



**TECNOLÓGICO  
DE MONTERREY®**

**Campus Ciudad de México**

**Escuela de Diseño, Ingeniería y Arquitectura**

**Maestría en Ciencias de la Computación**

*“Diseño e Implementación de un Sistema Gestor para  
Aplicaciones de Misión Crítica Basado en la Arquitectura  
ADSOA”*

Autor:

Marco Antonio Corona Ruiz

Director Interno:

Dr. Yoel Ledo Mezquita

Director Externo:

Dr. Carlos Pérez Leguízamo



**TECNOLÓGICO  
DE MONTERREY**

**Biblioteca**  
Campus Ciudad de México

Octubre 2014

# Contenido

1.	Introducción .....	1
1.1.	Antecedentes .....	1
1.2.	Definición del Problema .....	3
1.2.1.	Propuestas Existentes .....	3
1.3.	Objetivos .....	7
1.4.	Justificación .....	7
1.5.	Aportación.....	8
1.6.	Metodología .....	8
1.6.1.	FASE 1. Analizar las Distintas Tecnologías Utilizadas en la Arquitectura ADSOA. ....	9
1.6.2.	FASE 2. Definir las Tecnologías para el Proceso de Monitoreo. ....	9
1.6.3.	FASE 3. Diseño de las Tecnologías a Utilizar para el Proceso de Monitoreo. ....	10
1.6.4.	FASE 4. Seleccionar el Caso de Prueba para Implementar el Diseño y la Arquitectura para el Proceso de Monitoreo.....	10
2.	Marco Teórico .....	12
2.1.	Sistemas de Misión Crítica .....	12
2.1.1.	Descripción.....	12
2.1.2.	Propiedades .....	16
2.2.	ADS (Sistemas Autónomos Descentralizados).....	19
2.2.1.	Conceptos .....	19
2.2.2.	Arquitectura.....	20
2.3.	SOA (Arquitectura Orientadas a Servicios).....	22
2.3.1.	Conceptos .....	22
2.3.2.	Arquitectura.....	24
2.4.	ADSOA (Arquitectura Orientada a Servicios Autónoma Descentralizada).....	27
2.4.1.	Conceptos .....	27
2.4.2.	Atom.....	29
2.4.3.	Arquitectura.....	29
3.	Objetivos de investigación .....	37
3.1.	Planteamiento del Problema .....	37
3.2.	Requerimientos Funcionales.....	37
3.3.	Requerimientos no Funcionales.....	37
3.4.	Trabajos Relacionados (Monitores).....	38

3.4.1.	Hyperic HQ .....	38
3.4.2.	WSO2 BAM .....	40
3.4.3.	Monitoreo de Aplicaciones SOA Basado en JAVA .....	41
3.4.4.	Big Sister .....	43
3.5.	Propuesta .....	45
4.	Sistema Gestor para Aplicaciones de Misión Crítica Basado en la Arquitectura ADSOA ...	46
4.1.	Introducción .....	46
4.1.1.	Niveles de Diseño Arquitectónico .....	46
4.2.	Modelo de Negocio .....	48
4.3.	Modelo Conceptual .....	49
4.3.1.	Sistema Gestor para Aplicaciones de Misión Crítica .....	50
4.4.	Operatividad .....	52
4.4.1.	Flujo Básico .....	52
4.4.2.	Eventos .....	53
4.4.3.	Diagrama de Estado de los Procesos .....	54
4.4.4.	Diagrama de Estados de los Servicios .....	55
4.5.	Formato de los Mensajes para el Proceso de Gestión .....	56
4.5.1.	Mensajes Reservados .....	56
4.5.2.	Formato de los Mensajes .....	58
4.6.	Arquitectura .....	60
4.7.	Diseño .....	63
4.7.1.	Modelo de la Base de Datos .....	64
4.7.2.	Diagrama Entidad Relación .....	65
4.7.3.	Diagramas de Clases .....	66
4.8.	Configuración General .....	71
4.8.1.	Configuración del Atom .....	71
4.8.2.	Configuración del Integrador .....	72
4.8.3.	Configuración del Conector Empresarial .....	73
4.8.4.	Configuración del MDB y WebSocket .....	75
5.	Pruebas y Resultados .....	79
5.1.	Introducción .....	79
5.2.	Selección de Caso de Prueba .....	79
5.3.	Escenario de Caso Práctico .....	80
5.4.	Estrategia Adoptada .....	81
5.5.	Resultados Esperados .....	81

5.6.	Integración del Sistema Gestor al Prototipo de Subasta de Títulos Gubernamentales...	82
5.6.1.	Caso de Prueba I.....	82
5.6.1.1.	Sistema Monolítico .....	82
5.6.1.2.	Resultados del Caso de Prueba I .....	88
5.6.2.	Caso de Prueba II .....	89
5.6.2.1.	Sistema Distribuido .....	89
5.6.2.2.	Resultados del Caso de Prueba II.....	96
5.6.3.	Caso de Prueba III.....	97
5.6.3.1.	Sistema Tolerante a Fallos .....	97
5.6.3.2.	Resultados del Caso de Prueba III.....	110
5.6.4.	Caso de Prueba IV.....	111
5.6.4.1.	Sistema Especializado .....	111
5.6.4.2.	Resultados del Caso de Prueba IV .....	122
6.	Conclusiones y Líneas Futuras.....	123
6.1.	Conclusiones.....	123
6.2.	Líneas Futuras .....	127
6.2.1.	Herramienta Complementaria para Modelar los Proceso de Negocio de Manera Gráfica.	127
6.2.2.	Aportación al Protocolo de Foliado de la Arquitectura ADSOA.....	127
6.2.3.	Línea de Investigación Sobre el Cómputo Autónomo. ....	128
6.2.4.	Incorporar el Monitoreo de Sistemas con Arquitectura Distinta a ADSOA. ....	128

## Lista de Figuras

Figura 1.1 Dos tipos de mecanismos de monitorco.....	4
Figura 1.2 Gartner Hype Cycle.....	6
Figura 2.1 Curva costo/confiabilidad .....	14
Figura 2.2 Propiedades de la confiabilidad .....	16
Figura 2.3 Arquitectura ADS.....	21
Figura 2.4 Formato de mensaje incluido el código de contenido .....	22
Figura 2.5 Arquitectura SOA.....	25
Figura 2.6 Modelo conceptual ADSOA .....	28
Figura 2.7 Arquitectura ADSOA.....	31
Figura 2.8 Secuencialidad y transaccionalidad.....	32
Figura 2.9 Sincronización con otras Instancias .....	33
Figura 2.10 Encabezado del mensaje .....	34
Figura 2.11 Cuerpo del mensaje .....	36
Figura 3.1 Arquitectura de software de Hyperic HQ.....	39
Figura 3.2 Arquitectura de software de WSO2 BAM .....	41
Figura 3.3 Diagrama de despliegue de la aplicación monitoreada .....	43
Figura 3.4 Componentes Big Sister.....	44
Figura 4.1 Niveles del diseño arquitectónico .....	47
Figura 4.2 Modelo de negocio.....	49
Figura 4.3 Modelo conceptual.....	52
Figura 4.4 Flujo básico .....	54
Figura 4.5 Diagrama de estado de los procesos.....	54
Figura 4.6 Diagrama de estados de los servicios.....	55
Figura 4.7 Diagrama de arquitectura nivel de recursos .....	61
Figura 4.8 Diagrama de arquitectura nivel de integración .....	62
Figura 4.9 Diagrama de arquitectura nivel de negocio.....	63
Figura 4.10 Modelo de base de datos .....	64
Figura 4.11 Diagrama entidad relación .....	65
Figura 4.12 Diagrama de clases del gestor .....	66
Figura 4.13 Diagrama de clases del integrador .....	66
Figura 4.14 Diagrama de clases del conector empresarial .....	67
Figura 4.15 Diagrama de clases del message driven bean .....	67
Figura 4.16 Diagrama de clases de la clase ServiciosBean .....	68

Figura 4.17 Diagrama de clases de la clase ConsultasBean .....	68
Figura 4.18 Diagrama de clases del ServerEndPoint .....	69
Figura 4.19 Diagrama de clases del mensaje MsjRecibeGestor.....	69
Figura 4.20 Diagrama de clases del mensaje MsjRecibeGestorIntegrador .....	70
Figura 4.21 Diagrama de clases del mensaje MsjRecibeGestorConector .....	70
Figura 4.22 Archivo de configuración dtd.....	72
Figura 4.23 Archivo hibernate.cfg.xml .....	73
Figura 4.24 Constantes para el JMS Topic Client .....	74
Figura 4.25 Inicializar JMSTopicClient .....	74
Figura 4.26 Envío del mensaje de aviso .....	75
Figura 4.27 Message Driver Bean .....	76
Figura 4.28 CDI.....	76
Figura 4.29 ServerEndPoint .....	78
Figura 5.1 Arquitectura del sistema monolítico .....	82
Figura 5.2 Grafo inicial del sistema monolítico .....	83
Figura 5.3 Etapa de asignación del sistema monolítico.....	84
Figura 5.4 Grafo del sistema monolítico en la etapa de asignación .....	85
Figura 5.5 Consulta de la base de datos del sistema monolítico .....	86
Figura 5.6 Bitácora del gestor para el sistema monolítico .....	88
Figura 5.7 Arquitectura del sistema distribuido .....	89
Figura 5.8 Grafo inicial del sistema distribuido .....	90
Figura 5.9 Entidad subasta validador del sistema distribuido .....	91
Figura 5.10 Entidad subasta del sistema distribuido .....	92
Figura 5.11 Grafo del sistema distribuido en la etapa de publicación.....	93
Figura 5.12 Consulta de la base de datos del sistema distribuido .....	94
Figura 5.13 Bitácora del gestor para el sistema distribuido .....	96
Figura 5.14 Arquitectura del sistema tolerante a fallas .....	98
Figura 5.15 Grafo inicial del sistema tolerante a fallas .....	99
Figura 5.16 Entidad Subasta del sistema tolerante a fallas.....	100
Figura 5.17 Error en la etapa de convocatoria del sistema tolerante a fallas .....	101
Figura 5.18 Registro del evento erróneo en la base de datos .....	102
Figura 5.19 Registro del evento erróneo en la bitácora del servidor posturas.....	103
Figura 5.20 Entidad subasta validador, tolerancia a fallos .....	104
Figura 5.21 Entidad subasta, tolerancia a fallos .....	104
Figura 5.22 Instancia del servidor posturas .....	105
Figura 5.23 Entidad Subasta Validador en la etapa de publicación. ....	106

Figura 5.24 Entidad Subasta en la etapa de publicación. ....	106
Figura 5.25 Grafo del sistema tolerante a fallos en la etapa de publicación.....	107
Figura 5.26 Consulta de la base de datos del sistema tolerante a fallos. ....	108
Figura 5.27 Bitácora del gestor para el sistema tolerante a fallos. ....	110
Figura 5.28 Arquitectura del sistema especializado. ....	111
Figura 5.29 Grafo inicial del sistema distribuido. ....	112
Figura 5.30 Instancias especializadas.....	113
Figura 5.31 Entidad Subasta y el ingreso de una postura para CETES.....	114
Figura 5.32 Instancias especializadas para CETES.....	114
Figura 5.33 Entidad Subasta y el ingreso de una postura para BONOS. ....	115
Figura 5.34 Entidades especializadas para BONOS.....	115
Figura 5.35 Grafo del sistema especializado en la etapa de cierre. ....	116
Figura 5.36 Entidad subasta validador del sistema especializado en la etapa de publicación. ...	117
Figura 5.37 Entidad Subasta del sistema especializado en la etapa de publicación. ....	117
Figura 5.38 Grafo del sistema especializado en la etapa de publicación. ....	118
Figura 5.39 Consulta de la base de datos del sistema especializado. ....	119
Figura 5.40 Bitácora del gestor para el sistema especializado. ....	121

## Lista de Tablas

Tabla 1.1 Cronograma de actividades .....	11
Tabla 2.1 Terminología de fiabilidad. ....	15
Tabla 4.1 Estados de los procesos .....	55
Tabla 4.2 Estados de los servicios .....	56
Tabla 4.3 Tipos de datos.....	57
Tabla 4.4 Mensajes reservados.....	57
Tabla 4.5 Formato mensaje CambioEstado .....	58
Tabla 4.6 Formato mensaje IntegrarCambioEstado .....	59
Tabla 4.7 Formato mensaje NotificarConector .....	60
Tabla 5.1 Descripción de los servicios .....	81



# 1.Introducción

## 1.1.Antecedentes

En las últimas décadas los avances tecnológicos han evolucionado rápidamente, estos avances han sido la pauta en la manera de planear un negocio o servicio, también se ven reflejados en el aumento en la demanda del rendimiento y escalabilidad de las aplicaciones de software. Además, las exigencias del mercado son cada vez mayores y las principales necesidades que solicita cualquier consumidor al adquirir algún producto o servicio son la calidad y el tiempo de respuesta.

Por otro lado, para las empresas la información con la que cuentan es la parte esencial del negocio, su principal activo. Por esta razón, es de suma importancia que estas empresas cuenten con procesos y sistemas **confiables** y bien definidos, que estén disponibles en cualquier momento cuando se les requiera.

Por ejemplo, en el sistema bancario mexicano, diariamente se generan miles de transacciones realizadas por cajeros automáticos, pagos con tarjetas, transferencias bancarias, entre otros, lo que exige que la red financiera esté **disponible** durante el período de operación de la banca, además de proporcionar **los tiempos de respuesta** que se necesitan [26]. Por lo consiguiente este tipo de sistemas se convierten en críticos ya que cualquier fallo que pudiera existir, generaría pérdidas económicas importantes.

Otro ejemplo son, los sistemas de control de las aerolíneas por donde viajan millones de personas diariamente, los cuales deben estar listos y **disponibles** las 24 horas del día durante todo

el año. Estos sistemas cumplen con las propiedades de **fiabilidad, seguridad y protección** ya que cualquier fallo en su sistema de navegación puede poner en riesgo las vidas de las personas que utilizan este servicio.

Los tipos de sistemas antes mencionados se les conocen como “Sistemas de Misión Crítica”, de los cuales dependen tanto las personas como los negocios que los utilizan. Si este tipo de sistemas no ofrecen sus servicios como se espera o como fueron especificados pueden ocasionar pérdidas importantes y graves problemas.

Actualmente existe una arquitectura novedosa que combina las características de SOA (Arquitectura Orientada a Servicios) y ADS (Sistemas Autónomos Descentralizados) llamada ADSOA, diseñada para sistemas de misión crítica. ADSOA está conformado por entidades autónomas que pueden proporcionar o solicitar servicios mediante mensajes.

Con base en los conceptos de “Sistemas de Misión Crítica” y los requerimientos de los mismos, la propuesta de este trabajo es la implementación de un sistema gestor para aplicaciones de misión crítica que utilicen una arquitectura ADSOA, ya que actualmente no existe un mecanismo de monitoreo que cumpla con este fin.

## **1.2. Definición del Problema**

¿Por qué se necesita diseñar e implementar un sistema gestor para aplicaciones de misión crítica que utilizan la arquitectura ADSOA?

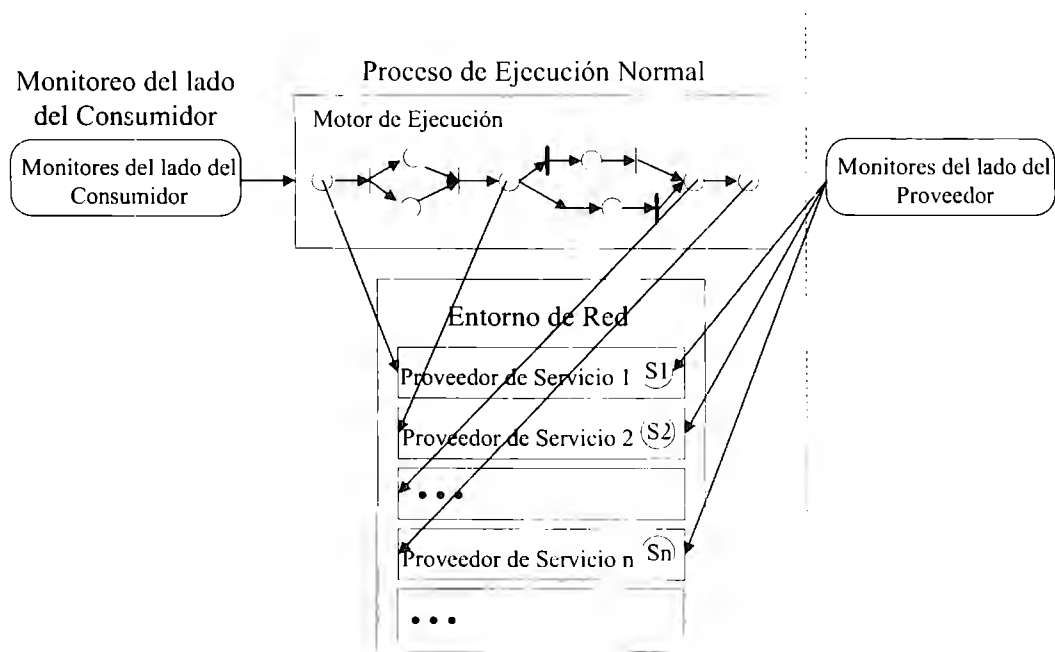
Actualmente, las empresas buscan que sus procesos de negocio y los sistemas de TI con los que cuentan coincidan en algún punto, es decir, cuando cualquier proceso de negocio requiera un cambio, se pueda realizar sin la intervención manual de los expertos del sistema. El escenario ideal es cuando los expertos del negocio configuran el sistema y los cambios se efectúan por sí mismos sin otra intervención.

La necesidad de integración de sistemas de misión crítica en entornos que puedan cumplir con todas las características que estos necesitan va en aumento. De ahí la importancia en contar con mecanismos de gestión que proporcione información veraz y oportuna sobre el estado de los procesos del negocio, y que esto permita tomar mejores decisiones para la empresa.

### **1.2.1. Propuestas Existentes**

La Arquitectura Orientada a Servicios (SOA) ayuda a cerrar esta brecha entre los problemas del negocio y TI, gracias a sus principios de composición de servicios. A continuación se describe de manera general el análisis realizado sobre opciones existentes de monitoreo en una arquitectura SOA.

En [4] se realizó una investigación que describe los tipos de mecanismos de monitoreo que se pueden realizar sobre una composición de servicios y su impacto sobre la disponibilidad de los mismo. En la figura 1.1 se describen los tipos de monitoreo que se pueden realizar tanto del lado del consumidor como del proveedor de servicios.



**Figura 1.1** Dos tipos de mecanismos de monitoreo

En el monitoreo del lado del proveedor, para asegurar la disponibilidad de la composición de los servicios se propone mediante los modelos *Continuous Time Markov Chain model (CTMC)* y *Discrete Time Markov Chain(DTMC)*, para obtener el intervalo de monitoreo óptimo y el conjunto de servicios con mayor impacto para minimizar el uso de los recursos utilizados para el monitoreo.

Por otro lado, el mecanismo de monitoreo de lado del consumidor considera la disponibilidad y el tiempo de respuesta para su estudio, seleccionando solamente parte de los servicios y calculando el número de servicios redundantes para estos. Utiliza los modelos DTMC y QoS (ejecución de servicios en paralelo).

La principal diferencia entre estos mecanismos, es el número de servicios redundantes utilizados y el intervalo de monitoreo calculado. Los resultados de esta investigación muestran que a pesar de optimizar los recursos utilizados, aún existen problemas con la disponibilidad y los tiempos de respuesta.

Con respecto a la gestión de aplicaciones SOA, existen soluciones que proporcionan este servicio, pero tienen la desventaja de que solo se enfocan a ciertos tipos de problemas. Una opción es Hyperic HQ[10], el cual es un sistema de administración y monitoreo de infraestructura web. Es utilizado para gestionar la complejidad de la infraestructura, agiliza las operaciones y mejora los niveles de servicio.

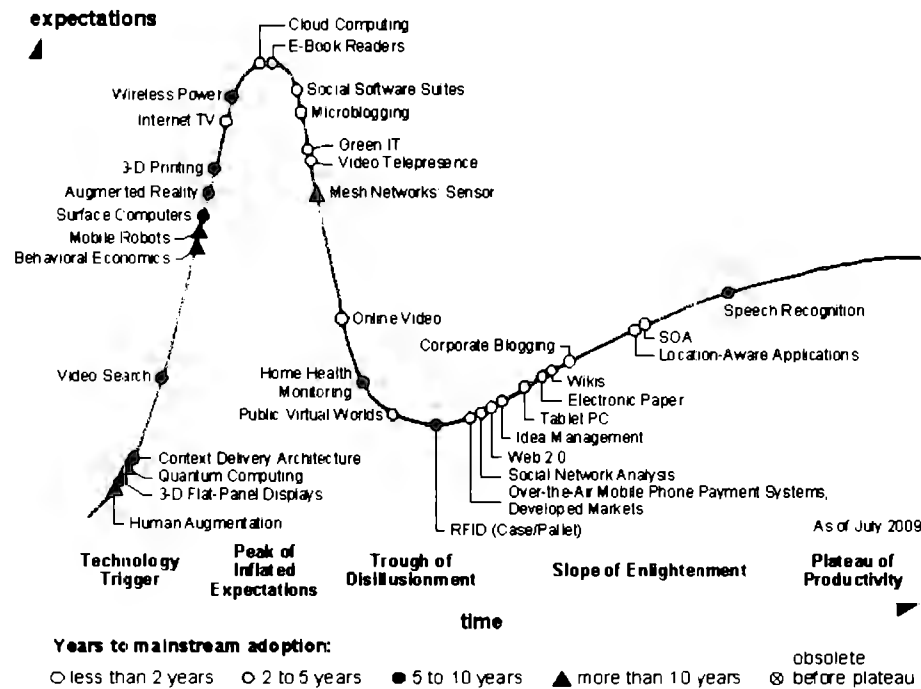
Otra opción es WSO2 BAM[11] la cual es una plataforma para desarrollo y monitoreo de sistemas SOA dividida en módulos. El usuario interactúa con el sistema mediante una interface, donde se visualiza la información del negocio y las métricas proporcionadas por el servidor.

En [5] se ofrece una propuesta para el monitoreo de sistemas basados en SOA dividido en componentes, su principal objetivo es la simplicidad e independencia de la plataforma que se utilice, ya que está basado en JAVA y es soportado por cualquier servidor de aplicaciones. Se pueden utilizar cualquier *framework* para implementar los servicios (Spring MVC, WS-JAX-WS, entre otros).

También se encontró una infraestructura de monitoreo de sistemas que implementa las características de Big Sister[7], esta opción proporciona un monitoreo sobre los siguientes aspectos: el espacio disponible en disco, el uso de la memoria, el espacio disponible en las bases

de datos, la ejecución propia de las diferentes aplicaciones, etc. Pero no proporciona las características de un monitoreo para sistemas de misión crítica.

En una investigación realizada por Garner[3], SOA se ha utilizado en más del 50% de los desarrollos nuevos para el diseño e implementación de aplicaciones críticas, ver figura 1.2. A pesar de la aceptación de este modelo, ha presentado ciertos problemas de disponibilidad, rendimiento y flexibilidad. Por lo que se necesita explorar otros modelos que cumplan con estas necesidades.



*Figura 1.2 Gartner Hype Cycle*

Como lo indica esta referencia [14] y las propuestas mostradas anteriormente:

*“El monitoreo SOA se aborda en varias investigaciones pero aún tiene retos en las técnicas utilizadas para este fin. Esos retos difieren de acuerdo a la naturaleza de la aplicación. Además, no existe una arquitectura o modelo a seguir que pueda cubrir todos los escenarios para el monitoreo de este tipo de aplicaciones.”*

### 1.3. Objetivos

El objetivo general de esta tesis es diseñar e implementar un sistema gestor para aplicaciones de misión crítica que utilizan la arquitectura ADSOA, que permita conocer el estado actual de los procesos y servicios de negocio.

Los objetivos específicos del proyecto son los siguientes:

- Diseñar la arquitectura a utilizar para el sistema gestor.
- Diseñar e implementar las tecnologías que permitan conocer el estado actual de los procesos y servicios.
- Diseñar e implementar las tecnologías a utilizar para la detección de fallos en los procesos y la visualización de los mismos.

### 1.4. Justificación

Empresas pequeñas y medianas están en busca de soluciones de monitoreo para sus procesos de negocio basadas en servicios SOA [5]. Este tipo de compañías tienen sistemas distribuidos basados en servicios simples, lo cual exige tener un monitoreo que les permita saber si existe algún problema con la ejecución de alguno de sus servicios o saber con anticipación si existe algún problema con el rendimiento de los mismos, lo cual actualmente no es fácil de diagnosticar.

En las investigaciones realizadas, se encontraron arquitecturas y propuestas para la gestión de recursos, métodos y sistemas, pero en general no cumplen en su totalidad con las propiedades de la **confiabilidad** para sistemas críticos.

Como ya se mencionó, el uso de aplicaciones de misión crítica ha ido en aumento, la importancia en tener un sistema **confiable** es esencial. La arquitectura ADSOA está diseñada para cumplir con las exigencias de este tipo de sistemas.

Se llegó a la conclusión de que era imperativo tener un mecanismo de gestión para sistemas de misión crítica que utilizan la arquitectura ADSOA, ya que no existe actualmente alguna propuesta para este fin. Las ventajas que se pueden obtener con la implementación de esta propuesta, se verán reflejadas en la reducción en el tiempo de respuesta en caso de presentarse algún fallo que ponga en riesgo los procesos del negocio y en la prevención del impacto que pueda producirse si estos incidentes no llegan a ser evitados o controlados.

## **1.5. Aportación**

La implementación de un sistema gestor de aplicaciones de misión crítica con base en la arquitectura ADSOA, el diseño de las tecnologías necesarias que permita conocer el estado de cada proceso y servicio de negocio del sistema, además de un mecanismo que permita saber si existe algún fallo con los procesos y servicios mencionados.

## **1.6. Metodología**

La metodología para diseñar e implementar un sistema gestor para aplicaciones de misión crítica que utilizan una arquitectura ADSOA, está conformada por cuatro fases las cuales tienen una serie de actividades a seguir y que deben cumplirse de manera secuencial. Estas fases se describen a continuación:



### **1.6.1.FASE 1. Analizar las Distintas Tecnologías Utilizadas en la Arquitectura ADSOA.**

Esta fase tiene como objetivo principal el análisis de los diferentes protocolos que conforman esta arquitectura, así como de los componentes que la integran.

- Paso 1. Comprender los conceptos y la arquitectura de ADS, SOA y ADSOA.
- Paso 2. Estudio y análisis de los protocolos de comunicación.
- Paso 3. Estudio y análisis del protocolo de foliado.
- Paso 4. Estudio y análisis del protocolo de auto recuperación.
- Paso 5. Estudio y análisis del prototipo existente implementado con la tecnología ADSOA.

### **1.6.2.FASE 2. Definir las Tecnologías para el Proceso de Monitoreo.**

Esta fase es esencial ya que se definen los elementos a ser monitoreados, también se especifica el diseño de la arquitectura para el proceso de monitoreo, así como el método a emplear que cumpla con esta tarea.

- Paso 6. Identificar los elementos que conforman el proceso de monitoreo.
- Paso 7. Modelar la arquitectura para el sistema gestor.

### **1.6.3.FASE 3. Diseño de las Tecnologías a Utilizar para el Proceso de Monitoreo.**

En esta fase se deberá establecer los protocolos necesarios para el monitoreo de los procesos, así como el diseño de los componentes necesarios para la tecnología que permita esta tarea.

Paso 8. Establecer el protocolo para el proceso de monitoreo.

Paso 9. Diseño de los componentes para las tecnologías definidas en la arquitectura.

### **1.6.4.FASE 4. Seleccionar el Caso de Prueba para Implementar el Diseño y la Arquitectura para el Proceso de Monitoreo.**

En esta fase, se selecciona el prototipo para implementar las tecnologías que fueron definidas en la fase anterior, se realizan las pruebas necesarias para asegurar el correcto funcionamiento del proceso de monitoreo.

Paso 10. Seleccionar el prototipo a utilizar.

Paso 11. Implementar el diseño y las tecnologías para el sistema gestor.

Paso 12. Establecer el caso de prueba.

Paso 13. Pruebas de concepto.

En la tabla 1.1 se describe el Cronograma de Actividades a seguir para la implementación del Sistema Gestor para Aplicaciones Críticas que utilizan una arquitectura ADSOA.

Nombre	Comienzo	Fin
<b>Proyecto Tesis</b>	<b>12/05/2014</b>	<b>17/10/2014</b>
<b>FASE 1 Analizar las distintas tecnologías utilizadas en la arquitectura ADSOA</b>	<b>12/05/2014</b>	<b>15/06/2014</b>
Paso 1 Comprender los conceptos y la arquitectura de ADS, SOA y ADSOA.	12/05/2014	16/05/2014
Paso 2 Estudio y análisis de los protocolos de comunicación.	17/05/2014	22/05/2014
Paso 3 Estudio y análisis de los protocolos de foliado.	23/05/2014	29/05/2014
Paso 4 Estudio y análisis del protocolo de auto recuperación.	30/05/2014	02/06/2014
Paso 5 Estudio y análisis del prototipo existente implementado con la tecnología ADSOA	03/06/2014	13/06/2014
<b>FASE 2 Definir las tecnologías para el proceso de monitoreo.</b>	<b>16/06/2014</b>	<b>06/07/2014</b>
Paso 6 Identificar los elementos que conforman el proceso de monitoreo.	16/06/2014	23/06/2014
Paso 7 Modelar la arquitectura para el sistema gestor.	24/06/2014	04/07/2014
<b>FASE 3 Diseño de las tecnologías para el proceso de monitoreo.</b>	<b>07/07/2014</b>	<b>17/08/2014</b>
Paso 8 Establecer el protocolo para el proceso de monitoreo y control.	07/07/2014	25/07/2014
Paso 9 Diseño de los componentes para las tecnologías definidas en la arquitectura.	28/07/2014	15/08/2014
<b>FASE 4 Seleccionar el caso de prueba para implementar el diseño y la arquitectura para el proceso de monitoreo.</b>	<b>18/08/2014</b>	<b>17/10/2014</b>
Paso 10 Seleccionar el prototipo a utilizar.	18/08/2014	25/08/2014
Paso 11 Implementar el diseño y las tecnologías para el sistema gestor.	26/08/2014	26/09/2014
Paso 12 Establecer el caso de prueba.	27/09/2014	06/10/2014
Paso 13 Pruebas de concepto.	07/10/2014	17/10/2014

**Tabla 1.1 Cronograma de actividades**

## 2.Marco Teórico

En este capítulo se describen las características de los Sistemas de Misión Crítica, también los conceptos de Arquitectura Orientada a Servicios (SOA), las propiedades de los Sistemas Autónomos Descentralizados (ADS), así como distintas propuestas de monitoreo y finalmente el concepto de Arquitectura Orientada a Servicios Autónomos Descentralizados (ADSOA).

### 2.1.Sistemas de Misión Crítica

#### 2.1.1. Descripción

Los fallos en los sistemas de software pueden ser considerados comunes. En la mayor parte de los casos, estos fallos pueden provocar retrasos e inconvenientes, pero no generan problemas de alto impacto. Por otro lado, en algunos sistemas un fallo puede ser causa de pérdidas económicas importantes, provocar daños físicos e incluso pérdidas humanas. A este tipo de sistemas se les conoce como Sistemas de Misión Crítica.

Los Sistemas Críticos son sistemas socio-técnicos<sup>1</sup> de los que dependen los negocios o las personas y son diseñados para ofrecer los servicios de la manera que se espera o específica, ya que de no cumplir con esto pueden provocar graves problemas y pérdidas importantes. A continuación se describen los tres tipos principales de sistemas críticos.

- **Sistemas de seguridad críticos:** Se caracterizan por el daño que pueden ocasionar al medio ambiente u ocasionar pérdidas humanas si tienen algún fallo (sistema de control para una planta de fabricación de productos químicos).

---

<sup>1</sup> Los sistemas socio-técnicos están constituidos por uno o más sistemas técnicos, incluyen personas (operadores) como parte inherente del sistema, son gobernados por políticas y reglas organizacionales y pueden ser afectados por restricciones como leyes nacionales y políticas regulatorias.

- Sistemas de misión críticos: Se caracterizan por los errores que pueden provocar en los sistemas en algunas actividades dirigidas por objetivos si ocurre un fallo de funcionamiento (sistema de navegación para una nave espacial).
- Sistemas de negocio críticos: Se caracterizan por los altos costos que pueden ocasionar al negocio al ocurrir algún fallo en el sistema (sistemas financieros o bancarios).

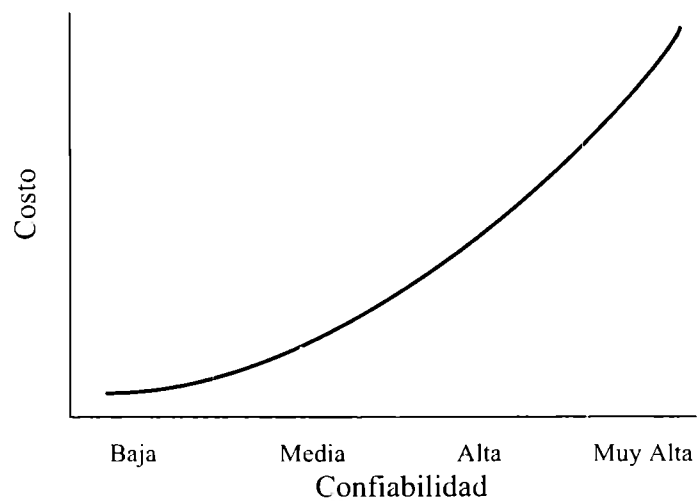
Para los sistemas críticos, la confiabilidad es la característica más importante que deben cumplir. A continuación se detallan algunos puntos para explicar la importancia de la confiabilidad en este tipo de sistemas:

- Cuando un sistema que no es confiable y además inseguro, por lo regular es rechazado por los usuarios que los utilizan, y esto se debe por la falta de confianza que le tienen.
- Por los costos que pueden ocasionar los fallos de funcionamiento del sistema. Siendo el costo por el fallo mayor en magnitud que el costo del sistema mismo.
- Cuando se tiene un sistema no confiable puede ocasionar pérdida de información importante. Esto se debe al costo que implica el mantenimiento de los datos; en algunas ocasiones es mayor que el sistema de software que los procesa.

Como se mencionó anteriormente, el costo por un fallo en el funcionamiento en los sistemas críticos es elevado, por lo cual se deben utilizar métodos y técnicas confiables para su desarrollo.

Los métodos matemáticos formales<sup>2</sup> de desarrollo de software son utilizados con éxito para este tipo de sistema, además de obtener una reducción en el número de pruebas requeridas para validarlos.

Un aspecto importante en los sistemas críticos, es que los costos de verificación y validación son muy caros, - aproximadamente del 50% del costo del desarrollo del sistema [1]. Cuanto mayor sea la confiabilidad que se necesita, se tendrá que gastar más en probar y verificar que se alcanzaron los niveles deseados, tal y como se muestra en la figura 2.1



**Figura 2.1 Curva costo/confiabilidad**

Por lo tanto, se debe tener atención en los siguientes componentes del sistema que son los más susceptibles a generar algún fallo importante:

---

<sup>2</sup> Los métodos formales se usan para referirse a cualquier actividad relacionada con las representaciones matemáticas del software, incluyendo la especificación formal de sistema, análisis y demostración de la especificación, el desarrollo transformacional y la verificación de programas.

- Hardware del sistema. Puede fallar por errores de fabricación, de diseño o debido a que sus componentes cumplieron con su vida útil.
- Software del sistema. Puede fallar por errores de diseño, especificación o implementación.
- Operadores del sistema. Pueden ocasionar fallos debido al uso incorrecto del sistema.

Estos fallos, defectos y errores pueden ser clasificados para su mayor control, como se puede observar la Tabla 2.1. Además se definen tres conceptos que permiten evitarlos lo mejor posible y así obtener una buena fiabilidad en el sistema.

<b>Tipo</b>	<b>Descripción</b>
Fallo del sistema	Momento en el sistema cuando no funciona como esperan los usuarios.
Error del sistema	Comportamiento inesperado del sistema, lo que produce un estado erróneo.
Defecto del sistema	Provoca que un sistema de software no cumpla de manera completa y efectiva aquello para lo que fue creado.
Error humano	Acción del usuario que trae como consecuencia defectos en el sistema.

***Tabla 2.1 Terminología de fiabilidad.***

Para evitar los defectos, se hace uso de técnicas que permiten evitar la intrusión de defectos en el sistema, con la intención de minimizarlos. Un buen ejemplo, es trabajar con lenguajes de programación que no sean propensos a errores, que no utilicen punteros.

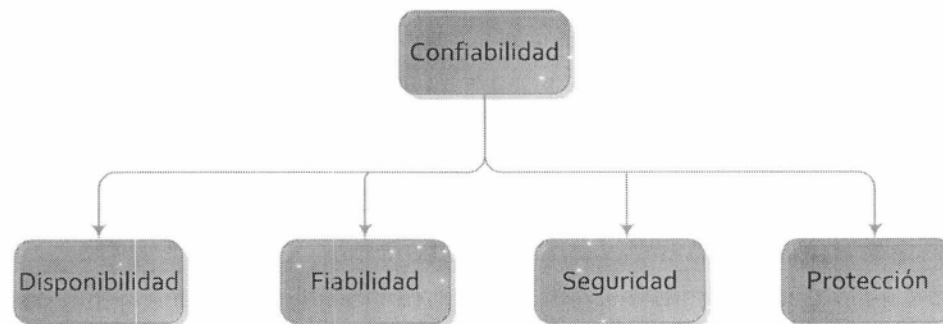
Para eliminar los defectos, se pueden usar técnicas de verificación y validación para detectarlos y eliminarlos antes de hacer usos del sistema. Un ejemplo son los tipos de pruebas a realizar en un sistema de software (límites, estrés, etc.)

Con la tolerancia a defectos, se intenta asegurar que los errores en el sistema no causen fallos en el funcionamiento del mismo.

Con base en estos aspectos, es importante que el diseño de cualquier sistema crítico se implemente de manera separada, es decir, componentes separados sin tener en cuenta los puntos vulnerables de cada uno de estos. De esta forma los errores solo ocurrirán en las interfaces de los componentes del sistema, lo que permite un mayor control.

### 2.1.2. Propiedades

Como ya se mencionó, la confiabilidad es la característica más importante con la que debe contar un sistema crítico. Esta característica se divide en cuatro propiedades principales, como se muestra en la figura 2.2.



*Figura 2.2 Propiedades de la confiabilidad*



A continuación se describen cada una de estas:

La **disponibilidad** es la probabilidad de que el sistema esté activo y en funcionamiento proporcionando los servicios que ofrece en cualquier momento.

La **fiabilidad** de un sistema, es la probabilidad de que un sistema funcione correctamente como lo espera el usuario, durante un período determinado. Esta propiedad está integrada además por otras características, las cuales se presentan en seguida.

- **Corrección:** se encarga de asegurar que los servicios que se ofrecen por el sistema, tengan las especificaciones correctas.
- **Precisión:** se encarga de asegurar el nivel de detalle de la información sea el óptimo al ser entregada al usuario.

La **seguridad** es la valoración de la probabilidad de que el sistema cause algún daño a las personas o a su entorno.

La **protección** es la valoración de la probabilidad de que el sistema pueda resistir intrusiones. Esta propiedad está integrada además de otras características, que se muestran a continuación:

- **Integridad:** se encarga de asegurar la información que los programas utilizados en un sistema no sufran algún daño.
- **Confidencialidad:** asegura que solo el personal autorizado pueda hacer uso del sistema.

Es importante mencionar que además de estas cuatro dimensiones que componen la confiabilidad, existen otras propiedades que la complementan, las cuales se explican a continuación:

- **Reparabilidad:** es la capacidad de diagnosticar un fallo producido en el sistema, acceder al componente que lo produce y realizar los cambios necesarios en el mismo. Esto permite minimizar la interrupción en los procesos del sistema.
- **Mantenibilidad:** es la capacidad de adaptación del sistema a nuevos requerimientos sin producir errores al realizar los cambios correspondientes.
- **Supervivencia:** es la capacidad de un sistema para dar continuidad en los servicios que ofrece al estar siendo atacado y principalmente cuando parte del sistema no está en funcionamiento.
- **Tolerancia a errores:** es la capacidad del sistema en repararse automáticamente, al detectar un error por los datos introducidos por el usuario.

En conclusión, todas estas propiedades son utilizadas de manera indistinta en la integración de sistemas críticos, no siempre se implementa cada una de estas, esto depende del entorno y del propósito del mismo.

## **2.2.ADS (Sistemas Autónomos Descentralizados)**

### **2.2.1. Conceptos**

Un sistema autónomo descentralizado (ADS) es un sistema conformado por módulos o componentes que están diseñados para funcionar de manera independiente, pero son capaces de interactuar entre sí para alcanzar el objetivo global del sistema (analogía de seres vivos). Este paradigma le permite al sistema seguir funcionando en caso de algún fallo en los componentes. Otra característica que tiene son el mantenimiento y reparación en línea [8].

Como ya se mencionó, ADS es una opción que proporciona tolerancia a fallos, así como expansión y mantenimiento en línea [15]. Para que un sistema implementado con ADS pueda cumplir con estas propiedades, cada subsistema que lo compone deben cumplir con las siguientes características:

Control autónomo. Si cualquier subsistema llega a fallar, está en reparación o está siendo agregado, los demás subsistemas deben seguir funcionando sin interrumpir sus responsabilidades.

Coordinación autónoma (subsistema autónomo). Si cualquier subsistema llega a fallar, está en reparación o está siendo agregado, los otros subsistemas pueden seguir coordinando sus objetivos individuales entre sí y así continuar operando de manera normal.

Se sugiere que cada subsistema autónomo debe tener cierta inteligencia para administrarse por sí mismo sin dirigir o ser dirigido por otros subsistemas. Para realizar un sistema autónomo descentralizado con auto controlabilidad y coordinabilidad, cada uno de sus subsistemas tienen que satisfacer las siguientes condiciones:

Es importante mencionar que además de estas cuatro dimensiones que componen la confiabilidad, existen otras propiedades que la complementan, las cuales se explican a continuación:

- **Reparabilidad:** es la capacidad de diagnosticar un fallo producido en el sistema, acceder al componente que lo produce y realizar los cambios necesarios en el mismo. Esto permite minimizar la interrupción en los procesos del sistema.
- **Mantenibilidad:** es la capacidad de adaptación del sistema a nuevos requerimientos sin producir errores al realizar los cambios correspondientes.
- **Supervivencia:** es la capacidad de un sistema para dar continuidad en los servicios que ofrece al estar siendo atacado y principalmente cuando parte del sistema no está en funcionamiento.
- **Tolerancia a errores:** es la capacidad del sistema en repararse automáticamente, al detectar un error por los datos introducidos por el usuario.

En conclusión, todas estas propiedades son utilizadas de manera indistinta en la integración de sistemas críticos, no siempre se implementa cada una de estas, esto depende del entorno y del propósito del mismo.

## **2.2.ADS (Sistemas Autónomos Descentralizados)**

### **2.2.1. Conceptos**

Un sistema autónomo descentralizado (ADS) es un sistema conformado por módulos o componentes que están diseñados para funcionar de manera independiente, pero son capaces de interactuar entre sí para alcanzar el objetivo global del sistema (analogía de seres vivos). Este paradigma le permite al sistema seguir funcionando en caso de algún fallo en los componentes. Otra característica que tiene son el mantenimiento y reparación en línea [8].

Como ya se mencionó, ADS es una opción que proporciona tolerancia a fallos, así como expansión y mantenimiento en línea [15]. Para que un sistema implementado con ADS pueda cumplir con estas propiedades, cada subsistema que lo compone deben cumplir con las siguientes características:

Control autónomo. Si cualquier subsistema llega a fallar, está en reparación o está siendo agregado, los demás subsistemas deben seguir funcionando sin interrumpir sus responsabilidades.

Coordinación autónoma (subsistema autónomo). Si cualquier subsistema llega a fallar, está en reparación o está siendo agregado, los otros subsistemas pueden seguir coordinando sus objetivos individuales entre sí y así continuar operando de manera normal.

Se sugiere que cada subsistema autónomo debe tener cierta inteligencia para administrarse por sí mismo sin dirigir o ser dirigido por otros subsistemas. Para realizar un sistema autónomo descentralizado con auto controlabilidad y coordinabilidad, cada uno de sus subsistemas tienen que satisfacer las siguientes condiciones:

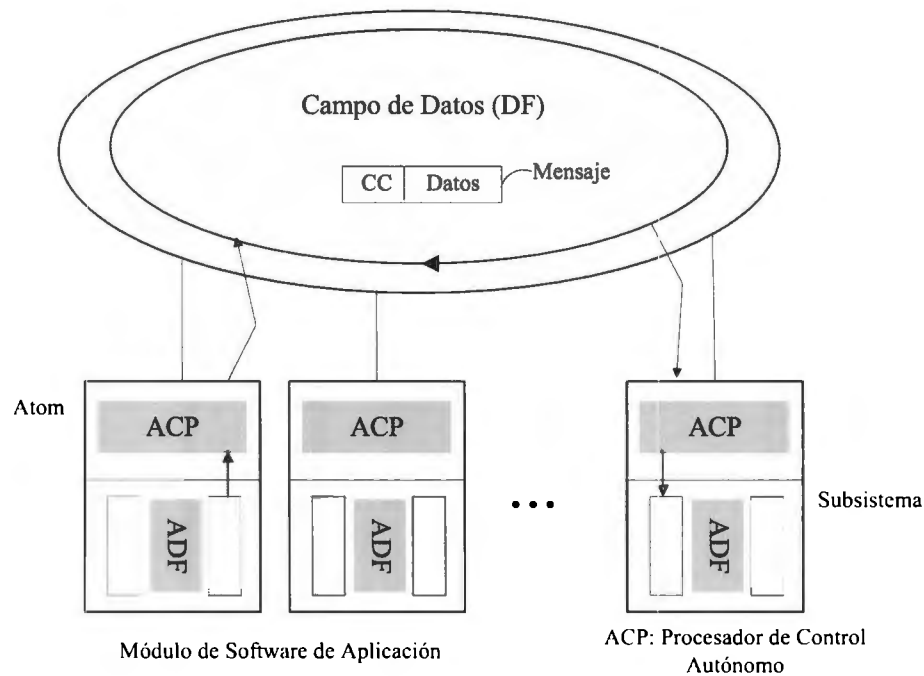
- **Igualdad:** se refiere a que cada subsistema debe ser capaz de gestionarse a sí mismo, sin necesidad que ser dirigido por otro (no existe la relación maestro-esclavo).
- **Localidad:** cada subsistema es capaz de administrarse a sí mismo para coordinarse con los demás, con base en su información local.
- **Autonomía:** cada subsistema es autónomo en funciones al administrarse a sí mismo y al tener la capacidad de coordinarse con los demás subsistemas.

Al conseguir estas características, se asegura que cualquier subsistema del sistema global, si llegara a fallar, los demás subsistemas pueden seguir trabajando y operando sin ser afectados por este incidente, lo que permite una operación continua.

## 2.2.2. Arquitectura

Como ya se mencionó, los sistemas que cumplan con las características de localidad, igualdad y autonomía, pueden ser implementados bajo una arquitectura ADS, donde no se permite el uso de operaciones centralizadas y de sistemas de coordinación. Con esta arquitectura, cada subsistema tiene su propio sistema de gestión conocido como **Procesador de Control Autónomo (ACP)**, el cual se encarga de la administración y la comunicación con otros ACP's.

Adicionalmente, cada subsistema tiene módulos de aplicación de software, y en conjunto con su ACP conforman una unidad autónoma llamada *Atom* [14], observar figura 2.4.

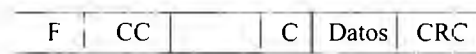


**Figura 2.3 Arquitectura ADS**

En esta arquitectura, todos los subsistemas están conectados mediante una red llamada **Campo de Datos (DF)**, por esta red los mensajes son transmitidos de forma lógica viajando a través de esta [15]. Los mensajes son recibidos por cada *Atom* conectado al DF, los datos se mueven a través de los módulos de aplicación con ayuda de un componente llamado **Atom Data Fiel (ADF)**.

Cada mensaje tiene un formato, como se muestra en la figura 2.5, donde los datos siempre van acompañados de su **Código de Contenido (CC)**, el cual es definido de manera única con respecto al contenido de los datos. Al llegar un mensaje a cualquier subsistema o Atom, decide tomarlo o no con base en la información proporcionada por el CC.

La comunicación mediante el Código de Contenido permite que cada subsistema no necesite conocer la relación entre el origen y el destino, donde solo se toman y procesan los mensajes que necesita y conoce, desechando los demás.



F: Bandera

CC: Código de Contenido

C: Código de Control

CRC: Comprobación de Redundancia Ciclica

*Figura 2.4 Formato de mensaje incluido el código de contenido*

## 2.3.SOA (Arquitectura Orientadas a Servicios)

### 2.3.1. Conceptos

SOA es una arquitectura de TI que ayuda en la transformación de los procesos de negocio en un conjunto de servicios vinculados, los cuales están disponibles en una red informática ya sea local o remota (pueden estar en diferentes zonas geográficas). Esta arquitectura también permite la integración de distintas tecnologías, lo cual es una gran ventaja ya que el negocio puede ser adaptado eficazmente a los cambios en las condiciones del entorno o a nuevos requerimientos.

IBM definió cinco puntos de entrada importantes con base en las expectativas y experiencia de sus clientes [9]. Estos puntos son utilizados de acuerdo a las necesidades presentadas, ya sean de negocio (personas, procesos y entradas de información) o de TI (conectividad o reúso de servicios). A continuación se describen cada uno de estos:



- **Personas:** este punto de entrada busca mejorar la productividad del negocio, con base en la experiencia de los usuarios para generar innovación y una mayor colaboración, lo que permite una consistente interacción entre las personas y los procesos.
- **Procesos:** este punto de entrada permite mejorar los procesos de negocio existente en una empresa, al conocer y entender lo que ocurre en su negocio. Al implementar SOA los procesos de negocio se transforman en servicios flexibles y reutilizables, además de permitir la optimización de los nuevos procesos a utilizar.
- **Información:** con este punto de entrada, la información se aprovecha y proporciona de una manera coherente y fiable. Al lograr proporcionar la información con estas características en todas las áreas de negocio dentro de una empresa, se logra generar innovación entre estas, y así poder competir de manera más eficaz.
- **Conectividad:** la esencia de este punto de entrada es crear una infraestructura de conexión efectiva que permita integrar a las personas, los procesos y la información. Al contar con una conexión flexible con SOA entre los servicios y el entorno de negocio, se puede enviar un proceso existente a través de una canal de negocio distinto, con diferente infraestructura y ser entregado sin problemas.
- **Reutilizar Servicios:** al reutilizar los servicios existentes, se logra optimizar los procesos de negocio, reduciendo la duplicidad de la funcionalidad de los servicios, y así obtener ahorro en tiempo y dinero.

### **2.3.2. Arquitectura**

Como se describió, SOA puede ser percibido de acuerdo con la funcionalidad que se necesite en cada negocio. Para un mayor entendimiento, a continuación se hace una definición de la Arquitectura Orientada a Servicios.

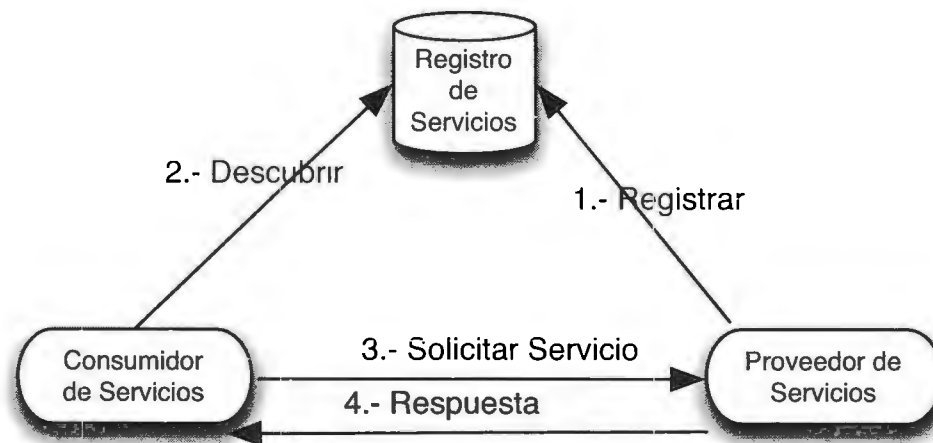
**Servicio:** es una tarea de negocio repetible y es utilizada para ejecutar una actividad. El detalle de su implementación debe ser independiente de los demás servicios ya que no deben existir dependencias entre estos.

**Orientación al Servicio:** mecanismo que utilizan los servicios que están relacionados e integrados para generar un resultado.

**Arquitectura Orientada a Servicios:** es una arquitectura que se basa en la orientación al servicio. Desde un enfoque de negocio, SOA es un mecanismo mediante el cual los servicios de negocio son usados e integrados dentro de la empresa o fuera de ella por aplicaciones empresariales.

SOA visto desde un nivel de implantación, es una arquitectura que utiliza estándares basados en infraestructura, modelos de programación y protocolos web. Por otro lado, en una perspectiva operativa, SOA se define como una serie de acuerdos entre los servicios a proveer y los consumidores de los servicios.

Esta arquitectura está conformada por tres roles: Un proveedor de servicios, un consumidor y un registro de servicios. La figura 2.3 describe la arquitectura SOA.



**Figura 2.5 Arquitectura SOA**

Proveedor de servicios: es el encargado de proporcionar tanto el servicio como el detalle del mismo. También el proveedor indica si el servicio puede ser usado por cualquier persona o no, además de proporcionar una interfaz para tener acceso a dicho servicio.

Registro de servicios: es el encargado de registrar los servicios proporcionados por el proveedor de servicios.

Consumidor de servicios: invoca los servicios definidos por el proveedor, asegurándose que puede localizarlos en el registro de servicios para que así genere una solicitud de servicio al proveedor de servicios.

La secuencia de acciones al solicitar un servicio se muestra a continuación:

1. El proveedor de servicios registra sus servicios en el registro de servicios.
2. Un consumidor de servicios consulta los servicios existentes en el registro de servicios.

3. El consumidor de servicios envía una solicitud de servicio al proveedor que tienen el servicio de su interés.
4. Finalmente el proveedor de servicios recibe y atiende la solicitud para enviar la petición al consumidor que lo requirió.

Los servicios en SOA tienen varias características pero las más importantes se definen como: **reutilización, bajo acoplamiento, autónomos y no tienen un estado.**

Para cumplir con el proceso de solicitud y entrega, SOA necesita de los siguientes elementos:

1. Un medio de comunicación que permita el flujo de los mensajes.
2. Un protocolo de comunicación.
3. La descripción de los servicios que indique las reglas.
4. Un conjunto de servicios disponibles.
5. Los procesos de negocio (reglas de los requerimientos del negocio).
6. Un registro del servicio.

En esta arquitectura se hace uso de protocolos estándar e interfaces que permiten el acceso a la lógica de negocio. Como ya se mencionó una característica de esta arquitectura es la reutilización de los servicios entre los distintos procesos de negocio, lo que puede producir la siguiente problemática [2]:

- Bajo desempeño y baja disponibilidad al aumentar la demanda.
- Los procesos se detienen cuando existe una falla en el servicio.

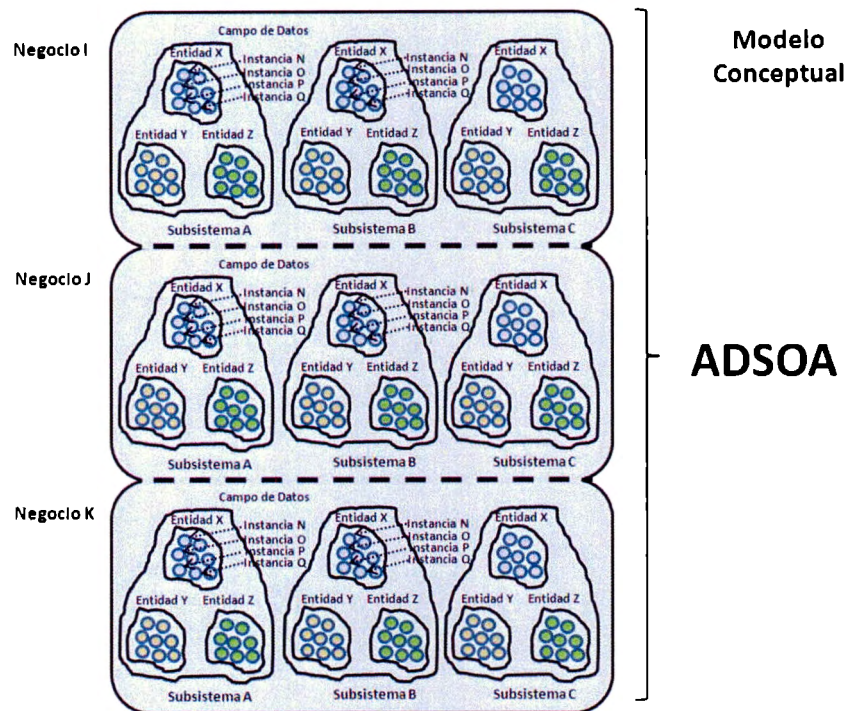
A pesar de su aceptación este modelo, ha presentado ciertos problemas de disponibilidad, rendimiento y flexibilidad [2]. Por lo que se necesita explorar otros modelos que cumplan con estas necesidades lo mejor posible.

## **2.4. ADSOA (Arquitectura Orientada a Servicios Autónoma Descentralizada)**

### **2.4.1. Conceptos**

Actualmente existe una novedosa propuesta que cumple con todas las necesidades en el diseño e implementación de sistemas críticos, combinando las características de ADS y SOA [2]. Esta arquitectura es llamada ADSOA (Arquitectura Orientada a Servicios Autónoma y Descentralizada).

Con ADSOA, cada **Negocio** está conformado por **Subsistemas**, los cuales a su vez están constituidos por un conjunto de **Entidades Autónomas** que tienen la característica de ser capaces de solicitar y ofrecer servicios mediante mensajes, ver figura 2.6.



**Figura 2.6 Modelo conceptual ADSOA**

Cada entidad se identifica por tres elementos:

1. Id Negocio
2. Id Subsistema
3. Id Entidad

Las entidades pueden tener varias **Instancias Independientes**, con la característica de tener la misma funcionalidad que la entidad que representan, lo que permite tener tolerancia a fallos y alta disponibilidad. ADSOA al igual que ADS, utiliza el Campo de Datos (DF) como canal de comunicación entre cada negocio, subsistemas y entidades, además de ser el medio por donde se transmiten los mensajes. Cada entidad, así como sus instancias deben de ser capaces de realizar las mismas tareas para las que fueron diseñadas, independientemente del orden en que reciben los mensajes.

### **2.4.2. Atom**

Dentro de la Arquitectura ADSOA, se implementó un componente que funge como la capa intermedia que realiza las funciones de comunicación entre cada entidad conectada al campo de datos. A este elemento se le conoce como *Atom*. Esta capa ofrece servicios de acceso al campo de datos, de protocolo de foliado y de servicios (necesarios para la alta disponibilidad).

### **2.4.3. Arquitectura**

Como ya se mencionó, ADSOA está constituida de entidades, estas entidades pueden tener varias instancias conectadas simultáneamente a un campo de datos ya que cumplen con la característica de autonomía. Este DF está conformado por procesos interconectados entre si y las entidades están conectadas a estos procesos para solicitar u ofrecer servicios mediante mensajes, ver figura 2.7.

Cada entidad tiene un identificador único, el cual es compartido entre cada una de sus instancias [2]. Para que una instancia pueda conectarse a la DF, debe implementar un protocolo de autenticación que se describe en los siguientes pasos:

1. Una instancia de una entidad solicita autenticarse al campo de datos.
2. El campo de datos reenvía un mensaje con un reto de conexión para la instancia solicitante.
3. La instancia responde el reto mediante otro mensaje.
4. Si el reto es contestado de manera correcta la conexión se realiza de lo contrario es rechazada.

Las instancias al estar conectadas al DF, ya están listas para recibir mensajes. Estos mensajes pueden llegar en un orden distinto al esperado o incluso podría no llegar alguno de estos, lo que puede ocasionar pérdida de información. Para evitar esta situación, ADSOA propone una tecnología de sincronización y transacción de entrega de bajo acoplamiento.

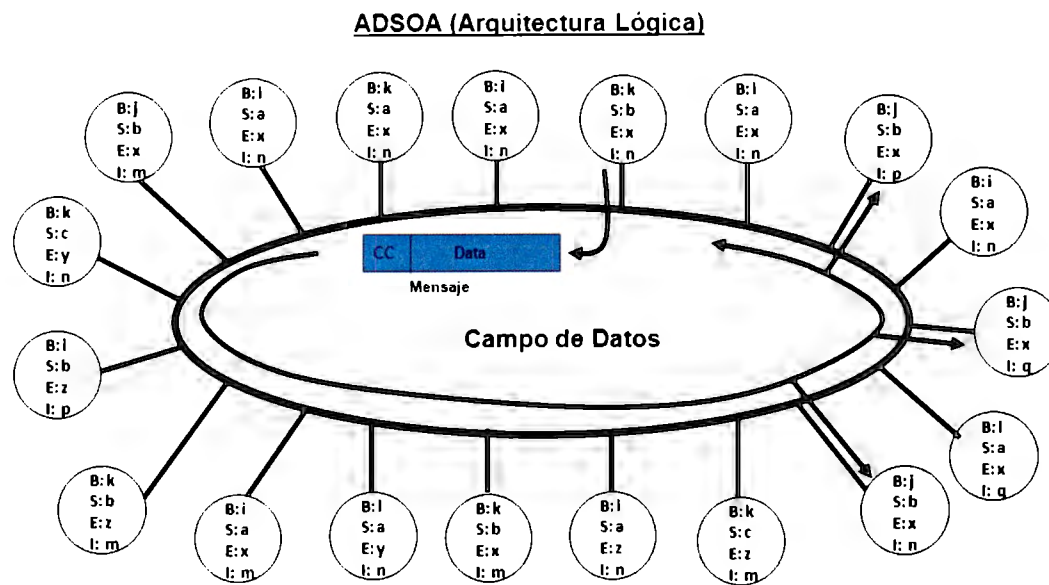
Esta tecnología, define el concepto de **Transacción** al proceso de solicitud de servicio realizado por una entidad a otra, donde la entidad solicitante necesita conocer si su mensaje fue recibido por la entidad receptora. La solicitud se considera entregada hasta que la entidad solicitante recibe un acuse de recibo por parte de la entidad receptora.

También se utiliza el concepto de **Secuencialidad**<sup>3</sup>, que hace referencia al número mínimo de acuses que debe recibir una entidad solicitante por parte de las entidades receptoras antes de que pueda mandar la siguiente solicitud de servicio.

---

<sup>3</sup> El número mínimo de acuses de recibo es configurable para cada entidad. Se recomienda que el número mínimo de acuses sean 2 para evitar la pérdida de información.



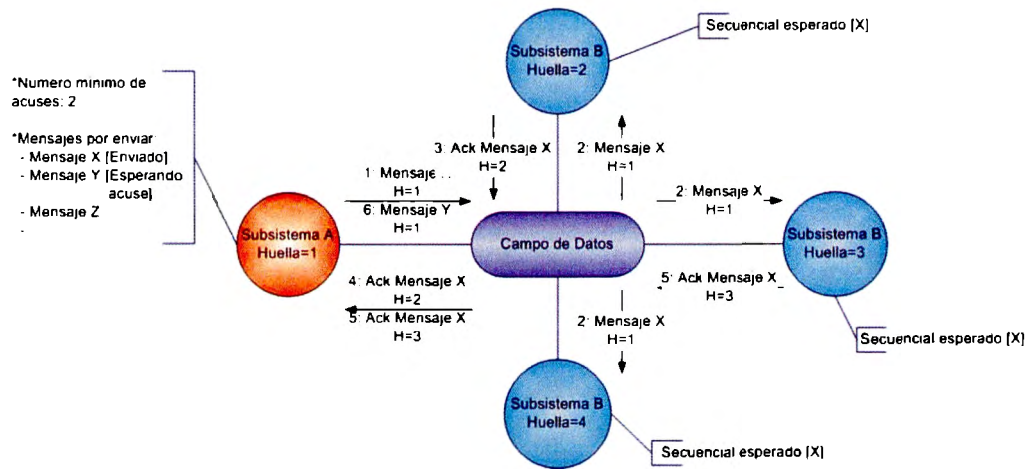


**Figura 2.7 Arquitectura ADSOA**

La estructura de la información de la solicitud de servicio está organizada por los siguientes elementos:

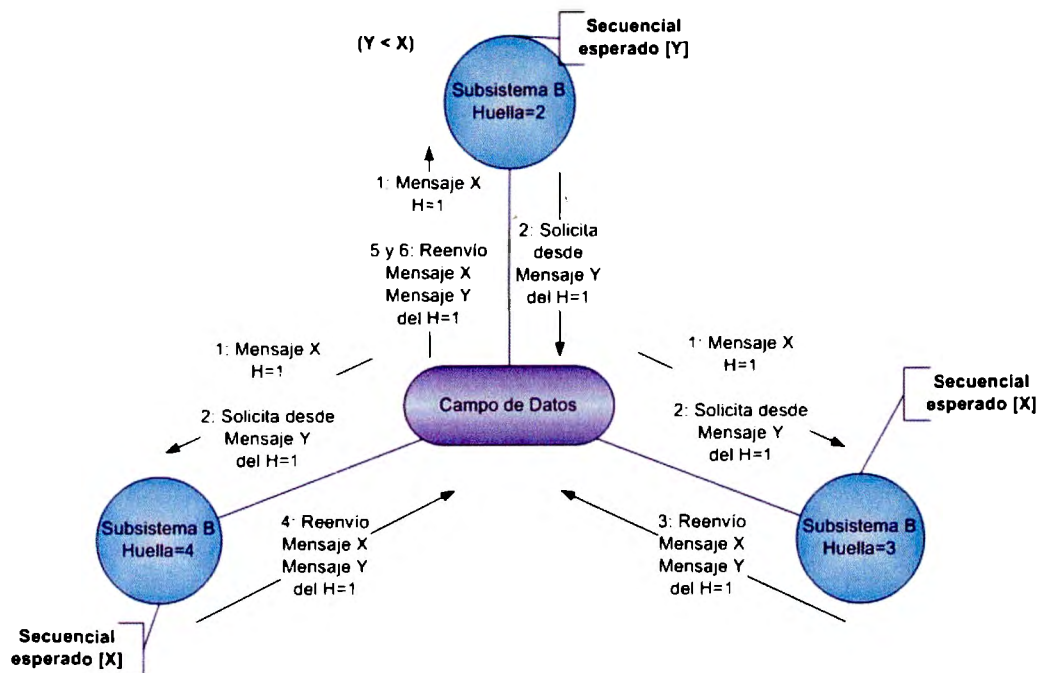
1. Información de la solicitud.
2. Código de Contenido (CC) que especifica el contenido y define el servicio solicitado.
3. Estructura de Foliado que se encarga de identificar la transacción, con base en:
  - Id del solicitante
  - Id de la tarea especializada que se solicita (Pivote)
  - Número de secuencial
  - Número de evento que es el identificador generado por la relación de la información original de la solicitud.
  - Id de la instancia relacionada con una entidad (huella).

Con esta estructura se asegura la secuencialidad de las múltiples solicitudes, además de garantizar la identificación de la cantidad de acuses recibidos por la entidad, como se muestra en la figura 2.8.



**Figura 2.8 Secuencialidad y transaccionalidad**

Cuando una instancia recibe un mensaje con una solicitud de servicio con un número de secuencia mayor al que espera, con el principio de secuencialidad tiene el conocimiento de que otra instancia de su entidad tiene los mensajes que le faltan, así que manda una solicitud de sincronización para que le manden los mensajes que le hacen falta y así contar con la misma información que las demás instancias [2], ver figura 2.9.



**Figura 2.9 Sincronización con otras Instancias**

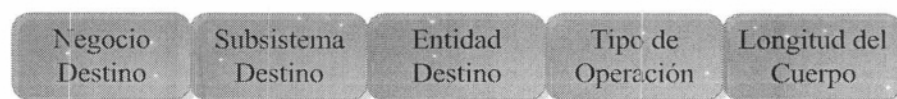
Otro punto importante, es si una entidad recibe varias veces una solicitud de servicio, es capaz de discernir mediante la huella del mensaje si esta solicitud proviene de la misma instancia solicitante o una instancia diferente a la misma entidad. Con esto la entidad receptora puede determinar si todas las solicitudes recibidas cumplen con el número mínimo de solicitudes requeridas.

En esta arquitectura, se utilizan dos tipos de mensajes, transaccionales y no transaccionales. La diferencia entre éstos es que en los mensajes transaccionales siempre se asegura la entrega del mismo a su destino, mientras que los no transaccionales no requieren asegurar esa entrega.

La estructura de estos mensajes se divide en dos partes, encabezado y cuerpo. El encabezado contiene la información del destinatario, mientras que el cuerpo contiene la estructura de la información del mensaje. A continuación se describe cada uno de los elementos que los conforman.

En la figura 2.10 se muestra la estructura del encabezado del mensaje, donde:

- **Negocio Destino.** Es el identificador del negocio que representa el sistema global del destinatario.
- **Subsistema Destino.** Es el identificador del grupo o subsistema dentro del sistema global del destinatario.
- **Entidad Destino.** Es el identificador que representa a una entidad o elemento de un grupo (o subsistema) del sistema global.
- **Tipo de Operación.** Es el identificador para el tipo de operación a procesar en el campo de datos.
- **Longitud del Cuerpo.** Representa la longitud en bytes del cuerpo del mensaje.

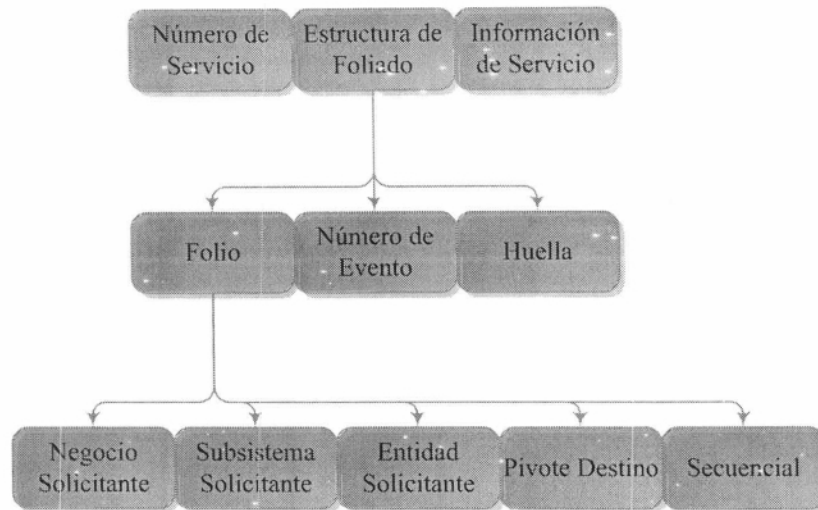


**Figura 2.10 Encabezado del mensaje**

Por otro lado, el cuerpo del mensaje se conforma de varios elementos como se muestra en la figura 2.11.

- **Número Servicio.** Se refiere al servicio que se solicita. Este servicio debe de estar implementado en la tabla de servicios que ofrece la entidad a la que se enviará el mensaje.

- Estructura Foliado. Es una estructura con varios datos relacionados con quien emite el mensaje y, en caso de ser transaccional, algunos elementos que indican la secuencia de los mensajes.
- Folio. Permite garantizar la entrega y secuencialidad de los mensajes
- Número de Evento. Permite identificar una operación. Este número de evento se obtiene mediante la combinación de campos con una función hash. Identificar la relación entre una solicitud y la operación que la desencadenó.
- Huella. Identifica a cada instancia de una entidad de manera única.
- Negocio Solicitante. Es el identificador del negocio que representa el sistema global del solicitante.
- Subsistema Solicitante. Es el identificador del grupo o subsistema dentro del sistema global del solicitante.
- Entidad Solicitante. Es el identificador que representa a una entidad o elemento de un grupo (o subsistema) del sistema global.
- Pivote Destino. Es el identificador que indica la especialización de una entidad, además permite conocer a qué subconjunto de la entidad se dirige el mensaje.
- Secuencial. Indica el número que se le asigna a cada mensaje de forma ascendente para indicar la secuencia de los mensajes que se emitan.
- Información Servicio. En el bloque de información de servicio se incluye información que requiere para que el servicio sea atendido.



**Figura 2.11** *Cuerpo del mensaje*

## **3.Objetivos de investigación**

### **3.1.Planteamiento del Problema**

Diseñar e implementar un sistema gestor para aplicaciones de misión crítica con base en la arquitectura ADSOA, que permita el monitoreo de los procesos y servicios de negocio de los sistemas.

### **3.2.Requerimientos Funcionales**

Los principales requerimientos funcionales que se deben cubrir con el sistema gestor para procesos y servicios son los siguientes:

- Interfaz gráfica con un despliegue rápido.
- Fácil uso para el usuario.
- Proporcionar información oportuna y veras sobre los procesos y servicios de los sistemas conectados al campo de datos.
- Persistencia de la información.
- Notificación de inconvenientes en los procesos y servicios de los sistemas.
- Aseguramiento de la entrega de los mensajes a su destino.
- Permitir el modelado de un proceso a través de los servicios que lo componen.

### **3.3.Requerimientos no Funcionales**

Los requerimientos no funcionales deben de cumplir con las características de confiabilidad y alta disponibilidad para tener una operación continua, que sea tolerante a fallos para que no detener su operación. El sistema debe tener la característica de usabilidad por el tipo de información que se maneja, otra característica es la capacidad de procesar un alto volumen de información, además de ser un sistema portable para visualizar la información en cualquier lugar.

### **3.4. Trabajos Relacionados (Monitores)**

En la investigación realizada, se consultaron varias propuestas para monitoreo de sistemas, el principal objetivo fue encontrar alguna alternativa que cumpliera con las necesidades que requieren cubrir los sistemas de misión crítica. El resultado fue el siguiente:

#### **3.4.1. Hyperic HQ**

[10] Es un sistema de administración y monitoreo de infraestructura web. Es utilizado para gestionar la complejidad de la infraestructura, agiliza las operaciones y mejora los niveles de servicio.

Hyperic HQ utiliza un agente que automáticamente verifica el software que está en ejecución y periódicamente escanea el sistema para obtener mediante métricas la información de la disponibilidad, utilización y el rendimiento del mismo. Posteriormente este agente manda esta información al servidor HQ y las almacena en la base de datos HQ.

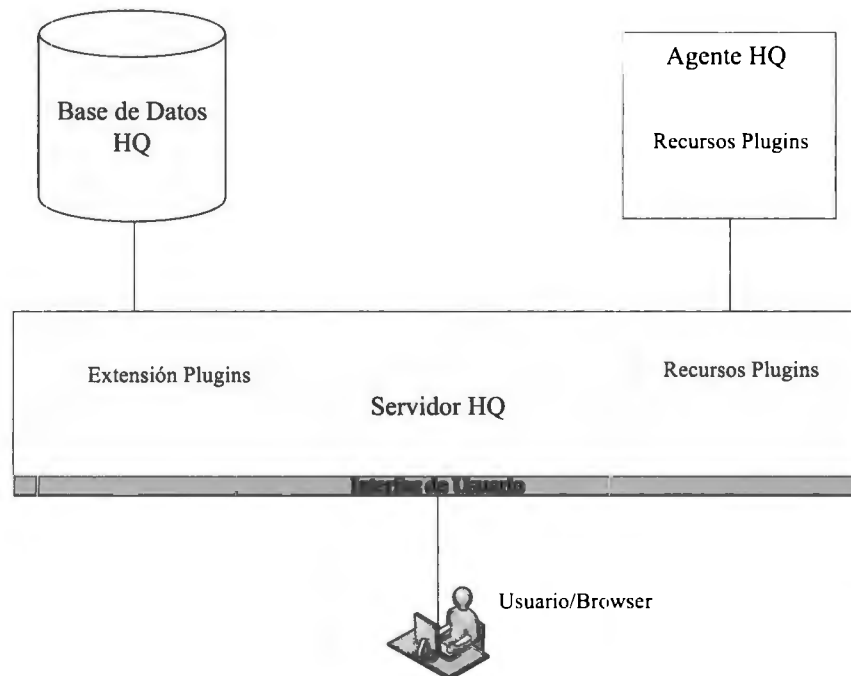
El servidor HQ agrupa el software de manera que es más sencilla su administración y control; detecta las alertas que se disparan y se encarga de realizar las notificaciones o los procesos de escalamiento que previamente fueron definidos; también procesa las acciones que son inicializadas desde una interfaz de usuario o mediante web servicios Hyperic HQ.

HQ proporciona una interfaz de usuario donde se observa el inventario de software, el estado de los recursos, las alertas recientes y las métricas en gráficas sobre los recursos. HQ ofrece un API que provee acceso programático mediante servicios web a la funcionalidad e información que contiene el servidor HQ.



Se pueden extender las capacidades de Hyperic HQ mediante dos plugins. El primero es el de recursos utilizado por el Agente HQ para monitorear, controlar y descubrir los recursos de software. El segundo es utilizado por la Interfaz de Usuario HQ para implementar scripts que ayuden en la automatización de procesos.

Estas características permiten a HQ ofrecer un buen monitoreo y control, pero debido a esto proporciona un bajo rendimiento. En la figura 3.1 se muestra la arquitectura de software de esta solución.



**Figura 3.1** *Arquitectura de software de Hyperic HQ*

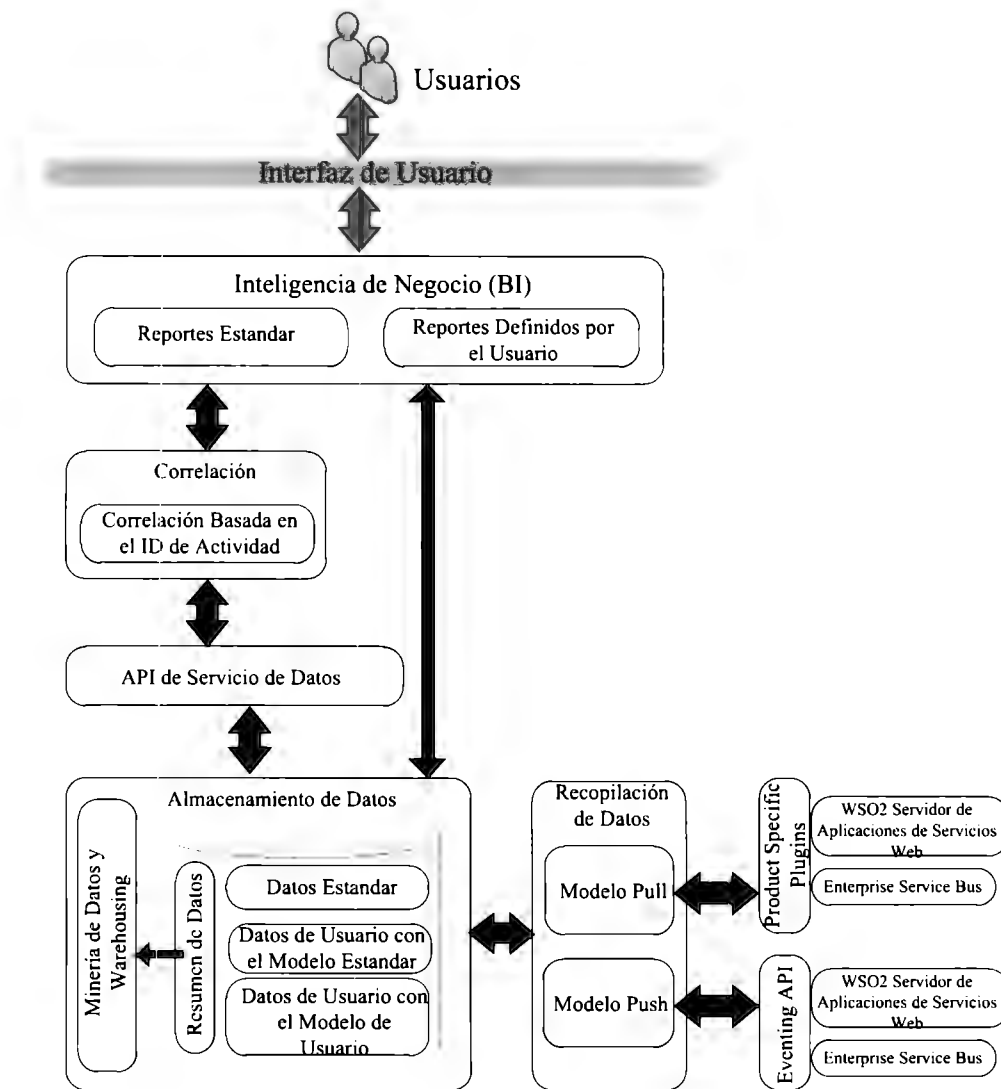
### 3.4.2. WSO2 BAM

Business Activity monitoring (BAM) se describe como un software que permite dar seguimiento a las actividades del negocio, las cuales son implementadas y ejecutadas en sistemas informáticos.

BAM es un término creado por Gartner [12], que se centra en los conceptos estratégicos de la empresa con latencia cero, a través del procesamiento directo. La referencia de latencia cero es la brecha entre el instante de obtener los datos y cuando están disponibles para la toma de decisiones, en otras palabras, los datos deben de estar disponibles de inmediato. Y el procesamiento directo se refiere a evitar usar pasos ineficientes como la entrada manual.

WSO2 BAM[11] es una plataforma para desarrollo y monitoreo de sistemas SOA dividida en módulos. El usuario interactúa con el sistema mediante una interfaz, donde se visualiza la información del negocio y las métricas proporcionadas por el servidor. También proporciona una API de servicios datos para obtener la información que se requiera de la base de datos. Es importante mencionar que la información de negocio proviene de sistemas externos.

Con BAM se pueden generar reportes sobre los estados del sistema utilizan un módulo de negocio (BI) que obtiene la información de una base de datos. Un punto importante de BAM es que no utiliza agentes, pero esto hace que el sistema sea menos flexible pero fácil de mantener. En la figura 3.2 se ilustra la arquitectura de software de WSO2 BAM.



**Figura 3.2** Arquitectura de software de WSO2 BAM

### 3.4.3. Monitoreo de Aplicaciones SOA Basado en JAVA

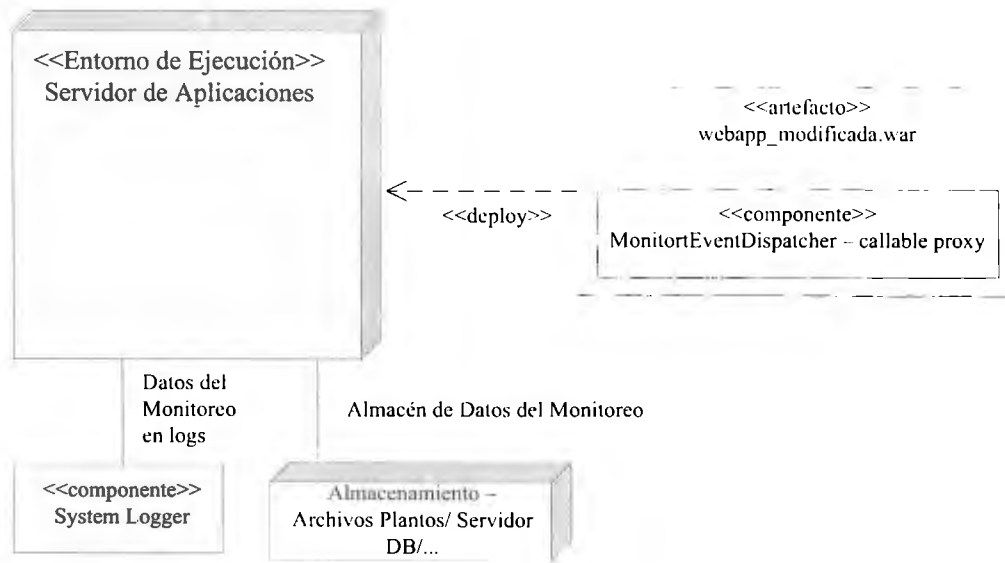
En [5] ofrece una propuesta para el monitoreo de sistemas basados en SOA dividido en componentes, su principal objetivo es la simplicidad e independencia de la plataforma que se utilice, ya que está basado en JAVA y es soportado por cualquier servidor de aplicaciones. Se pueden utilizar cualquier *framework* para implementar los servicios (Spring MVC, WS-JAX-WS, entro otros).

La gestión y control ocurre en tiempo de ejecución, después de que se hizo *deploy* a la aplicación de monitoreo, así todos los métodos del negocio públicos y protegidos pueden ser candidatos a ser monitoreados. Los casos que cubre esta propuesta son los siguientes:

- Almacenamiento estadístico de la información de negocio de los servicios en ejecución.
- Actividades de monitoreo de la aplicación modificada.
- Cuando existen modificaciones en la aplicación web.

Esta solución está dividida en dos partes. Una estática (tiempo de diseño) la cual es responsable del análisis y de las posibles modificaciones que puede sufrir la aplicación monitoreada. La segunda es activa (tiempo de ejecución) responsable del monitoreo en sí.

Para que la aplicación pueda monitorear algún servicio, se debe generar una descomposición, modificación y recomposición de las clases binarias en JAVA, esto se realiza con el *framework* ASM que en realidad es un API que provee estos servicios. Otra opción para esto es la implementación de OAP con AspectJ , se detectan los puntos de corte donde se insertaran las llamadas de los métodos a ser monitoreados. La figura 3.3 describe la parte responsable de la modificación y análisis de la aplicación a ser monitoreada.

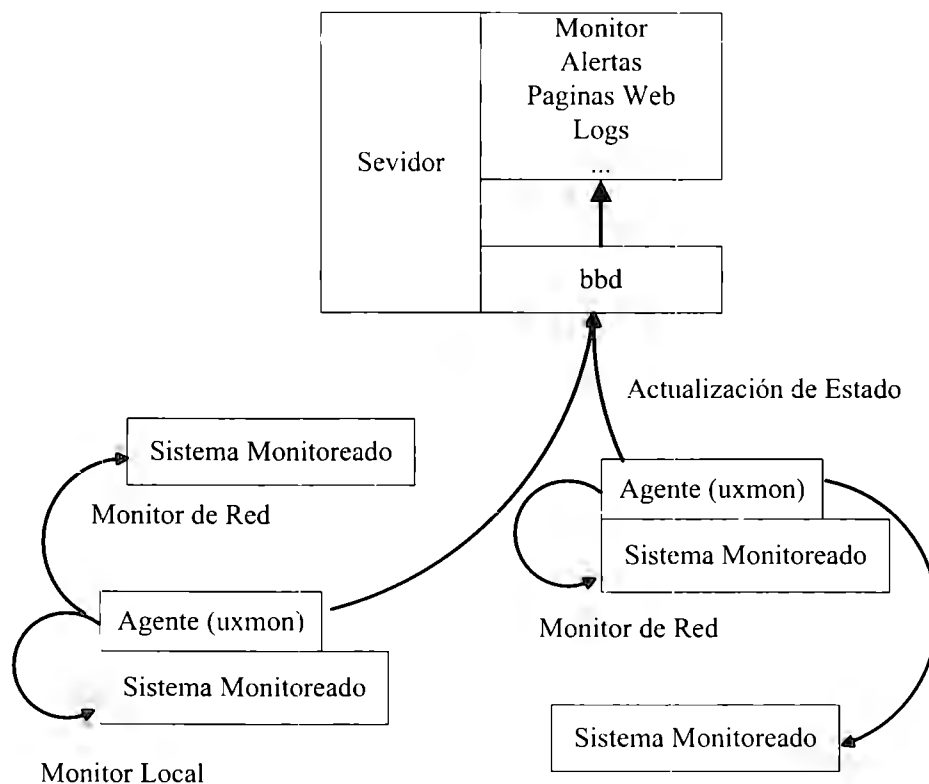


**Figura 3.3 Diagrama de despliegue de la aplicación monitoreada**

Las restricciones que se deben tomar en cuenta para esta propuesta son: la solución desarrollada solo puede monitorear los servicios implementados en JAVA (archivo WAR). El almacenamiento de la información del monitoreo se guarda en archivos planos separado por comas, en lugar de utilizar una base de datos, esto genera una desventaja porque restringe el tamaño de la información almacenada y no ofrece un monitoreo sobre la calidad de los servicios ofrecidos. Como se puede observar esta no es una solución genérica, existen siempre casos que nos son cubiertos.

#### **3.4.4. Big Sister**

Big Sister[7], es una opción de monitoreo limitado. Un entorno completo que utiliza esta herramienta está conformada por dos componentes principales, un Servidor que ejecuta el Big Sister Server y un Monitor Big Sister, ver figura 3.4.



**Figura 3.4 Componentes Big Sister**

Big Sister se basa en el uso de agentes. Estos agentes pueden estar colocados en distintos lugares de una red de acuerdo a las necesidades de monitoreo, por ejemplo, pueden existir agentes instalados en diferentes subredes encargados de verificar tanto los recursos de la subred como el estado de los procesos en ejecución.

Para cada agente se configuran métricas, las cuales ayudan en la recopilación de la información de los estados de los procesos y recursos. Esta información se envía al Servidor Big Sister que es el encargado de almacenarla en la base de datos, procesarla y mandarla al Monitor Big Sister, el cual mediante una interfaz de usuario despliega toda esta información. Es posible también definir los niveles de alertas de acuerdo al impacto que se defina por cada métrica existente.

Las métricas configuradas en cada agente permiten monitorear los siguientes aspectos: el espacio disponible en disco, el uso de la memoria, el espacio disponible en las bases de datos, la ejecución propia de las diferentes aplicaciones, entre otros. Pero un punto importante, es que no proporciona las características de un monitoreo para sistemas de misión crítica.

### **3.5. Propuesta**

La propuesta es extender la arquitectura existente en ADSOA, al diseñar e implementar los elementos necesarios para incorporar a dicha arquitectura un componente gestor que cumpla con la tarea de monitoreo de procesos y servicios de negocio. Este sistema gestor, junto con la infraestructura propia de ADSOA, fungirá como un sistema nervioso central, donde todos los sistemas que componen el negocio puedan comunicar el estado de sus procesos y servicios, y en caso de existir alguna anomalía poderlo comunicar visualmente.

Esto permitirá identificar los eventos de falla de manera oportuna y a su vez tomar las acciones necesarias para su atención, esto proporcionará continuidad a la operación evitando los riesgos.

Con el fin de poder ofrecer una alta disponibilidad, confiabilidad y tolerancia a fallos, este monitor de negocio ADSOA estará conformado de subsistemas, que a su vez estos subsistemas pueden contener una o varias entidades que estarán conectadas al campo de datos como los demás sistemas.

## **4.Sistema Gestor para Aplicaciones de Misión Crítica Basado en la Arquitectura ADSOA**

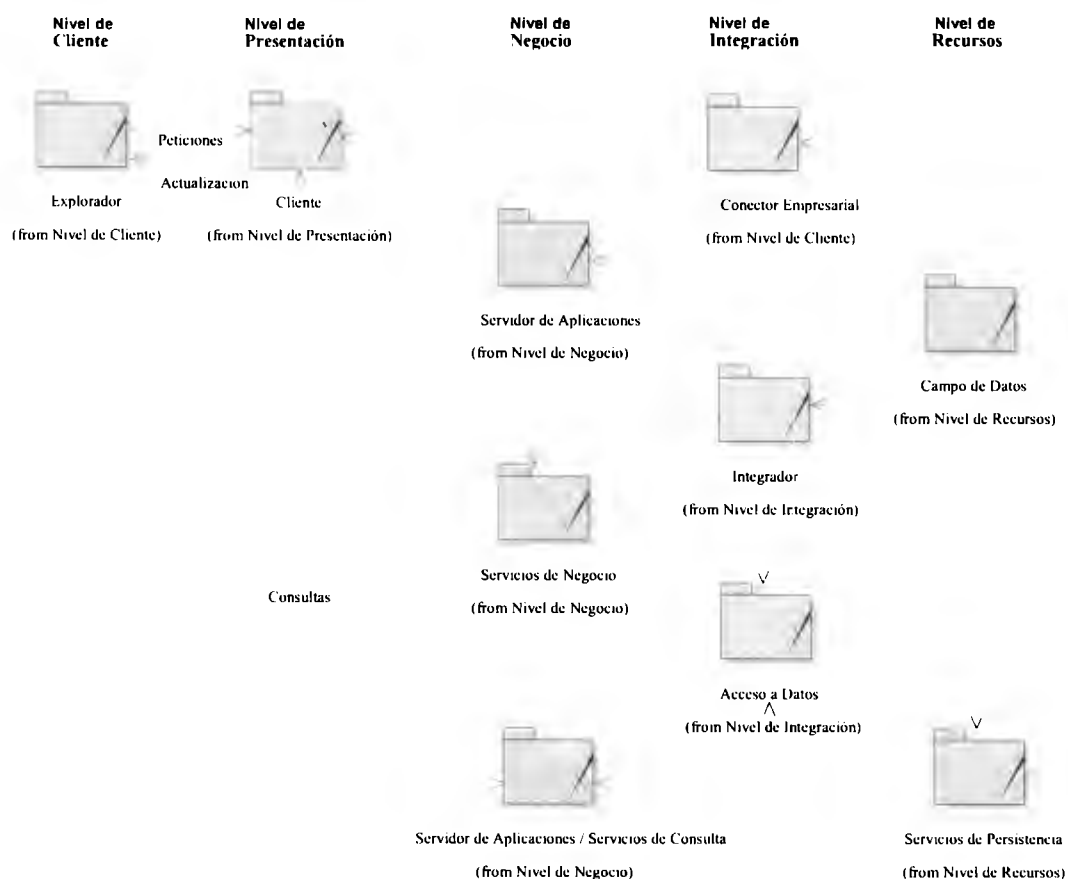
### **4.1.Introducción**

Tomando en consideración el problema planteado sobre la necesidad de un sistema de monitoreo para aplicaciones de misión crítica basados en una arquitectura ADSOA, en este capítulo se definen los elementos que conforman el proceso de monitoreo y el modelado de la arquitectura a implementar, así como los detalles de las especificaciones técnicas.

#### **4.1.1. Niveles de Diseño Arquitectónico**

Básicamente el concepto arquitectónico de la solución puede visualizarse a través de una arquitectura de niveles, como se muestra en la figura 4.1:





**Figura 4.1 Niveles del diseño arquitectónico**

Estos niveles se describen a continuación:

- Nivel Cliente:** Representa los servicios clientes, como los navegadores donde se visualizarán las consultas históricas de los procesos y sus servicios, así como el estado actual del proceso seleccionado.
- Nivel de Presentación:** Representa toda la lógica de presentación como los componentes del cliente que direccionan las peticiones que envía el navegador.
- Nivel de Negocio:** Representa toda la lógica de negocio como los servicios para procesar las peticiones del cliente, aquellas peticiones de consulta y los mensajes de actualización a

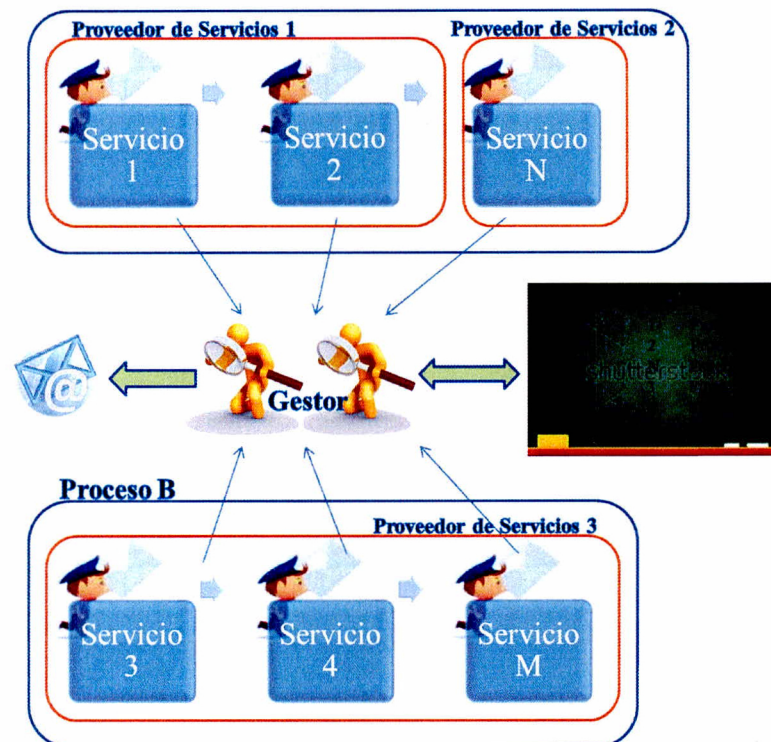
través del servidor de aplicaciones.

- d) Nivel de Integración: Representa toda la lógica de acceso a los recursos como el conector empresarial, *Atom* [14], y los drivers de acceso a las bases de datos.
- e) Nivel de Recursos: Representa todo aquel recurso que el sistema utiliza como son las bases de datos y el campo de datos.

## 4.2. Modelo de Negocio

En la figura 4.2 se muestra el modelo de negocio propuesto para el sistema de monitoreo y control. Los elementos que participan son los siguientes:

1. Proveedor de Servicios: representación de cualquier negocio.
2. Proceso: conjunto de servicios relacionados que logran un fin común.
3. Servicio: recursos o tareas específicas que obtienen un resultado parcial.
4. Gestor: elemento encargado del monitoreo de los procesos y sus servicios. Además de gestionar la ejecución de cada etapa de los procesos y notifica si existe algún error en los mismos.



*Figura 4.2 Modelo de negocio*

### 4.3. Modelo Conceptual

Con base en el modelo de negocio y utilizando la arquitectura ADSOA, se hace la representación del modelo conceptual en la figura 4.3. Es importante mencionar que se debe hacer referencia al punto 2.4 “ADSOA”.

1. Proveedor de Servicios: es la representación de cualquier Negocio del Sistema.
2. Negocio: está compuesto por Entidades autónomas que realizan tareas específicas.
3. Entidad: es encargada de la comunicación con otras entidades, mediante el Campo de Datos, además de ofrecer y solicitar Servicios.
4. Campo de Datos: medio por el cual se comunican cada una de las entidades de diversos negocios.
5. Base de Datos: repositorio que contiene la información de los estados de los procesos, así

como de cada uno de los servicios que solicitan.

6. Servidor de Aplicaciones: elemento encargado de las solicitudes sobre el estado de los procesos.
7. Subsistema Gestor de Procesos de Negocio: es el subsistema encargado del monitoreo de los estados tanto de los procesos, como de los servicios solicitados entre entidades.

### **4.3.1. Sistema Gestor para Aplicaciones de Misión Crítica**

A continuación se describen los componentes que conforman el sistema gestor, las relaciones que existen entre estos y las funciones que realizan para cumplir con el monitoreo de procesos y servicios de los sistemas conectados al campo de datos.

**Gestor:** este componente es el encargado de recibir los mensajes de las entidades que son gestionables, que proveen un servicio y que pertenecen a un proceso de negocio. Este mensaje trae la información del identificador del proceso en ejecución, del servicio que provee la entidad y el estado en el que se encuentra (ver punto 4.2.2).

Después de la recepción del mensaje, el Gestor le notifica al Integrador de la llegada del mismo.

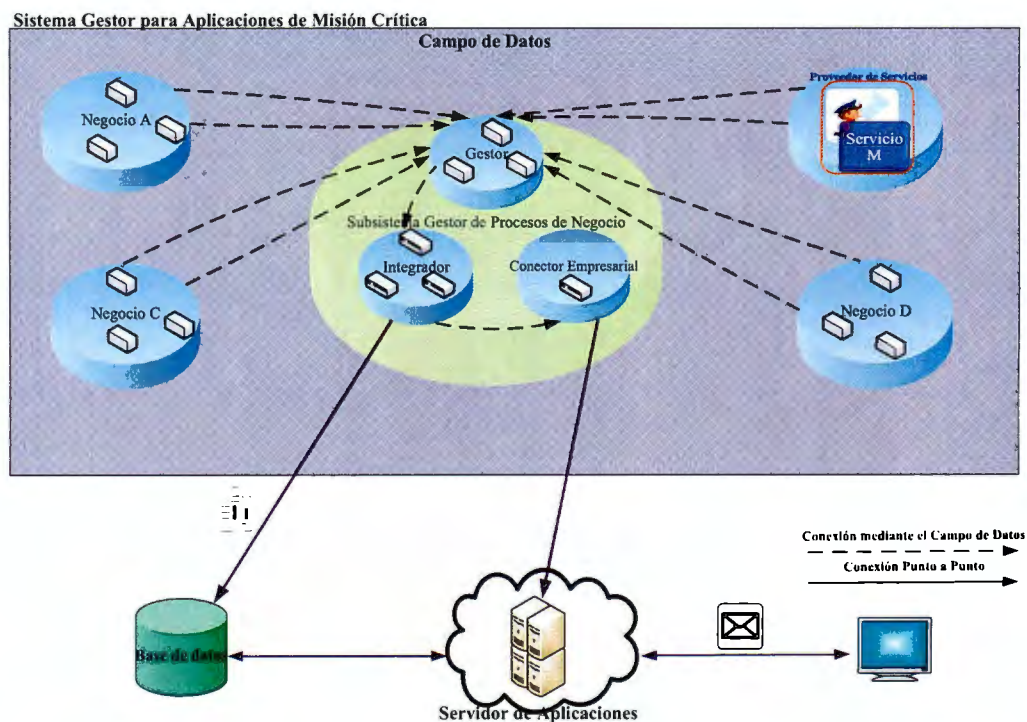
**Integrador:** este componente recibe el mensaje de notificación del Gestor de cualquier cambio en el estado de los procesos y servicios de negocio. Al ser el puente de comunicación entre el campo de datos y la base de datos del sistema, es el encargado de registrar de los mismos.

Después de registrar estos cambios en la base de datos, el Integrador le confirma al Conector Empresarial mediante un mensaje que el proceso fue exitoso.

**Conector Empresarial:** este componente recibe el mensaje de notificación del Integrador que le indica que la persistencia sobre los cambios de estado de los procesos y servicios de negocio se realizó con éxito. Al ser el puente de comunicación entre el campo de datos y el servidor de aplicaciones, envía un mensaje en formato Json [25] hacia la cola JMS [18] del servidor de aplicaciones con la información del identificador del proceso en ejecución, el identificador del servicio actual y el estado del mismo.

**Base de Datos:** repositorio de información donde se registran los cambios de estado relacionados con los procesos y servicios de negocio gestionables, lo que permite llevar el histórico de la ejecución de los mismos.

**Servidor de Aplicaciones:** es el encargado de recibir la información de actualización del cambio de estado de los procesos y servicios de negocio gestionables, para después desplegar su ejecución mediante una aplicación web portable.



*Figura 4.3 Modelo conceptual*

## 4.4. Operatividad

El Sistema Gestor debe cumplir con los requerimientos especificados en el capítulo 3.

### 4.4.1. Flujo Básico

A continuación se presenta el flujo básico del “Proceso de Monitoreo” que esta propuesta integra a los sistemas de misión crítica basados en una arquitectura ADSOA.

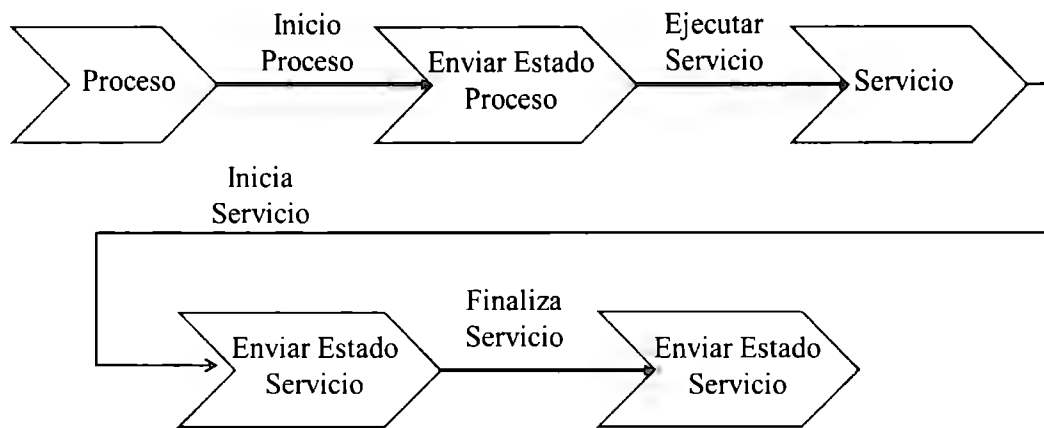
Existe una condición inicial para cumplir con este cometido y se especifica a continuación:

- El Sistema Gestor debe conocer los procesos y sus servicios relacionados. Para cumplir con esta condición, el Sistema Gestor obtendrá esta información de su repositorio para mapear cada etapa de cualquier proceso.

## 4.4.2. Eventos

Estos eventos muestran los instantes exactos cuando cada Entidad gestionable debe notificar al Sistema Gestor sobre los cambios de estado de los servicios y procesos a su cargo, ver figura 4.4.

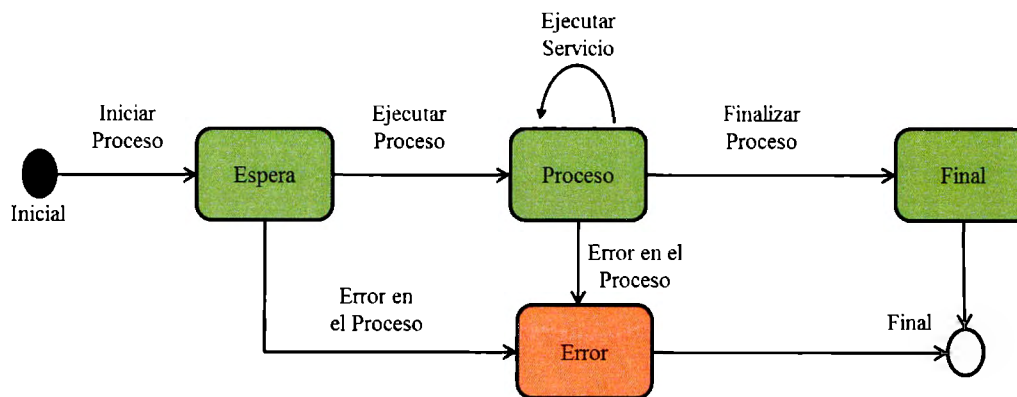
1. Inicio de proceso.
  - 1.1. Ocurre cuando se ejecuta el servicio con el que inicia un proceso, la Entidad responsable debe enviar al Gestor el identificador relacionado con el proceso actual.
2. Inicio de servicio.
  - 2.1. Ocurre al iniciar la ejecución de cualquier servicio, la Entidad responsable debe enviar al Gestor el aviso de inicio del servicio actual.
  - 2.2. Si llegara a existir algún error en la ejecución del servicio, la Entidad responsable del servicio actual debe enviar el estado de error al Sistema Gestor para que este mande el mensaje de aviso del mismo.
3. Fin de servicio.
  - 3.1. Ocurre al finalizar la ejecución de cualquier servicio, la Entidad responsable debe enviar al Sistema Gestor el aviso de finalización del servicio actual.
4. Fin de proceso.
  - 4.1. Ocurre al finalizar la ejecución del último servicio de un proceso, la Entidad responsable debe enviar al Sistema Gestor el aviso de finalización del proceso actual.



*Figura 4.4 Flujo básico*

### 4.4.3. Diagrama de Estado de los Procesos

En la figura 4.5 se muestran los diferentes estados que pueden tener los procesos del negocio.



*Figura 4.5 Diagrama de estado de los procesos*

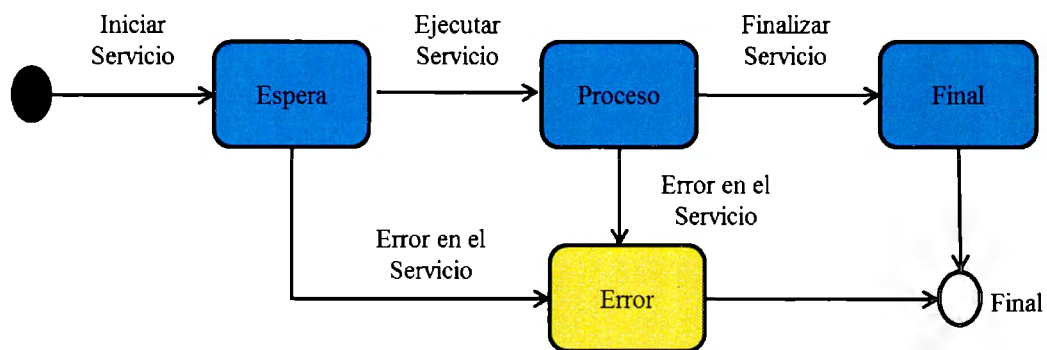


Estado	Evento	Descripción
<b>Espera</b>	Antes de iniciar el proceso	Estado inicial del proceso. Puede ocurrir un error.
<b>Proceso</b>	El proceso está en ejecución	El proceso está en ejecución. En este momento se procesan los servicios que componen dicho proceso. Puede ocurrir un error.
<b>Final</b>	Fin del proceso	Termina el proceso. siempre que los servicios que lo forman hayan tenido estados de "Finalizado"
<b>Error</b>	Se trunca el proceso	Debido a un estado de "Error" en la ejecución de un servicio, provoca un estado erróneo en alguna etapa del proceso. Se debe notificar le error.

*Tabla 4.1 Estados de los procesos*

#### 4.4.4. Diagrama de Estados de los Servicios

En la figura 4.6 se muestran los diferentes estados que pueden tener los servicios relacionados con los procesos del negocio.



*Figura 4.6 Diagrama de estados de los servicios*

Estado	Evento	Descripción
<b>Espera</b>	Antes de iniciar el servicio	Estado inicial del servicio. Puede ocurrir un error o iniciar su auto recuperación.
<b>Proceso</b>	El servicio está en ejecución	Procesando el servicio. Puede ocurrir un error.
<b>Final</b>	Fin del servicio	Termina el servicio.
<b>Error</b>	Se trunca el proceso del servicio	Estado erróneo del servicio

*Tabla 4.2 Estados de los servicios*

## **4.5. Formato de los Mensajes para el Proceso de Gestión.**

En esta sección se describen los mensajes implementados para realizar el proceso de gestión.

### **4.5.1. Mensajes Reservados**

Dentro del campo de datos existe un grupo de mensajes reservados para mensajes específicos, por ejemplo, mensajes de control, mensajes sobre el estado del campo de datos, entre otros.

El formato de un mensaje es una secuencia de tipos de datos básicos. Para crear esta secuencia a cada tipo de dato se le asignó un símbolo, como se muestra en la siguiente tabla 4.3.

<b>Símbolo</b>	<b>Tamaño</b>	<b>Rango</b>
<b>1B</b>	1 byte	0 – 255
<b>2B</b>	2 byte	0 – 65535
<b>4B</b>	4 byte	0 – 4294967295
<b>AB</b>	Arreglo variable de bytes.  Este campo solo puede ir como último elemento en un mensaje y su longitud queda definida por los bytes restantes.	

**Tabla 4.3 Tipos de datos**

En la tabla 4.4 se muestran los números de mensajes reservados para el proceso de gestión:

<b>65527</b>	<b>CambioEstado</b>
DESCRIPCIÓN	Mensaje de Gestión de los Procesos y Servicios
INTERPRETACIÓN	Mensaje de notificación al Gestor cuando ocurre un cambio de estado en los procesos y servicios.
FORMATO	{2B} {2B} {2B}
<b>65526</b>	<b>IntegrarCambioEstado</b>
DESCRIPCIÓN	Mensaje de integración.
INTERPRETACIÓN	Mensaje de control para almacenar la información del cambio de estado en la base de datos..
FORMATO	{2B} {2B} {2B}
<b>65525</b>	<b>NotificarConector</b>
DESCRIPCIÓN	Mensaje de notificación
INTERPRETACIÓN	Mensaje de notificación al conector empresarial sobre el existo en la persistencia de la información
FORMATO	{2B} {2B} {2B}

**Tabla 4.4 Mensajes reservados**

## 4.5.2. Formato de los Mensajes

**CambioEstado:** Esta estructura de información indica que hubo un cambio de estado en los procesos o servicios del negocio. Es enviado desde cualquier Entidad hacia el Gestor de Procesos y Servicios. Este mensaje debe ser transaccional, ver tabla 4.5.

Formato			
	Tipo	Dato	Significado
Núm Serv	2B	Número de servicio	Es el número reservado para el aviso de cambio de estado de los procesos y servicios
Estructura del foliado	2B	Identificador del Negocio	Es la triada de números que identifica a la entidad que envía el mensaje. Es el identificador definido para el campo de datos.
	2B	Identificador del Subsistema	
	2B	Identificador de la Entidad	
	1B	Identificador Pivote	Es el identificador que indica el pivote. Este se define al nivel del negocio.
	4B	Secuencial	Es el número secuencial del mensaje por Identificador Pivote en la entidad origen o emisora.
	AB1	Evento	Es un número único que se calcula aplicando la función HASH seleccionada.
	4B	Huella	Es un número único asignado dinámicamente a cada instancia de las instancias de los participantes del Negocio.
Info Serv	2B	Id del Proceso	Identificadores del proceso y los servicios, y el estado actual en el que se encuentran
	2B	Id del Servicio	
	2B	Estado Actual	

**Tabla 4.5 Formato mensaje CambioEstado**

**IntegrarCambioEstado:** Esta estructura de información le notifica al Integrador que debe almacenar la información del servicio en el repositorio. Se envía desde el Gestor hacia el Integrador. Este mensaje debe ser transaccional, ver tabla 4.6.

Formato			
Tipo	Dato	Significado	
Núm Serv	2B	Número de servicio de la integración del cambio de estado	Es el número reservado para el proceso de almacenamiento de la información
Estructura del foliado	2B	Identificador del Negocio	Es la triada de números que identifica a la entidad que envía el mensaje. Es el identificador definido para el campo de datos.
	2B	Identificador del Subsistema	
	2B	Identificador de la Entidad	
	1B	Identificador Pivote	Es el identificador que indica el pivote. Este se define al nivel del negocio.
	4B	Secuencial	Es el número secuencial del mensaje por Identificador Pivote en la entidad origen o emisora.
	AB1	Evento	Es un número único que se calcula aplicando la función HASH seleccionada.
Info Serv	4B	Huella	Es un número único asignado dinámicamente a cada instancia de las instancias de los participantes del Negocio.
	2B	Id del Proceso	Identificadores del proceso y los servicios, y el estado actual en el que se encuentran
	2B	Id del Servicio	
2B	Estado Actual		

**Tabla 4.6 Formato mensaje IntegrarCambioEstado**

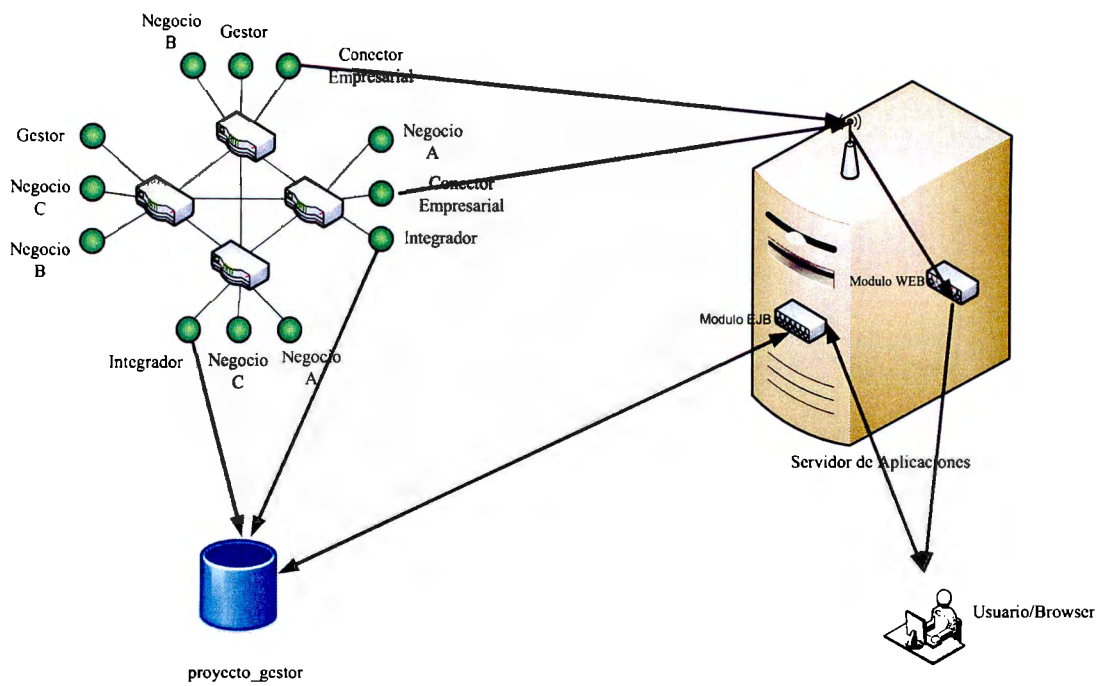
**NotificarConector:** Esta estructura de información le notifica al Conector Empresarial que la persistencia de la información fue exitosa. Este mensaje se envía desde el Integrador hacia el Conector Empresarial, además debe ser transaccional, ver tabla 4.7.

Formato			
	Tipo	Dato	Significado
Núm Serv	2B	Número de servicio	Es el número reservado para el aviso de cambio de estado de los procesos y servicios
Estructura del foliado	2B	Identificador del Negocio	Es la triada de números que identifica a la entidad que envía el mensaje. Es el identificador definido para el campo de datos.
	2B	Identificador del Subsistema	
	2B	Identificador del Entidad	
	1B	Identificador Pivote	Es el identificador que indica el pivote. Este se define al nivel del negocio.
	4B	Secuencial	Es el número secuencial del mensaje por Identificador Pivote en la entidad origen o emisora.
	AB1	Evento	Es un número único que se calcula aplicando la función HASH seleccionada.
	4B	Huella	Es un número único asignado dinámicamente a cada instancia de las instancias de los participantes del Negocio.
Info Serv	2B	Id del Proceso	Información que le indica al conector empresarial del existo de la persistencia.
	2B	Id del Servicio	
	2B	Estado Actual	

**Tabla 4.7 Formato mensaje NotificarConector**

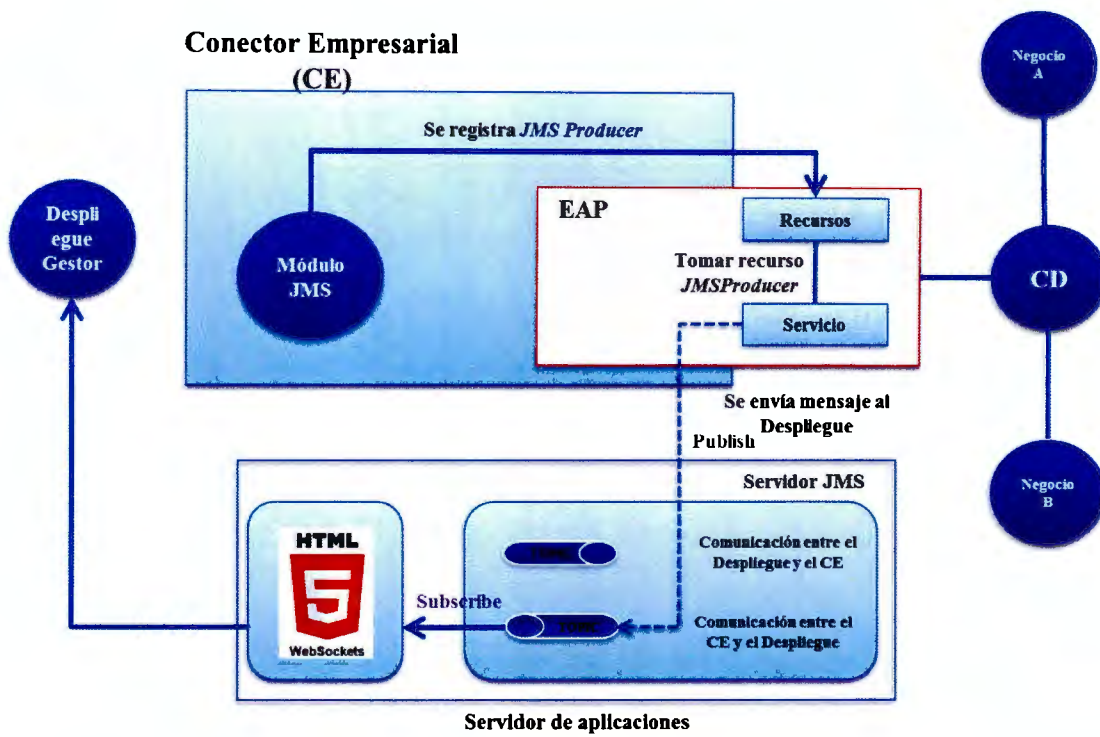
## 4.6. Arquitectura

En esta sección se muestra la arquitectura propuesta para el desarrollo del Sistema Gestor para Aplicaciones de Misión Crítica que utilizan una Arquitectura ADSOA. La figura 4.9 muestra la arquitectura utilizada a nivel de recursos (ver sección 4.1.1).



**Figura 4.7 Diagrama de arquitectura nivel de recursos**

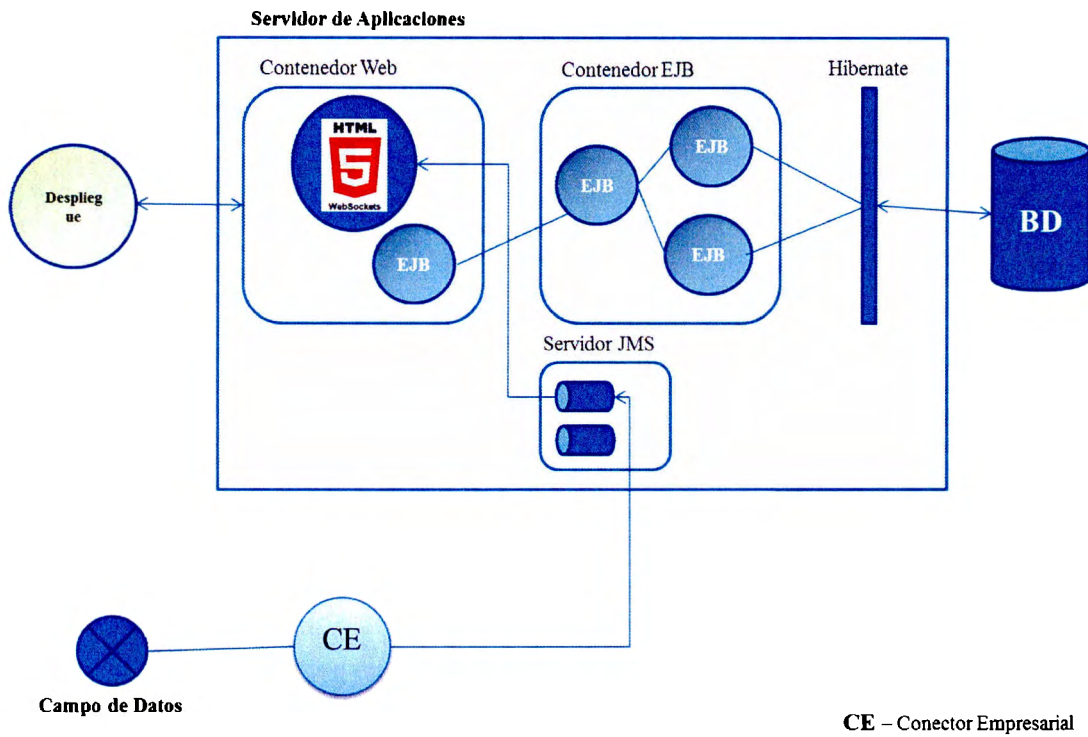
La figura 4.8 muestra la arquitectura utilizada a nivel de integración (ver sección 4.1.1).



*Figura 4.8 Diagrama de arquitectura nivel de integración*

La figura 4.9 muestra la arquitectura utilizada a nivel de negocio (ver sección 4.1.1).





*Figura 4.9 Diagrama de arquitectura nivel de negocio*

## 4.7. Diseño

Las figuras 4.10 y 4.11 muestran el diseño de la base de datos que permitirá llevar el registro de los eventos de cada proceso y servicios de negocio que son gestionables. Además de proporcionar la información histórica del proceso a consultar.

### 4.7.1. Modelo de la Base de Datos

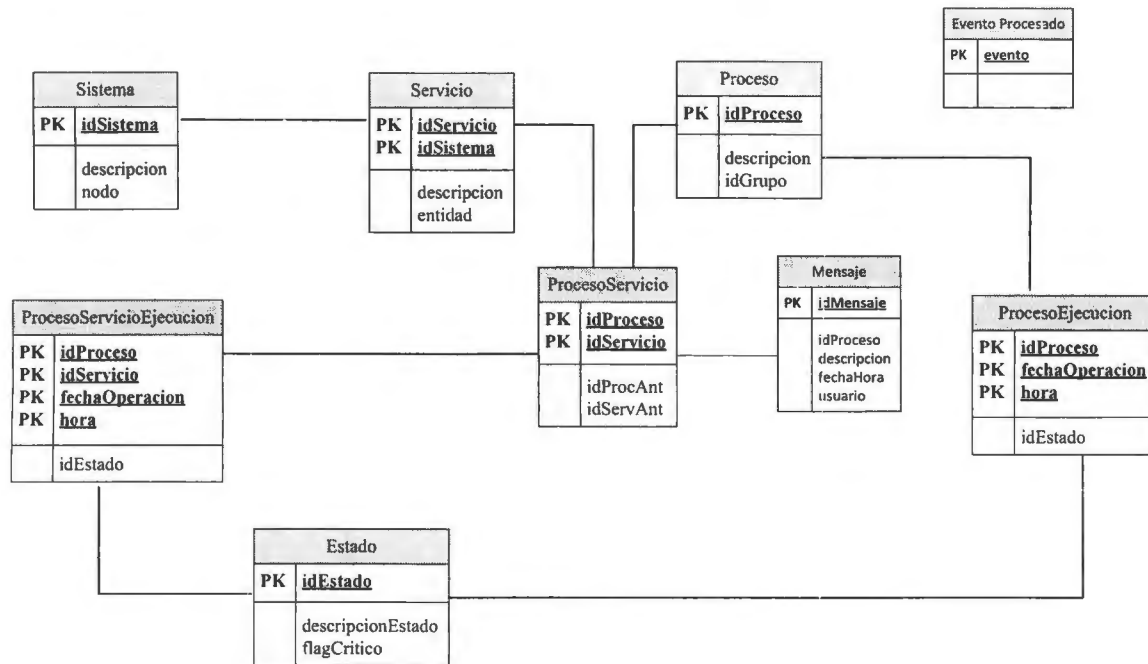


Figura 4.10 Modelo de base de datos

### 4.7.2. Diagrama Entidad Relación

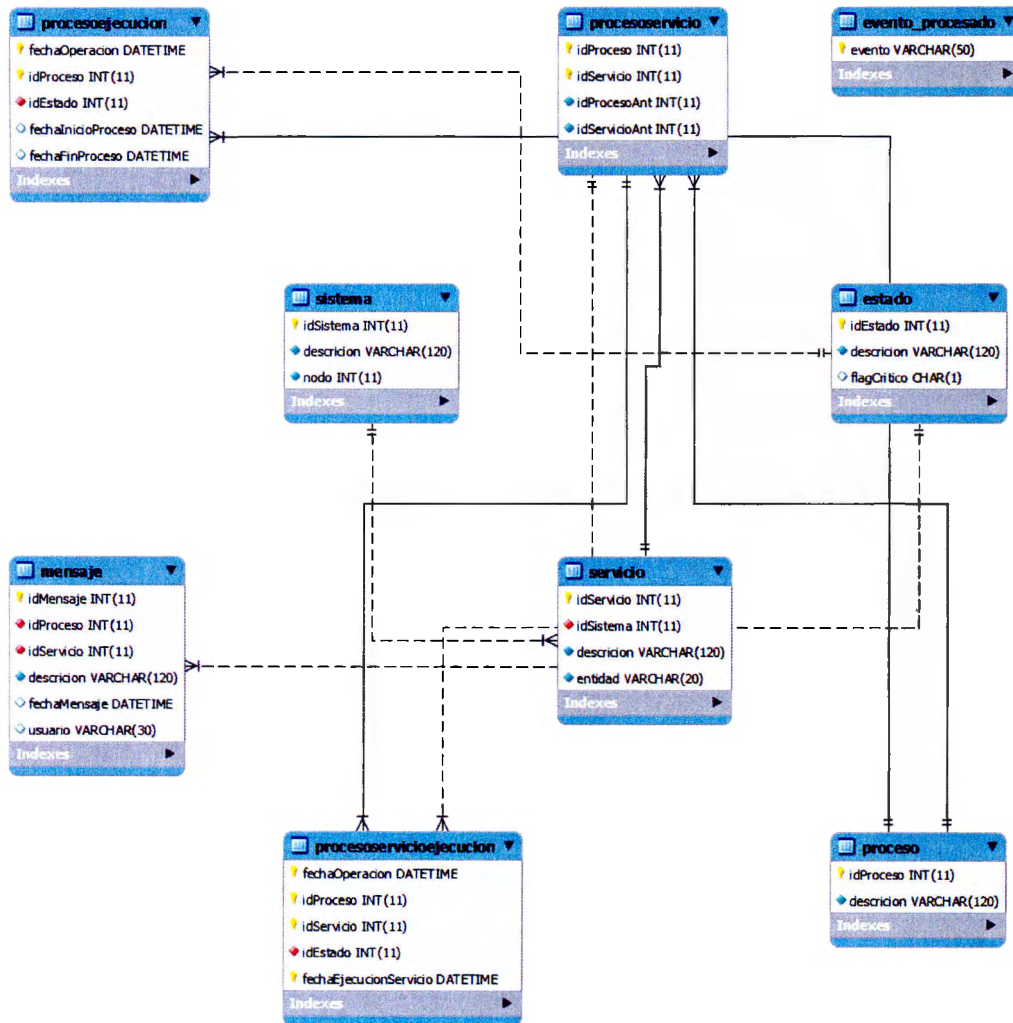


Figura 4.11 Diagrama entidad relación

### 4.7.3. Diagramas de Clases

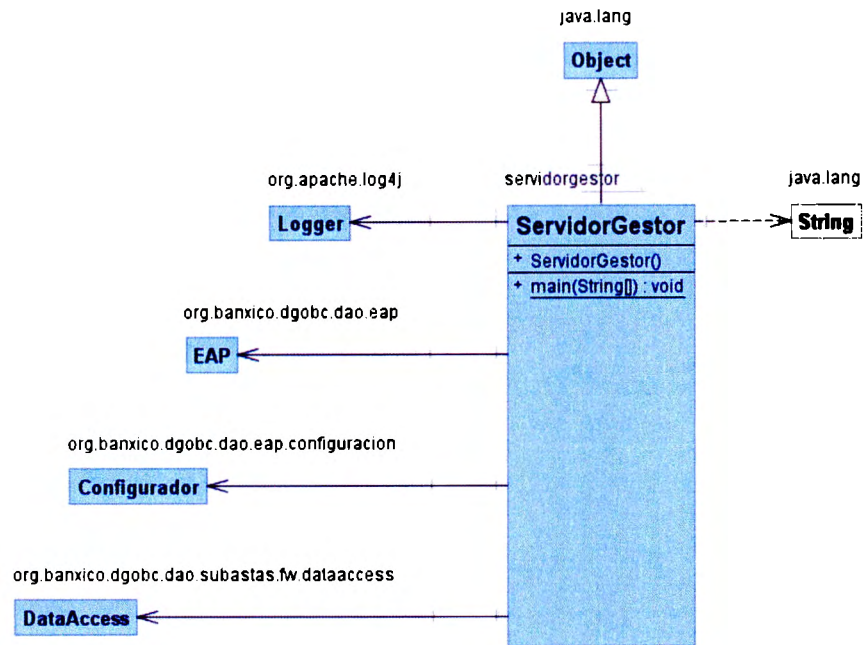


Figura 4.12 Diagrama de clases del gestor

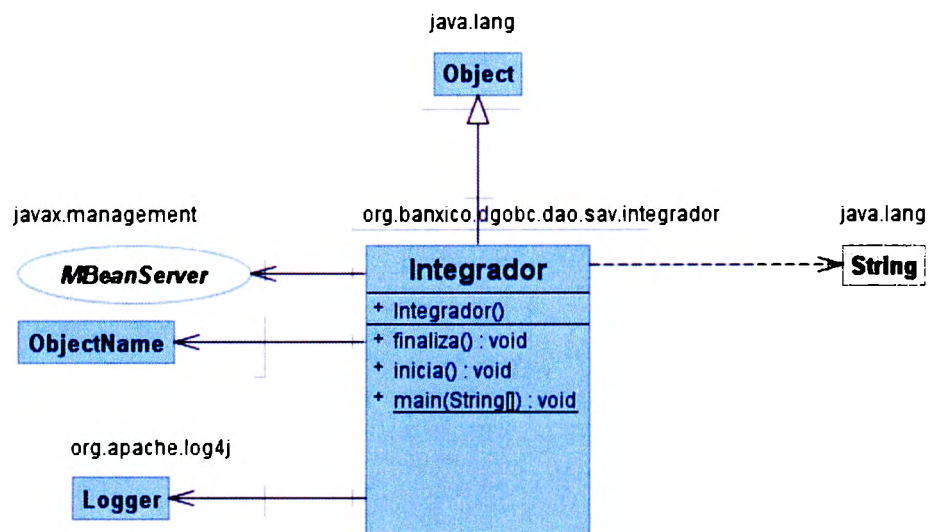
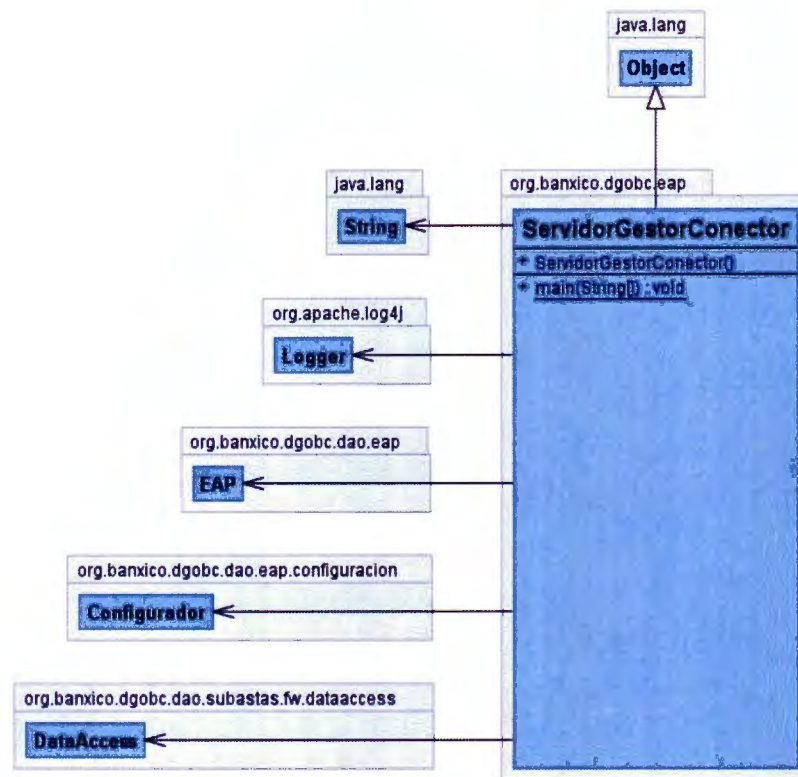
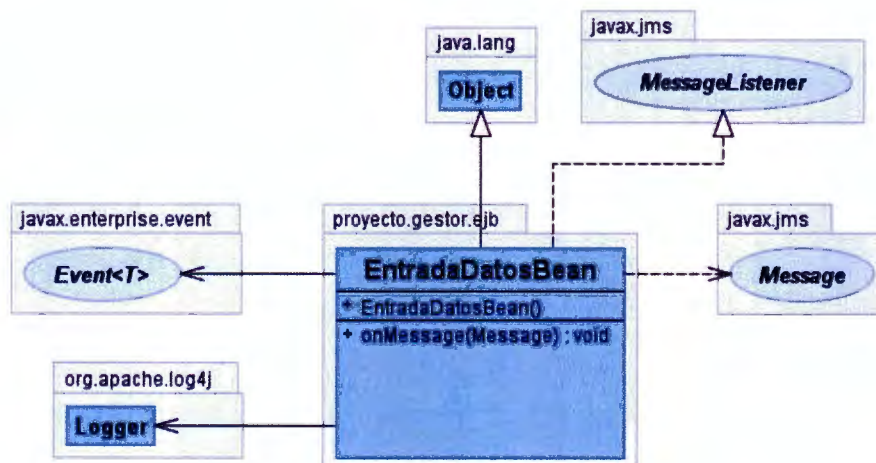


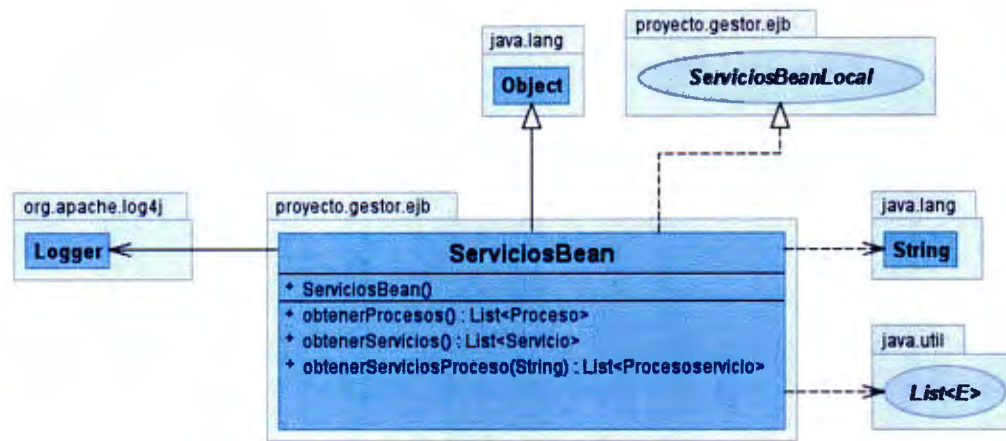
Figura 4.13 Diagrama de clases del integrador



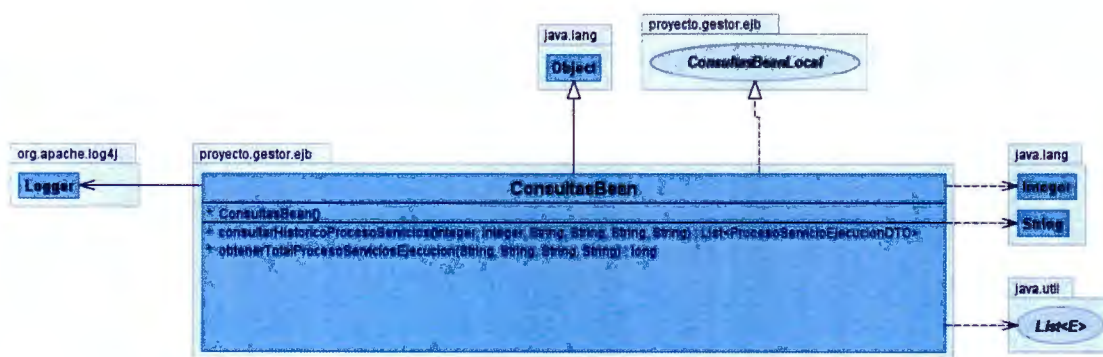
*Figura 4.14 Diagrama de clases del conector empresarial*



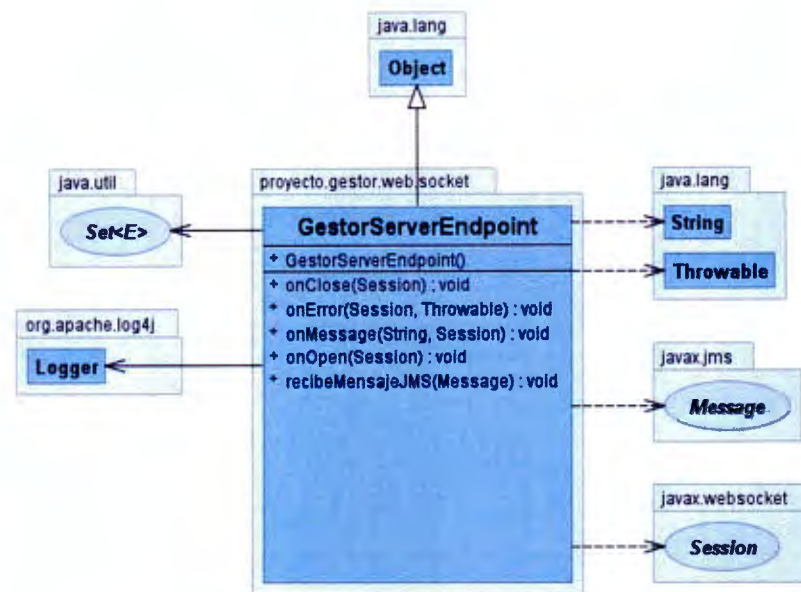
*Figura 4.15 Diagrama de clases del message driven bean*



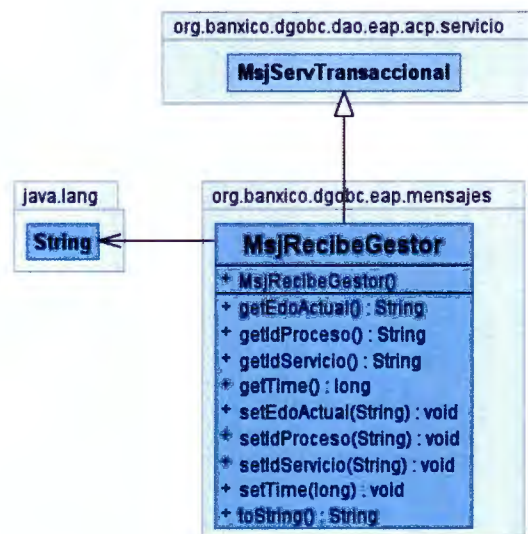
*Figura 4.16 Diagrama de clases de la clase ServiciosBean*



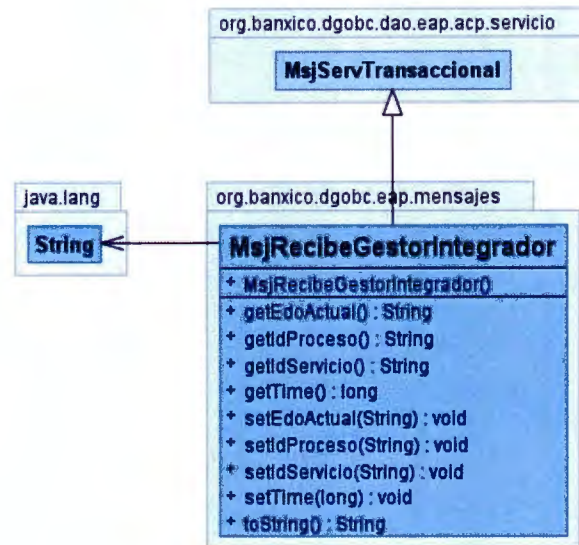
*Figura 4.17 Diagrama de clases de la clase ConsultasBean*



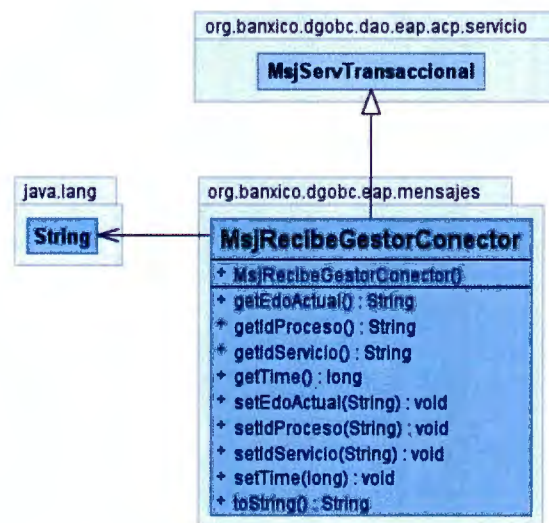
*Figura 4.18 Diagrama de clases del ServerEndPoint*



*Figura 4.19 Diagrama de clases del mensaje MsjRecibeGestor*



*Figura 4.20 Diagrama de clases del mensaje MsjRecibeGestorIntegrador*



*Figura 4.21 Diagrama de clases del mensaje MsjRecibeGestorConector*



## **4.8. Configuración General**

En esta sección se especifica la configuración necesaria para cumplir con el objetivo de contar con un Sistema Gestor para Aplicaciones de Misión Crítica que utilizan una Arquitectura ADSOA.

### **4.8.1. Configuración del Atom**

En el análisis se observó que la configuración del Atom fue poco invasiva, es decir, para que cada Entidad tenga la inteligencia para saber que es gestionable o no, solamente se incorporaron dos parámetros necesarios para este fin “esGestionable y numeroProceso”. Con esta actualización se puede llevar a cabo el proceso de monitoreo.

```

<?xml version='1.0' encoding='UTF-8'?>

<!ELEMENT SERVICIOS_OFRECIDOS (SERVICIO|PIVOTE)*>

<!ELEMENT PIVOTE (#PCDATA)>

<!ELEMENT SERVICIO (CLASE_SERVICIO|CLASE_MENSAJE)*>
<!-- ATTLIST SERVICIO
    numeroProceso CDATA #IMPLIED
    esGestionable CDATA #IMPLIED
    esControl CDATA #IMPLIED
    numeroSolicitudes CDATA #IMPLIED
    numeroServicio CDATA #IMPLIED
-->

<!ELEMENT CLASE_MENSAJE (#PCDATA)>

<!ELEMENT CLASE_SERVICIO (#PCDATA)>

```

*Figura 4.22 Archivo de configuración dtd*

## 4.8.2. Configuración del Integrador

Este componente funge como el puente entre el campo de datos y el repositorio de información, para su correcto funcionamiento requiere de un archivo de configuración (hibernate.cfg.xml) que contiene las etiquetas necesarias para el acceso a la base de datos; además de incluir el mapeo de las clases a las que hace referencia. La figura 4.13 muestra este archivo.

```

<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>

    <property name="hibernate.connection.url">

      jdbc:mysql://localhost:3306/proyecto_gestor
    </property>
    <property name="hibernate.connection.driver_class">

      com.mysql.jdbc.Driver
    </property>
    <property name="hibernate.connection.username">

      root
    </property>
    <property name="hibernate.connection.password">

      Administr4d0r
    </property>

    <property name="hibernate.dialect">org.hibernate.dialect.MySQLInnoDBDialect </property>

    <property name="hibernate.enable_sql">true</property>
    <property name="hibernate.format_sql">true</property>
    <property name="hibernate.current_session_context_class">thread</property>
    <property name="hibernate.transaction.factory_class">org.hibernate.transaction.JDBCTransactionFactory</property>
    <property name="hibernate.hbm2ddl.auto">update</property>
    <property name="hibernate.hbm2ddl.max_size">5</property>
    <property name="hibernate.hbm2ddl.max_lines">20</property>
    <property name="hibernate.hbm2ddl.max_statements">1800</property>
    <property name="hibernate.hbm2ddl.max_statements">50</property>
    <property name="hibernate.hbm2ddl.schema_drop_statements">true</property>

    <mapping class="proyecto_gestor.persistencia.entity.EventoPersonal"></mapping>
    <mapping class="proyecto_gestor.persistencia.entity.Estado"></mapping>
    <mapping class="proyecto_gestor.persistencia.entity.Mensaje"></mapping>
    <mapping class="proyecto_gestor.persistencia.entity.Pedido"></mapping>
    <mapping class="proyecto_gestor.persistencia.entity.PedidoDetalle"></mapping>
    <mapping class="proyecto_gestor.persistencia.entity.PedidoServicio"></mapping>
    <mapping class="proyecto_gestor.persistencia.entity.PedidoServicioDetalle"></mapping>
    <mapping class="proyecto_gestor.persistencia.entity.Servicio"></mapping>
    <mapping class="proyecto_gestor.persistencia.entity.Sistema"></mapping>

  </session-factory>
</hibernate-configuration>

```

**Figura 4.23** Archivo *hibernate.cfg.xml*

### 4.8.3. Configuración del Conector Empresarial

Este componente funge como el puente entre el campo de datos y el servidor JMS. Este conector utiliza un cliente JMS para realizar la conexión con las colas de mensajes y así poder enviar la información de aviso de cambio de estado de los procesos y servicios gestionables. Esta información se envía en formato Json [25], ver figuras, 4.14, 4.15 y 4.16.

```

    private static final String IP_JMS = "127.0.0.1";
    private static final String PUERTO_JMS = "3700";
    private static final String URL_JMS_SERVER = "http://127.0.0.1:3700";
    private static final String JNDI_JMS_FACTORY = "jms/entradaDatosCF";
    private static final String URL_JMS_TOPIC = "jms/entradaDatos";

```

**Figura 4.24 Constantes para el JMS Topic Client**

```

private JMSTopicClient inicializarJMS() throws JMSException {
    // Inicializar el logger para el cliente
    Logger logger = Logger.getLogger(JMSTopicClient.class);

    // Definir las propiedades de configuración
    Properties properties = new Properties();
    properties.put(Context.PROVIDER_URL, URL_JMS_SERVER);
    properties.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.enterprise.naming.impl.SerialInitContextFactory");
    properties.put(Context.URL_PKG_PREFIXES, "com.sun.enterprise.naming");
    properties.put(Context.STATE_FACTORIES, "com.sun.corba.ee.impl.presentation.rmi.JNDIStateFactoryImpl");
    properties.put(Context.SECURITY_PRINCIPAL, "admin");
    properties.put(Context.SECURITY_CREDENTIALS, "admin");

    // Definir las propiedades de configuración de JNDI
    System.setProperty("org.omg.CORBA.ORBInitialHost", IP_JMS);
    System.setProperty("org.omg.CORBA.ORBInitialPort", PUERTO_JMS);

    try {
        // Crear el cliente JMS
        JMSTopicClient jmsClient = new JMSTopicClient(properties,
            JNDI_JMS_FACTORY,
            URL_JMS_TOPIC);

        // Iniciar la conexión JMS
        jmsClient.startJMSConnection();
        logger.info("Se inició la conexión a JMS...");

        // Mostrar información de configuración
        logger.info("----- 1: " + jmsClient.getMessageProducer().getDeliveryMode());
        logger.info("----- 1: " + jmsClient.getMessageProducer().getTimeToLive());

        // Configurar el tiempo de vida y el modo de entrega
        jmsClient.getMessageProducer().setTimeToLive(1000 * 60 * 20);
        jmsClient.getMessageProducer().setDeliveryMode(DeliveryMode.NON_PERSISTENT);

        logger.info("----- 2: " + jmsClient.getMessageProducer().getDeliveryMode());
        logger.info("----- 2: " + jmsClient.getMessageProducer().getTimeToLive());

        return jmsClient;
    } catch (NamingException neCon) {
        logger.error("Error: Falló en el lookup: "
            + neCon.getMessage());
    } catch (RuntimeException re) {
        logger.error("Error: No se pudo realizar la conexión: "
            + re.getMessage());
        re.printStackTrace();
    }

    return null;
}

```

**Figura 4.25 Inicializar JMSTopicClient**

```

@Override
protected void procesaMensaje() throws InvalidRuleException, EAPEException {

    logger.info("***** Envío de mensaje *****");
    logger.info("***** + mensaje.getIdProceso()");
    logger.info("***** + mensaje.getIdServicio()");
    logger.info("***** + mensaje.getEdoActual()");

    String obj;
    String msj;

    JMSTopicClient jmsClient = (JMSTopicClient)recursos.getRecurso(Constants.LLAVE_JMS);

    try {
        obj = "***** + mensaje.getIdProceso() + " + "***** + mensaje.getIdServicio() + " + "***** + mensaje.getEdoActual() + "*****";
        |
        msj = "***** + obj + "*****";
        logger.info("***** + msj");
        jmsClient.sendJMSMessage(msj);

        System.out.println("***** Envío de mensaje *****");

    } catch (javax.jms.JMSException jmseMsg) {
        System.err.println("***** Envío de mensaje *****");
        + jmseMsg.getMessage());
    }
}

```

*Figura 4.26 Envío del mensaje de aviso*

#### 4.8.4. Configuración del MDB y WebSocket

Para lograr la recepción y el despliegue del mensaje de aviso sobre el cambio de estado de los procesos y servicios gestionables, se implementó el uso de un canal comunicación bidireccional entre el servidor de aplicaciones y el browser de los usuarios llamado *WebSocket*, ver figura 4.19.

Para integrar el *WebSocket* con JMS es necesario el uso de CDI Events, ver figura 4.18, ya que de forma nativa no interactúan [20]. Con esta implementación es posible recibir la información proveniente de la cola JMS y desplegarla en el browser, ver figura 4.17.

```

@MessageDriven(activationConfig = {
    @ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Topic"),
    @ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "jms/entradaDatos"),
    @ActivationConfigProperty(propertyName = "clientId", propertyValue = "jms_entradaDatos"),
    @ActivationConfigProperty(propertyName = "subscriptionName", propertyValue = "jms/entradaDatos")
})
public class EntradaDatosBean implements MessageListener {

    @Inject
    @NotifyMessage
    Event<Message> jmsEvent;

    private static final Logger logger = Logger.getLogger(EntradaDatosBean.class);

    public EntradaDatosBean() {
    }

    @Override
    public void onMessage(Message message) {
        try {
            logger.debug("-->Llego mensaje a la cola timestamp: "+message.getJMSTimestamp());

            String mensaje = message.getBody(String.class);
            logger.debug("-->Se mando mensaje al WS desde EntradaDatosBean: " + mensaje);
            jmsEvent.fire(message);

        } catch (Exception ex) {
            logger.debug("Error al enviar la informacion a WS" + ex.getMessage());
        }
    }
}

```

*Figura 4.27 Message Driver Bean*

```

package proyecto.gestor.cdi;

import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
import javax.inject.Qualifier;

...
-
-
- @author ...
-

@Qualifier
@Retention(RetentionPolicy.RUNTIME)
@Target({ElementType.METHOD, ElementType.FIELD, ElementType.PARAMETER, ElementType.TYPE})
public @interface NotifyMessage {
}

```

*Figura 4.28 CDI*

```

@Singleton
@ServerEndpoint(value = "/gestorServerEndpoint")
public class GestorServerEndpoint {

    private static Set<Session> clientes = Collections.synchronizedSet(new HashSet<Session>());
    private static final Logger logger = Logger.getLogger(GestorServerEndpoint.class);

    
        * Método que está como observador de los mensajes enviados desde la cola JMS - destino @PUSH
        * @Param msg
        * @Throws Exception
    

    public void recibeMensajeJMS(@Observes @NotifyMessage Message msg) throws Exception {

        logger.debug("GestorServerEndpoint-> Mensaje recibido de la cola JMS-EntradaDatosBean: " + msg.getBody(String.class));
        try {
            for (Session cliente : clientes) {
                cliente.getBasicRemote().sendText(msg.getBody(String.class));
            }
        } catch (Exception ex) {
            logger.debug("Error al enviar el mensaje de la cola JMS desde WS: " + ex.getMessage());
        }
    }

    @OnOpen
    public void onOpen(Session session) {
        logger.info("Server Gestor-> Nueva conexión con el cliente: " + session.getId());
        clientes.add(session);
        logger.debug("Server Gestor-> Usuarios conectados [{"getOpenSessions:" + session.getOpenSessions().size() +
            ", clientes: " + clientes.size() + "}]);
    }

    
        * @Param peer : clientes
        * @Param cveUsuario = String.valueOf(peer.getUserProperties().get("name"));
        * @Param string
    
    for (Session peer : clientes) {
        logger.debug("Server Gestor-> {" + peer.getId() + ", " + cveUsuario + "}");
    }
    }

    @OnMessage
    public void onMessage(String message, Session session) throws IOException {

        logger.debug("Server Gestor-> Nuevo mensaje del cliente{" + session.getId() + "}: " + message + "!");
        logger.debug("Server Gestor-> getOpenSessions: " + session.getOpenSessions().size() + ", clientes: " + clientes.size());

        synchronized (clientes) {
            
                * @Param session = session
                * @Param message = message
            
            for (Session cliente : clientes) {
                if (!cliente.equals(session)) {
                    cliente.getBasicRemote().sendText("{" + cliente.getId() + "}: " + message);
                }
            }
        }
    }

    @OnClose
    public void onClose(Session session) {
        logger.debug("Server Gestor-> Cerrada la conexión para el cliente: " + session.getId() + "!");

        Set<Session> sesionesAbiertas = session.getOpenSessions();
        for (Session peer : sesionesAbiertas) {
            String cveUsuario = String.valueOf(peer.getUserProperties().get("name"));
        }
    }
}

```

```
        logger.debug("Server Restor-> getOpenSessions [" + peer.getId() + ". " + cveUsuario + "]:");
    }

    clientes.remove(session);

    // Se eliminan los peers de la lista
    Set<Session> peersRemove = new HashSet();
    for (Session peer : clientes) {
        if (session.getId().equals(peer.getId())) {
            peersRemove.add(peer);
        }
    }
    clientes.removeAll(peersRemove);
}

@OnError
public void onError(Session clientes, Throwable ex) {
    logger.debug("Server Restor-> Error del cliente: " + clientes.getId() + "-", ex);
}
}
```

**Figura 4.29 ServerEndPoint**



## **5.Pruebas y Resultados**

### **5.1.Introducción**

El principal objetivo que tiene el banco central de México es conservar el valor de la moneda nacional a lo largo del tiempo, logrando así contribuir a mejorar el bienestar económico de los mexicanos [31].

Para este propósito, el banco central realiza distintas actividades y una de las principales es la venta y compra pública de títulos gubernamentales mediante subastas y colocación de valores. Por ejemplo, la subasta primaria de valores gubernamentales se realiza con propósitos de fondeo, donde el gobierno federal emite pagares por los cuales obtiene dinero y al término del plazo debe pagar un interés por estos.

Por otro lado, existe la subasta de regulación monetaria donde el propósito esencial es buscar un equilibrio en la liquidez del país fomentando el ahorro y así evitar en lo más posible la inflación. Esto se logra emitiendo títulos gubernamentales para que la banca los pueda comprar y así el banco central pueda retirar dinero en circulación [32].

### **5.2.Selección de Caso de Prueba**

En este capítulo se describen las pruebas realizadas para comprobar el funcionamiento del Sistema Gestor para Aplicaciones de Misión Crítica basado en una Arquitectura ADSOA. Para este fin, se desarrolló un prototipo que simula una subasta tradicional de títulos gubernamentales, en este caso para CETES y BONOS [34].

Este prototipo permitirá comprobar que la inclusión del sistema gestor en aplicaciones de misión crítica que utilizan una arquitectura ADSOA funciona correctamente proporcionando los resultados esperados.

### **5.3. Escenario de Caso Práctico**

Como ya se mencionó, este caso práctico se basa en una subasta tradicional donde los participantes presentan sus posturas pero no tienen la posibilidad de ver las ofertas de los demás postores hasta que se publican los resultados de la subasta [33].

El proceso de la subasta se divide en cuatro etapas que se describen a continuación:

1. Convocatoria: etapa en la cual se indica el inicio de la subasta y el precio del título a subastar.
  - 1.1. Al estar la subasta activa, es el momento de ingresar las posturas. Estas posturas están conformadas por la institución interesada en estos títulos y el monto de participación.
2. Cierre: etapa que indica el final de la subasta y de la recepción de posturas.
3. Asignación: esta etapa realiza el cálculo de los títulos a recibir de acuerdo al monto de la postura.
4. Publicación: etapa que simula la publicación los resultados de todas las asignaciones.

Para los casos de pruebas, se tienen cinco servicios gestionables que indican la etapa en la que se encuentra la subasta, la tabla 5.1 describe estos servicios.

Servicio	Descripción
Serv 1	Convocatoria
Serv 2	Posturas
Serv 3	Cierre
Serv 4	Asignación
Serv 5	Publicación

*Tabla 5.1 Descripción de los servicios*

## 5.4. Estrategia Adoptada

Se planteó la ejecución de cuatro casos de prueba con diferentes arquitecturas, integrando al sistema gestor a cada uno de estos y así comprobar que su inclusión en sistemas de misión crítica basados en una arquitectura ADSOA es simple y lo más importante es que permite llevar el seguimiento de los procesos y servicios de negocio de una manera eficaz y oportuna.

## 5.5. Resultados Esperados

Los resultados se obtienen de la ejecución de cada subasta, el monitoreo de cada proceso de negocio se visualiza mediante una aplicación empresarial que despliega un grafo donde se observa la etapa en la que se encuentra, además es posible obtener una consulta a la base de datos que describe la hora ejecución de cada proceso y sus servicios, el estado en el que se encuentran, entre otras cosas.

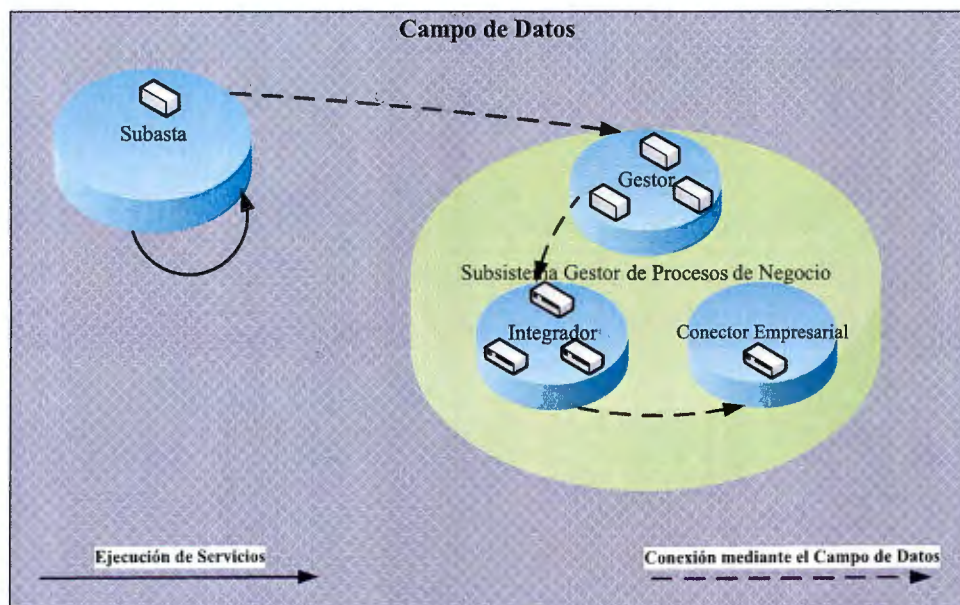
En este capítulo se describen las pruebas realizadas para comprobar el funcionamiento del Sistema Gestor para Aplicaciones de Misión Crítica basado en una Arquitectura ADSOA. Estas pruebas se realizaron mediante un prototipo conectado al campo de datos que simula la subasta de títulos gubernamentales (CETES Y BONOS).

## 5.6. Integración del Sistema Gestor al Prototipo de Subasta de Títulos Gubernamentales

### 5.6.1. Caso de Prueba I

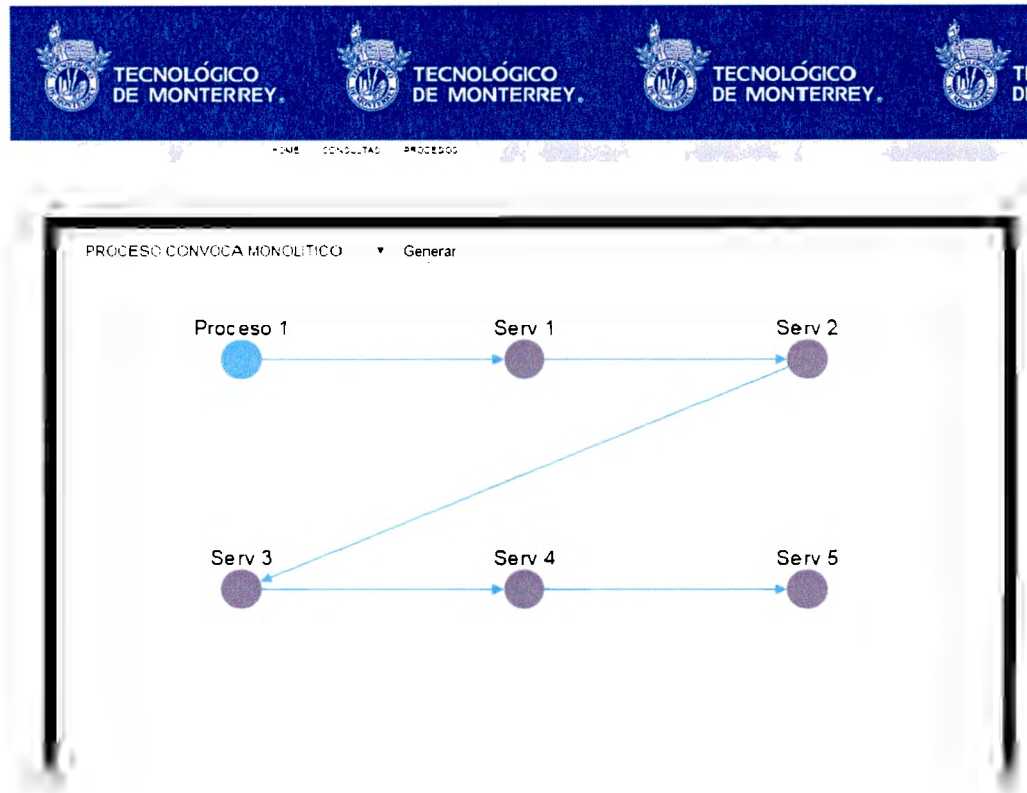
#### 5.6.1.1. Sistema Monolítico

En esta prueba se simula la subasta en un sistema tradicional con una arquitectura cliente-servidor, donde una sola Entidad (Subasta) es encargada de ofrecer y solicitar los servicios necesarios para este fin. La figura 5.1 muestra la arquitectura utilizada.



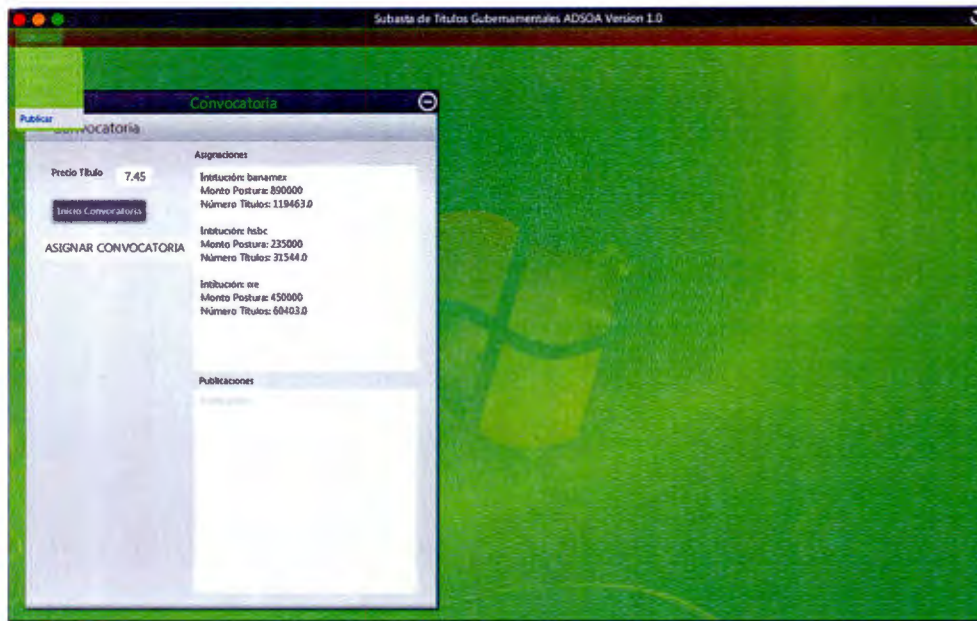
*Figura 5.1 Arquitectura del sistema monolítico*

En esta prueba se demuestra la implementación del sistema gestor en una arquitectura donde solo una Entidad es la encargada de ofrecer y solicitar los servicios necesarios para el proceso de negocio. La figura 5.2 muestra cómo se visualiza el inicio del proceso.



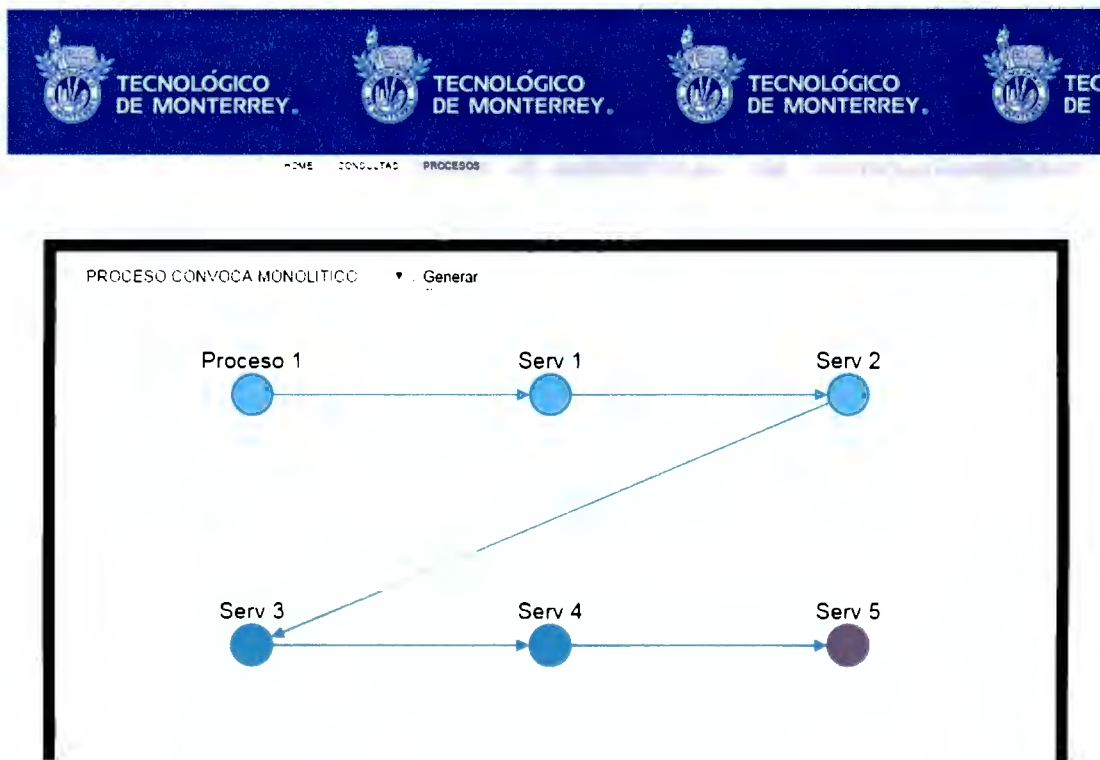
*Figura 5.2 Grafo inicial del sistema monolítico*

Este prototipo tiene un menú que ejecuta cada etapa de la subasta, la convocatoria, el cierre, la asignación y finalmente la publicación. En la figura 5.3 se muestra este proceso, en este caso se tomó la muestra de la etapa de asignación.



**Figura 5.3 Etapa de asignación del sistema monolítico**

En la figura 5.4 se muestra como el grafo avanzó hasta la etapa de asignación, quedando pendiente la publicación.



*Figura 5.4 Grafo del sistema monolítico en la etapa de asignación*

Para respaldar lo anterior, se realizó la consulta a la base de datos del proceso actual y se obtuvo el siguiente resultado.

PROCESO CONVOCA MONOLITICO    Selecionar Servicio    2014-10-26  
2014-11-28   

Historia

Proceso	Descripción Proceso	Servicio	Descripción Servicio	Estado	Fecha Operación	Fecha Ejecución Servicio
PROCESO CONVOCA MONOLITICO		CONVOCA		PROCESO	2014-10-28	2014-10-28 11:47:12 PM
PROCESO CONVOCA MONOLITICO				FINAL	2014-10-28	2014-10-28 11:47:12 PM
PROCESO CONVOCA MONOLITICO				FINAL	2014-10-28	2014-10-28 11:47:42 PM
PROCESO CONVOCA MONOLITICO		BITACORA		FINAL	2014-10-28	2014-10-28 11:47:12 PM
PROCESO CONVOCA MONOLITICO		POSTULANTE		FINAL	2014-10-28	2014-10-28 11:47:12 PM
PROCESO CONVOCA MONOLITICO		SELECCION		FINAL	2014-10-28	2014-10-28 11:47:12 PM
PROCESO CONVOCA MONOLITICO		ASIGNAR		FINAL	2014-10-28	2014-10-28 11:47:12 PM
PROCESO CONVOCA MONOLITICO		5	PUBLICAR	FINAL	2016-10-28	Oct 28, 2016 11:46:49 PM

Mostrando registros 5 de 6 de 6

**Figura 5.5 Consulta de la base de datos del sistema monolítico**

Como se puede observar, esta consulta muestra la descripción de la ejecución del proceso de negocio. Se detalla la hora de ejecución de los servicios, así como el estado en el que se encuentran.

En cuanto al gestor, éste registra en su bitácora los cambios de estado de los servicios gestionables, llevando el seguimiento de los mismos. En la figura 5.6 se muestra la bitácora.



```
2014-10-28 23:29:07,604 [INFO] [main] - ***Servidor Gestor Operando
2014-10-28 23:37:12,125 [DEBUG] [pool-3-thread-2] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(1)evento(cbbd8c05c4b5842e3ba72018fb47
2014-10-28 23:37:12,229 [DEBUG] [pool-3-thread-2] - El evento (cbbd8c05c4b5842e3ba72018fb47ab4475b84fe), proveniente de la huella (1500735243) es nuevo
2014-10-28 23:37:12,239 [DEBUG] [pool-3-thread-2] - Se registra el origen (Origenes(1)ave=13-2-1-1negocioSolicitante=13subSistemaSolicitante=2entidadSolicita
2014-10-28 23:37:12,246 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(1)evento(cbbd8c05c4b5842e3ba
2014-10-28 23:37:12,247 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:37:12,247 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 1
2014-10-28 23:37:12,247 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-28 23:37:12,254 [INFO] [pool-3-thread-2] - E 24944f80afaa8f4cbbd305e2904ffa83346f26b0 1 1 143
2014-10-28 23:37:12,298 [DEBUG] [pool-3-thread-2] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(2)evento(24944f80afaa8f4cbbd305e2904ffa83346f26b0), proveniente de la huella (1500735243) es nuevo
2014-10-28 23:37:12,301 [DEBUG] [pool-3-thread-2] - El evento (24944f80afaa8f4cbbd305e2904ffa83346f26b0), proveniente de la huella (1500735243) es nuevo
2014-10-28 23:37:12,303 [DEBUG] [pool-3-thread-2] - Se registra el origen (Origenes(1)ave=13-2-1-1negocioSolicitante=13subSistemaSolicitante=2entidadSolicita
2014-10-28 23:37:12,311 [INFO] [pool-3-thread-2] - E 24944f80afaa8f4cbbd305e2904ffa83346f26b0 2 1 37
2014-10-28 23:37:12,334 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(bb29539c79ef71f9c7e4ef588e
2014-10-28 23:37:12,355 [DEBUG] [DepachadorDeSolicitudes] - AT cbbd8c05c4b5842e3ba72018fb47ab66768b4fe 1 1 133 109
2014-10-28 23:37:12,356 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(2)evento(24944f80afaa8f4cbbd3
2014-10-28 23:37:12,364 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:37:12,364 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 1
2014-10-28 23:37:12,364 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-28 23:37:12,386 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(7656a970cb295a8cb8202405e
2014-10-28 23:37:12,404 [INFO] [DepachadorDeSolicitudes] - AT 24944f80afaa8f4cbbd305e2904ffa83346f26b0 2 1 85 45
2014-10-28 23:37:42,576 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:37:42,576 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-28 23:37:42,576 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-28 23:37:42,587 [DEBUG] [pool-3-thread-4] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(4)evento(400a31781f9c96e5eaf981436c4ide
2014-10-28 23:37:42,589 [DEBUG] [pool-3-thread-4] - El evento (400a31781f9c96e5eaf981436c4ide760fc6e13), proveniente de la huella (1500735243) es nuevo
2014-10-28 23:37:42,591 [DEBUG] [pool-3-thread-4] - Se registra el origen (Origenes(1)ave=13-2-1-1negocioSolicitante=13subSistemaSolicitante=2entidadSolicita
2014-10-28 23:37:42,594 [INFO] [pool-3-thread-4] - E 400a31781f9c96e5eaf981436c4ide760fc6e13 4 1 7 27
2014-10-28 23:37:42,602 [INFO] [DepachadorDeSolicitudes] - AT caa44d6f7298c465aa503608f0c2faca93b730 3 1 27 27
2014-10-28 23:37:42,602 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(4)evento(400a31781f9c96e5eaf98
2014-10-28 23:37:42,603 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:37:42,603 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-28 23:37:42,603 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-28 23:37:42,607 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(10e1e616f00b58d05d010b2d455
2014-10-28 23:37:42,609 [INFO] [DepachadorDeSolicitudes] - AT 400a31781f9c96e5eaf981436c4ide760fc6e13 4 1 16 7
2014-10-28 23:38:07,430 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:38:07,430 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-28 23:38:07,430 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-28 23:38:07,437 [DEBUG] [pool-3-thread-6] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(6)evento(74a2acfa8735bd3605aa2d01b0729
2014-10-28 23:38:07,440 [DEBUG] [pool-3-thread-6] - El evento (74a2acfa8735bd3605aa2d01b0729d3b81f27), proveniente de la huella (1500735243) es nuevo
2014-10-28 23:38:07,442 [DEBUG] [pool-3-thread-6] - Se registra el origen (Origenes(1)ave=13-2-1-1negocioSolicitante=13subSistemaSolicitante=2entidadSolicita
2014-10-28 23:38:07,445 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(d862296f31fb1848f0fc283a230
2014-10-28 23:38:07,448 [INFO] [pool-3-thread-6] - AT 8388b4e937282726d6d36fefa2904d6e6c9f25 5 1 34 15
2014-10-28 23:38:07,452 [INFO] [pool-3-thread-6] - E 74a2acfa8735bd3605aa2d01b0729d3b81f27 6 1 16 16
2014-10-28 23:38:07,463 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(6)evento(74a2acfa8735bd3605aa2
2014-10-28 23:38:07,463 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:38:07,463 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-28 23:38:07,463 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-28 23:38:07,467 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(0b70749edcb353e6f7630aa232993
2014-10-28 23:38:07,458 [INFO] [DepachadorDeSolicitudes] - AT 74a2acfa8735bd3605aa2d01b0729d3b81f27 6 1 16 5
2014-10-28 23:38:32,661 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:38:32,661 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-28 23:38:32,661 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-28 23:38:32,670 [DEBUG] [pool-3-thread-6] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(8)evento(27c329597d06d144113119796f9a66e
2014-10-28 23:38:32,672 [DEBUG] [pool-3-thread-6] - El evento (27c329597d06d144113119796f9a66e467b76fab), proveniente de la huella (1500735243) es nuevo
2014-10-28 23:38:32,673 [DEBUG] [pool-3-thread-6] - Se registra el origen (Origenes(1)ave=13-2-1-1negocioSolicitante=13subSistemaSolicitante=2entidadSolicita
2014-10-28 23:38:32,675 [INFO] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(0r9c5f2956474b16482425e6e117
2014-10-28 23:38:32,678 [INFO] [pool-3-thread-6] - E 27c329597d06d144113119796f9a66e467b76fab 8 1 5 22
2014-10-28 23:38:32,682 [INFO] [DepachadorDeSolicitudes] - AT 2d47827821aa2c6737ab62d821246a3db5db9c 7 1 13 22
2014-10-28 23:38:32,683 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(8)evento(27c329597d06d144113119
2014-10-28 23:38:32,684 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:38:32,684 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-28 23:38:32,684 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-28 23:38:32,686 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(cc93b59f770371042eda302d654
2014-10-28 23:38:32,692 [INFO] [DepachadorDeSolicitudes] - AT 13119796f9a66e467b76fab 8 1 18 9
2014-10-28 23:38:40,108 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:38:40,109 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 3
2014-10-28 23:38:40,109 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-28 23:38:40,131 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(bb6d8e1c265e6978274bedf01e8
2014-10-28 23:38:40,134 [DEBUG] [pool-3-thread-6] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(10)evento(ba7abb6d9f4e57620cd5dc4506c80
2014-10-28 23:38:40,136 [DEBUG] [pool-3-thread-6] - El evento (ba7abb6d9f4e57620cd5dc4506c80c020f7bd9014), proveniente de la huella (1500735243) es nuevo
2014-10-28 23:38:40,136 [DEBUG] [pool-3-thread-6] - Se registra el origen (Origenes(1)ave=13-2-1-1negocioSolicitante=13subSistemaSolicitante=2entidadSolicita
2014-10-28 23:38:40,139 [INFO] [DepachadorDeSolicitudes] - AT 83594d0e4e18956442af8875e4b8ec67f07b5d 9 1 18 38
2014-10-28 23:38:40,137 [INFO] [pool-3-thread-6] - E ba7abb6d9f4e57620cd5dc4506c80c020f7bd9014 10 1 13 13
2014-10-28 23:38:40,140 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(10)evento(ba7abb6d9f4e57620cd5d
2014-10-28 23:38:40,142 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:38:40,142 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 3
2014-10-28 23:38:40,142 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-28 23:38:40,143 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(9)evento(bb6d8e1c265e6978274bedf01e8
2014-10-28 23:38:40,147 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(41d27bd9a7164eef8f83e0f76240
2014-10-28 23:38:40,157 [INFO] [DepachadorDeSolicitudes] - AT ba7abb6d9f4e57620cd5dc4506c80c020f7bd9014 10 1 16 17
```

```

2014-10-28 23:40:16,030 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-28 23:40:16,030 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:40:16,030 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 4
2014-10-28 23:40:16,030 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-28 23:40:16,029 [INFO] [pool-3-thread-6] - E 4608ca30e0975fc5cafb794bb28d7d67e728cfa 11 1 50
2014-10-28 23:40:16,048 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(174db9cc92cd361c046c0d42898
2014-10-28 23:40:16,047 [DEBUG] [pool-3-thread-6] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(12)evento(063cbf246079bc4868f66fb182674a
2014-10-28 23:40:16,048 [DEBUG] [pool-3-thread-6] - El evento (063cbf246079bc4868f66fb182674a7efc605671), proveniente de la huella (1500736243) es nuevo
2014-10-28 23:40:16,049 [DEBUG] [pool-3-thread-6] - Se registra el origen (Origenes(1)llave=13-2-1-InegocioSolicitante=13subSistemaSolicitante=2entidadSolicita
2014-10-28 23:40:16,060 [INFO] [DepachadorDeSolicitudes] - AT 4608ca30e0975fc5cafb794bb28d7d67e728cfa 11 1 50 21
2014-10-28 23:40:16,064 [INFO] [pool-3-thread-6] - E 063cbf246079bc4868f66fb182674a7efc605671 12 1 8
2014-10-28 23:40:16,066 [DEBUG] [DepachadorDeSolicitudes] - Depachador toma el ma para atender (numeroServicio(1)secuencial(12)evento(063cbf246079bc4868f66
2014-10-28 23:40:16,066 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-28 23:40:16,066 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:40:16,066 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 4
2014-10-28 23:40:16,066 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-28 23:40:16,066 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(4aa4a3693d6599fd91f0ed277436
2014-10-28 23:40:16,066 [INFO] [DepachadorDeSolicitudes] - AT 063cbf246079bc4868f66fb182674a7efc605671 12 1 9 11
2014-10-28 23:44:49,648 [DEBUG] [DepachadorDeSolicitudes] - Depachador toma el ma para atender (numeroServicio(1)secuencial(13)evento(13660a0063014d9866d36
2014-10-28 23:44:49,649 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-28 23:44:49,649 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:44:49,649 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 6
2014-10-28 23:44:49,649 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-28 23:44:49,666 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(32e821036e12b115883779e9162d
2014-10-28 23:44:49,672 [DEBUG] [pool-3-thread-8] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(14)evento(41217c97fde49bac694b3bfeaf33a1
2014-10-28 23:44:49,673 [INFO] [DepachadorDeSolicitudes] - AT 13660a0063014d9866d3692c6c79bb75ddf08bfb 13 1 14 26
2014-10-28 23:44:49,677 [DEBUG] [pool-3-thread-8] - El evento (41217c97fde49bac694b3bfeaf33a16faf793fb4), proveniente de la huella (1500736243) es nuevo
2014-10-28 23:44:49,678 [DEBUG] [pool-3-thread-8] - Se registra el origen (Origenes(1)llave=13-2-1-InegocioSolicitante=13subSistemaSolicitante=2entidadSolicita
2014-10-28 23:44:49,683 [INFO] [pool-3-thread-8] - E 41217c97fde49bac694b3bfeaf33a16faf793fb4 14 12
2014-10-28 23:44:49,685 [DEBUG] [DepachadorDeSolicitudes] - Depachador toma el ma para atender (numeroServicio(1)secuencial(14)evento(41217c97fde49bac694b3
2014-10-28 23:44:49,687 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-28 23:44:49,687 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 1
2014-10-28 23:44:49,687 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 5
2014-10-28 23:44:49,687 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-28 23:44:49,686 [DEBUG] [Enviador(ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(13)evento(32e821036e12b115883779e9162d
2014-10-28 23:44:49,696 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(b6c0862234060072c127b1aae21
2014-10-28 23:44:49,707 [INFO] [DepachadorDeSolicitudes] - AT 41217c97fde49bac694b3bfeaf33a16faf793fb4 14 1 14 22

```

*Figura 5.6 Bitácora del gestor para el sistema monolítico*

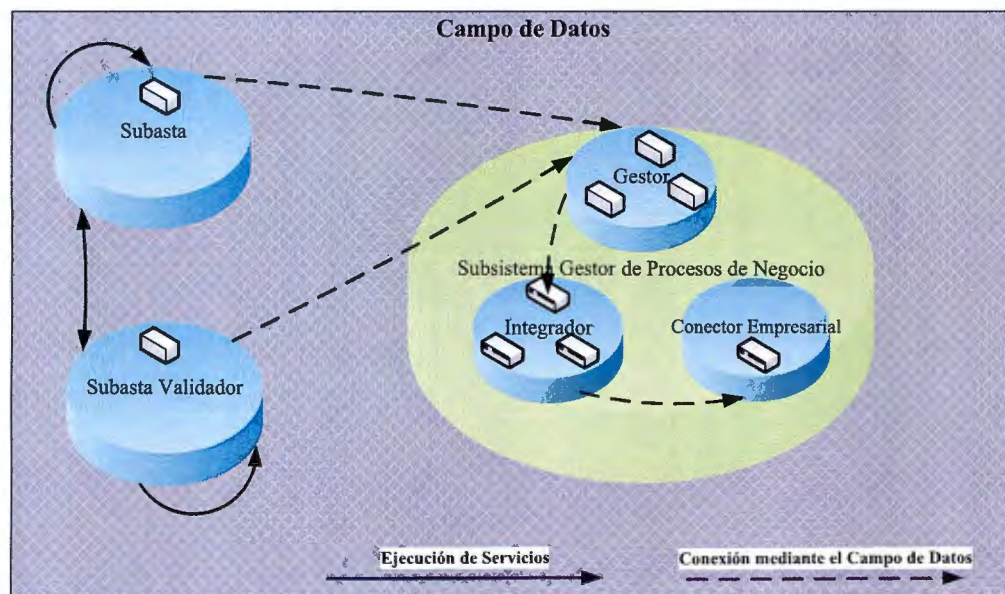
### 5.6.1.2. Resultados del Caso de Prueba I

Se pudo demostrar que la integración del sistema gestor en una arquitectura tradicional donde una sola Entidad solicita y ofrece los servicios a utilizar es transparente, permitiendo llevar el seguimiento del proceso de negocio sin dificultades, proporcionando información oportuna y veraz. Además de visualizar cada etapa del proceso de manera clara.

## 5.6.2. Caso de Prueba II

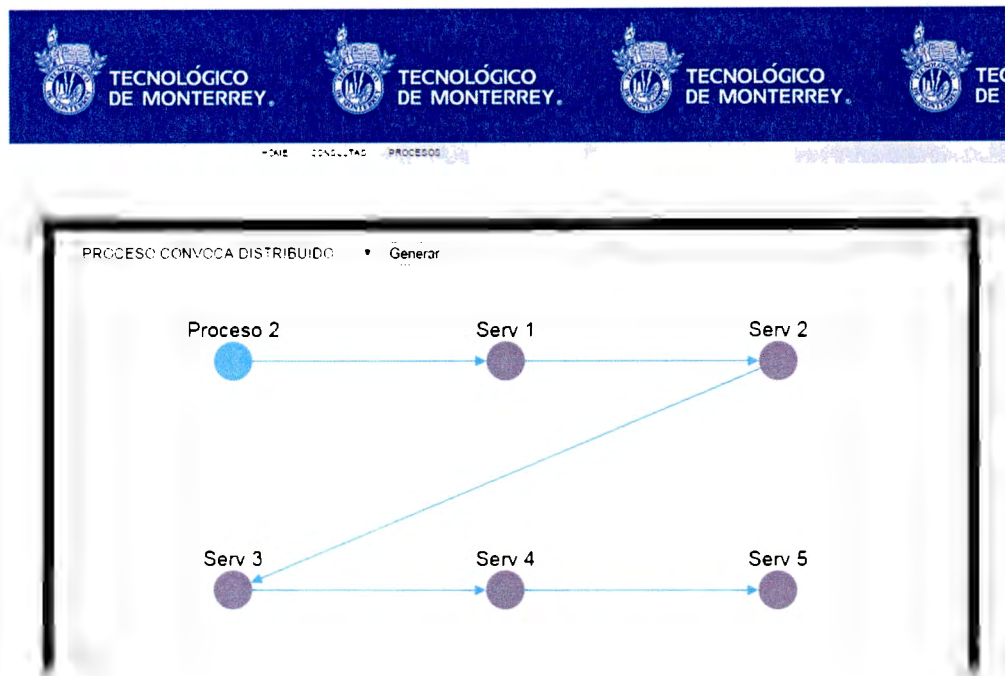
### 5.6.2.1. Sistema Distribuido

En esta prueba se utiliza una arquitectura distribuida, donde los servicios a utilizar se dividieron y configuraron en dos Entidades. La Entidad Subasta se encarga de iniciar la convocatoria, de recibir las posturas y de realizar el cierre de la subasta. La Entidad Subasta Validador tiene la tarea de realizar la asignación y la publicación. La figura 5.7 muestra la arquitectura utilizada.



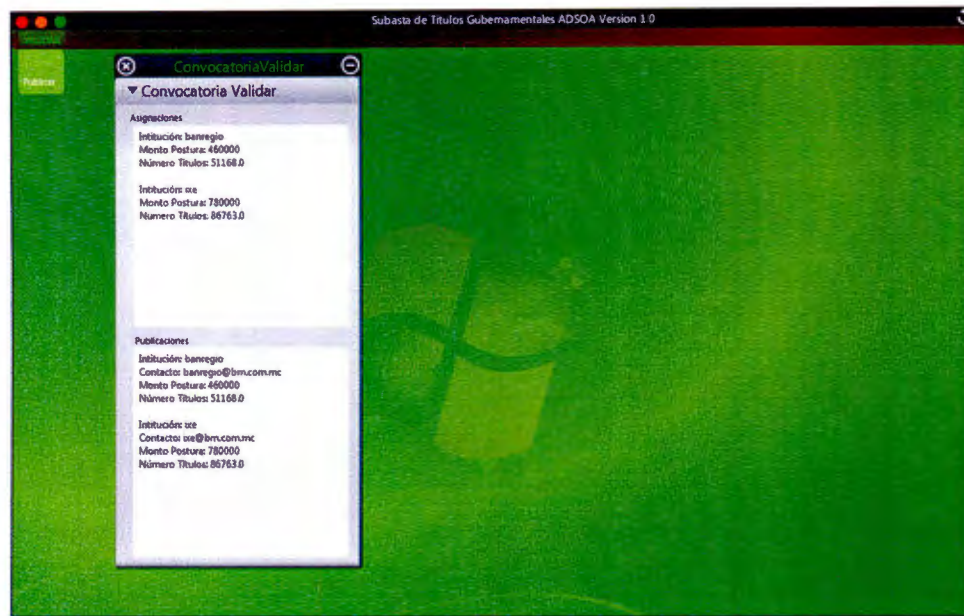
*Figura 5.7 Arquitectura del sistema distribuido*

Para esta prueba se demuestra la implementación del sistema gestor en esta arquitectura distribuida, realizando el monitoreo del proceso de negocio aún cuando existan dos Entidades separadas de manera lógica. La imagen 5.8 muestra cómo se visualiza el inicio del proceso.



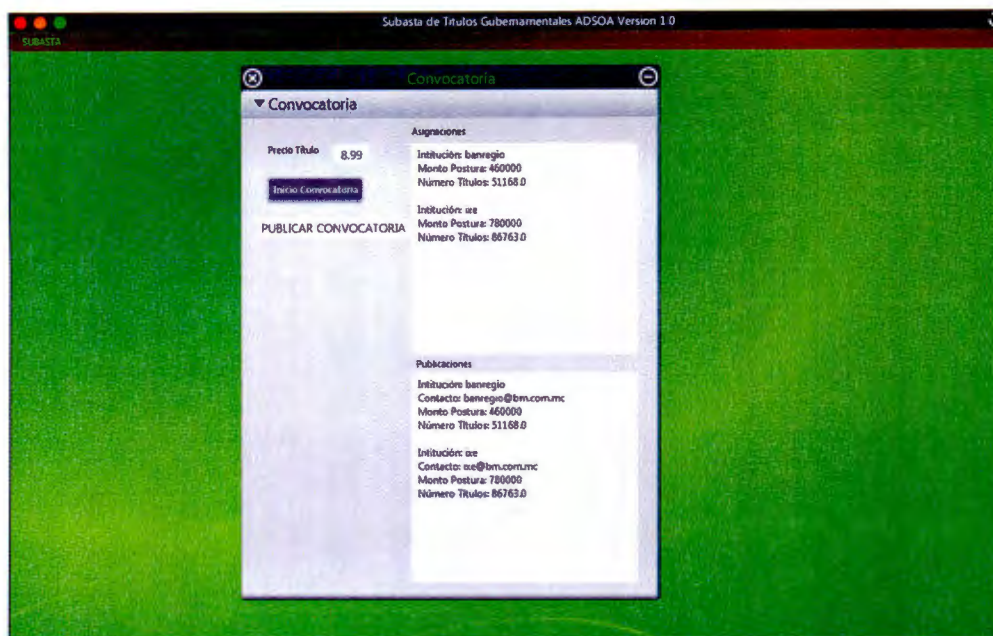
***Figura 5.8 Grafo inicial del sistema distribuido***

Las figura 5.9 muestra a la Entidad Subasta Validador encargada de la asignación y publicación de los resultados de la subasta. En este caso la subasta llegó a la etapa de publicación.



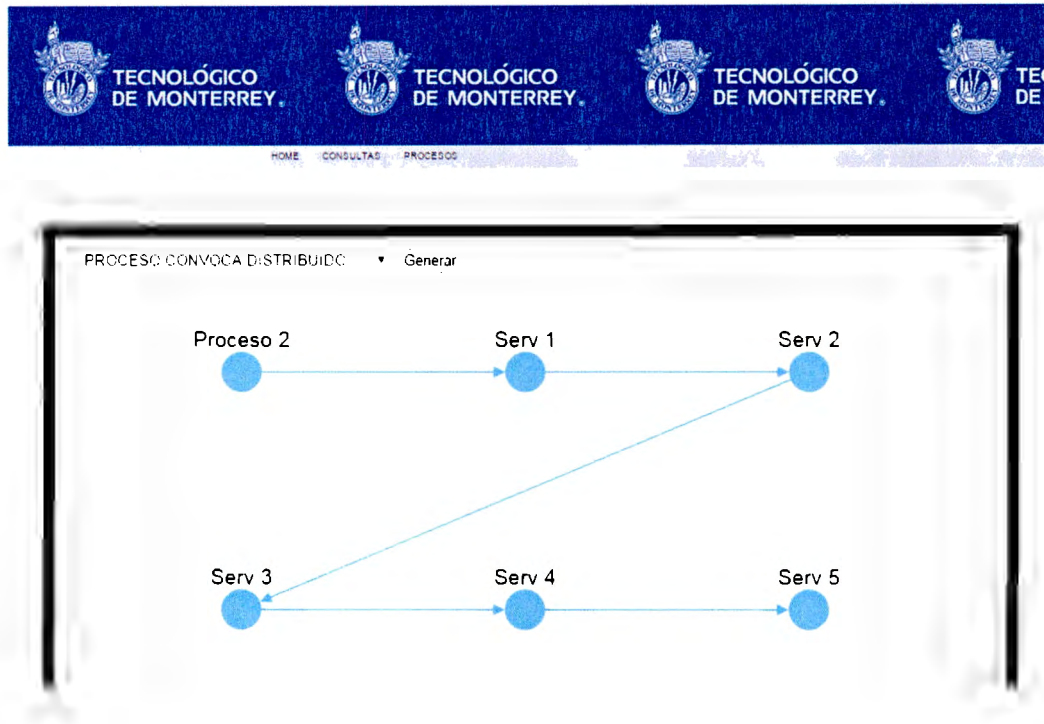
**Figura 5.9 Entidad subasta validador del sistema distribuido**

En la imagen 5.10 se observa a la Entidad Subasta también en la etapa de publicación, se visualiza la misma información que en la Entidad Subasta Validador, lo que indica que existe consistencia en la información.



*Figura 5.10 Entidad subasta del sistema distribuido*

En la figura 5.11, se puede observar que el grafo llegó a la etapa final, lo que indica que fue exitosa la ejecución del proceso de negocio.



*Figura 5.11 Grafo del sistema distribuido en la etapa de publicación*

Para respaldar lo anterior, se realizó la consulta a la base de datos del proceso actual y se obtuvo el siguiente resultado.

HOME CONSULTAS PROCESOS

TECNOLOGICO DE MONTERREY. TECNOLOGICO DE MONTERREY. TECNOLOGICO DE MONTERREY. TECNOLOGICO DE MONTERREY.

PROCESO CONVOCADA DISTRIBUIDO Selecionar Servicio 2014-10-29

2014-10-30

Proceso	Descripción Proceso	Servicio	Descripción Servicio	Estado	Fecha Operación	Fecha Ejecución Servicio
PROCESO CONVOCADA DISTRIBUIDO		5	PUBLICAR	FINAL	2014-10-29	2014-10-29 12:27:16 AM
PROCESO CONVOCADA DISTRIBUIDO		5	PUBLICAR	FINAL	2014-10-29	2014-10-29 12:27:16 AM
PROCESO CONVOCADA DISTRIBUIDO		5	PUBLICAR	FINAL	2014-10-29	2014-10-29 12:27:16 AM
PROCESO CONVOCADA DISTRIBUIDO		5	PUBLICAR	FINAL	2014-10-29	2014-10-29 12:27:16 AM
2	PROCESO CONVOCADA DISTRIBUIDO	5	PUBLICAR	PRMAL	2014-10-29	Oct 29, 2014 12:27:16 AM

1 de 7 registros por página. 10 registros por página. Muestre registros 1 a 7 de 7

**Figura 5.12 Consulta de la base de datos del sistema distribuido**

Como se puede observar, esta consulta muestra la descripción de la ejecución del proceso de negocio. Se detalla la hora de ejecución de los servicios, así como el estado en el que se encuentran.

En cuanto al gestor, éste registra en su bitácora los cambios de estado de los servicios gestionables, llevando el seguimiento de los mismos. En la figura 5.13 se muestra la bitácora.



```
2014-10-29 00:21:14,541 [INFO ] [main] - ***Servidor Gestor Operando
2014-10-29 00:26:56,491 [DEBUG] [pool-3-thread-2] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(1)evento(b7834f707b4fd46fb6764
2014-10-29 00:26:56,526 [DEBUG] [pool-3-thread-2] - El evento (b7834f707b4fd46fb676463a412ea63d91cae731), proveniente de la huella (82826701) es n
2014-10-29 00:26:56,533 [DEBUG] [pool-3-thread-2] - Se registra el origen (Origenes(llave=13-2-1-1negocioSolicitante=13subSistemaSolicitante=2entid
2014-10-29 00:26:56,541 [INFO ] [pool-3-thread-2] - E b7834f707b4fd46fb676463a412ea63d91cae731 1 1 66
2014-10-29 00:26:56,570 [DEBUG] [pool-3-thread-2] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(2)evento(271c50b4871daef11ae75
2014-10-29 00:26:56,573 [DEBUG] [pool-3-thread-2] - El evento (271c50b4871daef11ae7590ae62dic3e61b05567), proveniente de la huella (82826701) es n
2014-10-29 00:26:56,576 [DEBUG] [pool-3-thread-2] - Se registra el origen (Origenes(llave=13-2-1-1negocioSolicitante=13subSistemaSolicitante=2entid
2014-10-29 00:26:56,578 [INFO ] [pool-3-thread-2] - E 271c50b4871daef11ae7590ae62dic3e61b05567 2 1 9
2014-10-29 00:26:56,617 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(1)evento(b7834f707b4f
2014-10-29 00:26:56,619 [INFO ] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(a32cd4710d0e025d43
2014-10-29 00:26:56,619 [INFO ] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:26:56,619 [INFO ] [DepachadorDeSolicitudes] - ***Servicio: 1
2014-10-29 00:26:56,620 [INFO ] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 00:26:56,620 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(a32cd4710d0e025d43
2014-10-29 00:26:56,700 [INFO ] [DepachadorDeSolicitudes] - AT b7834f707b4fd46fb676463a412ea63d91cae731 1 1 140 86
2014-10-29 00:26:56,704 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(2)evento(271c50b4871d
2014-10-29 00:26:56,704 [INFO ] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(d71cb0ce7703b39cad
2014-10-29 00:26:56,705 [INFO ] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:26:56,705 [INFO ] [DepachadorDeSolicitudes] - ***Servicio: 1
2014-10-29 00:26:56,705 [INFO ] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 00:26:56,718 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(d71cb0ce7703b39cad
2014-10-29 00:26:56,726 [INFO ] [DepachadorDeSolicitudes] - AT 271c50b4871daef11ae7590ae62dic3e61b05567 2 1 135 22
2014-10-29 00:26:13,863 [INFO ] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:26:13,863 [INFO ] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-29 00:26:13,864 [INFO ] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 00:26:13,866 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(fbf7b10026c7be988ff
2014-10-29 00:26:13,877 [INFO ] [DepachadorDeSolicitudes] - AT 138bb2b7e253c388ed5fd8af1b495f47e4f4c44 3 1 23 15
2014-10-29 00:26:13,895 [DEBUG] [pool-3-thread-3] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(4)evento(ab1ee5a8429d2ed4fac58
2014-10-29 00:26:13,899 [DEBUG] [pool-3-thread-3] - El evento (ab1ee5a8429d2ed4fac5842f0b8a062f397999cd), proveniente de la huella (82826701) es n
2014-10-29 00:26:13,899 [DEBUG] [pool-3-thread-3] - Se registra el origen (Origenes(llave=13-2-1-1negocioSolicitante=13subSistemaSolicitante=2entid
2014-10-29 00:26:13,901 [INFO ] [pool-3-thread-3] - E ab1ee5a8429d2ed4fac5842f0b8a062f397999cd 4 1 6
2014-10-29 00:26:13,906 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(4)evento(ab1ee5a8429d
2014-10-29 00:26:13,906 [INFO ] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(7fb7b10026c7be988ff
2014-10-29 00:26:13,906 [INFO ] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:26:13,906 [INFO ] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-29 00:26:13,906 [INFO ] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 00:26:13,908 [DEBUG] [Enviador (ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(3)evento(fbf7b10026c7be988ff
2014-10-29 00:26:13,946 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(7fb7b10026c7be988ff
2014-10-29 00:26:14,005 [INFO ] [DepachadorDeSolicitudes] - AT ab1ee5a8429d2ed4fac5842f0b8a062f397999cd 4 1 11 99
2014-10-29 00:26:33,237 [INFO ] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:26:33,237 [INFO ] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-29 00:26:33,237 [INFO ] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 00:26:33,242 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(1cea0127ab6de99a939
2014-10-29 00:26:33,247 [INFO ] [DepachadorDeSolicitudes] - AT 0299adae21dcb01c3842fb78a405da19266c0af 5 1 50 13
2014-10-29 00:26:33,249 [DEBUG] [Enviador (ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(5)evento(1cea0127ab6de99a939
2014-10-29 00:26:33,256 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(6)evento(61f2d711397d
2014-10-29 00:26:33,259 [INFO ] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:26:33,260 [INFO ] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-29 00:26:33,261 [INFO ] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 00:26:33,262 [INFO ] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(1cea0127ab6de99a939
2014-10-29 00:26:33,264 [INFO ] [pool-3-thread-3] - S 1ced01978b3d99393505138ad1c009c98666 5 1 8 17
2014-10-29 00:26:33,272 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(0bcd0e30b5dff87047
2014-10-29 00:26:33,277 [INFO ] [DepachadorDeSolicitudes] - AT 61f2d711397d9fd0146316f280a5e5fcfc7e1ab4 6 1 33 21
2014-10-29 00:26:52,216 [INFO ] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:26:52,216 [INFO ] [DepachadorDeSolicitudes] - ***Servicio: 3
2014-10-29 00:26:52,216 [INFO ] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 00:26:52,252 [DEBUG] [pool-3-thread-2] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(2)evento(c99cd4dfab13d8ec9c7
2014-10-29 00:26:52,254 [DEBUG] [pool-3-thread-2] - El evento (c99cd4dfab13d8ec9c7c8f69021862b1a79fa3b), proveniente de la huella (1196736335) es
2014-10-29 00:26:52,254 [DEBUG] [pool-3-thread-2] - Se registra el origen (Origenes(llave=10-1-2-1negocioSolicitante=10subSistemaSolicitante=1entid
2014-10-29 00:26:52,286 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(61f2d7078136c31
2014-10-29 00:26:52,296 [INFO ] [pool-3-thread-2] - E c99cd4dfab13d8ec9c7c8f69021862b1a79fa3b 2 1 44
2014-10-29 00:26:52,339 [INFO ] [DepachadorDeSolicitudes] - AT c1cd0da5ae886c08d8a23ff5192d5f9728b896d 1 1 13 126
2014-10-29 00:26:52,342 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el msg para atender (numeroServicio(1)secuencial(2)evento(c99cd4dfab:
2014-10-29 00:26:52,343 [INFO ] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:26:52,343 [INFO ] [DepachadorDeSolicitudes] - ***Servicio: 3
2014-10-29 00:26:52,343 [INFO ] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 00:26:52,350 [DEBUG] [Enviador (ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(7)evento(61f2d7078136c31
2014-10-29 00:26:52,356 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(98f49924b261a2fdd:
2014-10-29 00:26:52,379 [INFO ] [DepachadorDeSolicitudes] - AT c99cd4dfab13d8ec9c7c8f69021862b1a79fa3b 2 1 90 37
```

```

2014-10-29 00:27:06,507 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-29 00:27:06,507 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:27:06,507 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 4
2014-10-29 00:27:06,507 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-29 00:27:06,516 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio:1)secuencial(0)evento(iabce37fec2f16d1b
2014-10-29 00:27:06,518 [INFO] [DepachadorDeSolicitudes] - AT ef056f0be88012b16efdf31cadled507719840ac7 7 1 8 12
2014-10-29 00:27:06,532 [DEBUG] [Enviador (ACP)] - Enviador envia el mensaje transaccional (numeroServicio(1)secuencial(9)evento(iabce37fec2f16d1bb
2014-10-29 00:27:06,610 [INFO] [pool-3-thread-3] - S iabce37fec2f16d1bb2137234151297174e7080 9 1 12 86
2014-10-29 00:27:06,655 [DEBUG] [pool-3-thread-2] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(8)evento(i6c9952127a4f5e21bdb0
2014-10-29 00:27:06,656 [DEBUG] [pool-3-thread-2] - El evento i6c9952127a4f5e21bdb044784ec2fbd2a2cf6e2, proveniente de la huella i382826701 es n
2014-10-29 00:27:06,667 [DEBUG] [pool-3-thread-2] - Se registra el origen (Origenes(llave=13-2-1-lnegocioSolicitante=13subSistemaSolicitante=2entid
2014-10-29 00:27:06,664 [INFO] [pool-3-thread-2] - E i6c9952127a4f5e21bdb044784ec2fbd2a2cf6e2 8 1 9
2014-10-29 00:27:06,667 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el mas para atender (numeroServicio(1)secuencial(8)evento(i6c9952127a4
2014-10-29 00:27:06,670 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-29 00:27:06,670 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:27:06,670 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 4
2014-10-29 00:27:06,670 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 00:27:06,724 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio:1)secuencial(0)evento(86ff6d060f2df1e9f2
2014-10-29 00:27:06,751 [DEBUG] [Enviador (ACP)] - Enviador envia el mensaje transaccional (numeroServicio(1)secuencial(10)evento(86ff6d060f2df1e9f2
2014-10-29 00:27:06,764 [INFO] [DepachadorDeSolicitudes] - AT i6c9952127a4f5e21bdb044784ec2fbd2a2cf6e2 8 1 12 97
2014-10-29 00:27:16,226 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-29 00:27:16,226 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:27:16,226 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 5
2014-10-29 00:27:16,226 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-29 00:27:16,229 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio:1)secuencial(0)evento(a512665094eddc3e7
2014-10-29 00:27:16,233 [INFO] [DepachadorDeSolicitudes] - AT 51e0112124d34e94f98cd2fd34971f4101e928a3f 9 1 12 7
2014-10-29 00:27:16,247 [DEBUG] [Enviador (ACP)] - Enviador envia el mensaje transaccional (numeroServicio(1)secuencial(11)evento(a512665094eddc3e7
2014-10-29 00:27:16,271 [INFO] [pool-3-thread-3] - S a512665094eddc3e739d0fd5ff4a40fbae29125 11 1 13 31
2014-10-29 00:27:16,302 [DEBUG] [pool-3-thread-3] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(10)evento(adf262950bc4203e6b68
2014-10-29 00:27:16,303 [DEBUG] [pool-3-thread-3] - El evento (adf262950bc4203e6b68ef8197180e97c3beae43), proveniente de la huella i382826701 es n
2014-10-29 00:27:16,303 [DEBUG] [pool-3-thread-3] - Se registra el origen (Origenes(llave=13-2-1-lnegocioSolicitante=13subSistemaSolicitante=2entid
2014-10-29 00:27:16,307 [INFO] [pool-3-thread-3] - E adf262950bc4203e6b68ef8197180e97c3beae43 10 1 6
2014-10-29 00:27:16,308 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el mas para atender (numeroServicio(1)secuencial(10)evento(adf262950bc
2014-10-29 00:27:16,309 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-29 00:27:16,309 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 2
2014-10-29 00:27:16,309 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 6
2014-10-29 00:27:16,309 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 00:27:16,314 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio:1)secuencial(0)evento(f0da65994e7861b887f
2014-10-29 00:27:16,316 [INFO] [DepachadorDeSolicitudes] - AT adf262950bc4203e6b68ef8197180e97c3beae43 10 1 7 8

```

Figura 5.13 Bitácora del gestor para el sistema distribuido

### 5.6.2.2. Resultados del Caso de Prueba II

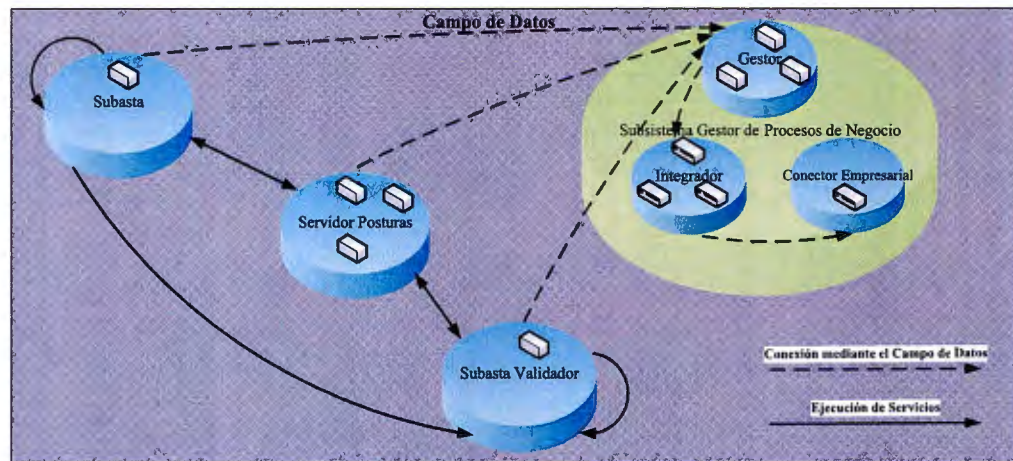
Se demostró que la integración del sistema gestor en una arquitectura distribuida es sencilla y segura, permitiendo llevar el monitoreo de los procesos de negocio aún cuando se tengan configurados los servicios en Entidades separadas de manera lógica. Además de visualizar cada etapa del proceso de manera fácil.

### **5.6.3. Caso de Prueba III**

#### **5.6.3.1. Sistema Tolerante a Fallos**

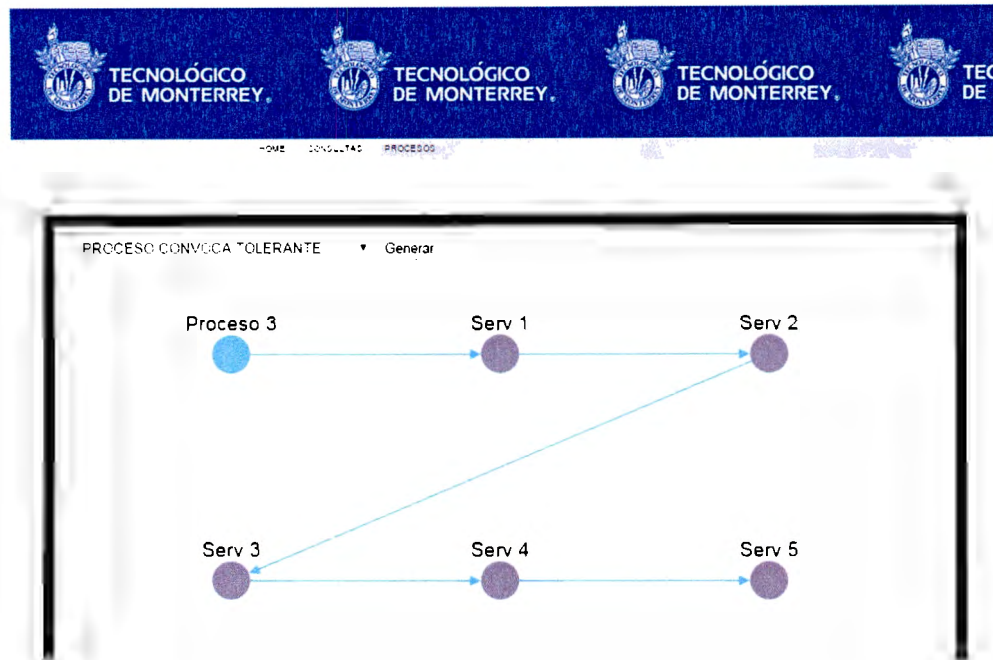
Para esta prueba se utiliza una arquitectura distribuida, pero tiene la característica de ser también tolerante a fallas. Para demostrar esta característica, se utilizan tres Entidades. La Entidad Subasta se encarga de iniciar la convocatoria y del cierre de la misma. El Servidor Posturas se encarga de la recepción de las posturas. La Entidad Servidor Validador es encargada de solicitar la asignación y publicación de las posturas recibidas. La figura 5.14 muestra su arquitectura utilizada.

Para cumplir con la característica de tolerancia a fallas, la Entidad Subasta se configura para que sólo acepte el servicio de asignación y de publicación, al recibir solamente dos solicitudes para este fin. En otras palabras, si sólo recibe un mensaje de solicitud para la asignación o publicación no es tomado como completo, necesita recibir otra solicitud para ser válido y esto se hace teniendo una Instancia más del Servidor Posturas.



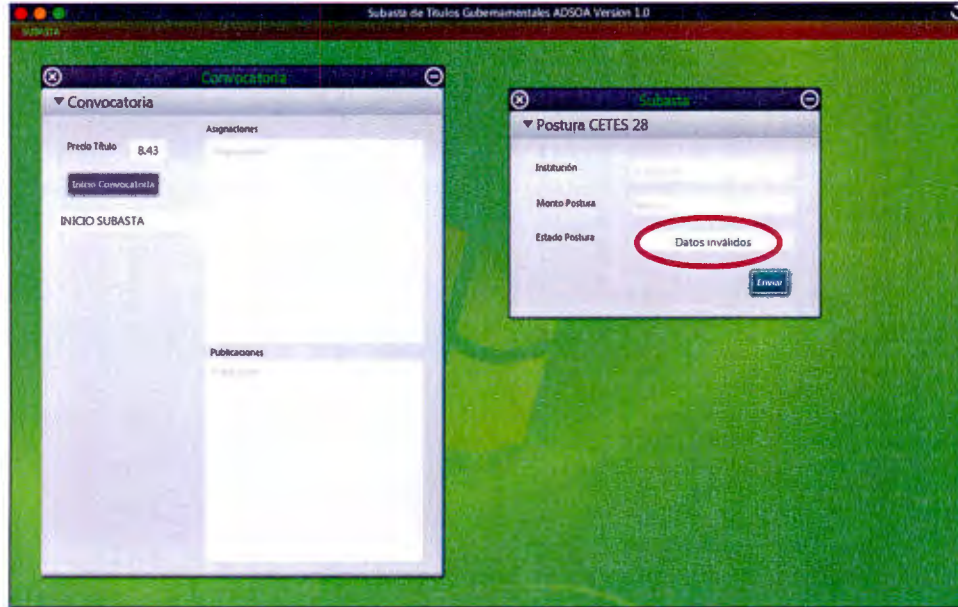
**Figura 5.14** Arquitectura del sistema tolerante a fallas

Para esta prueba se demuestra la implementación del sistema gestor en esta arquitectura distribuida y tolerante a fallas, realizando el monitoreo del proceso de negocio aún cuando existan tres Entidades separadas de manera lógica. La imagen 5.14 muestra cómo se visualiza el inicio del proceso.



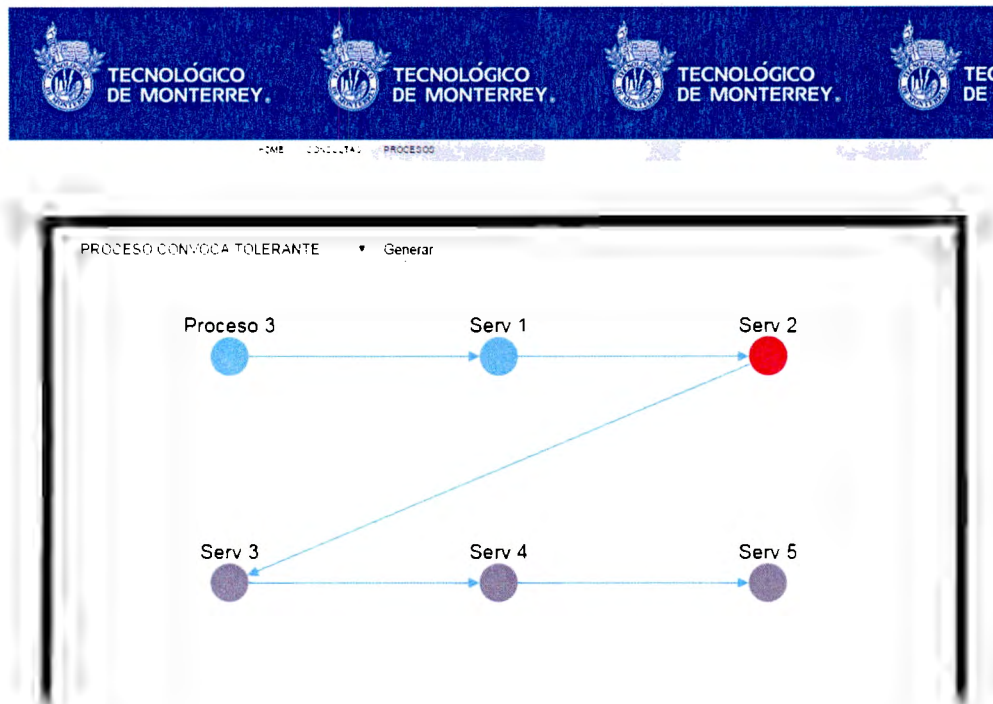
*Figura 5.15 Grafo inicial del sistema tolerante a fallas*

Para esta prueba, se integra un evento de error, este se realiza en la etapa de apertura de la convocatoria, al recibir una postura con datos erróneos. La figura 5.16 indica que se ingresaron datos inválidos para una postura.



**Figura 5.16 Entidad Subasta del sistema tolerante a fallas**

A continuación se observa que la aplicación indica que en esta etapa existe un error y el grafo lo dibuja en rojo, figura 5.17.



***Figura 5.17 Error en la etapa de convocatoria del sistema tolerante a fallas***

Se realizó la consulta de la base de datos hasta ese momento y se observa que efectivamente el evento de error se registró, ver figura 5.18

TECNOLÓGICO DE MONTERREY. TECNOLÓGICO DE MONTERREY. TECNOLÓGICO DE MONTERREY. TEC DE

HOME CONSULTAS PROCESOS

PROCESO CONVOCA TOLERANTE Selecionar Servicio 2014-10-29

2014-10-30 Buscar

Id	Proceso	Descripción Proceso	Servicio	Descripción Servicio	Estado	Fecha Operación	Fecha Ejecución Servicio
1	PROCESO CONVOCA TOLERANTE	PROCESO CONVOCA TOLERANTE	1	POSTURAS	ERROR	2014-10-29	Oct 29, 2014 4:55:11 PM
2	PROCESO CONVOCA TOLERANTE	PROCESO CONVOCA TOLERANTE	1	POSTURAS	FINAL	2014-10-29	Oct 29, 2014 4:55:14 PM
3	PROCESO CONVOCA TOLERANTE	PROCESO CONVOCA TOLERANTE	2	POSTURAS	ERROR	2014-10-29	Oct 29, 2014 5:00:52 PM

1 de 4 páginas 1 Registros por página. 1/1 Muestre registros 1 a 4 de 4

**Figura 5.18 Registro del evento erróneo en la base de datos**

También se muestra este evento de error en la bitácora del Servidor Posturas, que indica la validación de los datos erróneos. En este caso se ingresó un valor no numérico para el monto de la postura.



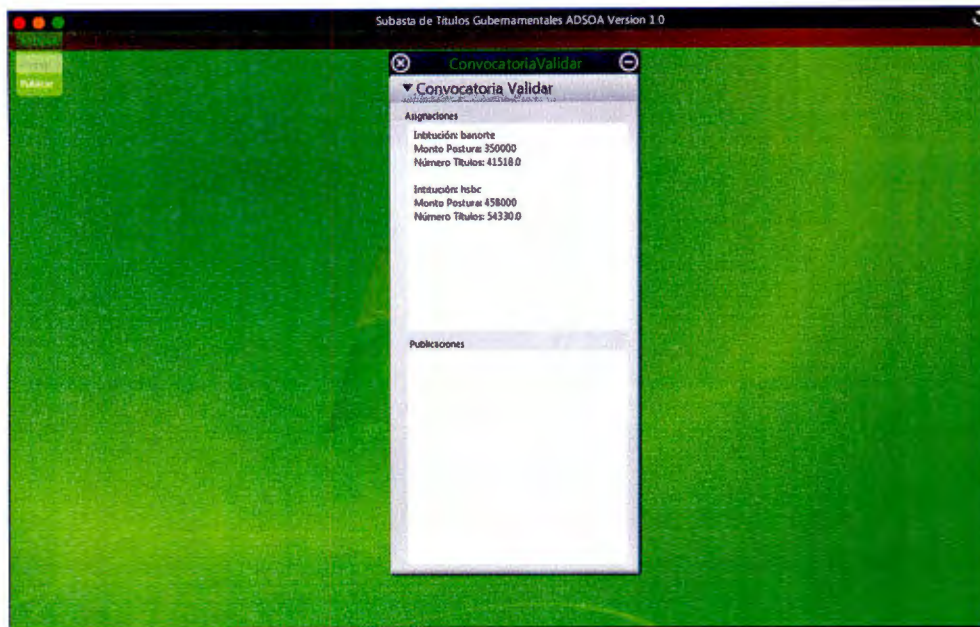
```

C:\Windows\system32\cmd.exe
2014-10-29 17:00:52,529 [INFO ] [pool-3-thread-4] - E 1efab1c7899f6d1af88f2286
7bca3311d4145b80 2 2 70
2014-10-29 17:00:52,532 [INFO ] [DepachadorDeSolicitudes] - ***Se manda mensaje
al Gestor***
2014-10-29 17:00:52,534 [INFO ] [DepachadorDeSolicitudes] - ***idProceso: 3
2014-10-29 17:00:52,535 [INFO ] [DepachadorDeSolicitudes] - ***idServicio: 2
2014-10-29 17:00:52,535 [INFO ] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-29 17:00:52,536 [INFO ] [DepachadorDeSolicitudes] - ***tipoServicio: RES
PUESTA_GESTOR
2014-10-29 17:00:52,537 [INFO ] [DepachadorDeSolicitudes] - ***tipoPivote: PIVOTE
E_MONITOREO
2014-10-29 17:00:52,537 [INFO ] [DepachadorDeSolicitudes] - ***NEvento: 2d0243b3
8ace997c339be444a19502fe415faae2
2014-10-29 17:00:52,540 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF
el mensaje (numeroServicio(1)secuencial(0)evento(2d0243b38ace997c339be444a19502f
e415faae2)solicitante(10.1.3)pivote(1)tiempoCreacion(1414623652538)tiempoEnvio(0
)idMensaje(3)huella(1571433370)), para el destinatario (10.1.5) y el numero de s
ervicio (1)
2014-10-29 17:00:52,542 [DEBUG] [DepachadorDeSolicitudes] - Despachador tomo el
msj para atender (numeroServicio(2)secuencial(2)evento(1efab1c7899f6d1af88f22867
bca3311d4145b80)solicitante(13.2.1)pivote(1)tiempoCreacion(1414623652459)tiempoE
nvio(0)idMensaje(2)huella(362539572))
2014-10-29 17:00:52,543 [INFO ] [DepachadorDeSolicitudes] - ***SE RECIBIO LA POS
TURA***
2014-10-29 17:00:52,543 [INFO ] [DepachadorDeSolicitudes] - ***Institucion: bana
mex
2014-10-29 17:00:52,544 [INFO ] [DepachadorDeSolicitudes] - ***monto: 756000
2014-10-29 17:00:52,544 [INFO ] [DepachadorDeSolicitudes] - ***SE ENVIA EL ERROR
DE LA POSTURA***
2014-10-29 17:00:52,555 [DEBUG] [Enviador(ACP)] - Enviador envia el mensaje tran
saccional (numeroServicio(1)secuencial(3)evento(2d0243b38ace997c339be444a19502fe
415faae2)solicitante(10.1.3)pivote(1)tiempoCreacion(1414623652538)tiempoEnvio(14
14623652547)idMensaje(3)huella(1571433370))
2014-10-29 17:00:52,561 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF
el mensaje (numeroServicio(11)secuencial(0)evento(7b184874ec17f2dcddaf84f316a37e1
f258ffddb)solicitante(10.1.3)pivote(1)tiempoCreacion(1414623652551)tiempoEnvio(
0)idMensaje(4)huella(1571433370)), para el destinatario (13.2.1) y el numero de
servicio (11)
2014-10-29 17:00:52,562 [INFO ] [DepachadorDeSolicitudes] - AT 1efab1c7899f6d1a
f88f22867bca3311d4145b80 2 72 31
2014-10-29 17:00:52,563 [INFO ] [DepachadorDeSolicitudes] - ***Se manda mensaje
al Gestor***
2014-10-29 17:00:52,563 [INFO ] [DepachadorDeSolicitudes] - ***idProceso: 3
2014-10-29 17:00:52,566 [INFO ] [DepachadorDeSolicitudes] - ***idServicio: 2
2014-10-29 17:00:52,566 [INFO ] [DepachadorDeSolicitudes] - ***Estado: ERROR
2014-10-29 17:00:52,567 [INFO ] [DepachadorDeSolicitudes] - ***tipoServicio: RES
PUESTA_GESTOR
2014-10-29 17:00:52,567 [INFO ] [DepachadorDeSolicitudes] - ***tipoPivote: PIVOTE
E_MONITOREO
2014-10-29 17:00:52,567 [INFO ] [DepachadorDeSolicitudes] - ***NEvento: 72e48114
0891802703de53f8bc425b416c818649
2014-10-29 17:00:52,579 [DEBUG] [Enviador(ACP)] - Enviador envia el mensaje tran
saccional (numeroServicio(11)secuencial(1)evento(7b184874ec17f2dcddaf84f316a37e1
f258ffddb)solicitante(10.1.3)pivote(1)tiempoCreacion(1414623652551)tiempoEnvio(1
414623652572)idMensaje(4)huella(1571433370))
2014-10-29 17:00:52,580 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF
el mensaje (numeroServicio(1)secuencial(0)evento(72e481140891802703de53f8bc425b4

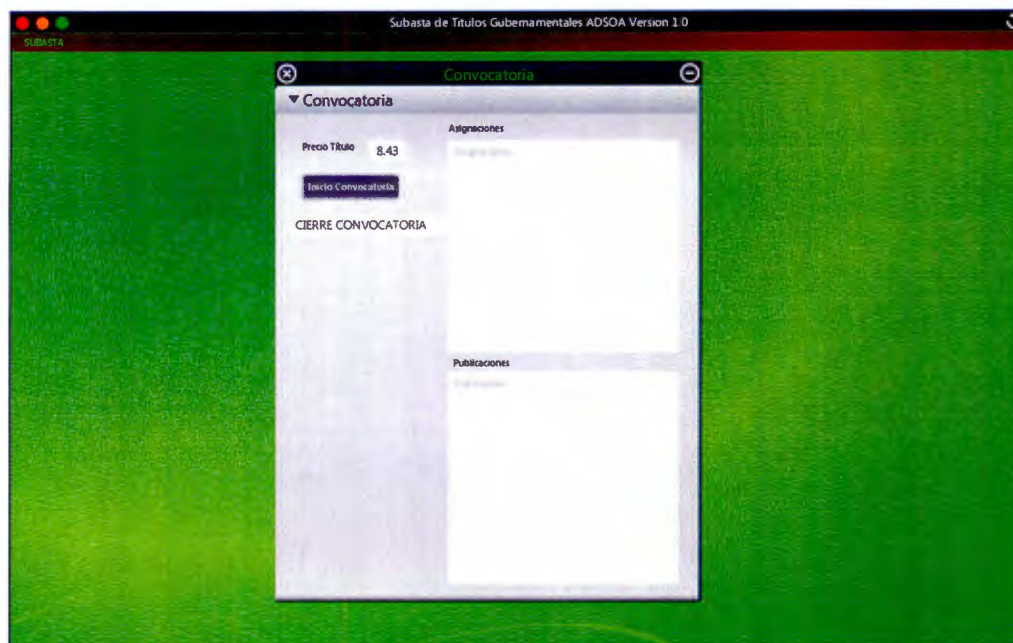
```

**Figura 5.19 Registro del evento erróneo en la bitácora del servidor posturas**

Como ya se mencionó, la Entidad Subasta se configuró para que sólo acepte la asignación y publicación al recibir dos solicitudes del Servidor Posturas. Para esta prueba se realizó la asignación sin instancias, dando como resultado que en la Entidad Subasta no se despliegue la información concerniente a las asignaciones. La figura 5.20 y 5.21 muestran este caso.



*Figura 5.20 Entidad subasta validador, tolerancia a fallos*



*Figura 5.21 Entidad subasta, tolerancia a fallos*

Para continuar con este caso, se inició una nueva instancia del Servidor Posturas, tal y como lo muestra la figura 5.22

```

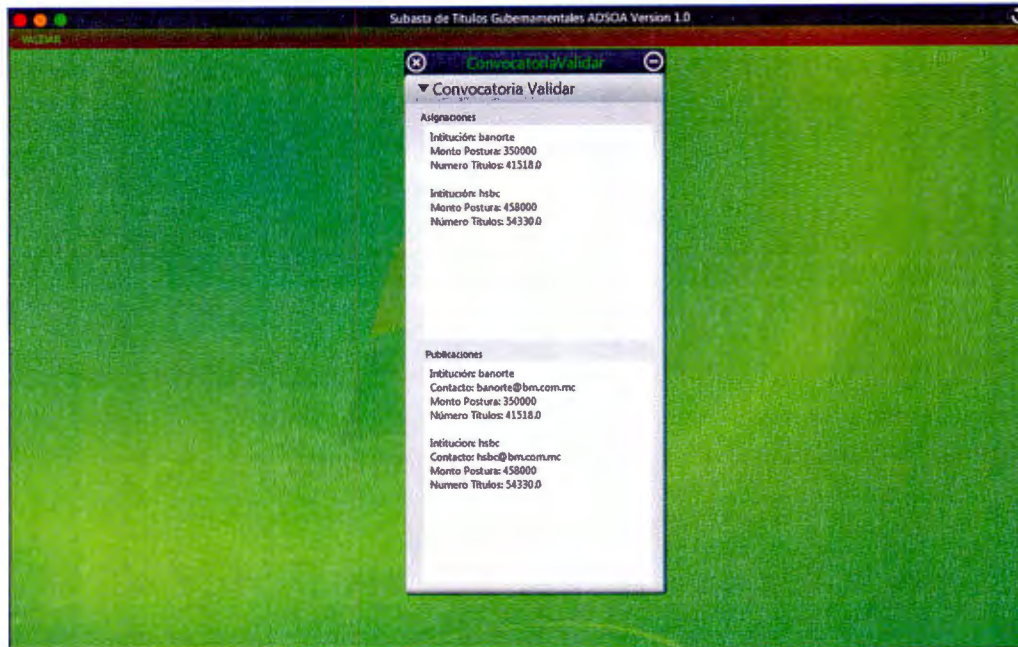
C:\Windows\system32\cmd.exe
2014-10-29 17:22:00,676 [INFO ] [main] - Creando ADF
2014-10-29 17:22:00,677 [INFO ] [main] - Cambiando a estado <INICIALIZADO> al AD
F
2014-10-29 17:22:00,685 [INFO ] [main] - Creando Despachador
2014-10-29 17:22:00,687 [INFO ] [main] - Cambiando a estado <INICIALIZADO> al De
spachador
2014-10-29 17:22:00,688 [INFO ] [main] - Registrando el EAP como un recurso, med
iante la llave <EAP>
2014-10-29 17:22:00,689 [INFO ] [main] - Registrando la Persistencia local del E
AP como un recurso, mediante la llave <PERSISTENCIA_EAP>
2014-10-29 17:22:00,690 [INFO ] [main] - Cambiando a estado <INICIALIZADO> al EA
P
2014-10-29 17:22:00,799 [INFO ] [main] - Creando la sesion con el Campo de Datos
2014-10-29 17:22:03,630 [INFO ] [main] - Creando Enviador
2014-10-29 17:22:03,633 [INFO ] [main] - Cambiando a estado <INICIALIZADO> al AC
P
2014-10-29 17:22:03,635 [INFO ] [main] - Cambiando a estado <OPERANDO> al ACP
2014-10-29 17:22:03,638 [INFO ] [main] - Cambiando a estado <OPERANDO> al ACP
2014-10-29 17:22:03,639 [INFO ] [main] - Cambiando a estado <OPERANDO> al ADF
2014-10-29 17:22:03,640 [INFO ] [main] - Cambiando a estado <OPERANDO> al Despac
hador
2014-10-29 17:22:03,644 [INFO ] [main] - Cambiando a estado <OPERANDO> al EAP
2014-10-29 17:22:03,645 [INFO ] [main] - ***Servidor Posturas Operando

```

**Figura 5.22** Instancia del servidor posturas

Al realizar la etapa de publicación, la Instancia del Servidor Posturas recibe esa solicitud, internamente se percata que tiene un mensaje pendiente y que no lo tiene registrado, en ese momento envía un mensaje a su Entidad (Servidor Posturas) solicitándole la información faltante, de este modo envía la información correspondiente a la publicación y de la asignación pendiente, dando como resultado la consistencia de la información en todo el sistema.

Las figuras 5.23 y 5.24 muestran este resultado.

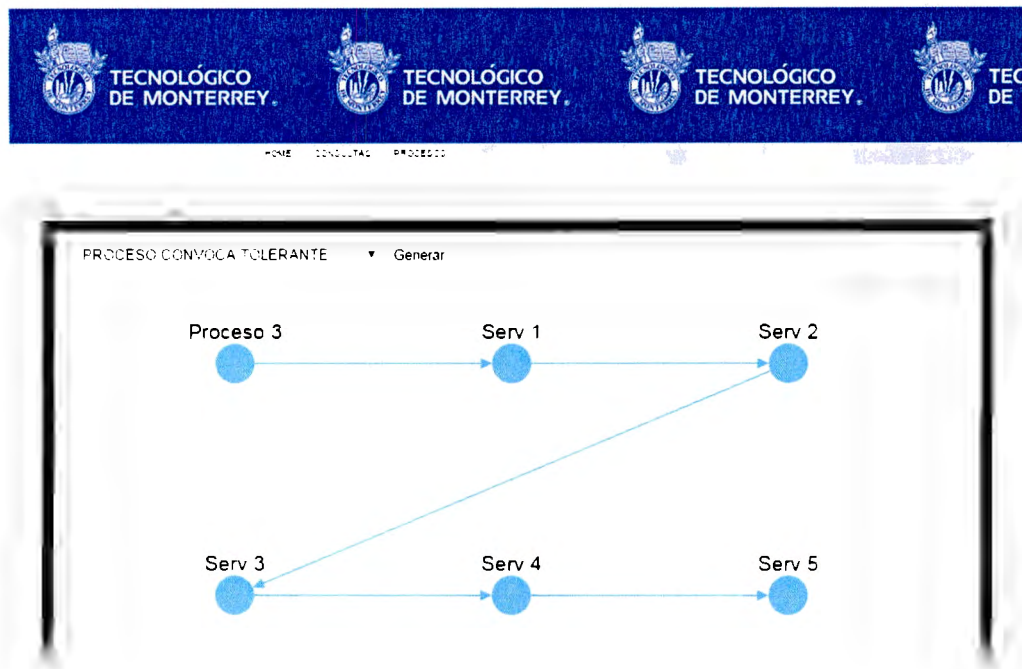


*Figura 5.23 Entidad Subasta Validador en la etapa de publicación.*



*Figura 5.24 Entidad Subasta en la etapa de publicación.*

La figura 5.25 indica que le proceso actual término con éxito.



***Figura 5.25 Grafo del sistema tolerante a fallas en la etapa de publicación.***

Para respaldar lo anterior, se realizó la consulta a la base de datos del proceso actual y se obtuvo el siguiente resultado.

PROCESO CONVOCA TOLERANTE    Selecionar Servicio    2014-10-29

2014-10-26    Buscar

Proceso	Descripción Proceso	Servicio	Descripción Servicio	Estado	Fecha Operación	Fecha Ejecución Servicio
PROCESO CONVOCA TOLERANTE	CONVOCA TOL	CONVOCA TOL	CONVOCA TOL	FINAL	2014-10-29	2014-10-29 00:59:47 PM
PROCESO CONVOCA TOLERANTE	CONVOCA TOL	CONVOCA TOL	CONVOCA TOL	FINAL	2014-10-29	2014-10-29 00:59:47 PM
3	PROCESO CONVOCA TOLERANTE	2	POSTURAS	ERROR	2014-10-29	Oct 29, 2014 5:00:52 PM
3	PROCESO CONVOCA TOLERANTE	2	POSTURAS	FINAL	2014-10-29	Oct 29, 2014 5:01:58 PM
	PROCESO CONVOCA TOLERANTE			FINAL		
	PROCESO CONVOCA TOLERANTE	40	SNAP	FINAL	2014-10-29	
	PROCESO CONVOCA TOLERANTE	40	SNAP	FINAL	2014-10-29	
3	PROCESO CONVOCA TOLERANTE	5	PUBLICAR	FINAL	2014-10-29	Oct 29, 2014 5:23:55 PM

1 a página    Registros por página:    Mostrando registros 1 a 18 de 50

**Figura 5.26 Consulta de la base de datos del sistema tolerante a fallas.**

En la consulta se observa la descripción de la ejecución del proceso de negocio, así como la hora de ejecución de los servicios y el estado en el que se encuentran.

En cuanto al gestor, éste registra en su bitácora los cambios de estado de los servicios gestionables, incluyendo el evento de error en la postura, realizando el monitoreo de manera oportuna. En la figura 5.27 se muestra la bitácora.

```
2014-10-29 16:57:05,807 [INFO] [main] - ***Servidor Gestor Operando
2014-10-29 16:59:18,169 [DEBUG] [pool-3-thread-2] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(1)evento(9f111e7d48c1620a146.
2014-10-29 16:59:18,183 [DEBUG] [pool-3-thread-2] - El evento (9f111e7d48c1620a146290894f9a4d2152f0721), proveniente de la huella (362539672) es :
2014-10-29 16:59:18,198 [DEBUG] [pool-3-thread-2] - Se registra el origen (Origenes(llave=13-2-1-InegocioSolicitante=13subSistemaSolicitante=2ent.
2014-10-29 16:59:18,194 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(1)evento(9f111e7d48c
2014-10-29 16:59:18,201 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:18,201 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:18,201 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:18,225 [INFO] [pool-3-thread-2] - E 9f111e7d48c1620a146290894f9a4d2152f0721 1 1 86
2014-10-29 16:59:18,241 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(1)evento(a32cd4710d0e026d4.
2014-10-29 16:59:18,243 [DEBUG] [pool-3-thread-3] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(2)evento(170c26353c27bdfbb657.
2014-10-29 16:59:18,246 [DEBUG] [pool-3-thread-3] - El evento (170c26353c27bdfbb657c9003cf564ac23235e8c , proveniente de la huella (362639672) es :
2014-10-29 16:59:18,261 [DEBUG] [pool-3-thread-3] - Se registra el origen (Origenes(llave=13-2-1-InegocioSolicitante=13subSistemaSolicitante=2ent.
2014-10-29 16:59:18,266 [INFO] [pool-3-thread-3] - E 170c26353c27bdfbb657c9003cf564ac23235e8c 2 1 13
2014-10-29 16:59:18,267 [INFO] [DepachadorDeSolicitudes] - AT 9f111e7d48c1620a146290894f9a4d2152f0721 1 1 54 64
2014-10-29 16:59:18,269 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(2)evento(170c26353c2
2014-10-29 16:59:18,269 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:18,269 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:18,269 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:18,269 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:18,296 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(d71cb0ce2703b88ca.
2014-10-29 16:59:18,298 [INFO] [DepachadorDeSolicitudes] - AT 170c26353c27bdfbb657c9003cf564ac23235e8c 2 1 16 39
2014-10-29 16:59:38,482 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:38,482 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:38,482 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:38,482 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:38,486 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(fbf7b10026c7be988
2014-10-29 16:59:38,499 [INFO] [DepachadorDeSolicitudes] - AT 160b3664021abedd3fdd601988e23bd0262b6947 1 1 16 18
2014-10-29 16:59:38,606 [DEBUG] [pool-3-thread-3] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(2)evento(e60cbfd5d8964a3bc495.
2014-10-29 16:59:38,608 [DEBUG] [pool-3-thread-3] - El evento (e60cbfd5d8964a3bc4952c2acab4de7f0ad5fd8), proveniente de la huella (1671433370) es :
2014-10-29 16:59:38,608 [DEBUG] [pool-3-thread-3] - Se registra el origen (Origenes(llave=10-1-3-InegocioSolicitante=10subSistemaSolicitante=1enti.
2014-10-29 16:59:38,614 [INFO] [pool-3-thread-3] - E e60cbfd5d8964a3bc4952c2acab4de7f0ad5fd8 2 1 8
2014-10-29 16:59:38,616 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(2)evento(e60cbfd5d89
2014-10-29 16:59:38,616 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:38,616 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:38,616 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:38,616 [INFO] [DepachadorDeSolicitudes]
2014-10-29 16:59:38,619 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(7fb3396a2c1c3d49.
2014-10-29 16:59:38,621 [INFO] [DepachadorDeSolicitudes] - AT e60cbfd5d8964a3bc4952c2acab4de7f0ad5fd8 2 1 10 5
2014-10-29 17:00:52,581 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:00:52,581 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:00:52,581 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:00:52,581 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:00:52,581 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:00:52,590 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(1c5e01278eb6da99a9
2014-10-29 17:00:52,596 [INFO] [DepachadorDeSolicitudes] - AT 2d0243b38ace597c339be444a19502fe415faa2 3 1 20 15
2014-10-29 17:00:52,605 [DEBUG] [Enviodor(ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(5)evento(1c5e01278eb6da99a9
2014-10-29 17:00:52,675 [INFO] [pool-3-thread-5] - E 1c5e01278eb6da99a939da13add015c609c86556 5 1 17 73
2014-10-29 17:00:52,676 [DEBUG] [pool-3-thread-5] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(4)evento(72e481140891802703de
2014-10-29 17:00:52,677 [DEBUG] [pool-3-thread-5] - El evento (72e481140891802703de53f8bc425b416c018649), proveniente de la huella (1671433370) es :
2014-10-29 17:00:52,691 [DEBUG] [pool-3-thread-5] - Se registra el origen (Origenes(llave=10-1-3-InegocioSolicitante=10subSistemaSolicitante=1enti.
2014-10-29 17:00:52,707 [INFO] [pool-3-thread-5] - E 72e481140891802703de53f8bc425b416c018649 4 1 31
2014-10-29 17:00:52,708 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(4)evento(72e48114089
2014-10-29 17:00:52,708 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:00:52,708 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:00:52,708 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:00:52,708 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:00:52,716 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(0bcd0e30b9d9f8704
2014-10-29 17:00:52,718 [INFO] [DepachadorDeSolicitudes] - AT 72e481140891802703de53f8bc425b416c018649 4 1 32 10
2014-10-29 17:16:56,094 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:16:56,094 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:16:56,094 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:16:56,093 [INFO] [pool-3-thread-6] - E cb4fad8735663b624ce798d3bed88759b8e308b 6 1 10
2014-10-29 17:16:56,111 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(61c6b0f707313e6c3
2014-10-29 17:16:56,116 [DEBUG] [pool-3-thread-6] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(6)evento(cb4fad8735663b624ce7
2014-10-29 17:16:56,116 [DEBUG] [pool-3-thread-6] - El evento (cb4fad8735663b624ce798d3bed88759b8e308b), proveniente de la huella (1671433370) es :
2014-10-29 17:16:56,117 [DEBUG] [pool-3-thread-6] - Se registra el origen (Origenes(llave=10-1-3-InegocioSolicitante=10subSistemaSolicitante=1enti.
2014-10-29 17:16:56,121 [INFO] [pool-3-thread-6] - E cb4fad8735663b624ce798d3bed88759b8e308b 6 1 7
2014-10-29 17:16:56,126 [INFO] [DepachadorDeSolicitudes] - AT cd637c6e4163598140693a178d92467fab184e7 6 1 10 32
2014-10-29 17:16:56,126 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(6)evento(cb4fad87356
2014-10-29 17:16:56,127 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:16:56,127 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:16:56,127 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:16:56,127 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:16:56,127 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:16:56,129 [DEBUG] [Enviodor(ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(7)evento(61c6b0f707313e6c35
2014-10-29 17:16:56,135 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(982499246261a2fdd
2014-10-29 17:16:56,138 [INFO] [DepachadorDeSolicitudes] - AT cb4fad8735663b624ce798d3bed88759b8e308b 6 1 12 12
2014-10-29 17:18:15,704 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:18:15,704 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:18:15,704 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:18:15,704 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:18:15,704 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:18:15,709 [INFO] [pool-3-thread-8] - E 6dfe729121f85ca4697ccc5b1d56d152c87d4d 1 1 29
2014-10-29 17:18:15,716 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(1abce37fec2f16d1
2014-10-29 17:18:15,718 [INFO] [DepachadorDeSolicitudes] - AT 6dfe729121f85ca4697ccc5b1d56d152c87d4d 1 1 29 15
2014-10-29 17:18:15,732 [DEBUG] [Enviodor(ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(9)evento(1abce37fec2f16d1b
2014-10-29 17:18:15,771 [DEBUG] [pool-3-thread-8] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(2)evento(f00ef903fd9fad02271c
2014-10-29 17:18:15,773 [DEBUG] [pool-3-thread-8] - El evento (f00ef903fd9fad02271c9420df03b400a11582df), proveniente de la huella (1389367608) es :
2014-10-29 17:18:15,774 [DEBUG] [pool-3-thread-8] - Se registra el origen (Origenes(llave=10-1-2-InegocioSolicitante=10subSistemaSolicitante=1enti.
2014-10-29 17:18:15,780 [INFO] [pool-3-thread-8] - E f00ef903fd9fad02271c9420df03b400a11582df 2 1 9
2014-10-29 17:18:15,781 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(2)evento(f00ef903fd9
2014-10-29 17:18:15,781 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:18:15,781 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:18:15,781 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:18:15,781 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:18:15,781 [INFO] [DepachadorDeSolicitudes]
2014-10-29 17:18:15,787 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(86ff6d0602df1e9f
2014-10-29 17:18:15,788 [INFO] [DepachadorDeSolicitudes] - AT f00ef903fd9fad02271c9420df03b400a11582df 2 1 10 7
```

```

2014-10-29 17:18:36,843 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-29 17:18:36,843 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 3
2014-10-29 17:18:36,843 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 4
2014-10-29 17:18:36,843 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-29 17:18:36,847 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADP el mensaje {numeroServicio(1)secuencial(0)evento(a51266509cddca3e
2014-10-29 17:18:36,849 [INFO] [DepachadorDeSolicitudes] - AT c68e47c7c0f3cc32a9701c1282d10e056c9241b" 7 1 32 -
2014-10-29 17:18:36,862 [DEBUG] [Enviaador (ACP)] - Enviador envio el mensaje transaccional {numeroServicio(1)secuencial(11)evento(a51266509cddca3e
2014-10-29 17:18:36,889 [INFO] [pool-3-thread-8] - S a51266509cddca3e739d0f05df4a406baa29125 11 1 16 28
2014-10-29 17:18:36,890 [DEBUG] [pool-3-thread-8] - Se recibio el mensaje transaccional {numeroServicio(1)secuencial(8)evento(19ccb7edda05168ec2f
2014-10-29 17:18:36,891 [DEBUG] [pool-3-thread-8] - El evento {19ccb7edda05168ec2f67f3706a22dff79280b , proveniente de la huella {1571433370} es
2014-10-29 17:18:36,891 [DEBUG] [pool-3-thread-8] - Se registra el origen {Origenes{llave=10-1-3-InegocioSolicitante=10subSistemaSolicitante=lent
2014-10-29 17:18:36,898 [INFO] [pool-3-thread-8] - E 19ccb7edda05168ec2f67f3706a22dff79280b 8 1 9
2014-10-29 17:18:36,946 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender {numeroServicio(1)secuencial(8)evento(19ccb7edda
2014-10-29 17:18:36,947 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-29 17:18:36,947 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 3
2014-10-29 17:18:36,947 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 4
2014-10-29 17:18:36,947 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 17:18:37,078 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADP el mensaje {numeroServicio(1)secuencial(0)evento(f0da5994e7861b887
2014-10-29 17:18:37,171 [DEBUG] [Enviaador (ACP)] - Enviador envio el mensaje transaccional {numeroServicio(1)secuencial(12)evento(f0da5994e7861b887
2014-10-29 17:18:37,217 [INFO] [DepachadorDeSolicitudes] - AT 19ccb7edda05168ec2f67f3706a22dff79280b 8 1 56 272
2014-10-29 17:23:55,434 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-29 17:23:55,434 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 3
2014-10-29 17:23:55,434 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 5
2014-10-29 17:23:55,434 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-29 17:23:55,441 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADP el mensaje {numeroServicio(1)secuencial(0)evento(32f2d684381620674
2014-10-29 17:23:55,446 [INFO] [DepachadorDeSolicitudes] - AT d6ef9589f3abb5b340fbddda7deb7145cc51664 9 1 28 13
2014-10-29 17:23:55,482 [DEBUG] [Enviaador (ACP)] - Enviador envio el mensaje transaccional {numeroServicio(1)secuencial(13)evento(32f2d684381620674
2014-10-29 17:23:55,629 [DEBUG] [pool-3-thread-11] - Se recibio el mensaje transaccional {numeroServicio(1)secuencial(10)evento(64d9c452958319f00
2014-10-29 17:23:55,532 [DEBUG] [pool-3-thread-11] - El evento {64d9c452958319f00602141e4abal2461e6515d}, proveniente de la huella {1571433370} e
2014-10-29 17:23:55,534 [DEBUG] [pool-3-thread-11] - Se registra el origen {Origenes{llave=10-1-3-InegocioSolicitante=10subSistemaSolicitante=lent
2014-10-29 17:23:55,541 [INFO] [pool-3-thread-11] - E 64d9c452958319f00602141e4abal2461e6515d 10 1 12
2014-10-29 17:23:55,594 [INFO] [pool-3-thread-10] - S 32f2d6843816206741fb95f6c40caef70f3a8e53 13 1 20 135
2014-10-29 17:23:55,605 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender {numeroServicio(1)secuencial(10)evento(64d9c4529
2014-10-29 17:23:55,606 [INFO] [DepachadorDeSolicitudes] - ***Gestor atiende cambio de estado
2014-10-29 17:23:55,606 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 3
2014-10-29 17:23:55,606 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 5
2014-10-29 17:23:55,606 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-29 17:23:55,642 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADP el mensaje {numeroServicio(1)secuencial(0)evento(df68f55335bb1f32b
2014-10-29 17:23:55,719 [DEBUG] [Enviaador (ACP)] - Enviador envio el mensaje transaccional {numeroServicio(1)secuencial(14)evento(df68f55335bb1f32b
2014-10-29 17:23:55,777 [INFO] [DepachadorDeSolicitudes] - AT 64d9c452958319f00602141e4abal2461e6515d 10 1 76 172

```

Figura 5.27 Bitácora del gestor para el sistema tolerante a fallos.

### 5.6.3.2. Resultados del Caso de Prueba III

Se demostró que el sistema gestor integrado en una arquitectura distribuida y tolerante a fallas funciona como se esperaba, llevando el monitoreo del proceso en ejecución, registrando los cambios de estados de los servicios gestionables, ya sea por su ejecución o error, visualizando cada etapa de manera clara.

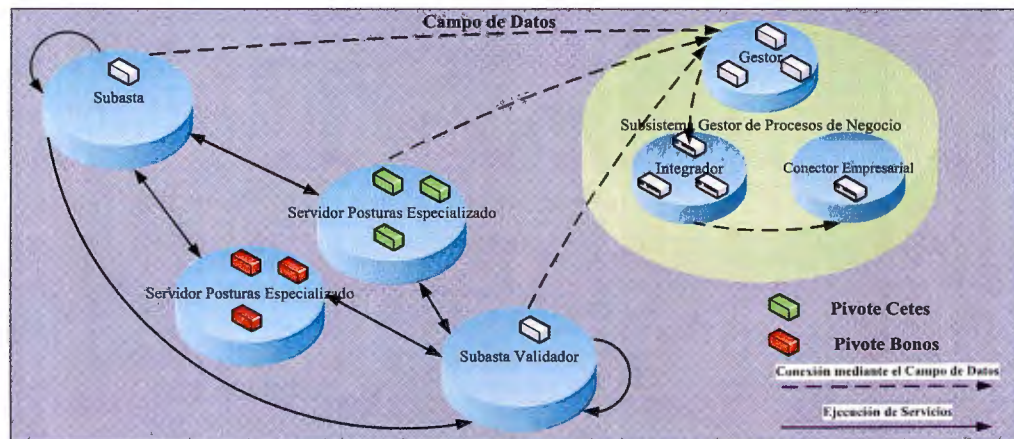


## 5.6.4. Caso de Prueba IV

### 5.6.4.1. Sistema Especializado

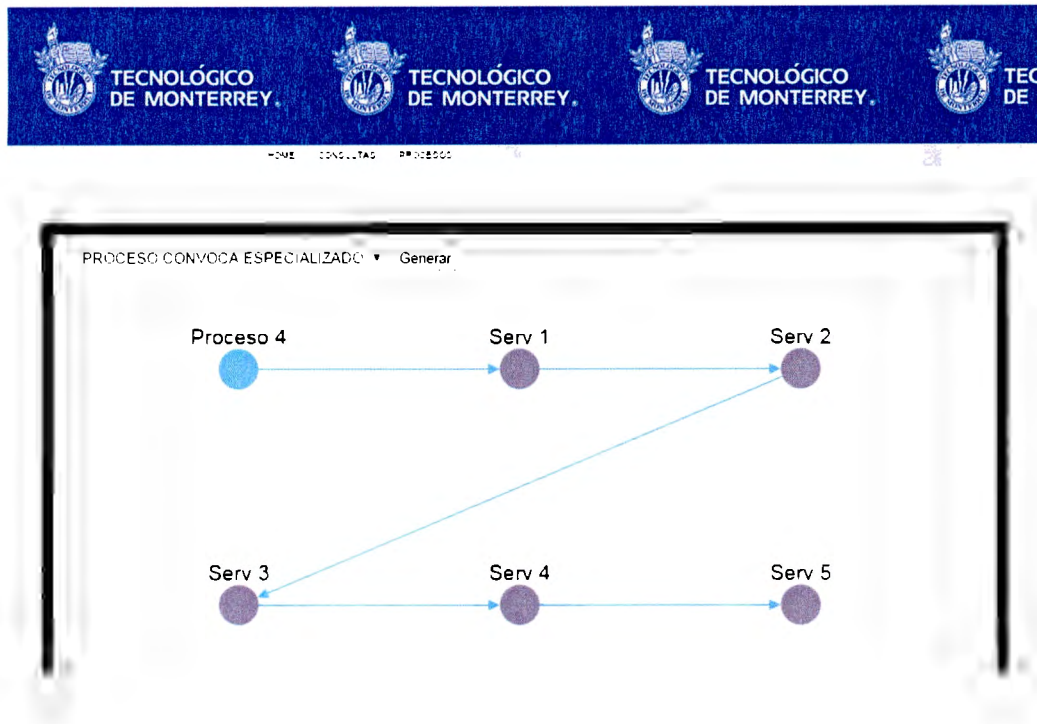
Para esta prueba se utiliza una arquitectura distribuida, con la característica de tolerancia a fallas, pero además es un sistema especializado. Como se indicó en la prueba anterior, está conformado por tres Entidades, La figura 5.28 muestra la arquitectura utilizada.

Para cumplir con la característica de especialización, el Servidor Posturas se configura de tal manera que solo atenderá las posturas para las que está configurado ya sean posturas para CETES o BONOS y esto se hace mediante su pivote, logrando así la distribución de la carga.



*Figura 5.28 Arquitectura del sistema especializado.*

Para esta prueba se valida la implementación del sistema gestor en una arquitectura distribuida, tolerante a fallos y especializada, realizando el monitoreo del proceso de negocio. La imagen 5.29 muestra cómo se visualiza el inicio del proceso.



***Figura 5.29 Grafo inicial del sistema distribuido.***

Para esta prueba, se hace uso de la tolerancia a fallas como el caso de la prueba anterior, incluyendo la especialización de las Instancias. Como se puede ver en la figura 5.30, se tienen dos Instancias especializadas para atender posturas de CETES y otras dos para BONOS.

The figure displays four screenshots of Windows command prompts, arranged in a 2x2 grid. Each window shows the output of a program named 'Servidor Posturas Operando'. The logs are as follows:

- Top-left window:** Shows the initial setup, including creating the ERP, Despachador, and Envaiador, and registering the ERP as a resource. The main process starts in 'INICIALIZADO' state and then transitions to 'OPERANDO'.
- Top-right window:** Shows the continuation of the setup, including creating the session with the ERP, and the main process transitioning to 'OPERANDO'.
- Bottom-left window:** Shows the creation of the Envaiador and the main process transitioning to 'OPERANDO'.
- Bottom-right window:** Shows the final state where the main process is 'OPERANDO' and the server is ready to receive postures.

**Figura 5.30** Instancias especializadas

A continuación se muestra que la convocatoria está activa y se ingresa una postura para CETES, en la figura 5.31 y 5.32 se confirma que solamente las Instancias especializadas para la recepción de este tipo de posturas registran su ingreso al sistema.

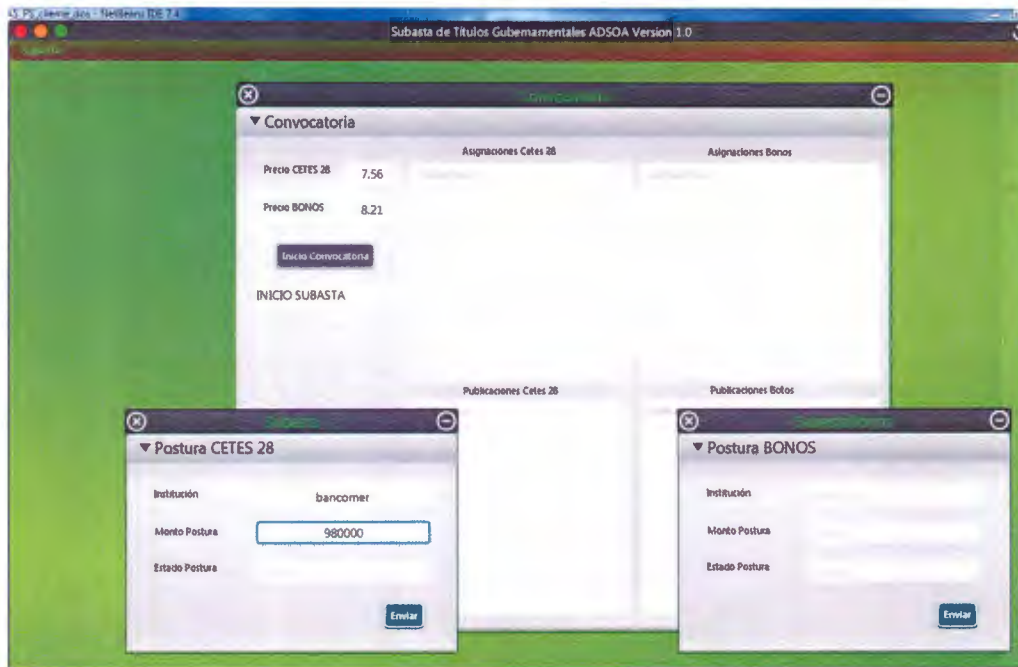


Figura 5.31 Entidad Subasta y el ingreso de una postura para CETES

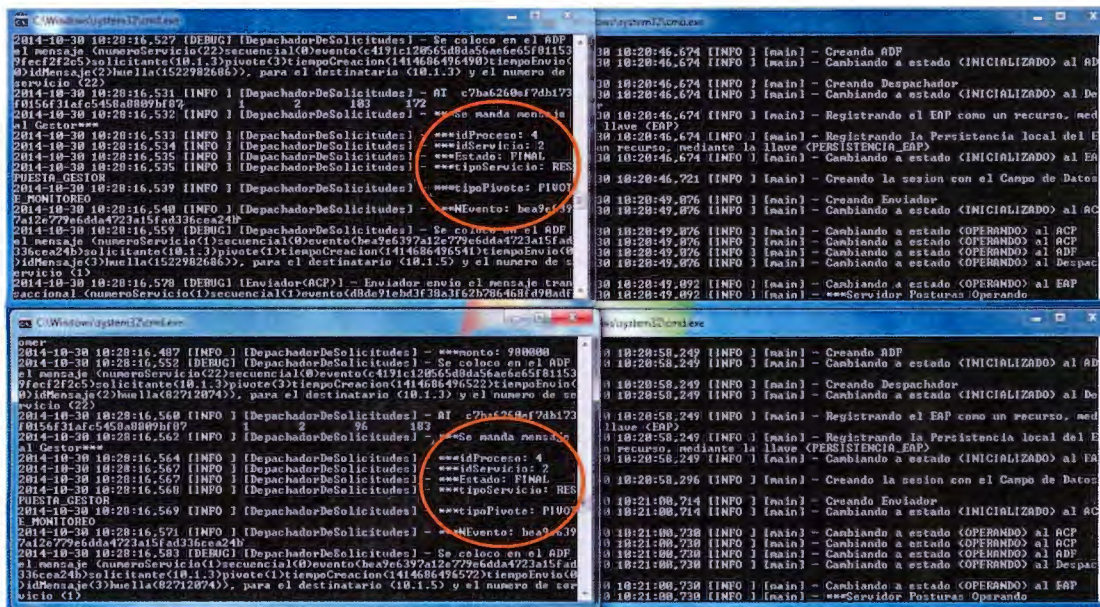


Figura 5.32 Instancias especializadas para CETES.

El siguiente paso fue ingresar una postura para BONOS y como se esperaba las Instancias especializadas en la recepción de este tipo de posturas, reciben y registran el evento, ver figuras 5.33 y 5.34.

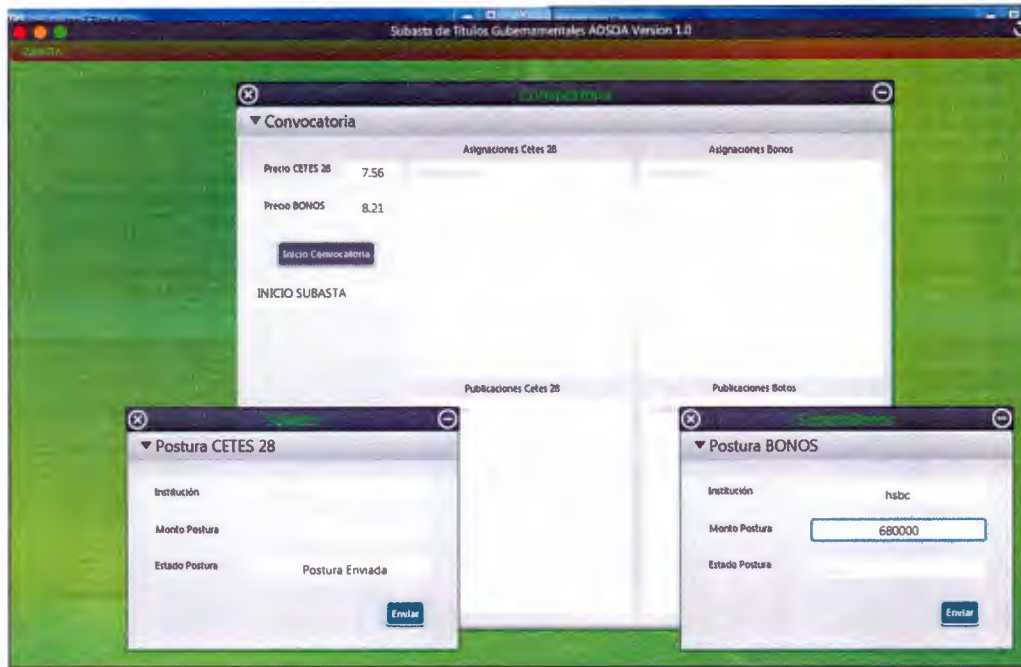


Figura 5.33 Entidad Subasta y el ingreso de una postura para BONOS.

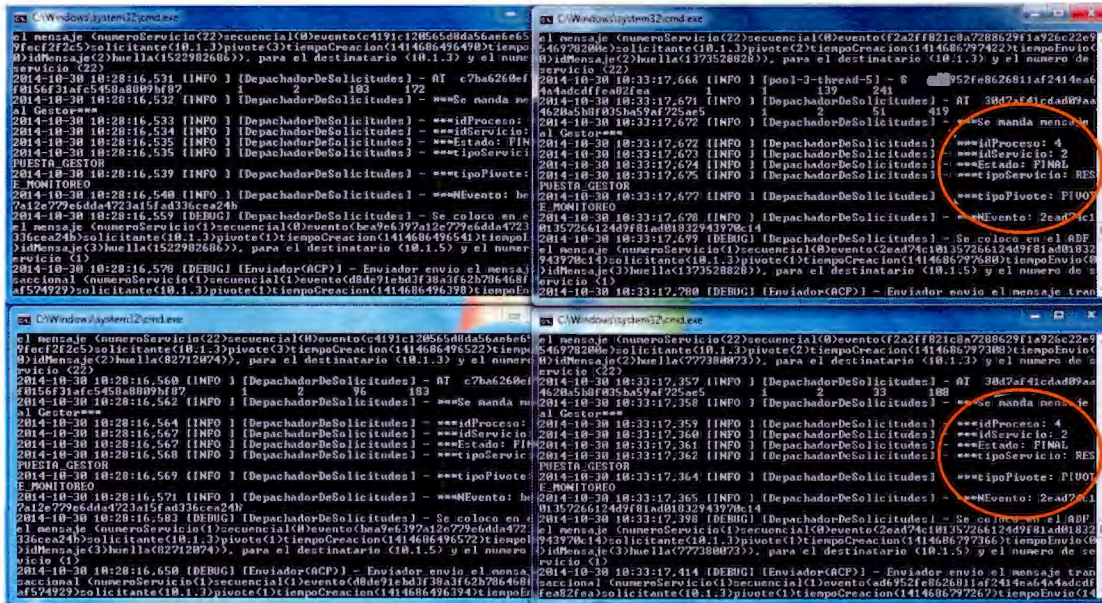
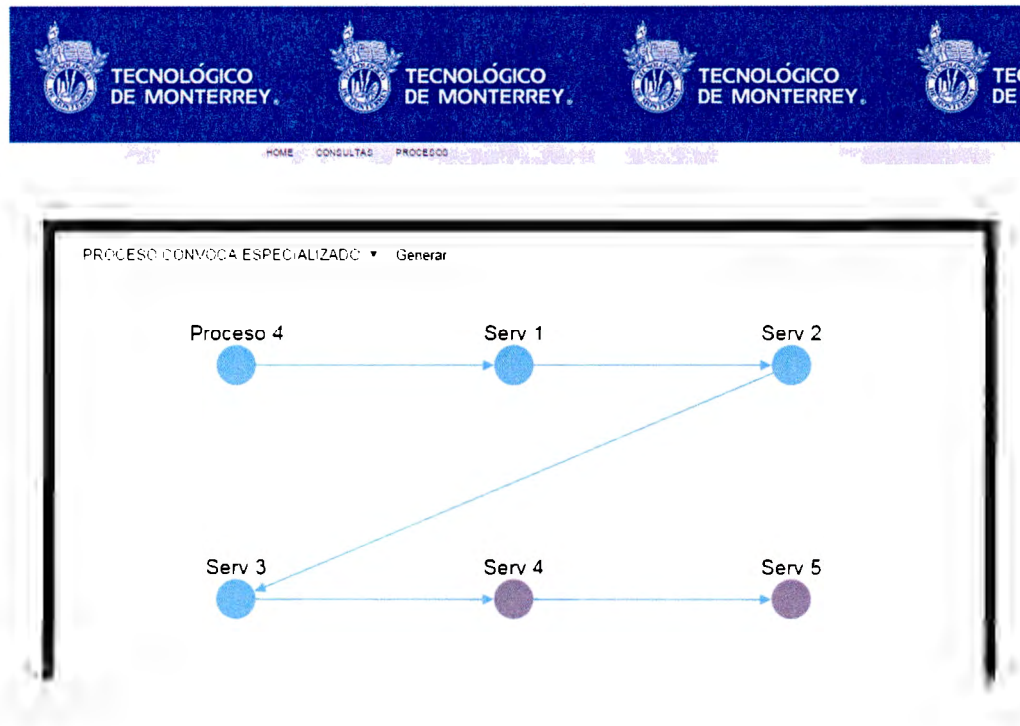


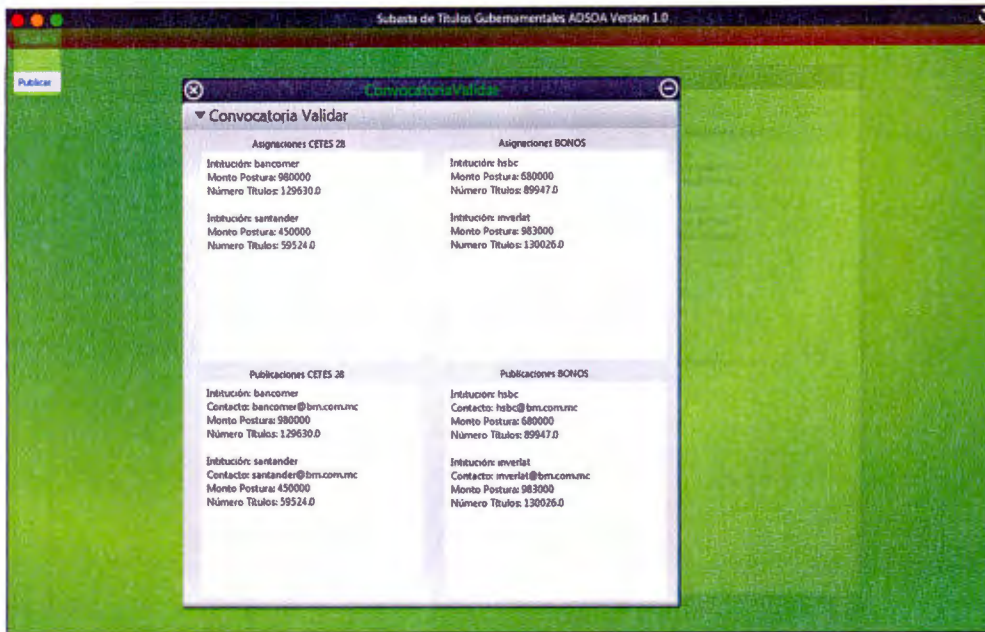
Figura 5.34 Entidades especializadas para BONOS.

Después del ingreso de las posturas, se realiza el cierre de la subasta, en la figura 5.35 se observa esta etapa.

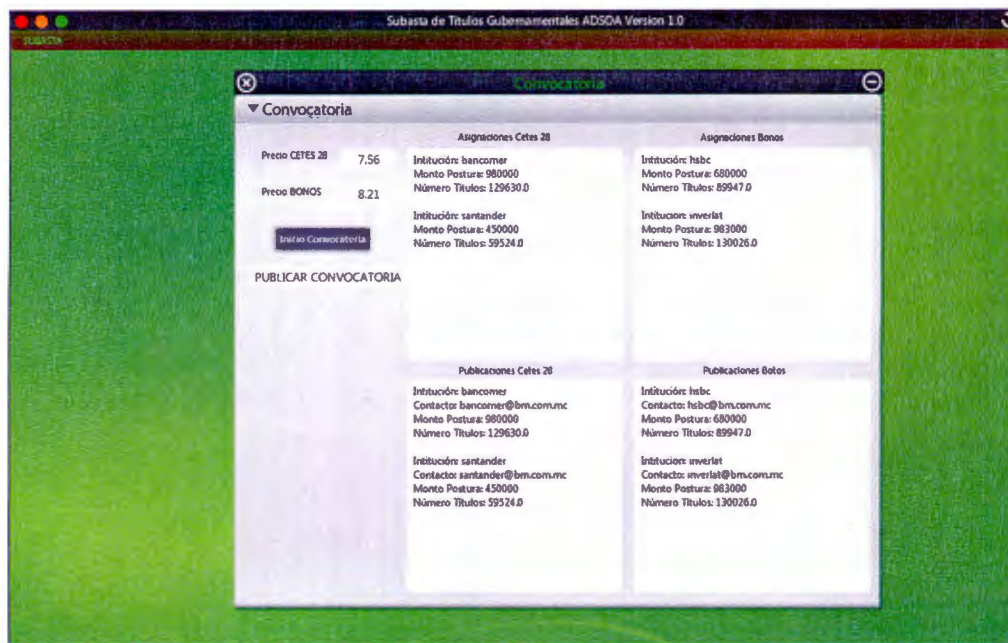


*Figura 5.35 Grafo del sistema especializado en la etapa de cierre.*

Posteriormente se realiza la asignación y la publicación de la subasta como se observa en las figuras 5.36 y 5.37.

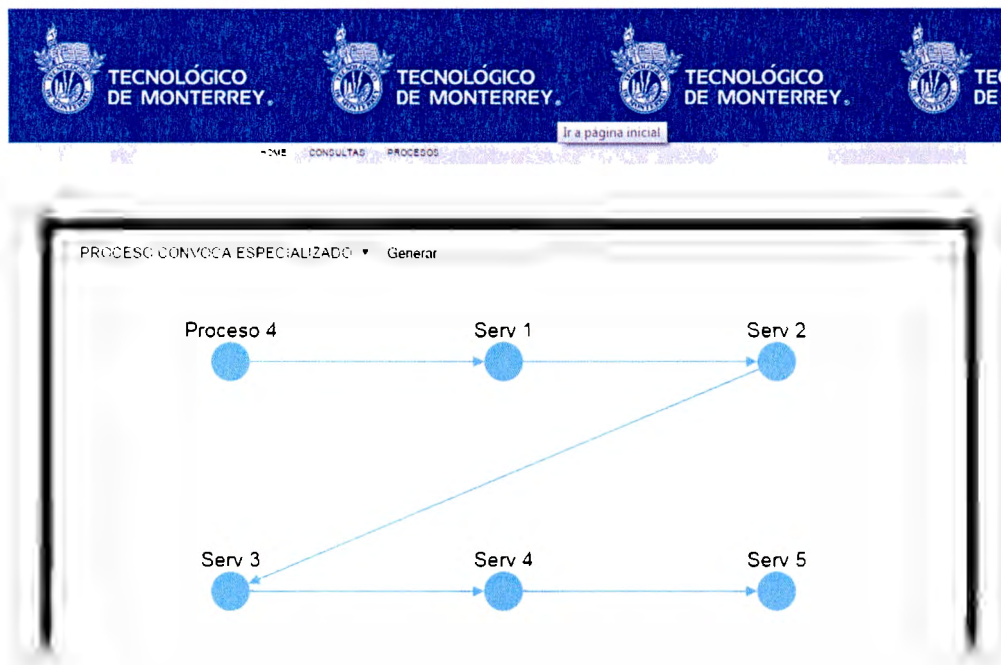


**Figura 5.36** Entidad subasta validador del sistema especializado en la etapa de publicación.



**Figura 5.37** Entidad Subasta del sistema especializado en la etapa de publicación.

La figura 5.38 indica que el proceso actual término con éxito.



**Figura 5.38 Grafo del sistema especializado en la etapa de publicación.**

Para respaldar lo anterior, se realizó la consulta a la base de datos del proceso actual y se obtuvo el siguiente resultado.



ID	Descripción Proceso	Servicio	Descripción Servicio	Estado	Fecha Operacion	Fecha Ejecucion Servicio
	PROCESO CONVOCA ESPECIALIZADO		CONVOCA	PRUAL		Oct 30, 2014 10:25:17 AM
	PROCESO CONVOCA ESPECIALIZADO		POSTULAR	PRUAL		Oct 30, 2014 10:25:17 AM
	PROCESO CONVOCA ESPECIALIZADO		POSTULAR			Oct 30, 2014 10:35:17 AM
	PROCESO CONVOCA ESPECIALIZADO		POSTULAR			Oct 30, 2014 10:38:52 AM
	PROCESO CONVOCA ESPECIALIZADO		POSTULAR		2014-10-30	Oct 30, 2014 10:39:22 AM
	PROCESO CONVOCA ESPECIALIZADO			PRUAL		Oct 30, 2014 10:37:12 AM
4	PROCESO CONVOCA ESPECIALIZADO	4	CONVOCAR	PRUEBO	2014-10-30	Oct 30, 2014 10:39:59 AM
	PROCESO CONVOCA ESPECIALIZADO		CONVOCAR	PRUAL	2014-10-30	Oct 30, 2014 10:30:14 AM
	PROCESO CONVOCA ESPECIALIZADO		PUBLICAR	PRUEBO	2014-10-30	Oct 30, 2014 10:42:11 AM

*Figura 5.39 Consulta de la base de datos del sistema especializado.*

Como se puede observar, esta consulta muestra la descripción de la ejecución del proceso de negocio, se detalla la hora de ejecución de los servicios, así como el estado en el que se encuentran.

En cuanto al gestor, durante la subasta registra en su bitácora los cambios de estado de los servicios gestionables, realizando el monitoreo de manera clara. En la figura 5.40 se muestra esta bitácora.

```
2014-10-30 10:19:35,276 [INFO] [main] - ***Servidor Gestor Operando
2014-10-30 10:26:26,311 [DEBUG] [pool-3-thread-2] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(1)evento(0587a69d796d00ab64ed5b7ef1
2014-10-30 10:26:26,396 [DEBUG] [pool-3-thread-2] - El evento (0587a69d796d00ab64ed5b7ef183e64d121230b2), proveniente de la huella (360501950) es nuevo
2014-10-30 10:26:26,407 [DEBUG] [pool-3-thread-2] - Se registra el origen (Origenes(llave=13-2-1-InegocioSolicitante=13subSistemaSolicitante=2entidadSo
2014-10-30 10:26:26,416 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(1)evento(0587a69d796d00ab
2014-10-30 10:26:26,418 [INFO] [pool-3-thread-2] - E 0587a69d796d00ab64ed5b7ef183e64d121230b2 1 1 135
2014-10-30 10:26:26,430 [INFO] [DepachadorDeSolicitudes] - ***Destino sistema cambio de estado
2014-10-30 10:26:26,433 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:26:26,436 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 1
2014-10-30 10:26:26,436 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-30 10:26:26,446 [DEBUG] [pool-3-thread-2] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(2)evento(d1b1ba87709df66b2a4644e1
2014-10-30 10:26:26,449 [DEBUG] [pool-3-thread-2] - El evento (d1b1ba87709df66b2a4644e1968f6a6b87afc), proveniente de la huella (360501950) es nuevo
2014-10-30 10:26:26,454 [DEBUG] [pool-3-thread-2] - Se registra el origen (Origenes(llave=13-2-1-InegocioSolicitante=13subSistemaSolicitante=2entidadSo
2014-10-30 10:26:26,459 [INFO] [pool-3-thread-2] - E d1b1ba87709df66b2a4644e1968f6a6b87afc 2 1 14
2014-10-30 10:26:26,486 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(calb43a5a05039d993a514
2014-10-30 10:26:26,516 [INFO] [DepachadorDeSolicitudes] - AT 0587a69d796d00ab64ed5b7ef183e64d121230b2 1 1 132 101
2014-10-30 10:26:26,519 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(2)evento(d1b1ba87709df66
2014-10-30 10:26:26,522 [INFO] [DepachadorDeSolicitudes] - ***Destino sistema cambio de estado
2014-10-30 10:26:26,522 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:26:26,523 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 1
2014-10-30 10:26:26,524 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-30 10:26:26,586 [DEBUG] [Enviaador (ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(1)evento(calb43a5a05039d993a514)
2014-10-30 10:26:26,637 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(b0a347a6fbd54761f7481)
2014-10-30 10:26:26,682 [INFO] [DepachadorDeSolicitudes] - AT d1b1ba87709df66b2a4644e1968f6a6b87afc 2 1 74 163
2014-10-30 10:26:16,717 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(1)evento(d8de91abdf39a3c
2014-10-30 10:26:16,719 [INFO] [DepachadorDeSolicitudes] - ***Destino sistema cambio de estado
2014-10-30 10:26:16,720 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:26:16,721 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-30 10:26:16,721 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-30 10:26:16,742 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(b1b781a62bb5dfe7a96d
2014-10-30 10:26:16,791 [INFO] [DepachadorDeSolicitudes] - AT d8de91abdf39a3c262b7864ed90ad2af574929 1 1 79 74
2014-10-30 10:26:16,846 [DEBUG] [Enviaador (ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(3)evento(b1b781a62bb5dfe7a96d
2014-10-30 10:26:16,854 [DEBUG] [pool-3-thread-4] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(2)evento(bea9e6397a12e779e6dda4723
2014-10-30 10:26:16,858 [DEBUG] [pool-3-thread-5] - El evento (bea9e6397a12e779e6dda4723a15fad336ca24b), proveniente de la huella (1622982686) es nuev
2014-10-30 10:26:16,860 [DEBUG] [pool-3-thread-5] - Se registra el origen (Origenes(llave=10-1-3-InegocioSolicitante=10subSistemaSolicitante=1entidadSo
2014-10-30 10:26:16,919 [INFO] [pool-3-thread-6] - E bea9e6397a12e779e6dda4723a15fad336ca24b 2 1 65
2014-10-30 10:26:16,934 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(2)evento(bea9e6397a12e779e
2014-10-30 10:26:16,939 [INFO] [DepachadorDeSolicitudes] - ***Destino sistema cambio de estado
2014-10-30 10:26:16,961 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:26:16,962 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-30 10:26:16,963 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-30 10:26:16,977 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(e1e0abe12b54beb0c44bc6f
2014-10-30 10:26:16,978 [INFO] [pool-3-thread-4] - S b1b781a62bb5dfe7a96d70829b647f006d825 3 1 59 178
2014-10-30 10:26:16,983 [INFO] [DepachadorDeSolicitudes] - AT bea9e6397a12e779e6dda4723a15fad336ca24b 2 1 80 55
2014-10-30 10:33:17,619 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(1)evento(ad6952fe626811af
2014-10-30 10:33:17,622 [INFO] [DepachadorDeSolicitudes] - ***Destino sistema cambio de estado
2014-10-30 10:33:17,623 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:33:17,624 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-30 10:33:17,625 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-30 10:33:17,630 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(423d4be30051082f638f6e4
2014-10-30 10:33:17,636 [INFO] [DepachadorDeSolicitudes] - AT ad6952fe626811af2414ea64adcdffea82fea 1 1 46 16
2014-10-30 10:33:17,619 [INFO] [pool-3-thread-6] - E ad6952fe626811af2414ea64adcdffea82fea 1 1 46
2014-10-30 10:33:17,666 [DEBUG] [Enviaador (ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(6)evento(42964bf30051082f638f6e4
2014-10-30 10:33:17,673 [DEBUG] [pool-3-thread-6] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(1)evento(ad6952fe626811af2414ea64
2014-10-30 10:33:17,676 [DEBUG] [pool-3-thread-6] - El evento (ad6952fe626811af2414ea64adcdffea82fea), proveniente de la huella (1373528828) ya se h
2014-10-30 10:33:17,678 [DEBUG] [pool-3-thread-6] - Se registra el origen (Origenes(llave=10-1-3-InegocioSolicitante=10subSistemaSolicitante=1entidadSo
2014-10-30 10:33:17,653 [INFO] [pool-3-thread-6] - E ad6952fe626811af2414ea64adcdffea82fea 1 1 80
2014-10-30 10:33:17,654 [DEBUG] [pool-3-thread-6] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(2)evento(2ead74c101357266124d9f81ac
2014-10-30 10:33:17,673 [DEBUG] [pool-3-thread-6] - El evento (2ead74c101357266124d9f81ad01832943970c14), proveniente de la huella (777380073) es nuevo
2014-10-30 10:33:17,688 [DEBUG] [pool-3-thread-6] - Se registra el origen (Origenes(llave=10-1-3-InegocioSolicitante=10subSistemaSolicitante=1entidadSo
2014-10-30 10:33:17,753 [INFO] [pool-3-thread-6] - E 2ead74c101357266124d9f81ad01832943970c14 2 1 99
2014-10-30 10:33:17,753 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(2)evento(2ead74c1013572661
2014-10-30 10:33:17,764 [INFO] [DepachadorDeSolicitudes] - ***Destino sistema cambio de estado
2014-10-30 10:33:17,764 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:33:17,764 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-30 10:33:17,765 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-30 10:33:17,768 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(aac02cb4aa37198fb76393
2014-10-30 10:33:17,806 [INFO] [pool-3-thread-6] - S 42964bf30051082f638f6e4e0054811562d748396 5 1 17 262
2014-10-30 10:33:17,807 [INFO] [DepachadorDeSolicitudes] - AT 2ead74c101357266124d9f81ad01832943970c14 2 1 99 54
2014-10-30 10:36:58,306 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(3)evento(aee9ac0f2ae37c9
2014-10-30 10:36:58,309 [INFO] [DepachadorDeSolicitudes] - ***Destino sistema cambio de estado
2014-10-30 10:36:58,309 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:36:58,309 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-30 10:36:58,309 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-30 10:36:58,327 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(894b33803893dbfa3b143f2
2014-10-30 10:36:58,333 [INFO] [DepachadorDeSolicitudes] - AT aee9ac0f2ae37c9209c7a1fd9144c8d10c16778 3 1 31 27
2014-10-30 10:36:58,333 [DEBUG] [Enviaador (ACP)] - Enviador envio el mensaje transaccional (numeroServicio(1)secuencial(7)evento(894b33803893dbfa3b143f2
2014-10-30 10:36:58,361 [DEBUG] [pool-3-thread-8] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(4)evento(2cbf4092d33a8f9c6478a73b7
2014-10-30 10:36:58,362 [DEBUG] [pool-3-thread-8] - El evento (2cbf4092d33a8f9c6478a73b7e656e058b06f41f), proveniente de la huella (82712074) es nuevo
2014-10-30 10:36:58,364 [DEBUG] [pool-3-thread-8] - Se registra el origen (Origenes(llave=10-1-3-InegocioSolicitante=10subSistemaSolicitante=1entidadSo
2014-10-30 10:36:58,362 [INFO] [pool-3-thread-8] - E 2cbf4092d33a8f9c6478a73b7e656e058b06f41f 4 1 11
2014-10-30 10:36:58,366 [DEBUG] [DepachadorDeSolicitudes] - Despachador como el msg para atender (numeroServicio(1)secuencial(4)evento(2cbf4092d33a8f9c
2014-10-30 10:36:58,367 [INFO] [DepachadorDeSolicitudes] - ***Destino sistema cambio de estado
2014-10-30 10:36:58,367 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:36:58,367 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-30 10:36:58,367 [INFO] [DepachadorDeSolicitudes] - ***Estado: FINAL
2014-10-30 10:36:58,377 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(d664948b12c6dcdf9f5e6f
2014-10-30 10:36:58,384 [INFO] [DepachadorDeSolicitudes] - AT 2cbf4092d33a8f9c6478a73b7e656e058b06f41f 4 1 15 18
```

```
2014-10-30 10:36:22,197 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el nro para atender (numeroServicio(1)secuencial(3)evento(a750ba1764c2c71c
2014-10-30 10:36:22,200 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:36:22,201 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-30 10:36:22,202 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-30 10:36:22,203 [INFO] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(2d8738c8e2299872abdc97
2014-10-30 10:36:22,216 [DEBUG] [DepachadorDeSolicitudes] - AT a760ba1764c2c71cdd280cd3f6e43db7e216bd9 3 1 22 19
2014-10-30 10:36:22,232 [DEBUG] [Enviador (ACP)] - Enviador envia el mensaje transaccional (numeroServicio(1)secuencial(9)evento(2d8738c8e2299872abdc977
2014-10-30 10:36:22,309 [INFO] [pool-3-thread-9] - S 2d8738c8e2299872abdc9773cf9cdd1b08af109b 9 1 16 86
2014-10-30 10:36:22,310 [DEBUG] [pool-3-thread-9] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(3)evento(a760ba1764c2c71cdd280cd3
2014-10-30 10:36:22,311 [DEBUG] [pool-3-thread-9] - El evento (a760ba1764c2c71cdd280cd3f6e43db7e216bd9), proveniente de la huella (777380073) ya se ha
2014-10-30 10:36:22,312 [DEBUG] [pool-3-thread-9] - Se registra el origen (Origenes(llave=10-1-3-inegocioSolicitante=10subSistemaSolicitante=1entidadS
2014-10-30 10:36:22,321 [INFO] [pool-3-thread-9] - E a760ba1764c2c71cdd280cd3f6e43db7e216bd9 3 1 11
2014-10-30 10:36:22,322 [DEBUG] [pool-3-thread-9] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(4)evento(07f6482f163325f41ce095696
2014-10-30 10:36:22,324 [DEBUG] [pool-3-thread-9] - El evento (07f6482f163325f41ce095696fbd0f4c66d8aac9), proveniente de la huella (137528828) es nuev
2014-10-30 10:36:22,325 [DEBUG] [pool-3-thread-9] - Se registra el origen (Origenes(llave=10-1-3-inegocioSolicitante=10subSistemaSolicitante=1entidadS
2014-10-30 10:36:22,336 [INFO] [pool-3-thread-9] - E 07f6482f163325f41ce095696fbd0f4c66d8aac9 4 1 15
2014-10-30 10:36:22,357 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el nro para atender (numeroServicio(1)secuencial(4)evento(07f6482f163325f4
2014-10-30 10:36:22,359 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:36:22,360 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 2
2014-10-30 10:36:22,361 [INFO] [DepachadorDeSolicitudes] - ***Estado: FIN
2014-10-30 10:36:22,362 [INFO] [pool-3-thread-8] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(4)evento(07f6482f163325f41ce095696
2014-10-30 10:36:22,360 [DEBUG] [pool-3-thread-8] - El evento (07f6482f163325f41ce095696fbd0f4c66d8aac9), proveniente de la huella (777380073) ya se ha
2014-10-30 10:36:22,369 [DEBUG] [pool-3-thread-8] - Se registra el origen (Origenes(llave=10-1-3-inegocioSolicitante=10subSistemaSolicitante=1entidadS
2014-10-30 10:36:22,447 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(2ce971a1285666ab30361)
2014-10-30 10:36:22,449 [INFO] [pool-3-thread-8] - E 07f6482f163325f41ce095696fbd0f4c66d8aac9 4 1 89
2014-10-30 10:36:22,664 [INFO] [DepachadorDeSolicitudes] - AT 07f6482f163325f41ce095696fbd0f4c66d8aac9 4 1 36 157
2014-10-30 10:37:10,425 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el nro para atender (numeroServicio(1)secuencial(1)evento(340c76d8b8d852d
2014-10-30 10:37:10,430 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:37:10,430 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 3
2014-10-30 10:37:10,430 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-30 10:37:10,450 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(7489fa2b76a0a089e993
2014-10-30 10:37:10,453 [INFO] [DepachadorDeSolicitudes] - AT 340c76d8b8d852d792e67132d81ab3534bc008 1 1 26 28
2014-10-30 10:37:10,486 [DEBUG] [pool-3-thread-11] - Enviador envia el mensaje transaccional (numeroServicio(1)secuencial(1)evento(7489fa2b76a0a089e993)
2014-10-30 10:37:10,486 [DEBUG] [pool-3-thread-11] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(2)evento(7dfca76d063513940aed813
2014-10-30 10:37:10,487 [DEBUG] [pool-3-thread-11] - El evento (7dfca76d063513940aed81306899e023a9e36ab), proveniente de la huella (2040662576) es nue
2014-10-30 10:37:10,488 [DEBUG] [pool-3-thread-11] - Se registra el origen (Origenes(llave=10-1-3-inegocioSolicitante=10subSistemaSolicitante=1entidadS
2014-10-30 10:37:10,498 [INFO] [pool-3-thread-11] - E 7dfca76d063513940aed81306899e023a9e36ab 2 1 13
2014-10-30 10:37:10,502 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el nro para atender (numeroServicio(1)secuencial(2)evento(7dfca76d0635139
2014-10-30 10:37:10,502 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:37:10,502 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 3
2014-10-30 10:37:10,502 [INFO] [DepachadorDeSolicitudes] - ***Estado: FIN
2014-10-30 10:37:10,509 [INFO] [pool-3-thread-11] - S 7dfca76d063513940aed81306899e023a9e36ab 11 1 12 49
2014-10-30 10:37:10,512 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(bb413389a9fa659cb86656f
2014-10-30 10:37:10,516 [INFO] [DepachadorDeSolicitudes] - AT 7dfca76d063513940aed81306899e023a9e36ab 2 1 17 14
2014-10-30 10:38:35,369 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el nro para atender (numeroServicio(1)secuencial(6)evento(cae75276dbcb93b8
2014-10-30 10:38:35,360 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:38:35,360 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 4
2014-10-30 10:38:35,360 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-30 10:38:35,369 [INFO] [pool-3-thread-12] - E cae75276dbcb93b882830aed43292c913e721b88 6 1 64
2014-10-30 10:38:35,373 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(2c323646872ab6e6c2560b)
2014-10-30 10:38:35,380 [INFO] [DepachadorDeSolicitudes] - AT cae75276dbcb93b882830aed43292c913e721b88 5 1 64 21
2014-10-30 10:38:35,389 [DEBUG] [pool-3-thread-12] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(6)evento(4e3e189bbdb86f76c78a30bc
2014-10-30 10:38:35,390 [DEBUG] [pool-3-thread-12] - El evento (4e3e189bbdb86f76c78a30bc104528dbfa05fa), proveniente de la huella (82712074) es nuevo
2014-10-30 10:38:35,392 [DEBUG] [pool-3-thread-12] - Se registra el origen (Origenes(llave=10-1-3-inegocioSolicitante=10subSistemaSolicitante=1entidadS
2014-10-30 10:38:35,396 [INFO] [pool-3-thread-12] - E 4e3e189bbdb86f76c78a30bc104528dbfa05fa 6 1 8
2014-10-30 10:38:35,399 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el nro para atender (numeroServicio(1)secuencial(6)evento(4e3e189bbdb86f77
2014-10-30 10:38:35,400 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:38:35,400 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 4
2014-10-30 10:38:35,400 [INFO] [DepachadorDeSolicitudes] - ***Estado: FIN
2014-10-30 10:38:35,409 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(a0eac96c0cd93997afe87)
2014-10-30 10:38:35,416 [INFO] [DepachadorDeSolicitudes] - AT 4e3e189bbdb86f76c78a30bc104528dbfa05fa 6 1 11 17
2014-10-30 10:42:10,658 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el nro para atender (numeroServicio(1)secuencial(7)evento(82f3b9f8117ab74e36a0d21c
2014-10-30 10:42:10,658 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:42:10,658 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 5
2014-10-30 10:42:10,658 [INFO] [DepachadorDeSolicitudes] - ***Estado: PROCESO
2014-10-30 10:42:10,664 [INFO] [pool-3-thread-14] - E 82f3b9f8117ab74e36a0d21c7ab71ae5a460a8e 7 1 15
2014-10-30 10:42:10,666 [DEBUG] [pool-3-thread-14] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(7)evento(82f3b9f8117ab74e36a0d21c
2014-10-30 10:42:10,670 [DEBUG] [pool-3-thread-14] - El evento (82f3b9f8117ab74e36a0d21c7ab71ae5a460a8e), proveniente de la huella (777380073) ya se ha
2014-10-30 10:42:10,672 [DEBUG] [pool-3-thread-14] - Se registra el origen (Origenes(llave=10-1-3-inegocioSolicitante=10subSistemaSolicitante=1entidadS
2014-10-30 10:42:10,684 [INFO] [pool-3-thread-14] - E 82f3b9f8117ab74e36a0d21c7ab71ae5a460a8e 7 1 20
2014-10-30 10:42:10,686 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(2202fa2bfce042e9da1b1)
2014-10-30 10:42:10,697 [INFO] [pool-3-thread-14] - Se recibio el mensaje transaccional (numeroServicio(1)secuencial(8)evento(c44a376e89566144ddcf87019
2014-10-30 10:42:10,702 [DEBUG] [pool-3-thread-14] - El evento (c44a376e89566144ddcf8701974112cc651605365) proveniente de la huella (82712074) es nuevo
2014-10-30 10:42:10,704 [DEBUG] [pool-3-thread-14] - Se registra el origen (Origenes(llave=10-1-3-inegocioSolicitante=10subSistemaSolicitante=1entidadS
2014-10-30 10:42:10,706 [DEBUG] [pool-3-thread-14] - Se registra el origen (Origenes(llave=10-1-3-inegocioSolicitante=10subSistemaSolicitante=1entidadS
2014-10-30 10:42:10,708 [INFO] [pool-3-thread-14] - E c44a376e89566144ddcf8701974112cc651605365 8 1 9
2014-10-30 10:42:10,709 [DEBUG] [DepachadorDeSolicitudes] - Depachador como el nro para atender (numeroServicio(1)secuencial(8)evento(c44a376e89566144
2014-10-30 10:42:10,710 [INFO] [DepachadorDeSolicitudes] - ***Proceso: 4
2014-10-30 10:42:10,710 [INFO] [DepachadorDeSolicitudes] - ***Servicio: 5
2014-10-30 10:42:10,710 [INFO] [DepachadorDeSolicitudes] - ***Estado: FIN
2014-10-30 10:42:10,719 [DEBUG] [DepachadorDeSolicitudes] - Se coloco en el ADF el mensaje (numeroServicio(1)secuencial(0)evento(6e9571e76c8081a61e1d)
2014-10-30 10:42:10,729 [INFO] [DepachadorDeSolicitudes] - AT c44a376e89566144ddcf8701974112cc651605365 8 1 10 20
```

Figura 5.40 Bitácora del gestor para el sistema especializado.



#### **5.6.4.2. Resultados del Caso de Prueba IV**

Se demostró que la integración del sistema gestor en una arquitectura distribuida, tolerante a fallas y especializado no es intrusiva y funciona como se esperaba, monitoreando cada etapa del proceso en ejecución y registrando los cambios de estados de los servicios gestionables.

## 6. Conclusiones y Líneas Futuras

### 6.1. Conclusiones

Para cualquier empresa la información con la que cuenta es la parte esencial de su negocio, su principal activo. En muchos casos esa información es administrada por sistemas críticos ya que de ello depende la organización. De aquí nació la necesidad de contar con un sistema para aplicaciones de misión crítica que permitiera monitorear los procesos de negocio de manera fácil, rápida, sencilla y portable.

Se analizaron varias propuestas para este fin, y se encontró por ejemplo que WSO2 BAM, permite dar seguimiento a las actividades del negocio basadas en una arquitectura SOA dividida en módulos. Esta opción no utiliza agentes para la administración, pero no es flexible si existiera algún cambio en los procesos de la organización.

Otra propuesta es realizar un monitoreo de aplicaciones SOA basado en JAVA, dividido en componentes, su principal objetivo es la simplicidad e independencia de la plataforma ya que está basada en JAVA. La gestión ocurre en tiempo de ejecución y tiene la restricción de que solamente realiza el monitoreo en los servicios implementados en JAVA. Otra desventaja es que almacena la información del monitoreo en archivos planos separados por comas, lo que restringe el tamaño de los datos a almacenar.

También se encontró con una herramienta llamada BigSister que realiza un monitoreo limitado ya que utiliza agentes, los cuales deben ser configurados mediante métricas para la recopilación de la información sobre los estados de los procesos y recursos del sistema. Esta

herramienta está más enfocada hacia la administración de los recursos como: espacio en disco, memoria disponible, bases de datos, etc.

Como se pudo observar ninguna de estas propuestas cumple con las características para la administración de sistemas de misión crítica. Al continuar con la investigación se encontró una novedosa propuesta que combina lo mejor de dos arquitecturas SOA y ADS obteniendo ADSOA.

Este novedoso paradigma está diseñado para el desarrollo de aplicaciones de misión crítica, pero no cuenta con un sistema gestor que le permita dar seguimiento a los procesos de negocio de manera clara y sencilla. Por lo cual se tomó la decisión de realizar el trabajo de este proyecto en dicha arquitectura.

Con esta propuesta se lograron culminar los objetivos planeados con éxito, cumpliendo con cada una de sus especificaciones como se detallan a continuación:

El diseño de la arquitectura a utilizar para el sistema gestor se realizó con el estudio de los protocolos y arquitectura que conforman ADSOA, esto permitió la identificación de los elementos a ser gestionados y la manera de hacerlo, ya que la intención era lograr la integración de este sistema gestor de manera transparente, es decir, que su incorporación no afectara los procesos y servicios en su funcionamiento. Además en esta etapa se modeló la arquitectura del sistema gestor en cinco componentes: gestor, integrador, conector empresarial, base de datos y un servidor de aplicaciones.

El diseño e implementación de las tecnologías que permitieron conocer el estado actual de los procesos y servicios de negocio, se lograron al establecer el protocolo a utilizar para el proceso de monitoreo, extendiendo el protocolo de foliado y el del campo de datos. Esto fue un reto ya que se buscó que la extensión del protocolo fuera lo más simple posible y se obtuvo al incorporar solo tres mensajes con la información siguiente: identificador del proceso en ejecución, identificador del servicio actual, identificador del estado del mismo.

También se implementaron en este objetivo, cuatro componentes de la arquitectura del sistema gestor (gestor, integrador, conector empresarial y la base de datos) con esto se confirmó el funcionamiento y flujo de los mensaje ya mencionados.

Para la implementación y diseño de las tecnologías a utilizar en la detección de fallos en los procesos de negocio y visualización de los mismos, se lograron al modificar y configurar el *Atom* de manera que fuera capaz de identificar si un proceso y sus servicios eran gestionables, para esto se agregaron dos parámetros “esGestionable y numeroProceso”.

Con esto cada Entidad tiene la inteligencia para discernir si el servicio actual es gestionable o no, por lo consiguiente al existir algún fallo en el mismo se puede detectar en el instante y así poderlo informar. Del lado de los servicio a ofrecer por cada Entidad, solamente se indica en su configuración si son gestionables y el identificador del proceso al que pertenecen.

Un punto a enfatizar es que este cambio fue poco intrusivo, es decir, si en algún momento se requiere que un servicio gestionable ya no funja como tal, solamente se eliminan de su

configuración los parámetros “esGestionable y numeroProceso”, y lo más importante no se ve afectado su funcionamiento.

En esta etapa también, se implementó el componente que faltaba, una aplicación web que permitiera el despliegue del estado de los procesos de negocio y sus servicios. Esta aplicación muestra el flujo del proceso, el estado y la etapa en la que se encuentra mediante un grafo.

Con base en la analogía de los organismos vivos, el sistema gestor funge como el sistema nervioso central, donde su tarea es monitorear el estado de los órganos a su alrededor, permitiendo conocer con oportunidad de la existencia de alguna falla en alguno de los órganos, para así tomar la decisión que permita la continuidad del proceso.

Además, este sistema gestor cuenta con las características de ser autónomo, permite la distribución de carga, tiene la capacidad de auto coordinarse y auto controlarse. Cuenta con una tecnología que ofrece alta disponibilidad al permitir el auto recuperación de sus Instancias.

Como se puede observar, este sistema gestor es la mejor propuesta para el monitoreo de procesos de negocio implementados en aplicaciones de misión crítica basadas en una arquitectura ADSOA. El monitoreo permite dar seguimiento a los procesos de manera clara y oportuna, además de que puede ser visto en cualquier lugar y en cualquier momento, ya que fue concebida con la característica de portabilidad.



## **6.2. Líneas Futuras**

### **6.2.1. Herramienta Complementaria para Modelar los Proceso de Negocio de Manera Gráfica.**

Durante la implementación del sistema gestor para aplicaciones de misión crítica basadas en una arquitectura ADSOA, se encontró la necesidad de contar con una herramienta complementaria que permita el modelado de los procesos de negocio de manera gráfica, fácil y sencilla.

Esta herramienta debe permitir crear, eliminar y modificar cualquier proceso de negocio, ya que en cualquier empresa los cambios en los procesos y servicios son constantes y se requiere realizarlos de manera rápida y confiable.

Por otro lado, con esta tecnología también deberá ser posible alimentar la base de datos del sistema gestor con la información necesaria del proceso de negocio para su administración, lo que permitiría dar continuidad en la operación.

### **6.2.2. Aportación al Protocolo de Foliado de la Arquitectura ADSOA.**

Durante la investigación y el análisis, se encontró que el protocolo actual de foliado, no permite la reutilización de los números de servicio entre diferentes entidades. El protocolo tiene un límite máximo de asignación, por lo que el organismo tiene un tope de crecimiento, además que requiere de una administración para la asignación de los números de servicio para garantizar que no se traslapen.

Sobre estas limitaciones, se requiere revisar los cambios necesarios al protocolo de manera que los números de servicios tengan la capacidad de ser reutilizados para evitar estos inconvenientes.

### **6.2.3. Línea de Investigación Sobre el Cómputo Autonómico.**

Esta propuesta está encaminada para que el sistema gestor pueda tener la inteligencia para reparar las fallas que pueda identificar, además de que pueda contar con las herramientas necesarias para tomar la mejor decisión y realizar la reparación sin la necesidad de la intervención humana.

Actualmente existe un antecedente de este tipo de paradigmas y es el protocolo de auto recuperación de ADSOA. Este protocolo permite saber si alguna Instancia dentro del campo de datos no está funcionando correctamente, al confirmar este evento se intenta la recuperación de esa Instancia y de no ser posible se genera una nueva automáticamente [35].

### **6.2.4. Incorporar el Monitoreo de Sistemas con Arquitectura Distinta a ADSOA.**

En esta investigación se llegó a la conclusión de que era posible incorporar el monitoreo en sistemas diseñados en una arquitectura distinta a ADSOA. El diseño del sistema gestor permite esta incorporación ya que solo se necesita conocer como está conformado el proceso de negocio, es decir, que servicios tiene que atender. Solamente se necesita el mecanismo de que alimente al sistema gestor para poder llevar a cabo esta tarea.

## Bibliografía

- [ 1 ] I. Sommerville, *Ingeniería del software*. Ed. Pearson Educación, Madrid, 2005.
- [ 2 ] L. C. Coronado, J. A. González, P. J. Hernández, C. P. Leguizamo, *An Autonomous Decentralized Service Oriented Architecture for High Reliable Service Provision*, Tenth International Symposium on Autonomous Decentralized Systems, 2011
- [ 3 ] S. Hayward, *Positions 2005: SOA adds flexibility to business processes*, Gartner, February 2005.
- [ 4 ] P. H. Kaigui Wu, J. Wen, Jie Xu, *Monitoring Resources Allocation for Service Composition under Different Monitoring Mechanisms*, International Conference on Complex, Intelligent and Software Intensive Systems, 2011
- [ 5 ] B. Bonev, S. Llieva, *Monitoring Java based SOA applications*, International Conference on Computer Systems and Technologies, 2012
- [ 6 ] M. M. Polycarpou, *Monitoring and Control of Critical Infrastructure Systems*, IEEE Computational Intelligence Society, University of Cyprus: [http://www.dic.uchile.cl/iecc-cis/files/DLP\\_Chile\\_M\\_Polycarpou.pdf](http://www.dic.uchile.cl/iecc-cis/files/DLP_Chile_M_Polycarpou.pdf)
- [ 7 ] BigSister, sitio web [en línea]. Disponible en: <http://www.bigsister.ch> y <http://www.bigsister.ch/bigsister.pdf>
- [ 8 ] Waseda University, Departamento de Ciencias e Ingeniería, Sitio web [en línea]. Disponible en: <http://www.aoni.waseda.jp/kinji.mori/ads.html>
- [ 9 ] JSEEBrain, sitio web [en línea]. Disponible en: <http://www.j2cebrain.com/java-J2ee-basic-components-of-soa.html>
- [ 10 ] Hyperic sitio web [en línea]. Disponible en: <http://www.hyperic.com> y <https://support.hyperic.com/download/attachments/59375779/HypericHQTour.pdf>
- [ 11 ] WSO2 sitio web [en línea]. Disponible en: <http://wso2.com> y <http://wso2.com/library/articles/business-activity-monitoring-introduction/>

- [ 12 ] Gartner Inc., *Business Activity Monitorig: Calm Before the Storm*, Research, 2002:  
<http://www.gartner.com/resources/105500/105562/105562.pdf>
- [ 13 ] P. Hershey, D. Runyon, *SOA Monitoring for Enterprise Computing Systems*, 11<sup>th</sup> IEEE International Enterprise Distributed Object Computing Conference, 2007
- [ 14 ] F. Zakaria Safy, M. El-Ramly, A. Salah, *Runtime Monitoring of SOA Applications: Importance, Implementations and Challenges*, IEEE Seventh International Symposium on Service-Oriented System Engineering, 2013
- [ 15 ] K. Mori, *Autonomous Decentralized Systems: Concept, Data Field Architecture and Future Trends*, International Symposium on, 1993
- [ 16 ] M. J. Mendez, F. A. Silva, *Mobile Agents in Autonomous Decentralized Systems, Integration of Heterogeneous Systems*, Fourth International Symposium on 1999
- [ 17 ] C. Bauer, *Hibernate in Action*. Ed. Manning Publications Co., USA, 2005.
- [ 18 ] Open Message Qucuc, *Developer's Guide for Java Clients*. Oracle America., USA, 2013.
- [ 19 ] Tutorial Java Servlets, sitio web [en línea]. Disponible en:  
<http://tutorials.jenkov.com/java-servlets/index.html>
- [ 20 ] Integrating WebSockets and JMS with CDI Events in Java EE7, sitio web [en línea]. Disponible en:  
[https://blogs.oracle.com/brunoborges/entry/integrating\\_websockets\\_and\\_jms\\_with](https://blogs.oracle.com/brunoborges/entry/integrating_websockets_and_jms_with) y  
<http://technology.amis.nl/2013/06/22/java-ee-7-cjb-publishing-cdi-events-that-are-pushed-over-websocket-to-browser-client/>
- [ 21 ] JavaScript Reference, sitio web [en línea]. Disponible en:  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>
- [ 22 ] jQuery API, sitio web [en línea]. Disponible en: <http://api.jquery.com/>
- [ 23 ] jTable, sitio web [en línea]. Disponible en: <http://www.jtable.org/> y  
<http://www.programming-freec.com/2014/02/jtable-pagination-in-java-web.html>
- [ 24 ] Cytoscape.js, sitio web [en línea]. Disponible en: <http://cytoscape.github.io/cytoscape.js/>
- [ 25 ] Json, sitio web [en línea]. Disponible en: <http://json.org/>

- [ 26 ] Banco de México, Sistemas de pago de bajo valor, sitio web [en línea]. Disponible en: <http://www.banxico.org.mx/SicInternet/consultarDirectorioInternetAction.do?accion=consultarDirectorioCuadros&sector=21&sectorDescripcion=Sistemas%20de%20pago%20de%20bajo%20valor&locale=es>
- [ 27 ] Caniuse, sitio web [en línea]. Disponible en: <http://caniuse.com/>
- [ 28 ] Z. Ruvalcaba, A. Boehm, , *murach's HTML5 and CSS3*. Ed. Murach, United States, 2012.
- [ 29 ] The WebSocket API, sitio web [en línea]. Disponible en: <http://www.w3.org/TR/websockets/>
- [ 30 ] WebSockets, sitio web [en línea]. Disponible en: <https://developer.mozilla.org/es/docs/WebSockets>
- [ 31 ] Banco de México, Misión, sitio web [en línea]. Disponible en: <http://www.banxico.org.mx/acerca-del-banco-de-mexico/acerca-del-banco-mexico.html>
- [ 32 ] Banco de México, Subastas, sitio web [en línea]. Disponible en: <http://www.banxico.org.mx/sistema-financiero/material-educativo/intermedio/subastas-y-colocacion-de-valores/subastas-colocacion-valores.html>
- [ 33 ] Banco de México, Glosario, sitio web [en línea]. Disponible en: <http://www.banxico.org.mx/divulgacion/glosario/glosario.html>
- [ 34 ] Banco de México, Circular 6/2012, sitio web [en línea]. Disponible en: <http://www.banxico.org.mx/disposiciones/circulares/{EC72B309-BBF4-B520-C6BD-3553AEBA7B4}.pdf>
- [ 35 ] R. Garcia Gomez, S. Godinez Borja, P. Hernandez Torre, C. Perez Leguizamo, *Self-Recovery Technology in Distributed Service-Oriented Mission Critical Systems for Fault Tolerance*, The Sixth International Conference on Dependability, 2013