

***Universidad Nacional Mayor de San Marcos
Unidad de Post Grado de la Facultad de Ingeniería de Sistemas e Informática
(UPGFISI)
Maestría en Ingeniería de Sistemas e Informática
Mención en Ingeniería de Software***



**“Sistema multi-agente para medir la confiabilidad, disponibilidad y
fiabilidad de un sistema ERP”**

Autor: Ing. Angel Hermoza Salas

LIMA 2007

Ing. Angel Hermoza Salas

“Sistema multi-agente para medir la confiabilidad, disponibilidad y fiabilidad de un sistema ERP”

Disertación presentada en la Maestría de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional Mayor de San Marcos.

Aprobada por:

BANCA EXAMINADORA

Prof. Luis Rivera Escriba, Ph.D. – Orientador

Prof. David Mauricio Sanchez, Ph.D. – Co-orientador

LIMA 2007

A mis padres Lucas y Lidia.

Especialmente a mi esposa Edith, mis hijos Angel
y Luana.

AGRADECIMIENTOS

Un agradecimiento a mis padres Lucas y Lidia por apoyar mi educación y decidieron en el lejano 1981 enviarme a estudiar a la capital, Lima, lejos de mi lugar de origen, la ciudad del Cusco.

El agradecimiento especial a mi esposa Edith, quien con su colaboración y comprensión permitió que pueda quitar un tiempo grande a mis responsabilidades de padre para dedicárselo a la maestría.

Un mensaje para mis hijos Angel y Luana para que en su futuro como estudiantes puedan continuar con la senda del estudio y de la investigación que pretendo inculcarles.

Sin la invalorable asesoría profesional y colaboración del Dr. Luis Rivera Escriba desde Brasil no se hubiera podido conseguir avanzar durante estos meses, todo lo avanzado es fruto de sus comentarios, observaciones y revisiones permanentes lo que demostró que aún a lo lejos se puede asesorar en el desarrollo de una tesis a través de un medio moderno denominado Internet.

Un reconocimiento muy especial a mi amigo el Dr. David Mauricio Sanchez, quien con sus conversaciones y charlas me convenció de seguir estudiando y tuvo la visión de crear la maestría que hoy es una grata realidad.

INDICE

CAPÍTULO 1:	INTRODUCCIÓN.....	14
1.1	Monitoreo de los sistemas ERP.....	15
1.2	Formulación del problema	16
1.3	Objetivos	17
1.4	Propuesta.	18
1.5	Justificativa.....	19
1.6	Estructura del documento	19
CAPÍTULO 2:	ANTECEDENTES	20
2.1	Procedimiento típico	21
2.2	Software de apoyo	22
2.3	Trabajos relacionados existentes	25
2.4	Análisis de métodos.....	26
CAPÍTULO 3:	SISTEMAS CRÍTICOS	27
3.1.1	Confiabilidad en un sistema	30
3.1.2	Disponibilidad y fiabilidad en sistemas	32
CAPÍTULO 4:	SISTEMAS ERP	36
4.1	Posicionamiento mundial en soluciones ERP.....	37

4.2	Ventajas de un sistema ERP para empresas de servicios	38
-----	---	----

4.3	Ventajas un sistema ERP para las industrias de procesos.....	39
-----	--	----

CAPÍTULO 5: AGENTES DE SOFTWARE 40

5.1	Metodología de construcción de agentes	42
-----	--	----

5.1.1	¿Qué son los agentes colaborativos?	43
-------	---	----

5.1.2	La herramienta Zeus	44
-------	---------------------------	----

5.1.3	La librería de componentes del agente	45
-------	---	----

5.1.4	El software de construcción de agentes	48
-------	--	----

5.1.5	La metodología de desarrollo de agentes Zeus	49
-------	--	----

5.1.6	El software generador de agentes Zeus	52
-------	---	----

CAPÍTULO 6: IDENTIFICACIÓN DE INDICADORES..... 53

6.1	Indicadores contables y financieros.....	53
-----	--	----

6.2	Indicadores estratégicos o de gestión.....	53
-----	--	----

6.3	Dimensiones de los Sistemas Críticos	53
-----	--	----

6.3.1	Indicadores de disponibilidad	54
-------	-------------------------------------	----

6.3.1.1	Indicadores de base de datos.....	54
---------	-----------------------------------	----

6.3.1.2	Indicadores de sistema operativo	54
---------	--	----

6.3.2	Indicadores de fiabilidad	55
-------	---------------------------------	----

6.3.2.1	Indicadores del sistema ERP.....	55
---------	----------------------------------	----

6.3.3	Indicador de confiabilidad	55
-------	----------------------------------	----

6.4	Indicadores	55
6.5	Indicadores CODA.....	56
CAPÍTULO 7: MODELO DE AGENTES.....		59
7.1	Modelo general de agentes.....	59
7.2	Meta-modelo de organización.	59
7.3	Modelamiento de agentes	61
7.3.1	Modelamiento del agente de base de datos	61
7.3.1.1	Casos de Uso para el agente de base de datos.....	61
7.3.1.2	Agente de base de datos.....	63
7.3.2	Modelamiento del agente sistema operativo	65
7.3.2.1	Casos de uso para el agente de sistema operativo.....	65
7.3.2.2	Agente del sistema operative	66
7.3.3	Modelamiento de agente ERP	68
7.3.3.1	Casos de uso para agente del sistema ERP	68
7.3.3.2	Agente ERP	69
CAPÍTULO 8: EL SISTEMA CODA.....		72
8.1	Propósito	72
8.2	Alcance	72
8.3	Definiciones y abreviaturas.	72
8.4	Descripción de las perspectivas del producto	73

8.5	Requerimientos para el rendimiento adecuado del sistema	75
8.6	Dependencias	77
8.7	Requerimientos.....	77
8.7.1	Requerimientos específicos	77
8.7.2	Requerimientos funcionales	77
8.7.3	Requerimientos no funcionales	81
8.8	Diagrama de Casos de Uso	81
8.8.1	Diagramas de casos de uso general	81
8.8.2	Casos de uso de Tablas.....	82
8.8.3	Casos de uso Administración	82
8.8.4	Casos de uso Indicadores	83
8.8.5	Casos de uso alertas	83
 CAPÍTULO 9: EXPERIMENTOS NUMÉRICOS DEL SISTEMA CODA		85
9.1	Organización e instancias de prueba	85
9.2	Resultados numéricos	85
9.2.1	Indicador confiabilidad del sistema CODA (CODA).....	85
9.2.2	Indicador fiabilidad del sistema ERP (ERP001)	86
9.2.3	Indicador disponibilidad de la base de datos (BD001).....	87
9.2.4	Indicador disponibilidad del sistema operativo (SO001).....	87
 CAPÍTULO 10: CONCLUSIONES Y TRABAJOS FUTUROS.....		89

10.1	Conclusiones	89
10.2	Trabajos futuros.....	89
10.3	Posibles aplicaciones en las empresas	90
BIBLIOGRAFIA		92

Anexos:

1. Manual del usuario del módulo intranet.
2. Manual del usuario del módulo extranet.
3. Manual de Especificación de los requerimientos de software y análisis del sistema CODA

Lista de Figuras

Figura 1.1: Sistema actual.
Figura 1.2: Arquitectura del modelo propuesto.
Figura 3.1 Dimensiones de la confiabilidad.
Figura 3.2 Curva de costo/confiabilidad.
Figura 4.1 Sistemas ERP.
Figura 4.2 Módulos de un sistema ERP.
Figura 5.1 Grupos de Componentes.
Figura 5.2 Etapas de la metodología Zeus.
Figura 6.1 Arquitectura del sistema CODA
Figura 7.1 Modelo general de agentes.
Figura 7.2 Metamodelo de organización.
Figura 7.3 Casos de uso del agente de base de datos.
Figura 7.4 Modelo de agente base de datos.
Figura 7.5 Casos de uso agente sistema operativo.
Figura 7.6 Modelo de agente sistema operativo.
Figura 7.7 Casos de uso del agente ERP.
Figura 7.8 Modelo de agente ERP.
Figura 8.1 Casos de Uso General.
Figura 8.2 Casos de Uso Tablas.
Figura 8.3 Casos de Uso Administración.
Figura 8.4 Casos de Uso Indicadores.
Figura 8.5 Casos de uso alertas.
Figura 8.6 Modelo Conceptual.
Figura 9.1 Indicador de confiabilidad sistema CODA.
Figura 9.2 Indicador de fiabilidad del sistema ERP.
Figura 9.3 Indicador de disponibilidad de la base de datos.
Figura 9.4 Indicador de disponibilidad del sistema operativo.
Figura 10.1 Solución final (generalización).

Lista de Tablas

Tabla 2.1. Aplicativos de apoyo a especialista en sistemas de software.
Tabla 3.3. Terminología de la confiabilidad.
Tabla 6.1 Representación de los nodos CODA.
Tabla 6.2 Indicadores CODA.
Tabla 6.3. Indicadores del sistema CODA
Tabla 7.1 Instancias de Entidades de Meta-Modelo de Organización.
Tabla 7.2 Instancias de entidades de meta-modelo agente base de datos.
Tabla 7.3 Instancias de Entidades de Meta-Modelo.
Tabla 7.4 Instancias de entidades de meta-modelo agente sistema operativo.
Tabla 7.5 Instancias de entidades de meta-modelo.
Tabla 7.6 Instancias de entidades de meta-modelo agente ERP.
Tabla 7.7 Instancias de entidades de meta-modelo.

RESUMEN

El presente trabajo busca desarrollar un modelo y un prototipo del sistema multi-agente para medir la confiabilidad, disponibilidad y fiabilidad de un sistema ERP. Se describe cómo se monitorea, actualmente, los sistemas ERP, sea manualmente o con apoyo de otros sistemas; se menciona cuales son los roles que actualmente administran cada uno de estos sistemas; se muestra el problema actual de manera gráfica; se formula el problema; se detalla los objetivos del presente trabajo; se muestra gráficamente la solución final al problema; se determina la justificación del presente trabajo; se muestra los procedimientos actuales utilizados para resolver parcialmente el problema y se identifica los programas de software que dan solución parcial al problema, siendo los más usados por las empresas; se detalla la funcionalidad de los sistemas ERP; se revisa los sistemas ERP, el posicionamiento mundial de estos sistemas; se explica la teoría de sistemas críticos y la metodología de construcción de agentes Zeus así como su utilización en el desarrollo de los agentes del presente trabajo; se explica detalladamente la identificación de indicadores de desempeño del sistema ERP; se detalla el modelamiento de cada uno de los agentes de software utilizados en el desarrollo del sistema CODA utilizando la metodología Ingenias; se detalla la implementación de la aplicación sistema CODA y finalmente; se detalla las conclusiones y trabajos futuros.

Palabras Clave: Sistemas Críticos, Indicadores, Agentes de Software, Sistemas ERP.

ABSTRACT

“Multi-agent system to measure the trustworthiness, availability and reliability of an ERP system”.

This article aims to describe a system model and a software named system CODA, which is intended to recover information from the main server where is installed the Enterprise Resource Planning System (ERP Baan IV), the data base (Oracle) and the operating system (AIX), those tasks are performed by software agents, then this information will be processed and validated by the agents, finally the critical information is send to the system “CODA”; this system has two sub systems, the Intranet sub system will be managed by an Administrator, who will coordinate the creation of the indicators, maintain the users and clients, the parameters, basic tables. The Extranet sub system will be used by the clients (General Manager, Operations Manager, Information Manager), they will see graphs for each indicator defined in the system, also they will see how the system is running and also if there are problems, they can drill down the problems and see the detail. The system send mails and alerts to the technical persons who will solve the problems, each mail is associated with the error code, a description called “sintoma”, the real problem and a propose solution, when a new problem appears the administrator must add the solution to the data base, so if in the future the problem occurs, the system propose the solution.

Keywords: Critical systems, Indicators, Software Agents, Enterprise Resource Systems.

Capítulo 1: Introducción

El monitoreo de los sistemas es una tarea muy antigua. Consiste en estar atento a los errores, alertas o problemas reportados por los sistemas software, como los sistemas de Planeamiento de Recursos Empresariales (ERP - *Enterprise Resource Planning*), bases de datos y sistemas operativos. Para continuar operando, esos problemas deben tener solución inmediata: Los especialistas identifican la causa del problema dependiendo del problema, realizan determinadas acciones para resolverlos. Mientras tanto las operaciones se interrumpen y/o el sistema queda inoperativo por el tiempo que demande su atención. Ese lapso de tiempo disminuye con la ayuda de sistemas inteligentes alertando de posibles fallas, determinando e indicando las posibles alternativas a ser tomadas, si no se cuenta con ese tipo de apoyo, los especialistas tienen que aplicar procedimiento manuales o solicitar servicio de terceros (*outsourcing*) que muchas veces pueden demandar tiempo y dinero para resolverlo.

El Outsourcing es una modalidad, en la que determinadas organizaciones, grupos o personas ajenas a la compañía son contratadas para hacerse cargo de "parte del negocio" o de un servicio puntual dentro de ella. La compañía delega la gerencia y la operación de uno de sus procesos o servicios a un prestador externo (Outsourcer), con el fin de agilizarlo, optimizar su calidad y/o reducir sus costos.

Tradicionalmente los problemas de los sistemas se resuelven reactivamente. Es decir a partir del problema sucedido, el especialista busca una solución recurriendo a sus experiencias en problemas similares.

En contraste, las organizaciones modernas requieren que los sistemas críticos (core systems) estén operativos el mayor tiempo posible (si es necesario las 24 horas los 365 días del año), que existan sistemas de apoyo que permitan monitorear, medir y mostrar indicadores bajo los cuales la organización garantiza la atención continua a sus clientes, que estos indicadores puedan ser interpretados por los niveles de decisión a fin de tomar acciones oportunas.

1.1 Monitoreo de los sistemas ERP

El monitoreo de un sistema software puede ser realizado de dos formas: manualmente y/o con apoyo de sistemas inteligentes. El proceso manual es costoso y demora en comparación con los sistemas inteligentes que permiten realizar acciones oportunas, una vez detectada automáticamente el problema. Este enfoque puede minimizar grandemente el tiempo de tratamiento del problema y minimizar los costos.

Si se cuenta con personal especializado para el sistema software mencionado, es necesario contar con los siguientes especialistas: administrador del sistema ERP, quien revisa los archivos *logs* de errores del sistema ERP para ver qué sucedió luego que el usuario ha reportado un error por teléfono, *e-mail* o requerimiento escrito; un administrador de base de datos, quien procesa *queries* (lenguaje utilizado para recuperar información de la base de datos) para ver el origen del error en la base de datos, y un administrador del sistema operativo, quien revisa los *logs* de errores de los problemas de hardware o software.

Los logs desempeñan un papel importante en este esquema de monitoreo, pues en ellos se registran todos los eventos o mensajes ocurridos en el sistema. El log es un archivo creado por un sistema, así el sistema operacional crea un log que no necesariamente va a ser log de banco de datos, o de ERP. Estos archivos contienen una serie de mensajes, que para su interpretación, deben ser clasificados y asociados a tipos de indicadores para sus respectivos tratamientos.

La Figura 1.1 ilustra la situación actual de la operación del sistema ERP a través de un servidor que el usuario accede para recopilar información de los eventos que generó el problema, asimismo resolverlos. En esa situación, los sistemas ERP no cuentan con soporte software que permita recopilar, procesar y resumir la información, así mismo mostrar la información estadística respectiva de los acontecimientos, también no dispone de alertas, pues actualmente se reacciona cuando la falla ya ocurrió.

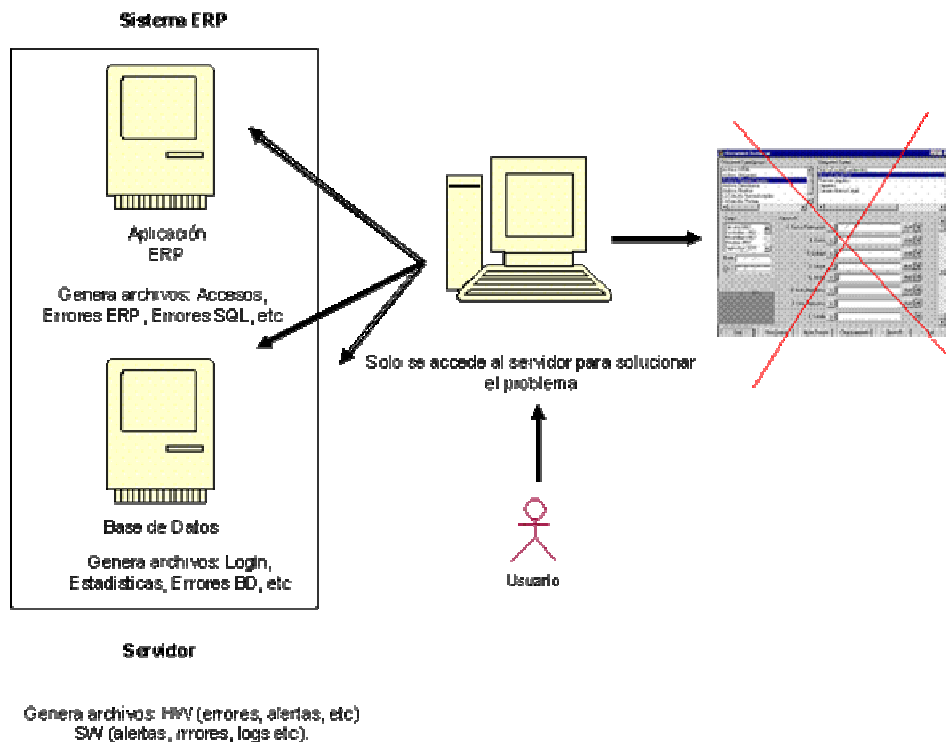


Figura 1.1: Sistema actual.

1.2 Formulación del problema

Los sistemas de software requieren un monitoreo continuo de sus actividades, debido a que realizan operaciones importantes dentro de la empresa. Bajo de estos sistemas, también se ejecutan otros componentes de sistemas de gestión y operaciones automatizadas de la empresa. Un ejemplo de caso de estudios, considerado en este trabajo, es el sistema ERP, la base de datos y el sistema operativo que demandan monitoreo continuo de sus actividades. Así, cuando ocurre algún problema (error o alerta operativa) se debe resolver o tomar acciones pertinentes por los especialistas humanos en estos sistemas.

En vista que actualmente las soluciones de esos problemas son frecuentes y aparentemente temporarios, produce un desgaste en muchos factores, como económicos, producción, etc. a la empresa. En virtud de estas deficiencias, la solución adoptada debe ser definitiva y la toma de acción debe permitir que las tareas operativas del sistema ERP siga el curso planeado. Para esto, se debe proveer al especialista con información integral, precisa y sencilla de entender, que indique tendencias, que alerte situaciones, que defina estándares y

patrones de comportamiento. Si fuera posible que dé soluciones automáticas, al menos en eventos de menor envergadura.

El tiempo de recuperación en caso de fallas graves pone en peligro la operatividad de la empresa ya que todas las operaciones son registradas en línea en este sistema. Esas deficiencias podrían resolverse con el establecimiento de indicadores que nos permitan conocer cómo está operando el sistema ERP o los procesos críticos de la empresa. Esto, por un lado, aliviará los problemas de contratación, por turno, de monitores humanos exclusivamente para esos sistemas.

1.3 Objetivos

Para la identificación e interpretación de las fallas de los sistemas software, será necesario la definición de los indicadores, estos serán categorizados de forma que permitan medir la confiabilidad, disponibilidad y fiabilidad del sistema ERP, base de datos y sistema operativo.

Modelará un sistema multi-agente que permita generar e interpretar los indicadores de confiabilidad, disponibilidad y fiabilidad del sistema ERP y mostrar estos indicadores a través de diagramas y tablas. Para este objetivo, se desea modelar agentes¹ de software. La definición de los indicadores de confiabilidad, disponibilidad y fiabilidad será tomada como referencia del sistema ERP, base de datos y sistema operativo. Al final, se definirá tablas de valores mínimos, normales y máximos a fin de mostrar las variaciones de los indicadores antes mencionados.

El sistema integral que permita el monitoreo de sistemas software, desde el punto de vista de medir la confiabilidad, disponibilidad y fiabilidad será desarrollado con el nombre de Sistema CODA. Para ello se elegirá un sistema ERP, base de datos y sistema operativo reconocido mundialmente, probar e implementarlo en un entorno web que permitirá mostrar los resultados a través de consultas, reportes, diagramas y correo electrónico.

¹ Un agente es un programa de software que puede realizar alguna acción definida para un usuario, con un nivel de inteligencia que le permite desarrollarla de alguna manera autónoma y que puede interactuar con su entorno.

1.4 Propuesta.

Construir un modelo basado en multi-agentes para reconocimiento de fallas y alertas del sistema ERP, base de datos y sistema operativo, denominado CODA.

La Figura 1.2 muestra el modelo propuesto en forma gráfica donde un solo sistema recolecta, a través de agentes de software, la información del sistema *ERP*, base de datos y sistema operativo; almacena esta información en otro servidor, procesa de acuerdo a parámetros establecidos, muestra la información a través de diagramas. Además permite tener la información histórica en reportes y consultas para el análisis.

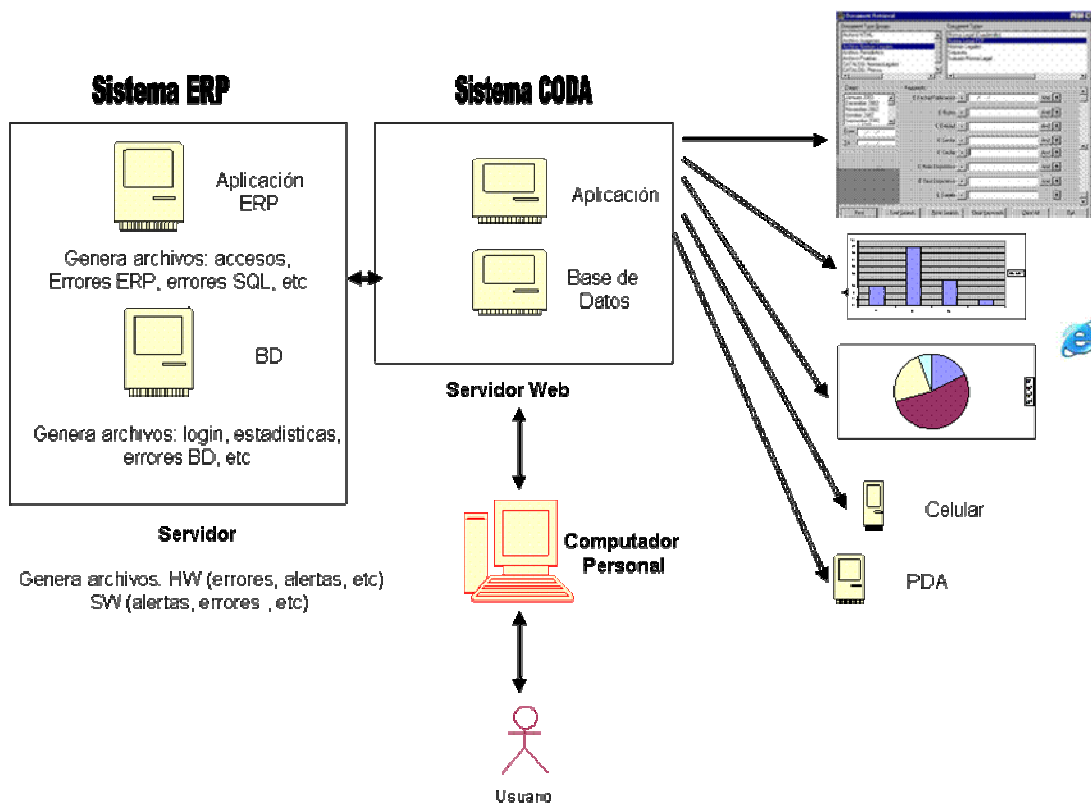


Figura 1.2: Arquitectura del modelo propuesto.

1.5 Justificativa

Con la definición de indicadores de confiabilidad, disponibilidad y fiabilidad y el modelamiento de agentes de software se podrá desarrollar un sistema a medida para el sistema ERP, base de datos y sistema operativo de cualquier empresa.

El desarrollo del sistema CODA permitirá a la empresa conocer los errores, problemas, soluciones, métricas del Sistema ERP, mostrar resultados de sustento y apoyo a la toma de decisiones, además servirá como modelo en el desarrollo de soluciones para otros sistemas de la empresa.

Este sistema enviará alertas y mensajes que permitirán tener informado en el instante a los técnicos encargados de mantener el sistema en operación.

1.6 Estructura del documento

En el capítulo 2 se analizan los antecedentes y los procedimientos actuales utilizados para resolver parcialmente el problema y se identifica los programas de software que dan solución parcial al problema, siendo los más usados por las empresas; en el capítulo 3 se explica la teoría de sistemas críticos; en el capítulo 4 se detalla la funcionalidad de los sistemas ERP; se revisa los sistemas ERP, el posicionamiento mundial de estos sistemas; en el capítulo 5 se detalla el modelamiento de cada uno de los agentes de software utilizados en el desarrollo del sistema CODA utilizando la metodología Ingenias, la metodología de construcción de agentes Zeus así como su utilización en el desarrollo de los agentes en el presente trabajo; en el capítulo 6 se explica detalladamente la identificación de indicadores del sistema CODA; en el capítulo 7 se muestra el modelo de agentes; en el capítulo 8 se muestra el sistema CODA; en el capítulo 9 se muestran los resultados numéricos del sistema CODA y en el capítulo 10 finalmente se detalla las conclusiones y trabajos futuros.

Capítulo 2: Antecedentes

Problemas que demanden soporte de fallas de sistemas software son ampliamente estudiados en diferentes niveles de concepción de la Ingeniería de Software. La Ingeniería de Software establece las técnicas de testeo de sistemas, programas y módulos durante su implementación y desarrollo; con eso, aparentemente los programas deberían funcionar sin defecto, pero siempre ocurren casos no contemplados, problemas de comunicación con otros componentes, problemas de sincronía, problemas de recursos, etc.

Las metodologías de verificación de programas establecen una serie de actividades para detectar errores de programas descritas en la etapa de testing o pruebas [Maldonado] del ciclo de vida del software además se utilizan modelos de verificación de programas (model checking) [Hung]. La programación modular establece metodologías de detección de fallas y resoluciones como el caso del uso de lenguajes semánticos y su verificación [George], así como la reingeniería trata de buscar alternativas de solución a partir del comportamiento de sistemas en funcionamiento, analizando sus defectos, sus fallas, etc.

Sin duda, existen procedimientos manuales de detección y resolución de problemas de fallas [Baan] [Oracle] [IBM] los que requieren buscar soluciones a problemas similares y a través de este medio tratar de solucionar el problema; como también las tendencias de automatización de algunos aspectos de esos procedimientos que serán expuestos más adelante, de forma que mejore el tiempo de atención como mejora del performance² de proceso de resolución de problemas.

² El término performance se tomó de la expresión inglesa “performance art”. El arte de la performance es aquel en el que el trabajo lo constituyen las acciones de un individuo o un grupo, en un lugar determinado y un tiempo concreto. Puede ocurrir en cualquier lugar, iniciarse en cualquier momento y puede tener cualquier duración. Cualquier situación que involucre cuatro elementos básicos: tiempo, espacio, el cuerpo del performer y una relación entre el performer y la audiencia, puede constituir el arte de la performance. En este sentido se opone a la pintura o la escultura, por ejemplo, en las que un objeto constituye el trabajo.

2.1 Procedimiento típico

Típicamente, el procedimiento de detección y resolución de fallas es seguido los manuales proporcionados por la empresa proveedora de los sistemas, por ejemplo de ERP: Verificación de posibles errores, eliminación de temporales, *backup* y *restore*, levantar servicios, bajar servicios, eliminar sesiones, determinar significado de errores.

Similar forma, son seguidos los procedimientos proporcionados por la empresa proveedora de la base de datos: Verificación de posibles errores, eliminación de temporales, observación de la Base de Datos, monitoreo de espacio de tablas, monitoreo de espacio de índices, monitoreo de espacio de extents³, *backup* y *restore*, corrección de errores de rollback⁴, levantar servicios, bajar servicios, aumentar espacio en tablas, índices, verificación de *backups* (diario, semanal, mensual), determinar significado de errores.

Lo mismo ocurre con los sistemas operativos, son seguidos los procedimientos proporcionados por la empresa proveedora del sistema operativo: verificación de posibles errores, eliminación de espacios temporales, *backup* y *restore*, levantar servicios, bajar servicios, verificación de backups (diario, semanal, mensual), determinar significado de errores.

En particular, para el caso específico de estudio como ejemplo, los sistemas de software, son seguidos los procedimientos adicionales dados por:

- Cursos proporcionados por la empresa proveedora del sistema *ERP Baan*, por cada módulo (capacitación funcional), herramientas de desarrollo (*tools*).
- Cursos proporcionados por la empresa proveedora de la base de datos Oracle: Administración de base de datos, *backup*, *restore*, seguridad.
- Cursos proporcionados por la empresa proveedora del Sistema Operativo AIX: administración, seguridad y comunicaciones.

³ Espacios reservados por la base de datos Oracle para almacenar la información, cuando excede el tamaño máximo definido la base de datos se detiene.

⁴ Proceso por el que la base de datos Oracle reversa las transacciones al estado anterior.

En cada procedimiento se necesita personal especializado para atender cada una de estas funciones.

2.2 Software de apoyo

Se han identificado software de apoyo a los especialistas en detección y resolución de fallas de los sistemas de software en estudio. Esas empresas de software son BMC [BMC], Quest Software [QUEST], Symantec [Symantec] e IBM [IBM].

Esas aplicaciones están orientadas a apoyar a los especialistas (administradores) en banco de datos y sistemas operativos. Tal como se muestra en la Tabla 2.1, ellos participan en el monitoreo de ERP, Base de datos y sistemas operativos, sólo BMC-SAP NetWeaver y Symantec i3 for SAP consideran el monitoreo de los tres sistemas de software, mientras los otros sólo consideran dos sistemas de software.

Empresa – Aplicativo	ERP	Base de Datos	Sistema Operativo
BMC – SAP NetWeaver	X	X	X
Queso Software -SAP		X	X
Symantic i3 for SAP	X	X	X
IBM – Tivoli		X	X

Tabla 2.1: Aplicativos de apoyo a especialista en sistemas de software.

A continuación procedemos a describir la funcionalidad de cada una de ellas.

BMC (SAP NetWeaver) [BMC]

Este programa administra el entorno del sistema ERP SAP, Base de Datos, y sistema operativo. No existe ninguna explicación técnica respecto a cómo se realiza esta administración. Incluye aplicaciones, data, calendarización y claves de acceso.

Millones de usuarios usan SAP para correr sus operaciones de negocios como finanzas, logística y comercio electrónico. Si la compañía tiene SAP, las caídas pueden afectar al personal, los procesos y la tecnología y su ventaja competitiva.

El programa BMC se encarga de:

- Administrar la disponibilidad: usando un administrador centralizado para reducir la degradación de la performance de la aplicación y asegurar la productividad.
- Administrar la complejidad: maneja todo los componentes de su sistema SAP desde una ubicación centralizada, automatiza los backups y restore para reducir los tiempos no planeados.
- Al minimizar el tiempo de caídas o fallas del sistema maximiza la productividad de la aplicación empresarial.
- Mejora los niveles de servicio proactive: incorpora medidas y reportes para detectar problemas antes que afecten al usuario.
- Envía la información vía Web a cada uno de los responsables del monitoreo del sistema SAP, para que pueda tomar conocimiento y si es necesario tomar acción respecto a problemas del sistema.
- Administración de base de datos para la aplicación empresarial asegura óptima performance, administración y recuperación de datos críticos en el entorno SAP, utilizando software de apoyo a los administradores se asegura la operatividad de la base de datos.

En este caso se trata de una herramienta que interactúa directamente con el sistema ERP SAP, la base de datos y el sistema operativo.

Quest Software para SAP [QUEST]

Quest software complementa la optimización, performance y disponibilidad del sistema SAP y de BMC (SAP NetWeaver), incluyendo la base de datos, aplicación, redes y la plataforma de sistemas. Provee las herramientas que el profesional de Tecnología de Información necesita para controlar eficientemente y administrar los entornos críticos.

Para ello ha desarrollado los siguientes sistemas:

Foglight : es una consola que monitorea cada tarea en su aplicación tecnológica crítica que incluye la base de datos, redes, aplicaciones, servidores, procesos automatizados, y servidores web, alertando sobre problemas antes que impacten en los usuarios. Estas alertas se muestran en la consola dependiendo del problema (rojo, verde o ambar).

LiveReorg: es un componente para Oracle, reorganiza su base de datos sin bajarla, mejora la performance aún en entornos sin mantenimiento. Todos los métodos de reorganización son diseñados para ayudar a mantener las bases de datos SAP en la capacidad óptima. Muchos usuarios de SAP lo utilizan para reorganizar las tablas de SAP sin impactar la disponibilidad de sus sistemas de producción.

En este caso se trata de herramientas que no interactúan directamente con el sistema ERP SAP. Sólo con la base de datos y el sistema operativo.

Symantec i3 for SAP

Características:

- Captura, mide y correlaciona las métricas de performance de cada unidad del sistema SAP (Servidor, aplicaciones, base de datos y almacenamiento).
- Muestra una imagen clara del sistema SAP desde la perspectiva del usuario final.
- Almacena la información para mostrar tendencias históricas.

Beneficios:

- Optimiza el tiempo de respuesta a los usuarios y mejora sobre todo la calidad del servicio encontrando la raíz de los problemas que causan la degradación del sistema.
- Reduce el tiempo de implementación de los proyectos de mejora del sistema SAP.
- Resuelve rápidamente los problemas y los elimina usando el aviso de un sistema experto.

En este caso se trata de una herramienta que interactúa directamente con el sistema ERP SAP, la base de datos y el sistema operativo.

IBM – Tivoli [IBM]

Esta herramienta brinda soporte a los sistemas ejecutados, las bases de datos y los sistemas operativos de una empresa, es de propósito general y brinda las siguientes prestaciones:

- Disponibilidad: asegurando el funcionamiento eficiente y saludable de sus recursos de TI.
- Administración de las aplicaciones empresariales: aísla, diagnostica y arregla los problemas de performance de las transacciones de negocios críticos.
- Administración de servicios de negocio: alinea los recursos de TI con las prioridades críticas del negocio.
- Orquestamiento y administración de recursos: provee y configura los servidores, software y dispositivos de redes.
- Administración de la seguridad: asegura la confidencialidad y la integridad de los datos.
- Administración del almacenamiento: protege y maximiza la utilidad y disponibilidad de sus datos.

En este caso se trata de una herramienta que interactúa directamente con el sistema, la base de datos y el sistema operativo.

2.3 Trabajos relacionados existentes

Problemas que demanden soporte de fallas de sistemas software son ampliamente estudiados en diferentes niveles de concepción de la Ingeniería de Software, estableciendo técnicas de desarrollo y pruebas. Supuestamente, los programas desarrollados considerando esas técnicas deberían funcionar sin defecto, pero siempre ocurren casos no contemplados, problemas de comunicación con otros componentes, problemas de sincronía, problemas de recursos, etc.

Existen modelos de sistemas inteligentes que permiten un monitoreo automático de detección de fallas de software y conocerlos en lo posible, en mayor parte, generando alertas para el usuario. Algunos de los últimos trabajos analizados son presentados en esta sección.

En cuanto a trabajo de investigación en monitoreo y mantenimiento de ERP, Kwon [Kwon 01] presenta un modelo inteligente para detección y mantenimiento de fallas de ERP. Su modelo se basa en agentes que interactúan, obedeciendo un control de sincronismo de

redes de Petri, para interpretar y analizar los errores de un sistema ERP; en algunos casos, dependiendo de la política, un agente trata el error del sistema. El modelo es eficiente para sistema ERP, pero no considera que los errores podrían ser relacionados en la interacción de ERP con sistemas operativos o banco de datos. También no se detallan posibles alertas que podrían generarse para uso de los administradores, en el caso que el sistema no pueda corregir el error automáticamente.

Chandrashekhar et al. [Chandrashekhar 03] enfocan el análisis completo del entorno de negocios con la ayuda de la tecnología de información, toma en cuenta varias interacciones entre diferentes procesos de negocio mientras se maneja un Sistema ERP, también provee una marco de referencia basado en diagramas de eventos para registrar los flujos lógicos y otros detalles relevantes, los que pueden ser utilizados para el diseño de un sistema de información. Los diagramas de evento son útiles para el diseño del sistema de información inteligente. Esta metodología puede ser utilizada para desarrollar sistemas. Cada evento puede ser analizado para requerimientos de datos, operaciones y recursos. Las interacciones entre muchas fuerzas, eventos, módulos, aplicaciones pueden ser vistas a través de los diagramas de interacción.

El uso de multiagentes en problemas de monitoreo es común en diversas líneas de computación, como gestión de negocios electrónicos, en detección de fallas y problemas de funcionamiento usados en auditoria, abordados por Ling-yu et al. [Ling-yu 07] en el modelo de auditoría ABCAM.

2.4 Análisis de métodos

En el caso de los programas existentes, sólo están disponibles para el sistema ERP SAP, que es el de mayor venta en el mundo y del mismo modo para las bases de datos Oracle y generalmente para sistemas operativos Unix, quedando sin resolver para las demás sistemas ERP (Baan, Oracle, etc), base de datos (MS SQL, DB2, Informix, etc) y sistemas operativos (Windows, Linux, etc).

El uso de multi-agentes de software para recopilar información, es el método de recopilación utilizado en los documentos técnicos revisados.

Capítulo 3: Sistemas críticos

Según Sommerville [Sommerville 02], todos estamos familiarizados con los problemas de las caídas en sistemas de cómputo. Por razones no obvias, algunas veces los sistemas de cómputo colapsan y dejan de suministrar los servicios que se les han solicitado. Los programas que se ejecutan sobre estas computadoras no operan como se espera; ocasionalmente, adulteran los datos administrados por el sistema. Hemos aprendido a vivir con estas caídas y pocos de nosotros confiamos completamente en las computadoras que utilizamos todos los días.

Los sistemas críticos son los sistemas no confiables, que con sus efectos de caídas pueden dar perjuicios a la institución.

La confiabilidad de un sistema de cómputo es una propiedad del sistema que es igual a su fidelidad. La fidelidad esencialmente significa el grado de confianza esperado del usuario en el sistema que operará y en el que el sistema no "caerá" al utilizarlo normalmente. Esta propiedad no se puede expresar numéricamente sino que se utilizan términos como "no confiable", "muy confiable" y "ultraconfiable" para reflejar diferentes grados de confianza en un sistema.

Por supuesto, la fiabilidad no es lo mismo que la usabilidad. El procesador de texto que se utilizó para escribir este trabajo no es, en la opinión del autor, un sistema muy confiable pero sí muy útil. Sin embargo para reflejar la falta de confianza que tengo en el sistema con frecuencia se almacena el trabajo y se mantiene varias copias de respaldo de él. Por lo tanto, se compensa a la falta de confianza en el sistema por medio de acciones que limitan el daño que podría generarse por una caída del sistema.

Existen cuatro dimensiones principales de la confiabilidad [Sommerville 02], como se muestra en la Figura 3.1, que son: Disponibilidad, Fiabilidad, Seguridad y Protección. Esas cuatro dimensiones, se puede conceptuar como:

1. La **disponibilidad** de un sistema es la probabilidad de que esté activo y en funcionamiento y le sea posible suministrar servicios útiles en cualquier momento.
2. La **fiabilidad** de un sistema es la probabilidad de que en un período, el sistema suministre los servicios correctamente tal como lo espera el usuario.
3. La **seguridad** en un sistema es una valoración de que tan probable es que el sistema provoque un daño a la gente o a su entorno.
4. La **protección** de un sistema es una valoración de que tan probable es que el sistema resista a una intrusión accidental o provocada.

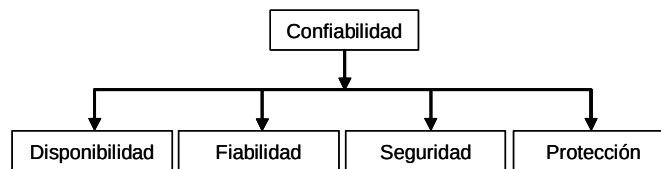


Figura 3.1: Dimensiones de la confiabilidad.

La disponibilidad y la fiabilidad son, esencialmente, probabilidades y por lo tanto, se pueden expresar de manera cuantitativa. La seguridad y la protección son valoraciones que se hacen basándose en evidencias del sistema. Estas raramente se expresan como valores numéricos, más bien se expresan en términos de niveles de seguridad, o sea por efecto.

Debido a los diseños adicionales y a la sobrecarga en la implementación y en la validación, incrementar la confiabilidad de un sistema puede incrementar dramáticamente los costos de desarrollo.

En la Figura 3.2, se muestra, [Sommerville 02], la relación entre los costos y las mejoras crecientes en la confiabilidad. Ese comportamiento se cumple para todos los aspectos de la confiabilidad, fiabilidad, disponibilidad, seguridad y protección. Debido a la naturaleza exponencial de esta curva costo/confiabilidad no es posible demostrar que un sistema es

100 por ciento confiable puesto que los costos de aseguramiento de la confiabilidad serían infinitos. Por lo general es cierto que los altos niveles de confiabilidad sólo se pueden lograr a costa del desempeño del sistema.

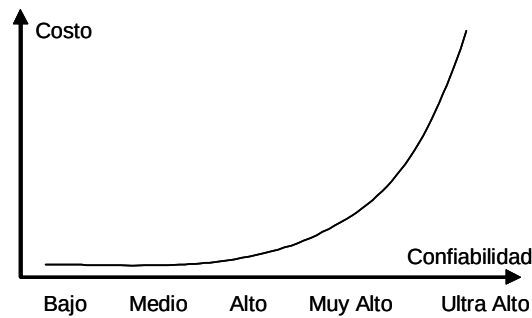


Figura 3.2: Curva de costo/confiabilidad.

El software confiable contiene código extra, a menudo redundante, para llevar a cabo la verificación necesaria respecto a los estados del sistema excepcionales y recuperarse de las caídas del sistema.

Esto reduce el desempeño del sistema e incrementa la cantidad de espacio de almacenamiento requerido por el software. Sin embargo, existen varias razones por las que la confiabilidad es por lo general un atributo más importante que el desempeño. Las razones pueden ser:

- A menudo, los sistemas que no son confiables, poco seguros o inseguros no se utilizan. Si los usuarios no confían en un sistema, por lo regular se rehúsan a utilizarlo. Más aún se rehúsan a utilizar los productos de la compañía que produjo el sistema no confiable puesto que tampoco estos son confiables.
- Los costos en las caídas del sistema son enormes: para algunas aplicaciones, como un sistema de reactores o un sistema de navegación aérea, los costos de las caídas del sistema son de órdenes de magnitud más grande que el costo de los sistemas de control.

- Es difícil reajustar la confiabilidad, por lo general es posible ajustar un sistema ineficiente puesto que mucho de tiempo de ejecución se invierte en secciones de programas pequeños. Un sistema que no es confiable es muy difícil de mejorar puesto que la no confiabilidad tiende a estar distribuida a lo largo del sistema.
- A menudo es posible compensar la falta de desempeño del sistema, algunas veces los usuarios ajustan su forma de trabajar acorde al desempeño pobre del sistema. Una falta de confiabilidad, en contraste, por lo general sorprende al usuario. El software no confiable destruye al sistema y a los datos del usuario sin ningún aviso, lo cual puede traer caídas con consecuencias que se manifiesten posteriormente.
- Los sistemas no confiables provocan la pérdida de información, es muy caro recolectar y dar mantenimiento a los datos, algunas veces cuesta más que el sistema de cómputo que se procesan. Se tiene que hacer un gran esfuerzo o invertir mucho dinero para duplicar los datos importantes a fin de protegerlos de la corrupción.

La confiabilidad del producto de software se ve influenciada por el proceso de software utilizado para desarrollar ese producto. Un proceso repetitivo orientado a evitar defectos probablemente desarrolle un sistema confiable. Sin embargo, no existe una relación entre producto y calidad del proceso.

Llevar a cabo un proceso en particular no permite que se hagan valoraciones cuantitativas de la calidad del producto.

3.1.1 Confiabilidad en un sistema

Los aspectos de confiabilidad son importantes para los sistemas críticos. Conseguir un alto nivel de confiabilidad es por lo general el requerimiento más importante de los sistemas críticos. Según [Sommerville 02], existen tres tipos principales de sistemas críticos:

1. **Sistemas de seguridad críticos:** Un sistema cuyo fallo provoca una lesión, pérdida de vida o un daño importante al medio ambiente. Un ejemplo de un sistema de seguridad crítico es un sistema de control para una planta química.

2. **Sistemas de misión crítica:** Un sistema cuya anomalía provoca caídas en algunas actividades dirigidas por objetivos. Un ejemplo de un sistema de misión crítica es el sistema de navegación para una nave aérea.
3. **Sistemas de negocios críticos:** Un sistema cuya anomalía provoca una caída en los negocios que utilizan el sistema. Un ejemplo de un sistema de negocios crítico es un sistema de cuentas de clientes de banco.

Los costos de las caídas de un sistema crítico a menudo son muy altos. Estos costos incluyen los costos directos de las caídas que requieren que el sistema se reemplace y los costos indirectos como son la litigación y las pérdidas en los negocios consecuencia de que el sistema no esté disponible. El alto costo potencial de las caídas significa que por lo regular la fidelidad de las técnicas y procesos de desarrollo es más importante que los costos de aplicar estas técnicas.

En consecuencia, los sistemas críticos se desarrollan por lo regular utilizando técnicas probadas en lugar de técnicas nuevas que no hayan estado sujetas a pruebas prácticas a fondo. Por ejemplo, hasta hace muy poco las técnicas orientadas a objetos se han estado utilizando para el desarrollo de sistemas críticos ya que el desarrollo de muchos temas críticos aún se basa en el diseño orientado a funciones.

Sin embargo, las técnicas de ingeniería de software que por lo regular no son costeables se utilizan para el desarrollo de sistemas críticos, por ejemplo el uso de especificación y verificación formales para la especificación de un programa. Una razón por la que se utilizan es que ayudan a reducir la cantidad de pruebas requeridas. Para los sistemas críticos, los costos de verificación y validación son muy altos y pueden consumir el 50 por ciento de los costos del desarrollo del sistema.

Cuando la confiabilidad es considerada en los sistemas críticos, existen tres tipos de componentes de sistemas susceptibles a caerse:

1. El hardware que puede fallar debido a errores en su diseño, debido a que los componentes se caen como resultado de errores en su manufactura o debido a que los componentes del hardware han llegado al final de su vida útil
2. El software que puede fallar debido a errores en su especificación, diseño o implementación.
3. Los operadores humanos de sistemas que pueden tener problemas al operar el sistema correctamente.

Por lo tanto, si la meta es mejorar la confiabilidad de un sistema, se tienen que considerar estos aspectos y sus interacciones.

3.1.2 Disponibilidad y fiabilidad en sistemas

La disponibilidad de un sistema es la probabilidad de que le sea posible suministrar servicios a sus usuarios cuando éstos se requieran. La fiabilidad es la probabilidad de que los servicios del sistema se suministren tal como se especificaron. Obviamente, la fiabilidad presupone la disponibilidad debido a que si un servicio especificado no se suministra, entonces el sistema no se comporta de acuerdo con su especificación.

Sin embargo, es útil distinguir entre estas características puesto que los requerimientos para la disponibilidad y la fiabilidad son diferentes. Por ejemplo, algunos sistemas pueden tolerar caídas relativamente frecuentes de las que se pueden recuperar muy rápidamente. Por lo tanto tienen requerimientos de fiabilidad relativamente bajos. Sin embargo, los mismos sistemas pueden tener requerimientos de disponibilidad muy altos debido a que los usuarios esperan que siempre estén en servicio continuo.

Un buen ejemplo de tal sistema es un interruptor de intercambio telefónico. Los usuarios esperan el tono de marcar cuando toman un teléfono de tal forma que el sistema tiene requerimientos de disponibilidad altos. Sin embargo si una falla en el sistema provoca una caída en la conexión, ésta es a menudo recuperable. Los interruptores de intercambio por lo general incluyen recursos de reparación que pueden reiniciar el sistema y tratar nuevamente el intento de conexión. Esto se hace muy rápidamente y el usuario del teléfono ni siquiera nota que ocurrió la caída. Por lo tanto, en tal situación, la disponibilidad más que la fiabilidad es un requerimiento clave de la confiabilidad.

Una diferencia adicional entre estas características es que la disponibilidad no depende sólo del sistema mismo, sino del tiempo que se requiere para reparar las fallas que provocan la falta de disponibilidad del sistema. Por lo tanto, si un sistema A se cae una vez por año y un sistema B se cae una vez por mes entonces A es más confiable que B. Sin embargo si al sistema A le toma tres días reiniciar mientras que al sistema B le toma 10 minutos, entonces la disponibilidad de B en todo el año es considerablemente más grande que A. Los usuarios probablemente preferirían el sistema B al sistema A.

Por consiguiente, podemos redefinir, en forma más precisa, la fiabilidad y la disponibilidad como:

- **Fiabilidad:** La probabilidad de que se tengan operaciones libres de caídas en un tiempo definido y en un entorno dado para un propósito establecido.
- **Disponibilidad:** La probabilidad de que el sistema, en cierto momento, esté operando y le sea posible suministrar los servicios requeridos.

La definición de fiabilidad establece que el entorno en el cual el sistema se utiliza y el propósito para el que se utiliza deben tomarse en cuenta al considerar su fiabilidad. Por

lo tanto, las medidas de fiabilidad en un entorno no son necesariamente transferibles a otro entorno en donde el sistema se utiliza de forma diferente.

Ejemplo: la fiabilidad de un sistema de software se puede medir en un entorno de oficina donde muchos usuarios no están interesados en la operación y no tratan de experimentar con él. En el entorno universitario, la fiabilidad es muy diferente. Aquí los estudiantes exploran las fronteras del sistema y lo utilizan de formas inesperadas. Esto conduce a caídas en el sistema que no ocurren en muchos entornos de oficina restringidos.

Una dificultad adicional con estas definiciones es que no toman en cuenta la severidad de las caídas o las consecuencias de la ausencia de disponibilidad. Naturalmente las personas están más concientes de las caídas en el sistema que tienen consecuencias serias y en las que la percepción de la fiabilidad del sistema se vea influenciada por estas consecuencias. Por ejemplo, es molesto el motor de un auto que se apaga inmediatamente después de haberse encendido, pero que inevitablemente funciona de manera correcta después de reiniciar. Sin embargo, esto no afecta la operación normal del auto. Y muchos conductores no pensarían que es no fiable. En contraste, muchos conductores pensarán que una máquina que se apaga mientras se conduce a alta velocidad una vez por mes es no fiable.

Una definición estricta de fiabilidad relaciona a la implementación del sistema con su especificación. Esto es, el sistema se comporta de forma fiable si su comportamiento es consistente con el definido en la especificación. Sin embargo una causa común percibida de ausencia de confiabilidad es que la especificación del sistema no cumpla las expectativas de los usuarios del sistema. Desafortunadamente, muchas especificaciones son incompletas o incorrectas y se deja a los ingenieros de software que interpreten cómo se debería comportar el sistema. Puesto que ellos no son expertos en el dominio, por lo

tanto no pueden implementar el comportamiento que esperan los usuarios. Las terminologías de fiabilidad, en ingeniería de software son ilustrados en la Tabla 3.3.

Por lo general, la fiabilidad y la disponibilidad se consideran las dimensiones más importantes de la confiabilidad. Si el sistema no es fiable, es difícil asegurar la seguridad o protección del sistema puesto que dependen de las caídas del sistema. Si un sistema no está disponible. Las consecuentes pérdidas económicas pueden ser muy altas. El software no fiable provoca costos altos a los usuarios finales. Los desarrolladores de sistemas no fiables adquirirán una mala reputación debido a la calidad y perderán oportunidades futuras de negocios.

Términos	Descripción
Falla del sistema	Un evento que ocurre en algún punto en el tiempo cuando el sistema no entrega un servicio como se esperaba por los usuarios
Error del sistema	Un estado erróneo del sistema que puede conducir comportamiento del sistema que es no esperado por los usuarios del sistema
Error humano	Comportamiento humano que resulta en la introducción de fallos en el sistema

Tabla 3.3: Terminología de la fiabilidad.

La fiabilidad está relacionada con las caídas del sistema. Estas pueden ser una caída al proveer el servicio, una caída en el suministro de un servicio especificado o la entrega de un servicio que sea inseguro. Algunas de estas caídas son una consecuencia de los errores en la especificación o de caídas en los sistemas asociados, como los sistemas de telecomunicaciones. Sin embargo muchas caídas son consecuencia de comportamientos erróneos en el sistema que conducen a fallas en el sistema. Cuando se discute la fiabilidad es de gran utilidad distinguir entre los términos falla, error y caída.

Capítulo 4: Sistemas ERP

Un sistema *ERP* (*Enterprise Resource Planning*), según Kwon [Kwon 01], es “un paquete de software amplio integrado empresarial diseñado para mantener los más altos estándares de calidad de los procesos empresariales”. Según APICS [APICS 02], un sistema ERP “es un sistema de información orientado al registro contable que sirve para identificar y planear los recursos de toda la empresa necesarias para comprar, producir, embarcar y registrar las órdenes del cliente”.

Un sistema ERP difiere de un sistema MRPII⁵ en los requerimientos técnicos tales como la interfase gráfica del usuario, base de datos relacional, uso de lenguaje de cuarta generación y herramientas de desarrollo e ingeniería asistidos por computadora, arquitectura cliente servidor y portabilidad. En general un método para el planeamiento efectivo y control de todos los recursos necesarios para comprar, producir, embarcar y registrar las órdenes del cliente en los servicios de manufactura, distribución o servicio de una compañía. Un sistema ERP está considerado como parte integrante de la Administración de la Cadena de Suministro (SCM).⁶ Sirve como columna vertebral de transacciones para otras aplicaciones de la cadena de suministro.

⁵ Los sistemas de Planificación de Requerimientos de Materiales (MRP) integran las actividades de producción y compras. Programan las adquisiciones a proveedores en función de la función de la producción programada.

⁶ Cadena de suministro o Cadena logística (en inglés, Supply Chain) es una expresión figurada para designar en la compleja serie de procesos de intercambio o flujos de materiales y de información que se establecen en cada organización o empresa con sus respectivos clientes y sus mismos proveedores.

La historia de los sistemas ERPs, enfocado por BDO [BDO 00], es que las compañías proveedoras se fundaron para proveer servicios de consultoría administrativa y financiera. Luego el objetivo siguiente fue proveer sistemas de información para toda la empresa, ampliamente aplicables y que son fácilmente adaptables a tecnologías cambiantes y necesidades de usuarios finales. Desde entonces han introducido nuevas generaciones de productos y han expandido sus operaciones para cubrir a la mayoría de los más grandes mercados del mundo. Hoy en día estos proveedores están catalogados como empresas líderes del mercado y como las más avanzadas en cuanto a tecnología se refiere.

Han buscado organizaciones enfocadas a reingeniería de procesos de negocios, diseñando las soluciones de software para brindar la flexibilidad y funcionalidad requeridas para soportar el cambio continuo e implementaciones rápidas. El software desarrollado soporta múltiples unidades y procesos operativos, es fácilmente escalable para apoyar niveles de operación cambiantes y puede ser reconfigurado para cubrir necesidades particulares.

La misión de estos proveedores es proveer a las empresas soluciones de software de negocios que sean innovadoras y que estén alineadas con las estructuras organizacionales, las prácticas de negocios y los procedimientos operacionales. Con una visión de construir la mejor organización de ventas y servicios de la industria, donde las personas trabajen conjuntamente y en un ambiente de grupo. Al lograr esto se conseguirán niveles de satisfacción incomparables y enfocar la empresa totalmente hacia **los** clientes.

4.1 Posicionamiento mundial en soluciones ERP

Los productos ERP existentes son presentados en la Figura 4.1, publicada por Gartner Group [GartnerGroup], en la que se muestra en cuatro cuadrantes a los principales proveedores de software *ERP* de clase mundial. En el cuadrante superior derecho,

denominado “cuadrante de los Líderes”, se encuentran a las casas de software más exitosas del mundo. En este cuadrante se encuentra los sistemas *ERP* que ofrecen las más completas e integradas funcionalidades y que al mismo tiempo aprovechan lo mejor de la tecnología, como SAP, Baan, Oracle. Los otros productos, próximos a los mencionados son SSA, IFS, Avalon, JBA, QAD, entre otros.

Un sistema ERP típicamente está conformado por varios módulos, tal como ilustra la Figura 4.2, como manufactura, distribución, finanzas, proyectos, servicios y transportes. También existen las conexiones con los otros paquetes (DEM, herramientas, interface del cliente, internet), así como con otros sistemas complementarios tales como administración de la cadena de suministro (*supply chain management*).

4.2 Ventajas de un sistema ERP para empresas de servicios

Las soluciones ERP poseen en conjunto la posibilidad de gestionar eficientemente sus necesidades de información en el área financiera, como lo son: cuentas por pagar; cuentas por cobrar; tesorería; contabilidad general; estados financieros; activos fijos; presupuestos; costos (ABC); control de proyectos; costo real del mantenimiento; análisis financiero.

Asimismo, se considera otras funciones vitales de soporte y administración que apoyan el área comercial. Así como cumplen importantes funciones dentro del entorno administrativo financiero aplicables al caso del cliente, aquí se presentan otras funcionalidades claves para las áreas de compras, ventas y almacenes, por ejemplo: contratos de compra; control de compras; cotizaciones de compra; control de órdenes de reabastecimiento (RPL); planeamiento de recursos de distribución (DRP); intercambio electrónico de datos (EDI); control de ubicación de inventarios; facturación; control de lotes; control de instalaciones; control de horas y de gastos; contratos de ventas; control de calidad; EIS - sistema de información gerencial.

4.3 Ventajas un sistema ERP para las industrias de procesos

Poseen en conjunto la posibilidad de gestionar en entornos integrados problemáticas tan disímiles, desde el punto de vista de sus necesidades de información, como son: la fabricación por procesos; las etapas de distribución y transporte, en las que se controla la producción discreta para optimizar economías de escala; la fabricación por lotes con características especiales para elaborar productos finales con la calidad deseada y para satisfacer pedidos específicos de clientes. Además de proporcionar funcionalidad para soportar fabricación continua (por procesos) y fabricación discreta (manufactura).

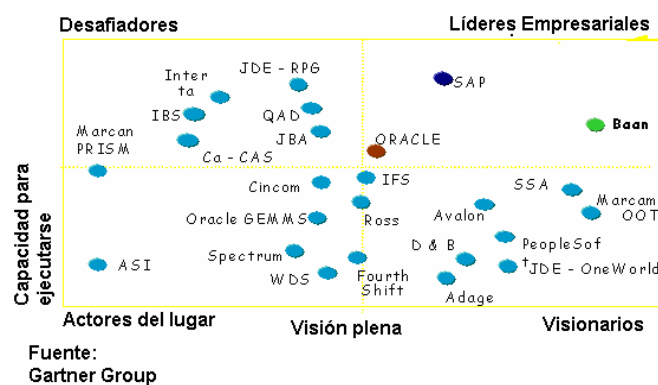


Figura 4.1 Sistemas ERP.

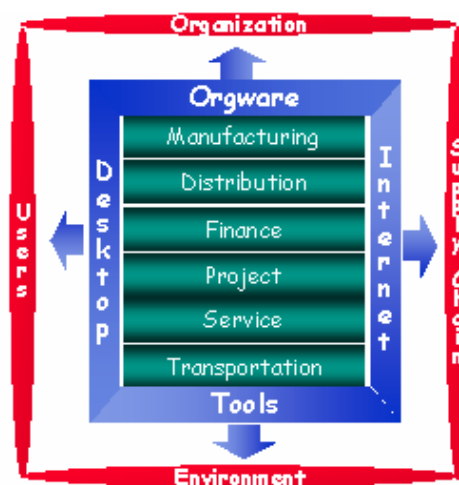


Figura 4.2 Módulos de un sistema ERP.

Capítulo 5: Agentes de software

Los agentes son un objeto de estudio popular de muchas investigaciones hoy en día en campos como Psicología, Informática en particular dentro del área de Inteligencia Artificial e Ingeniería de Software en donde se estudia más intensamente para solucionar problemas del mundo real, en una amplia gama de usos industriales y comerciales.

Según el resumen de la Universidad de Cambridge [Cambridge 96], los agentes de software son un área de investigación muy desarrollada. Sin embargo el sobre uso de la palabra agente ha tendido a enmascarar el hecho de que hay una verdad y un grupo de investigación bajo este término. La tipología de agentes servirá para colocarlo en el contexto, definirlos y una visión de cada caso para la crítica racional, mostrar las hipótesis, metas, retos y el estado del arte.

Los agentes de software han evolucionado de los sistemas multi-agente (MAS), los que a su turno forman una de las tres áreas en las que cae la inteligencia artificial distribuida (DAI), las otras dos son la resolución de problemas distribuidos (DPS) y la inteligencia artificial paralela (PAI). Por lo tanto, como con los sistemas multi-agente, heredan muchas de las motivaciones de DAI, los objetivos y beneficios potenciales.

El concepto de un agente según Hewitt es “objeto contenido en sí mismo, interactivo y ejecutado concurrentemente, denominado actor. Este objeto tiene algún estado interno encapsulado y puede responder los mensajes de otro objeto similar: un actor”.

¿Qué es un agente?

Los investigadores de agentes no manejan este término como es usado diariamente como agentes de viaje, agentes del estado, aún dentro de la fraternidad de software, la palabra agente es realmente un término paraguas para un cuerpo de investigación y desarrollo. La respuesta de algunos investigadores de agentes a esta falta de definición ha sido inventar otros sinónimos que incluyen Knowbots (knowledge-based-robots), softbots (software-robots), taskbots (task-based-robots), userbots, robots, personal agents y asistentes personales. Hay muy buenas razones para tener estos sinónimos. Los agentes vienen de

manera física, por ejemplo, aquellos que habitan el mundo físico, algunas fábricas los llaman robots, los que habitan en los computadores se denominan softbots, los que realizan tareas específicas taskbots, y los agentes autónomos se refieren típicamente a los agentes móviles o robots que operan en ambientes dinámicos e inciertos. Los agentes pueden jugar muchos roles, por ejemplo los knowbots, que tienen conocimiento experto. Debido a la multiplicidad de roles de los agentes hay una variedad de adjetivos que preceden a la palabra agente, search agents, report agents, presentation agents, navigation agents, role-playing agents, management agents, search and retrieval agents, domain-specific agents, development agents, analysis and design agents, testing agents, packaging agents and help agents.

Definimos un agente refiriéndonos a un componente de software el cual es capaz de actuar en razón de complementar tareas o comportamientos de su usuario.

Tipología de agentes

Una tipología se refiere al estudio de los tipos de entidades. Hay muchas dimensiones para clasificar los agentes de software existentes.

Los agentes pueden ser clasificados por su movilidad, por su habilidad para moverse alrededor de alguna red. Este cubre las clases de agentes estáticos o agentes móviles.

Pueden haber clases como deliberativas o reactivas. Los agentes deliberativos derivan del paradigma del pensamiento deliberativo: los agentes poseen una simbología interna, modelo razonado y se relacionan en el planeamiento y la negociación para lograr coordinación con otros agentes. Los agentes reactivos no tienen nada interno, modelos simbólicos de su entorno, y actúan usando el tipo de comportamiento estímulo/respuesta para presentar el estado del entorno en el que se encuentra. Los agentes pueden ser clasificados sobre muchos atributos ideales y primarios los que los agentes deben mostrar.

Se ha identificado una lista mínima de tres: autonomía, aprendizaje y cooperación. La autonomía se refiere al principio que los agentes pueden operar por sí solos sin la necesidad de guía humana aún cuando esto sea no valorado. Estos tienen estados internos individuales y metas, actúan de manera que logran los objetivos. Un elemento clave de su autonomía es su proactividad, su habilidad para tomar la iniciativa en lugar de sólo actuar en respuesta a su entorno. La cooperación con otros agentes es el modo de tener múltiples

agentes, estos necesitan poseer una habilidad social, la habilidad de interactuar con otros agentes y posiblemente humanos a través de un lenguaje de comunicación. Por lo que es posible para los agentes coordinar sus acciones sin cooperación. Los agentes inteligentes tienen que aprender como reaccionar y/o interactuar con el entorno externo.

Según [Gomez 02], se han identificado siete tipos de agentes:

1. Agentes colaborativos
2. Agentes de interfase
3. Agentes móviles
4. Agentes de información/internet
5. Agentes reactivos
6. Agentes híbridos
7. Agentes inteligentes

Existen aplicaciones que combinan agentes de una o más categorías y nos referimos a ellos como sistemas de agentes heterogéneos.

Otro asunto a tomar en cuenta es que los agentes necesitan ser benevolentes unos a otros. Es muy probable que los agentes pueden estar en competencia con otros, o quizás ser antagonista. Sin embargo se ven agentes competitivos como subclases potenciales de estas clases.

5.1 Metodología de construcción de agentes

Idealmente los desarrolladores deben emplear menos tiempo acerca de lo intrincado de una tecnología en particular, y más tiempo implementando soluciones para esos problemas. Mientras que las tecnologías maduren las soluciones genéricas independientes de la aplicación tienden a ser desarrolladas para ayudar a los usuarios de la tecnología.

La tecnología de agentes no es la excepción a este principio general, y hay un consenso emergente sobre los investigadores de agentes de la necesidad de desarrollar metodologías y herramientas para desarrollar sistemas distribuidos de agentes. Muchas herramientas han sido desarrolladas para ayudar en el desarrollo de los agentes.

La filosofía de transitar de las soluciones puntuales a los marcos generales, arquitecturas y metodologías fue la motivación para desarrollar la herramienta de desarrollo de agentes Zeus. La intención ha sido facilitar a la ingeniería de futuras aplicaciones de agentes proveyendo una librería de agentes a nivel de componentes, junto con una metodología de desarrollo de agentes que describe como los agentes pueden ser hechos utilizando los niveles básicos de esta librería.

Hay sin embargo diferentes tipos de agentes. La herramienta Zeus fue desarrollada para asistir a la creación de sistemas de agentes colaborativos.

5.1.1 ¿Qué son los agentes colaborativos?

Según [Zeus], las características esenciales de un agente colaborativo son:

- Ser capaces de comunicarse: esto es actualmente un problema a resolver, se requiere no sólo un protocolo de comunicación, sino también un lenguaje de comunicación común inter-agentes y una ontología compartida que represente los conceptos del dominio a ser comunicados.
- Ser capaces de razonar: cuando el control de un recurso (como un servicio o suma de valores) es delegada a un agente, se necesita suficiente inteligencia para decidir cuando dejar el control del recurso a un usuario potencial, o utilizar el recurso para lograr un objetivo, por tanto los agentes colaborativos necesitan determinar cuanto mejor lograr sus objetivos dado sus limitados recursos.

- Ser capaces de cooperar: porque un simple agente posee limitadas competencias. Recursos o información, será capaz de resolver sólo una pequeña porción del problema, y así un agente colaborativo se encuentra con el problema que no puede solo, de modo que debe encontrar otros agentes que poseen habilidad, delegar a otros las tareas, coordinación del comportamiento de diferentes agentes de modo que actuando juntos den solución al problema.

Dado que los requerimientos descritos a la funcionalidad de los agentes colaborativos, agentes en lugar de aplicaciones particulares, no son fáciles de satisfacer, consecuentemente estos agentes tienden a ser grandes, complejos y difíciles de implementar.

5.1.2 La herramienta Zeus

Fue motivada por la necesidad de proveer herramienta de construcción de agentes colaborativos, genéricos, customizable y escalable. En sí es un paquete de clases implementada en Java, que permite ejecutarse en una variedad de plataformas de hardware. Java es un lenguaje ideal para el desarrollo de aplicaciones multiagente porque es orientada a objetos y multihilos y cada agente consiste de muchos objetos y varios hilos de ejecución. Java también tiene la ventaja de ser portable a una variedad de librerías de clases que incluye excelentes facilidades de comunicación.

Las clases de la herramienta Zeus pueden ser categorizadas en tres grupos:

4. Una librería de componentes del agente.
5. Una herramienta de construcción de agentes.
6. Una herramienta de visualización de agentes.

Cada uno de estos grupos de componentes se muestran en la siguiente Figura 5.1:

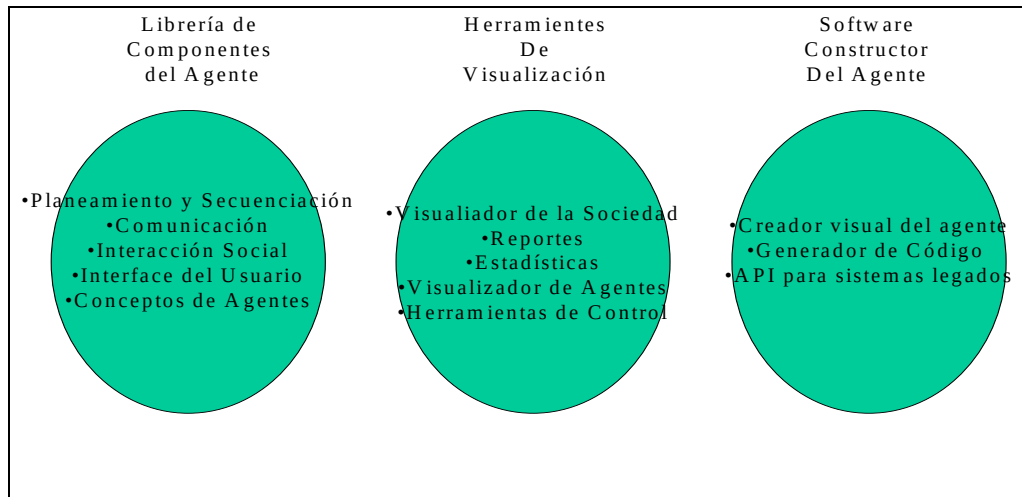


Figura 5.1: Grupos de Componentes.

5.1.3 La librería de componentes del agente

Es un paquete de clases de Java que forman los bloques de construcción de los agentes. Juntas estas clases implementan la funcionalidad de nivel de agente requerida por los agentes colaborativos. Así la herramienta de comunicación provee:

- Una preformativa basada en el lenguaje de comunicación interagente.
- Representación del conocimiento y almacenaje usando ontologías.
- Un sistema de pase de mensajes asíncrona basada en sockets.

De modo que se maximice la futura compatibilidad, los componentes de la herramienta Zeus utiliza tecnología estándar donde sea posible: por ejemplo, la comunicación toma lugar a través de sockets TCP/IP usando un lenguaje basado en KQML (Knowledge Query Management Language) y para el ACL (Agent Communication Language) se ha adoptado la especificación FIPA (Foundation for Intelligent Physical Agents).

Luego para proveer a los agentes de las facilidades de razonar, la herramienta provee:

- Un sistema genérico de planeamiento y secuenciación.

- Un modelo de eventos con una interface de aplicación de programación (API) que permite a los programadores de aplicación monitorear los cambios en el estado interno de un agente, y externamente controlar su comportamiento.

Para activar la cooperación interagentes, ofrece un motor de coordinación que controla el comportamiento de un agente – la funcionalidad de este motor es influenciada por el conocimiento del contexto del agente.

Para que un agente participe en actividades orientadas al objetivo con otros agentes su motor de coordinación necesita un o más estrategias de comportamiento y alguna idea de que relaciones del agente tienen con cada uno de sus pares, de modo que la herramienta también provee:

- Una librería de protocolos de coordinación predefinidos.
- Una librería de relaciones organizacionales predefinida.

Juntos los componentes de la librería de componentes del agente activan la construcción de una aplicación genérica – agente independiente Zeus que puede ser customizada para aplicaciones específicas.

Lógicamente, cada agente Zeus está compuesto de tres niveles:

- o Un nivel definición: contiene las habilidades de razonamiento del agente, sus metas, recursos, habilidades, creencias, preferencias, etc.
- o Un nivel de organización: describe las relaciones de los agentes con otros agentes, que habilidades cree que los otros agentes tienen, y que relaciones de autoridad existen entre agentes.

- o Un nivel de coordinación: cada agente es modelado como una entidad social, están los protocolos que implementan las comunicaciones interagente.

La forma como este modelo lógico de agente actúa se muestra en la siguiente figura, usando las clases de la librería de componentes de agente de Zeus. El agente genérico incluye los siguientes componentes:

- Mailbox: mantiene la comunicación entre el agente y los agentes externos. Es una entidad compleja que consiste de muchos hilos de ejecución tales como el servidor, el que acepta y almacena los mensajes entrantes.
- Manejador de mensajes: Este procesa los mensajes entrantes del mailbox, distribuyéndolos a otros componentes dentro del agente.
- Motor de coordinación: Este toma decisiones referidas a las metas del agente (cómo deben proceder o cuándo abandonarlos) es responsable por la coordinación de todas las actividades del agente con otros agentes usando sus propias estrategias de coordinación. Esto no sólo activa tareas a ser contraídas y delegadas, pero también asegura que los agentes trabajen en conjunto a través de metas compartidas.
- Modelo de reconocimiento: Este describe las relaciones del agente con otros agentes en la sociedad y sus creencias acerca de las capacidades de estos agentes.
- Planeador y secuenciador: Este planea las tareas del agente basadas en las decisiones tomadas por el motor de coordinación y los recursos y la definición de tareas disponible al agente.
- Base de datos de recursos: mantiene una lista de recursos que son propiedad y disponibles a cada agente.

- Base de datos ontológica: Almacena la definición lógica de cada tipo de hecho – sus atributos legales, el rango de los valores legales de cada atributo, cualquier restricción entre los valores de los atributos y cada cualquier relación entre los atributos de los hechos, Los agentes deben usar la misma información ontológica si se quieren entender.
- Base de datos de tareas/planes: provee descripciones lógicas de planeamiento de operadores (o tareas) conocidas al agente. También almacena un conjunto de scripts que el agente usa para el comportamiento reactivo para eventos.
- Monitor de ejecución: Esto inicia, detiene y monitorea los sistemas externos que hayan sido secuenciadas para ejecución o término por el planeador y secuenciador. También informa al motor de coordinación del éxito y las condiciones de terminación excepcional de las tareas y su monitoreo.

Estos componentes genéricos actúan juntos para proveer la funcionalidad necesaria de los agentes. La comunicación es facilitada por el mailbox y la base de datos ontológica, el formador provee agentes con la habilidad de transmitir mensajes en un formato de reconocimiento universal, mientras que se activa cada agente para entender que se corresponde con la comunicación. La siguiente sección describe la metodología de construcción de agentes que ayuda a crear los agentes Zeus.

5.1.4 El software de construcción de agentes

La intención es proveer clases para implementar comunicación, razonamiento y cooperación, dejando a los desarrolladores concentrarse en la definición de los agentes

necesarios para su aplicación y proveer el código⁷ que implemente las habilidades específicas del dominio particular del agente.

Porque muchos de los aspectos del sistema convencional de agentes desarrollado son manejados por la herramienta Zeus, los desarrolladores necesitan un conocimiento de lo que se espera de ellos. Consecuentemente la herramienta Zeus fue desarrollada en paralelo con una metodología de desarrollo de agentes, esta guiará a través del proceso de creación.

5.1.5 La metodología de desarrollo de agentes Zeus

Una metodología es un sistema de métodos y principios que manejan la conducta y productos de algún proceso. Facilitan el pensamiento y actuación necesaria para ejecutar el proceso, pero también recomiendan métodos y técnicas que deberán ser empleadas para encaminar el proceso. La metodología de desarrollo de agentes Zeus está orientada al desarrollador, permitiéndole concentrarse en la solución del problema en cuestión.

Las seis etapas de la metodología de desarrollo Zeus se muestra en la Figura 5.2, las etapas se muestran en el orden que se procuran, donde los desarrolladores se mueven hacia abajo las actividades referidas hasta que concluyen.

⁷ El código fuente es un conjunto de líneas de código que conforman un bloque de texto que normalmente genera otro código mediante un compilador o intérprete para ser ejecutado por una computadora.

Normalmente se refiere a la programación de software. Un único programador o un equipo de ellos escriben el código fuente en el lenguaje de programación elegido. Posteriormente en un proceso de compilación el código fuente se traduce en código objeto.

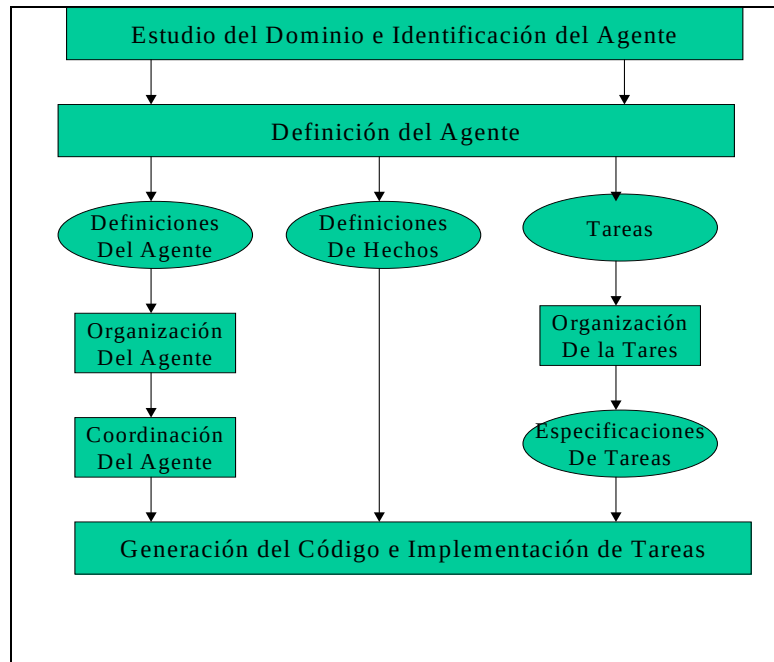


Figura 5.2: Etapas de la metodología Zeus.

La etapa inicial es el dominio de estudio y la identificación del agente, durante el cual el desarrollador analiza el dominio del problema para identificar los agentes potenciales.

Pero ¿Qué es un agente?: Esto depende en gran parte de la granularidad en la que el dominio de modelado, pero los mejores candidatos serán aquellas entidades que son autónomas por lo tanto capaces principalmente de funcionar independientemente de la interacción humana, y aquellos que son sociales, por lo tanto capaces de interactuar inteligentemente y constructivamente con otros agentes o humanos.

Una vez que el candidato a agentes son identificados la fase de definición puede comenzar. Esto involucra al desarrollador identificar los atributos significativos de cada agente. Una metáfora para un agente utilizado en esta etapa de la metodología es de los administradores de fábrica, quienes controlan muchas líneas de producción. Cada agente inicia su vida con un conjunto limitado de recursos (llamados hechos) y la habilidad para ejecutar un número diferente de tareas simultáneamente. La producción de que un item dure por un periodo de

tiempo finito y consuma alguno de los recursos de un agente. Esta etapa involucra la identificación del conjunto inicial de recursos propios de un agente (y la definición de cada tipo), tan bien como las tareas que el agente ejecuta.

La fase de definición continúa hasta que todos los agentes hayan sido considerados y sus recursos iniciales y tareas hayan sido identificadas. La siguiente etapa es la organización del agente donde el desarrollador identifica al agente. Para reconocer a un agente A con otro agente B, los recursos que el agente A cree del agente B pueden producir son especificadas, también con las relaciones que el agente A cree que comparte con el agente B. Los agentes que son definidos como superiores a otros subordinados pueden delegar tareas a estos. Los agentes pueden también pertenecer a la misma comunidad estática y ser coempleados quienes prefieren interactuar con otros.

Como la etapa de organización requiere información acerca de las habilidades de los agentes, puede ser ejecutada en conjunto con la fase de definición de tareas. Durante esta etapa, las tareas que fueron identificadas durante la fase de definición del agente son definidas en términos de sus costos, duración, precondiciones, productos y restricciones. Las tareas pueden ser primitivas o sumarias.

Cuando todas las tareas son especificadas, la fase de coordinación del agente puede comenzar. Esto involucra identificar los protocolos apropiados de coordinación que cada agente requerirá para la interacción social con otros agentes cuando ejecute sus labores rutinarias.

Como la metodología de diseño de agentes es soportada por el software generador de agentes, la etapa final de esta metodología es la generación del código y la implementación de las tareas.

5.1.6 El software generador de agentes Zeus

Es una suite de editores integrados que soportan la metodología de desarrollo de agentes Zeus. Para facilitar el uso los editores han sido diseñados para que el usuario pueda interactivamente crear agentes visualizando específicamente sus atributos. Estos son:

- **Ontología:** para definir los items de ontología en un dominio (los hechos reales, los atributos de estos hechos, los valores legales de cada atributo y cualquier restricción entre los hechos de los atributos)
- **Hechos y/o variables:** para describir específicamente las instancias de hechos y variables, usando las plantillas ya creadas usadas en el editor de ontología.
- **Descripción de la tarea:** Para ingresar los atributos de las tareas y también activar la composición de tareas sumarias, las cuales consisten de varias subtareas ordenadas de algún modo.
- **Restricciones:** invocadas por el editor de tareas no sólo para definir las restricciones entre las precondiciones y efectos de las tareas, pero puede también restringir los efectos de una tarea precedente y las precondiciones de una tarea resultante en una tarea sumaria.
- **Organización:** para definir las relaciones organizacionales entre los agentes.
- **Definición de agentes:** para describir lógicamente a los agentes lo que involucra nombrar a cada tarea del agente y especificar sus recursos iniciales.
- **Coordinación:** para seleccionar el conjunto de protocolos de coordinación con los cuales el agente será equipado.

Para generar el código para una aplicación específica, la herramienta de generación hereda código de la librería de componentes del agente y la integra con la data de los editores. El programa resultante Java es luego compilado y ejecutado normalmente.

Capítulo 6: Identificación de indicadores

6.1 Indicadores contables y financieros

Estos indicadores son los que normalmente se utilizan para tomar decisiones, básicamente se obtienen de la contabilidad de la empresa (Estos indicadores no forman parte del presente trabajo).

6.2 Indicadores estratégicos o de gestión

Para el presente trabajo se utilizará la definición de indicadores de gestión no funcionales, es decir aquellos indicadores que no se orientan a medir lo que la empresa produce, sino a medir los sistemas que dan apoyo a los procesos de la empresa.

6.3 Dimensiones de los sistemas críticos

En el Capítulo II, Sommerville enuncia [Sommerville 02]: “Existen cuatro dimensiones principales de la confiabilidad (como se muestra en la Figura 3.1)”.

1. Disponibilidad: Informalmente la disponibilidad de un sistema es la probabilidad de que esté activo y en funcionamiento y le sea posible suministrar servicios útiles en cualquier momento.
2. Fiabilidad: Informalmente la fiabilidad de un sistema es la probabilidad de que en un período, el sistema suministre los servicios correctamente tal como lo espera el usuario.
3. Seguridad: Informalmente, la seguridad en un sistema es una valoración de que tan probable es que el sistema provoque un daño a la gente o a su entorno.

4. Protección: Informalmente la protección de un sistema es una valoración de que tan probable es que el sistema resista a una intrusión accidental o provocada.

Para el presente trabajo consideramos que las dimensiones de seguridad y protección se mantienen constantes. Por lo tanto se considera que los indicadores o métricas del presente trabajo se realizarán a las dimensiones de disponibilidad, fiabilidad y éstas se consolidarán en la dimensión resultado denominada confiabilidad.

6.3.1 Indicadores de disponibilidad

Identificación de eventos, alertas, mensajes por servidor debido a caídas del sistema, pérdida de comunicación, desconexión de recursos, calculándose el tiempo no disponible, para el presente trabajo se tomará la información de la base de datos y sistema operativo.

6.3.1.1 Indicadores de base de datos

Dentro de la base de datos se pueden medir y obtener los siguientes datos: *tablespaces, extends, users, freespace, locks, shutdown, system, cursors, license, procedures, sessions, dumps, SGA, jobs, logs*, etc.

6.3.1.2 Indicadores de sistema operativo

Dentro del sistema operativo se pueden medir y obtener los siguientes: memoria, discos duros, CPU, usuarios, *logs*, etc.

6.3.2 Indicadores de fiabilidad

6.3.2.1 Indicadores del sistema ERP

Existen muchos indicadores a definir teniendo en cuenta los módulos existentes en el sistema ERP: módulo de finanzas, módulo de distribución, módulo de manufactura, módulo de proyectos, módulo de servicios, módulo de transportes, módulo de recursos humanos (no *ERP*), módulo de desarrollo y mantenimiento (*tools*), etc.

Nota: estos indicadores se determinarán hasta el nivel del usuario cuando sea necesario.

6.3.3 Indicador de confiabilidad

Este indicador nos mostrará el grado de confiabilidad de acuerdo a los indicadores de disponibilidad y fiabilidad definidos para la base de datos, sistema operativo y sistema ERP.

6.4 Indicadores

Una representación formal del modelo de confiabilidad considerando la fiabilidad y disponibilidad se esquematiza mediante una estructura de árbol, tal como es ilustrado por la Figura 6.1, en la que el nodo de nivel superior representa a la Confiabilidad (CODA) y los nodos hijos a cada uno de los componentes a medir en este caso el Sistema ERP (CODA_ERP), la base de datos (CODA_BD) y al sistema operativo (CODA_SO), a su vez cada uno tendrá sus respectivos hijos denominados aspectos como son la confiabilidad (CODA_ERP01), la disponibilidad (CODA_ERP02) y la fiabilidad (CODA_ERP03), en los demás casos se repite estas definiciones como se muestra en la Tabla 6.1.

Nodos	Descripción
CODA	Sistema CODA
CODA_BD	Base de Datos
CODA_SO	Sistema Operativo
CODA_ERP	Sistema ERP
CODA_BD01	BD Confiabilidad
CODA_BD02	BD Disponibilidad
CODA_BD03	BD Fiabilidad
CODA_SO01	SO Confiabilidad
CODA_SO02	SO Disponibilidad
CODA_SO03	SO Fiabilidad
CODA_ERP01	ERP Confiabilidad
CODA_ERP02	ERP Disponibilidad
CODA_ERP03	ERP Fiabilidad

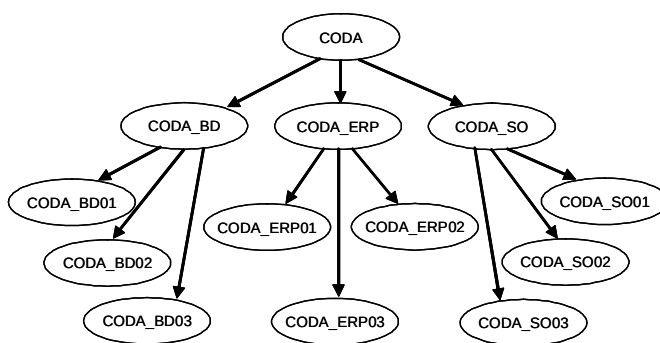


Tabla 6.1: Representación de los nodos CODA. Figura 6.1: Arquitectura del sistema CODA.

6.5 Indicadores CODA

En la Tabla 6.2 se muestra los indicadores definidos para los nodos del sistema CODA. Estos se denominan indicadores, los que se definen para cada uno de los componentes (ERP, SO y BD). Los valores que tendrán estos indicadores se muestran en la Tabla 6.3 y son el valor real que se genera a partir de los datos extraídos por el sistema, el valor normal o promedio de los valores históricos al igual que el valor mínimo y máximo.

Indicador	Descripción
BD001	BD Confiabilidad
BD002	BD Disponibilidad
BD003	BD Fiabilidad
SO001	SO Confiabilidad
SO002	SO Disponibilidad
SO003	SO Fiabilidad
ERP001	ERP Confiabilidad
ERP002	ERP Disponibilidad
ERP003	ERP Fiabilidad
BD002002	Indicador BD 002
BD002003	Indicador BD 003
BD002004	Indicador BD 004
SO002001	Indicador SO 001
SO002002	Indicador SO 002
SO002003	Indicador SO 003
ERP001001	Indicador ERP 001
ERP001002	Indicador ERP 002
ERP001003	Indicador ERP 003
ERP001004	Indicador ERP 004
ERP001005	Indicador ERP 005
BD002001	Indicador BD 001

Tabla 6.2: Indicadores CODA

Fecha	Indicador	Año	Periodo	Real	Normal	Minimo	Maximo
01/01/2006	BD0020001	2006	1	10	9	8	10
01/01/2006	BD0020002	2006	1	9	9	7	10
01/01/2006	BD0020003	2006	1	10	9	8	9
01/01/2006	BD0020004	2006	1	8	9	8	10
01/01/2006	SO0020001	2006	1	9	9	8	10
01/01/2006	SO0020002	2006	1	9	9	8	9
01/01/2006	SO0020003	2006	1	9	9	8	10
01/01/2006	ERP0010001	2006	1	7	8	5	9
01/01/2006	ERP0010002	2006	1	8	8	7	10
01/01/2006	ERP001	2006	1	93	90	78	98
01/01/2006	SO001	2006	1	90	90	90	97
01/01/2006	BD001	2006	1	65	80	80	95
01/01/2006	CODA	2006	1	83	87	73	96

Tabla 6.3:: Indicadores del sistema CODA.

Indicador de confiabilidad (CODA)

Este indicador nos mostrará el grado de confiabilidad de acuerdo a los indicadores de disponibilidad y fiabilidad definidos para la base de datos, sistema operativo y sistema ERP. El indicador de CODA es la media de los indicadores de contabilidad de los tres sistemas (ERP, BD, SO), dado por:

$$CODA = \frac{1}{N} \sum_{i=1, \dots, 999} (ERP_i + BD_i + SO_i)$$

donde N es total de indicadores diferentes de cero de los tres sistemas del mismo nivel.

Por ejemplo, si $N = 3$ la expresión sería: $CODA = (ERP001 + SO001 + BD001) / 3$, con $i = 001$.

Por lo que se afirma que este indicador es el resultado del procesamiento de los demás indicadores de Nivel 1 que a su vez son resultado de los del Nivel 2, además, cada indicador de tiene un valor real, normal o promedio, mínimo y máximo, se define que la media de los indicadores mínimos será el indicador mínimo de CODA; de la misma forma el indicador máximo de CODA es la media de los indicadores máximos de CODA (tal como se muestra en la Tabla 6.3).

Generalizando, para nivel 2, o indicador ERP_i o BD_i o SO_i puede ser como SIS_i calculado como:

$$SIS_i = \frac{10}{N} \sum_{j=1, \dots, 999} SIS_{ij}$$

Aquí, como en el nivel superior, el N es total de indicadores a procesar diferente de cero; los valores de los indicadores $1 \leq SIS_{ij} \leq 10$, representan los indicadores respectivos del sistema SIS_i en nivel 2.

Indicadores de disponibilidad

Identificación de eventos, alertas, mensajes por servidor debido a caídas del sistema, pérdida de comunicación, desconexión de recursos, calculándose el tiempo no disponible, para el presente trabajo se tomará la información de la base de datos y sistema operativo. Dentro de la base de datos se pueden medir y obtener los siguientes datos: *tablespaces, extends, users, freespace, locks, shutdown, system, cursors, license, procedures, sessions, dumps, SGA, jobs, logs*, etc. Dentro del sistema operativo se pueden medir y obtener los siguientes datos: memoria, discos duros, CPU, usuarios, logs, etc.

Por ejemplo, el cálculo del indicador BD001, para N = 4, sería:

$$BD001 = (BD0020001 + BD0020002 + BD0020003 + BD0020004) / 4 * 10$$

Mientras del indicador SO001, para N=3, sería:

$$SO001 = (SO0020001 + SO0020002 + SO0020003) / 3 * 10$$

Indicadores de fiabilidad

Existen muchos indicadores de fiabilidad del sistema ERP a definir teniendo en cuenta los módulos existentes en el sistema ERP: módulo de finanzas, módulo de distribución, módulo de manufactura, módulo de proyectos, módulo de servicios, módulo de transportes, módulo de recursos humanos (no *ERP*), módulo de desarrollo y mantenimiento (*tools*), etc.

Por ejemplo, el cálculo del indicador ERP001 para N = 2, sería:

$$ERP001 = ((ERP0010001 + ERP0010002) / 2) * 10$$

Cuando los valores de los indicadores son 0, no son considerados en la fórmula, la multiplicación por 10 permite tener un valor porcentual (%).

Capítulo 7: Modelo de Agentes

7.1 Modelo general de agentes

En la Figura 7.1 se muestra el modelo general de agentes que se desarrollará como parte de este trabajo, donde se identifica los siguientes agentes: agente ERP, agente base de datos, agente sistema operativo y agente recolector. Dentro del modelamiento de los agentes se distinguirá dos secciones, una relacionada con el meta-modelo de Organización y el modelamiento de cada uno de los agentes.

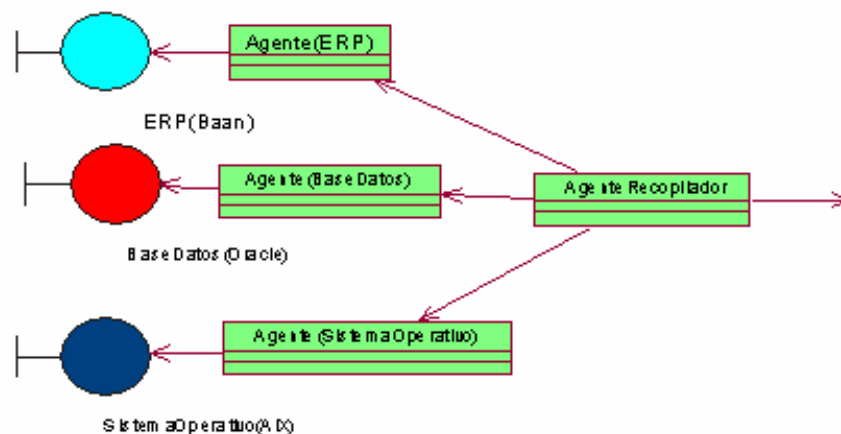


Figura 7.1 Modelo general de agentes.

7.2 Meta-modelo de organización

Incluye los grupos y agentes los que componen la organización y las aplicaciones, también muestra los flujos de la organización de modo que satisfaga sus objetivos.

Este sistema está compuesto por sólo una organización: Sistema Coda, este tiene objetivos globales, esta organización está compuesta a su vez por dos grupos tal como se muestra en la Figura 7.2.

Administración: principalmente compuesto por la plataforma de desarrollo a la que se le agregará una Base de Datos local para almacenar la información recuperada de los servidores del Sistema ERP. Comunidad: a los que pertenecen los agentes del Sistema CODA. Sus actividades están organizadas en muchos flujos. El grupo de administración está relacionado con las tareas globales del sistema; eso es, aquellas compartidas entre los diferentes tipos y recursos de información.

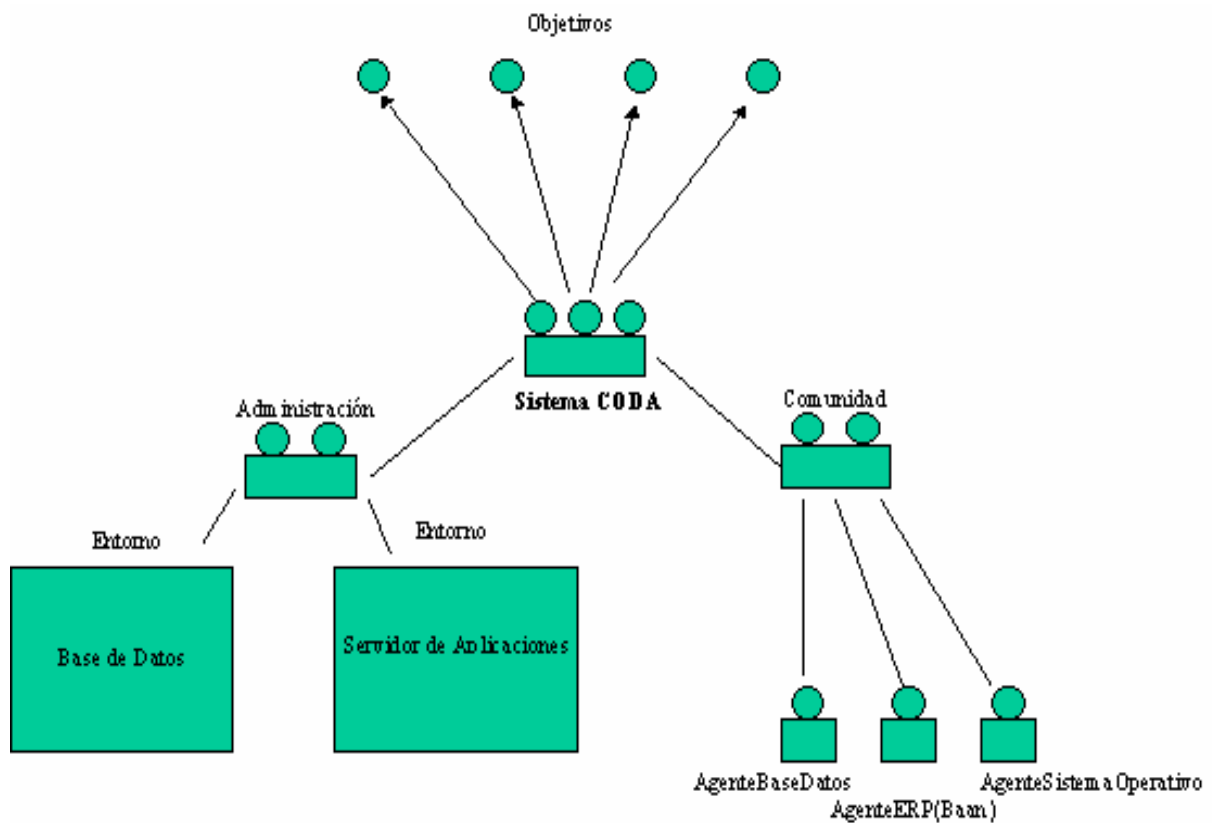


Figura 7.2 Metamodelo de organización.

La descripción al detalle de lo mostrado en la Figura 4.2 se puede observar en la siguiente Tabla 7.1.

Entidad	Entidad del meta-modelo	Documentación
Agente Base Datos	Agente	Representa el agente que recuperará la información de la Base de Datos donde se guarda la información del Sistema ERP
Agente ERP (Baan)	Agente	Representa el agente que recuperará la información de la aplicación del sistema ERP
Agente Sistema Operativo	Agente	Representa el agente que recuperará la información del sistema operativo donde se ha instalado el sistema ERP
Sistema CODA	Grupo	Es el grupo general dentro del que se encuentran los agentes
Administración	Grupo	Principalmente compuesto por la plataforma de construcción de agentes
Comunidad	Grupo	Compuesto por los agentes base de datos, ERP y sistema operativo

Tabla 7.1: Instancias de Entidades de Meta-Modelo de Organización.

7.3 Modelamiento de agentes

7.3.1 Modelamiento del agente de base de datos

7.3.1.1 Casos de Uso para el agente de base de datos

Los casos de uso describen la funcionalidad que el agente de base de datos provee. Algunos de ellos están relacionados a la interacción en la cual son especificados en mayor detalle con los conceptos paradigmáticos de agentes. Los casos de uso ilustran la funcionalidad relevante es la gestión del Agente de Base de Datos, refleja las acciones que toma cuando se desea ver si hay información relevante a recuperar.

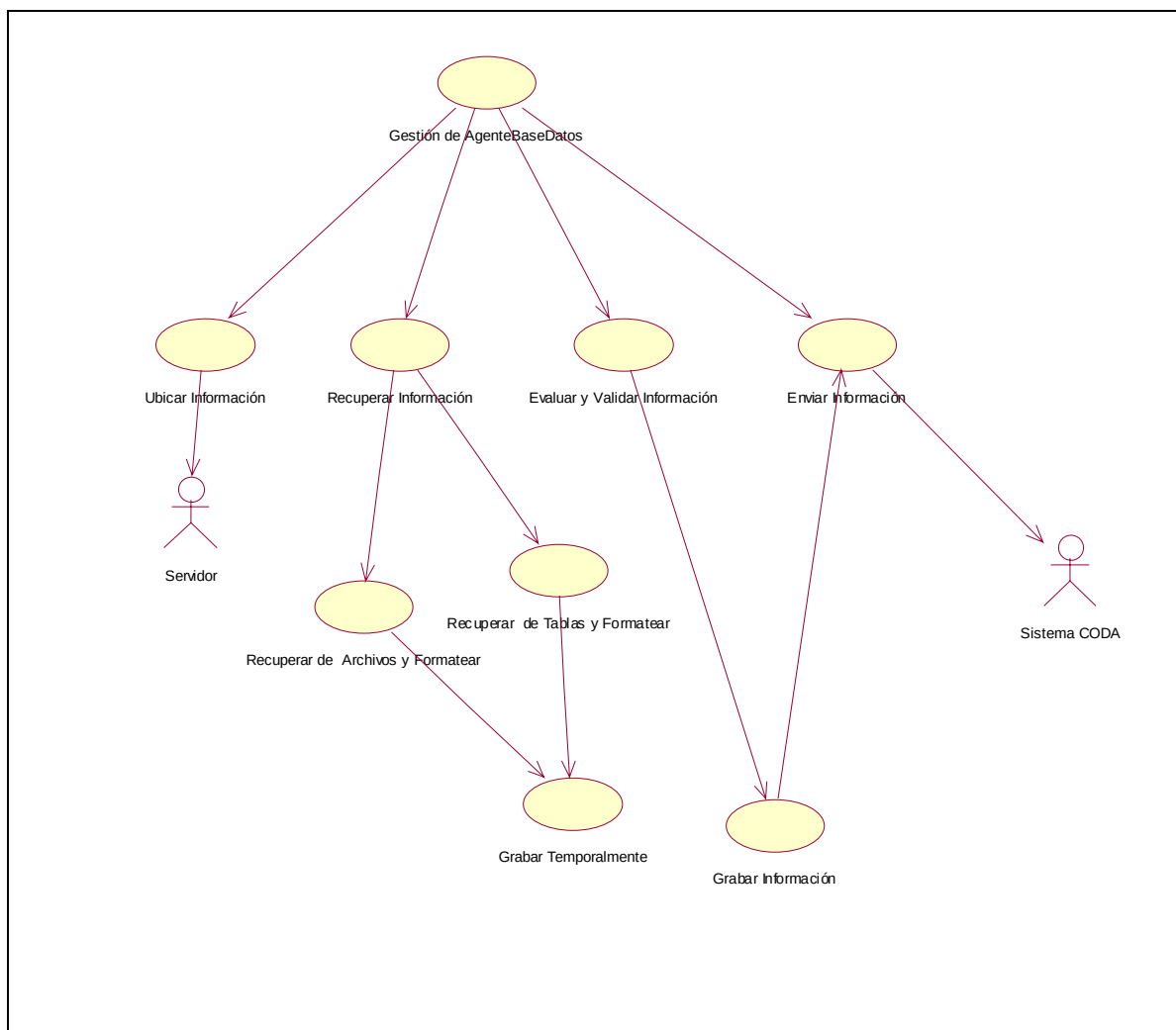


Figura 7.3 Casos de uso del agente de base de datos.

En la Figura 7.3 se muestra los casos de uso del Agente de Base de Datos, los que a continuación en la Tabla 7.2 son descritas en detalle.

Entidad	Entidad del meta-modelo	Documentación
Gestión de base de datos	CasoUso	Involucra los casos de uso que permitirán recuperar información relevante de la base de datos y enviarla al sistema CODA
Ubicar información	CasoUso	Mediante este caso de uso el agente identificará el lugar donde se encuentra la información a recuperar
Recuperar información	CasoUso	Mediante este caso de uso el agente recuperará la información
Recuperar de archivos y formatear	CasoUso	Cuando se trate de archivos se realizará la recuperación mediante un procedimiento y luego se formateará para que pueda ser luego evaluada.
Recuperar de tablas y formatear	CasoUso	Cuando se trate de tablas se realizará la recuperación mediante scripts y luego se formateará para que pueda ser luego evaluada.

Grabar temporalmente	CasoUso	Mediante este caso de uso se consolidará la información tanto de archivos o tablas
Evaluar y validar información	CasoUso	Toda la información no podrá ser informada a menos que no sea relevante o prioritaria, para ello se utilizará tablas de validación
Grabar información	CasoUso	Al final se debe proceder a grabar la información validada
Enviar información	CasoUso	Esta información será enviada a la base de datos del sistema CODA
Copiar files	Interacción	Describe la interacción de copiar del servidor origen los archivos que se utilizarán para recuperar la información de un folder específico

Tabla 7.2 Instancias de entidades de meta-modelo agente base de datos.

7.3.1.2 Agente de base de datos

El agente de base de datos (AgenteBaseDatos) es el que gestiona la ubicación, recuperación, validación y envío de la información que reside en los archivos de gestión de la base de datos, tablas de la base de datos o archivos de Baan relacionados a la base de datos. Este modelo muestra su procesador de estado mental, el administrador, las metas que persigue, los roles, las tareas que lleva a cabo. El AgenteBaseDatos persigue las siguientes metas: estos son ubicar, recuperar, validar y enviar la información, estos lo hacen responsable de administrar la información que genera la base de datos donde reside el sistema ERP Baan. Tiene muchos roles: procesador la ubicacion de archivos y/o tablas, organizar la información, obtener la información validada, grabar la información validada, cada uno de ellos se desagregará en rutinas especializadas para realizar las funciones asignadas. La tarea más importante que realiza es la de informar al Sistema CODA que ha cumplido con encontrar uno o más eventos.

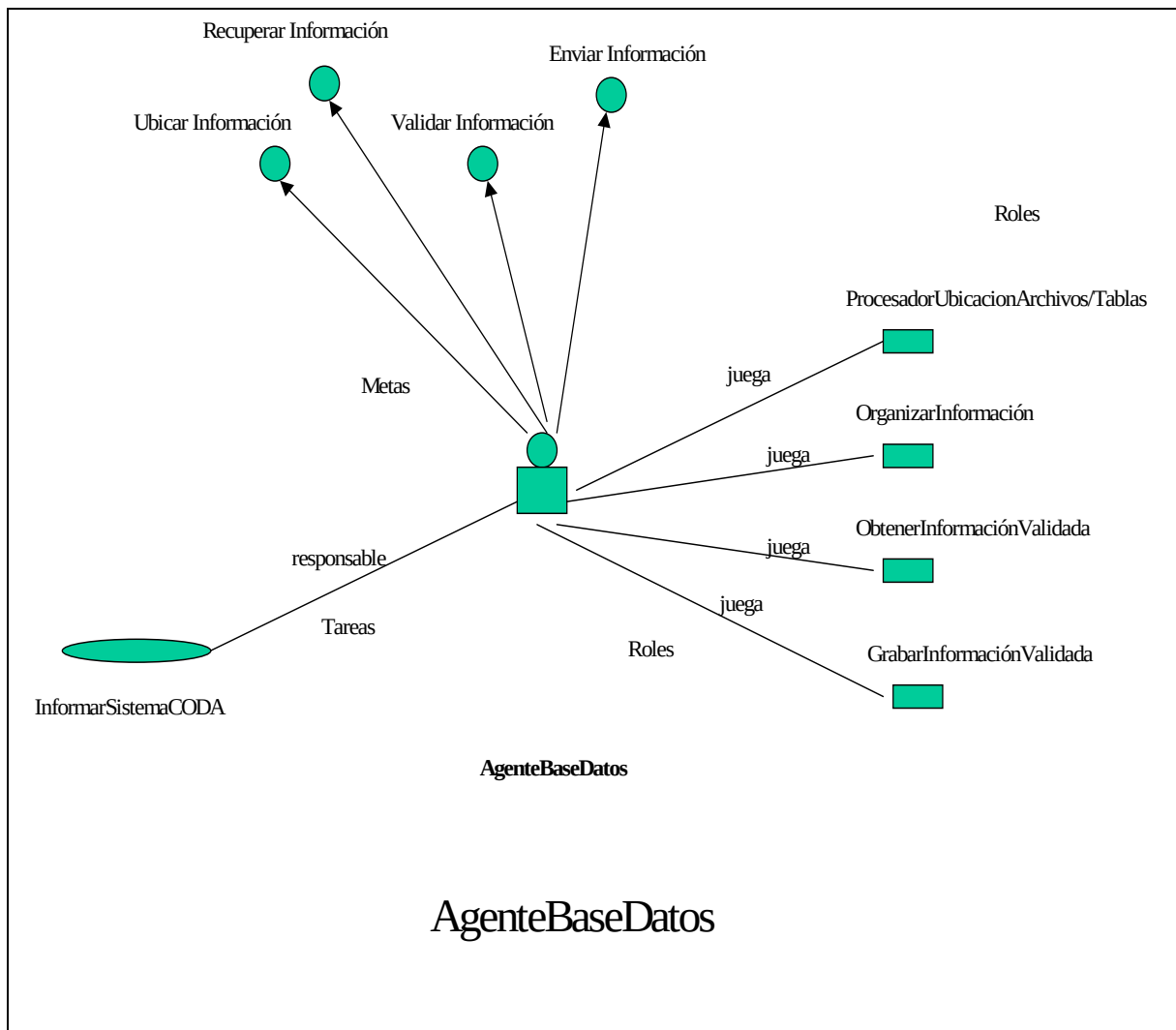


Figura 7.4 Modelo de agente base de datos.

La descripción detallada de la Figura 7.4 se muestra en la siguiente Tabla 7.3.

Entidad	Entidad del meta-modelo	Documentación
AgenteBaseDatos	Agente	Representa el agente que administra la información relevante de la Base de Datos del ERP a monitorear.
Ubicar información	Tarea	Se debe ubicar el lugar donde residen los archivos a ser recuperados o las tablas desde donde se recuperará la información
Recuperar información	Tarea	Una vez ubicada se debe proceder a recuperar la información
Validar información	Tarea	De acuerdo al tipo de indicador se deberá validar si el grado de importancia de la información
Enviar información	Tarea	Se debe enviar la información relevante a una base de datos del sistema CODA
ProcesadorUbicacionArchivos/Tablas	Rol	Busca la información ya sea en los archivos o tablas de la Base de Datos

OrganizarInformación	Rol	Ordena, clasifica para luego ser procesada y validada
ObtenerInformaciónValidada	Rol	De acuerdo a criterios establecidos se procede a solamente obtener información importante
GrabarInformaciónValidada	Rol	Una vez ubicada se procede a formatearla para ser enviada y grabada en el Sistema CODA
Informar al Sistema CODA	Tarea	Una vez obtenida la información y grabada en el sistema CODA se procede a informar de que la tarea ha sido terminada

Tabla 7.3: Instancias de Entidades de Meta-Modelo.

7.3.2 Modelamiento del agente sistema operativo

7.3.2.1 Casos de uso para el agente de sistema operativo

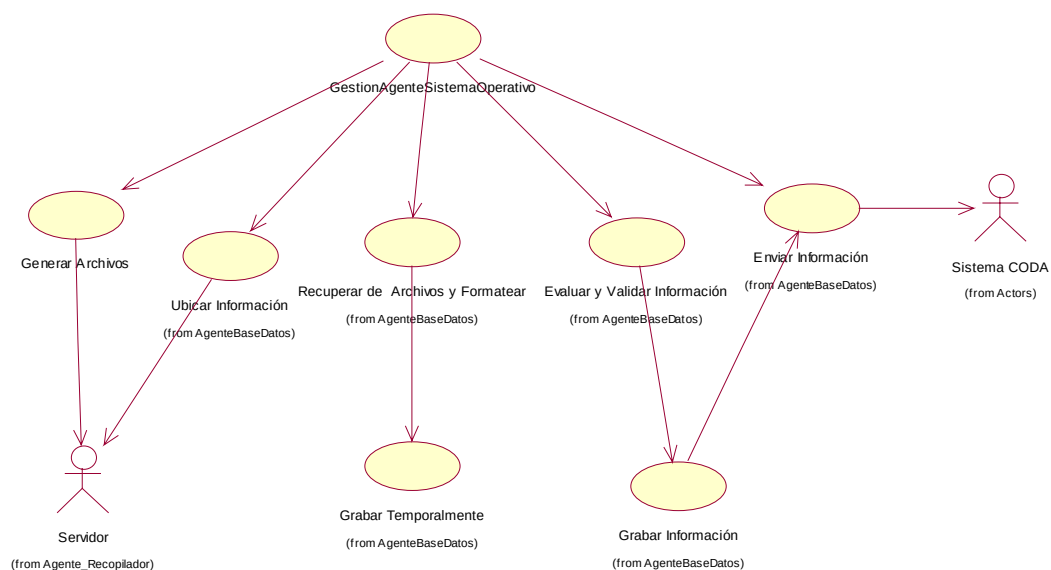


Figura 7.5 Casos de uso agente sistema operativo.

Los casos de uso describen la funcionalidad que el agente de sistema operativo provee. Algunos de ellos están relacionados a la interacción en la cual son especificados en mayor detalle con los conceptos paradigmáticos de agentes. Los casos de uso ilustran la funcionalidad relevante en la gestión del agente de sistema operativo, refleja las acciones que toma cuando se desea ver si hay información relevante a recuperar. La Figura 7.5

muestra cada uno de los casos de uso que el agente del sistema operativo gestiona, el resumen de estos casos de uso son descritos en la Tabla 7.4.

Entidad	Entidad del meta-modelo	Documentación
Gestión de sistema operativo	CasoUso	Involucra los casos de uso que permitirán recuperar información relevante del sistema operativo y enviarla al sistema CODA
Ubicar información	CasoUso	Mediante este caso de uso el agente identificará el lugar donde se encuentra la información a recuperar
Recuperar información	CasoUso	Mediante este caso de uso el agente recuperará la información
Recuperar de archivos y formatear	CasoUso	Cuando se trate de archivos se realizará la recuperación mediante un procedimiento y luego se formateará para que pueda ser luego evaluada.
Grabar temporalmente	CasoUso	Mediante este caso de uso se consolidará la información tanto de archivos o tablas
Evaluar y validar información	CasoUso	Toda la información no podrá ser informada a menos que no sea relevante o prioritaria, para ello se utilizará tablas de validación
Grabar información	CasoUso	Al final se debe proceder a grabar la información validada
Enviar información	CasoUso	Esta información será enviada a la base de datos del sistema CODA
Copiar files	Interacción	Describe la interacción de copiar del servidor origen los archivos que se utilizarán para recuperar la información de un folder específico

Tabla 7.4 Instancias de entidades de meta-modelo agente sistema operativo.

7.3.2.2 Agente del sistema operativo

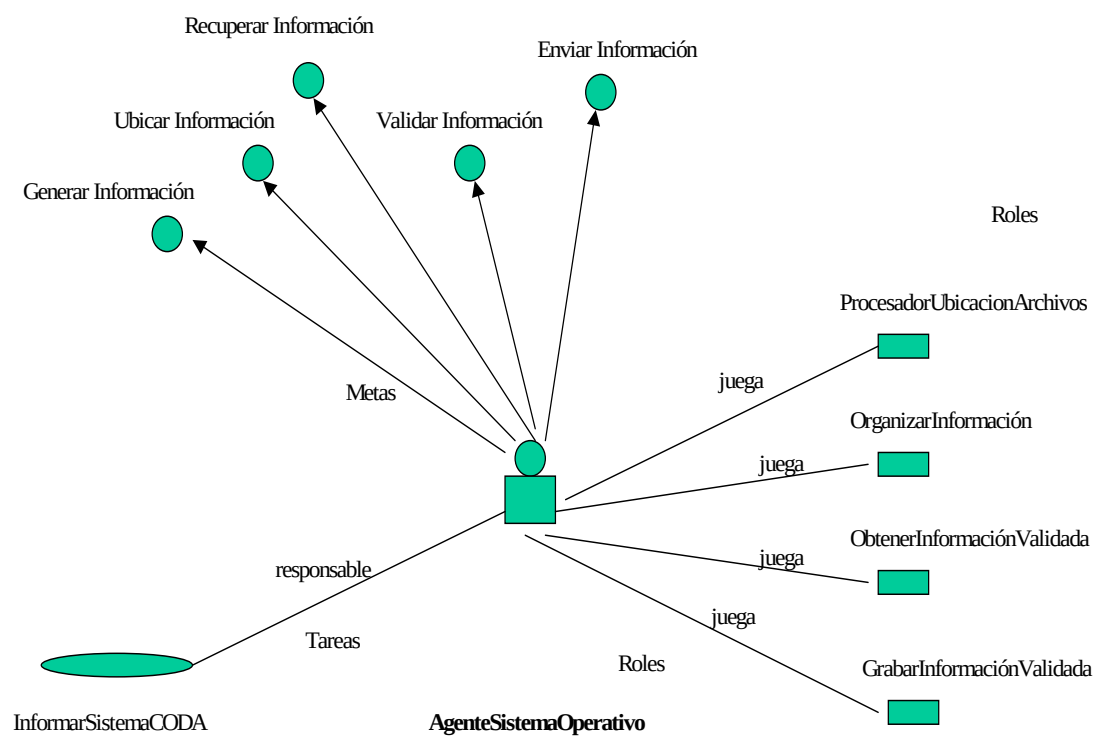
El agente del sistema operativo (AgenteSistemaOperativo) es el que gestiona la ubicación, recuperación, validación y envío de la información que reside en los archivos del sistema operativo o archivos de Baan relacionados con el sistema operativo.

Este modelo muestra su procesador de estado mental, el administrador, las metas que persigue, los roles, las tareas que lleva a cabo.

El AgenteSistemaOperativo persigue las siguientes metas: Estos son ubicar, recuperar, validar y enviar la información, estos lo hacen responsable de administrar la información que genera el sistema operativo donde reside el sistema ERP Baan.

Tiene muchos roles: ProcesadorUbicacionArchivos, OrganizarInformación, ObtenerInformaciónValidada, GrabarInformaciónValidada, cada uno de ellos se desagregará en rutinas especializadas para realizar las funciones asignadas.

En la Figura 7.6 se muestra gráficamente lo descrito anteriormente, el detalle se muestra en la Tabla 7.5. La tarea más importante que realiza es la de informar al sistema CODA que ha cumplido con encontrar uno o más eventos.



AgenteSistemaOperativo

Figura 7.6: Modelo de agente sistema operativo.

Entidad	Entidad del meta-modelo	Documentación
AgenteSistemaOperativo	Agente	Representa el agente que administra la información relevante del sistema operativo a monitorear.
Ubicar información	Tarea	Se debe ubicar el lugar donde residen los archivos a ser recuperados o las tablas desde donde se recuperará la información
Recuperar información	Tarea	Una vez ubicada se debe proceder a recuperar la información

Validar información	Tarea	De acuerdo al tipo de indicador se deberá validar si el grado de importancia de la información
Enviar información	Tarea	Se debe enviar la información relevante a una base de datos del sistema CODA
ProcesadorUbicacionArchivos	Rol	Busca la información en los archivos
OrganizarInformación	Rol	Ordena, clasifica para luego ser procesada y validada
ObtenerInformaciónValidada	Rol	De acuerdo a criterios establecidos se procede a solamente obtener información importante
GrabarInformaciónValidada	Rol	Una vez ubicada se procede a formatearla para ser enviada y grabada en el Sistema CODA
Informar al Sistema CODA	Tarea	Una vez obtenida la información y grabada en el sistema CODA se procede a informar de que la tarea ha sido terminada

Tabla 7.5: Instancias de entidades de meta-modelo.

7.3.3 Modelamiento de agente ERP

7.3.3.1 Casos de uso para agente del sistema ERP

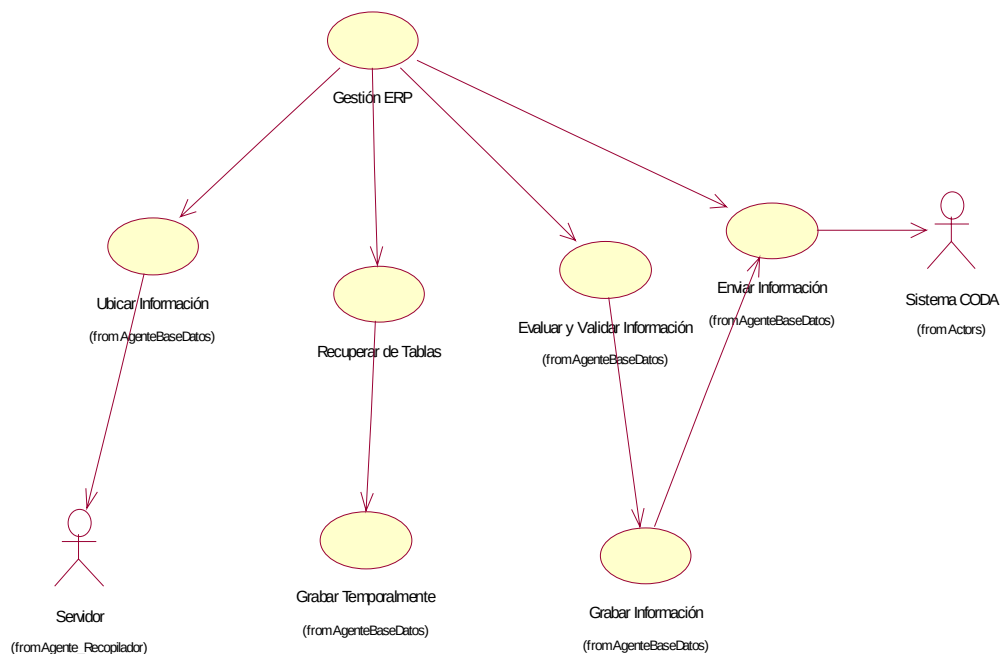


Figura 7.7 Casos de uso del agente ERP.

Los casos de uso describen la funcionalidad que el agente ERP provee. Algunos de ellos están relacionados a la interacción en la cual son especificados en mayor detalle con los conceptos paradigmáticos de agentes. Los casos de uso ilustran la funcionalidad relevante en la gestión del agente ERP, refleja las acciones que toma cuando se desea ver si hay información relevante a recuperar. En la Figura 7.7 se muestra cada uno de los casos de uso del agente ERP y en la Tabla 7.6 se muestra el resumen.

Entidad	Entidad del meta-modelo	Documentación
Gestión de ERP	CasoUso	Involucra los casos de uso que permitirán recuperar información relevante del ERP y enviarla al sistema CODA
Ubicar información	CasoUso	Mediante este caso de uso el agente identificará el lugar donde se encuentra la información a recuperar
Recuperar información	CasoUso	Mediante este caso de uso el agente recuperará la información
Recuperar de tablas y formatear	CasoUso	Se realizará la recuperación mediante un procedimiento almacenado para que pueda ser luego evaluada.
Grabar temporalmente	CasoUso	Mediante este caso de uso se consolidará la información
Evaluar y validar información	CasoUso	Toda la información no podrá ser informada a menos que no sea relevante o prioritaria, para ello se utilizará tablas de validación
Grabar información	CasoUso	Al final se debe proceder a grabar la información validada
Enviar información	CasoUso	Esta información será enviada a la base de datos del sistema CODA
Copiar files	Interaccion	Describe la interacción de copiar del servidor origen los archivos que se utilizarán para recuperar la información de un folder específico

Tabla 7.6: Instancias de entidades de meta-modelo agente ERP.

7.3.3.2 Agente ERP

El agente ERP (AgenteERP) es el que gestiona la ubicación, recuperación, validación y envío de la información que reside en los archivos del sistema operativo o archivos de Baan relacionados con sus tablas. Este modelo muestra su procesador de estado mental, el administrador, las metas que persigue, los roles, las tareas que lleva a cabo. El agente ERP

persigue las siguientes metas: Estos son ubicar, recuperar, validar y enviar la información, estos lo hacen responsable de administrar la información que generan las tablas del sistema ERP Baan. Tiene muchos roles: ProcesadorUbicacionTablas, OrganizarInformación, ObtenerInformaciónValidada, GrabarInformaciónValidada, cada uno de ellos se desagregará en rutinas especializadas para realizar las funciones asignadas. En la Figura 7.8 se muestra gráficamente lo descrito anteriormente y un resumen de cada actividad se muestra en la Tabla 7.7. La tarea más importante que realiza es la de informar al Sistema CODA que ha cumplido con encontrar uno o más eventos.

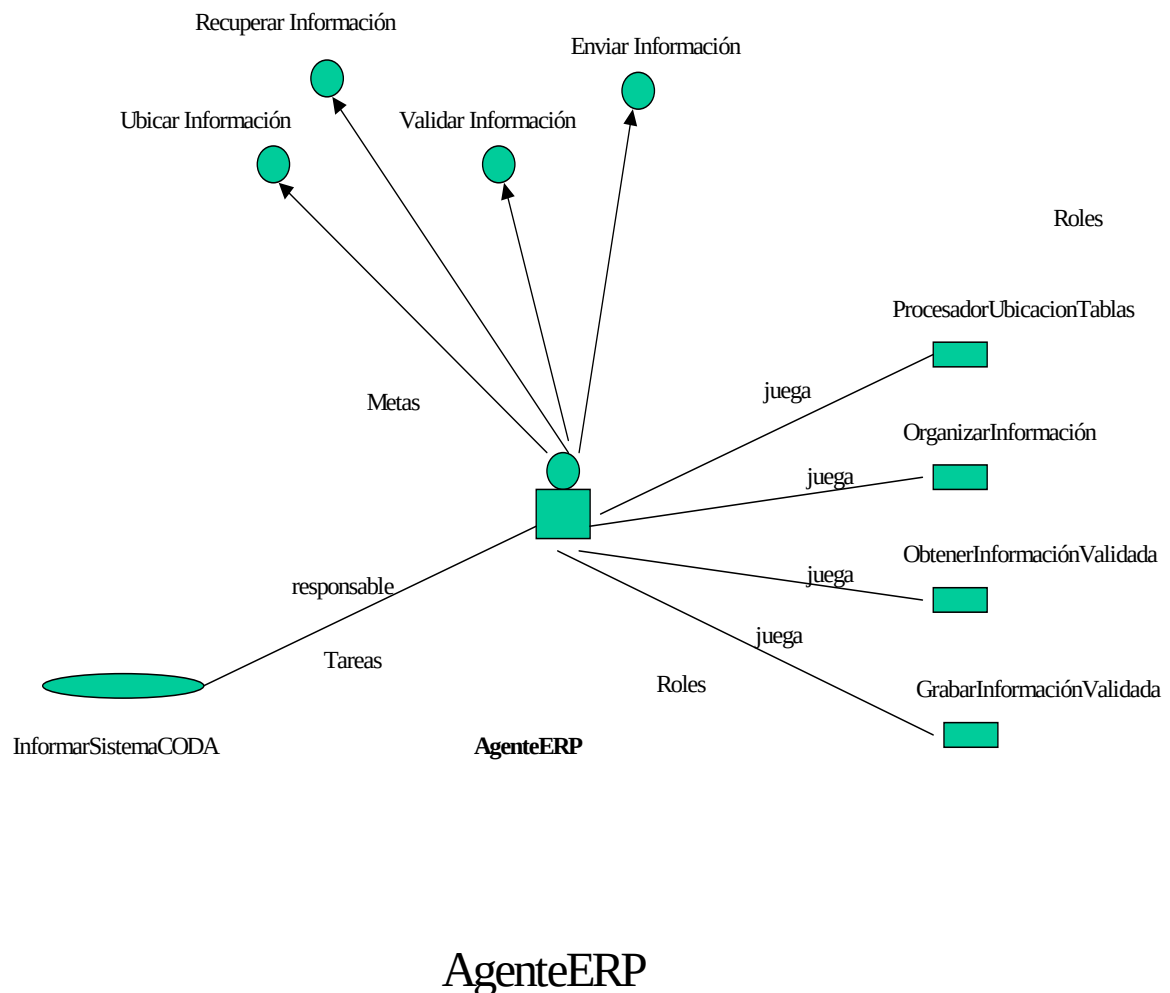


Figura 7.8 Modelo de agente ERP.

Entidad	Entidad del meta-modelo	Documentación
AgenteERP	Agente	Representa el agente que administra la información relevante del ERP a monitorear.
Ubicar información	Tarea	Se debe ubicar las tablas a ser recuperada desde donde se recuperará la información
Recuperar información	Tarea	Una vez ubicada se debe proceder a recuperar la información
Validar información	Tarea	De acuerdo al tipo de indicador se deberá validar si el grado de importancia de la información
Enviar información	Tarea	Se debe enviar la información relevante a una base de datos del sistema CODA
ProcesadorUbicacionArchivos	Rol	Busca la información en los archivos
OrganizarInformación	Rol	Ordena, clasifica para luego ser procesada y validada
ObtenerInformaciónValidada	Rol	De acuerdo a criterios establecidos se procede a solamente obtener información importante
GrabarInformaciónValidada	Rol	Una vez ubicada se procede a formatearla para ser enviada y grabada en el Sistema CODA
Informar al Sistema CODA	Tarea	Una vez obtenida la información y grabada en el sistema CODA se procede a informar de que la tarea ha sido terminada

Tabla 7.7 Instancias de entidades de meta-modelo.

Capítulo 8: El Sistema CODA

En este capítulo se mostrarán los principales requerimientos de software así como la lógica del negocio que sea necesario para la realización de los módulos del sistema CODA.

8.1 Propósito

El propósito es tener registrada las funcionalidades de la lógica del proceso del sistema CODA, de tal forma que puedan ser llevadas a la fase de análisis y posteriormente verse clarificadas en el diseño. Además, se tendrán las especificaciones técnicas para el desarrollo del sistema.

8.2 Alcance

Se persigue que el sistema provea los medios necesarios para:

- Desarrollar los agentes que recuperen la información de la base de datos Oracle 8.0, sistema operativo AIX de IBM y sistema ERP Baan IV.
- Registro de los indicadores de confiabilidad, disponibilidad y fiabilidad.
- Registro de los usuarios que tendrán acceso al sistema a los módulos Intranet y Extranet.
- Mostrar los indicadores al usuario final (Cliente).

8.3 Definiciones y abreviaturas.

Agentes de Software: No existe una definición exacta para el término agente debido a su utilización en diferentes áreas de las ciencias de la computación. Todos los agentes poseen las siguientes características: autonomía, inteligencia e interacción. Un agente es un

programa de software que puede realizar alguna acción definida para un usuario, con un nivel de inteligencia que le permite desarrollarla de alguna manera autónoma y que puede interactuar con su entorno.

Indicadores: Son muy utilizados para describir o medir eventos, procesos, tareas de modo que el usuario final pueda tomar decisiones.

Sistema ERP: Un sistema *ERP (Enterprise Resource Planning)*, según Kwon [Kwon 01], es un paquete de software amplio integrado empresarial diseñado para mantener los más altos estándares de calidad de los procesos empresariales.

Sistema operativo AIX: Es un sistema operativo propietario de la empresa IBM, es muy similar al sistema operativo UNÍX.

Base de datos (Oracle): Es un repositorio de datos que es muy utilizado en la implementación de sistemas ERP y por contar con versiones que corren en el sistema operativo AIX.

8.4 Descripción de las perspectivas del producto

Perspectivas del producto

Mediante la descripción detallada de los indicadores de confiabilidad, disponibilidad y fiabilidad se podrá desarrollar un sistema a medida para cada tipo de *ERP*.

Permitirá conocer los errores, problemas, soluciones, métricas de este tipo de sistemas, base de datos y sistema operativo.

Servirá como modelo a seguir en el desarrollo de soluciones para otro tipo de sistemas.

El desarrollo de un prototipo basado en el sistema *ERP* Baan IV permitirá demostrar los beneficios obtenidos en la implementación de este sistema (CODA)

Sistema que muestra resultados y de sustento que apoye la toma de decisiones.

Sistema de conocimiento vía alarmas o alertas para detectar y prevenir problemas.

Interfases con otros sistemas

Casi de inmediato se pueden identificar una interfase con una hoja de cálculo para mostrar los datos gráficamente u otra herramienta de visualización gráfica.

Características de las interfases de usuario

Esto implica que las páginas web cumplan con lo siguiente:

- No navegar por muchas páginas y tener casi toda la funcionalidad en un número reducido de páginas.
- Tener la posibilidad de acceder a las *todas* las opciones.
- Posibilidad de corrección de data errónea sin necesidad de intervención constante de administradores del sistema o súper usuarios.
- Mensajes de error suficientemente claros para determinar el curso de acción necesario. en cada caso. Aún los que requieran la intervención de otras autoridades.

Interfases de hardware

Especificando los componentes de hardware generales sobre los que se configuraría el sistema podemos afirmar que:

Se implementará sobre la red que disponga la empresa donde reside el sistema ERP Baan IV.

Al ser un sistema orientado a Web el acceso está disponible utilizando un código de usuario y una clave de seguridad sin necesidad de instalación de ejecutables y disponible a todos los usuarios a través de un navegador web.

Interfases de software

Especificando principalmente los componentes de software con los que trabajará el sistema tenemos:

- Sistema Operativo: Windows 2000 para el servidor (incluido IIS(Internet Information Services))
- Explorador de Internet (Internet Explorer)
- Manejador de base de datos MSSQL
- Sistema Operativo AIX v 4.3
- Manejador de Base de Datos Oracle v 8.0
- Sistema ERP BAAN IV C2
- ASP (Active Server Pages)

8.5 Requerimientos para el rendimiento adecuado del sistema

Se describirán las características de los usuarios del sistema y algunas regulaciones con relación a las políticas del uso del sistema de carácter administrativo y de auditoria.

Políticas de regulación de uso

Sumadas a las políticas de seguridad, que se explicarán técnicamente más adelante, es la misma empresa donde reside el sistema la que determina el alcance de cada tipo de usuario sobre el sistema. De la misma manera, determina el tiempo por el cual va a estar en condición de activo el sistema.

Limitaciones de hardware y de espacio

Dependiendo de la naturaleza y el alcance del proceso. Cuando el proceso se realiza en varios servidores y éstos están alejados, el rendimiento de la aplicación final dependerá de la configuración de la red establecida.

Igualmente si en el proceso se realizan consultas con muchos terminales es también el tiempo de acceso a las bases de datos deberá medirse a fin de mejorar su tiempo de respuesta.

Operaciones paralelas

Luego de haber recibido la información del sistema CODA los usuarios deberán resolver los problemas o informar del problema para buscar solucionarlos.

Este trabajo en los casos de problemas graves se deberá resolver convocando a los proveedores del software o hardware.

Características de los usuarios

Para este sistema existen dos tipos de usuario:

Usuario de Sistemas (Usuario): Al recibir las alertas y mails tiene que tomar acción y solucionar los problemas y tiene acceso al módulo Intranet.

Usuario Final (Cliente): que tiene acceso al módulo de extranet donde puede estar informado respecto a cuándo y porqué está el sistema inoperativo o con problemas y desde cuándo y tiene acceso al módulo extranet.

8.6 Dependencias

Existen factores que afectan al rendimiento del sistema como ya se mencionó. Listaremos entonces entre otros a:

- Configuración de los mecanismos de red de la institución.
- Convenios o estándares en cuanto al empleo de determinadas marcas de exploradores.
- Factores legales, administrativos que se dicten durante el desarrollo del sistema.

8.7 Requerimientos

8.7.1 Requerimientos específicos

Se analizarán características técnicas en varios aspectos: hardware, software, seguridad y funcionalidad.

8.7.2 Requerimientos funcionales

Módulo intranet

Para acceder a este módulo se necesita contar con un código de usuario y una contraseña, en el tercer intento el sistema mostrará un mensaje.

Una vez que el acceso ha sido satisfactorio se mostrará la siguiente pantalla donde se puede acceder al menú: Administración, Consultas, Tablas y Salir del Sistema.

Submódulo de tablas

Dentro de esta opción se podrá ingresar:

- Usuarios: Esta opción permite el ingreso de los usuarios del sistema.
- Clientes: Esta opción permite ingresar a los que se les enviará mensajes o alertas.

- Áreas: Cada usuario está relacionado a un área de la empresa.
- Empleados: Cada usuario tiene un código de empleado en la planilla.
- Errores: Existe un código de error del fabricante para cada caso : sistema operativo, base de datos y sistema ERP.
- Origen : Idéntifica de donde procede el error.

Submódulo de Administración

Mediante esta opción se puede ingresar a los Indicadores, Movimientos, Nodos, Problemas, Síntomas y Soluciones.

- Indicadores: En esta opción se ingresarán los indicadores de disponibilidad y fiabilidad.
- Nodos: En esta opción se podrá ingresar los nodos de cada uno de los indicadores.
- Síntomas: Cuando un problema está por originarse existen en la mayoría de los casos alertas previas que se consideran síntomas que evidencian la generación de un problema.
- Problemas: En esta opción se detallan los problemas que el sistema operativo, base de datos o el sistema ERP informan, se puede ver los síntomas o soluciones de cada problema.
- Soluciones: Para cada problema o síntoma generado deberá existir la solución o ingresarla para que si vuelve a ocurrir se pueda actuar o dar solución de manera más rápida.

Módulo extranet

Para acceder a este módulo se necesita contar con un código de usuario y una contraseña, en el tercer intento el sistema mostrará un mensaje.

Una vez que el acceso ha sido satisfactorio se mostrará la siguiente pantalla donde se puede dar click en el menú de Indicadores, Alertas/Mensajes, Estadísticas o Salir.

Submódulo de indicadores

Dentro de esta opción se podrá ingresar a ver los indicadores, nodos o resultados consolidados de los indicadores por diariamente.

- **Indicadores** : Esta opción permite mostrar una lista de los indicadores y luego ir al detalle.
- **Nodos** : Esta pantalla muestra todos los nodos disponibles dentro del sistema, se puede ir al detalle de cada nodo.
- **Síntomas** : Cuando un problema está por originarse existen en la mayoría de los casos alertas previas que se consideran síntomas que evidencian la generación de un problema.
- **Problemas** : En esta opción se detallan los problemas que el sistema operativo, base de datos o el sistema ERP informan, se puede ver los síntomas o soluciones de cada problema.
- **Soluciones** : Para cada problema o síntoma generado deberá existir la solución o ingresarla para que si vuelve a ocurrir se pueda actuar o dar solución de manera más rápida.

Submódulo de mails/alertas/mensajes celular

- Dentro de esta opción se consideran las Alertas, Celular, Mails que el usuario recibió.
- Cuando un evento ha ocurrido el sistema enviará este tipo de mails a fin de dar cuenta de la realización de un problema.
- **Alertas** : para el monitoreo local, el usuario podrá ver todos los eventos del día.

Submódulo de estadísticas

Dentro de esta opción se podrá ver las consultas o reportes

Entradas y salidas del sistema

Entradas

En lo referente a las entradas, el sistema necesitará previamente los siguientes datos:

- Usuarios: Esta opción permite el ingreso de los usuarios del sistema.
- Clientes: Esta opción permite ingresar a los que se les enviará mensajes o alertas.
- Areas: Cada usuario está relacionado a un área de la empresa.
- Empleados: Cada usuario tiene un código de empleado en la planilla.
- Errores: Existe un código de error del fabricante para cada caso: sistema operativo, base de datos y sistema ERP.
- Origen: Identifica de dónde procede el error.
- Indicadores: En esta opción se ingresarán los indicadores de disponibilidad y operatividad.
- Nodos : En esta opción se podrá ingresar los nodos de cada uno de los indicadores.
- Síntomas: Cuando un problema está por originarse existen en la mayoría de los casos alertas previas que se consideran síntomas que evidencian la generación de un problema.
- Problemas: En esta opción se detallan los problemas que el sistema operativo, base de datos o el ERP informan, se puede ver los síntomas o soluciones de cada problema.
- Soluciones: Para cada problema o síntoma generado deberá existir la solución o ingresarla para que si vuelve a ocurrir se pueda actuar o dar solución de manera más rápida.

Salidas

Como todo sistema, debe estar en la capacidad de brindar casi a manera de reportes o consultas por pantalla los siguientes datos:

- Indicadores: permite mostrar una lista de los indicadores.
- Nodos: muestra todos los nodos disponibles dentro del sistema, se puede ir al detalle de cada nodo.
- Síntomas: Cuando un problema está por originarse existen en la mayoría de los casos alertas previas que se consideran síntomas que evidencian la generación de un problema.
- Problemas: se detallan los problemas que el sistema operativo, base de datos o el ERP informan, se puede ver los síntomas o soluciones de cada problema.

- Soluciones: Para cada problema o síntoma generado deberá existir la solución o ingresarla para que si vuelve a ocurrir se pueda actuar o dar solución de manera más rápida.
- Mailes: Cuando un evento ha ocurrido el sistema enviará este tipo de mails a fin de dar cuenta de la realización de un problema.
- Alertas: Para el monitoreo local, el usuario podrá ver todos los eventos del día.

8.7.3 Requerimientos no funcionales

Terminales: Usualmente trabajará con varios terminales conectados vía Internet.

8.8 Diagrama de Casos de Uso

8.8.1 Diagramas de casos de uso general

En la Figura 8.1, mostramos un diagrama de alto nivel donde se apreciarán los procesos fundamentales, tales como: Tablas, Administración, Indicadores, Alertas y Estadísticas que se detallan adelante.

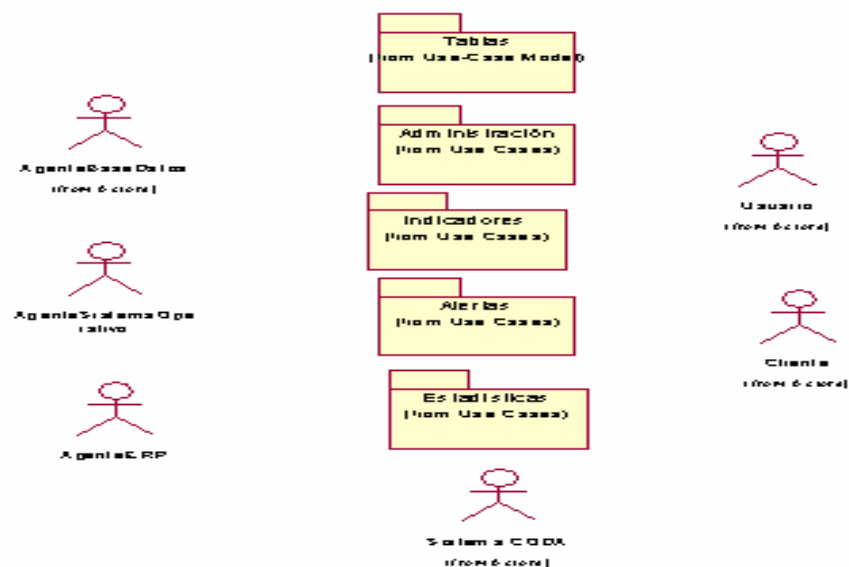


Figura 8.1 Casos de Uso General.

8.8.2 Casos de uso de Tablas

La Figura 8.2 muestra los casos de uso denominado Tablas donde se puede observar las actividades de registrar usuarios, clientes, áreas, empleados, errores, origen.

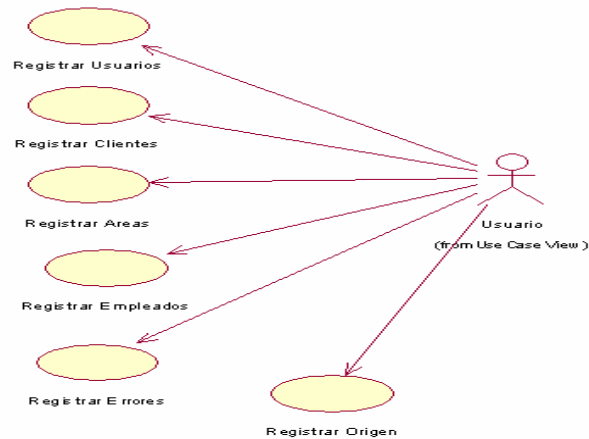


Figura 8.2 Casos de Uso Tablas.

8.8.3 Casos de uso Administración

La Figura 8.3 muestra los casos de uso denominado Administración donde se puede observar las actividades de registro de nodos, registro de indicadores, actualizar movimientos, registrar sistemas, registrar problemas, registrar soluciones.

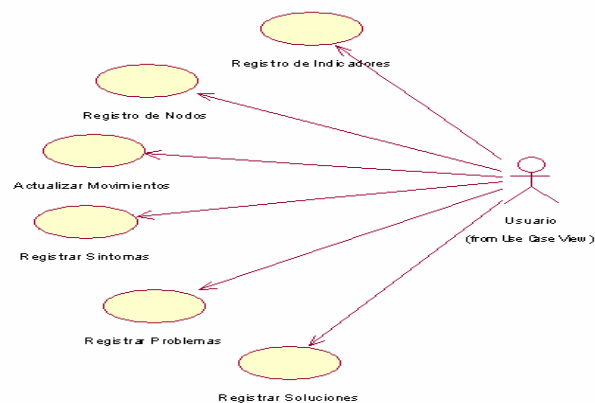


Figura 8.3 Casos de Uso Administración.

8.8.4 Casos de uso Indicadores

La Figura 8.4 muestra los casos de uso denominado Indicadores donde se puede observar las actividades de mostrar indicadores, nodos, movimientos, sistemas, soluciones.

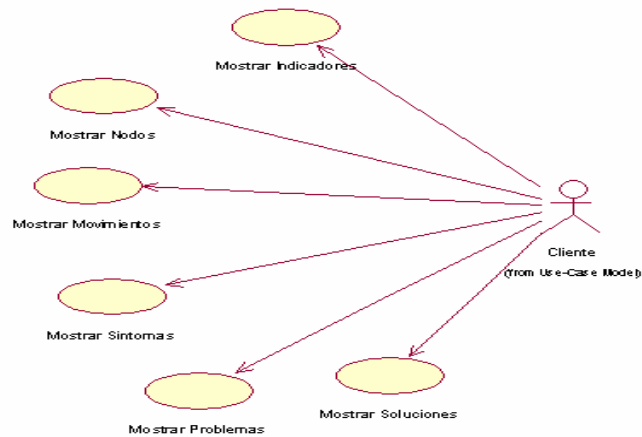


Figura 8.4 Casos de Uso Indicadores.

8.8.5 Casos de uso alertas

La Figura 8.5 muestra los casos de uso denominado Indicadores donde se puede observar las actividades de enviar mail, alertas y mensajes de celular.

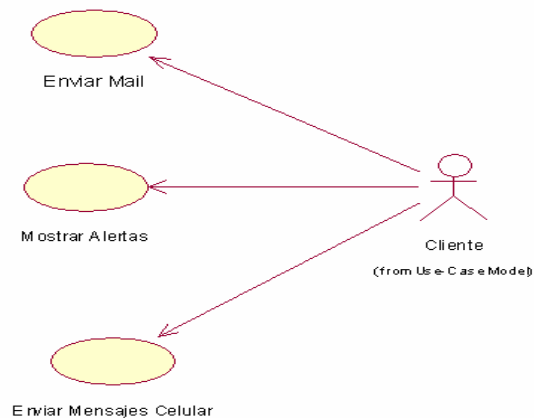


Figura 8.5 Casos de uso alertas.

Capítulo 9: Experimentos numéricos del sistema CODA

9.1 Organización e instancias de prueba

Se ha considerado los siguientes días de proceso: 01, 02 y 03 de enero del año 2006, esto en razón de que se tiene el cierre mensual del mes de diciembre, el cierre del año anterior y existen procesos anuales que sólo se ejecutan estos días.

Los componentes de hardware con los que trabajará el sistema son: Servidor compatible, Servidor IBM X Series Modelo H71, SAN Storage HP. Mientras que los componentes de software con los que trabajará el sistema son: Sistema CODA, Sistema Operativo: Windows 2000 para el servidor (incluido IIS(Internet Information Services)), Explorador de Internet (Internet Explorer), Manejador de base de datos MSSQL 2000, ASP (Active Server Pages), Sistema ERP, Sistema Operativo AIX v 4.3, Manejador de Base de Datos Oracle v 8.0, Sistema ERP BAAN IV C2.

9.2 Resultados numéricos

9.2.1 Indicador confiabilidad del sistema CODA (CODA)

La interpretación de la Figura 9.1, que muestra el indicador de confiabilidad del sistema CODA (CODA) del presente trabajo es el siguiente: El día 01 de enero del 2006, el indicador muestra el valor más bajo con un 82% de confiabilidad; el 03 de enero del 2006, el indicador muestra el valor más alto con un 84% de confiabilidad, ambos valores se encuentran por debajo del intervalo del valor normal y el valor mínimo considerado (87% y 73%), por lo que se necesita revisar todos los indicadores de disponibilidad y fiabilidad de los días 01 al 03 de enero de 2006.

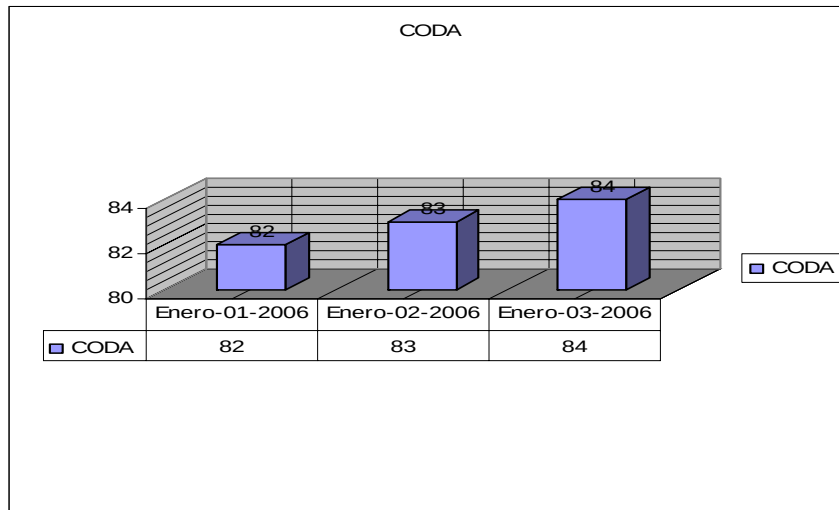


Figura 9.1 Indicador de confiabilidad sistema CODA.

9.2.2 Indicador fiabilidad del sistema ERP (ERP001)

La interpretación de la Figura 9.2, que muestra el indicador de fiabilidad del sistema ERP (ERP001) del presente trabajo es la siguiente: El día 01 de enero del 2006, el indicador muestra el valor más alto con un 93% de fiabilidad; el 02 y 03 de enero del 2006, el indicador muestra el valor más bajo con un 92% de fiabilidad, ambos valores se encuentran por encima del intervalo del valor normal y el valor máximo considerado (90% y 98%), por lo que podemos inferir que el problema no está en el sistema ERP.

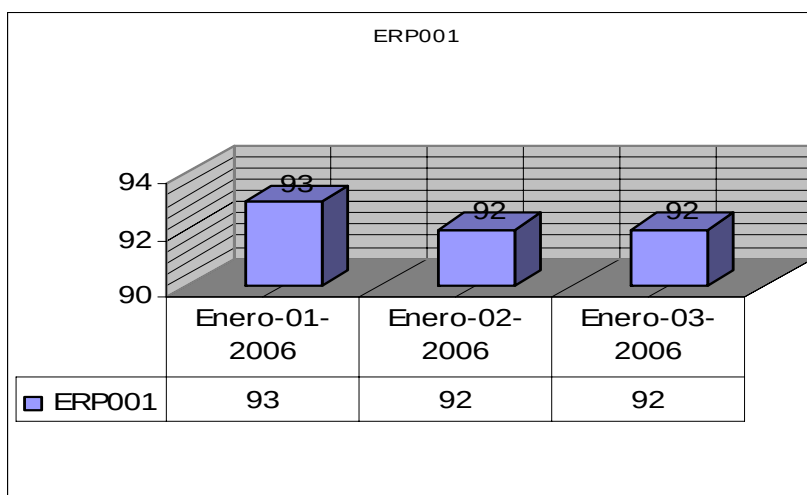


Figura 9.2 Indicador de fiabilidad del sistema ERP.

9.2.3 Indicador disponibilidad de la base de datos (BD001)

La interpretación de la Figura 9.3 que muestra el indicador de disponibilidad de la base de datos (BD001) del presente trabajo es la siguiente: El día 01 de enero del 2006, el indicador muestra el valor más bajo con un 65% de disponibilidad; el 03 de enero del 2006, el indicador muestra el valor más alto con un 75% de disponibilidad, ambos valores se encuentran por debajo del intervalo del valor normal y el valor mínimo considerado (80% y 95%), por lo que podemos inferir que debemos revisar los eventos que ocasionaron la baja en este indicador en el día 01 de enero del 2006, que influyeron en el indicador de confiabilidad total.

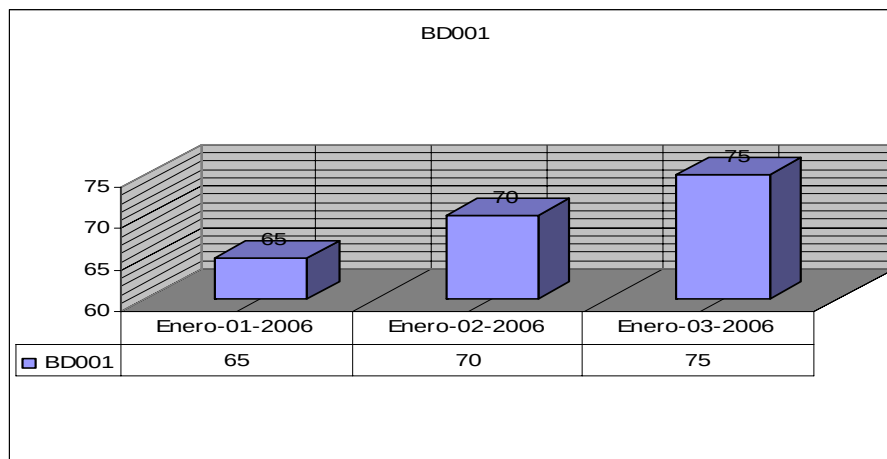


Figura 9.3 Indicador de disponibilidad de la base de datos.

9.2.4 Indicador disponibilidad del sistema operativo (SO001)

La interpretación de la Figura 9.4 que muestra el indicador de disponibilidad del sistema operativo (SO001) del presente trabajo es la siguiente: El día 01 de enero del 2006, el indicador muestra el valor más alto con un 92% de disponibilidad; el 02 de enero del 2006, el indicador muestra el valor más bajo con un 89% de disponibilidad, el primer valor coincide con el valor normal, en cambio el otro está por debajo del valor normal y el

mínimo considerado (90%), aún cuando el valor del día 02 de enero del 2006 es bajo se debería revisar las causas de este resultado.

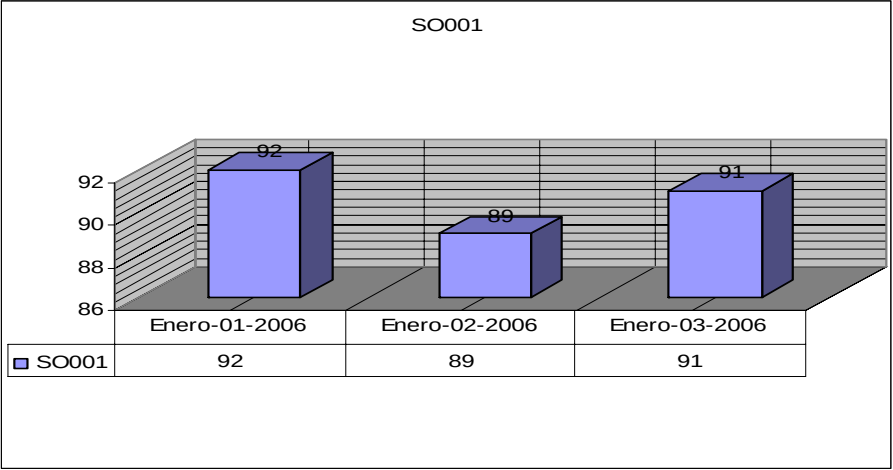


Figura 9.4 Indicador de disponibilidad del sistema operativo.

Capítulo 10: Conclusiones y trabajos futuros

10.1 Conclusiones

Se ha modelado un sistema multi-agente denominado sistema CODA que permite generar los indicadores de confiabilidad, disponibilidad y fiabilidad del sistema ERP Baan y mostrar estos indicadores a través de diagramas y tablas. También, el sistema CODA desarrollado permite al usuario final monitorear la actividad del sistema ERP mediante los indicadores, gráficos, alertas y mails.

Se ha modelado los agentes de software para cada uno de los indicadores del Sistema CODA, y analizado los valores resultantes de confiabilidad, disponibilidad y fiabilidad del sistema ERP Baan IV, base de datos Oracle 8.0 y sistema operativo AIX de IBM. Los valores mínimos, normales y máximos, como mostrados en las tablas respectivas, indican las variaciones de los indicadores antes mencionados.

En la Figura 10.1 se extiende la solución del problema a todas las aplicaciones de la empresa o corporación.

10.2 Trabajos futuros

Multi-compañía: en el caso de que el Sistema ERP Baan trabaje en esta modalidad deberá de hacerse las modificaciones necesarias para capturar los movimientos y consolidarlos de modo que se pueda mostrar índices por cada compañía.

Otras bases de datos / sistemas operativos: El sistema ERP Baan provee soporte para otras bases de datos (MS SQL, Db2, etc) y sistemas operativos (Linux, Windows 2000, etc), se deberá pensar en desarrollar agentes que permitan acceder y obtener la información necesaria.

Mensajes vía celular: se propone como una mejora la implementación de enviar los mensajes o alertas vía mensajes de texto. Acceso vía PDA (Personal Device Assistance): se deberán elaborar los programas que permitan acceder a través de este medio.

10.3 Posibles aplicaciones en las empresas

Las empresas grandes, medianas y pequeñas pueden contar con este modelo para implementarlo dentro de su organización. Estas empresas desarrollarán los sistemas de acuerdo a los sistemas existentes, se ahorrará tanto en número de recursos humanos como por la compra de software, se contará con indicadores de disponibilidad, confiabilidad y fiabilidad para tomar medidas en caso de que estos se desvíen del curso normal. Los especialistas tendrán un patrón o modelo para enfrentar un problema en el caso no cuenten con el software desarrollado, debido a que será una guía para ubicar el problema y darle solución, mediante los diagramas y tablas el usuario final (no necesariamente conocedor de TI) podrá conocer el estado de los sistemas que apoyan a su trabajo diario, se contará con datos históricos de manera que se pueden tomar medidas para resolver problemas recurrentes. La empresa no tendrá que depender de terceros para identificar los problemas y resolverlos.

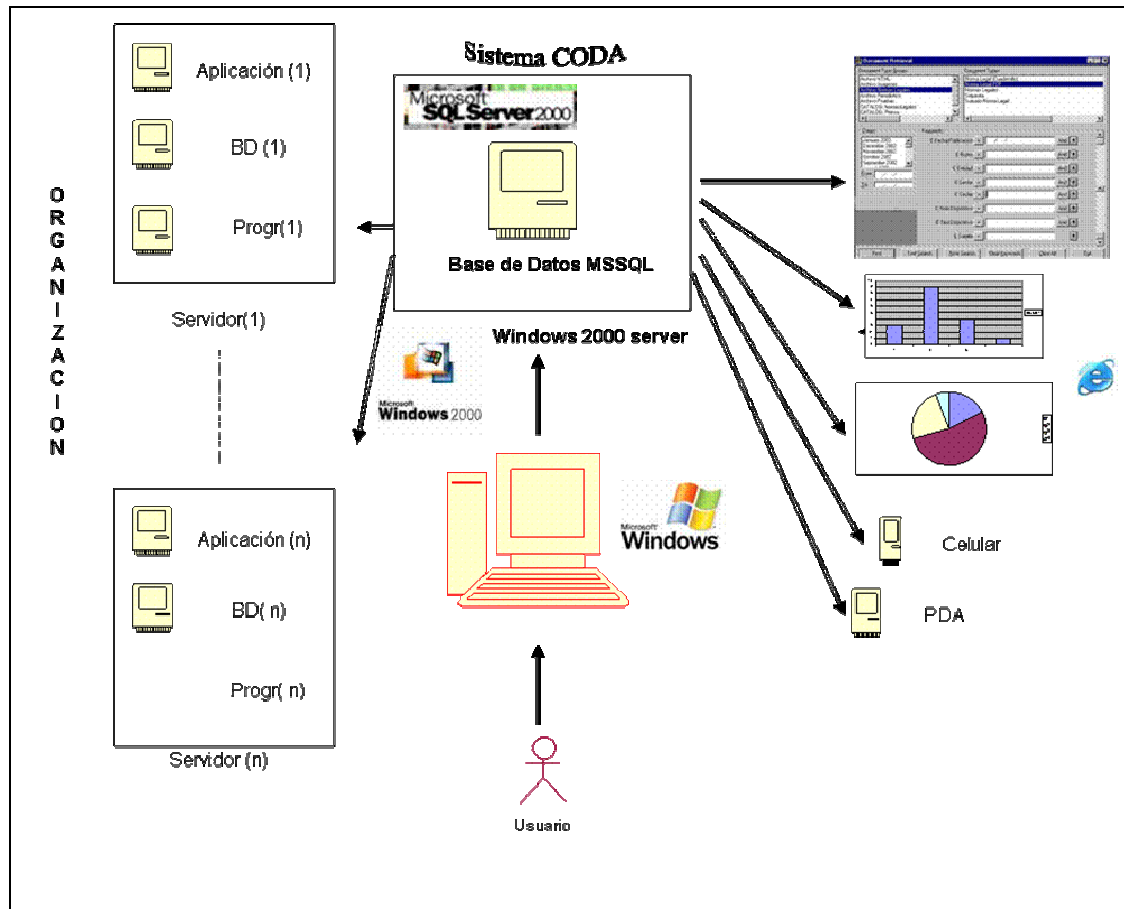


Figura 10.1 Solución final (generalización).

BIBLIOGRAFIA

[**Maldonado**] Juan C. Maldonado, site: <http://www.icmc.usp.br/php/>, Universidad de Sao Paolo, Brasil, accesado en Abril 2007.

[**Hung**] Hung, site: <http://www.iist.unu.edu/~dvh/>, United Nations University, Macao, China, accesado en Abril 2007.

[**George**] George, site: <http://www.iist.unu.edu/~cwg/>, United Nations University, Macao. China, accesado en Abril 2007.

[**Baan**] ERP Baan, site: <http://www.support.baan.com>, Base de datos de conocimiento, accesado en Abril 2007.

[**Oracle**] Oracle, site: <https://metalink.oracle.com/>, Base de datos de conocimiento, accesado en Abril 2007.

[**IBM**] IBM, site: <http://www.ibm.com/support/us/>, Soporte on line, accesado en Abril 2007.

[**BMC**] BMC, site: <http://www.bmc.com>, accesado en diciembre 2007.

[**QUEST**] QUEST SOFTWARE, site: <http://www.quest.com/sap/>, accesado en diciembre 2007.

[**Symantec**] Symantec, site: <http://www.symantec.com/enterprise/products>, accesado en diciembre 2007.

[**Kwon 01**] Kwon, O. B. and Leeb, J. J. A multi-agent intelligent system for efficient ERP maintenance, Instituto Handong de Informacion Tecnologica, Expert System and Application, Volume 21, Issue 4, November 2001, Pages 191-202.

[**Chandrashekhar 03**] Chandrashekhar Ch., Deshmukh, S. G. and Chattopadhyay, R. Application of principles of event related open systems to business process

reengineering, ACM: Computers and Industrial Engineering, Volume 45, Issue 3, October 2003, Pages 347-374.

[Ling-yu 07] Ling-yu, Ch. Ch., Dua, T and Lai, V. S. Continuous auditing with a multi-agent system, Decision Support Systems, volume 42, issue 4, January 2007, Pages 2274-2292.

[Zeus] Zeus Agent Builder Toolkit, site: <http://labs.bt.com/projects/agents/zeus/>., accesado en abril 2007.

[Gomez 02] Gomez Sans J.; Análisis y Diseño de Sistema Multi-Agente, Curso de Doctorado, Dep. Sistemas Informáticos y Programación, Universidad Complutense de Madrid: 2002.

[Sommerville 02] Sommerville I.; Ingeniería de Software, Mexico 2002, Pearson Education; pags. 400.

[APICS 02], APICS Dictionary, 9th edition, 2002

[GartnerGroup] GartnerGroup, site: www.gartnergroup.com, accesado en Diciembre 2006.

[BDO 00] BDO; Términos de Referencia de la Implementación del Sistema Baan en Editora Perú, implementado por BDO Consulting SAC: 2000.

[Cambridge 96], Knowledge Engineering Review, Vol. 11, No 3, pp.1-40, Sept 1996. © Cambridge University Press, 1996.

ANEXOS

Consultar el capítulo completo en formato impreso