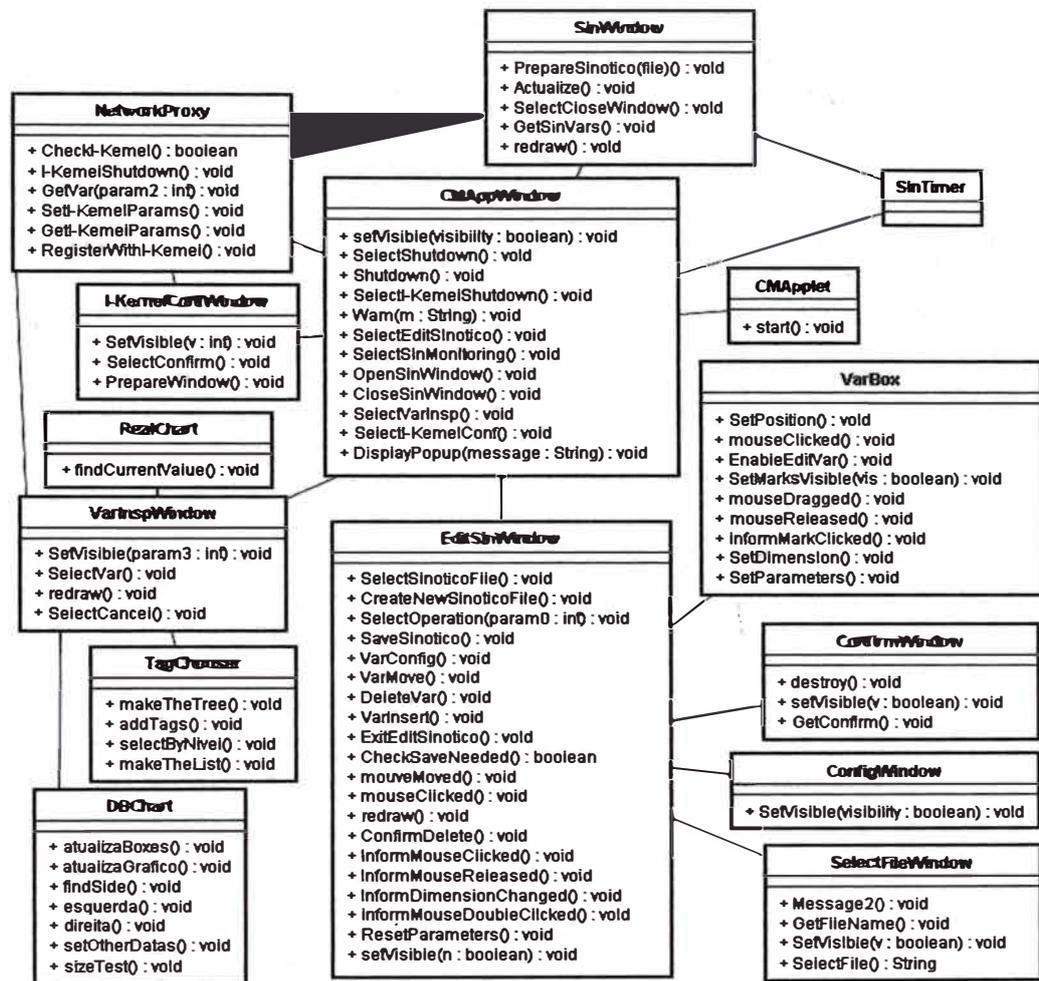


Figura 5.4- Clases del *Confmonitool*.

### 5.3.2 Clases del *Confmonitool*

Las principales clases de la herramienta *Confmonitool* mostradas en la Figura 5.4 son:

- *NetworkProxy*: a través de esta clase el cliente solicita y recibe informaciones del servidor;
- *VarInspWindo*: presenta una interface de inspección de variables, con columnas de valor, calidad y fecha que fue colectada;

- *TagChooser*: la selección de las tags del árbol jerárquico para inspección de variables es seleccionada a través de esta clase;
- *RealChart*: soporta los gráficos de variables en tiempo real;
- *DBChart*: gráficos de las variables históricas colectadas del banco de datos.

## **CAPÍTULO 6**

### **APLICACIÓN EN UNA CENTRAL HIDROELÉCTRICA**

En este capítulo se presenta la aplicación del software en una central hidroeléctrica para demostrar los métodos y algoritmos desarrollados en la metodología e implementados en el SIMPREBAL. Estudiaremos los sistemas turbina, mancal y generador eléctrico de la central hidroeléctrica de Balbina. Además presentaremos como ejemplo, las anomalías ocurridas en el sistema de enfriamiento del generador.

#### **6.1 CENTRAL HIDROELÉCTRICA Y SUS SISTEMAS**

La central hidroeléctrica de Balbina es en donde se desarrolla el prototipo del SE, ubicada en el municipio de Presidente Figueiredo en el estado de Amazonas a 180km de la ciudad de Manaus, en el río Uatuma, uno de los afluentes del río Amazonas. Dicha central está caracterizada por tener una capacidad de generación de 250MW de energía eléctrica (cinco generadores de 50MW). Los generadores son de tipo Umbrella de baja rotación (105,88rpm), capacidad nominal de 55,5MVA, tensión nominal de 13,8kV, y son numerados de 1 a 5, como se observa en la Figura 6.1.

Todos los equipos y sistemas monitoreados por el SIMPREBAL están indicados en la Figura 6.2. En esta sección se presenta una breve descripción de las

funciones de cada equipo y de sus instrumentos. Los equipos de la central de Balbina están divididos en los siguientes sistemas: Sistema mancal (SM), sistema turbina (ST) y sistema generador (SG).

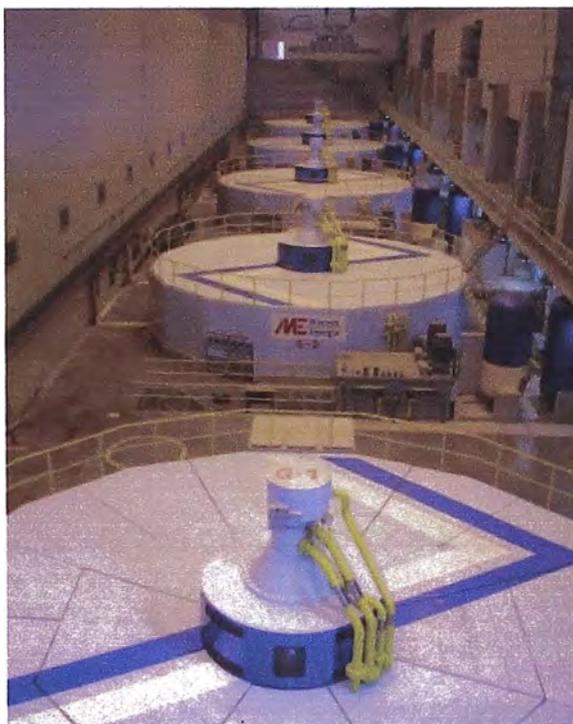


Figura 6.1- Vista superior de los 5 generadores.

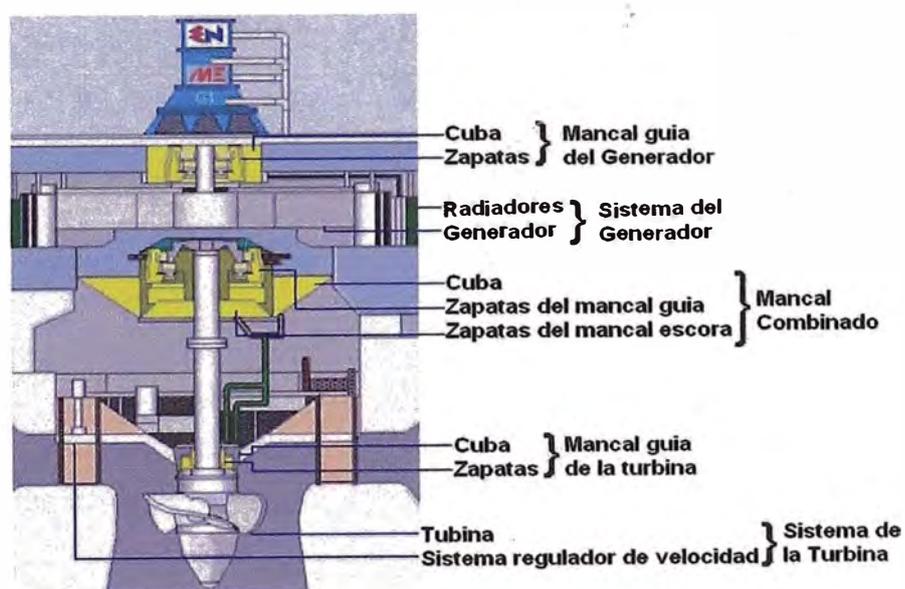


Figura 6.2- Ilustración del grupo turbina generador.

### 6.1.1 Sistema mancal

Los mancales son responsables por transferir los esfuerzos radiales y axiales del eje de la turbina al concreto, evitando así una vibración excesiva del mismo y el desgaste prematuro de todo el grupo. Los componentes de cada mancal son básicamente cuba y zapatas. La cuba es responsable por el almacenamiento de aceite que lubrica las partes activas del mancal y las zapatas son estructuras metálicas responsables por la formación de una película de aceite, por efecto hidrodinámico, entre el metal patente y el eje de la turbina durante el funcionamiento de la unidad generadora. La Figura 6.2 muestra el eje de la turbina, mostrando también la tapa, que brinda estanqueidad a los compartimentos del rotor Kaplan, y la región superior de las zapatas del mancal guía de la turbina (MGT).



Figura 6.3- Eje y mancal guía de la turbina.

La Tabla 6.1 presenta toda la instrumentación del sistema de mancal de la central hidroeléctrica de Balbina incluyendo las *tags* con que son referenciadas.

Tabla 6.1- Instrumentación del sistema de mancal.

| Subsistema                                    | Tag     | Descripción  |
|---|---------|--|
| Mancal guía del generador (MGG)               | 38GMM1  | Temperatura del metal de las zapatas (sonda n° 1)              |
|   | 38GMM2  | Temperatura del metal de las zapatas (sonda n° 2)              |
|   | 38GMM3  | Temperatura del metal de las zapatas (sonda n° 3)              |
|   | 38GMO1  | Temperatura del aceite en la cuba (sonda n° 1)                 |
|   | 38GMO2  | Temperatura del aceite en la cuba (sonda n° 2)                 |
| Mancal guía de la turbina (MGT)               | 38MK1   | Temperatura del metal de las zapatas (sonda n° 1)              |
|   | 38MK2   | Temperatura del metal de las zapatas (sonda n° 2)              |
|   | 38MK3   | Temperatura del metal de las zapatas (sonda n° 3)              |
|   | 38MJ1   | Temperatura del aceite en la cuba (sonda n° 1)                 |
|   | 38MJ2   | Temperatura del aceite en la cuba (sonda n° 2)                 |
| Mancal combinado (MCB)                        | 38MG1   | Temperatura del metal del mancal guía intermediario n°1        |
|   | 38MG2   | Temperatura del metal del mancal guía intermediario n°2        |
|   | 38MG3   | Temperatura del metal del mancal guía intermediario n°3        |
|   | 38ME1   | Temperatura del metal del mancal escora (sonda n°1)            |
|   | 38ME2   | Temperatura del metal del mancal escora (sonda n°2)            |
|   | 38ME3   | Temperatura del metal del mancal escora (sonda n°3)            |
|   | 38MI    | Temperatura del aceite en la cuba                              |
| Sistema de enfriamiento y lubricación del MCB | DT302-I | Densidad del aceite del mancal combinado                       |
|   | DT302-T | Temperatura del aceite calculada por el transmisor de densidad |

### 6.1.2 Sistema de la turbina

El sistema de la turbina se subdivide en: turbina hidráulica, sistema de drenaje de la tapa, sistema de aducción y descarga, y SRV (Sistema de Regulación de Velocidad). El SRV cual es mostrado en la Figura 6.4.

El SRV es el sistema responsable por controlar la velocidad de rotación de la turbina. Este control es realizado a través de comandos de

abertura y cerradura de las paletas del distribuidor. El distribuidor es un dispositivo mecánico compuesto por un conjunto de láminas verticales (paletas) formando un círculo.



Figura 6.4- Sistema de regulación de velocidad.

El sistema SIMPREBAL monitorea este proceso por medio de la adquisición de datos de sensores de temperatura del aceite de regulación y temperatura del agua de enfriamiento en los intercambiadores cuyas *tags* son mostradas en la Tabla 6.2.

Tabla 6.2- Instrumentación del sistema de la turbina.

| Subsistema                         | Tag  | Descripción                          |
|------------------------------------|------|--------------------------------------|
| Sistema de regulación de velocidad | 26AR | Temperatura del agua de enfriamiento |
|                                    | 26LK | Temperatura del aceite de regulación |

### 6.1.3 Sistema generador

La energía mecánica de rotación de la turbina es convertida en energía eléctrica en el generador. Cuando la turbina gira mueve al rotor y este a su vez mueve a una serie de grandes electroimanes que, una vez alimentados por una corriente de excitación, al girar producen un campo magnético variable. Este campo magnético induce corriente eléctrica en el estator. El estator es la parte fija de la máquina montada entorno del rotor. El rotor esta constituido de un material ferromagnético (núcleo) envuelto por un conjunto de enrolamientos. Para evitar el calentamiento excesivo del generador, el aire caliente que sale de los enrolamientos es enfriado por radiadores.

Tabla 6.3- Instrumentos del sistema generador.

| <b>Subsistema</b>                            | <b>Tag</b> | <b>Descripción</b>                              |
|--|------------|---|
| <b>Generador eléctrico principal</b>         | 49G1A      | Temperatura del enrolamiento del estator fase A |
|  | 49G1B      | Temperatura del enrolamiento del estator fase B |
|  | 49G1V      | Temperatura del enrolamiento del estator fase V |
|  | 49G2A      | Temperatura del núcleo del estator fase A       |
|  | 49G2B      | Temperatura del núcleo del estator fase B       |
|  | 49G2V      | Temperatura del núcleo del estator fase V       |
| <b>Sistema de enfriamiento del generador</b> | 26GAF1     | Temperatura del aire frio del radiador n° 1     |
|  | 26GAF2     | Temperatura del aire frio del radiador n° 2     |
|  | 26GAF3     | Temperatura del aire frio del radiador n° 3     |
|  | 26GAF4     | Temperatura del aire frio del radiador n° 4     |
|  | 26GAF5     | Temperatura del aire frio del radiador n° 5     |
|  | 26GAF6     | Temperatura del aire frio del radiador n° 6     |
|  | 26GAF7     | Temperatura del aire frio del radiador n° 7     |
|  | 26GAF8     | Temperatura del aire frio del radiador n° 8     |
|  | 26GAQ1     | Temperatura del aire caliente                   |

El sistema del generador es monitoreado por sensores de temperatura del núcleo y de enrolamiento del estator, además de sensores de temperatura del aire caliente y aire frío de los radiadores, conforme se especifica en la Tabla 6.3.

## 6.2 MÉTODO DE ANÁLISIS

La validación de los diagnósticos de falla y sugerencias de mantenimiento producidos por el SIMPREBAL fue realizada a través de la comparación entre las señales de falla en el sistema y las OTM ejecutadas por el personal de mantenimiento de la empresa. De este modo se puede verificar que a cada falla diagnosticada, si la misma es válida, su sugerencia de mantenimiento tiene que ser adoptada.

Tabla 6.4- Ejemplos de registros de OTM.

| Ítem | Descripción de Inicio del Servicio   | Nº      | Inicio   | Termino  |
|------|--|---------|----------|----------|
|      | Descripción de cierre del Servicio   | Tipo    | Hora     | Hora     |
| 14   | Corregir fuga de aceite en el sistema de circulación del MGG   | 3002036 | 26/02/08 | 27/02/08 |
|      | Fue corregido la fuga de aceite del sistema MGS con el ajuste de los tornillos de las válvulas y las juntas de las MBs 01 y 02 | O       | 09:58    | 09:33    |
| 15   | Efectuar lubricación y limpieza en la MB AG  | 3002047 | 27/02/08 | 27/02/08 |
|      | Inspección de limpieza y lubricación efectuada   | O       | 08:20    | 14:39    |

Los formularios de OTM son documentos que relatan todas las acciones de mantenimiento de la empresa. Los registros son numerados secuencialmente en una planilla y contiene las siguientes informaciones: Descripción de inicio del servicio,

descripción del cerramiento, número del servicio, tipo (ordinario, extraordinario, especial o con parada de máquina), horario de inicio y horario de término. La Tabla 6.4 presenta un ejemplo de dos órdenes de servicio.

### **6.3 RESULTADOS Y DISCUSIONES**

En esta sección se presenta los resultados obtenidos de la aplicación del SIMPREBAL a los diferentes sistemas de la central hidroeléctrica de Balbina. Estudios mas detallados están documentados en los trabajos de Amaya (2008), quien analizó la actuación del SIMPREBAL en el sistema generador, Tonaco (2008) analizó el sistema turbina, y Souza (2008) analizó el sistema mancal. Estos estudios serán comentados en el presente trabajo, comparando todos los diagnósticos del sistema con las órdenes de servicio ejecutadas por los funcionarios de la empresa.

#### **6.3.1 Análisis de la fallas**

Los diagnósticos de fallas registrados en el banco de datos del SIMPREBAL, durante los 75 días de operación del sistema, indican la ocurrencia de 22 estados TRIPS, 843 ALARMAS y 1541 ALERTAS, distribuidos entre los sistemas conforme se presenta en la Figura 6.5.

Se puede observar que en poco tiempo de operación del sistema, el número de anomalías detectadas fue bastante elevado. Esto se debe al establecer rangos rígidos de operación para el monitoreo de las condiciones funcionales de los equipos, no se llevo en consideración fluctuaciones de

valores de las variables monitoreadas y su influencia en el registro de las anomalías.

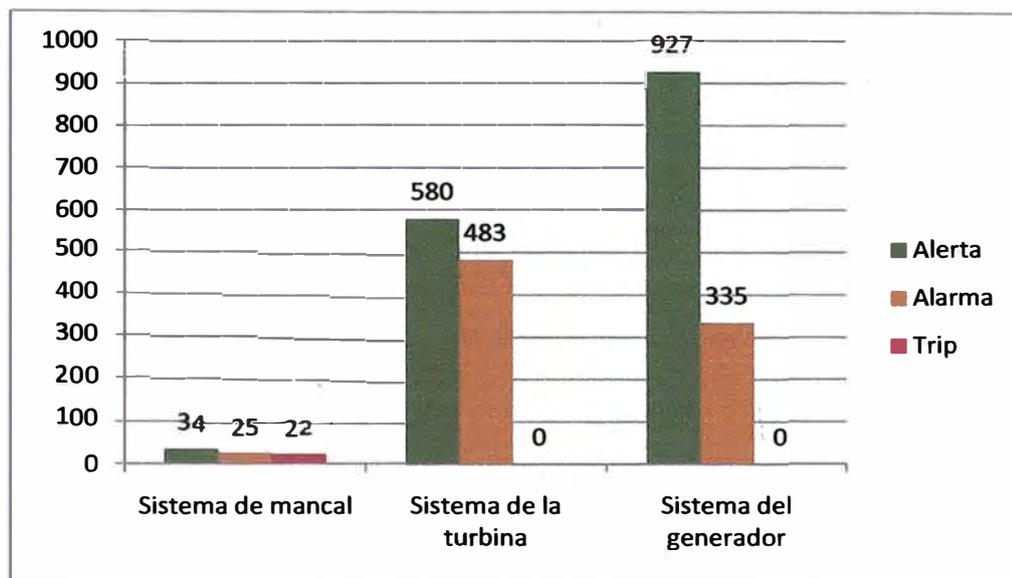


Figura 6.5- Anomalías detectas.

Un ejemplo es cuando el valor medido de una medición física como temperatura, oscila de tal forma que entra y sale varias veces de un determinado rango de operación, entonces esta medida es registrada en el banco de datos tantas veces entra en el referido rango de operación. La Figura 6.6 ilustra esta situación para la variable temperatura del aire frío del radiador n° 8 del sistema de enfriamiento del generador (SEG) de la UGH4. En el período del 28/03/2008 al 07/04/2008 fueron registrados 34 alertas asociados a esta variable.

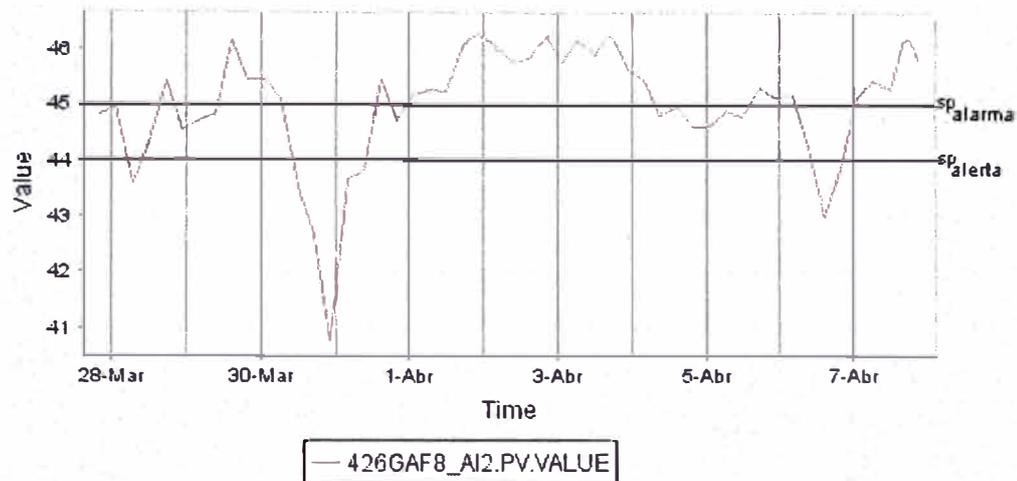


Figura 6.6- Oscilaciones de una variable monitoreada entre alerta y alarma.

El registro excesivo de una alerta podría reducirse si hubiese un tratamiento preliminar de los datos obtenidos vía OPC antes de procesar las reglas de producción. Tal tratamiento podría ser, redondear los valores a dos cifras decimales. Otra solución para este problema podría ser la introducción de lógica difusa en el procesamiento de señal. Los rangos de monitoreo de condición podrían ser *fuzzyficadas* para no indicar más condiciones estáticas para las variables, y sí condiciones de mayor o menor probabilidad de ocurrencia. De esta forma se evitaría los registros frecuentes de una misma anomalía en el banco de datos.

Una solución más simple que la utilización de lógica difusa, sería filtrar las variables procesadas a través de las reglas de producción por medio de histéresis. Por ejemplo, la variable temperatura podrá entrar en el rango de operación designado como alerta cuando es mayor que 45°C, mas sólo saldrá de este rango cuando esta por debajo de 44,5°C.

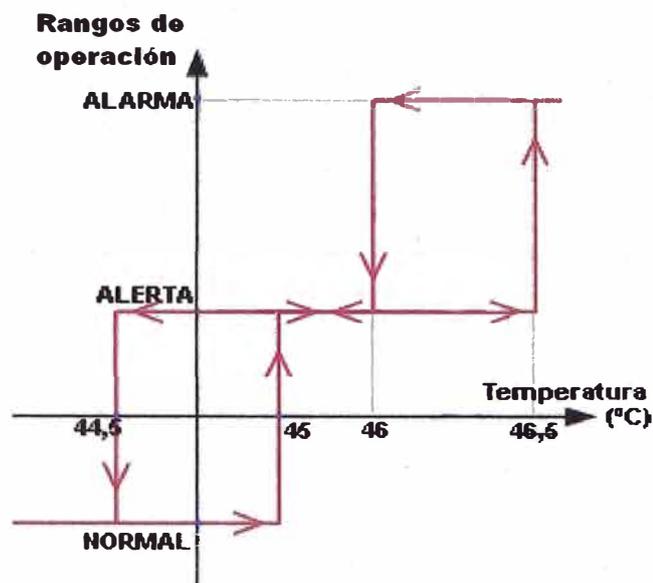


Figura 6.7- Curva de histéresis para manipulación de variables.

Por otro lado, esta misma variable entrará en el rango de alarma cuando sobrepasa  $46,5^{\circ}\text{C}$ , y sólo podrá salir de este rango si su valor está debajo de  $46^{\circ}\text{C}$ . La Figura 6.7 presenta una curva de histéresis para este proceso y la Figura 6.8 ejemplifica el proceso a través de un gráfico de evolución del valor de la variable monitoreada a lo largo del tiempo. Se pueden observar en la Figura 6.8 que la segunda alerta sólo es generada después que la señal monitoreada retorne al intervalo de valores normales.

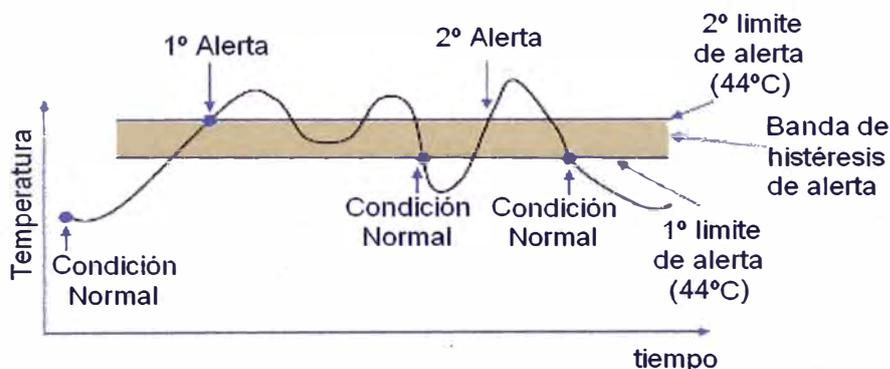


Figura 6.8- Zona de histéresis de manipulación de variables.

### 6.3.2 Ejemplo: Sistema de Enfriamiento del Generador

En esta sección vamos a analizar el caso del sistema de enfriamiento del generador de la UGH5. El proceso comienza con la adquisición en tiempo real de datos de la instrumentación desde el servidor OPC. En el momento que un valor sale de los límites de operación registrados en la base de reglas de producción, el gráfico correspondiente a la UGH comienza a parpadear en colores amarillo para alerta y alarma y rojo para trip, así como se muestra en la Figura 6.9.

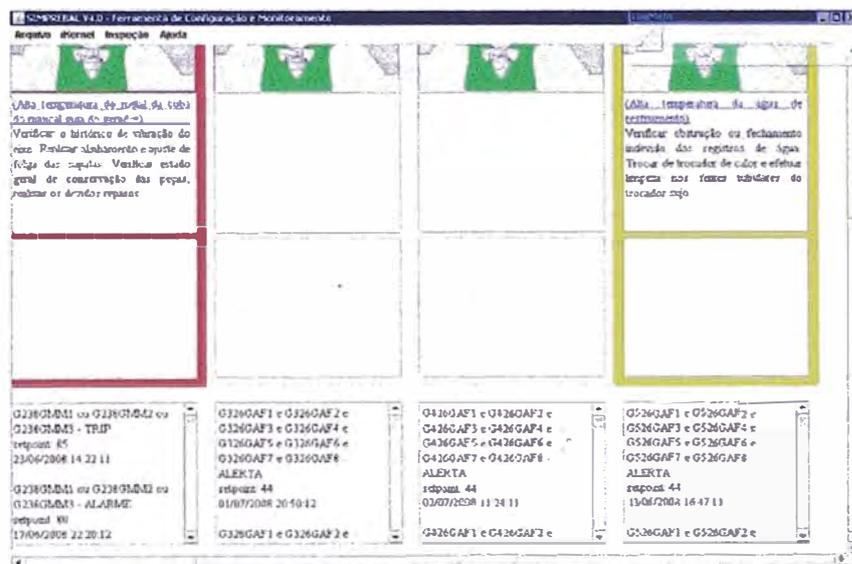


Figura 6.9- Interface mostrando las anomalías en tiempo real.

Una vez detectadas las anomalías por el sistema, son almacenadas en el banco de datos como se muestra en la Figura 6.10. El banco de datos contiene información como: código, descripción, causa, fecha de inicio, severidad y set point. Otra forma de informar sobre la ocurrencia de

anomalías al personal de operación y mantenimiento es a través de email, el formato de email es mostrado en la Figura 6.11.

|    | codigo    | descripcion                                | causa                  | data_inicio         | severidade | setpoint |
|----|-----------|--|------------------------|---------------------|------------|----------|
| 35 | G526GAFD1 | Alta temperatura de ar frio dos radiadores | Sujeira nos radiadores | 2008-06-13 09:29:11 | 4          | 44       |
| 36 | G526GAFD1 | 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 | G526GAFD1 | Alta temperatura de ar frio dos radiadores | Sujeira nos radiadores | 2008-06-13 10:42:11 | 4          | 44       |
| 39 | G526GAFD1 | Alta temperatura de ar frio dos radiadores | Sujeira nos radiadores | 2008-06-14 10:52:11 | 4          | 44       |
| 40 | G526GAFD1 | Alta temperatura de ar frio dos radiadores | Sujeira nos radiadores | 2008-06-14 11:01:11 | 4          | 44       |
| 41 | G526GAFD1 | Alta temperatura de ar frio dos radiadores | Sujeira nos radiadores | 2008-06-14 11:06:11 | 4          | 44       |

Figura 6.10- Anomalías almacenadas en el banco de datos.

|  |
|--|
| <p><b>Localización:</b> Sistema del Generador - Sistema de enfriamiento del generador.</p> <p><b>Modo de falla:</b> G526GAF1, G526GAF2, G526GAF3, G526GAF4, G526GAF5, G526GAF6, G526GAF7 y G526GAF8 - ALERTA.</p> <p><b>Severidad:</b> 4.</p> <p><b>Descripción:</b> Temperatura alta de aire frio en los radiadores.</p> <p><b>Causa:</b> Radiadores sucios.</p> <p><b>Decisión:</b> Alertar la posibilidad de efectuar limpieza interna de los radiadores.</p> <p>Manaus Energía<br/>SIMPREGAL</p> |
|--|

Figura 6.11- Email de anomalías enviado por el sistema.

Después de detectada la anomalía el operador puede entrar al sistema y analizar la tendencia de las variables asociadas a la anomalía. En la Figura 6.12 se muestra la selección de variables a partir del árbol jerárquico y la tabla inspección de variables. El grafico de tendencia de las variables es observada en la Figura 6.13.

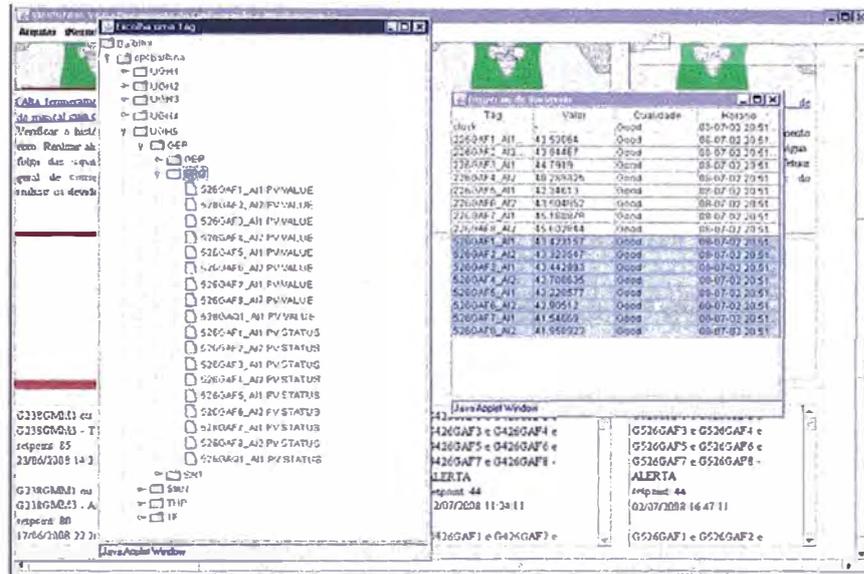


Figura 6.12- Selección e inspección de variables.

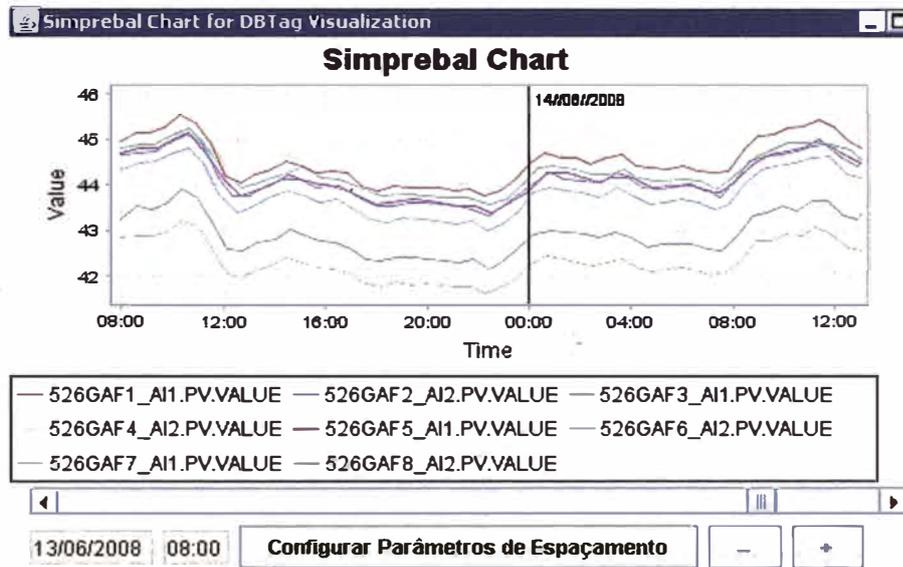


Figura 6.13- Tendencia de variables asociadas a anomalías del SEG.

### 6.3.3 Análisis Comparativo

Las informaciones disponibles antes y después de la instalación del sistema en el área de mantenimiento de la UGH son mostradas en la Tabla

6.5, esta tabla muestra la comparación de los resultados obtenidos por un operador de la planta y por el SE instalado.

Las ecuaciones para el cálculo de los indicadores son mostrados en el Apéndice D considerando un periodo de análisis de 75 días, estos indicadores son, el porcentaje de acierto descrito en la Ecuación D.4, los valores son extraídos de la Figura 6.14 la rapidez de diagnóstico es un tiempo estimado, siendo 1 segundo el ciclo de procesamiento del SE. Las variables analizadas son la cantidad de datos a los que tiene acceso y puede analizar en simultáneo. La tasa media de fallas se calcula según la Ecuación D.1 y el tiempo medio para reparo fue calculado siguiendo la Ecuación D.2.

Tabla 6.5- Cuadro comparativo entre un operador y el SE.

|                                    | <b>Operador</b>            | <b>Sistema Especialista</b> |
|------------------------------------|----------------------------|-----------------------------|
| Porcentaje de acierto              | 100%                       | 78.6%                       |
| Rapidez de diagnóstico             | 5 min                      | 1s                          |
| VARIABLES analizadas en simultáneo | Max. 10                    | 1500                        |
| Tasa media de fallas ( $\lambda$ ) | $3.5 \times 10^{-2}$ / día | $3.4 \times 10^{-2}$ / día  |
| El tiempo medio para reparo (MTTR) | 1.67 días                  | 1 día                       |

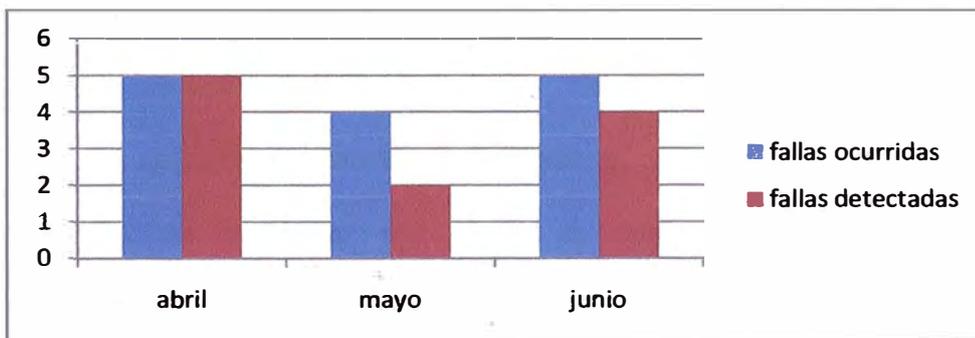


Figura 6.14- Fallas ocurridas y las detectadas por el sistema.

## CAPÍTULO 7

### APORTES PARA APLICACIONES ACADÉMICAS

En este capítulo se presentan los principales aportes de esta tesis para el desarrollo de aplicaciones relacionados al tema de investigación. El sistema dada su concepción en forma de un componente de software, podrá ser reutilizado en proyectos futuros que demanden soluciones de adquisición de datos vía servidor OPC, almacenamiento en banco de datos y procesamiento inteligente con SE a través de reglas de producción.

#### 7.1 ADQUISICIÓN DE DATOS

La adquisición de datos en tiempo real en aplicaciones Java usando servidores OPC con datos de instrumentación o simulados puede ser efectuada a través de *JOPClient* (<http://www.opi.dk/>) u *Openscada* (<http://openscada.org/>).

La librería *JOPClient* es una aplicación comercial, basada en Java que necesita configuraciones DCOM (*Distributed Component Object Model*). El *Openscada* es una aplicación desarrollada enteramente en Java, independiente de la configuración DCOM y puede ser usado en cualquier sistema operativo computacional.

## 7.2 BANCO DE DATOS

Aplicaciones académicas e industriales tienen la necesidad de almacenar los datos e informaciones generadas. Una herramienta ampliamente usada es MySQL (<http://www.mysql.com/>), el cual es un administrador de banco de datos gratuito y usa el lenguaje SQL estándar.

Una de las formas de interactuar con el banco de datos es vía PHP 4.4.9 (<http://www.php.net/>). A través de este método aplicaciones vía internet pueden comunicarse con banco de datos genéricos.

## 7.3 AMBIENTE DE DESARROLLO JAVA

Un ambiente de programación muy usado en el desarrollo de aplicaciones vía internet es Java. Más detalles de esta tecnología pueden ser encontrados en la página <http://www.sun.com/java/>.

Un ambiente para el desarrollo integrado (IDE - *Integrated Development Environment*) usado es Netbeans cuya última versión es 6.1 (<http://www.netbeans.org/>) el cual es una herramienta libre para desarrolladores. Además de permitir el desarrollo de aplicaciones en Java también permite la programación en C/C++. Puede ser ejecutado en diferentes plataformas como Windows, Linux, Mac OS X y Solaris.

#### **7.4 SISTEMAS ESPECIALISTAS**

La implementación de SE en ambiente Java puede ser efectuada usando la herramienta JESS, la cual es enteramente desarrollada en Java. La documentación y librerías a usar pueden ser descargadas a través de <http://herzberg.ca.sandia.gov/>. Esta herramienta es ampliamente usando en aplicaciones Java que involucren IA, haciendo al software desarrollado tener la capacidad de razonar. Para tener esta capacidad es necesario suministrar conocimiento en forma de reglas declarativas. En el lenguaje declarativo puede ser usada CLIPS (<http://clipsrules.sourceforge.net/>).

## **CONCLUSIONES Y RECOMENDACIONES**

Finalizamos esta tesis con las conclusiones asociadas a la metodología propuesta para el desarrollo de un sistema inteligente MBC y la implementación computacional del SIMPREBAL. También se presentan las recomendaciones para trabajos futuros que encaminan la implementación computacional de sistemas MBC.

### **CONCLUSIONES**

1. La metodología desarrollada en capas permite la implementación modular de una capa independiente de las otras, esto permite al sistema ser implementado por varios grupos de investigadores.
2. Los sistemas desarrollados a partir de esta metodología podrán ser aplicados en la industria para el auxilio a las decisiones del personal de operación y mantenimiento referente a las acciones de mantenimiento.
3. Los conceptos y herramientas usadas en esta tesis podrán ser reutilizadas en aplicaciones académicas e industriales que requieran adquisición y almacenamiento de datos, SE y aplicación vía internet.
4. La forma de representación de conocimiento utilizada en el SE favorece la expansión y la comunicación de información, aumentando el nivel de conocimiento de las personas involucradas en actividades de operación y mantenimiento.

5. Las principales ventajas obtenidas con el uso de SE en el sistema MBC son el aumento de confiabilidad, rapidez en el proceso de diagnóstico y mejoras en los indicadores de mantenimiento, mostrados en la Tabla 6.5.
6. La función principal del SIMPREBAL es auxiliar las tomas de decisión de mantenimiento, también es usado para análisis de tendencia y asociación de variables en tiempo real e históricas.
7. El uso de la tecnología *applet* Java y comunicación vía *socket* en la implementación permite al usuario utilizar el sistema independiente de la plataforma computacional.
8. El sistema puede procesar variables de sistemas convencionales 4-20mA, siendo necesario un software *bridge* para convertirlo a señales OPC.
9. El uso del internet permite el monitoreo a distancia de los estados de salud de los sistemas, subsistemas, equipos e instrumentos, permitiendo disponer informaciones de diagnósticos y sus tomas de decisiones.

## **RECOMENDACIONES**

1. La implementación de la camada de pronósticos lo cual permitirá predecir anomalías. Las informaciones generadas por esta camada ayudarán a tomar una decisión más acertada y anticipada ante una falla. Esta implementación puede ser realizada usando algoritmos de pronóstico basados en redes neuronales artificiales, lógica difusa, etc.
2. Implementar reglas de producción en áreas industrial o académica lo cual permitirá aplicar la metodología genérica propuesta en esta tesis.

3. Integrar las informaciones generadas por el sistema MBC y por otros sistemas de información de una planta de procesos a través del modelo OSA-EAI (*Open System Architecture for Enterprise Application Integration*). Esta integración permitirá a los operadores, personal de mantenimiento, administradores de logística, vendedores de repuestos e ingenieros, conocer las informaciones de las condiciones de los equipos. Centralizar en un computador diferentes tipos de información como: datos de monitoreo de condición, datos de fabricante, datos de instalación, datos de mantenimiento, datos operacionales, diagnóstico de salud de los activos y datos de confiabilidad.
4. Implementar un sistema de contraseñas de acceso para los usuarios del sistema considerando niveles de seguridad. Los niveles pueden variar desde administrador, programador, operador, etc. Con este sistema de seguridad el sistema de mantenimiento será menos vulnerable a modificaciones indeseadas y posibles ingresos de usuarios no autorizados.
5. Actualizar la base de conocimiento del Sistema Especialista corrigiendo reglas de producción existentes después de una evaluación y aumentando los casos que aun no se presentaron durante el funcionamiento de la planta. Este deberá un trabajo continuo que involucre a especialistas, ingenieros de conocimiento y programadores.

## BIBLIOGRAFÍA

- Abel, M. (1998), *Sistemas Especialistas*, Instituto de Informática, Universidade Federal do Rio Grande do Sul.
- 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.
- AMAYA, E.J., ÁLVARES, A., TONACO, R., SOUZA, R. y GUDWIN, R. (2007A), *An Intelligent Kernel for the Maintenance System of a Hydroelectric Power Plant*, 19th International Congress of Mechanical Engineering, COBEM2007, Universidade de Brasilia, Brasilia, Brasil.
- AMAYA, E.J., ÁLVARES, A., TONACO, R. y 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.
- 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, Departamento de Mecânica, UnB, Brasília, 179p.
- Anwar, M.R., Anwar, O., Shamim, S.F. y Zahid, A.A. (2004), *Human Machine Interface Using OPC (OLE for Process Control)*, Engineering, Sciences and Technology, Student Conference On, pp. 35- 40.
- Araújo, L.O.C., Grilo, L.M., Souza, U.E.L. y Melhado, S. (2001). *O Microplanejamento do Serviço de Concretagem: Análise e Aplicabilidade das Ferramentas da Qualidade*, In: II Sibrageq - 2º Simpósio Brasileiro de Gestão e Economia da Construção, Fortaleza/CE: ANTAC/UFC/Unifor/CEFET.
- Armitage, B., Dunlop, G., Hutchinson, D. y 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.
- 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.
- Bauchspiess, A. (2004), *Introdução aos Sistemas Inteligentes – Aplicações em Engenharia de Redes Neurais Artificiais, Lógica Fuzzy e Sistemas Neuro Fuzzy*, Apostila, Universidade de Brasília, UnB, 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. y 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.
- Bittencourt, G. (2001), *Inteligência artificial: ferramentas e teorias*, 2ed. 362 p, UFSC Florianópolis, Brasil.
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad P y 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. y Rumbaugh, J. (1999), *The Unified Modeling Language User Guide*, Addison-Wesley.

- Britto, T. (2006), *Metodologia da Manutenção Centrada em Confiabilidade Aplicada a Pára-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).
- 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/>.
- 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. y 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.
- Duarte, C., Figueiredo, L. y 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.
- 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., y Harrington, M. (2008), The Application of Open System Architecture for Condition Based Maintenance to Complete IVHM, Aerospace Conference IEEE , pp.1-9.

- Falqueto, J. y 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.
- 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.
- 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. y 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. y 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. y Riley, G. (1994), *Expert systems: principles and programming*, 2ed., PWS Publishing Company, Boston.
- Gonzalez, A. y Dankel, D. (1993), *The Engineering of knowledge-based systems*, Prentice Hall. New Jersey.
- 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. y 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. y 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.
- IEEE (1996), Transactions on Power Systems, New York, Vol. 11, No. 1, pp. 206-215, periódico.
- Jacobson, I., Booch, G. y Rumbaugh, J. (1999), *The Unified Software Development Process*, Addison Wesley.
- Javadpour, R. y Knapp, G. M. (2003), A fuzzy neural network approach to machine condition monitoring, Computers and Industrial Engineering, Vol. 45, No. 2, pp. 323–330.
- 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.
- Kothamasua, R. y Huang, S. H. (2007), Adaptive Mamdani fuzzy model for condition-based maintenance, Fuzzy Sets and Systems, Vol. 158, No. 24, pp. 2715–2733.
- Lacerda, J. y 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. y 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.
- Liebowitz, J. (1999), *The Handbook of Applied Expert Systems*, CRC Press, 868p.
- Mahalik, N.P. y Yen, M.(2008), *Extending fieldbus standards to food processing and packaging industry: A review*, Computer Standards & Interfaces.
- 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.

- Molina, J. M., Isasi, P., Berlanga, A. y 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. (1999), Maintenance Management: a new paradigm, In: <http://www.maintenanceresources.com/referencelibrary/rcm/maintparadigm.htm>, (06/05/2008).
- Moubray, J. (2000). RCM II: a manutenção centrada em confiabilidade, Gran Bretaña: Biddles Ltd., Guilford and King's Lynn. Edição Brasileira.
- NBR-5462 (1994), *Confiabilidade e Mantenabilidade, 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.
- Oliveira, C. H. P. (2002), SQL Curso prático, ed. Novatec, 272p, São Paulo, Brasil.
- 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.
- Rigoni, E., Peplow, L. A. y Silveira, P. R. (2004), *Sistema especialista de apoio à confiabilidade e a manutenção de sistemas técnicos automatizados*, In EMC 6610 Pós-Mec, Florianópolis.
- Schneider, G. y Winters, J. P. (1998), *Applying Use Cases: a practical guide*, Addison Wesley.
- 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.

- Sim, S. H. y Endrenyi, J. (1988), *Optimal preventive maintenance with repair*. IEEE Transactions on Reliability, Vol. 37, No. 1, pp. 92–96.
- 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. y 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.
- Souza, R.Q. (2008), *Metodologia e desenvolvimento de um sistema de manutenção preditiva visando à melhoria da confiabilidade de ativos de usinas hidrelétricas*, Dissertação de Mestrado em Sistemas Mecatrônicos, Departamento de Mecânica, UnB, Brasília, 131p.
- Tavares, L. y Filho, A. (2002), *A Manutenção Como Uma Atividade Corporativa*, Disponível na Internet < <http://www.abraman.org.br/publicações> >.
- 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.
- Tonaco, R.P. (2008), *Metodologia para Desenvolvimento de Base de Conhecimento Aplicada à Manutenção Baseada em Condição de Usinas Hidrelétricas*, Dissertação de Mestrado em Sistemas Mecatrônicos, Departamento de Mecânica, UnB, Brasília, 194p.
- 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. y 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.
- 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.

- Walter, R. (2006), *Open Systems Architecture for Condition-based Maintenance (OSA-CBM) Primer*, August.
- Waterman, D. A. (1986), *A guide to expert systems*, Addison-Wesley, p. 419, ISBN 0201083132.
- Yam, R., Tse, P., Li, L. y Tu, P. (2001), *Intelligent predictive decision support system for condition-based maintenance*, *Journal of Advanced Manufacturing Technology*, Vol. 17, No. 5, pp. 383–391.

## **APÉNDICES**

## **APÉNDICE A**

### **ARCHIVOS DE CONFIGURACIÓN**

En este apéndice se describen los tres archivos de configuración utilizados en el sistema. El archivo de configuración contiene los parámetros para la inicialización del servidor *I-kernel*. Los archivos códigos de falla y decisión contienen los códigos de todas las posibles fallas y su respectiva toma de decisión.

#### **A.1 ARCHIVO DE CONFIGURACIÓN**

El archivo de configuración posee los parámetros para la inicialización de la aplicación *I-kernel*. Cuando el *I-kernel* es iniciado, el sistema lee los parámetros contenidos en el archivo. Este archivo contiene los parámetros generales, de los servidores OPC, Banco de datos, e-mails, dispositivos DFI, tags OPC, tags de banco de datos y tags simuladas. A seguir se presentan los campos del archivo de configuración.

##### **A.1.1 Configuración general**

A seguir se describen todos los parámetros generales para inicialización del *I-kernel*, los posibles valores de estos parámetros son mostrados en la Figura A.1.

- **VERSION**, versión del SIMPREBAL;

- SOCKETPORT, puerta *socket* para comunicación del *SimprebalServer* con el *SimprebalClient*;
- OPCLIBRARY, biblioteca usada para obtener los datos vía OPC. Si el valor es uno, el sistema usa *JOPCClient*, y si es cero, usa *Openscada*;
- LOGLEVEL, forma de mostrar los eventos del *SimprebalServer*. Si el valor es uno, el sistema almacena los eventos en un archivo de texto, y si es cero, muestra en la interface del *prompt*;
- IKERNELTIMER, período de ejecución del procesamiento inteligente del *SimprebalServer*, expresado en milisegundos;
- PERCENTDEADBAND, porcentaje que el valor de una variable tiene que cambiar para ser almacenado en el banco de datos;
- SENDMAIL, permiso para envío de e-mails, si es uno, el *simprebalserver* enviara e-mails, y si es cero, no enviará emails;
- LOADINGTIMER, tiempo expresado en milisegundos para cargar los valores de todas las variables.

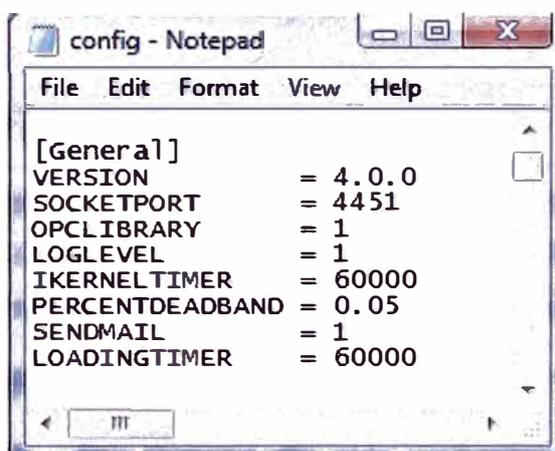


Figura A.1- Parámetros de configuración general.

```

config - Notepad
File Edit Format View Help

[OPCServers]
opcbaibina

[opcbaibina]
progid = Smar.Dfiolserver.0
host = 127.0.0.1
domain = manaus.com.br
user = usuario
password = senha
tags = opcbaibinatags

[opcbaibinatags]
; ***** U G H - 0 1 *****
; ----- GERADOR ELETRICO PRINCIPAL 01 -----
; ----- Gerador Elétrico Principal 01 -----
; -----
gl.gep.t.enrol.a = DF11A*UGH1.GEP.GEP*149G1A_AI1.PV.VALUE
gl.gep.t.enrol.b = DF11A*UGH1.GEP.GEP*149G1B_AI1.PV.VALUE
gl.gep.t.enrol.v = DF11A*UGH1.GEP.GEP*149G1V_AI1.PV.VALUE
gl.gep.t.nucleo.a = DF11A*UGH1.GEP.GEP*149G2A_AI2.PV.VALUE
gl.gep.t.nucleo.b = DF11A*UGH1.GEP.GEP*149G2B_AI2.PV.VALUE
gl.gep.t.nucleo.v = DF11A*UGH1.GEP.GEP*149G2V_AI2.PV.VALUE
gl.gep.st.t.enrol.a = DF11A*UGH1.GEP.GEP*149G1A_AI1.PV.STATUS
gl.gep.st.t.enrol.b = DF11A*UGH1.GEP.GEP*149G1B_AI1.PV.STATUS
gl.gep.st.t.enrol.v = DF11A*UGH1.GEP.GEP*149G1V_AI1.PV.STATUS
gl.gep.st.t.nucleo.a = DF11A*UGH1.GEP.GEP*149G2A_AI2.PV.STATUS
gl.gep.st.t.nucleo.b = DF11A*UGH1.GEP.GEP*149G2B_AI2.PV.STATUS
gl.gep.st.t.nucleo.v = DF11A*UGH1.GEP.GEP*149G2V_AI2.PV.STATUS
; ----- Sistema de Resfriamento do Gerador 01 -----
; -----
gl.srg.t.arfrio.rad1 = DF11A*UGH1.GEP.SRG*126GAF1_AI1.PV.VALUE
gl.srg.t.arfrio.rad2 = DF11A*UGH1.GEP.SRG*126GAF2_AI2.PV.VALUE
gl.srg.t.arfrio.rad3 = DF11A*UGH1.GEP.SRG*126GAF3_AI1.PV.VALUE
gl.srg.t.arfrio.rad4 = DF11A*UGH1.GEP.SRG*126GAF4_AI2.PV.VALUE
gl.srg.t.arfrio.rad5 = DF11A*UGH1.GEP.SRG*126GAF5_AI1.PV.VALUE
gl.srg.t.arfrio.rad6 = DF11A*UGH1.GEP.SRG*126GAF6_AI2.PV.VALUE
gl.srg.t.arfrio.rad7 = DF11A*UGH1.GEP.SRG*126GAF7_AI1.PV.VALUE
gl.srg.t.arfrio.rad8 = DF11A*UGH1.GEP.SRG*126GAF8_AI2.PV.VALUE

```

Figura A.2- Parámetros del Servidor OPC y tags asociadas.

### A.1.2 Servidores OPC y tags OPC

En la Figura A.2 se muestran los parámetros del servidor OPC. En el campo *OPCServers* están los nombres de los servidores OPC disponibles, puede ser más de un servidor. Cada servidor OPC tiene los siguientes parámetros descritos a seguir:

- *progid*, nombre del servidor OPC, depende del fabricante;
- *host*, número IP de la computadora donde está el servidor OPC;
- *domain*, dominio de la computadora donde está el servidor OPC;
- *user*, nombre del usuario que inicia la computadora del servidor OPC;

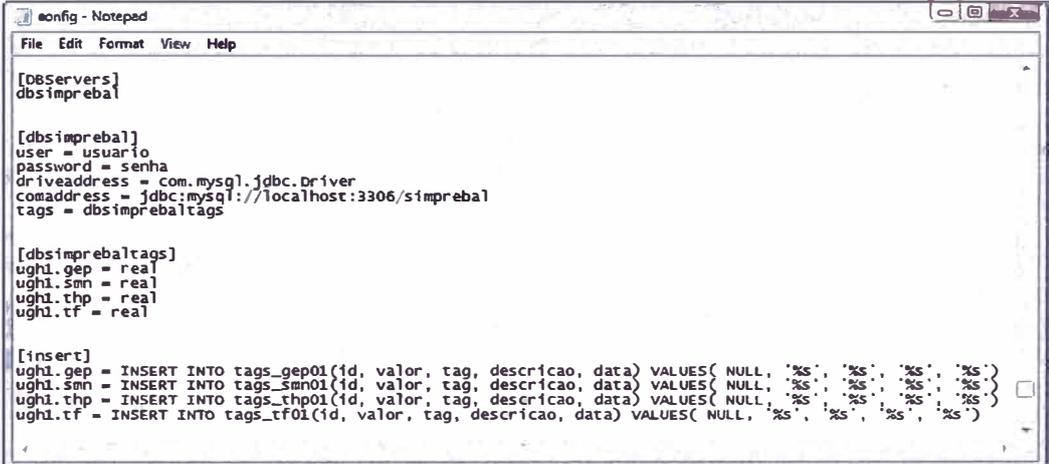
- *password*, clave del usuario;
- *tags*, lista de tags en el servidor OPC.

### A.1.3 Servidores de banco de datos

Los parámetros del servidor de banco de datos son mostrados en la Figura A.3. En el campo *DBServers* están los nombres de los servidores de banco de datos disponibles, puede ser más de uno. Cada servidor de banco de datos tiene los siguientes parámetros:

- *user*, usuario del banco de datos;
- *password*, clave del usuario del banco de datos;
- *driveaddress*, *driver* para conectar al banco de datos;
- *comaddress*, dirección electrónica del banco de datos;
- *tags*, lista de tags en el servidor de banco de datos.

Las tags del banco de datos pueden ser reales o simuladas. Las tags reales están relacionadas a los comandos *insert*, *select* o *update*.



```

[DBServers]
dbsimprebal

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

[dbsimprebaltags]
ughl.gcp = real
ughl.smn = real
ughl.thp = real
ughl.tf = real

[insert]
ughl.gcp = INSERT INTO tags_gcp01(id, valor, tag, descricao, data) VALUES( NULL, '%s', '%s', '%s', '%s')
ughl.smn = INSERT INTO tags_smn01(id, valor, tag, descricao, data) VALUES( NULL, '%s', '%s', '%s', '%s')
ughl.thp = INSERT INTO tags_thp01(id, valor, tag, descricao, data) VALUES( NULL, '%s', '%s', '%s', '%s')
ughl.tf = INSERT INTO tags_tf01(id, valor, tag, descricao, data) VALUES( NULL, '%s', '%s', '%s', '%s')

```

Figura A.3- Parámetros del servidor banco de datos y tags asociadas.

#### A.1.4 Tags simuladas

Las tags simuladas son mostradas en la Figura A.4. Estas tags son usadas en modo *offline* para probar el sistema en modo simulación.

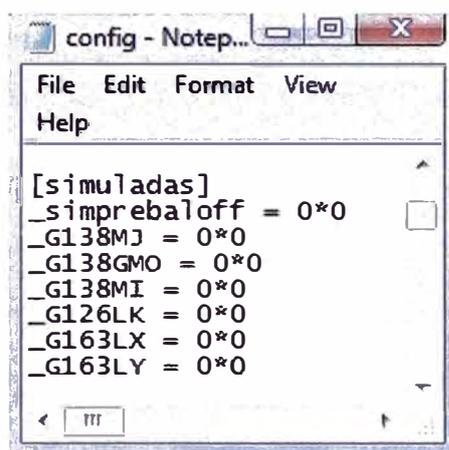


Figura A.4- Tags simuladas.

#### A.1.5 Dispositivos DFI

Todos los instrumentos están conectados a un dispositivo DFI. Estos parámetros incluyen el nombre del dispositivo DFI y el número de IP, así como se muestra en la Figura A.5.

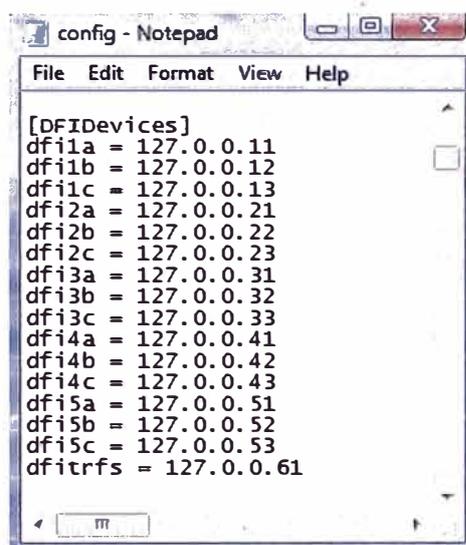


Figura A.5- Parámetros de los dispositivos DFI.

### A.1.6 Servidor de email

Los parámetros del servidor de e-mails, grupos de usuario y los emails de los miembros de los grupos son mostrados en la Figura A.6 y son descritos a seguir:

- *hostname*, número IP de la computadora donde está el servidor de email;
- *user*, usuario de la cuenta de email;
- *password*, clave del usuario de email;
- *servername*, nombre del servidor de e-mail;
- *email*, e-mail del usuario.



```

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 A.6- Parámetros del servidor de e-mail.

## A.2 ARCHIVO CÓDIGOS DE FALLA

El archivo códigos de falla mostrado en la Figura A.7 es usado para disponer de datos de falla de una manera codificada para el *I-kernel*. Este archivo es

serializado y usado como un objeto por la aplicación Java. Los datos de este archivo tiene la siguiente estructura: Código#descripción#id del equipo # modo #causa #detección #severidad #setpoint.

Nótese que el código anterior contiene códigos referidos a la falla y separados por el carácter #. El primero se refiere al código, el segundo es la descripción, el tercero es el ID del equipo conforme registrado en el banco de datos. El cuarto es el modo (tag - condición), el quinto es la causa, el sexto es el factor de detección, el séptimo es la severidad (estos dos últimos campos son valores referentes al FMEA) y finalmente el valor de *setpoint*.

```

FailureFile - Notepad
File Edit Format View Help
G1SIMPREBALOFF#Falha no servidor SIMPREBAL#415#SIMPREBAL Server - OFFLINE#0 servidor OPC esta inacessivel#1#3#
G2SIMPREBALOFF#Falha no servidor SIMPREBAL#415#SIMPREBAL Server - OFFLINE#0 servidor OPC esta inacessivel#1#3#
G3SIMPREBALOFF#Falha no servidor SIMPREBAL#415#SIMPREBAL Server - OFFLINE#0 servidor OPC esta inacessivel#1#3#
G4SIMPREBALOFF#Falha no servidor SIMPREBAL#415#SIMPREBAL Server - OFFLINE#0 servidor OPC esta inacessivel#1#3#
G5SIMPREBALOFF#Falha no servidor SIMPREBAL#415#SIMPREBAL Server - OFFLINE#0 servidor OPC esta inacessivel#1#3#
G1OPCSRV#Falha ao conectar com o servidor OPC#1#OPC Server - OFFLINE#0 cabo da rede esta desconectado#1#2#
G2OPCSRV#Falha ao conectar com o servidor OPC#1#OPC Server - OFFLINE#0 cabo da rede esta desconectado#1#2#
G3OPCSRV#Falha ao conectar com o servidor OPC#1#OPC Server - OFFLINE#0 cabo da rede esta desconectado#1#2#
G4OPCSRV#Falha ao conectar com o servidor OPC#1#OPC Server - OFFLINE#0 cabo da rede esta desconectado#1#2#
G5OPCSRV#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#

```

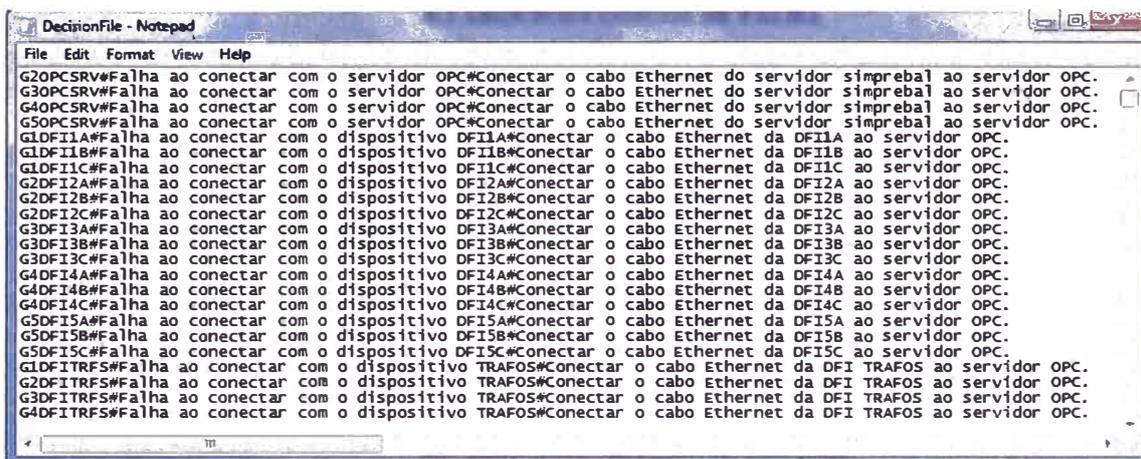
Figura A.7- Archivo códigos de falla.

### A.3 ARCHIVO CÓDIGOS DE DECISIÓN

El archivo códigos de decisión mostrado en la Figura A.8 es usado para disponer de los datos de toma de decisión de una manera fácil para la herramienta C&M. Este archivo es serializado y usado como un objeto por la aplicación Java. Para aumentar una nueva línea en este archivo es necesario que el usuario cree un código de falla, en el archivo de falla, y utiliza este código en el archivo de decisión,

referente a la nueva falla insertada. Los mensajes gravados en el archivo de decisión siguen la siguiente forma: Código#falla#toma de decisión.

Nótese que el código anterior contiene códigos separados por el carácter #. El primer se refiere al código de decisión, el segundo es la descripción de la falla, el tercero es la descripción de la decisión.



```

DecisionFile - Notepad
File Edit Format View Help
G2OPCSRv#Falha ao conectar com o servidor OPC#Conectar o cabo Ethernet do servidor simprebal ao servidor OPC.
G3OPCSRv#Falha ao conectar com o servidor OPC#Conectar o cabo Ethernet do servidor simprebal ao servidor OPC.
G4OPCSRv#Falha ao conectar com o servidor OPC#Conectar o cabo Ethernet do servidor simprebal ao servidor OPC.
G5OPCSRv#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 TRAFOS#Conectar o cabo Ethernet da DFI TRAFOS ao servidor OPC.
G2DFITRF5#Falha ao conectar com o dispositivo TRAFOS#Conectar o cabo Ethernet da DFI TRAFOS ao servidor OPC.
G3DFITRF5#Falha ao conectar com o dispositivo TRAFOS#Conectar o cabo Ethernet da DFI TRAFOS ao servidor OPC.
G4DFITRF5#Falha ao conectar com o dispositivo TRAFOS#Conectar o cabo Ethernet da DFI TRAFOS ao servidor OPC.
  
```

Figura A.8- Archivo códigos de decisión.

## APÉNDICE B

### CÁLCULO DE LA CLASE TAG

En este apéndice se muestran los procedimientos para el cálculo de los campos de la clase Tag. Esta clase es usada por las reglas de producción. Este proceso de cálculo es basado en la instrumentación *fieldbus* de Smar.

#### B.1 PROCESAMIENTO DEL ÍTEM *VALUE*

El proceso de cálculo de los campos de la clase Tag es mostrado en la Figura B.1. Un transmisor *fieldbus* (*i.e.* 149G1A) disponibiliza valores de sus parámetros a través de ítems, para el procesamiento seleccionamos dos ítems: *VALUE* (*i.e.* 149G1A\_AI1.PV.VALUE) y *STATUS* (*i.e.* 149G1A\_AI1.PV.STATUS). En el tipo *VALUE* están contenidas informaciones del valor de las medidas físicas, y la calidad de señal OPC. El tipo *STATUS* contiene información de calidad de señal de la instrumentación *fieldbus*.

En el ítem *VALUE* (*i.e.* 149G1A\_AI1.PV.VALUE) es procesada la propiedad *Quality*. Esta propiedad es de tipo *string* (*i.e.* *Good, non-specific*). Para mayor facilidad en el procesamiento de las reglas se convierte la propiedad *Quality* de *string* para números enteros *quality* y *subquality*. Los valores de *quality* y *subquality* son

obtenidos a partir de la Tabla B.1. Después de este procesamiento tendremos los valores de los campos *value*, *quality* y *subquality* de la clase Tag.

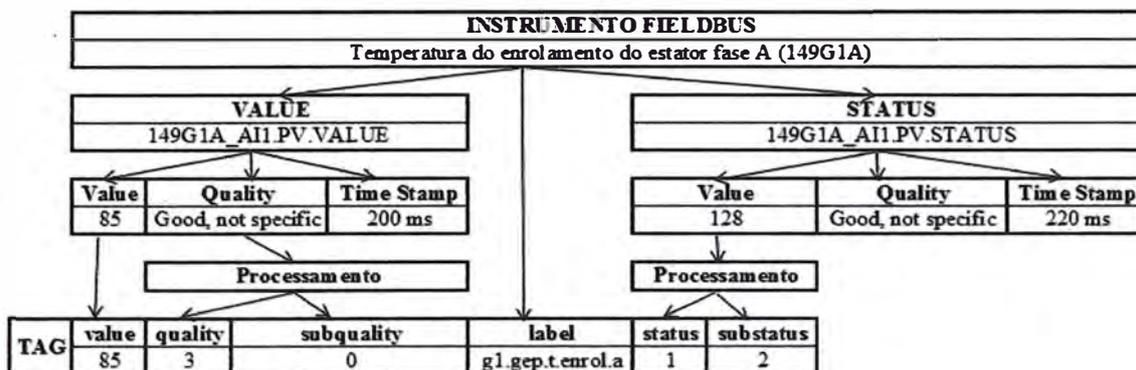


Figura B.1- Proceso de cálculo de la clase Tag.

Tabla B.1- Relación entre el *Quality* y valores *quality* y *subquality*.

| Quality                               | quality | subquality |
|---------------------------------------|---------|------------|
| 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          |

## B.2 PROCESAMIENTO DEL ÍTEM STATUS

Una característica muy útil disponibilizada por la instrumentación *fieldbus* es el ítem *STATUS* (i.e. 149G1A\_AI1.PV.STATUS). La gran ventaja de este ítem es que permite calificar el valor de la señal *fieldbus*, en caso de falla lleva al instrumento para una condición de seguridad (*Fail-Safe*) (Smar, 2005). La propiedad *value* del ítem *STATUS* es presentada en la Figura B.2, se puede observar que es compuesta de tres elementos: *Quality*, *SubStatus* y *Limits*. Estos tres elementos son relacionados matemáticamente en la Ecuación B.1.

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

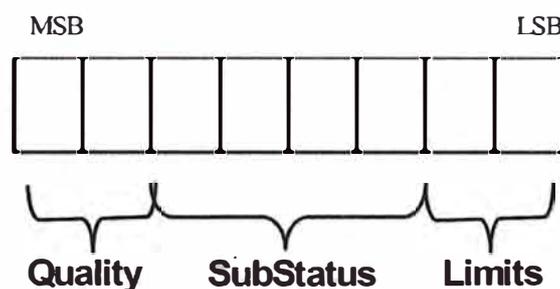


Figura B.2- Estructura de la propiedad *value* del ítem *STATUS* (Smar, 2005).

A seguir se presenta un ejemplo para el cálculo de los valores numéricos de *Quality*, *SubStatus* y *Limit*.

Valor del ítem *STATUS* = 78

Dividiendo el número por 64. El cociente será el *Quality* y almacenando el resto:

$Quality = 78 / 64 = 1$  (*Uncertain*, ver Tabla B.2)

Resto = 14

Dividiendo el resto por 4. El cociente será el *SubStatus* y el resto será el límite:

$SubStatus = 14 / 4 = 3$  (Valor inicial, ver Tabla B.2)

Limit = 2

Después de efectuar estas operaciones se tienen todos los valores de los campos de la clase Tag: *label*, *value*, *quality*, *subquality*, *status* y *substatus*.

Tabla B.2- Valores de *SubStatus* y *Quality* (Amaya *et al.*, 2007c).

|                  |   | Quality                                   |                                       |                                |                                    |
|------------------|---|---|---------------------------------------|--------------------------------|------------------------------------|
|                  |   | 0 = Bad                                   | 1 = Uncertain                         | 2 = Good Non Cascade           | 3 = Good Cascade                   |
| <b>SubStatus</b> | 0 | No específico                             | No específico                         | No específica (baja prioridad) | No específica                      |
|                  | 1 | Error de configuración                    | Último valor usable                   | Alarma activa de bloque        | Inicialización aprobada            |
|                  | 2 | No conectado                              | Substituto                            | Alarma activa de consulta      | Requisición de inicialización (IR) |
|                  | 3 | Falla en el Instrumento                   | Valor inicial                         | Alarma activa crítica          | No solicitado (NI)                 |
|                  | 4 | Falla en el Sensor                        | Conversión del sensor no exacto       | Alarma de bloque desconocida   | No seleccionado (NS)               |
|                  | 5 | Sin Comunicación, con último valor usable | Violación de límite de unidad técnica | Alarma de consulta desconocida | Cancelamiento local (LO)           |
|                  | 6 | Sin Comunicación, con valor no usable     | Sub-normal                            | Alarma crítica desconocida     | Estado de falla activo (FSA)       |
|                  | 7 | Fuera de servicio (alta prioridad)        |                                       |                                | Estado de falla iniciado (IFS)     |

## APÉNDICE C

### MÉTODOS DE ANÁLISIS FMEA Y FTA

En este apéndice se describen los conceptos de Análisis de Modos de Fallas y sus Efectos (FMEA) y Análisis de Árbol de Fallas (FTA).

#### C.1 ANÁLISIS DE MODOS DE FALLAS Y SUS EFECTOS - FMEA

Tabla C.1- Formulario estandarizado de análisis FMEA.

| Identificación del Equipo     |                               |                  |                    | Factores para Evaluación del Componente |                          |                |           |           |            |                          | Tarea<br>Propuesta<br>para<br>Mantenimiento |
|-------------------------------|-------------------------------|------------------|--------------------|---|--------------------------|----------------|-----------|-----------|------------|--------------------------|---|
| Identificación del Subsistema |                               |                  |                    | Seguridad y Medio Ambiente              | Perdida de Facturamiento | Corte de Carga | Detección | Severidad | Ocurrencia | Evaluación General (NPR) |   |
| Función                       |                               |                  |                    |   |                          |                |           |           |            |                          |   |
| Componente                    | Componente<br>Falla Funcional | Modo de la Falla | Efecto de la Falla |   |                          |                |           |           |            |                          |   |
|                               |                               |                  |                    |   |                          |                |           |           |            |                          |   |

El levantamiento de las funciones de los modos de fallas y sus efectos FMEA, se realiza a partir de la descripción textual del sistema desarrollado, de los registros de anomalías (OTM - Ordenes de Trabajo de Mantenimiento), de los planos de mantenimiento actuales y de los descriptivos funcionales y de instrumentación de los equipos y componentes. La documentación del análisis FMEA fue desarrollada

según el formulario estandarizado mostrado en la Tabla C.1. A continuación se explican cada uno los conceptos de las columnas del formulario presentado.

- **Función:** Descripción de la función del subsistema o equipo.
- **Componente:** Identificación del componente.
- **Función del componente:** Descripción sucinta y exacta de la tarea que el componente debe desempeñar.
- **Falla funcional:** Descripción de todas las posibles fallas pertinentes a cada componente.
- **Modo de falla:** Descripción simple y concisa de las ocurrencias (causas) que pueden dar origen al tipo de falla considerado.
- **Efecto de la falla:** Consecuencia de la ocurrencia de una falla, percibida o no por el usuario final. Puede ser local (no afecta los otros componentes) o global (puede afectar otras funciones o componentes).
- **Factores para evaluación del componente:** Consiste en una serie de criterios utilizados para evaluar la criticidad o prioridad de riesgo de un componente. En esta evaluación se considera la influencia de tres parámetros: severidad, ocurrencia y detección de fallas. Para estandarizar y tornar menos subjetiva la evaluación de la severidad de cada falla funcional, fueron categorizadas tres clases de efectos de fallas: los que afectan la seguridad y el medio ambiente, los que provocan pérdida de facturamiento, y los que provocan corte de carga. A seguir hay una descripción más detallada de las clases de efectos de las fallas.
- **Seguridad o medio ambiente:** Caracteriza la severidad de una falla de acuerdo con la intensidad con que ella puede afectar la seguridad de los funcionarios o

el medio ambiente, conforme establecido por las normas ISO de seguridad y protección al medio ambiente, ISO 18001 (seguridad) ISO 14001 (medio ambiente). La Tabla C.2 muestra el significado de los valores atribuidos a los índices de clasificación de las fallas cuanto a la intensidad con que afectan la seguridad o el medio ambiente.

- **Perdida de facturamiento:** Indica el grado con que una determinada falla afecta la economía de una planta. La Tabla C.3 presenta el significado de los valores atribuidos a los índices de pérdida de facturamiento.
- **Corte de carga:** Indica la probabilidad de una falla para provocar la parada de la planta. Los criterios de clasificación de los índices de corte de carga son mostrados en la Tabla C.4.

Tabla C.2- Significado de los índices de seguridad o medio ambiente.

| <b>Seguridad o Medio Ambiente</b> |  |
|-----------------------------------|--|
| 1                                 | No afecta la seguridad o el medio ambiente                       |
| 2-3                               | Remota posibilidad de afectar la seguridad o el medio ambiente   |
| 4-6                               | Posibilidad moderada de afectar la seguridad o el medio ambiente |
| 7-8                               | Grande posibilidad de afectar la seguridad o el medio ambiente   |
| 9                                 | Afecta la seguridad o el medio ambiente                          |
| 10                                | Afecta gravemente la seguridad o el medio ambiente               |

Tabla C.3- Significado de los índices de pérdida de facturamiento.

| <b>Perdida de Facturamiento</b> |  |
|---------------------------------|--|
| 1                               | No provoca pérdida de facturamiento  |
| 3-5                             | Puede provocar pérdida de facturamiento menor que 2,5% de la renta mensual       |
| 7-10                            | Puede provocar pérdida de facturamiento mayor o igual a 2,5% de la renta mensual |

Tabla C.4- Significado de los índices de corte de carga.

| <b>Corte de Carga</b> |   |
|-----------------------|---|
| 1                     | No provoca corte de carga   |
| 3                     | Riesgo remoto de provocar corte de carga  |
| 5                     | Riesgo moderado de provocar corte de carga                                      |
| 7                     | Provoca corte de carga de hasta 5% de la carga de instalación                   |
| 10                    | Provoca corte de carga mayor o igual a 5% de la capacidad máxima de instalación |

Tabla C.5- Significado de los índices de severidad.

| <b>Severidad</b> |  |
|------------------|--|
| 1                | Falla de menor importancia   |
| 2 – 3            | Provoca reducción de desempeño del componente                                |
| 4 – 6            | El componente sufrirá una degradación progresiva                             |
| 7 – 8            | El componente no desempeña su función  |
| 9                | Colapso del proceso  |
| 10               | Los problemas son catastróficos y pueden ocasionar daños a bienes o personas |

Los parámetros de evaluación de criticidad de fallas son descritos a seguir:

- **Severidad:** Se trata de un índice que refleja la gravedad de las consecuencias de una falla. Cuanto mayor es el índice, mayor es la gravedad. El índice de severidad fue asumido como el mayor índice entre los índices de las tres clases de efectos de fallas descritos anteriormente (seguridad y medio ambiente, pérdida de facturamiento o corte de carga). La Tabla C.5 presenta el padrón utilizado para cuantificación de la gravedad de las fallas en índices de severidad.
- **Ocurrencia:** Es un índice definido en función del número de ocurrencias de fallas registrados en un período considerado. La Tabla C.6 relaciona los valores y conceptos de los índices de ocurrencia.

- **Detección:** Es un índice construido con base en la estimativa de la probabilidad de una falla ser detectada, asumiéndose que haya ocurrido. La Tabla C.7 relaciona los valores y conceptos de los índices de detección.

Tabla C.6- Significado de los índices de ocurrencia de fallas.

| <b>Ocurrencia</b> |                                 |
|-------------------|---------------------------------|
| 1                 | Menor o igual a 1 en 8 años     |
| 2                 | 1 falla en el período analizado |
| 3                 | 2 fallas                        |
| 5                 | 3 fallas                        |
| 7                 | 4 fallas                        |
| 10                | 5 o mas fallas                  |

Tabla C.7- Significado dos índices de detección.

| <b>Detección</b> |                                    |
|------------------|------------------------------------|
| 1                | Probabilidad muy alta de detección |
| 2 – 3            | Probabilidad alta de detección     |
| 4 – 6            | Probabilidad moderada de detección |
| 7 – 8            | Probabilidad pequeña               |
| 9                | Probabilidad muy pequeña           |
| 10               | Probabilidad remota                |

## C.2 ANÁLISIS DE ÁRBOL DE FALLAS - FTA

Otra forma de identificar y catalogar las fallas funcionales de un sistema es a través del FTA. El FTA es un análisis deductivo detallado que generalmente requiere considerable volumen de informaciones sobre el sistema. Se trata de un modelo gráfico que permite mostrar la concatenación de los diferentes eventos relacionados con determinada falla.

De acuerdo con Araújo *et al.* (2001), el análisis de árbol de fallas consiste en la construcción de un diagrama lógico, a través de un proceso deductivo que partiendo de un evento indeseado predefinido, busca las posibles causas de tal evento. El proceso sigue investigando las sucesivas combinaciones de fallas de los componentes hasta llegar a las llamadas fallas básicas (o eventos básicos), las cuales constituyen el límite de resolución del análisis. El evento indeseado es comúnmente llamado de evento tope del árbol. Por tanto, el concepto fundamental de FTA consiste en la traducción de un sistema físico en un diagrama lógico estructurado, FT (*Fault Tree*), en que ciertas causas específicas conducen a un evento tope de intereses.

Estructurado el árbol, se procede a la definición de los cortes mínimos, esto es, de las combinaciones mínimas de eventos que cuando ocurren llevan a la falla del sistema.

La grande popularidad del FTA viene fundamentalmente de dos aspectos:

- Primero, da mayor flexibilidad de representación gráfica de sistemas complejos proporcionada por la simbología específica y;
- Segundo, da mayor facilidad computacional debido al menor número de algoritmos significativos necesarios para el cálculo de las probabilidades de fallas cuando es comparado a lo que requiere en el caso de los valores típicos de las probabilidades de suceso.

## APÉNDICE D

### INICIACIÓN Y OPERACIÓN DEL SIMPREBAL

En este Apéndice son descritos los pasos para iniciar el servidor y el cliente SIMPREBAL. También es presentada la descripción detallada de los componentes de la herramienta C&M, los procedimientos para la inicialización y operación y las actividades soportadas por el cliente tales como: cálculo de KPI, gráficos de variables *online* e históricas, mensajes de anomalías y toma de decisión.

#### D.1 SERVIDOR SIMPREBAL

La inicialización del *simprebalserver* es a través del archivo “server.bat” localizado en la carpeta principal del SIMPREBAL “...simprebal\simprebalserver”. Una vez inicializada aparece una ventana del *simprebalserver* en ejecución mostrada en la Figura D.1.



```
Simprebal Server
C:\Archivos de programas\Oracle Software Foundation\opclic2.2\ntdoc>SIMPREBAL\simprebalserver>java -cp .\dir\simprebalserver.jar;lib\activation.jar;lib\commons-email-1.0.jar;lib\jmail.jar;lib\jarspec.jar;lib\jclfa-1.2.9.jar;lib\jclfs-ntln-auth.jar;lib\jega.jar;lib\j-interop.jar;lib\log4j-1.2.12.jar;lib\mail.jar;lib\mysql-connector-java-5.0.6-bin.jar;lib\ocacn-ip-rcp.jar;lib\ocacn-mp.jar;lib\ntln-security.jar;lib\opencada-opc-dcon-0.2.0.jar;lib\opencada-opc-lib-0.2.0.jar;lib\opencada-opc-lib-test-0.2.0.jar;lib\opencada-util.jar;lib\opcio.jar;lib\sqljbc.jar;lib\kernel.ServerMain
Iniciando I-Kernel...
JOPClient (C) OPI 2000-2004. Single user license issued to:
'FINNIFC, Fundacio de Emp. Cient. e Tecnològics, alvarez@alvareztech.com' on 20
07-12-07
License serial no.: J00634
JOPClient 02.103b: server hostname=localhost prgid=8nar.DficleServer.0 threadId
=3176
Connecting to remote PC localhost
version2 server
```

Figura D.1- Servidor en ejecución.

## D.2 CLIENTE SIMPREBAL

Después de iniciado el *simprebalServer*, el lado cliente puede ser inicializado a través de un *browser*, puede ser Internet Explorer, Mozilla Firefox, etc. Escribir la dirección URL del SIMPREBAL y aparecerá en el browser la interface de *login* del SIMPREBAL, conforme se muestra en la Figura D.2, en la cual el usuario debe digitar la clave registrada.

Si el registro aún no ha sido realizado, clique en “*Cadastrar novo usuário*” y llene todos los campos solicitados. Efectuado el *login*, el usuario entrará en la interface inicial del SIMPREBAL mostrado en la Figura D.3.

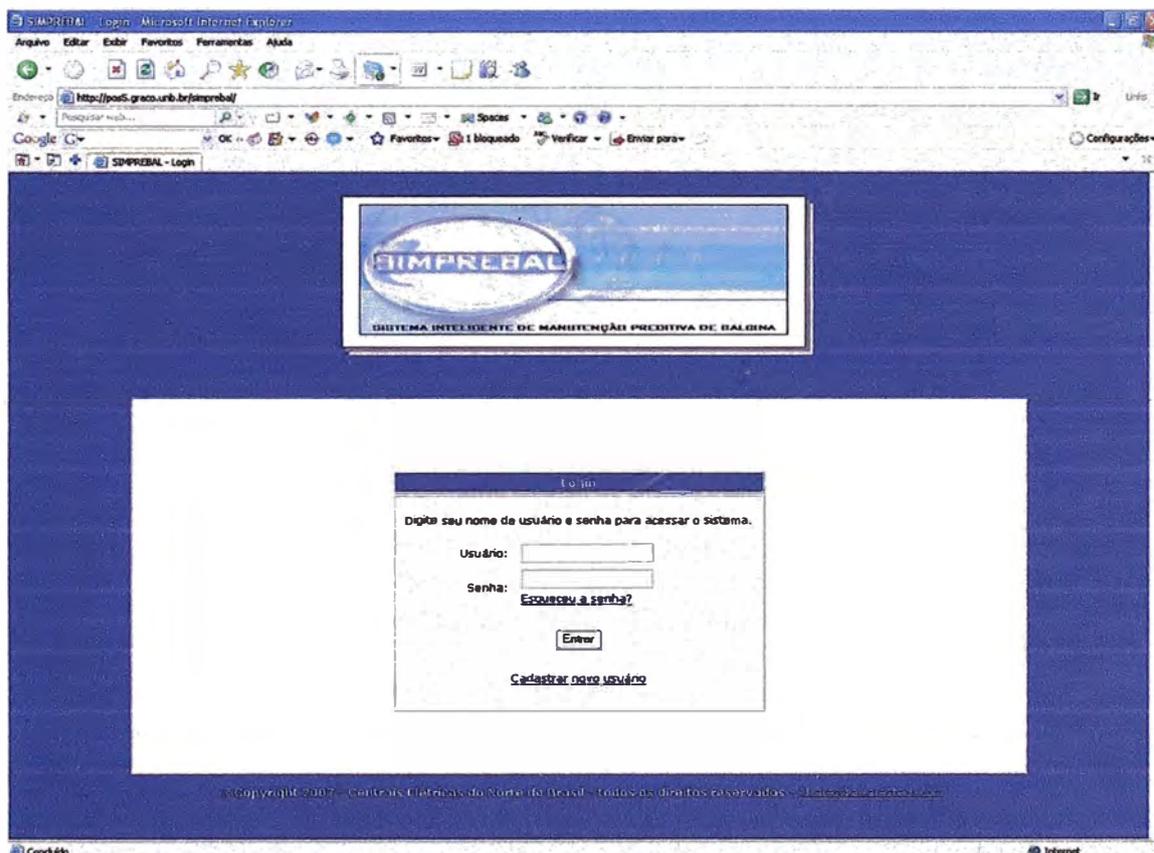


Figura D.2- Interface de *login*.

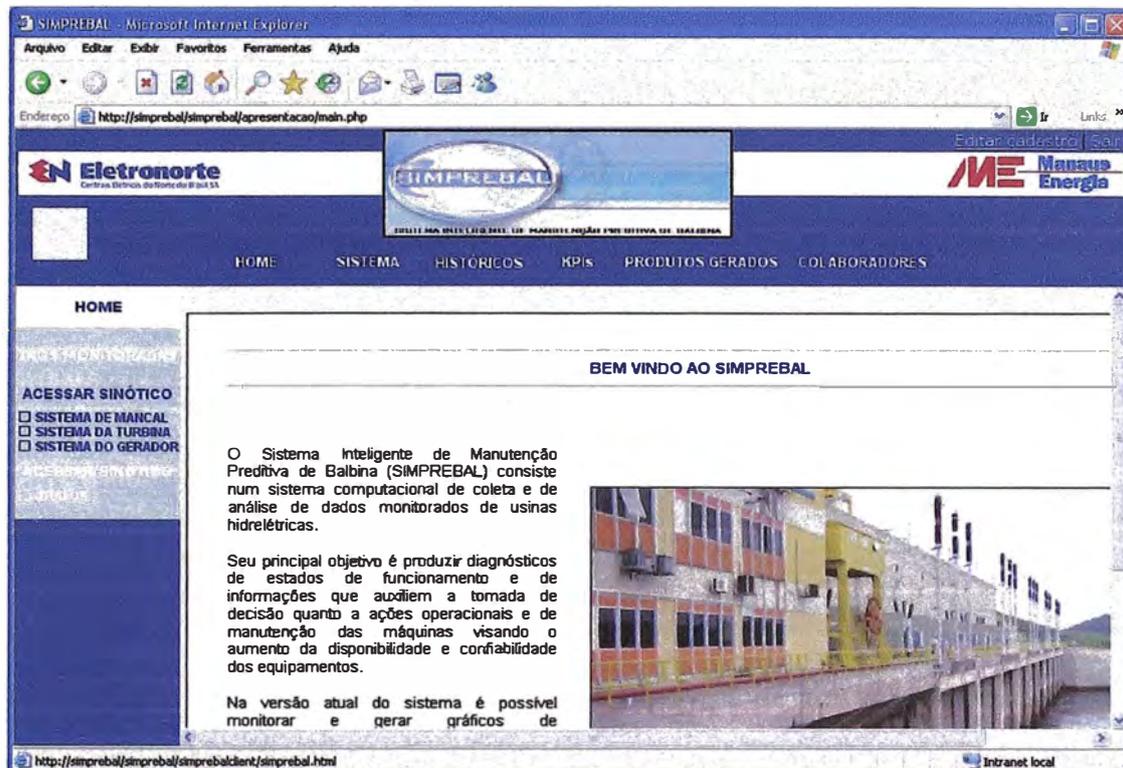


Figura D.3- Interface inicial.

A seguir se describe cada uno de los ítems del menú superior de la interface presentada en la Figura D.3.

### D.2.1 Home

Este menú contiene los siguientes submenús, a la izquierda:

- **Tags Monitoradas:** Permite visualizar cuales son las tags que actualmente están siendo monitoreadas por el SIMPREBAL y a cuales sistemas y equipos pertenecen.
- **Acessar Sinótico (Sistema de mancal, Sistema de la turbina, Sistema del generador):** Permite acceder al *SimprebalClient*, se trata de un supervisorio para mostrar las ocurrencias de fallas en el sistema, la variación de las tags, en tiempo real y a través de históricos, el

análisis de los modos falla y sus efectos de las anomalías ocurridas y sugerir OTM.

- Acessar Sinótico (Trafos): Este *link* corresponde a una expansión del sistema de transformadores.

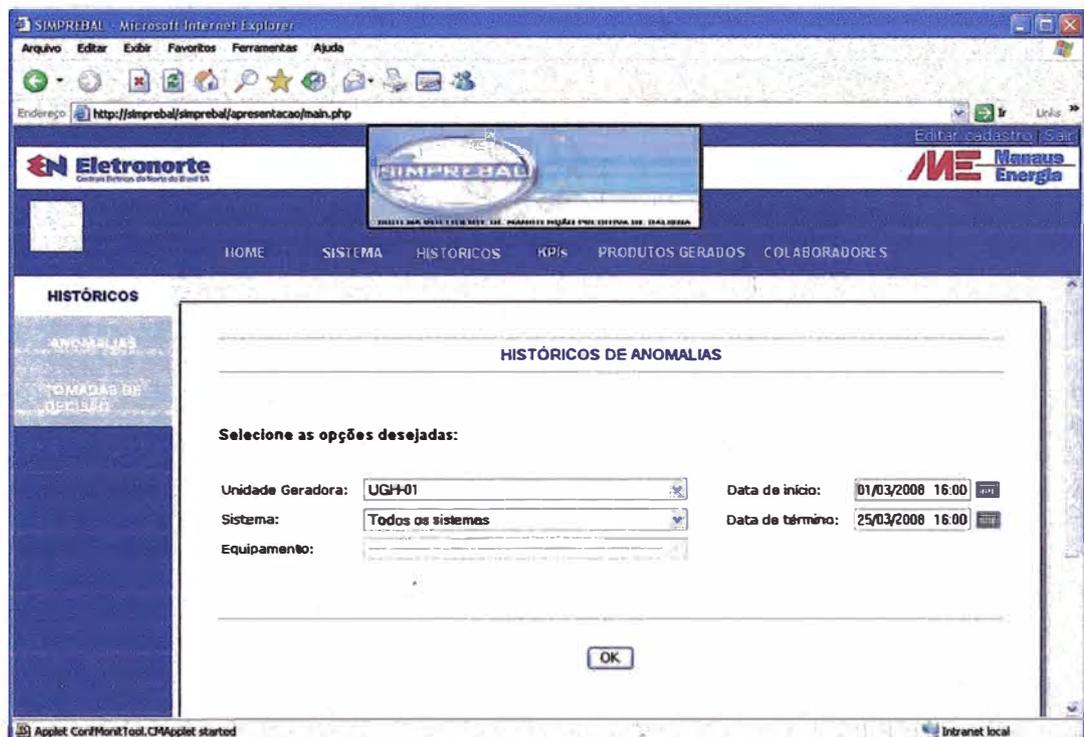


Figura D.4- Históricos de anomalías y selección de equipos.

### D.2.2 Sistema

Describe la metodología usada en el desarrollo del sistema. Contiene el submenú arquitectura que detalla la arquitectura de concepción del SIMPREBAL (arquitectura de siete capas del modelo OSA-CBM).

The screenshot shows a web browser window displaying the SIMPREBAL application. The page title is 'HISTÓRICOS DE ANOMALIAS'. Below the title, there are filters for 'Unidade Geradora: UGH-01', 'Sistema: Todos os sistemas', and 'Equipamento: Todos os equipamentos'. A table lists the anomaly records:

| Sistema            | Equipamento  | ID anomalia | Descrição                            | Causa                            | Data de início      | Data de término |
|--------------------|--------------|-------------|--------------------------------------|----------------------------------|---------------------|-----------------|
| Sistema de medição | Servidor OPC | 1283        | Falha ao conectar com o servidor OPC | O cabo da rede está desconectado | 2009-03-25 08:34:38 |                 |

Below the table is a form titled 'EDITAR DATA DE TÉRMINO' with fields for 'Digite o número da UGH:', 'Digite o ID Anomalia:', and 'Data de término:'. There is an 'Enviar informações' button at the bottom of the form.

Figura D.5- Históricos de anomalias.

### D.2.3 Históricos

Muestra los históricos de anomalías y de tomas de decisiones, tanto de los equipos de la central hidroeléctrica (mancales, generadores, turbinas) cuanto del sistema de medición (dispositivos, instrumentación *fieldbus*, servidor OP y servidor SIMPREBAL). Ingresando a los submenús de anomalías o decisiones surge una interface (Figura D.4) para selección del equipo, del sistema, de la unidad generadora y del período de tiempo deseado, siendo que es posible visualizar tanto las anomalías o decisiones de un equipo específico cuanto de todos los equipos de un determinado sistema o de todos

los sistemas de una determinada unidad generadora, de un único equipo de todas las unidades generadoras o de todos los equipos de todas las unidades generadoras de la central.

Seleccionada la opción deseada, los históricos podrán ser visualizados clicando en el botón OK. La interface de la Figura D.5 permite la visualización de un ejemplo de históricos de anomalías registrado en el banco de datos. Las informaciones de históricos son dispuestas en formato de tabla conteniendo los campos ID Anomalia, descripción de la anomalía, causa, fecha de inicio y fecha de término. El ID Anomalia es la llave primaria que identifica una anomalía en el banco de datos.

Además de mostrar los históricos de anomalías, la interface ofrece también la opción de editar una fecha de término de la falla. La importancia que las fallas estén con las fechas de término registradas para que sea posible calcular indicadores de desempeño tales como tiempo medio entre fallas, tiempo medio de reparo y tasa de falla.

EDITAR DATA DE TÉRMINO

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:

Figura D.6- Edición de la fecha de término de una anomalía.

Para editar la fecha de término de una determinada falla es necesario digitar el número de la unidad generadora y el ID de la anomalía en los respectivos campos de edición. Digitando un número de la UGH y un ID válido, aparecerán informaciones sobre la falla cuya fecha de término se desea editar en el cuadro debajo.

La fecha de término podrá ser editada seleccionando la fecha en que la referida falla paro de suceder, conforme mostrado en la Figura D.6. Después de seleccionar la fecha de término, se debe clicar en el botón para enviar informaciones y actualizar la nueva fecha de término de la falla en el banco de datos.

#### **D.2.4 KPIs**

KPIs (*Key Performance Indicators*) son indicadores clave de desempeño, o sea, indicadores que reflejan el progreso de la empresa en dirección a sus metas organizacionales. La Figura D.7 muestra una interface de cálculo de los KPIs para el sistema de enfriamiento y lubricación del mancal combinado. El menú KPIs contiene el submenú Calcular KPIs, que calcula los KPIs para los equipos seleccionados durante el intervalo de tiempo seleccionado. Las nueve KPIs son los siguientes:

Para cada una de las tags de un referido equipo:

1. Número de ocurrencias de ALERTA;
2. Número de ocurrencias de ALARMA;

3. Número de ocurrencias de TRIP;

Para cada equipo en si:

- 4. Número de ocurrencias de fallas;
- 5. Tasa de fallas;
- 6. Tiempo medio entre fallas;
- 7. Tiempo medio para reparo;
- 8. Número de prioridad de riesgo (factor de criticidad del equipo);

Y para el SIMPREBAL:

- 9. Porcentaje de decisiones acertadas con relación a las fallas del equipo escogido.

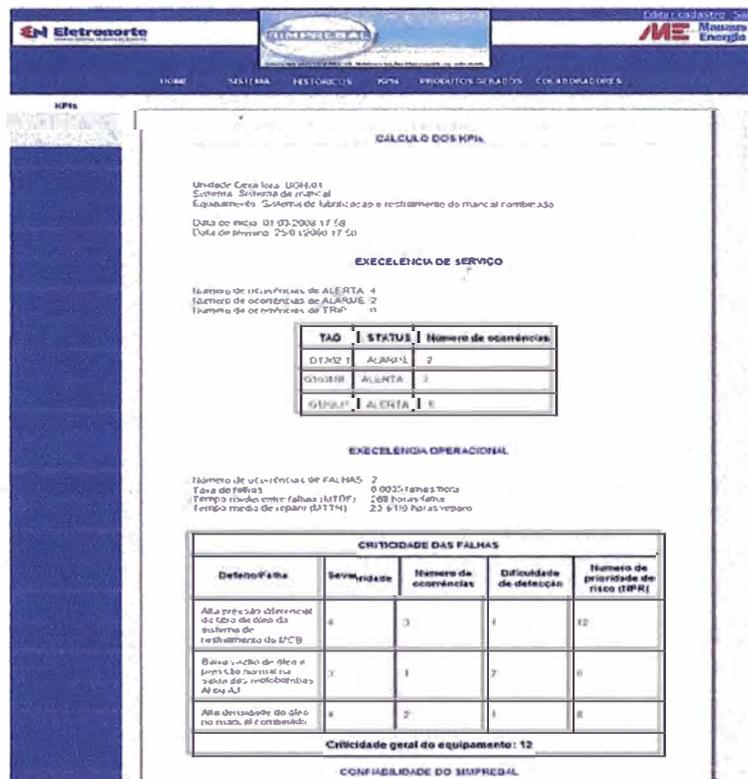


Figura D.7- Cálculo de los KPIs.

Una vez que una ALERTA no señala una falla, mas si un defecto, el número de ocurrencias de fallas consiste en el sumatorio de la cantidad de verificaciones de ocurrencias de ALARMES o TRIPS en cada una das tags o relaciones entre tags que permiten reconocer una falla, o sea, es el sumatorio de los modos de ALARMAS o TRIPS para un determinado equipo.

La tasa media de fallas es calculada usando la Ecuación D.1. El tiempo medio entre fallas es definido como el inverso de la tasa media de fallas.

$$\lambda = \frac{\text{n}^{\circ} \text{ de fallas del componente en el período considerado}}{\text{tiempo total que el componente estuvo operando}} \quad \text{D.1)}$$

El tiempo medio para reparo (MTTR) se calcula usando la Ecuación D.2.

$$\text{MTTR} = \frac{\text{tiempo total que el componente estuvo en reparo}}{\text{n}^{\circ} \text{ de reparos del componente en el período considerado}} \quad \text{(D.2)}$$

El número de prioridad de riesgo (NPR) de una falla es calculado usando la Ecuación D.3. El NPR es utilizado para la priorización de la toma de acción. Es una manera práctica de priorizar ciertas fallas y evaluar que precauciones deben ser consideradas en primer lugar. Vale resaltar que los índices de ocurrencia, severidad y detección mostrados en el Apéndice D fueron utilizados para evaluar la importancia de las fallas funcionales y, por tanto, atribuidos a cada una de las fallas funcionales de cada componente, en cuanto el NPR fue utilizado para evaluar los propios componentes. Consecuentemente, en los componentes que presentan más de una falla

funcional, el NPR del componente fue definida como igual al mayor NPR de sus fallas funcionales.

$$\text{NPR} = \text{Severidad} \times \text{Ocurrencia} \times \text{Detección} \quad (\text{D.3})$$

El factor de criticidad de un equipo corresponde al mayor NPR encontrado en las fallas de este mismo equipo dentro del intervalo de tiempo especificado.

La confiabilidad o porcentaje de aciertos del SIMPREBAL corresponde al porcentaje de fallas que fueron verificadas por un operador y señaladas como correctas, esto es, que corresponden a la realidad y es calculada usando la Ecuación D.4.

$$\% \text{Aciertos} = \frac{\text{n}^{\circ} \text{ de fallas acertadas}}{\text{n}^{\circ} \text{ total de fallas detectadas}} \quad \text{D.4)}$$

#### **D.2.5 Productos generados**

Este menú relata todos los documentos generados en función del proyecto de P&D Modernización del área de automatización de procesos de la central hidroeléctrica de Balbina, que resultó en el desarrollo del SIMPREBAL. Los submenús a la izquierda contienen informes de investigación, artículos publicados, manuales de mantenimiento y operación del sistema, y los cursos ofrecidos.

### **D.2.6 Colaboradores**

Este menú contiene los contactos del equipo de desarrolladores del SIMPREBAL, así como la identificación de los colaboradores *online*, además de los registros de los colaboradores que ingresaron el sistema en los últimos treinta días.

### **D.2.7 Editar Registro**

En el lado superior izquierdo de la interface hay una opción para el usuario editar su registro, pudiendo, por tanto, alterar nombre de usuario, nombre, apellido, email y cargo.

### **D.2.8 Sinóptico SIMPREBAL**

Al clicar en acceder sinóptico de mancal, turbina y generador, el usuario encuentra una interface semejante a la mostrada en la Figura D.8. En esta interface se pueden observar las figuras referentes a las cinco UGHs de la empresa. En cada figura están representados los equipos monitoreados por el SIMPREBAL. Los equipos que aparecen con la color verde están en funcionamiento normal, los equipos con la color amarilla están con defecto, o sea, poseen valores de las *tags* próximos a los valores de alarma (se dice que están en estado de alerta) y los equipos q están con la color roja (como es el caso del mancal guía de la turbina de la unidad generadora 1 de la Figura D.8) están en estado de alarma o *trip*.

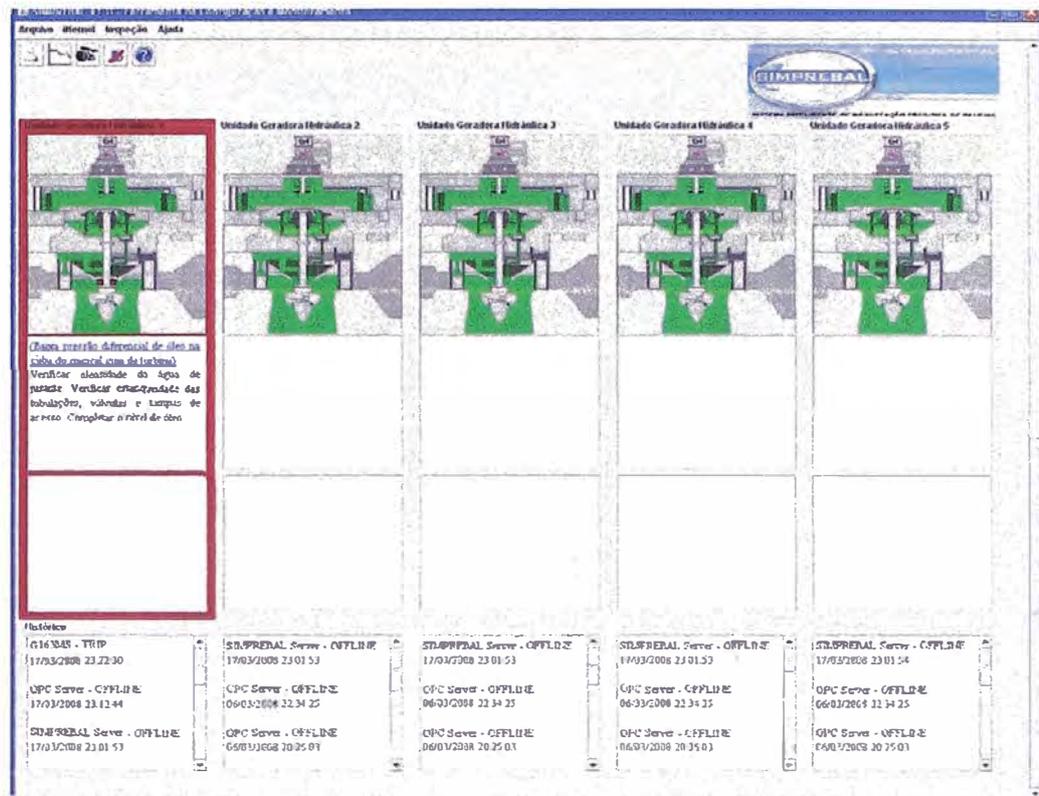


Figura D.8- Sinótico SIMPREBAL.

En el primer cuadrante abajo de la figura de cada UGH son presentadas las anomalías y las tomas de decisión referentes a los equipos del sistema de mancal, sistema de la turbina o sistema del generador. En el segundo cuadrante son presentadas las anomalías y las tomas de decisión referentes al sistema de medición, son, por tanto, fallas de procesamiento de señal. Y el tercer y último cuadrante muestra un histórico de las treinta últimas ocurrencias de defectos o fallas en cada UGH. El primer y el segundo cuadrante muestran la descripción de la anomalía entre paréntesis y en forma de un *link*, conforme observado en la Figura D.9.

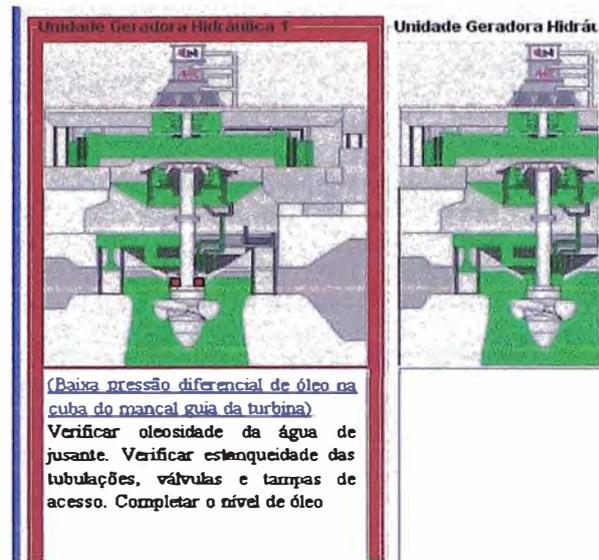


Figura D.9- Link para detallamiento de la anomalía.

Clicando en el *link* el usuario es direccionado para una página HTML conteniendo el análisis de los modos de falla y sus efectos (FMEA). Los botones de acceso, situados en la parte superior del SIMPREBAL, son respectivamente, de la izquierda para la derecha:

1. Botono Salir: Salir del sinóptico.
2. Botón Inspección de Variables: Inspeccionar las *tags* monitoreadas a través de visualización gráfica o por acompañamiento de cambio de valor.
3. Botón Camaras de seguridad: Accede a las cámaras de seguridad que eventualmente sean instaladas.
4. Botón *Shutdown*: Desconectar el servidor efectuando la operación de *shutdown* (este procedimiento sólo es posible mediante la digitación de una clave).
5. Botón Ayuda: Accede al manual de operación.

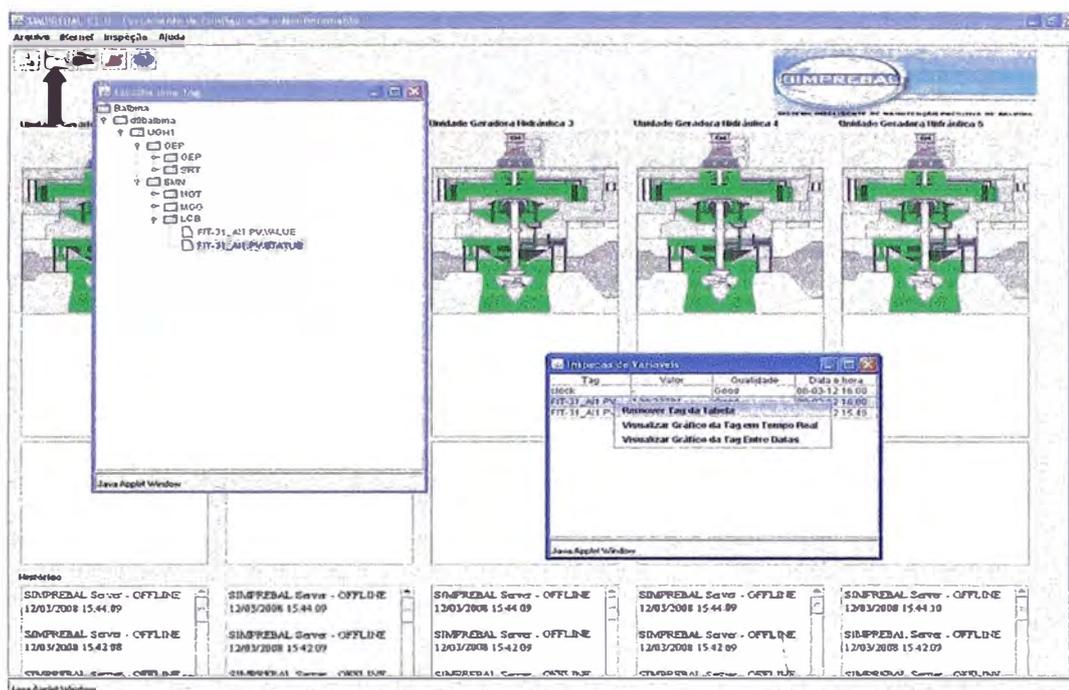


Figura D.10- Inspección de variables.

### D.2.9 Inspección de variables

Esta función permite que el usuario visualice el valor de las *tags* monitoreadas por el SIMPREBAL. Conforme mostrado en la Figura D.10 al seleccionar la opción inspección de variables el SIMPREBAL abrirá una ventana, con el título “*Escolha uma Tag*”, en la cual será presentado un árbol jerárquico con todas las 5 UGHs, los sistemas monitoreados, sus equipos y las *tags* de cada uno de ellos. Para visualizar los valores de las variables seleccione una *tag* y hacer doble *click* en ella. Aparecerá una ventana con el título inspección de variables. En esta ventana el usuario podrá acompañar la variación de los valores de las *tags* seleccionadas, así como verificar la calidad de la señal monitoreada.

Haciendo *click* con el botón derecho del mouse en una o más de estas variables, aparecerá un menú con las opciones remover *tag* de la tabla, visualizar gráfico de la *tag* en tiempo real y visualizar gráfico de la *tag* entre fechas. Para seleccionar el gráfico de más de una variable se debe seleccionar las variables deseadas manteniendo la tecla Ctrl presionada, conforme se muestra en la Figura D.11.

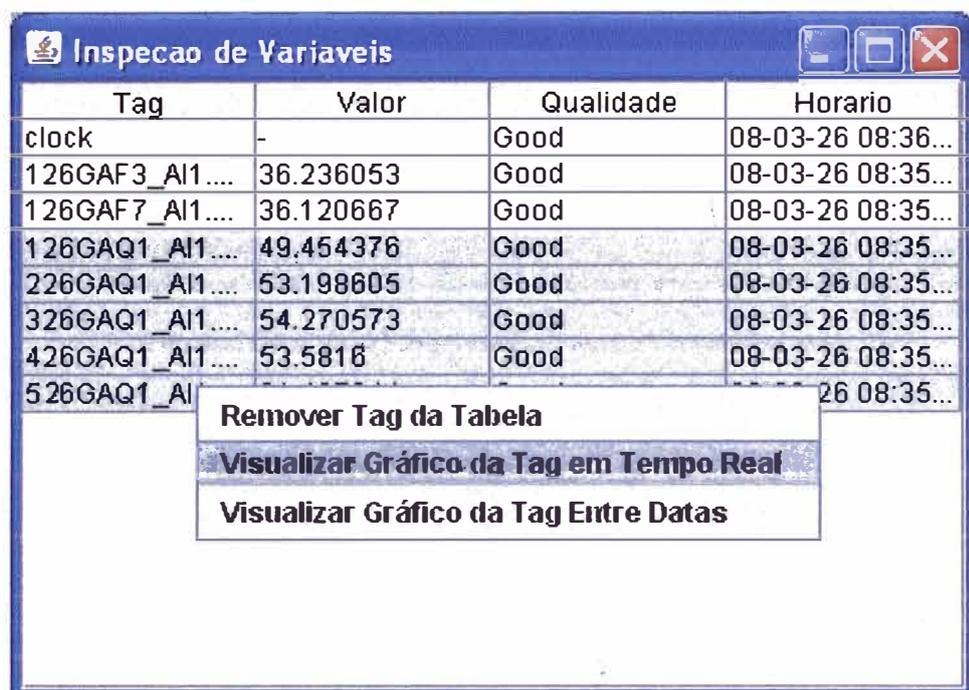


Figura D.11- Menú de la interface inspección de variables.

#### **D.2.10 Visualizar gráfico en tiempo real**

La opción visualizar gráfico de la Tag en tiempo real permite monitorear la variación de una o más tags a partir del instante en que se hace *click* en esta opción. Conforme se muestra en la Figura D.12.

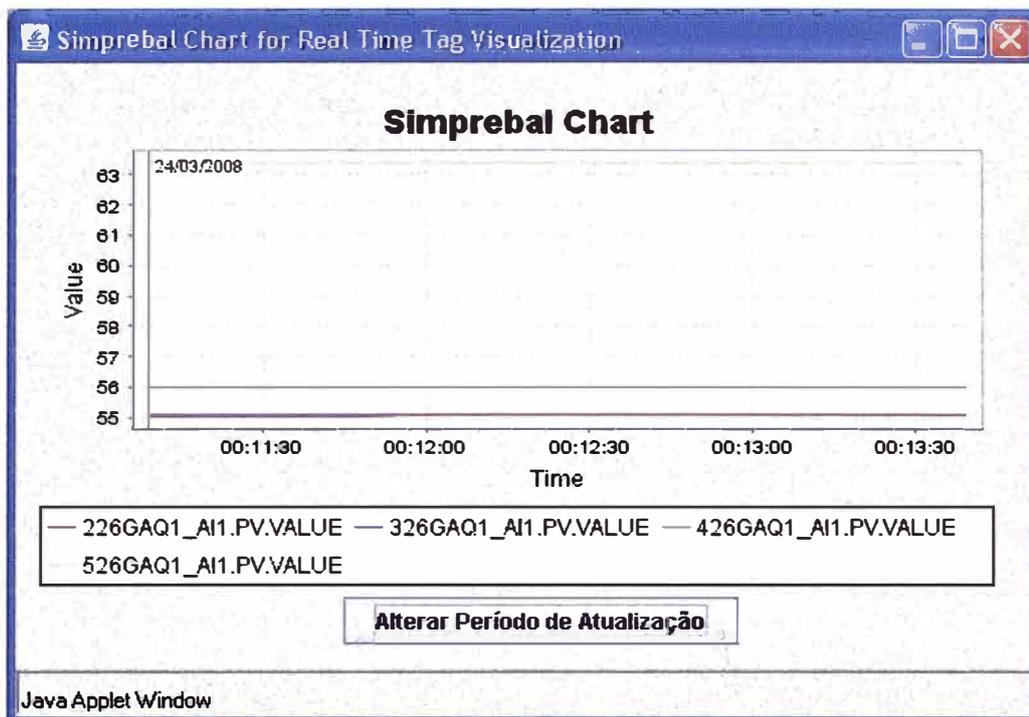


Figura D.12- Gráfico en tiempo real de temperaturas.

Figura D.13- Selección del intervalo de adquisición de los datos históricos.

### D.2.11 Visualizar gráfico histórico

Esta opción permite visualizar el gráfico de los valores de una o más tags entre dos instantes seleccionados, fechas de inicio y de término, conforme observado en la Figura D.13.

Haciendo *click* en el botón OK, el SIMPREBAL presentará un gráfico con la variación de los valores de las *tags* seleccionadas en el intervalo de tiempo determinado por el usuario, conforme es ilustrado en la Figura D.14.

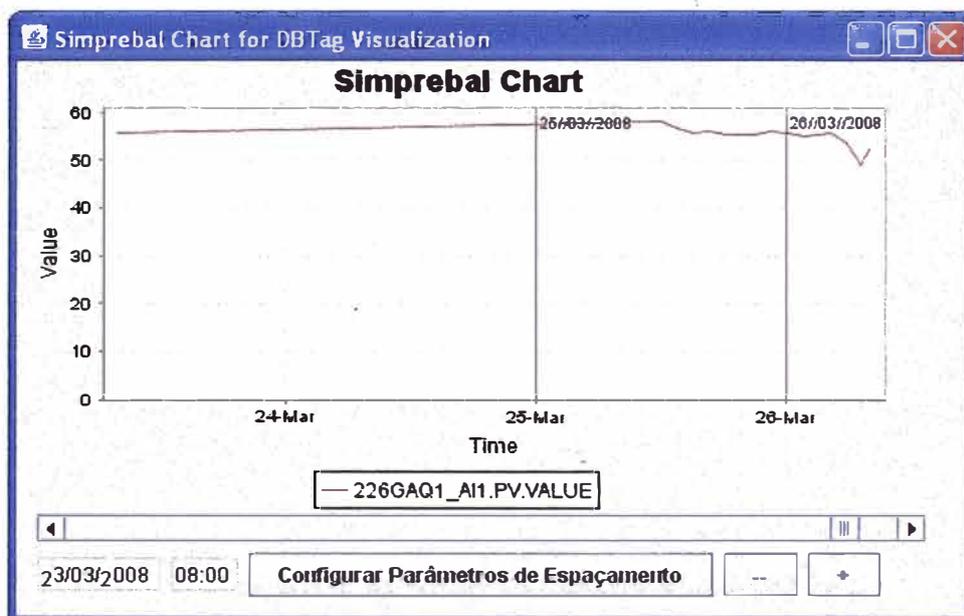


Figura D.14- Gráficos históricos