

**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

**FACULTAD DE INGENIERÍA**

**DIVISIÓN DE ESTUDIOS PROFESIONALES PARA EJECUTIVOS**

**CARRERA DE INGENIERÍA DE SISTEMAS**

**DESARROLLO DE UNA SOLUCION GENERAL DE INTEGRACION DE**

**PROCESOS DE NEGOCIOS BASADA EN MIDDLEWARES**

**PROYECTO PROFESIONAL PRESENTADO POR**

**Julio Martín Zelada Briceño**

**Juan Luis Toma Kiyamu**

**PARA OPTAR EL TÍTULO DE INGENIERO DE SISTEMAS**

**Lima, Noviembre de 2008**

## **RESUMEN**

El presente proyecto profesional tiene como finalidad el desarrollo de una solución flexible y estándar que permita acelerar la integración de los procesos de negocios que se encuentran contenidos dentro de dos o más sistemas informáticos, los cuales pueden residir en una o varias organizaciones, de tal forma que puedan colaborar entre sí en tiempo real. Esta solución será un complemento de un middleware de comunicación, hará uso de las funcionalidades de este para establecer conexión con diversos sistemas.

En los últimos años, los middlewares (Software que trabaja entre aplicaciones y la red. Administra la interacción entre diferentes aplicaciones corriendo en diferentes plataformas [FOLDOC:2008:1]) han sido la solución para integrar los sistemas informáticos. Ellos resuelven la complejidad derivada de los protocolos de comunicación y la plataforma operativa y dejan en manos de los desarrolladores el manejo de las reglas de negocios de la organización, lo que se traduce en el desarrollo de aplicaciones específicas para cada necesidad. Las nuevas soluciones se basan en el concepto de “procesos de negocio”. Es por ello que el objetivo principal de este proyecto radica en diseñar, desarrollar, interpretar y ejecutar un modelo de datos de propósito general, que se adapte a cualquier tipo de necesidad de integración de sistemas de la organización. La solución resultante servirá de apoyo para diversos sectores empresariales (financiero, gubernamental, salud, etc.).

El presente entregable comprende las etapas de fundamentación teórica, análisis de requerimientos y diseño de la solución. Se ha incluido el estudio de efectividad económica del proyecto propuesto

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>I</b>
<b>CAPÍTULO I FUNDAMENTACIÓN TEÓRICA</b> .....	<b>1</b>
1.1. INTRODUCCIÓN.....	1
1.2. MARCO TEÓRICO .....	1
1.2.1. Los nuevos paradigmas.....	1
1.2.2. Arquitectura de las soluciones .....	4
1.2.3. Procesos generales para el desarrollo de un proyecto de integración de sistemas basado en el modelamiento de los procesos de negocio.....	5
1.2.4. Complemento de las soluciones.....	10
1.2.5. Interacción con los sistemas externos.....	11
1.2.6. Recuperación de la información de los procesos de negocio .....	12
1.2.7. Seguridad .....	13
1.3. OBJETO DE ESTUDIO .....	14
1.4. CAMPO DE ACCIÓN .....	16
1.5. ANÁLISIS CRÍTICO DE LOS PROBLEMAS DE LA ORGANIZACIÓN .....	20
1.5.1. Situación problemática .....	20
1.5.2. Problema a resolver.....	25
1.6. CONCLUSIONES .....	25
<b>CAPÍTULO II PROPUESTA DE SOLUCIÓN</b> .....	<b>26</b>
2.1. INTRODUCCIÓN.....	26
2.2. OBJETIVOS DEL PROYECTO .....	26
2.2.1. Objetivo general.....	26
2.2.2. Objetivos específicos .....	27
2.2.3. Fundamentación de los objetivos propuestos .....	27
2.2.4. Indicadores de logro de los objetivos .....	28
2.3. BENEFICIOS DEL PROYECTO.....	28
2.3.1. Beneficios tangibles .....	28
2.3.2. Beneficios intangibles .....	29
2.4. CONCLUSIONES .....	29

<b>CAPÍTULO III MODELADO DEL NEGOCIO.....</b>	<b>30</b>
3.1. INTRODUCCIÓN.....	30
3.2. MODELO DE CASOS DE USO DEL NEGOCIO.....	30
3.2.1. Lista de actores del negocio.....	30
3.2.2. Diagrama de casos de uso del negocio .....	31
3.3. REGLAS DEL NEGOCIO .....	31
3.4. REALIZACIÓN DE LOS CASOS DE USO DEL NEGOCIO .....	32
3.4.1. Especificación de los casos de uso del negocio .....	32
3.4.2. Diagramas de procesos .....	39
3.4.3. Lista de actividades a automatizar .....	43
3.5. MODELO DE ANÁLISIS DEL NEGOCIO.....	43
3.5.1. Lista de trabajadores del negocio.....	43
3.5.2. Lista de las entidades del negocio.....	45
3.5.3. Diagrama de clases del negocio.....	49
3.6. CONCLUSIONES .....	49
<b>CAPÍTULO IV REQUERIMIENTOS.....</b>	<b>50</b>
4.1. INTRODUCCIÓN.....	50
4.2. ESPECIFICACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA .....	50
4.2.1. Funcionalidad .....	50
4.2.2. Usabilidad.....	51
4.2.3. Confiabilidad .....	52
4.2.4. Rendimiento .....	52
4.2.5. Soporte.....	52
4.2.6. Restricciones de diseño .....	52
4.2.7. Documentación de usuario y sistema de ayuda.....	53
4.2.8. Componentes adquiridos .....	53
4.2.9. Interfase .....	53
4.2.10. Licenciamiento .....	54
4.2.11. Requerimientos legales y derechos de autor.....	54
4.2.12. Estándares aplicables .....	54
4.3. SEGURIDAD DEL SOFTWARE.....	54
4.4. MODELO DE CASOS DE USO DEL SISTEMA.....	55
4.4.1. Lista de los actores del sistema.....	55
4.4.2. Diagrama de actores del sistema. ....	57
4.4.3. Diagrama de paquetes.....	57
4.4.4. Diagrama de casos de uso del sistema por paquete. ....	58
4.5. MATRIZ DE MODELO DEL NEGOCIO Y MODELO DEL SISTEMA. ....	61
4.6. ESPECIFICACIÓN DE ALTO NIVEL DE LOS CASOS DE USO DEL SISTEMA.....	63
4.7. DIAGRAMA DE MODELO CONCEPTUAL DEL SISTEMA.....	69
4.8. PRIORIZACIÓN DE LOS CASOS DE USO.....	72
4.8.1. Clasificación de los casos de uso.....	72
4.8.2. Ciclos de desarrollo. ....	73
4.9. ESPECIFICACIÓN DE CASOS DE USO DEL SISTEMA (CASOS DE USO DEL NÚCLEO CENTRAL).....	73
4.9.1. Caso de uso: Ejecutar proceso .....	73
4.9.2. Caso de uso: Validar proceso .....	75
4.9.3. Caso de uso: Gestionar proceso.....	77
4.11. CONCLUSIONES.....	81

<b>CAPÍTULO V ARQUITECTURA DE SOFTWARE.....</b>	<b>82</b>
5.1. INTRODUCCIÓN.....	82
5.2. METAS Y RESTRICCIONES DE LA ARQUITECTURA.....	82
5.3. VISTA DE CASOS DE USO.....	84
5.3.1 Diagrama de actores y sus relaciones.....	84
5.3.2 Diagrama de casos de uso relevantes a la arquitectura del sistema.....	85
5.4. MECANISMOS.....	86
5.5. VISTA DE MÓDULOS.....	87
5.6. VISTA DE COMPONENTES.....	88
5.7. VISTA DE DESPLIEGUE.....	89
5.7.1 Vista de despliegue de diseño.....	89
5.7.2 Vista de despliegue de ejecución.....	89
5.8. VISTA DE DATOS.....	90
5.9. CONCLUSIONES.....	91
<b>CAPÍTULO VI PRUEBAS.....</b>	<b>92</b>
6.1. INTRODUCCIÓN.....	92
6.2. PRUEBAS FUNCIONALES DEL CUS: EJECUTAR PROCESO.....	92
6.2.1. OBJETIVO DE LA PRUEBA.....	92
6.2.2. CASO DE PRUEBA: <EJEPROPF01>.....	92
6.2.3. CASO DE PRUEBA: <EJEPROPF02>.....	94
6.2.4. CASO DE PRUEBA: <EJEPROPF03>.....	95
6.2.5. CASO DE PRUEBA: <EJEPROPF04>.....	95
6.3. PRUEBAS FUNCIONALES DEL CUS: VALIDAR LÓGICA DE PROCESO.....	96
6.3.1. OBJETIVO DE LA PRUEBA.....	96
6.3.2. CASO DE PRUEBA: <VALPROPF01>.....	96
6.3.3. CASO DE PRUEBA: <VALPROPF02>.....	97
6.3.4. CASO DE PRUEBA: <VALPROPF03>.....	98
6.4. PRUEBAS FUNCIONALES DEL CUS: GESTIONAR PROCESO.....	100
6.4.1. OBJETIVO DE LA PRUEBA.....	100
6.4.2. CASO DE PRUEBA: <GESPROPF01>.....	100
6.4.3. CASO DE PRUEBA: <GESPROPF02>.....	101
6.4.4. CASO DE PRUEBA: <GESPROPF03>.....	101
6.5. MATRIZ DE TRAZABILIDAD ENTRE CASOS DE USO Y CASOS DE PRUEBA.....	102
6.6. CONCLUSIONES.....	103
<b>CAPÍTULO VII ADMINISTRACIÓN DEL PROYECTO.....</b>	<b>104</b>
7.1. INTRODUCCIÓN.....	104
7.2. PROJECT CHARTER.....	104
7.3. CRONOGRAMA DE EJECUCIÓN DEL PROYECTO.....	109
7.4. GESTIÓN DEL ALCANCE.....	113
7.5. CONCLUSIONES.....	114
<b>CONCLUSIONES.....</b>	<b>115</b>
<b>ANEXO 1 TIPOS ESPECIALIZADOS DE MIDDLEWARE.....</b>	<b>116</b>
<b>ANEXO 2 DESCRIPCIÓN DEL MIDDLEWARE SIX/TCL.....</b>	<b>118</b>
<b>ANEXO 3 ANÁLISIS.....</b>	<b>125</b>

<b>GLOSARIO DE TÉRMINOS .....</b>	<b>134</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>139</b>
<b>BIBLIOGRAFÍA .....</b>	<b>140</b>

## **LISTAS ESPECIALES**

### **Ilustraciones**

Ilustración 1: Herramienta de edición de las reglas de negocios.....	7
Ilustración 2: Esquema de una solución intrusiva .....	11
Ilustración 3: Esquema de una solución no intrusiva .....	12
Ilustración 4: Componentes de una solución de integración de Novatronic .....	19
Ilustración 5: Implicancias del tamaño del proyecto en la solución.....	20
Ilustración 6: Integración de estaciones cliente .....	22
Ilustración 7: Conexión a múltiples servidores .....	23
Ilustración 8: Interacción con sistemas externos .....	24
Ilustración 9: Diagrama de actores del sistema .....	57
Ilustración 10: Diagrama de paquetes del sistema.....	57
Ilustración 11: Casos de uso del paquete: MODELADOR .....	58
Ilustración 12: Casos de uso del paquete: MONITOR .....	59
Ilustración 13: Casos de usos del paquete SEGURIDAD .....	60
Ilustración 14: Casos de uso del paquete RUNTIME .....	60

## **INTRODUCCIÓN**

La integración de sistemas ha sido siempre una necesidad desde la aparición de las computadoras. Los problemas que plantea han requerido soluciones especializadas que permitan manejar distintos sistemas operativos y protocolos de comunicación. Dentro de la evolución natural de las soluciones que se ofrecían, surgieron los middlewares, desde entonces son la solución más aceptada para la integración de sistemas.

El mercado de la integración de los sistemas informáticos es muy competitivo a nivel mundial, En el Perú, generalmente las organizaciones que han requerido este tipo de servicio han solucionado sus problemas mediante middlewares desarrollados en el extranjero.

Dentro de este segmento de mercado nació Novatronic, empresa peruana que ha logrado desarrollar un middleware capaz de competir con los productos que provienen del extranjero, y ha logrado en estos veinte años posicionarse en el mercado nacional con proyecciones de expansión a nivel latinoamericano.

Actualmente, los proyectos de integración de sistemas desarrollados por Novatronic se ejecutan en dos fases bien definidas:

1. La instalación del middleware, responsable de manejar la capa de comunicación y que es la base para la integración de los sistemas involucrados en el proyecto.
2. El desarrollo de aplicaciones específicas que utilizan las funcionalidades que provee la capa de comunicación, y que son responsables de manejar las reglas de negocio. En algunos casos, Novatronic sólo es responsable de capacitar al cliente en el uso del middleware y de las facilidades que provee para el desarrollo de las aplicaciones, en otros casos, Novatronic es el responsable de desarrollar y mantener estas aplicaciones dando así un valor agregado al middleware de comunicación instalado en la fase 1.

El punto 2 implica que para cada proyecto desarrollado se tiene una solución a medida y el hecho de escribir el código de los programas para el desarrollo de estas aplicaciones, trae como consecuencia un costo adicional en tiempo, recursos y dinero que es cargado al cliente. Adicionalmente, la solución desarrollada necesitará ser modificada para reflejar los cambios en las reglas de negocio. Como se mencionó anteriormente, en algunos casos el mantenimiento de las aplicaciones es realizado por Novatronic, para lo cual debe destinar tiempo y recursos perdiendo así la oportunidad de desarrollar nuevos proyectos de integración.

Ante estos problemas, el presente proyecto plantea el desarrollo de una solución orientada a cualquier tipo de organización que necesite integrar sus sistemas informáticos de una

forma rápida y efectiva. La solución tendrá a un middleware como base de comunicación y será de propósito general para que se ajuste a los diferentes procesos de negocios contenidos dentro de los sistemas informáticos.

El objetivo principal del presente trabajo es elaborar el diseño del sistema.

Los objetivos que se esperan alcanzar con la solución planteada son: para el cliente, una reducción del tiempo de desarrollo, puesta en producción y mantenimiento de sus proyectos de integración, esto traerá como consecuencia una mejora en su eficiencia en el desarrollo de aplicaciones y mayor rapidez de adaptación a los cambios en las reglas del negocio; el cliente será más independiente de Novatronic, pues los cambios en los procesos, lo realizarán sus analistas de sistemas. Por otro lado Novatronic mejorará su posicionamiento en el mercado, con lo cual podrá competir tanto en el ámbito nacional e internacional.

## **CAPÍTULO I**

### **FUNDAMENTACIÓN TEÓRICA**

#### **1.1. Introducción**

Este capítulo se centrará en la definición y argumentación del problema a resolver dentro de las organizaciones que necesiten implementar la integración de sus sistemas informáticos con el fin de intercambiar información entre ellas.

#### **1.2. Marco teórico**

El modelo que se plantea ha tomado como base el esquema de administración y operación presentado en diversas soluciones que se han analizado como por ejemplo MQIntegrator de IBM, Talarian de Talarian NC, E2EHub de USInteractive, etc. Se han extraído las funcionalidades más importantes y se le ha añadido un valor agregado pensando en la gran variedad de casos que se pueden presentar durante un proyecto de integración de sistemas.

##### **1.2.1. Los nuevos paradigmas**

Para dar frente a los nuevos requerimientos de integración de sistemas, han surgido desde 1998 productos conocidos como EAI (Enterprise Application Integration) que implementan 3 conceptos básicos en las nuevas soluciones de integración:

- ❖ **Procesos de negocios:** Un proceso de negocios es un flujo de información entre varios sistemas. Los procesos de negocios se automatizan mediante la definición de una secuencia de eventos que deben realizarse para completar el ciclo de vida de los mensajes.

Tradicionalmente las organizaciones están divididas en departamentos. Las tareas al interior de estos departamentos son automatizadas mediante algún software empresarial. Sin embargo, muchos procesos de negocios sobrepasan las fronteras de los departamentos, por lo que ciertas tareas son completadas por una aplicación perteneciente a un departamento específico mientras otras tareas son ejecutadas en departamentos diferentes. Las soluciones de integración implementan estos procesos distribuidos como un único proceso donde diferentes tareas son realizadas por diferentes aplicaciones, de una forma atómica, es decir garantizando la ejecución del proceso desde su tarea inicial hasta su tarea final.

Un proceso de negocios es definido de esta manera como un conjunto de funciones de negocio. Se puede dividir en una serie de pasos (unidades de procesamiento) que se denominan tareas, las cuales describen cómo las funciones de negocio son ejecutadas.

Naturalmente, los procesos de negocio se basan en reglas de negocio que encapsulan las políticas de las organizaciones y los requerimientos del proceso, y definen la secuencia de tareas que deben ejecutarse para completar el proceso.

Ya que se ha definido a un proceso de negocios como una sucesión de pasos, es lógico que debe establecerse un evento inicial, el cual “dispara” una instancia en tiempo de ejecución del proceso de negocio. Los eventos son actividades específicas que dirigen la ejecución de tareas. Los demás eventos que forman parte del proceso son eventos servidores que se ejecutan en respuesta al evento inicial.

Para integrar sistemas de la manera que se ha presentado, la información debe pasar del sistema fuente a uno o varios sistemas destino. Las aplicaciones destino deben entender la data recibida, por lo que se requiere traducir la información de modo que ésta sea trasladada a los formatos que cada aplicación entiende.

- ❖ **Publicación/Suscripción:** Una aplicación envía un mensaje (lo publica) a una central de distribución (el middleware), el cual envía los mensajes a los subscriptores registrados. De esta manera las aplicaciones que publican mensajes no tienen que conocer los destinatarios de los mismos y la lista de subscriptores puede modificarse incluso con el sistema ejecutándose.
  
- ❖ **Objetos de negocio:** Un objeto de negocio es un conjunto lógico de datos que durante la ejecución del proceso de negocios es manipulado (transformado y/o filtrado) para ser entregado a los diferentes sistemas.

Un objeto de negocio debe tener las siguientes características:

Atributos, los cuales definen en su conjunto el objeto. Los atributos son las unidades mínimas de datos.

Persistencia, es la habilidad de un objeto de registrar su estado de modo que la información pueda ser reproducida en el futuro. Es importante entender que no es el objeto el que persiste físicamente, sino que la información encapsulada en el objeto se almacena en un medio permanente (tablas o archivos)

Manejador de Persistencia Administra las operaciones relacionadas a la persistencia de los objetos de negocio. Se implementan alrededor del patrón de diseño CRUD <sup>1</sup>. Los objetos de negocio asociados a los sistemas externos deben tener asociado un manejador de persistencia.

Relaciones Diferentes objetos pueden relacionarse unos con otros. Si un atributo es el nombre de otro objeto de negocio o una cadena de objetos, se tienen relaciones padre-hijo o una jerarquía de objetos

Estas 3 definiciones principales forman la base conceptual del nuevo paradigma de solución para la integración de sistemas.

### **1.2.2. Arquitectura de las soluciones**

Del análisis realizado sobre las soluciones existentes, se ha encontrado un patrón similar en la arquitectura de cada propuesta:

La arquitectura básica es la siguiente:

---

<sup>1</sup> Creación, Retrieval, Update and Delete, siglas en ingles de: operaciones de creación, recuperación, actualización y eliminación,

❖ Herramientas para la edición de procesos de negocios.

Estas herramientas son interfaces gráficas que permiten editar (también se utilizan los términos modelar o diseñar) un proceso de negocio como un diagrama de flujo que debe seguirse y adicionalmente administrar el repositorio.

❖ Un repositorio, donde se almacenan los modelos de procesos de negocios definidos y todas las entidades involucradas. El diseño de este repositorio (que en todos los casos es una base de datos) varia por cada propuesta, pero deben poder almacenar como mínimo :

- Las entidades de la organización (tareas, objetos, relaciones)
- La secuencia de un proceso de negocios
- Reglas de direccionamiento de los procesos
- Reglas de transformación de datos

❖ Servidor o Motor del sistema que ejecuta los procesos de negocios. En tiempo real estas aplicaciones interpretan las reglas almacenadas en el repositorio y controlan cada instancia del proceso de negocio que se ejecuta.

**1.2.3. Procesos generales para el desarrollo de un proyecto de integración de sistemas basado en el modelamiento de los procesos de negocio**

Es necesario entender el proceso en que está inmerso el modelo de referencia para el diseño, validación, interpretación y ejecución de las reglas de negocio

Los dos principales procesos que se observan durante el desarrollo de un proyecto de integración de sistemas basados en el modelamiento de procesos de negocio son :

### **1.2.3.1 Administración de los procesos de negocios**

La administración de los procesos de negocios involucra los procesos de diseño y validación de los “procesos de negocios”. En la etapa de desarrollo de la solución de integración se definen los procesos de negocios mediante el uso de una herramienta gráfica; una vez que se termina de definir un proceso de negocio, pasa por un proceso de validación que evalúa la consistencia de la definición realizada y se almacena en el repositorio central

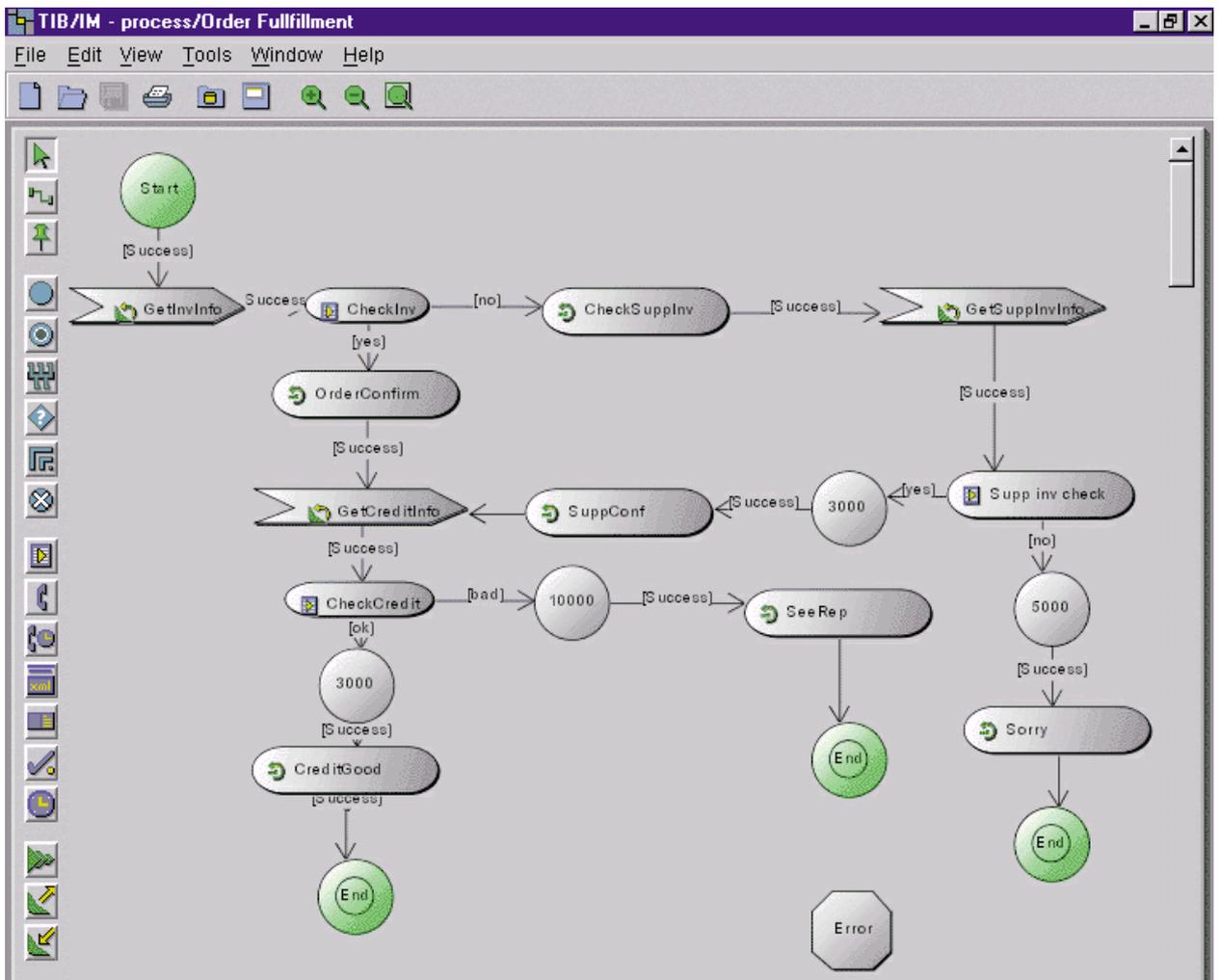
#### **1.2.3.1.1 Diseño de los procesos de negocio**

El diseño o edición de los procesos de negocio consiste en modelar la información que se ha obtenido del análisis del requerimiento particular de integración que se quiere implementar, en forma de reglas de negocio que puedan ser almacenadas en el repositorio central.

Los procesos de negocio se modelan usando los conceptos de eventos, tareas y objetos de negocio que se han descrito en el punto 1.2.1 y definiendo reglas de direccionamiento de la información (quién debe recibirla) y reglas de transformación de datos (qué y cómo debe recibirse).

Para acelerar el tiempo de desarrollo de las aplicaciones, es necesario contar con las herramientas que permitan modelar de una forma más intuitiva los procesos de negocios de tal forma que se reduzca y en algunos casos se elimine el desarrollo de código de los programas aplicativos. Los analistas de sistemas son los usuarios de las soluciones. Ellos podrán realizar el análisis de los procesos de negocios y los requerimientos de información

para traducir ese análisis en reglas de negocio que podrán ser modeladas en el sistema para un rápido desarrollo y puesta en producción (Ver Ilustración 1). De esta manera en la mayoría de los casos, se minimizará la participación de los programadores.



**Ilustración 1: Herramienta de edición de las reglas de negocios**

Las corporaciones desarrolladoras de soluciones de integración como IBM, Bea Systems, Tibco, US Interactive entre otros, han desarrollado soluciones de “procesos de negocios” que trabajan con estos conceptos y ofrecen herramientas visuales para la definición de procesos de negocios mediante la definición de eventos y objetos de negocios, mecanismos estándares para la extracción de la información de un sistema, reglas de transformación de

la información y reglas de enrutamiento de mensajes y/o registro de suscriptores, todo lo cual permite reducir el esfuerzo de codificación de los requerimientos de integración, ofrece mayor flexibilidad a las empresas para adicionar nuevos servicios y modificar los procesos de negocios ya existentes y reduce los tiempos de puesta en producción de las aplicaciones.

#### **1.2.3.1.2 Validación de las reglas de negocios definidas**

En el mercado existen muchas organizaciones, cada una de las cuales maneja diversos tipos de procesos de negocios dependiendo del giro de la empresa. Esta diversidad de procesos de negocios hace que la cantidad de opciones de integración de sistemas sea considerablemente grande.

Por lo tanto, la dificultad del presente proyecto consiste en elaborar un único modelo general de datos que sea capaz de cubrir todas las necesidades que existen de integración de sistemas rescatando las principales propiedades, atributos y características que formarán parte del modelo general de datos.

Debido a la complejidad que implica el diseño de una base de datos de estas características, es necesario validar que las reglas de negocios ingresadas por el analista sean consistentes y esté claramente definida la secuencia de eventos y tareas que deben ejecutarse para un proceso de negocios de modo de evitar que el intérprete y ejecutor de las reglas de negocios realice tareas no contempladas durante el diseño o quede en una condición en la que no pueda discernir cuál es el siguiente paso en la ejecución del proceso de negocio .

### **1.2.3.2 Operación de los procesos de negocios**

Los procesos de negocio se ejecutarán de acuerdo a las reglas de negocios almacenadas y validadas en el repositorio central. Adicionalmente, como todo sistema que opera en producción, el proceso de monitoreo permite controlar el estado del sistema y su funcionamiento.

#### **1.2.3.2.1 Interpretación y ejecución de las reglas de negocio**

Es necesario desarrollar una aplicación que permita interpretar y ejecutar las reglas de negocios definidas por el usuario que se encuentran almacenadas dentro del modelo general de datos. Cada vez que un sistema externo envía una petición al sistema, la petición debe ser interpretada según las reglas de negocio definidas y debe iniciarse la secuencia de eventos que conforman el proceso. La aplicación que ejecuta los procesos de negocio se encarga de distribuir la información a los sistemas que correspondan (tarea de direccionamiento de información) y es responsable de entregarla según se haya definido (tarea de transformación de datos) Al igual que la herramienta de diseño, esta aplicación debe tener la inteligencia suficiente para manejar las condiciones de error y de excepción que se puedan producir durante la interpretación o ejecución de las reglas definidas por los analistas.

#### **1.2.3.2.2 Monitoreo del sistema**

Para las tareas de monitoreo se han encontrado en las soluciones analizadas las siguientes características <sup>2</sup> :

---

<sup>2</sup> Información obtenida en la página de USInteractive - <http://www.usinteractive.com/technology/e2ehub.asp>

- Provee una vista centralizada de todos los procesos de negocios definidos en el modelo de datos
- Permite iniciar, finalizar y detener la aplicación que interpreta y ejecuta las reglas de negocios definidas en el modelo general de datos.
- Seguimiento del desarrollo de los procesos de negocios, tanto en la pantalla de la computadora como por medio de reportes.

Como un valor agregado al proyecto profesional presentado, incorporaremos la opción de elegir qué procesos se colocarán en la memoria del computador donde reside el motor del proceso, con esto se mejorará el tiempo de respuesta a la aplicación cliente.

#### **1.2.4. Complemento de las soluciones**

Las soluciones de software de integración de procesos de negocios analizadas en el mercado son en todos los casos un complemento de un middleware de comunicación ya existente que puede ser de la misma casa desarrolladora –Por ejemplo MQIntegrator utiliza como middleware de comunicación MQSeries, ambos productos de IBM Corporation- o de terceros – E2EHub desarrollado por la empresa US Interactive utiliza como middleware de comunicación a Tuxedo cuyo proveedor es Bea Systems.

Estas soluciones se instalan sobre un middleware transaccional para proveer la capa de comunicaciones y funcionalmente son una capa superior que potencia las funcionalidades del middleware. Por el hecho de formar una unidad funcional con el middleware estas soluciones son conocidas también como middlewares de procesos de negocios.

### 1.2.5. Interacción con los sistemas externos

Se ha observado que las soluciones ofrecidas para la integración de sistemas mediante el modelamiento de reglas de negocio están divididas en dos tipos de acuerdo a los mecanismos que ofrecen para la interacción con los sistemas de la empresa:

**Solución intrusiva** Es aquella en la que no se provee los programas del tipo conector (similares a los programas publicadores explicados en el Escenario 3 del punto 1.2.3), por lo tanto, es necesario modificar las aplicaciones. Para ello, el middleware de comunicación provee las APIs<sup>3</sup> necesarias para que las aplicaciones puedan conversar entre si (ver Ilustración 2). Como ejemplo de esto podemos mencionar al MQIntegrator de IBM, en las que deben modificarse las aplicaciones que desean establecer comunicación usando el middleware. Para habilitar este tipo de soluciones el middleware ofrece una API de comunicaciones (para plataformas Windows y Unix).

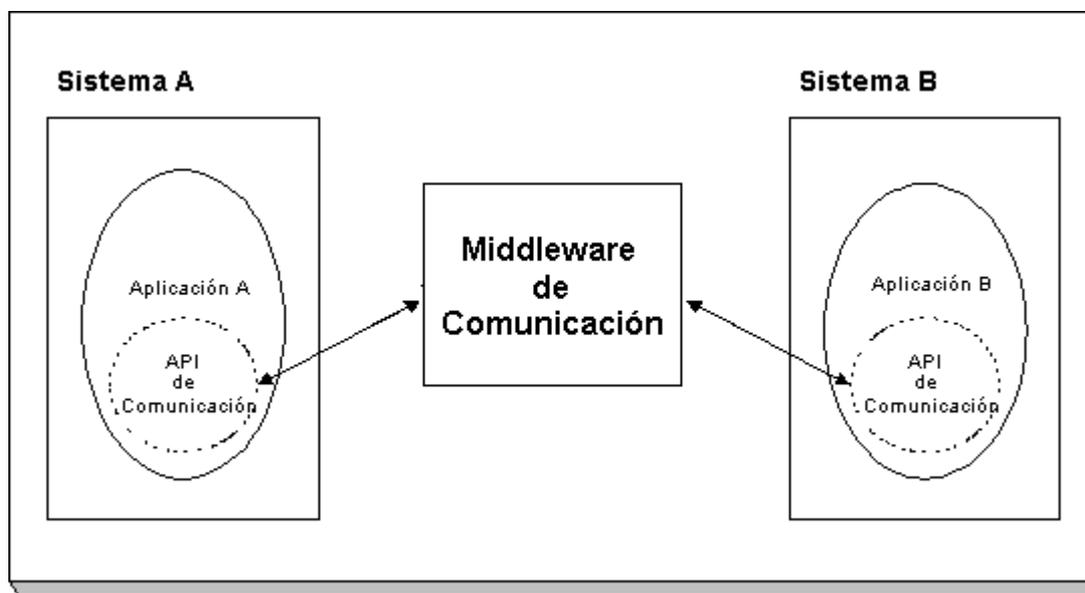
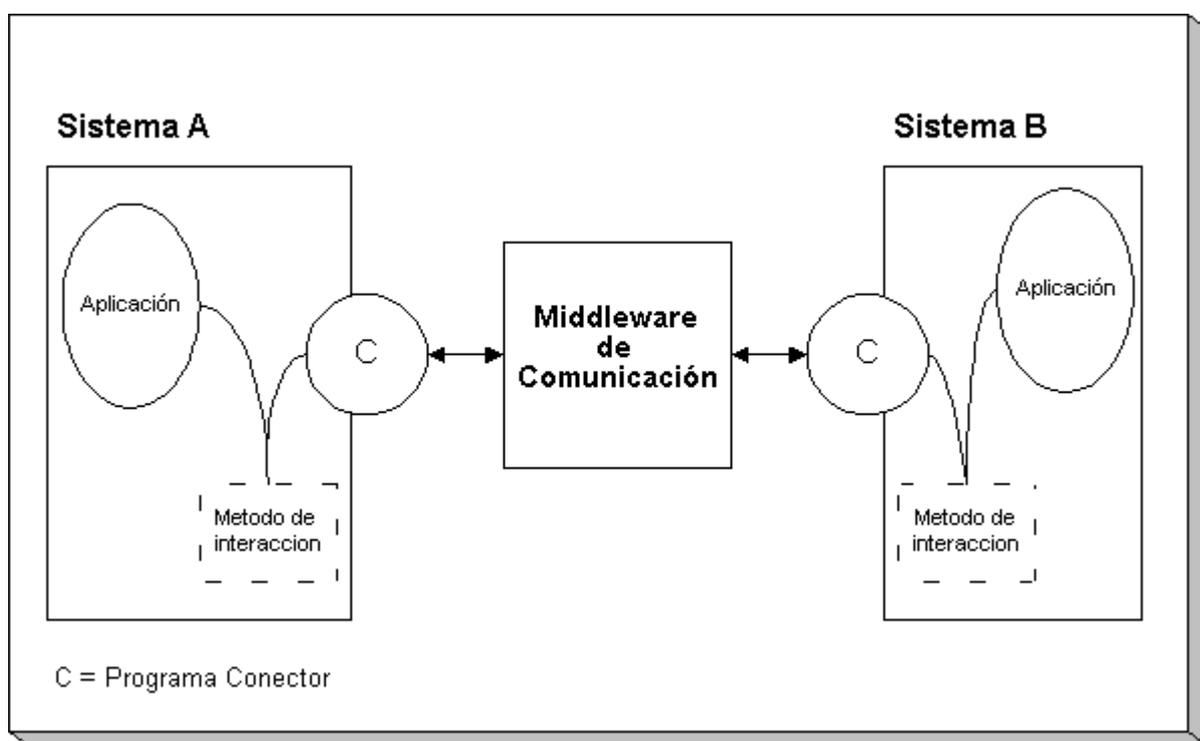


Ilustración 2: Esquema de una solución intrusiva

<sup>3</sup> Siglas de Application Program Interface

Las aplicaciones deben incorporar en su código las funciones que la API provee para enviar y recibir mensajes. Esto implica que es responsabilidad de la aplicación establecer la conexión, invocar a las funciones adecuadamente, preparar los mensajes a enviar, interpretar los mensajes que reciba , interpretar los códigos de error y manejar las excepciones.

**Solución no intrusiva:** En este caso, los sistemas de la empresa no sufren modificaciones. En este tipo de soluciones la conectividad con el middleware de comunicación se logra a través de programas llamados conectores o publicadores. Ver la Ilustración 3



**Ilustración 3 : Esquema de una solución no intrusiva**

**1.2.6. Recuperación de la información de los procesos de negocio**

Este proceso que involucra al motor de la solución tiene como finalidad recuperar la información de los procesos de negocio que se encuentra almacenada en el repositorio, para poder aplicar las reglas que se hayan definido para cada instancia del proceso que se

está ejecutando. Del análisis de las soluciones existentes se concluye que este proceso se realiza de una de dos maneras posibles:

Consulta permanente, bajo este esquema, el repositorio es consultado para cada instancia de un proceso de negocio, en el momento en que dicha instancia está ocurriendo. Tiene la ventaja que cualquier cambio en las reglas almacenadas en el repositorio se puede reflejar inmediatamente en los procesos, pero de otro lado incrementa el tiempo del respuesta del proceso, pues cada paso del proceso implica una operación de base de datos.

Consulta única, en este esquema, el motor de la solución consulta una única vez la información del repositorio, cuando el sistema es inicializado, y almacena todas las reglas de negocio definidas en la memoria principal del sistema. La ventaja principal de este esquema es que opera más rápidamente que el esquema anterior pues se están ejecutando operaciones sobre la memoria y no sobre un archivo físico.

### **1.2.7. Seguridad**

**A nivel de usuarios.** Las soluciones analizadas manejan un control de usuarios basado en contraseñas. Los permisos para cada usuario se otorgan individualmente. No se considera necesario establecer grupos de usuarios para la administración de la seguridad de acceso a los componentes del sistema. Los permisos otorgados definen roles de usuarios, con esto se puede administrar quiénes pueden crear/modificar un proceso de negocio y quién puede controlar el estado del sistema.

**A nivel de la información.** Las soluciones analizadas implementan funcionalidades de encriptación y desencriptación de datos usando técnicas estándares como DES. Esto se

realiza a nivel de un mensaje completo. No se ha encontrado en el análisis realizado la funcionalidad de poder calificar campos dentro de un mensaje como campos sensibles de modo que la encriptación solo se realice a los campos que el analista seleccione.

### **1.3. Objeto de estudio**

Por ser el sistema propuesto una solución basada en un producto específico existente en el mercado (el middleware SIX/TCL de Novatronic), el objeto de estudio es la empresa Novatronic S.A.C.

Novatronic, empresa peruana fundada en 1987, está dedicada a proveer servicios de desarrollo de aplicaciones, de integración de sistemas, de gerencia de proyectos y de consultoría.

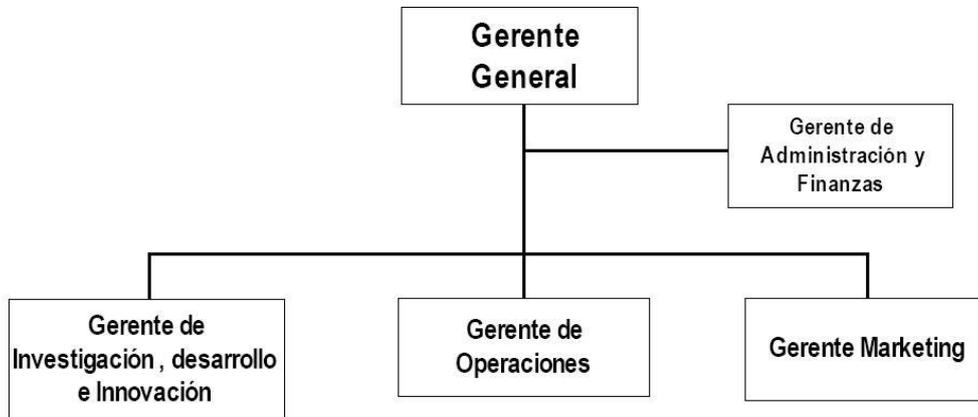
Las soluciones que provee Novatronic generalmente se enmarcan en el desarrollo de aplicaciones bajo arquitectura Cliente/Servidor, de sistemas distribuidos y de internet, utilizando tecnología de bases de datos, de seguridad, de comunicaciones y de ambiente visual.

Novatronic está conformada principalmente por una gerencia general y tres gerencias centrales: gerencia de investigación, desarrollo e innovación, gerencia de marketing y gerencia de operaciones.

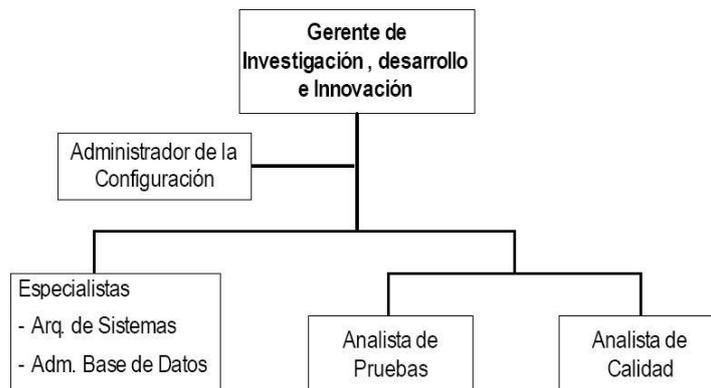
La gerencia general y las tres gerencias centrales conforman la “Alta Dirección” y a su vez integran el “Comité de Gerencia” de la empresa.

La estructura anteriormente mencionada se puede ver reflejada en el organigrama de la empresa el cual es el siguiente <sup>4</sup>

**Organigrama general**

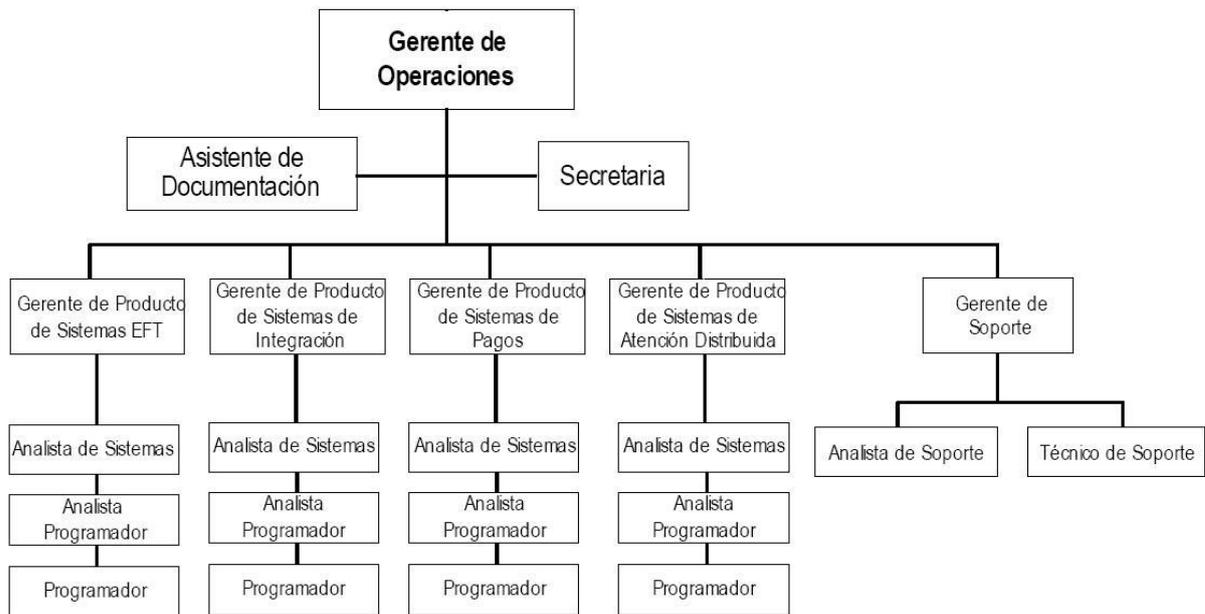


**Gerencia de investigación, desarrollo e innovación**



<sup>4</sup> Organigrama oficial proporcionado por Novatronic S.A.C.

**Gerencia de operaciones**



Cabe destacar que Novatronic cuenta con una organización matricial acorde con las nuevas formas de organización, por lo que cada gerencia posee responsabilidades concretas sobre su área y a su vez se cuenta con una organización por proyecto cuyos responsables tienen autonomía en todas las áreas de la organización.

**1.4. Campo de acción**

El campo de acción del presente trabajo está constituido por el proceso desarrollo de proyectos de la empresa Novatronic. Para la ejecución de un proyecto, Novatronic designa un equipo de trabajo, el cual está conformado por personal de diversas especialidades y conocimientos quienes, en base a la metodología, llevan a cabo las tareas necesarias para ejecutar el proyecto.

El equipo de trabajo está conformado por analistas programadores, quienes son dirigidos por un gerente de proyecto el cual debe administrar y ejecutar el plan de trabajo en forma integrada y es responsable frente al gerente de producto.

Como ya se mencionó, Novatronic utiliza un esquema matricial para la ejecución de proyectos, es decir se tiene una organización funcional dentro de la empresa y de acuerdo a los proyectos se define una organización temporal basada en roles, que se detallan a continuación:

Roles en la organización de los proyectos:

- Comité de proyectos
- Gerente de operaciones
- Gerente de producto
- Gerente de proyecto
- Equipo de trabajo
  - ✓ Analista programador
  - ✓ Analista de sistemas
  - ✓ Administrador de datos del proyecto
- Equipo de control
  - ✓ Administrador de la configuración
  - ✓ Analista de calidad

El ciclo de vida del proyecto está conformado por procesos, que se encuentran subdivididos por tareas con la finalidad de permitir que éstos sean controlados. Para ello se

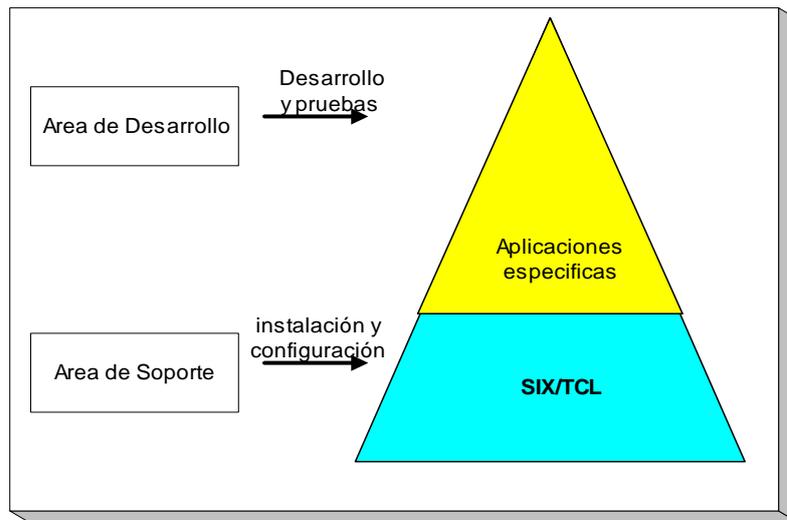
realizan evaluaciones del cumplimiento de los objetivos trazados. A continuación se describe de manera general la realización del ciclo de vida.

Nro.	NOMBRE DEL PROCESO	DESCRIPCIÓN DEL PROCESO	G E S T I Ó N  D E  P R O Y E C T O S
01	INICIACIÓN	<ul style="list-style-type: none"> <li>• Elaboración del Plan de Trabajo</li> <li>• Planificación del Proyecto</li> <li>• Determinación de los objetivos y alcance</li> <li>• Análisis de los Requerimientos</li> </ul>	
02	ANÁLISIS Y DISEÑO	<ul style="list-style-type: none"> <li>• Elaboración del Análisis y Diseño</li> <li>• Presentación y aprobación de Análisis y Diseño</li> </ul>	
03	PROGRAMACIÓN E INTEGRACIÓN	<ul style="list-style-type: none"> <li>• Programación y pruebas de módulos</li> <li>• Elaboración de manuales</li> <li>• Elaboración de documentos de pruebas</li> <li>• Ejecución de pruebas</li> </ul>	
04	CERTIFICACIÓN	<ul style="list-style-type: none"> <li>• Elaboración del documento de Pruebas de Certificación</li> <li>• Ejecución de las Pruebas de Certificación</li> </ul>	
05	IMPLEMENTACIÓN	<ul style="list-style-type: none"> <li>• Instalación</li> <li>• Capacitación</li> <li>• Documentación</li> <li>• Puesta en producción</li> <li>• Pase a soporte</li> </ul>	
06	CIERRE	<ul style="list-style-type: none"> <li>• Evaluación del Personal</li> <li>• Informe de Cierre del Proyecto</li> <li>• Evaluación de Proveedores</li> </ul>	
07	GESTIÓN DE PROYECTOS	<ul style="list-style-type: none"> <li>• Elaboración del Informe de Seguimiento (*)</li> <li>• Elaboración de la Matriz de Trazabilidad (*)</li> <li>• Elaboración del documento de Riesgos (*)</li> <li>• Actualización del Cronograma</li> </ul>	

Donde:  
(\*) : En caso aplique.

Típicamente, un proyecto desarrollado por Novatronic se divide en dos tareas principales:

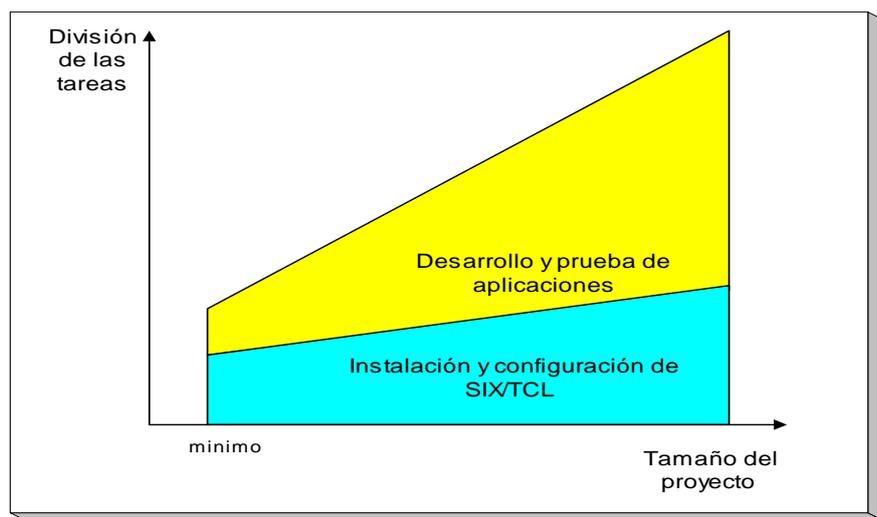
- El uso del SIX/TCL como base de comunicación
- El desarrollo de aplicaciones específicas que den encuentro a los requerimientos del proyecto y utilicen el SIX/TCL como medio de comunicación, como se muestra en la Ilustración 4.



**Ilustración 4: Componentes de una solución de integración de Novatronic**

La instalación y configuración de SIX/TCL es un proceso estándar cuyo tiempo de ejecución y costo en el proyecto depende del número y características de las plataformas de hardware y protocolos de comunicación involucrados en el proyecto. Este costo no varía substancialmente. El esfuerzo de las tareas de desarrollo de aplicaciones sí es directamente proporcional al tamaño del proyecto a realizar y las funcionalidades que se requiera implementar <sup>5</sup>. Como se observa en la Ilustración 5, la etapa de desarrollo y pruebas de aplicaciones consume una parte mayor de los recursos asignados al proyecto conforme aumenta el tamaño de este.

<sup>5</sup> Basado en la experiencia de proyectos en Novatronic



**Ilustración 5: Implicancias del tamaño del proyecto en la solución.**

## **1.5. Análisis crítico de los problemas de la organización**

### **1.5.1. Situación problemática**

Para comprender mejor la situación problemática actual, se describirán tres escenarios de estudio comunes que se presentan en los proyectos de integración de sistemas de Novatronic.

#### **Escenario 1: Clientes solicitando requerimientos de un servidor de negocios**

En este escenario se tiene un programa cliente que puede estar instalado en una o varias estaciones, y que debe intercambiar información con un servidor central (Ver Ilustración 6). Para que la estación cliente pueda intercambiar información con el servidor central de negocios, lo que se hace es colocar un middleware de comunicación entre ellos. La interacción entre el middleware y los programas cliente y servidor se realiza mediante el uso de las funciones que provee el middleware de comunicación, por ello la estación cliente debe incorporar dentro del código del programa las llamadas a dichas funciones con el fin de poder establecer una sesión con el middleware, enviarle la información de

requerimiento, recibir la confirmación de la recepción de la información y cerrar la sesión. Por otro lado, el programa servidor aplicativo también debe incorporar las llamadas a las funciones provistas por el middleware con el fin de recibir, procesar y retornar la respuesta a los requerimientos del cliente. Además, tanto el programa cliente como el programa servidor deben contemplar los códigos de retorno que devuelve el middleware para tomar las acciones respectivas en caso de producirse un error al momento de conectarse, enviar o recibir los mensajes o desconectarse.

Por lo tanto, en este escenario las tareas que el analista programador debe codificar son:

- Establecer la sesión con el middleware de comunicación y así mismo cerrar dicha sesión cuando no se requiera enviar la información.
- Generar la trama<sup>6</sup> de datos que se enviará al servidor de negocios.
- Descomponer la trama enviada por el cliente y actuar de acuerdo a las especificaciones solicitadas.
- Validar que el intercambio de información se ha realizado en forma satisfactoria.
- Modificar el programa cliente y servidor en caso que las reglas de negocio de la empresa cambien.

---

<sup>6</sup> Secuencia de caracteres que obedece a una especificación definida

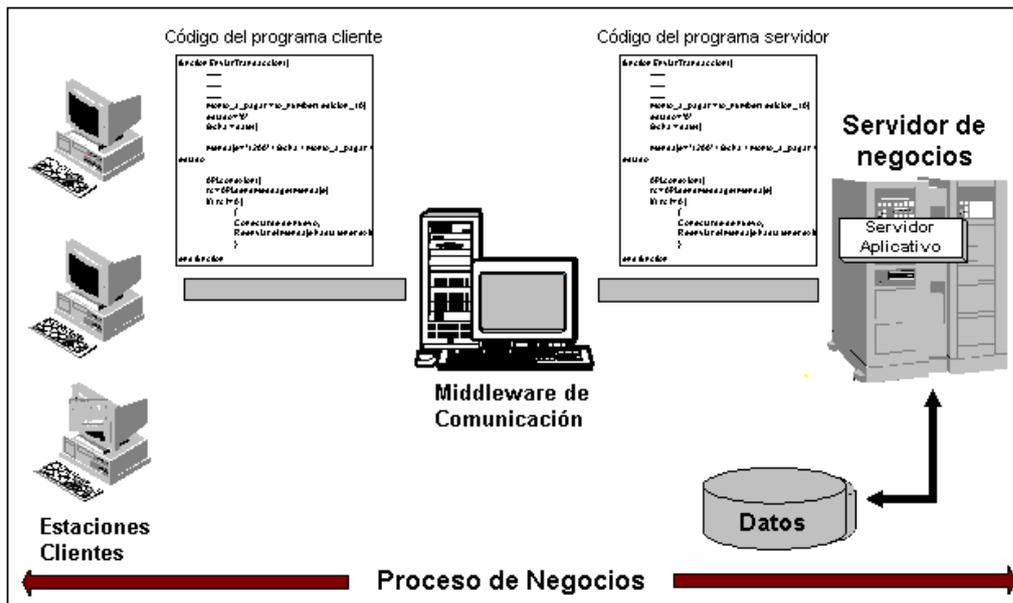


Ilustración 6: Integración de estaciones cliente

### Escenario 2: Múltiples servidores de atención

En este escenario, la aplicación cliente necesita intercambiar información con dos (2) o más servidores. En la Ilustración 7 se observa que para que las estaciones clientes puedan enviar la información hacia un determinado servidor lo que se hace es colocar dentro del middleware de comunicación un programa servidor aplicativo que tenga la inteligencia para enrutar los mensajes que provienen de las estaciones clientes.

Por lo tanto, en este escenario las tareas que el analista programador debe codificar son:

- Definir las reglas de enrutamiento de mensajes hacia un determinado servidor dependiendo de las características de la información enviada por el cliente.
- Transformar la información que proviene de las estaciones clientes y ajustarla de acuerdo con las características de cada uno de los servidores.

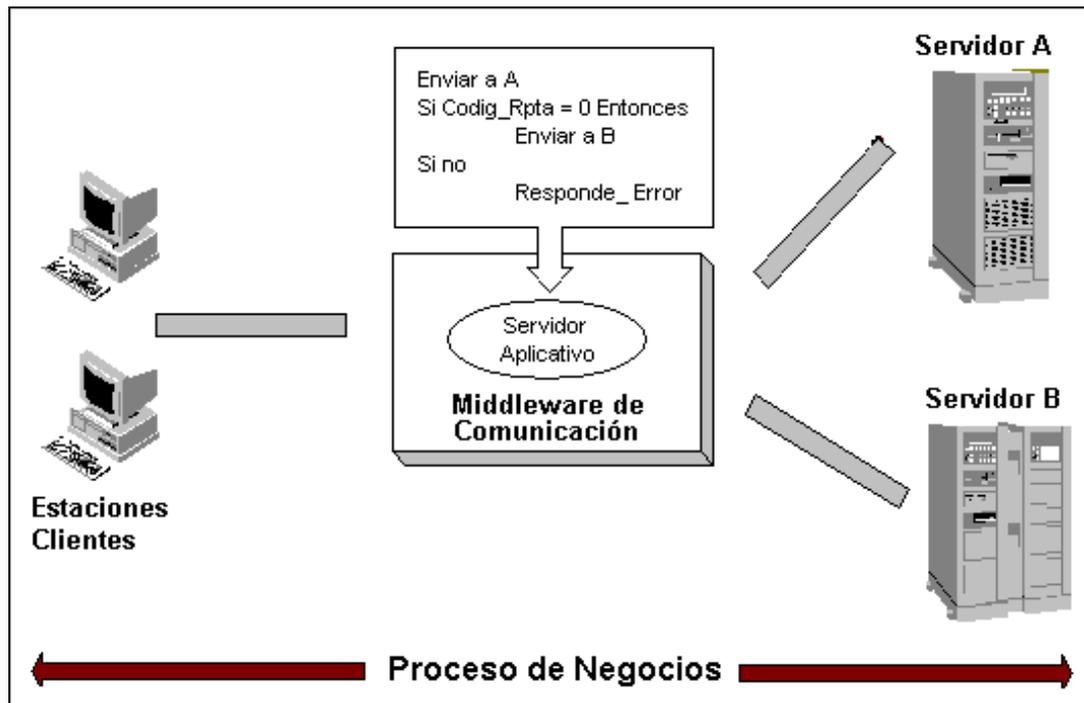


Ilustración 7: Conexión a múltiples servidores

### Escenario 3: Interacción con sistemas externos

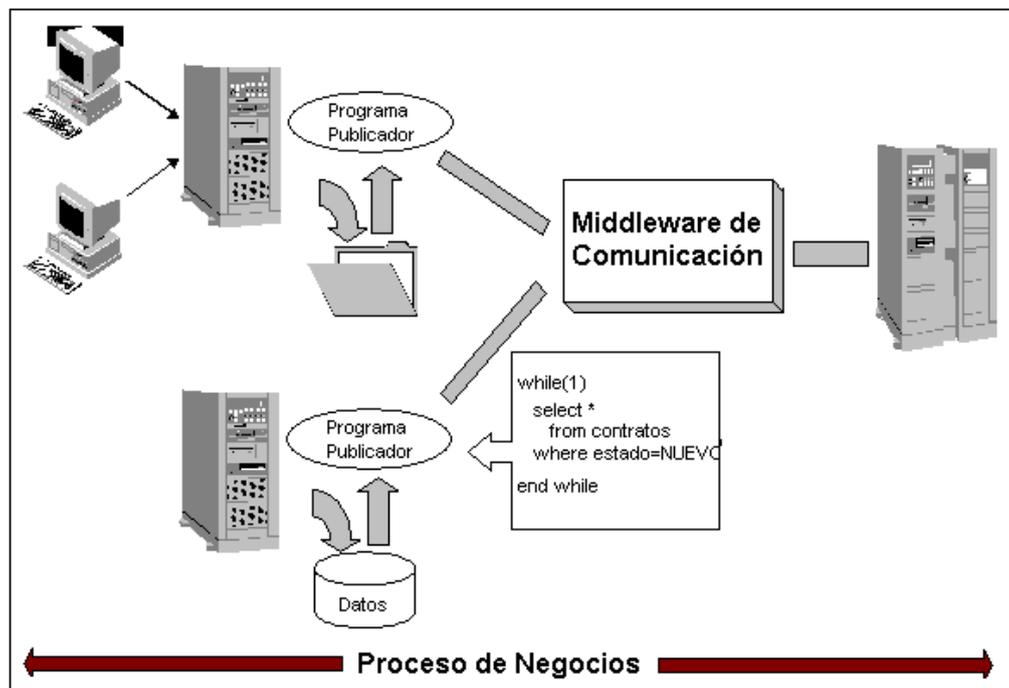
En el tercer escenario tenemos que los sistemas existentes en la organización no pueden ser modificados para añadir la funcionalidad de conexión al middleware por alguna de las siguientes razones:

- Porque no se poseen los códigos fuentes de dichos sistemas
- Porque son sistemas propietarios que usan protocolos particulares para la comunicación con otros sistemas (por ejemplo un sistema SAP, PeopleSoft, etc.),
- Porque la empresa no desea incrementar el riesgo de fallas en aplicaciones de producción.

En este escenario (ver Ilustración 8) se desarrollan programas publicadores que brindan la conectividad con el middleware. Estos programas manejan por un lado la conexión con el middleware de comunicación y por el otro lado están esperando permanentemente la activación de un evento que puede ser la generación de un nuevo archivo en un directorio o

la inserción de un nuevo registro en una base de datos determinada. Cuando este evento se activa, el programa publicador captura la información y lo envía al servidor destino mediante el uso de las funciones que provee el middleware para su proceso en el servidor de negocios.

Por lo tanto, el analista programador debe desarrollar un programa publicador específico dependiendo de la necesidad de cada proyecto.



**Ilustración 8: Interacción con sistemas externos**

En resumen, todo proyecto de integración implementado con SIX/TCL requiere un equipo de desarrolladores que escriba código para cubrir las tareas descritas en cada uno de los escenarios definidos con las siguientes consecuencias:

- Actividades repetitivas de desarrollo por cada proyecto de integración

- Tiempos de desarrollo de aplicaciones elevados (independientemente del tiempo que se ahorra por utilizar un middleware de comunicación)
- Limita la capacidad de las empresas para adaptarse en forma rápida a los cambios en las reglas de negocios o para implementar nuevos servicios.
- Mayores tiempos de prueba, depuración y certificación debido a que se desarrollan soluciones específicas para cada proyecto de integración.

Conforme los proyectos son más complejos en sistemas involucrados o requerimientos de intercambio de información, el desarrollo del código consume mayor tiempo ocasionando el encarecimiento del proyecto.

### **1.5.2. Problema a resolver**

La causa de la situación problemática descrita en el punto anterior es que los programadores escriben código para cada proyecto que desarrolla la empresa y no existe una herramienta que automatice el desarrollo del software.

### **1.6. Conclusiones**

La solución convencional para integrar sistemas es desarrollar programas específicos de acuerdo a las necesidades del proyecto, sin embargo, esto trae algunos inconvenientes como por ejemplo: los tiempos de desarrollo, la capacidad de las empresas de adaptarse a los cambios constantes de las reglas de negocio y el consumo de mayor tiempo en las pruebas y certificación de las aplicaciones desarrolladas.

## **CAPÍTULO II**

### **PROPUESTA DE SOLUCIÓN**

#### **2.1. Introducción**

El presente capítulo se centra en la propuesta de solución. Plantearemos los objetivos del proyecto con sus respectivos fundamentos. Finalmente se indican los beneficios que traerá el desarrollo del proyecto.

#### **2.2. Objetivos del proyecto**

##### **2.2.1. Objetivo general**

Conforme se ha analizado en la problemática a resolver, el objetivo fundamental del presente trabajo es construir una solución que permita obtener la información oportuna, moverla dentro de la empresa y utilizarla en forma adecuada permitiendo a la empresa definir las reglas de cómo mover la información, a dónde enviarla y cómo debe ser entregada. Conforme con lo analizado en el modelo de referencia esta solución se desarrollará bajo el paradigma de procesos de negocios. Esta solución debe permitir a la empresa reducir sus tiempos de desarrollo e implementación de proyecto de integración de sistemas.

### **2.2.2. Objetivos específicos**

Para ello, será necesario cumplir los siguientes objetivos específicos:

- ❖ **Diseñar un modelo general de datos para modelar los procesos de negocio** que se adapte a las reglas de negocios involucradas en los proyectos de integración de sistemas de la organización.
  
- ❖ **Construir un módulo cuya función será interpretar las reglas de negocio definidas** de modo que el sistema resuelva cada requerimiento que se reciba de una aplicación de acuerdo con las reglas definidas.
  
- ❖ **Garantizar la seguridad de los procesos de negocios almacenados en el repositorio** mediante el manejo de roles de usuarios que definirá quienes pueden crear/modificar un proceso de negocio y quienes pueden controlar el estado del sistema.

### **2.2.3. Fundamentación de los objetivos propuestos**

El cumplimiento de los objetivos nos permitirá:

- Eliminar las actividades repetitivas de desarrollo por cada proyecto de integración
- Reducir el tiempo de desarrollo de aplicaciones
- Incrementar la capacidad de las empresas para adaptarse en forma rápida a los cambios en las reglas de negocios o para implementar nuevos servicios.
- Reducir los tiempos de prueba, depuración y certificación.

#### **2.2.4. Indicadores de logro de los objetivos**

Los indicadores de logros de objetivos son:

- El documento de diseño del sistema
- El prototipo del sistema en su primera versión
- El acta de cierre del proyecto aprobada

### **2.3. Beneficios del proyecto**

#### **2.3.1. Beneficios tangibles**

Dadas las características particulares de la solución que se propone, es difícil estimar una métrica cuantitativa para evaluar el sistema en cuanto a su objetivo de reducción de tiempos de desarrollo de proyectos. El porcentaje de reducción de tiempos de desarrollo y/o adecuación que se pueden obtener con este sistema (o sistemas de este tipo) está relacionado con las características particulares del proyecto de integración. Es por ello que los fabricantes de soluciones ya existentes en el mercado no ofrecen en la información que brindan sobre sus productos mediciones cuantitativas de sus sistemas.

Teniendo esta consideración, los beneficios tangibles que el sistema brindará son:

- ❖ Reducir el esfuerzo de codificación de aplicaciones para atender los requerimientos de integración de la empresa.
- ❖ Reducir el tiempo de puesta en producción de los proyectos de integración de sistemas.
- ❖ Independencia del proveedor del sistema, lo que se traduce en menores costos de mantenimiento. El personal de la propia empresa podrá realizar los cambios que se requieran en sus procesos de negocios.

- ❖ Menores costos de desarrollo de los proyectos, por la reducción en el uso de recursos de la empresa y del tiempo de desarrollo de los proyectos. Este ahorro puede ser trasladado al cliente dándole a la empresa una ventaja económica u otorgarle a la empresa una mayor rentabilidad en los proyectos.

### **2.3.2. Beneficios intangibles**

Entre los beneficios intangibles tenemos:

- ❖ Proveer una vista gráfica a nivel de negocios de los flujos de los procesos de negocios de una empresa.
- ❖ Mejorar la competitividad de la empresa al brindar la flexibilidad para realizar las modificaciones y adopciones de los procesos de negocio.
- ❖ Automatizar la interacción entre los sistemas de una empresa y sistemas exteriores a ella como sus proveedores.
- ❖ Mejorar el posicionamiento de Novatronic como empresa especializada en integración de sistemas, al poder ofrecer menores tiempos de desarrollo de proyectos a los clientes y flexibilidad en el mantenimiento de las soluciones.

### **2.4. Conclusiones**

El proyecto propone un aporte en el desarrollo de productos de tecnologías de integración corporativa, diseñando y desarrollando un ambiente de mensajería inteligente para ayudar a las organizaciones a optimizar el valor total de sus sistemas de información, proveer un marco de trabajo coherente para manejar el comportamiento de los datos y servicios entre aplicaciones y recursos de información de todo tipo para dar encuentro a los requerimientos de negocios que se encuentran en constante cambio.

## **CAPÍTULO III**

### **MODELADO DEL NEGOCIO**

#### **3.1. Introducción**

El presente capítulo describe el modelado del negocio y se centra en el desarrollo de los casos de uso del negocio, en las reglas del negocio y en el modelo de análisis del negocio.

#### **3.2. Modelo de casos de uso del negocio**

##### **3.2.1. Lista de actores del negocio**

- **Comité de proyecto**

El comité de proyecto representa la escala más alta de la institución y los miembros que la componen están definidos en el documento “Descripción de los Comités Novatronic”. El comité de proyecto aprueba el inicio de la ejecución de un proyecto de acuerdo a las necesidades y metas de la organización y a los requerimientos de los clientes. Del mismo modo aprueba el cierre de un proyecto.

- **Gte Operaciones**

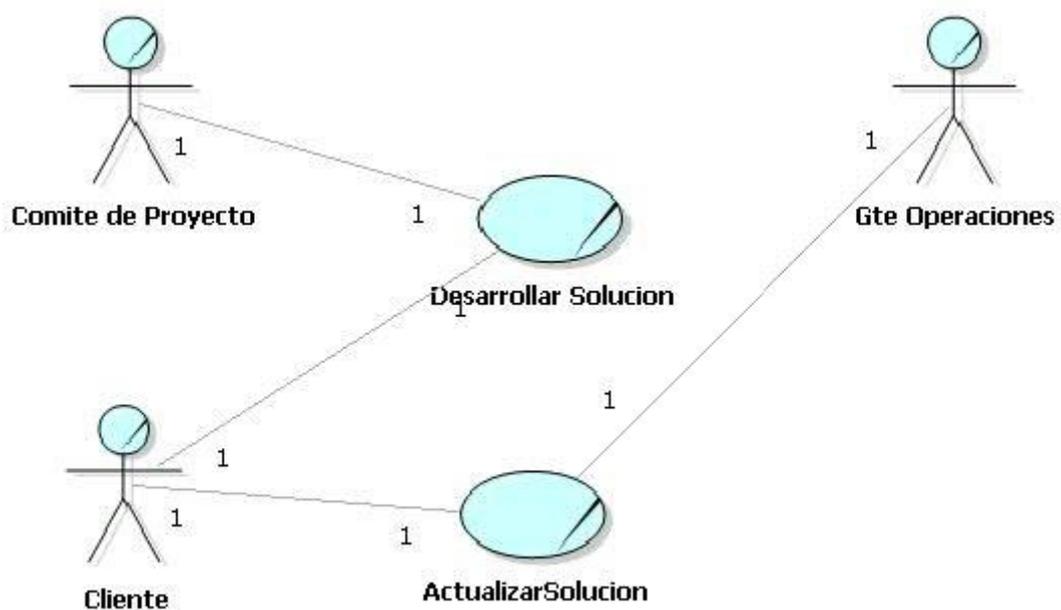
El gerente de operaciones aprueba las solicitudes de actualización o mantenimiento de los clientes y asegura una administración eficiente y productiva de los recursos de la empresa y sus operaciones en la ejecución de proyectos y de soporte técnico a los clientes y usuarios internos de la empresa.

### 3. Cliente

El cliente es el ente que solicita un servicio (que puede ser el desarrollo de un proyecto o la actualización de un producto previamente instalado) a Novatronic.

#### 3.2.2. Diagrama de casos de uso del negocio

El diagrama de casos de uso del negocio definido es el siguiente:



### 3.3. Reglas del negocio

Las reglas definidas por el negocio son:

- **Administración de la configuración**

Los programas fuentes generados durante el desarrollo del proyecto deberán ser entregados al administrador de la configuración antes de implementarse en el cliente.

- **Instalación en el cliente**

Para instalar la versión final de la aplicación en el cliente se requiere la aprobación del gerente de proyecto.

### **3.4. Realización de los casos de uso del negocio**

#### **3.4.1. Especificación de los casos de uso del negocio**

A continuación se detalla los casos de uso del negocio:

#### **A Especificación del caso de uso del negocio: Desarrollar solución**

##### **1. Actores**

Los actores que intervienen en el caso de uso son:

- Comité de proyecto
- Cliente

##### **2. Propósito**

Establecer los lineamientos y actividades a seguir para el desarrollo de las soluciones a los proyectos implementados por la organización.

##### **3. Breve descripción**

El caso de uso comienza cuando el comité de proyecto autoriza la ejecución de un proyecto y designa a sus responsables.

El equipo de proyecto designado procede con el desarrollo del proyecto siguiendo la metodología de la organización.

El caso de uso termina cuando el comité de proyecto aprueba el cierre del proyecto.

#### **4. Flujo básico de eventos**

- 4.1 El comité de proyecto autoriza la ejecución de un proyecto y designa a sus responsables (gerente de proyecto, equipo de trabajo, etc)
- 4.2 El gerente de proyecto revisa la propuesta técnica.
- 4.3 El gerente del proyecto de acuerdo a lo analizado en las propuestas solicita los recursos necesarios para ejecutar el proyecto.
- 4.4 Definidas las actividades, los tiempos y los recursos, el gerente del proyecto procede a elaborar el plan de trabajo y el documento de Planificación del Proyecto. Cuando el documento de planificación sea aprobado, el gerente de proyecto carga el cronograma del proyecto en el Sistema SARA para distribuir las actividades a cada uno de los responsables y recursos del proyecto.
- 4.5 El gerente del proyecto convoca a la reunión de Kick Off, con el equipo de proyecto
- 4.6 El gerente de proyecto elabora las alternativas de solución y luego se reúne con el comité técnico para seleccionar una de ellas
- 4.7 El equipo de trabajo valida los requerimientos de hardware, software y recursos de operación que han sido definidos en la propuesta técnica.
- 4.8 El gerente de proyecto elabora los documentos de interfaz applicativa y análisis y diseño
- 4.9 El gerente de proyecto presenta el documento de análisis y diseño al comité técnico y luego elaborará un acta de reunión de comité técnico
- 4.10 El gerente de proyecto, se reunirá con el equipo de trabajo con la finalidad de asignar las responsabilidades de programación.
- 4.11 Cada miembro del equipo de trabajo realiza la programación y pruebas unitarias de los módulos asignados
- 4.12 El equipo de trabajo y/o gerente de proyecto elabora el documento especificación de pruebas que permitirá validar que el producto cumpla con lo requerido.

4.13 Concluida las pruebas, el equipo de trabajo preparará el documento de “Resultado de Pruebas”.

4.14 El gerente de proyecto entregará al responsable de administración de la configuración lo siguiente:

- Software (Archivo en formato propietario)
- Simuladores (fuentes y ejecutables).
- Datos de prueba.
- Entorno de prueba.
- Otros

4.15 El gerente de proyecto elabora el documento de implantación del proyecto y solicita al Cliente la disponibilidad de requerimientos

4.16 El equipo de trabajo solicita a administración de la configuración los ítems de configuración necesarios para realizar la instalación.

4.17 El equipo de trabajo realizará la instalación, además verificará y probará que el sistema quede operativo en el cliente. Realizada las pruebas, el cliente realizará la certificación del sistema

4.18 El equipo de trabajo prepara la documentación requerida y realiza la capacitación. Al final de dicha capacitación entrega a los asistentes del curso el Formulario de Evaluación de la Capacitación, con la finalidad de que brinden su apreciación sobre la capacitación realizada

4.19 Terminada la certificación, el gerente de proyecto solicita al cliente la aceptación del sistema, asimismo coordina la fecha de pase a producción.

4.20 El equipo de trabajo conjuntamente con el cliente realizan el pase a producción del sistema

4.21 El gerente de proyecto coordina con el área de Soporte para hacer la capacitación de los productos instalados y/o actualizados del proyecto, a fin de que brinden el soporte adecuado cuando el cliente lo requiera o la situación lo amerite.

4.22 El gerente de proyecto evalúa a cada recurso del proyecto y registra dicha evaluación

4.23 El gerente de proyecto elabora un informe de cierre de proyecto. En él registra los resultados de cada etapa del proceso productivo así como las recomendaciones de mejora del proceso, y las lecciones aprendidas.

4.24 El gerente de proyecto verifica que todas las actividades relacionadas al proyecto estén cerradas en el SARA.

4.25 El gerente de proyecto comunica al comité de proyecto y a las áreas involucradas el cierre del proyecto para que tomen las acciones correspondientes.

4.26 El comité de proyecto revisa el informe de cierre y aprueba el cierre del proyecto

## **5. Subflujo**

No aplica

## **6. Flujos alternos**

- En 4.2, de ser necesario, el gerente de proyecto realizará reuniones con el Cliente para acordar y/o determinar las actividades principales y los tiempos esperados de la realización del proyecto
- En 4.9, Si el comité técnico presenta observaciones, el gerente de proyecto tendrá la responsabilidad de realizar el seguimiento del acta de comité con la finalidad que las observaciones hayan sido atendidas.
- En 4.13 si existen errores se debe volver a ejecutar las pruebas. El resultado de estas nuevas pruebas generará una nueva versión del documento de resultado de pruebas.

- En 4.17 si el Cliente reporta observaciones, el gerente de proyecto procede a revisarlos y determina:
  - ✓ Si son las fallas del software se procede a registrar una tarea de corrección de errores dentro del cronograma del proyecto.
  - ✓ Si son nuevos requerimientos, se informará y solicitará la autorización respectiva.

## **7. Precondiciones**

El Cliente debe haber aprobado la propuesta técnica enviada por Novatronic.

## **8. Poscondiciones**

La solución debe estar operativa en el cliente en ambiente de producción

## **9. Información adicional**

No aplica

## **B Especificación del caso de uso del negocio: Actualizar solución**

### **1. Actores**

Los actores que intervienen en el caso de uso son:

- Gerente de Operaciones
- Cliente

### **2. Propósito**

Establecer los lineamientos y actividades a seguir para atender los cambios o problemas en la solución instalada en el Cliente.

### **3. Breve descripción**

El caso de uso comienza cuando el cliente llama a Novatronic solicitando el mantenimiento de una solución instalada. El analista de soporte designado procede con el desarrollo de los ajustes, realiza las pruebas y lo instala en el Cliente.

El caso de uso termina cuando el analista de soporte cierra el requerimiento.

#### **4. Flujo básico de eventos**

4.1 El Cliente solicita la modificación de la solución instalada.

4.2 El área de marketing registra la atención de requerimiento y adjunta la propuesta técnica en el sistema SARA.

4.3 El gerente de soporte autoriza la ejecución de la atención de requerimiento

4.4 El analista de soporte realizará las actividades de acuerdo a la propuesta técnica indicada.

4.5 El analista de soporte realizará las pruebas de los ajustes teniendo como referencia el documento de especificación de pruebas y en caso sea necesario se deberá modificar los documentos de especificación y resultado de pruebas para incluir las pruebas adicionales que nos permitan verificar los ajustes y/o modificaciones realizadas..

4.6 El analista de soporte ajustará y/o modificará los documentos del producto que sean necesarios, entre estos se pueden considerar los siguientes:

- Documento de especificación de pruebas
- Documento de resultado de pruebas
- Documento de interfaz applicativa
- Documento de instalación
- Manual de usuario

Los documentos a modificar deberán de ser obtenidos de la administración de la configuración

4.7 Terminadas las pruebas de los ajustes o modificaciones realizadas, el analista de soporte prepara el documento de actualización del producto y coordina con el cliente la actualización del producto (vía e-mail, teléfono).

4.8 El analista de soporte brinda soporte en las pruebas y pase a producción en el Cliente.

4.9 El analista de soporte prepara el informe de ajuste o modificación.

4.10 El analista de soporte envía un mail al cliente solicitando la aceptación para dar por cerrado el requerimiento

4.11 El Cliente envía la conformidad

4.12 El analista de soporte cierra el requerimiento y adjunta en el sistema SARA el informe de ajuste y la conformidad del cliente

## **5. SubFlujo**

No Aplica

## **6. Flujos alternos**

- En 4.4, de ser necesario, el analista de soporte realizará reuniones con el Cliente para acordar y/o determinar las actividades principales y los tiempos esperados de la realización del proyecto
- En 4.5 si existen errores se debe volver a ejecutar las pruebas. El resultado de estas nuevas pruebas generará una nueva versión del documento de resultado de pruebas.
- En 4.10 si el Cliente reporta observaciones, el analista de soporte procede a revisar y corregir

## **7. Precondiciones**

El Cliente debe tener un contrato de mantenimiento y asistencia técnica vigente.

## **8. Poscondiciones**

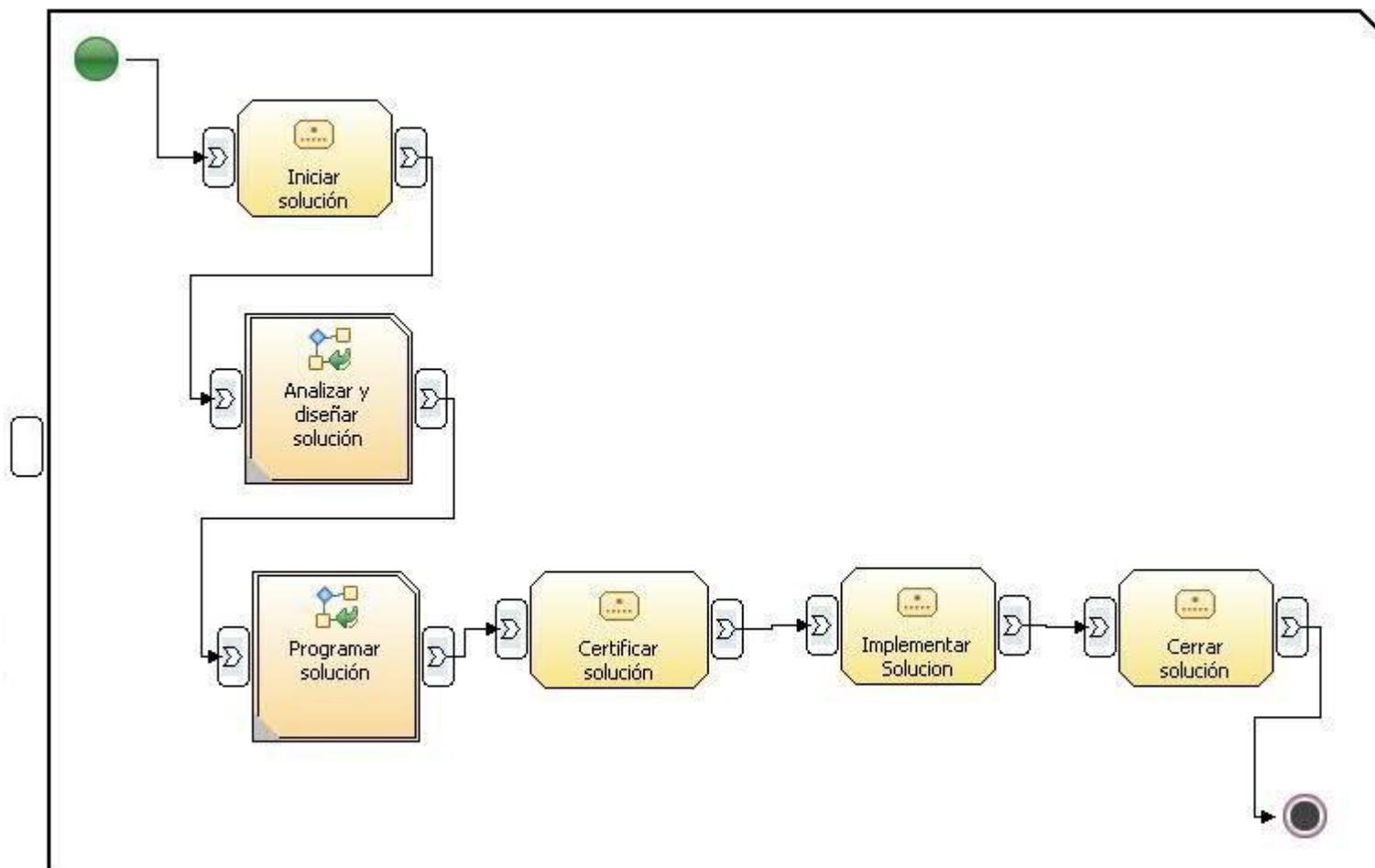
La solución modificada debe estar operativa en el cliente en ambiente de producción.

## **9. Información adicional**

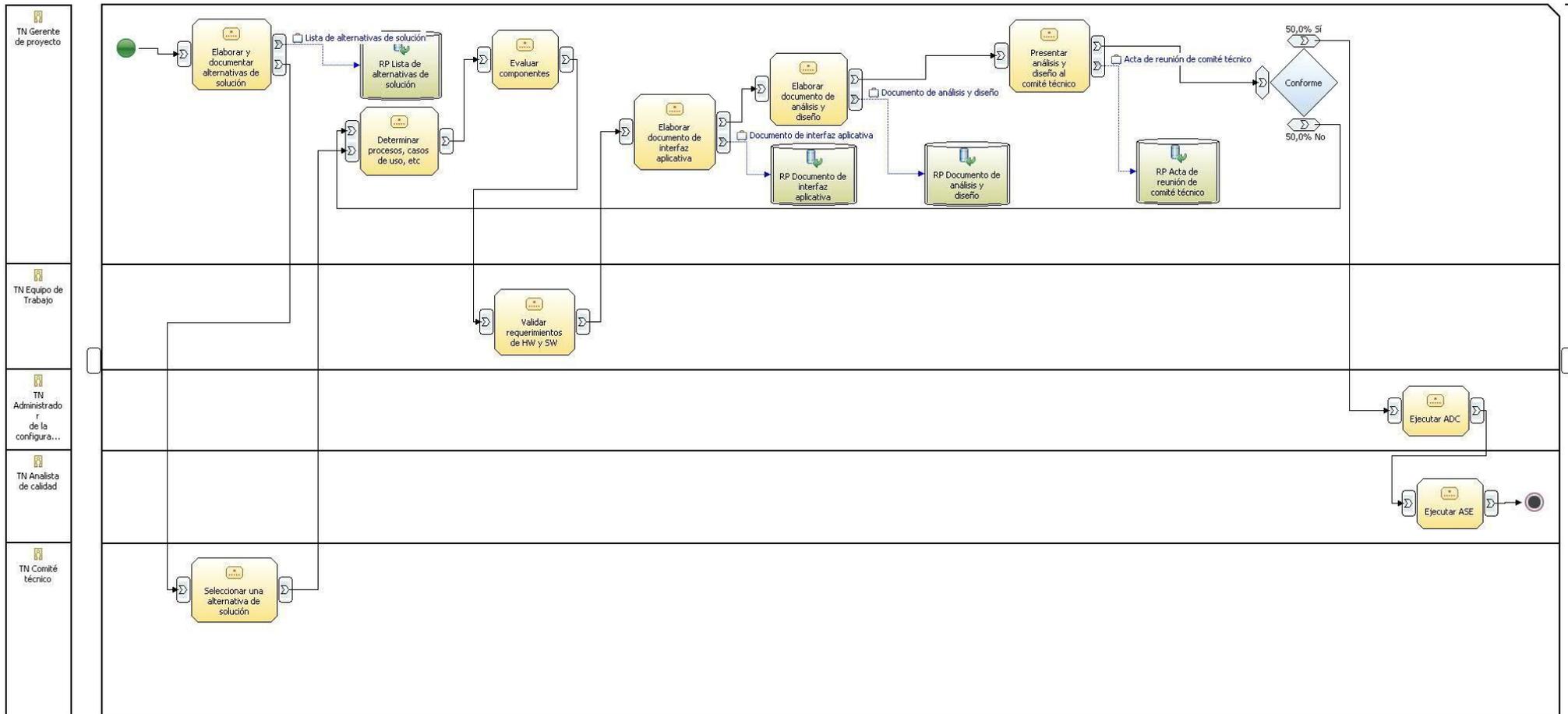
No Aplica

### 3.4.2. Diagramas de procesos

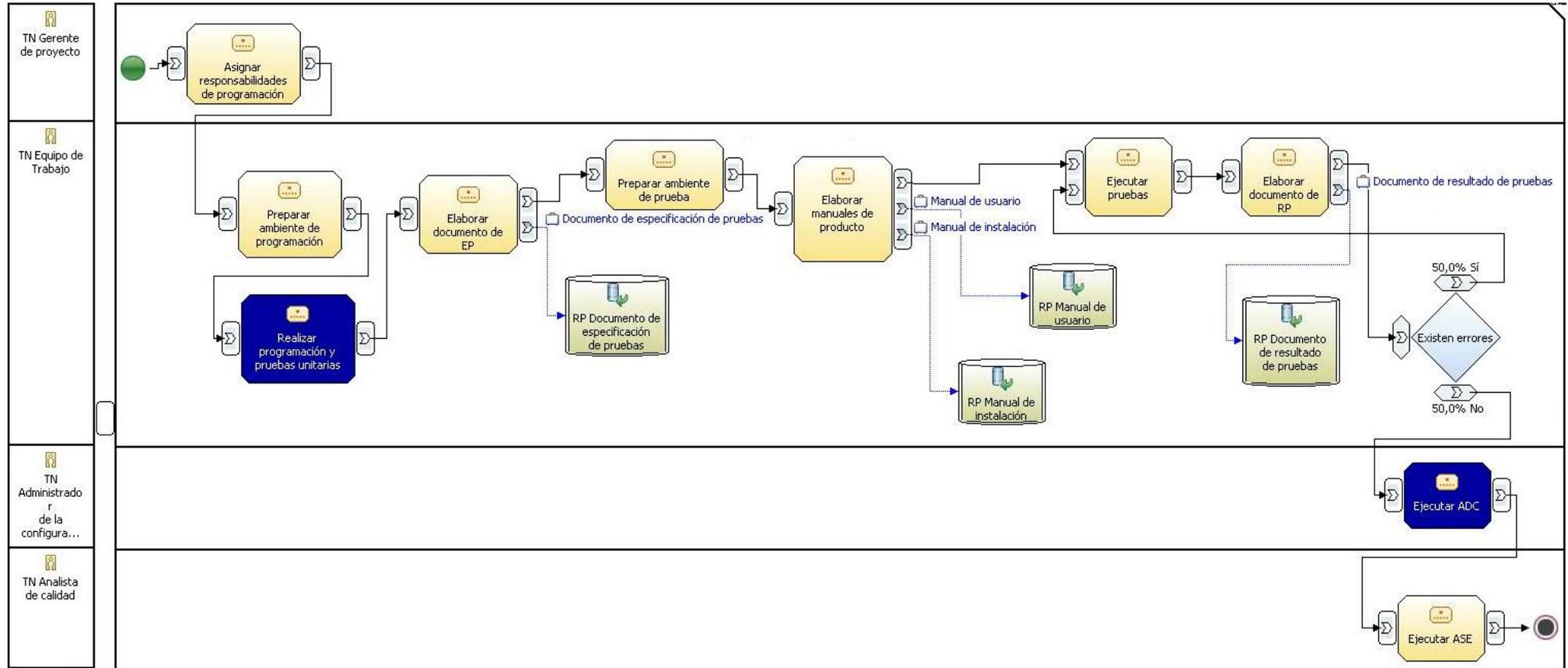
#### 1 Diagrama del proceso principal Desarrollar Solución



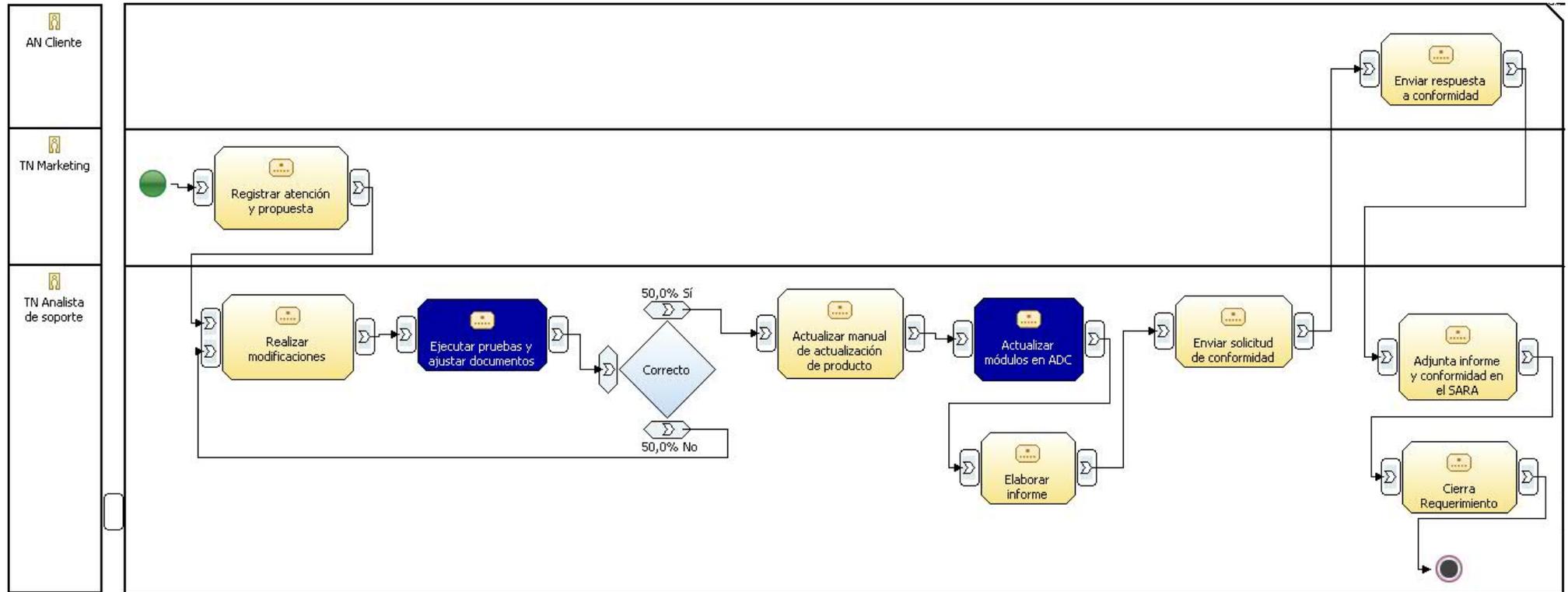
## 2 Diagrama del proceso Analizar y diseñar solución



### 3 Diagrama del proceso Programar solución



### 4 Diagrama del proceso Actualizar solución



### **3.4.3. Lista de actividades a automatizar**

Las actividades a automatizar se encuentran en el diagrama del proceso “Programar solución” y son:

- Realizar programación y pruebas unitarias
- Ejecutar ADC

## **3.5. Modelo de análisis del negocio**

### **3.5.1. Lista de trabajadores del negocio**

#### **1. TN Gerente de proyecto**

Tiene a su cargo las siguientes responsabilidades:

- Planea, dirige y controlar la ejecución del proyecto, velando que se cumpla con los objetivos establecidos.
- Gestiona y coordina los recursos adecuados para llevar a cabo las actividades del proyecto
- Hacer seguimiento de las actividades correspondientes a los procesos de iniciación, análisis y diseño, programación e integración, certificación, implementación y cierre de proyectos.
- Detecta e interpreta las posibles causas de problemas a fin de tomar a tiempo las decisiones que permitan corregir las desviaciones de los resultados esperados.
- Elabora y actualiza el plan de actividades del proyecto.
- Convoca y dirige las reuniones con los responsables de las tareas, así como participa en las reuniones de coordinación con el Cliente.
- Aprueba las pruebas ejecutadas por el equipo de trabajo designado.
- Coordina con el área de control para ejecutar las actividades de administración de la configuración y aseguramiento de calidad que corresponden al proyecto.
- Revisa y/o aprueba los documentos generados en los diferentes procesos del proyecto

## **2. TN Comité técnico**

Tiene a su cargo las siguientes responsabilidades:

- Evalúa y selecciona la alternativa de solución de un determinado proyecto presentado por el gerente de proyecto
- Evalúa el análisis y diseño del proyecto presentado por el gerente de proyecto y su equipo de trabajo

## **3. TN Equipo de trabajo**

Tiene a su cargo las siguientes responsabilidades:

- Participa en el planeamiento detallado de las tareas por las cuales va a adquirir responsabilidades.
- Establece y define la funcionalidad del sistema.
- Participa en el proceso de análisis y diseño.
- Llevar a cabo las actividades de evaluación y pruebas en caso de ser encomendadas.
- Elabora los documentos que se requieren en las diferentes etapas del proyecto.
- Reporta al gerente de proyecto los resultados obtenidos, el avance de sus tareas y solicita su aprobación respecto a productos entregables.

## **4. TN Administrador de la configuración**

Se encarga de administrar y controlar los cambios de los entregables en todos los procesos del ciclo de vida del proyecto

## **5. TN Analista de calidad**

Tiene a su cargo las siguientes responsabilidades:

- Ejecuta auditorias aleatorias a los diversos proyectos que se están ejecutando.

- Presenta los resultados de las auditorias de proyectos al gerente de producto, gerente de operaciones y comité de proyectos.
- Ejecuta las actividades de aseguramiento de la calidad.
- Presenta al comité de proyectos los resultados del cumplimiento de los procesos de aseguramiento de la calidad y administración de la configuración de los proyectos.

## **6. TN Marketing**

Tiene a su cargo elaborar las propuestas de ventas y los contratos de los clientes que se le asigne.

## **7. TN Analista de soporte**

Tiene a su cargo las siguientes responsabilidades:

- Atender al cliente en lo concerniente al mantenimiento correctivo y asistencia técnica de los productos brindados por la empresa y realizar los informes sobre el servicio brindado.
- Elaborar y actualizar la documentación generada por los cambios del producto y/o proyectos y capacitar a los usuarios, en el uso de los sistemas desarrollados en los casos que aplique.
- Realizar los mínimos cambios y/o nuevas funcionalidades que permitan la continuidad operativa del producto.

### **3.5.2. Lista de las entidades del negocio**

#### **1. EN Lista de alternativas de solución**

Documento donde se describen las alternativas a tomar para el desarrollo de un proyecto.

Nombre	Descripción	Tipo	Valor inicial
codigo	Código del documento	String	Vacio
fechaCreacion	Fecha de creación del documento	String	SystemDate
autor	Nombre de la persona que desarrolló el documento	String	Vacio
contenido	Contenido del documento	String	Vacio

### 3. EN DocumentoGenerico

Esta es una clase abstracta

Nombre	Descripción	Tipo	Valor inicial
codigo	Código del documento	String	Vacío
nombre	Nombre del documento	String	Vacío
versión	Versión del documento	String	“01.00”
fechaCreacion	Fecha de creación del documento	String	SystemDate
fechaRevision	Fecha de revisión del documento	String	Vacío
fechaAprobacion	Fecha de aprobación del documento	String	Vacío
autor	Nombre de la persona que desarrolló el documento	String	Vacío
revisor	Nombre de la persona que revisó el documento	String	Vacío
aprobador	Nombre de la persona que aprobó el documento	String	Vacío
titulo	Título del documento	String	“<Nombre del proyecto>”

### 4. EN Documento de análisis y diseño

Esta entidad es una especialización de la entidad DocumentoGenerico

En esta entidad se describe:

- La arquitectura del sistema a desarrollar
- El diseño global (procesos, pantallas, reportes, etc).

- Los requerimientos de operación (hardware, software, servicios)

Nombre	Descripción	Tipo	Valor inicial
analisis	Contiene el analisis de los requerimientos del cliente	String	Vacío
diseño	Contiene el diseño sugerido del sistema	String	Vacío

### 3. EN Documento de interfaz applicativa

Esta entidad es una especialización de la entidad DocumentoGenerico

En esta entidad se describe la estructura de los mensajes que se intercambian con los sistemas externos.

Nombre	Descripción	Tipo	Valor inicial
descripcionInterface	Contiene la especificación de la interfaz de mensajería	String	Vacío

### 4. EN Acta de reunión de comité técnico

Informe donde se definen los temas tratados en la exposición del análisis y diseño de la solución al comité técnico, así como las observaciones presentadas en dicha presentación

Nombre	Descripción	Tipo	Valor inicial
codigo	Código del documento	String	Vacío
fechaCreacion	Fecha de creación del documento	String	SystemDate
autor	Nombre de la persona que desarrolló el documento	String	Vacío
contenido	Contenido del documento	String	Vacío

### 5. EN Documento de especificación de pruebas

Esta entidad es una especialización de la entidad DocumentoGenerico

En esta entidad se definen las pruebas a realizar a la solución desarrollada

Nombre	Descripción	Tipo	Valor inicial
tipoPrueba	Tipo de prueba a realizar: funcional, de interface, de performance, entre otros	String	Vacío
prueba	Describe la prueba a realizar y las condiciones en que se realiza la prueba	String	Vacío

## 6. EN Documento de resultado de pruebas

Esta entidad es una especialización de la entidad DocumentoGenerico

En esta entidad se detallan los resultados de las pruebas definidas en el documento de especificación de pruebas.

Nombre	Descripción	Tipo	Valor inicial
cicloPrueba	Corresponde al numero de ciclo de prueba realizado	Integer	Vacío
tipoPrueba	Tipo de prueba a realizar: funcional, de interfaz, de performance, entre otros	String	Vacío
prueba	Describe la prueba a realizar y las condiciones en que se realiza la prueba	String	Vacío
resultado	Es el resultado de la prueba realizada	String	Vacío
ejecutadoPor	Nombre del trabajador que ejecuto la prueba	String	Vacío
fechaPrueba	Fecha de la prueba	String	Vacío

## 7. EN Manual de usuario

Esta entidad es una especialización de la entidad DocumentoGenerico

En esta entidad se describe cómo se utiliza el sistema desarrollado.

Nombre	Descripción	Tipo	Valor inicial
modulo	Contiene la explicacion funcional del modulo	String	Vacío

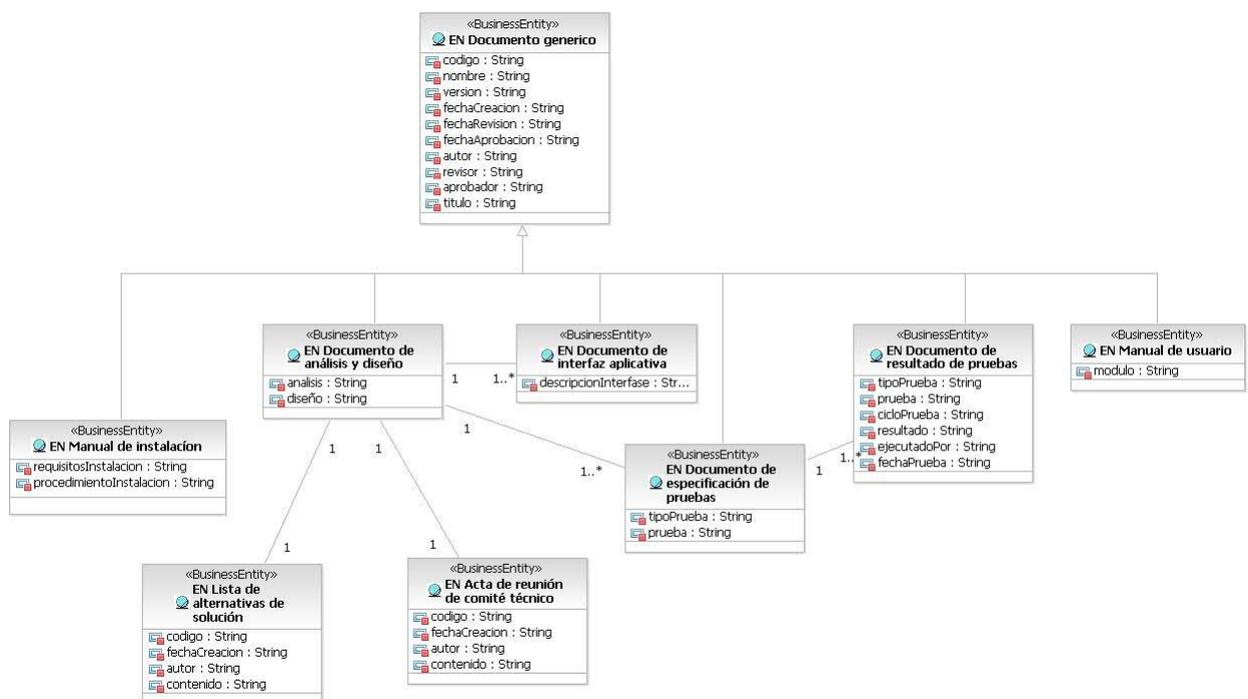
### 8. EN Manual de instalación

Esta entidad es una especialización de la entidad DocumentoGenerico

En esta entidad se describe cómo se instala y configura el sistema desarrollado

Nombre	Descripción	Tipo	Valor inicial
requisitoInstalacion	Define los requisitos de hardware, software, licenciamiento que se necesitan para la correcta instalación del sistema	String	Vacío
procedimientoInstalacion	Define los pasos de la instalación	String	Vacío

### 3.5.3. Diagrama de clases del negocio



### 3.6. Conclusiones

En este capítulo se ha realizado el estudio y análisis del negocio, describiendo los principales casos de uso y proceso. Se ha identificado los puntos del proceso que son sujeto de automatización por el sistema a desarrollar.

## **CAPÍTULO IV**

### **REQUERIMIENTOS**

#### **4.1. Introducción**

El presente capítulo se revisarán los requerimientos del sistema y el modelo del sistema.

#### **4.2. Especificación de los requerimientos del sistema**

Se han identificado los siguientes requerimientos del sistema:

##### **4.2.1. Funcionalidad**

1. Registrar los sistemas externos y el mecanismo de interacción
2. Registrar los eventos que conformarán los procesos de negocio
3. Modelar la información que se intercambia entre los sistemas y las transformaciones de esa información.
4. Modelar los procesos como una secuencia de tareas o pasos que deben ejecutarse.
5. Soportar flujos alternos en el modelamiento de procesos, o flujos condicionales
6. Consultar el modelo desarrollado para un proceso de negocio
7. Consultar referencias cruzadas entre Sistemas y Procesos de Negocio.
8. Consultar la relación de objetos definidos para un sistema, sus atributos y persistencia.

9. Extraer una imagen en tiempo real de los modelos de procesos registrados para administración de la configuración.
10. Extraer el modelo completo de un proceso de negocio
11. Detectar inconsistencias en el modelo de un proceso de negocio.
12. Generar la configuración de conectores para cada SISTEMA
13. Importar el modelo completo de un proceso de negocio
14. Detectar la condición de lanzamiento de un proceso de negocio.
15. Controlar cada paso del proceso negocio a ejecutar. En caso de errores, detener la ejecución del proceso.
16. Mantener un registro en tiempo real de la ejecución de un proceso de negocio. Para cada proceso de negocio en ejecución debe registrarse los tiempos consumidos en cada paso, los pasos ejecutados, el paso que se esta ejecutando.
17. Mantener un registro de los procesos de negocio ejecutados. Para cada ejecución realizada debe registrarse el resultado del proceso, la duración total del mismo, la duración de cada paso.
18. Habilitar / deshabilitar procesos de negocio
19. Detener o suspender temporalmente la ejecución de un proceso de negocio
20. Consultar las instancias de procesos de negocio ejecutadas.
21. Mantener un sistema de usuarios y perfiles de usuarios con poderes diferentes.
22. Obligar al usuario a identificarse con un login/contraseña para ingresar a los módulos.
23. Permitir la modificación de contraseña de un usuario.

#### **4.2.2. Usabilidad**

No se han definido requerimientos de usabilidad.

**4.2.3. Confiability**

1. El sistema debe estar disponible las 24 horas del día , los 7 días de la semana.

**4.2.4. Rendimiento**

1. El sistema debe soportar hasta un máximo de 50 procesos definidos.
2. El sistema debe soportar hasta un máximo de 1000 instancias por cada proceso definido
3. El tiempo de procesamiento directo del sistema en cada proceso de negocio no debe exceder los 2 segundos.

**4.2.5. Soporte**

1. El sistema debe almacenar información histórica de los procesos de hasta 6 meses de antigüedad.
2. El sistema podrá trabajar con las siguientes bases de datos:
  - Oracle
  - SQL Server
  - MySQL
  - Postgres

**4.2.6. Restricciones de diseño**

1. El módulo que interactúa con los sistemas externos se desarrollará en J2EE.
2. Todos los mensajes que se intercambien dentro del sistema en un flujo de procesos tendrán MAC.
3. El sistema debe soportar el cifrado de datos en los mensajes que se intercambien dentro de un flujo de procesos. En el diseño de un proceso de negocio, el analista programador podrá definir que información se cifra.

4. El sistema deberá generar correos para informar los errores ocurridos en la ejecución de un proceso

#### **4.2.7. Documentación de usuario y sistema de ayuda**

- 1 Se deberá proveer la documentación del sistema (manual de usuario y de instalación).

#### **4.2.8. Componentes adquiridos**

No aplica.

#### **4.2.9. Interfase**

##### **Interfase de usuario**

1. Se desarrollará una interfase de usuario para el modelado de los procesos.  
(MODELADOR)
2. Los procesos se modelarán en forma gráfica con la funcionalidad drag&drop
3. Se desarrollará una interfase de usuario para control operativo de los procesos.  
(MONITOR)

##### **Interfase de software**

4. El middleware de comunicación será el SIX/TCL en su versión 3.2
5. Soportar los siguientes mecanismos de interacción con los sistemas externos:
  - Base de datos
  - Archivos
  - TCP/IP
6. Soportar los siguientes tipos de archivo y/o mensajes TCP
  - CSV

- XML
- Mensajes con bitmap
- Campos de longitud fija.

#### **4.2.10. Licenciamiento**

1. Se dispondrá de 2 tipos de licencia:

- Licencia de operación: incluye únicamente el MOTOR y el MONITOR del sistema
- Licencia completa: incluye el MOTOR, MONITOR y el MODELADOR del sistema.

#### **4.2.11. Requerimientos legales y derechos de autor**

No aplica.

#### **4.2.12. Estándares aplicables**

No aplica.

### **4.3. Seguridad del software**

#### **Dentro del sistema**

Dentro del sistema la seguridad se centrará en lo siguiente:

- Manejo de MAC de seguridad en los registros de la base de datos
- Manejo de MAC de seguridad en los archivos externos de configuración

#### **Entrada al sistema**

Para la entrada del sistema se utilizará el control de usuario y clave. Adicionalmente se manejará perfiles de usuarios. El sistema deberá contar con un módulo que permita gestionar perfiles, usuarios y claves.

**Seguridad física**

La seguridad física dependerá de la organización donde se instale el sistema

**Controles administrativos**

Los controles administrativos lo define la organización donde se instale el sistema

**Aspectos legal y social**

No aplica

**4.4 Modelo de Casos de Uso del Sistema.****4.4.1. Lista de los actores del sistema.**

La lista de actores del sistema es la siguiente:

***Analista Programador***

El analista programador construye los procesos en la aplicación. Realiza el mantenimiento de sistemas y procesos.

***Operador***

El operador realiza las labores de implementación y monitoreo de la aplicación. Sus funciones principales son

- Implementa los procesos en ambiente de producción
- Configura los conectores
- Habilita/Deshabilita procesos
- Inicia/Termina la ejecución del motor de la aplicación
- Controla la operación de la aplicación

***Autorizador***

El autorizador realiza labores de control del desarrollo de procesos. Sus funciones principales son:

Aprobar la definición de un proceso

- Aprobar la modificación de la definición de un proceso.

***Gerente de Proyecto***

Es una especialización del actor “Autorizador”. Es el rol de gerente de proyectos en Novatronic

***Jefe***

Es una especialización del actor “Autorizador”. Es el rol de Responsable designado por el Cliente.

***Administrador del sistema***

El administrador del sistema es responsable del sistema de control de acceso. Mantiene los usuarios y los perfiles de usuario, asigna perfiles a cada usuario.

***Usuario***

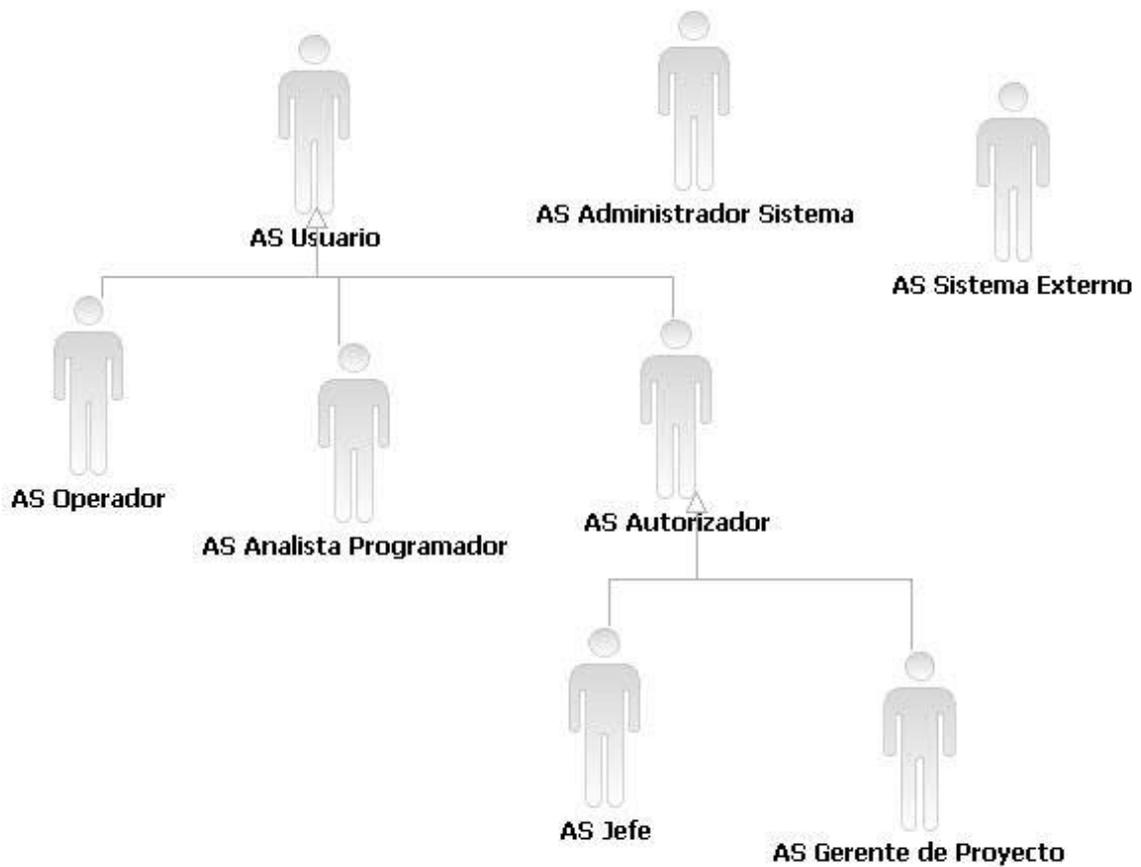
Usuario es una generalización de analista programador, operador y autorizador. Sus funciones son:

- Ingresar al sistema
- Cambiar su contraseña

***Sistema externo***

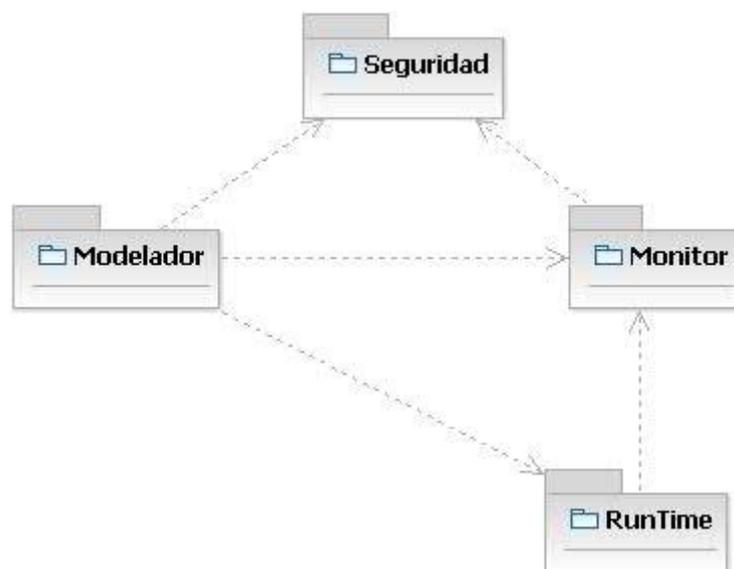
El sistema externo dispara la ejecución de un proceso determinado. El sistema externo realiza una acción que está configurada como disparador (evento inicial) de un proceso, iniciando de este modo el proceso.

**4.4.2. Diagrama de actores del sistema.**



**Ilustración 9: Diagrama de actores del sistema**

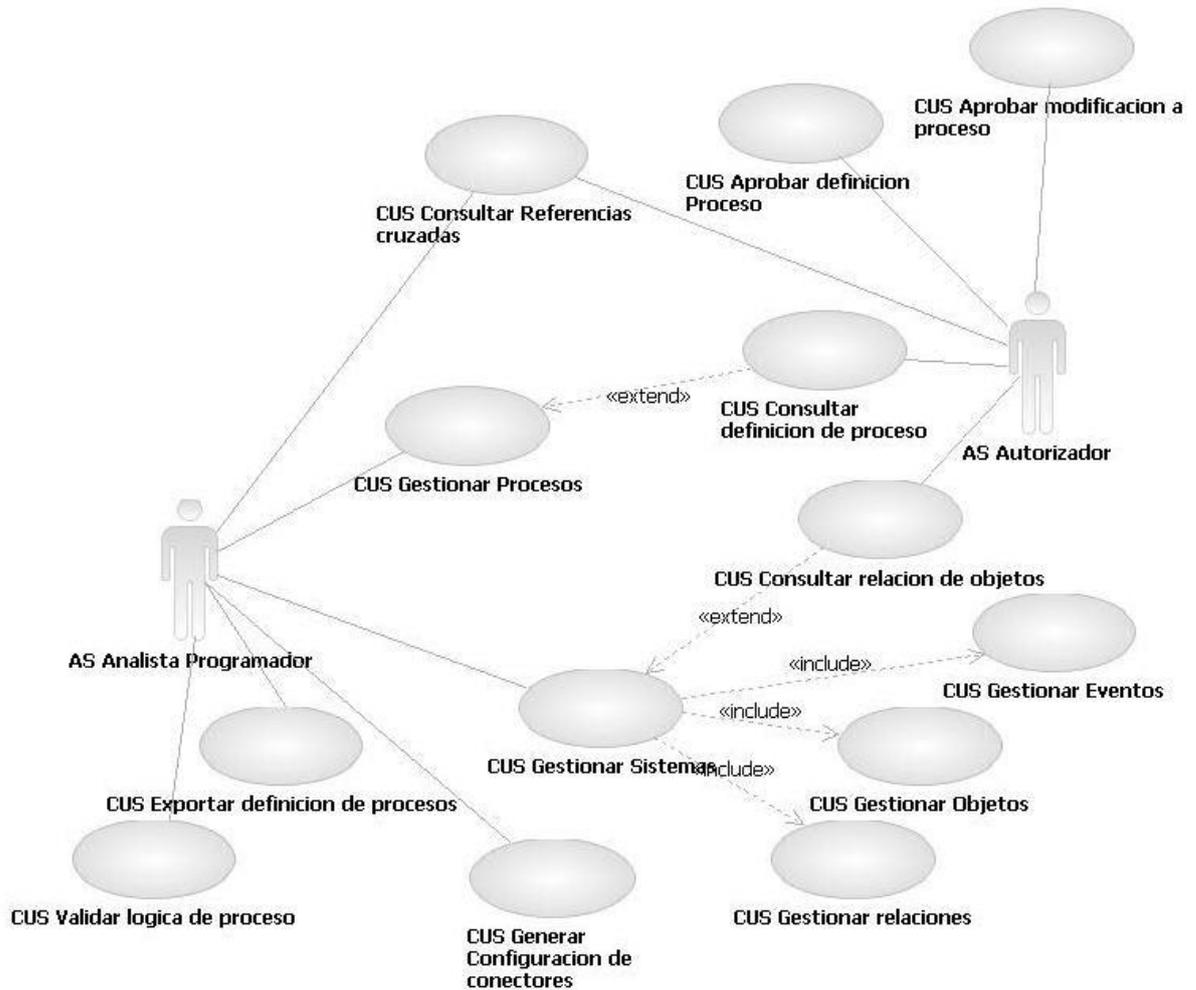
**4.4.3. Diagrama de paquetes.**



**Ilustración 10: Diagrama de paquetes del sistema**

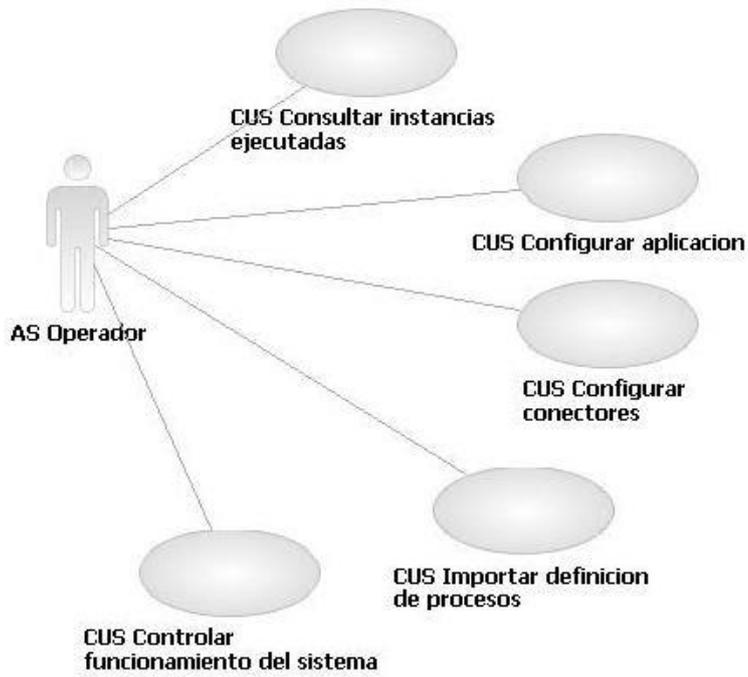
**4.4.4. Diagrama de casos de uso del sistema por paquete.**

*Paquete: MODELADOR*



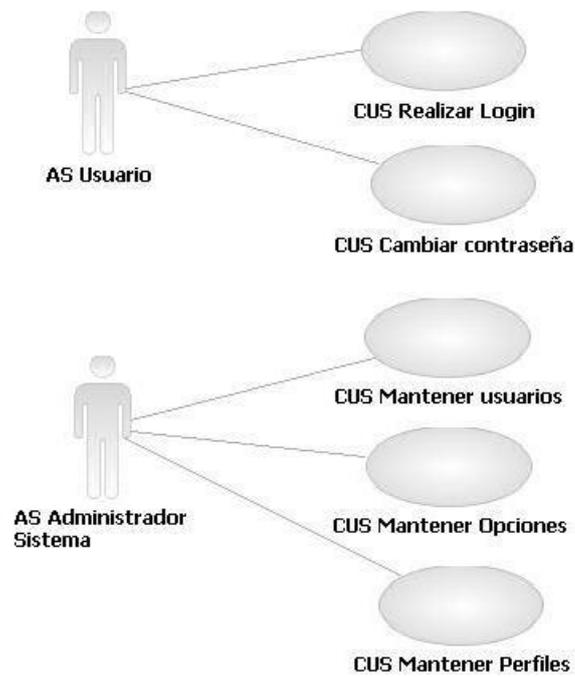
**Ilustración 11: Casos de uso del paquete: MODELADOR**

**Paquete: MONITOR**



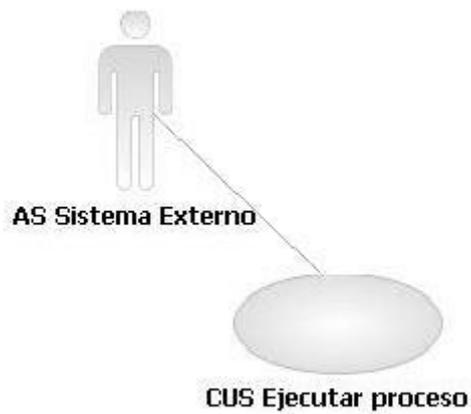
**Ilustración 12: Casos de uso del paquete: MONITOR**

**Paquete: SEGURIDAD**



**Ilustración 13: Casos de usos del paquete SEGURIDAD**

***Paquete: RUNTIME***



**Ilustración 14: Casos de uso del paquete RUNTIME**

**4.5. Matriz de modelo del negocio y modelo del sistema.**

Caso del uso del negocio		Actividad a automatizar			Requerimiento funcional				Caso de uso del sistema		
Nº	Nombre	Nº	Nombre	Trabajador	Nº	Nombre	Nº	Nombre	Actor		
1	Desarrollar solución y Actualizar Solución	1	Realizar programación y pruebas unitarias	Equipo de Trabajo	1	Registrar los sistemas externos y el mecanismo de interacción	1	Gestionar sistemas	AnalistaProgramador		
					2	Registrar los eventos que conformaran un proceso de negocio	2	Gestionar eventos			
					3	Modelar la información que se intercambia entre los sistemas y las transformaciones de esa información.	3	Gestionar objetos			
					4	Modelar los procesos como una secuencia de tareas o pasos que deben ejecutarse	4	Gestionar Procesos			
					5	Soportar flujos alternos en el modelado de procesos, o flujos condicionales					
					6	Consultar el modelo desarrollado para un proceso de negocio					
					7	Consultar referencias cruzadas entre Sistemas y Procesos de Negocio					
					8	Exportar el modelo completo de un proceso de negocio	5	Exportar definición de proceso			
					9	Detectar inconsistencias en el modelo de un proceso de negocio.	6	Validar proceso			
					10	Generar la configuración de conectores para cada SISTEMA involucrado en un proceso de negocio	7	Generar Configuración Conectores			
					11	Importar el modelo completo de un proceso de negocio	8	Importar definición de proceso	Operador		
					12	Mantener un registro en tiempo real de la ejecución de un proceso de negocio. Para cada proceso de negocio en ejecución debe registrarse los tiempos consumidos en cada paso, los pasos ejecutados, el paso que se esta ejecutando	9	Consultar instancias de procesos			
					13	Mantener un registro de los procesos de negocio ejecutados. Para cada ejecución realizada debe registrarse el resultado del proceso, la duración total del mismo, la duración de cada paso.					
					14	Consultar las instancias de procesos de negocio ejecutadas.					

				15	Detener o suspender temporalmente la ejecución de un proceso de negocio	10	Controlar el funcionamiento del sistema	
				16	Habilitar / deshabilitar procesos de negocio			
				17	Detectar la condición de lanzamiento de un proceso de negocio.	11	Ejecutar proceso	Sistema externo
				18	Controlar cada paso del proceso negocio a ejecutar. En caso de errores, detener la ejecución del proceso.			
	2	Ejecutar ADC	Administrador de la configuración	19	Extraer una imagen en tiempo real de los modelos de procesos registrados para administración de la configuración.	12	Exportar definición de proceso	AnalistaProgramador

**4.6. Especificación de alto nivel de los casos de uso del sistema.**

Caso de uso	Aprobar Definición Proceso
Actor(es)	Autorizador
Propósito	Marcar como aprobada una definición de proceso
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el autorizador solicita al sistema la lista de procesos pendientes de aprobación. Selecciona un proceso y firma la aprobación.
Clasificación	Secundario

Caso de uso	Aprobar modificación a proceso
Actor(es)	Autorizador
Propósito	Permitir la modificación de un proceso
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el autorizador solicita al sistema la lista de procesos aprobados. El autorizador selecciona uno de los procesos y firma la autorización para la modificación del proceso. El caso de uso termina cuando la autorización es registrada.
Clasificación	Secundario

Caso de uso	Cambiar contraseña
Actor(es)	Usuario
Propósito	Permitir al usuario cambiar la contraseña de acceso a la aplicación
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el usuario solicita la modificación de contraseña. El usuario ingresa la contraseña actual, la nueva contraseña y confirmación de la nueva contraseña. El caso de uso termina cuando se ha cambiado la contraseña.
Clasificación	Opcional

Caso de uso	Configurar aplicación
Actor(es)	Operador
Propósito	Realizar la configuración de la aplicación
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el operador solicita al sistema la relación de aplicaciones definidas. De acuerdo a su requerimiento, el operador puede registrar o modificar la información de la configuración. El caso de uso termina

	cuando la información de la configuración se ha registrado.
Clasificación	Secundario

Caso de uso	Configurar conectores
Actor(es)	Operador
Propósito	Realizar la configuración de los conectores
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el operador selecciona el conector a configurar y de acuerdo a su requerimiento el operador puede registrar o modificar la información de la configuración. El caso de uso termina cuando la información de la configuración se ha registrado.
Clasificación	Secundario

Caso de uso	Consultar definición de proceso
Actor(es)	Autorizador Analista Programador
Propósito	Mostrar la definición completa de un proceso incluyendo los objetos de negocio involucrados
Caso de uso asociado	Gestionar proceso (extendido)
Resumen	El caso de uso comienza cuando el actor solicita la definición de un proceso. El actor selecciona el proceso a consultar. La aplicación genera un reporte que incluye las tareas de que consta el proceso y los objetos (información) relacionados a estas tareas. El caso de uso termina con el reporte generado
Clasificación	Secundario

Caso de uso	Consultar instancias ejecutadas
Actor(es)	Operador
Propósito	Entregar información de los procesos de negocio ejecutados
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el operador ingresa un periodo para la consulta. El sistema entrega un reporte consolidado de los procesos ejecutados, número de cada proceso, tiempo promedio de duración, resultado del proceso. El caso de uso termina con el reporte generado.
Clasificación	Secundario

Caso de uso	Consultar referencias cruzadas
Actor(es)	Analista Programador Autorizador
Propósito	Entregar un reporte con las referencias cruzadas entre SISTEMAS y PROCESOS de NEGOCIO
Caso de uso asociado	

Resumen	El caso de uso comienza cuando el actor solicita las referencias cruzadas. El actor selecciona el sistema o proceso raíz. La aplicación genera las referencias a los procesos o los sistemas asociados al elemento raíz. El caso de uso termina con el reporte de referencias
Clasificación	Secundario

Caso de uso	Consultar relación de objetos
Actor(es)	Analista programador Autorizador
Propósito	Mostrar la relación de objetos definidos para un sistema
Caso de uso asociado	Gestionar sistemas (extendido)
Resumen	El caso de uso comienza cuando el actor selecciona el sistema a buscar. La aplicación genera un reporte con todos los objetos definidos dentro del sistema, con todos los atributos definidos. El caso de uso termina con el reporte generado.
Clasificación	

Caso de uso	Controlar funcionamiento del sistema
Actor(es)	Operador
Propósito	Controlar la aplicación
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el operador ingresa a la aplicación. Puede seleccionar entre varias acciones: detener un proceso, habilitar/deshabilitar un proceso, suspender un proceso por un periodo determinado entre otras. La aplicación ejecuta la tarea indicada por el operador. El caso de uso termina cuando la tarea se ha ejecutado
Clasificación	Secundario

Caso de uso	Ejecutar proceso
Actor(es)	Sistema externo
Propósito	Ejecutar un proceso definido en la aplicación
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el sistema externo ejecuta la acción definida como disparador de un proceso de negocio. La aplicación ejecuta el proceso de acuerdo a la definición almacenada. El caso de uso termina cuando la última tarea del proceso se ha realizado o cuando ocurre un error en el flujo del proceso.
Clasificación	Primario

Caso de uso	Exportar definición de procesos
-------------	---------------------------------

Actor(es)	Analista Programador
Propósito	Exportar la definición de un proceso para su posterior implementación
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el analista programador selecciona el proceso a exportar. El proceso debe estar previamente validado y aprobado por el Gerente de Proyecto. El analista programador realiza la exportación. El caso de uso termina cuando se ha creado el archivo de exportación
Clasificación	Primario

Caso de uso	Generar configuración de conectores
Actor(es)	Analista Programador
Propósito	Generar los archivos de configuración de los conectores
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el analista programador selecciona los sistemas para los cuales se van a generar los archivos de configuración y ejecuta la acción. El caso de uso termina cuando los archivos de configuración son generados
Clasificación	Secundario

Caso de uso	Gestionar Evento
Actor(es)	Analista Programador
Propósito	Mantener actualizado el registro de los eventos de los sistemas.
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el analista programador realiza el mantenimiento de eventos del sistema. Según el requerimiento, el analista programador puede agregar, modificar o eliminar la información de los eventos de un sistema. El caso de uso termina cuando se registra en la base de datos la información trabajada.
Clasificación	Primario

Caso de uso	Gestionar Objetos
Actor(es)	Analista Programador
Propósito	Mantener actualizado la información de los objetos del sistema
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el analista programador realiza el mantenimiento de objetos del sistema. Según el requerimiento, el analista programador puede agregar, modificar o eliminar la información de los objetos de un

	sistema. El caso de uso termina cuando se registra en la base de datos la información trabajada
Clasificación	Primario

Caso de uso	Gestionar Proceso
Actor(es)	Analista Programador
Propósito	Mantener actualizado la definición de los procesos
Caso de uso asociado	Consultar definición de proceso (extend)
Resumen	El caso de uso comienza cuando el analista programador selecciona el proceso. Según el requerimiento, el analista programador puede agregar, modificar o eliminar los procesos. El caso de uso termina cuando se registra en la base de datos la información trabajada de los procesos
Clasificación	Primario

Caso de uso	Gestionar Relaciones
Actor(es)	Analista Programador
Propósito	Mantener actualizada la información de relaciones de objetos
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el analista programador solicita al sistema la lista de objetos definidos. Según el requerimiento el analista programador puede agregar, modificar o eliminar la información de las relaciones de objetos de un sistema. El caso de uso termina cuando se registra en la base de datos la información trabajada
Clasificación	Primario

Caso de uso	Gestionar Sistema
Actor(es)	Analista Programador
Propósito	Mantener actualizada la información de los sistemas
Caso de uso asociado	Gestionar Eventos (include) Gestionar Objetos (include) Gestionar Relaciones (include) Consultar relación de objetos (extend)
Resumen	El caso de uso comienza cuando el analista programador solicita la lista de sistemas. Según el requerimiento el analista programador puede agregar, modificar o eliminar la información de los sistemas. El caso de uso termina cuando se registra en la base de datos la información trabajada
Clasificación	Primario

Caso de uso	Importar definición de procesos
Actor(es)	Operador
Propósito	Realizar la importación de las definiciones de los procesos
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el operador selecciona el archivo origen y ejecuta la acción. El caso de uso termina cuando la importación arroja un resultado. En caso satisfactorio se deben haber creado todas las entidades asociadas al proceso.
Clasificación	Primario

Caso de uso	Mantener opciones
Actor(es)	Administrador de sistema
Propósito	Mantener la relación de opciones que pueden ejecutarse en la aplicación
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el Administrador de sistema solicita la lista de opciones registradas. El administrador puede añadir nuevas opciones, modificar la información de opciones registrados, bloquear opciones. La aplicación ejecuta la operación seleccionada por el administrador. El caso de uso termina cuando se ha ejecutado la operación.
Clasificación	Opcional

Caso de uso	Mantener perfiles
Actor(es)	Administrador del sistema
Propósito	Mantener los perfiles que se asignan a los usuarios de la aplicación
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el Administrador de sistema solicita la relación de perfiles registrados. El administrador puede añadir nuevos perfiles, modificar la información de perfiles registrados, bloquear o eliminar perfiles. La aplicación ejecuta la operación seleccionada por el administrador. El caso de uso termina cuando se ha ejecutado la operación.
Clasificación	Opcional

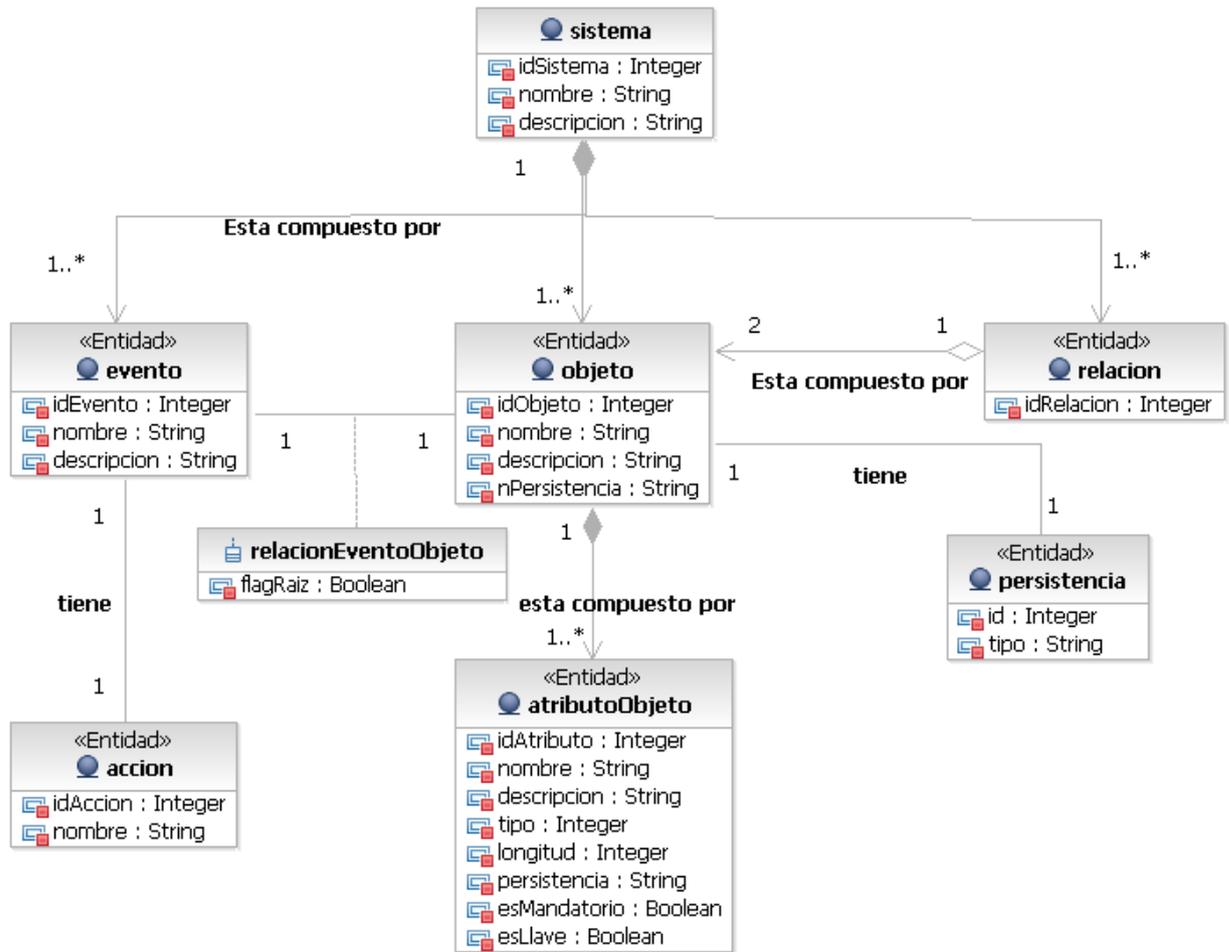
Caso de uso	Mantener usuarios
Actor(es)	Administrador del sistema
Propósito	Mantener la relación de usuario que pueden usar la aplicación
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el Administrador de

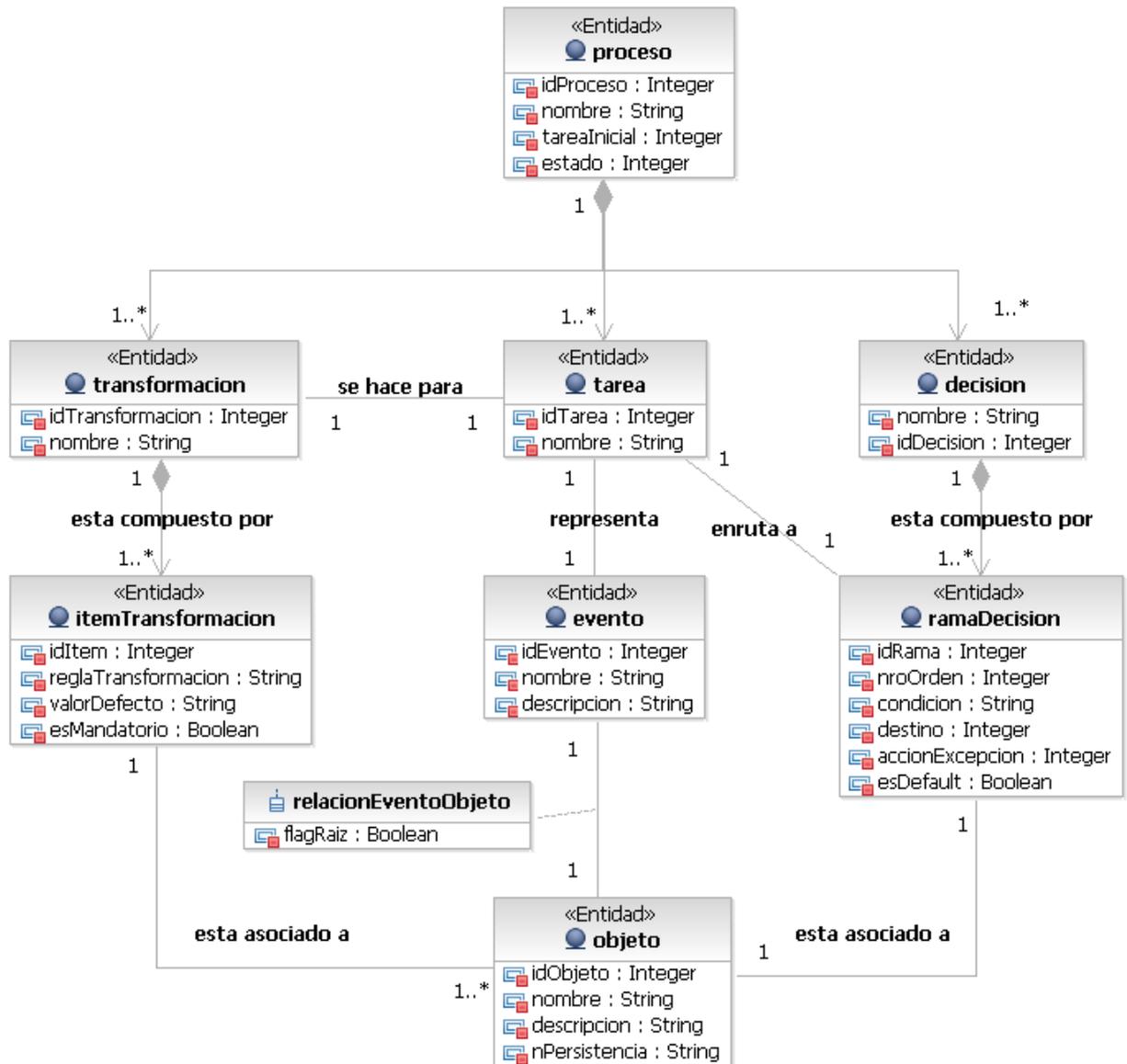
	sistema solicita la relación de usuarios registrados. El administrador puede añadir nuevos usuarios, modificar la información de usuarios registrados, bloquear o eliminar usuarios. La aplicación ejecuta la operación seleccionada por el administrador. El caso de uso termina cuando se ha ejecutado la operación.
Clasificación	Opcional

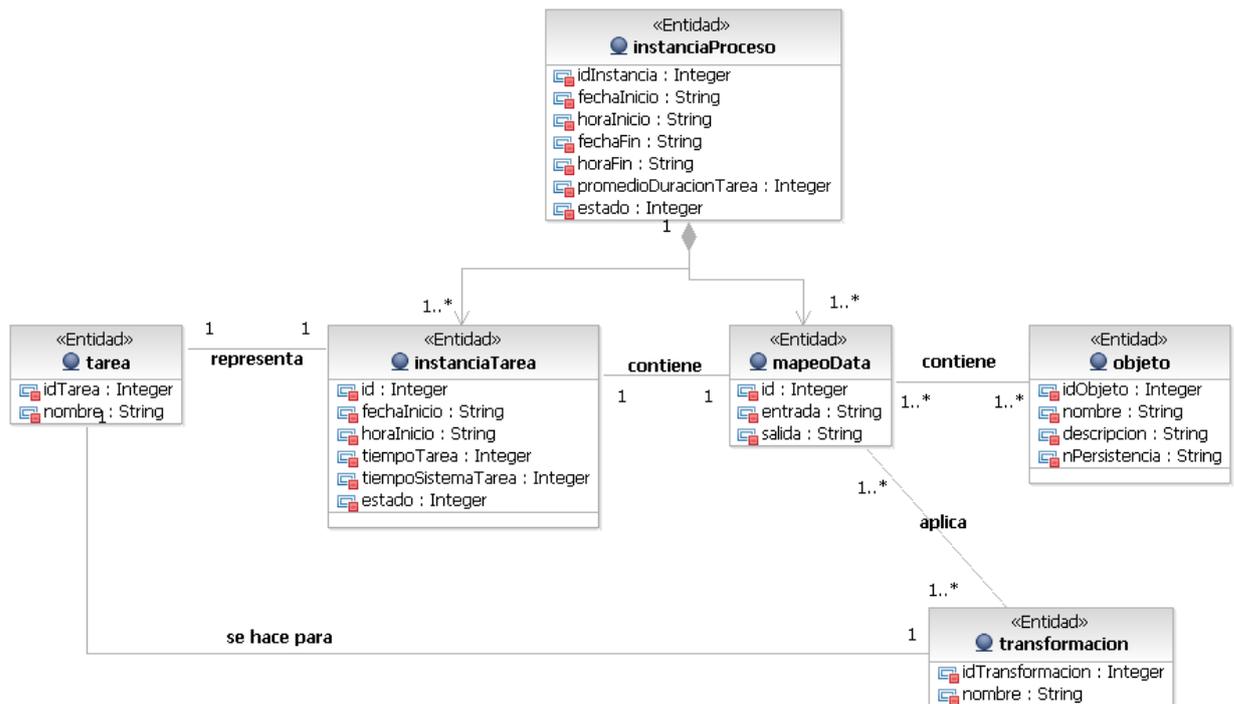
Caso de uso	Realizar login
Actor(es)	Usuario
Propósito	Obtener acceso a los módulos de la aplicación
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el usuario ingresa los el id de usuario y su contraseña. El caso de uso termina cuando el usuario es validado por el sistema y obtiene acceso.
Clasificación	Opcional

Caso de uso	Validar lógica de proceso
Actor(es)	Analista Programador
Propósito	Realizar la validación de las definiciones de los procesos que se hayan realizado
Caso de uso asociado	
Resumen	El caso de uso comienza cuando el analista programador selecciona el proceso a validar y ejecuta la acción. El caso de uso termina cuando el proceso de validación arroja un resultado.
Clasificación	Primario

**4.7. Diagrama de Modelo Conceptual del sistema.**







**4.8. Priorización de los casos de uso.**

En este punto se presenta la clasificación de los casos de uso y la definición de los ciclos de desarrollo en base a dicha clasificación

**4.8.1. Clasificación de los casos de uso.**

Nro	Nombre del caso de uso	Clasificación
1	Controlar funcionamiento del sistema	Primario
2	Ejecutar proceso	Primario
3	Exportar definición de procesos	Primario
4	Generar configuración de conectores	Primario
5	Gestionar proceso	Primario
6	Gestionar sistema	Primario
7	Importar definición de procesos	Primario
8	Validar lógica de proceso	Primario
9	Aprobar definición proceso	Secundario
10	Aprobar modificación a proceso	Secundario
11	Configurar aplicación	Secundario
12	Configurar conectores	Secundario
13	Consultar definición de proceso	Secundario
14	Consultar instancias ejecutadas	Secundario
15	Consultar referencias cruzadas	Secundario
16	Consultar relación de objetos	Secundario

17	Cambiar contraseña	opcional
18	Mantener opciones	opcional
19	Mantener perfiles	opcional
20	Mantener usuarios	opcional
21	Realizar login	opcional

**4.8.2. Ciclos de desarrollo.**

Ciclo de desarrollo	Nombre del caso de uso	Clasificación
Núcleo central	Ejecutar proceso	Primario
	Validar lógica de proceso	Primario
	Gestionar proceso	Primario
	Gestionar sistema	Primario
Ciclo 1	Generar configuración de conectores	Primario
	Importar definición de procesos	Primario
	Exportar definición de procesos	Primario
	Controlar funcionamiento del sistema	Primario
	Aprobar definición proceso	Secundario
	Aprobar modificación a proceso	Secundario
	Configurar aplicación	Secundario
	Configurar conectores	Secundario
Ciclo 2	Consultar definición de proceso	Secundario
	Consultar instancias ejecutadas	Secundario
	Consultar referencias cruzadas	Secundario
	Consultar relación de objetos	Secundario
	Cambiar contraseña	opcional
	Mantener opciones	opcional
	Mantener perfiles	opcional
	Mantener usuarios	opcional
	Realizar login	opcional
Casos de uso asociados	Gestionar eventos (include) Gestionar objetos (include) Gestionar relaciones (include) Consultar definición de proceso (extend) Consultar relacion de objetos (extend)	

**4.9. Especificación de casos de uso del sistema (casos de uso del núcleo central)**

**4.9.1. Caso de uso: Ejecutar proceso**

<p><b><u>Actores del sistema</u></b> Sistema Externo</p> <p><b><u>Propósito</u></b></p>
---

Ejecutar un proceso definido en la aplicación.

### **Breve descripción**

El caso de uso comienza cuando el sistema externo ejecuta la acción definida como disparador de un proceso de negocio. La aplicación ejecuta el proceso de acuerdo a la definición almacenada. El caso de uso termina cuando la última tarea del proceso se ha realizado o cuando ocurre un error en el flujo del proceso.

### **Flujo de eventos**

#### **Flujo básico**

1. El sistema externo realiza una acción que está definida en la aplicación como un disparador de un proceso de negocio
2. El CONECTOR (un componente de la aplicación), detecta que el sistema externo ha realizado la acción y debe iniciarse un proceso de negocio.
3. El CONECTOR consigue la información del sistema externo de acuerdo a su configuración
4. El CONECTOR envía la información obtenida al MOTOR (un componente de la aplicación).
5. El MOTOR recibe la información, la almacena y de acuerdo a la definición del proceso decide cual es el siguiente paso de proceso: Si es una decisión, se ejecuta el subflujo 1, si es una transformación de datos se ejecuta el subflujo 2
6. El MOTOR envía la información recibida al CONECTOR correspondiente
7. El CONECTOR recibe la información y la procesa de acuerdo a su configuración.
8. El CONECTOR devuelve de ser el caso nueva información al MOTOR y un resultado de la acción realizada
9. Cuando el subflujo correspondiente concluye, el MOTOR debe determinar si se ha ejecutado exitosamente el paso de proceso. Si ha ocurrido un error, el MOTOR registra el error en la auditoría y el caso de uso concluye.
10. Si el paso del proceso de negocio se ha ejecutado exitosamente, el MOTOR debe determinar si se ha ejecutado el paso final del proceso de negocio. De ser así, el caso de uso concluye.
11. De no ser el paso final del proceso de negocio, se regresa al punto 5

#### **Subflujos**

##### ***Subflujo 1: Decisión***

- El MOTOR procesa la información recibida del CONECTOR.
- El MOTOR evalúa la CONDICION definida en el paso “decisión” para determinar cual es la siguiente tarea a ejecutar en el proceso de negocio.

##### ***Subflujo 2: Transformación***

- El MOTOR procesa la información recibida del CONECTOR.
- El MOTOR encuentra la tarea destino de la transformación.
- El MOTOR evalúa cada REGLA DE TRANSFORMACION definida en el paso

“transformación” y la ejecuta sobre la información recibida, creándose nueva información.

**Flujos alternos**

No existen

**Precondiciones**

- El MOTOR debe estar en estado activo
- El “proceso de negocio” debe estar en estado activo.
- Los CONECTORES deben estar configurados y ejecutándose.
- El proceso de negocio tiene que estar definido y en estado “validado”.

**Poscondiciones**

- En caso de proceso normal, el proceso de negocio debe haberse ejecutado y existirán los registros del proceso en la auditoría. De estar configurado, el sistema externo habrá obtenido una respuesta.
- En caso de error, se registrará en la auditoría: el proceso fallido, el paso fallido y la condición de error

**Puntos de extensión**

No aplica

**Requerimientos especiales**

No aplica

**Información adicional**

No aplica

**4.9.2. Caso de uso: Validar proceso****Actores del Sistema**

Analista Programador

**Propósito**

Validar un proceso definido en la aplicación.

**Breve descripción**

El caso de uso comienza cuando el analista programador selecciona el proceso a validar. La aplicación realiza la validación del proceso en busca de inconsistencias. De encontrarlas, estas inconsistencias son mostradas al analista y el proceso no cambia de estado. De no encontrar inconsistencias, la aplicación cambia el estado del proceso a ser “validado”. El caso de uso termina cuando la validación se ha realizado.

## **Flujo de eventos**

### **Flujo básico**

1. El analista programador selecciona el proceso a validar.
2. La aplicación realiza una validación del flujo de proceso para encontrar iteraciones sin término.
3. De encontrarse una iteración sin término, se registra un error en el modelo. El proceso continúa.
4. La aplicación realiza una validación de las tareas de decisión del proceso para buscar condiciones ambiguas o con más de un resultado.
5. De encontrarse un error de ese tipo, se registra un error en el modelo. El proceso continúa
6. La aplicación realiza una validación de los objetos asociados a cada proceso para asegurar que exista la definición del objeto.
7. De no existir la definición del objeto, se registra un error en el modelo. El proceso continúa
8. La aplicación realiza una validación de las tareas de transformación de datos del proceso para buscar condiciones ambiguas.
9. De encontrarse un error de ese tipo, se registra un error en el modelo. El proceso continúa
10. Al final de la validación el estado del proceso es actualizado a “validado”.

### **Subflujos**

El caso de uso no tiene subflujos

### **Flujos alternos**

#### ***Flujo alternativo 1: Error en la definición del proceso***

Si se ha detectado uno o más errores en la definición del proceso de negocio, entonces:

1. Se genera una lista de los errores encontrados en el modelo del proceso.
2. El proceso de negocio no cambia de estado.

### **Precondiciones**

- Debe existir el proceso de negocio.

**Poscondiciones**

- En caso que la validación no encuentre errores, el estado del proceso es actualizado a “validado”.

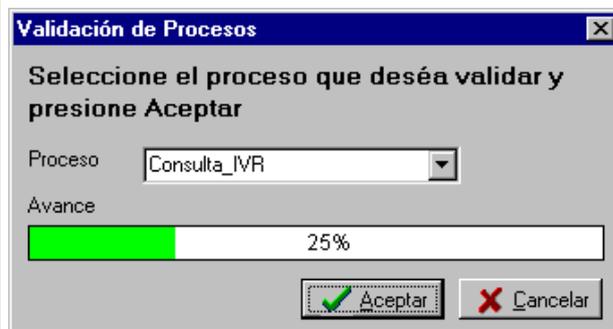
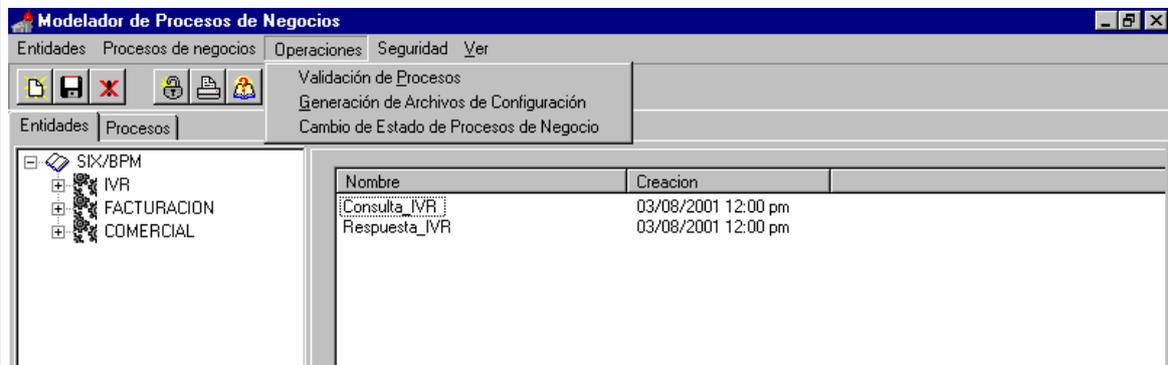
**Puntos de extensión**

No aplica

**Requerimientos especiales**

No aplica

**Información adicional**



**4.9.3. Caso de uso: Gestionar proceso**

**Actores del sistema**

Analista Programador

**Propósito**

Gestionar la definición de los procesos de negocio.

## **Breve descripción**

El caso de uso comienza cuando el analista programador selecciona un proceso o crea un nuevo proceso para trabajar. El analista programador va definiendo cada tarea del proceso, relaciona las tareas creando un flujo de proceso, define las transformaciones de datos que sean necesarios. El proceso no necesariamente debe quedar completo. El caso de uso termina cuando el actor graba los cambios realizados y sale de la aplicación.

## **Flujo de eventos**

### **Flujo básico**

1. El actor selecciona el proceso con el que va trabajar o registra un nuevo proceso.
2. Si se registra un nuevo proceso, el actor debe ingresar los atributos del nuevo proceso.
3. Si se selecciona un proceso existente, el actor puede modificar los atributos del proceso, eliminar el proceso o trabajar en el modelo del proceso.
4. Si elimina un proceso existente, se sigue el flujo alterno “eliminación de proceso”.
5. El actor decide selecciona el paso de proceso: evento, transformación o decisión.
6. El actor une cada “paso” de proceso creando un flujo de proceso (estableciendo precedencia entre los pasos del proceso)
7. Cuando los pasos del proceso han sido unidos, el actor decide cual paso trabajar.
8. Si es un “evento” se sigue el subflujo 1, Si es una transformación se sigue el subflujo 2, si es una decisión se sigue el subflujo 3.
9. Si se va a trabajar otro paso de negocio se regresa al punto 5.
10. El actor graba los cambios realizados y termina el caso de uso.

### **Subflujos**

#### ***Subflujo 1: Evento***

- El actor define la tarea a ejecutar: insert, delete, update, deep delete, retrieve para la tarea
- El actor define los atributos del objeto de negocio que se manipularán en la tarea.

#### ***Subflujo 2: Decisión***

- El actor define la expresión que se evaluará. La expresión debe estar generada en base a los atributos del objeto relacionado a la tarea origen.
- El actor define las salidas de la tarea “decisión”. Cada salida debe estar conformada por una tupla (valor, destino) siendo *valor* un resultado de la evaluación de la expresión, y *destino* una de las tareas conectadas a la tarea de decisión.

#### ***Subflujo 3: Transformación***

- El actor define los atributos “origen” que se trabajaran. Estos atributos deben provenir de del objeto(s) relacionado(s) a la tarea origen.
- El actor define los atributos “destino” que se trabajaran. Estos atributos deben provenir del objeto(s) relacionado(s) a la tarea destino.

- Para cada par (atributo origen, atributo destino) que se haya definido en el diseño, el actor registra la expresión de transformación que se aplicará al atributo origen para producir el valor del atributo destino.

### **Flujos alternos**

#### ***Flujo alternativo 1: Eliminación de proceso***

En el paso 4 del flujo básico.

- La aplicación solicitará confirmación de la operación
- El actor confirma la acción a realizar, la aplicación realiza la operación y el caso de uso termina.

### **Precondiciones**

- Deben existir “sistemas” registrados en la aplicación.
- Debe existir al menos 2 eventos y objetos de negocio relacionados a esos eventos para poder crear un proceso de negocio.

### **Poscondiciones**

### **Puntos de extensión**

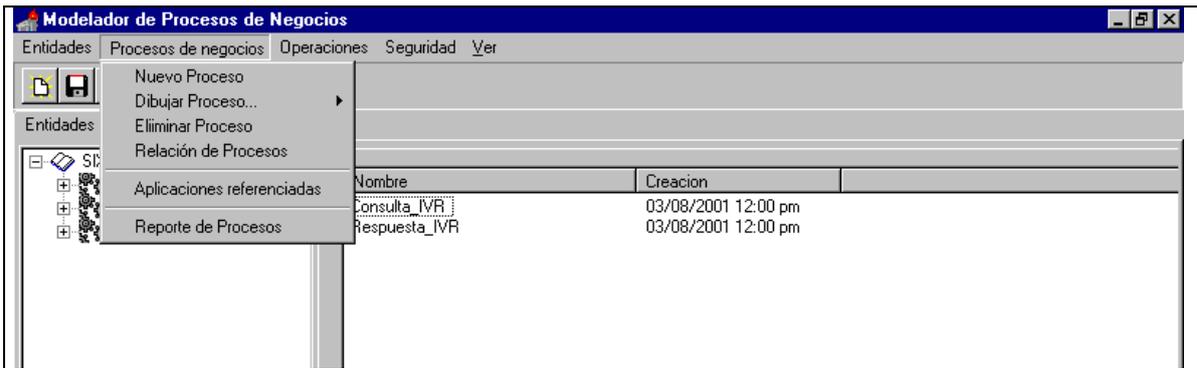
No aplica

### **Requerimientos especiales**

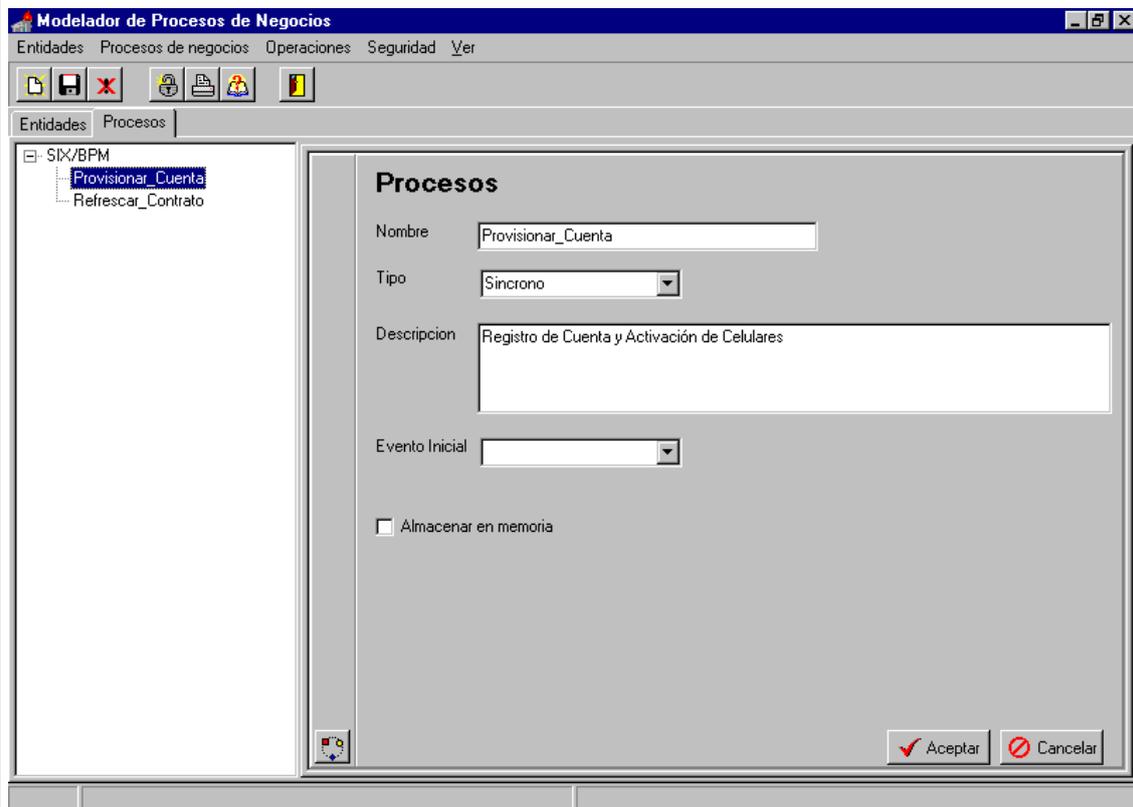
No aplica

### **Información adicional**

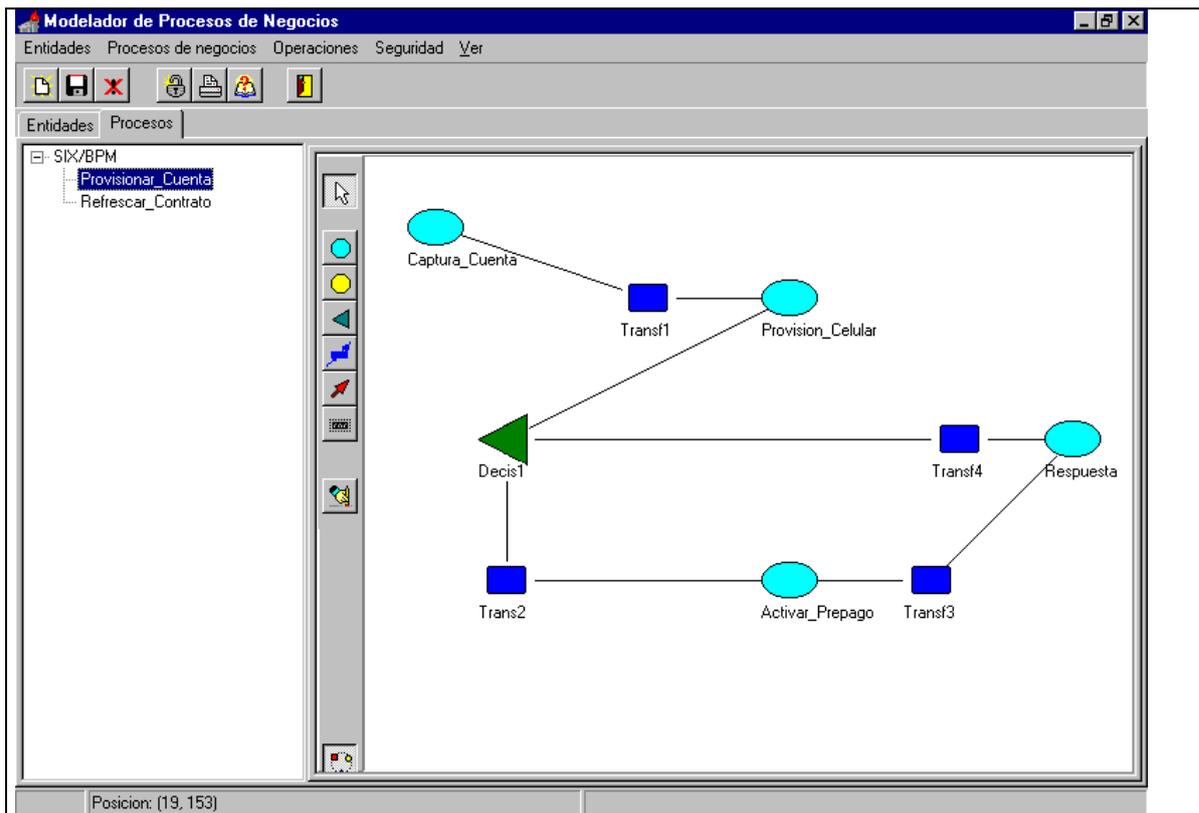
En el siguiente submenú se muestran las opciones disponibles para la administración de los Procesos de Negocio, con ello el Analista Programador podrá definir los Procesos de Negocios en base a las entidades definidas previamente.



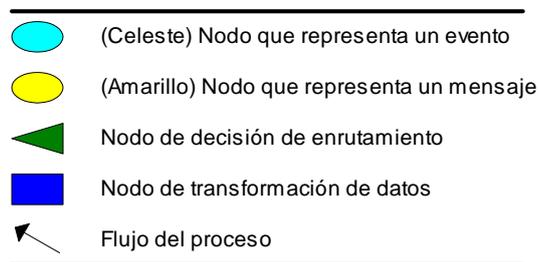
La siguiente pantalla muestra la relación de Procesos de Negocios definidos en la aplicación



La siguiente pantalla muestra el área de trabajo donde el actor diseña los Procesos de Negocio.



Para la versión inicial que se desarrollará, los nodos definidos son los siguientes:



Los objetos de negocio no se representan en el flujo del proceso pues están asociados a los eventos, sin embargo los nodos de transformación de datos y decisión de enrutamiento trabajan en base a ellos.

#### 4.11. Conclusiones.

En este capítulo hemos revisado las especificaciones del sistema y los casos de uso relevantes.

Se ha presentado una especificación inicial del prototipo del MODELADOR del sistema.

## **CAPÍTULO V**

### **ARQUITECTURA DE SOFTWARE**

#### **5.1. Introducción**

En este capítulo nos centraremos en la arquitectura del sistema. Se revisarán los requerimientos y casos de uso que son relevantes para la arquitectura.

#### **5.2. Metas y restricciones de la arquitectura**

##### **Requerimiento de Rendimiento 1**

El sistema debe soportar hasta un máximo de 50 procesos definidos.

##### **Requerimiento de Rendimiento 2**

El sistema debe soportar hasta un máximo de 1000 instancias por cada proceso definido

##### **Requerimiento de Rendimiento 3**

El tiempo de procesamiento directo del sistema en cada proceso de negocio no debe exceder los 2 segundos.

### Requerimiento de Soporte 2

El sistema soportará las siguientes bases de datos:

- Oracle
- SQL Server
- MySQL
- Postgres

### Restricción de diseño 1

El modulo que interactúa con los sistemas externos se desarrollará en J2EE.

### Requerimiento de interfaz de usuario 1

Se desarrollará una interfaz de usuario para el modelado de los procesos. (MODELADOR)

### Requerimiento de interfaz de usuario 3

Se desarrollara una interfaz de usuario para control operativo de los procesos.

(MONITOR)

### Requerimiento de interfaz de software 1

El middleware de comunicación será el SIX/TCL en su versión 3.2.1

### Requerimiento de interfaz de software 2

El sistema deberá soportar los siguientes mecanismos de interacción con los sistemas externos:

- Base de Datos
- Archivos

- TCP/IP

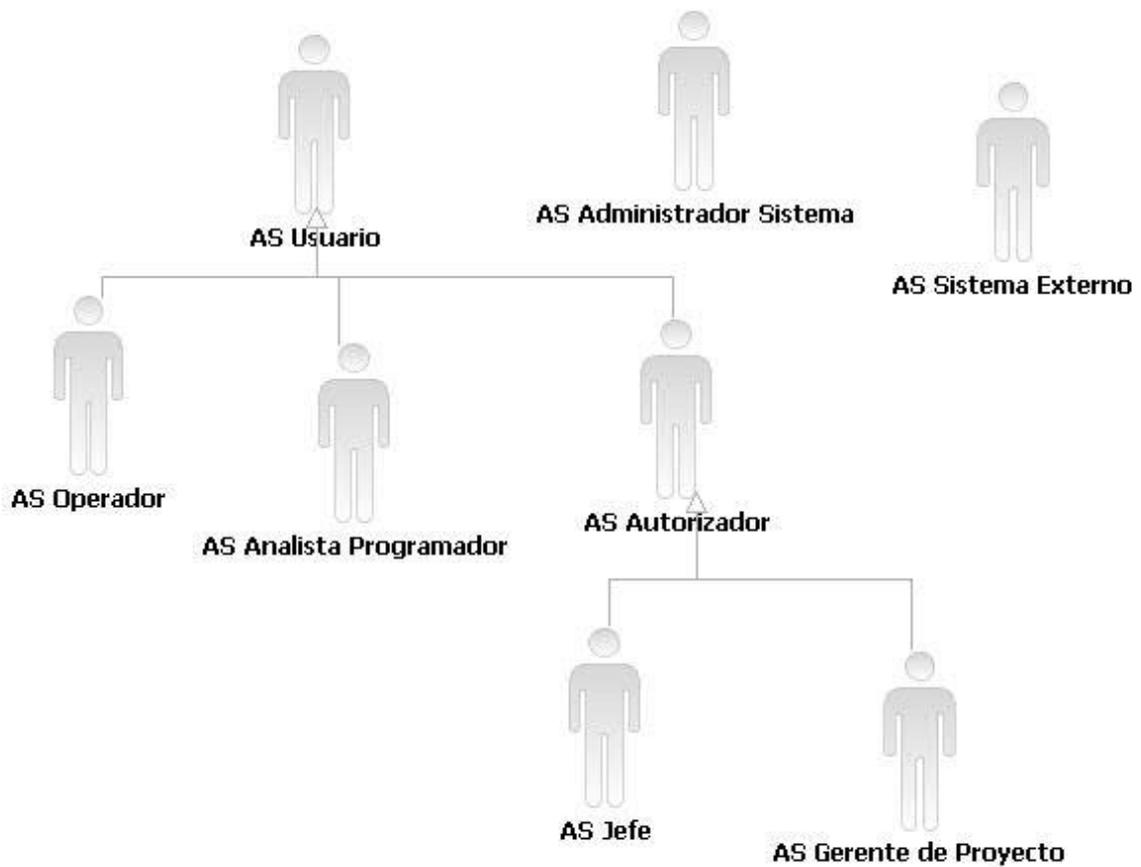
### Requerimiento de interfaz de software 3

El sistema deberá soportar los siguientes tipos de archivo y/o mensajes TCP

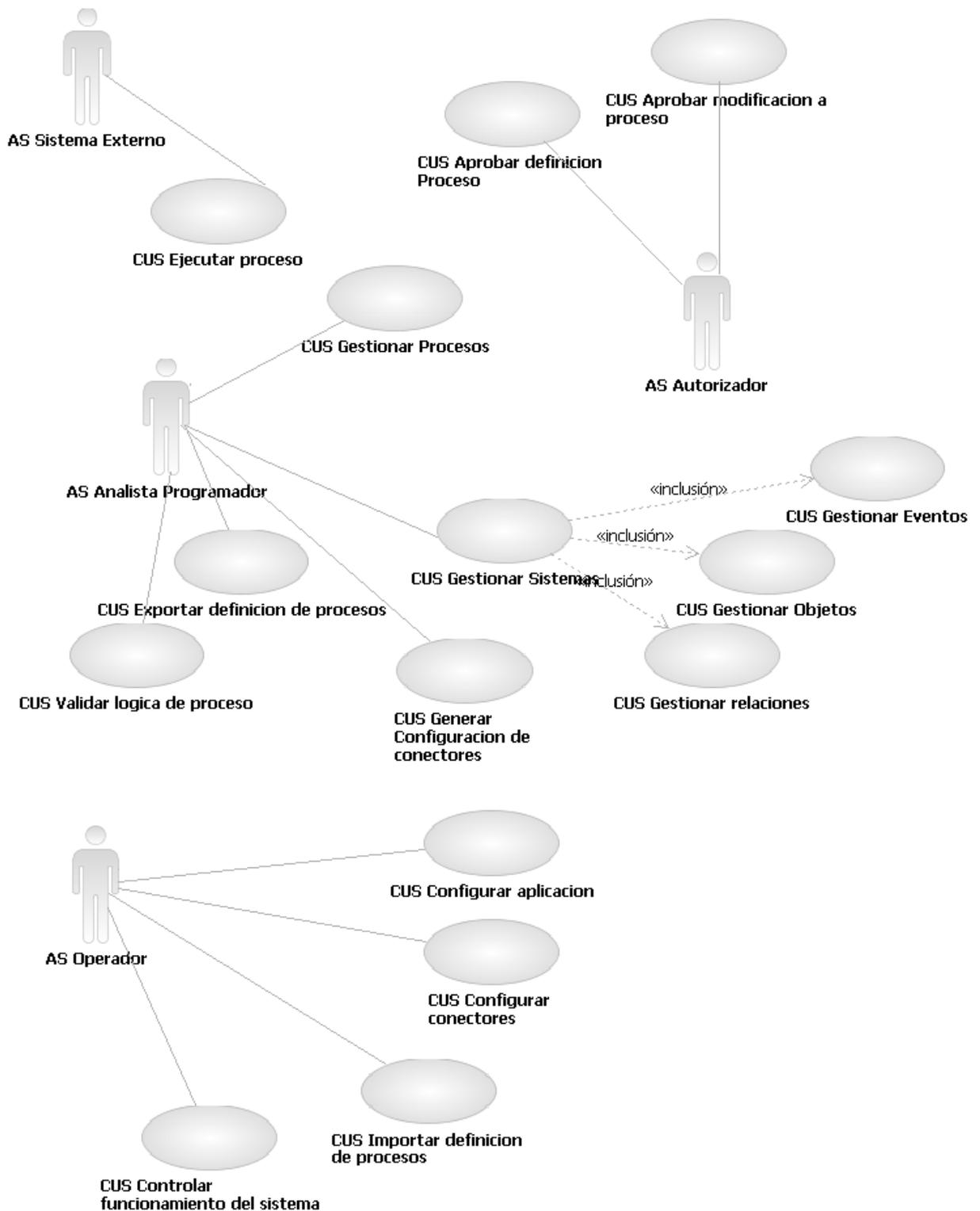
- CSV
- XML
- Mensajes con bitmap
- Campos de longitud fija.

## **5.3. Vista de casos de uso**

### **5.3.1 Diagrama de actores y sus relaciones**



**5.3.2 Diagrama de casos de uso relevantes a la arquitectura del sistema**



**5.4. Mecanismos**

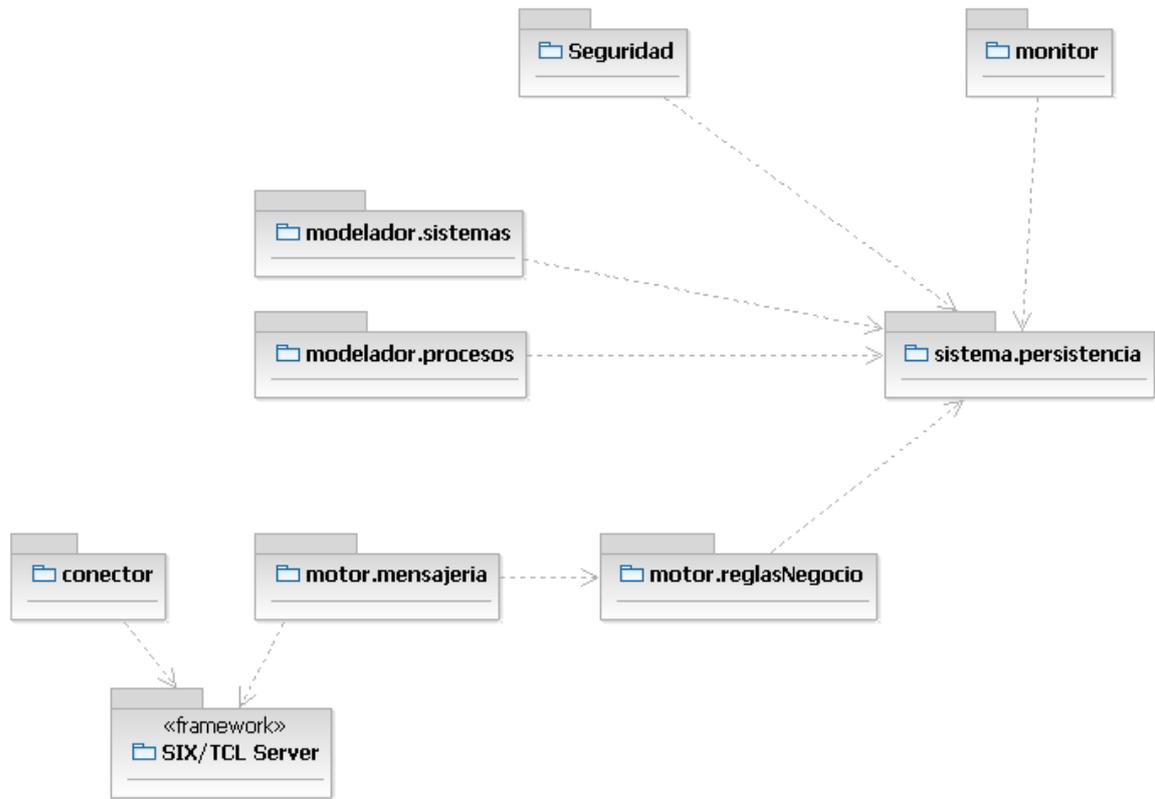
<b>Mecanismo</b>	<b>Envío de Correos Electrónicos</b>	
<b>Descripción</b>	<b>Restricciones al diseño del mecanismo</b>	<b>Solución</b>
Mecanismo que permite a la aplicación enviar correos electrónicos	[RNF10] El sistema deberá generar correos para informar los errores ocurridos en la ejecución de un proceso.	JavaMail

<b>Mecanismo</b>	<b>Registro de usuarios y perfiles de usuario</b>	
<b>Descripción</b>	<b>Restricciones al diseño del mecanismo</b>	<b>Solución</b>
Mecanismo que permite a la aplicación registrar los usuarios y los poderes de los usuarios en el sistema	[RF21] Mantener un sistema de usuarios y perfiles de usuarios con poderes diferentes.	SIX/Access v3.7

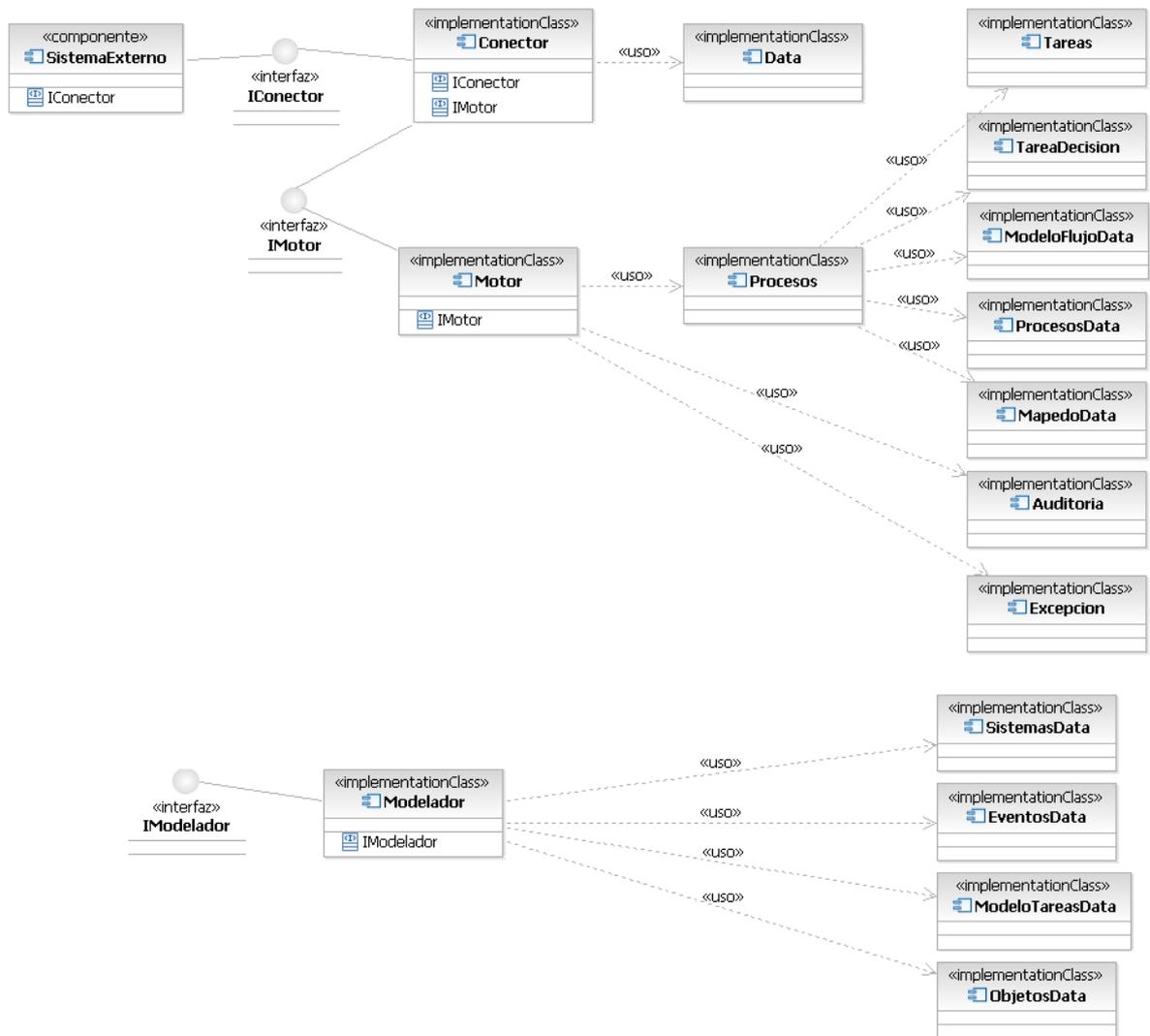
<b>Mecanismo</b>	<b>Cifrado de Datos y Generación de MAC</b>	
<b>Descripción</b>	<b>Restricciones al diseño del mecanismo</b>	<b>Solución</b>
Mecanismo que permite a la aplicación generar y validar MAC para los mensajes de los procesos y el cifrado de la data sensible	[RF8] Todos los mensajes que se intercambien dentro del sistema en un flujo de procesos tendrán MAC. [RF9] El sistema debe soportar el cifrado de datos en los mensajes que se intercambien dentro de un flujo de procesos. En el diseño de un proceso de negocio, el analista programador podrá definir que información se cifra.	Libnvseg_inter.dll Libnvseg_base.dll

<b>Mecanismo</b>	<b>Envío de mensajes entre los conectores y el motor de procesos</b>	
<b>Descripción</b>	<b>Restricciones al diseño del mecanismo</b>	<b>Solución</b>
Mecanismo que permite a la aplicación intercambiar mensajes entre los conectores y el motor de negocios en protocolo TCP/IP	[RNF14] El middleware de comunicación será el SIX/TCL en su versión 3.2.1	API cliente SIX

**5.5. Vista de módulos**

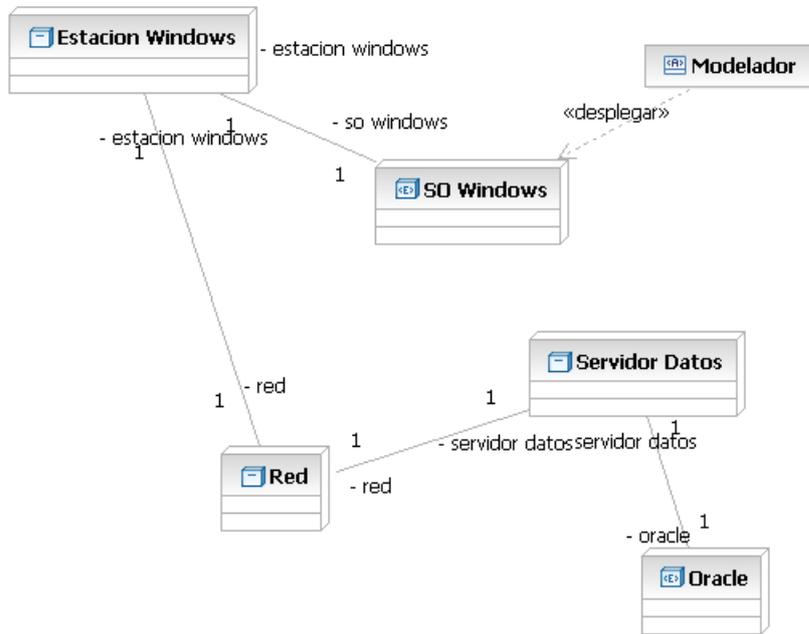


**5.6. Vista de componentes**

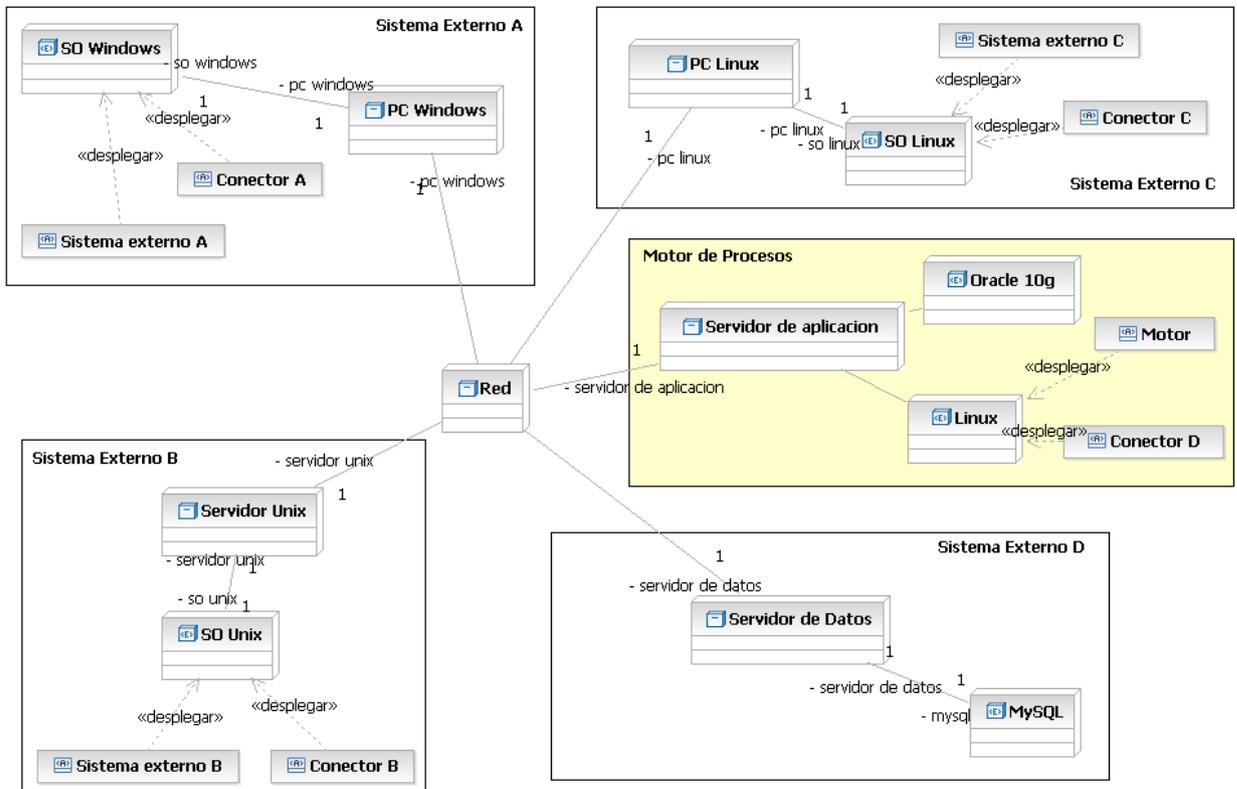


### 5.7. Vista de despliegue

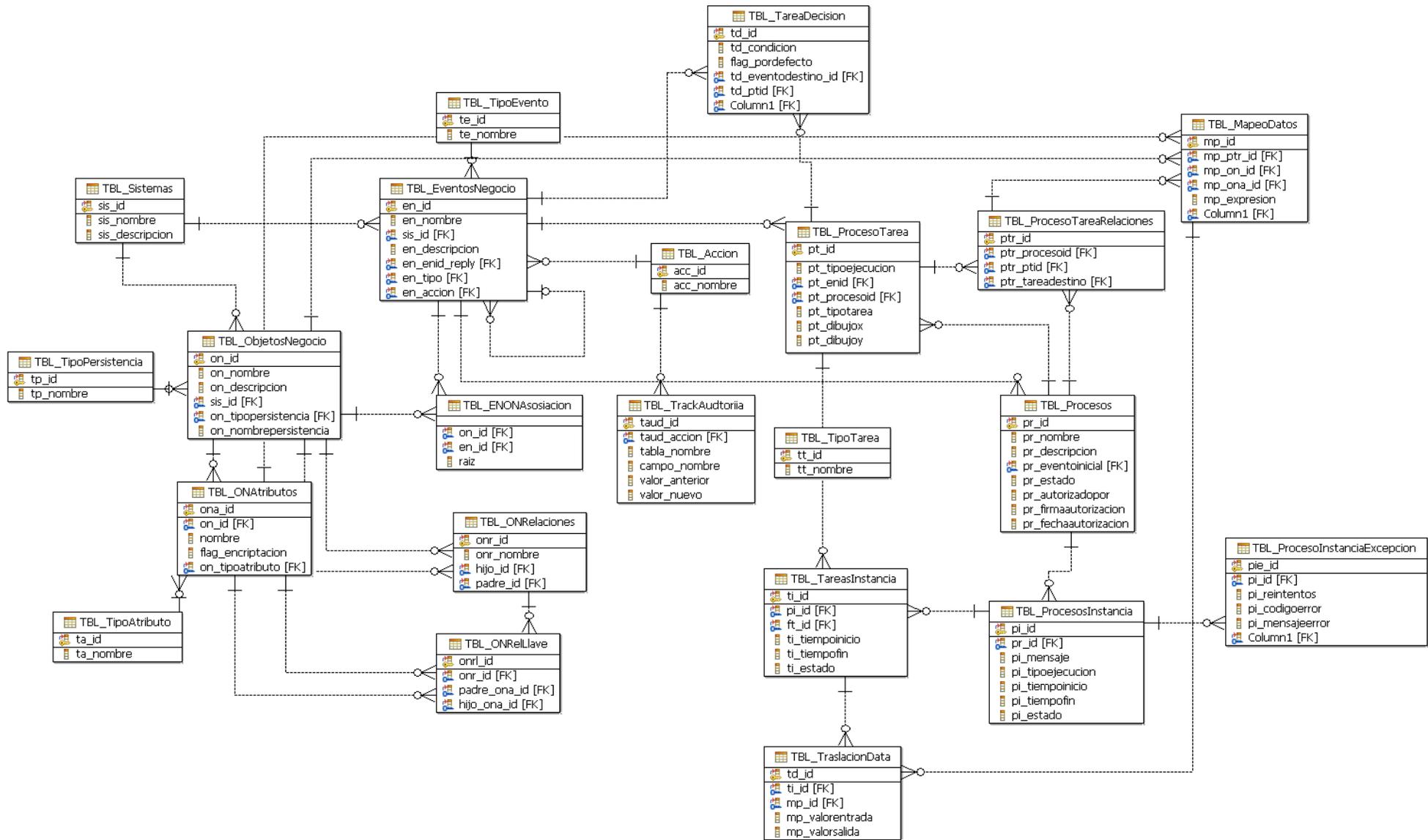
#### 5.7.1 Vista de despliegue de diseño



#### 5.7.2 Vista de despliegue de ejecución



5.8. Vista de datos



## **5.9. Conclusiones**

Las vistas de componentes y de despliegue reflejan la organización de los módulos del software en el entorno de desarrollo. Se ha definido también el diagrama entidad relación inicial del sistema.

## **CAPÍTULO VI**

### **PRUEBAS**

#### **6.1. Introducción**

Este capítulo contiene los escenarios con los casos de prueba que se utilizarán para validar la persistencia de la data y la funcionalidad de los casos de uso del núcleo central.

#### **6.2. Pruebas funcionales del CUS: Ejecutar proceso**

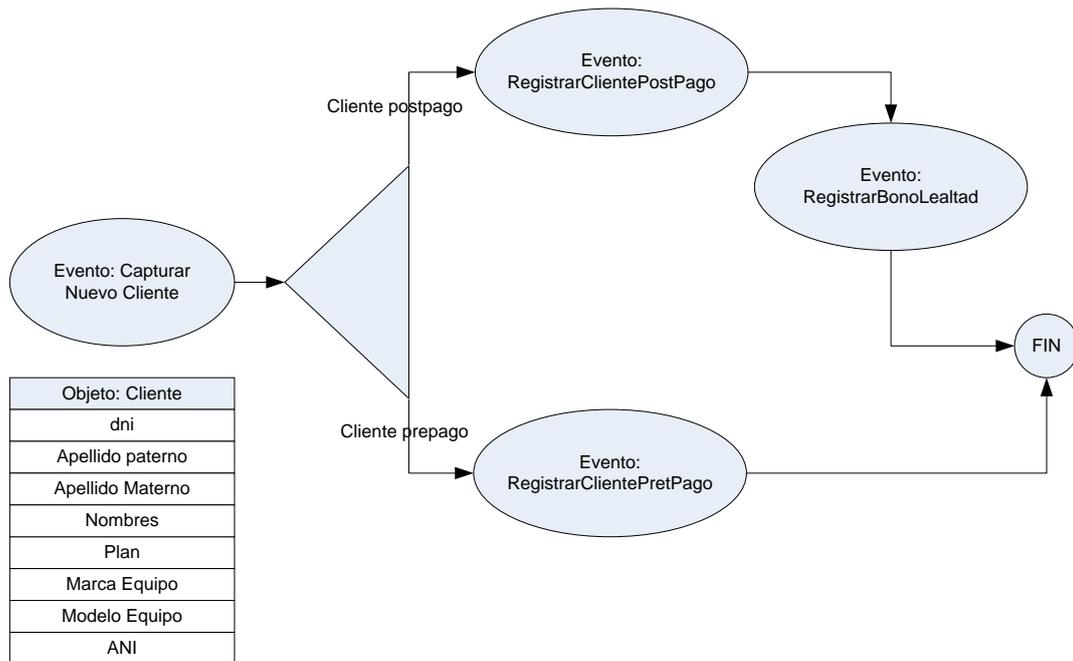
##### **6.2.1. Objetivo de la prueba**

Verificar que el proceso de negocios es ejecutado conforme fue diseñado.

##### **6.2.2. Caso de prueba: <EjeProPF01>**

###### **Data inicial**

Proceso de negocio



1. La aplicación de la TELCO registra una venta de celular a un cliente.
2. Si el plan del celular es prepago, debe registrarse en el sistema Prepago.  
(RegistrarClientePrepago)
3. Si el plan del celular es postpago, debe registrarse en el sistema Postpago  
(RegistrarClientePostPago)
4. Si el plan es postpago y el equipo es Apple/ iPhone, debe adjudicársele 1000 puntos en el sistema de lealtad.
5. Si el plan es postpago y el equipo es Motorola/Q, debe adjudicársele 800 puntos en el sistema de lealtad.

Datos de la “aplicación de la Telco”

Plan
Prepago
Postpago
Control

Marcas
Apple
Motorola
Control

Marcas	Modelo
--------	--------

Apple	iPhone
Motorola	C124
	Q

Dni	Ape paterno	Ape materno	Nombres	Plan	Marca equipo	Modelo	ANI
09301369	Zelada	Briceño	Julio	PostPago	Apple	Iphone	995879993
08998234	Toma	Kiyamu	Juan	Prepago	Motorola	C124	996761234
09289102	Kahatt	Bendezu	Sergio	Postpago	Motorola	Q	996687387
09898921	Sandoval	Malqui	Gilmer	Control	Motorola	C124	995212399
09898922	Tuesta	Sanchez	Luis	Prepago	Motorola	Q	812345

**Condiciones de entrada**

Condición de entrada	Valor
Dni	09301369
Apellido paterno	Zelada
Apellido materno	Briceño
Nombre	Julio
Plan	PostPago
Marca	Apple
Modelo	Iphone
ANI	995879993

**Resultado esperado**

1. Se ejecutaron los eventos: CapturarNuevoCliente, RegistrarClientePostPago y RegistrarBonoLealtad
2. Se registro el ANI “995879993” para el cliente “Zelada Briceño Julio” en el sistema postpago
3. Se registro una bonificación de 1000 puntos en el sistema de lealtada para el cliente “Zelada Briceño, Julio”.

**6.2.3. Caso de prueba: <EjeProPF02>**

**Data inicial**

La misma data del punto 6.2.2.

**Condiciones de entrada**

Condición de entrada	Valor
Dni	08998234
Apellido paterno	Toma
Apellido materno	Kiyamu
Nombre	Juan
Plan	Prepago

Marca	Motorola
Modelo	C124
ANI	996761234

**Resultado esperado**

1. Se ejecutaron los eventos: CapturarNuevoCliente y RegistrarClientePrePago
2. Se registro el ANI “08998234” para el cliente “Toma Kiyamu Juan” en el sistema pretpago

**6.2.4. Caso de prueba: <EjeProPF03>**

**Data inicial**

La misma data del punto 6.2.2.

**Condiciones de entrada**

Condición de entrada	Valor
Dni	09898921
Apellido paterno	Sandoval
Apellido materno	Malqui
Nombre	Gilmer
Plan	Control
Marca	Motorola
Modelo	C124
ANI	995212399

**Resultado esperado**

1. Se ejecuto el eventos: CapturarNuevoCliente
2. Se registro una excepción en el proceso de negocio porque el valor del campo “plan” no permite definir el siguiente evento a ejecutar.

**6.2.5. Caso de prueba: <EjeProPF04>**

**Data inicial**

La misma data del punto 6.2.2.

**Condiciones de entrada**

Condición de entrada	Valor
----------------------	-------

Dni	09898922
Apellido paterno	Tuesta
Apellido materno	Sanchez
Nombre	Luis
Plan	Prepago
Marca	Motorola
Modelo	Q
ANI	812345

**Resultado esperado**

1. Se ejecuto el eventos: CapturarNuevoCliente
2. Se registro una excepción en el proceso de negocio porque el valor del campo “ANI” no tiene formato correcto.

**6.3. Pruebas funcionales del CUS: Validar lógica de Proceso**

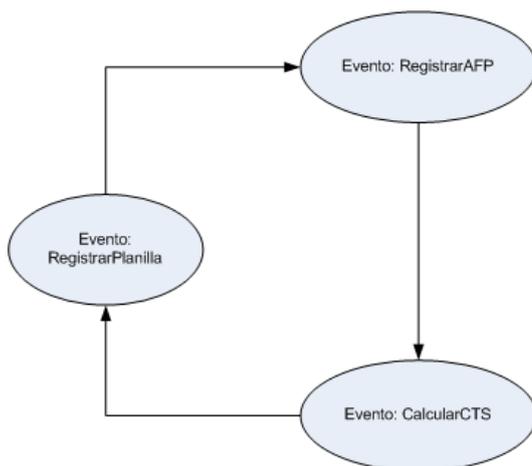
**6.3.1. Objetivo de la prueba**

Verificar que el sistema valida correctamente los procesos de negocio diseñados.

**6.3.2. Caso de prueba: <ValProPF01>**

**Data inicial**

Proceso de negocio



Datos

TBL_ProcesoTarea		
pt_id	pt_enid	pt_procesoid
1	11	1
2	12	1

3	13	1
---	----	---

TBL_EventoNegocio	
en_id	en_nombre
11	Registrar planilla
12	Registrar AFP
13	Calcular CTS

TBL_ProcesoTareaRelacion			
ptr_id	ptr_procesoid	ptr_tareainicial	ptr_tareafinal
111	1	11	12
112	1	12	13
113	1	13	11

**Nota.-** Los datos descritos en las tablas representan el proceso de negocio graficado

**Condiciones de entrada**

pt_procesoid	pr_estado
1	1

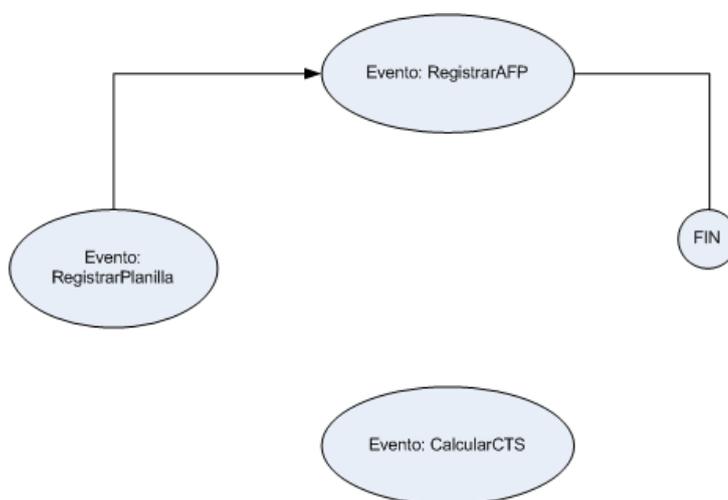
**Resultado esperado**

1. El sistema reporta “Proceso inválido. Motivo: loop sin término.”
2. Estado del proceso se mantiene en 1

**6.3.3. Caso de prueba: <ValProPF02>**

**Data inicial**

Proceso de negocio



Datos

TBL_ProcesoTarea		
pt_id	pt_enid	pt_procesoid
1	11	1
2	12	1
3	13	1

TBL_EventoNegocio	
en_id	en_nombre
11	Registrar planilla
12	Registrar AFP
13	Calcular CTS

TBL_ProcesoTareaRelacion			
ptr_id	ptr_procesoid	ptr_tareainicial	ptr_tareafinal
111	1	11	12
112	1	12	9999

**Nota.-** Los datos descritos en las tablas representan el proceso de negocio graficado

**Condiciones de entrada**

pt_procesoid	pr_estado
1	1

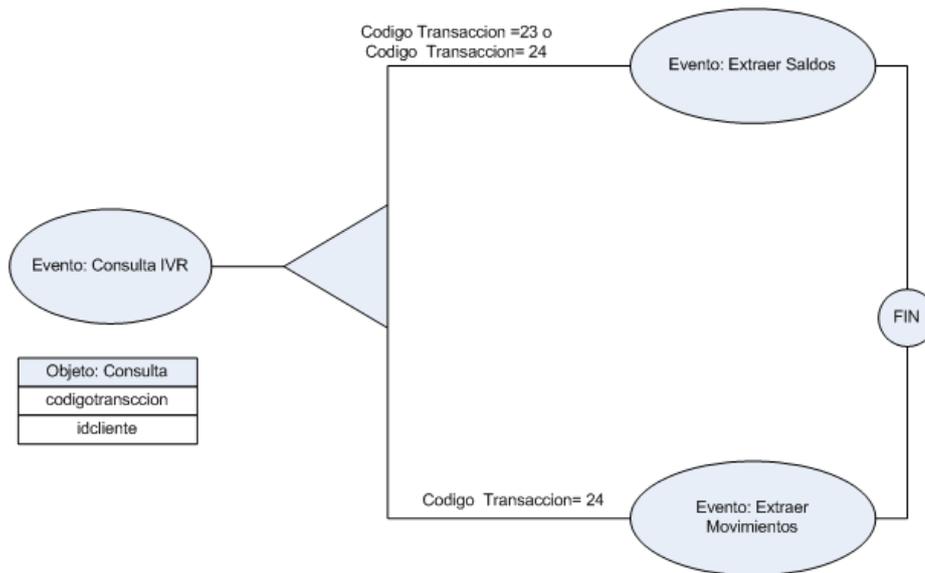
**Resultado esperado**

1. El sistema reporta “Proceso inválido. Motivo: evento aislado (no es parte de un flujo).”
2. Estado del proceso se mantiene en 1

**6.3.4. Caso de prueba: <ValProPF03>**

**Data inicial**

Proceso de negocio



**Datos**

TBL_ProcesoTarea			
pt_id	pt_enid	pt_tipoejecucion	pt_procesoid
1	11	1	1
2	12	1	1
3	13	1	1
4	14	2	1

TBL_TareaDecision				
td_id	td_ptid	td_condicion	td_eventoDestino	pt_procesoid
1	14	([VALOR][CONSULTA.CODIGOTRANSACCION]=23 ) o ([VALOR][CONSULTA.CODIGOTRANSACCION]=24)	12	1
2	14	([VALOR][CONSULTA.CODIGOTRANSACCION]=24)	13	1

TBL_EventoNegocio	
en_id	en_nombre
11	Consulta IVR
12	Extraer saldo
13	Extraer movimientos

TBL_ObjetoNegocio	
on_id	on_nombre
111	CONSULTA

TBL_ONAtributos		
ona_id	on_id	on_nombre
1111	111	CODIGOTRANSACCION

**Nota.-** Los datos descritos en las tablas representan el proceso de negocio graficado

**Condiciones de entrada**

pt_procesoid	pr_estado
1	1

**Resultado esperado**

1. El sistema reporta: “Proceso inválido. Motivo: Condición ambigua en tarea de decision”.
2. Estado del proceso se mantiene en 1

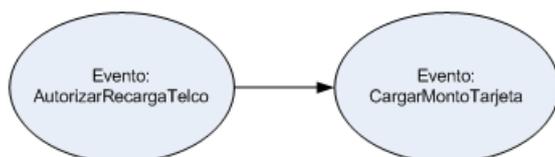
**6.4. Pruebas funcionales del CUS: Gestionar proceso**

**6.4.1. Objetivo de la prueba**

Verificar que el sistema gestiona correctamente los procesos de negocio diseñados.

**6.4.2. Caso de prueba: <GesProPF01>**

**Data inicial**



Un proceso de negocio con dos tareas: AutorizarRecargaTelco y CargarMontoTarjeta.

TBL_EventosNegocio		
en_id	en_Nombre	en_Descripcion
01	AutorizarRecargaTelco	Evento que envía un mensaje de autorización de una recarga a una Telco
02	VerificaPagoTarjeta	Evento que envía un mensaje al autorizador de la institución que emite la tarjeta de crédito

TBL_ProcesoTarea	
ft_id	ft_enid
01	01
02	02

**Condiciones de entrada**

- Se creará una nueva tarea.
- Asignar a la tarea el evento 01 (AutorizarRecargaTelco)

**Resultado esperado**

- El sistema muestra un mensaje de error. Evento duplicado en el proceso

**6.4.3. Caso de prueba: <GesProPF02>**

Se probará la opción “sistemas referenciados” para un proceso de negocio diseñado

**Data inicial**

TBL_Sistema	
sis_id	sis_nombre
01	IVR
02	Facturacion
03	Comercial

TBL_EventosNegocio		
sis_id	en_id	en_nombre
01	001	Consulta_IVR
01	002	Respuesta_IVR
02	003	Recarga_IVR
03	004	Consulta_Saldo
03	005	Consulta_Puntos_Bonus

TBL_Procesos	
pr_id	pr_Nombre
01	Recarga_Virtual

TBL_ProcesoTarea		
ft_id	ft_enid	pt_procesoid
01	001	1
02	002	1
03	003	1

**Condiciones de entrada**

Se obtendrán los sistemas a los cuales referencia el proceso 01

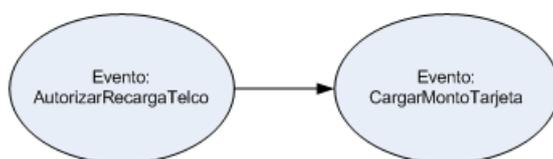
pr_id
01

**Resultado esperado**

Sistema	Evento
IVR	Consulta_IVR
IVR	Respuesta_IVR
Facturación	Recarga_IVR

**6.4.4. Caso de prueba: <GesProPF03>**

**Data inicial**



Un proceso de negocio con dos tareas: AutorizarRecargaTelco y CargarMontoTarjeta.

TBL_EventosNegocio		
en_id	en_Nombre	en_Descripcion
01	AutorizarRecargaTelco	Evento que envía un mensaje de autorización de una recarga a una Telco
02	VerificaPagoTarjeta	Evento que envía un mensaje al autorizador de la institución que emite la tarjeta de crédito

TBL_ProcesoTarea	
ft_id	ft_enid
01	01
02	02

**Condiciones de entrada**

- Se creará una nueva relación que parte de la tarea VerificarPagoTarjeta con destino a la tarea AutorizarRecargaTelco.

**Resultado esperado**

- El sistema muestra un mensaje de error. No se permite relaciones bidireccionales

**6.5. Matriz de trazabilidad entre Casos de uso y Casos de prueba**

Relationships: - direct only			
	CUS1: Ejecutar Proceso	CUS2: Gestionar Proceso	CUS3: Validar logica de proceso
CP1: Caso de prueba: <EjeProPF01>			
CP2: Caso de prueba: <EjeProPF02>			
CP3: Caso de prueba: <EjeProPF03>			
CP4: Caso de prueba: <EjeProPF04>			
CP5: Caso de prueba: <ValProPF01>			
CP6: Caso de prueba: <ValProPF02>			
CP7: Caso de prueba: <ValProPF03>			
CP8: Caso de prueba: <GesProPF01>			
CP9: Caso de prueba: <GesProPF02>			
CP10: Caso de prueba: <GesProPF03>			

## **6.6. Conclusiones**

Se han definido los escenarios de pruebas funcionales para los 3 principales casos de uso.

## CAPÍTULO VII

### ADMINISTRACIÓN DEL PROYECTO

#### **7.1. Introducción**

Este capítulo se centrará principalmente en el desarrollo del acta de constitución del proyecto y en la gestión del alcance.

#### **7.2. Project Charter**

##### **A. Información General**

<b>Nombre del Proyecto</b>	Solución general de integración de procesos de negocios basada en middlewares	<b>Fecha de Preparación</b>	31/05/2008
<b>Patrocinador:</b>	Novatronic S.A.C	<b>Fecha de Modificación:</b>	08/06/2008
<b>Preparado por:</b>	Toma K, Juan / Zelada, Julio	<b>Autorizado por:</b>	

##### **B. Propósito del Proyecto (descripción del producto o servicio)**

El producto del proyecto es el desarrollo de una aplicación que actuando como una capa superior del middleware SIX/TCL permita modelaren forma grafica procesos de negocios entre diferentes sistemas. El modelamiento incluye los flujos de los procesos y la data que será intercambiada entre los diferentes sistemas que conforman el proceso.

**C. Objetivo del Proyecto**

<b>Metas de la Organización</b>	<b>Objetivos del Proyecto</b>
Repotenciar el producto SIX/TCL	Diseñar la aplicación que soporte la definición de procesos de negocios
Reducir los tiempos de desarrollo de las aplicaciones de integración	Proporcionar una herramienta que permita al desarrollador diseñar en forma de un diagrama de flujos, los procesos de negocios involucrados dentro del proyecto de integración de sistemas, mediante el uso de una interface de usuario sencilla e intuitiva

**D. Alcance del Proyecto**

<p><b>Resultados del Proyecto.</b></p> <p>El resultado del proyecto será el desarrollo de un sistema de integración basada en procesos de negocios que opera sobre el middleware SIX/TCL. Mediante este sistema se podrá automatizar y manejar una secuencia estándar de eventos (un proceso de negocios) que ocurre entre 2 o más sistemas</p> <p><b>Contenido del Proyecto.</b></p> <ul style="list-style-type: none"> <li>- Fase de Fundamentación Teórica</li> <li>- Fase de Propuesta de Solución</li> <li>- Fase de Modelado del Negocio</li> <li>- Fase de Requerimientos</li> <li>- Fase de Administración de Proyecto</li> </ul> <p><b>Exclusiones.</b></p> <ul style="list-style-type: none"> <li>- Las tareas inherentes a la comunicación y manejo de protocolos de comunicación no son parte del proyecto</li> <li>- Las fases de construcción, certificación e implementación no forman parte del alcance del proyecto</li> </ul> <p><b>Stakeholders claves.</b></p> <ul style="list-style-type: none"> <li>- Gerente General</li> <li>- Gerente de Investigación y desarrollo</li> <li>- Consultor de negocio</li> </ul> <p><b>Hipótesis o Suposiciones.</b></p> <p>La versión del middleware base (SIX/TCL) se mantendrá en la 03.02.01 a lo largo del proyecto</p>
---

**Restricciones.**

- El diseño del producto se restringirá al uso del middleware SIX/TCL
- Se debe de cumplir los estándares de análisis y diseño de la metodología RUP.

**E. Factores Críticos de Éxito del Proyecto**

- Compromiso de la gerencia
- Gestión y seguimiento del proyecto con un control estricto del cumplimiento de los entregables.
- Colaboración plena de todo el personal involucrado en el proyecto.
- Contar con los recursos de hardware y software

**F. Planeamiento Inicial del Proyecto al alto nivel****Estimación de recursos requeridos :**

- 1 Gerente de proyecto
- 1 Analista funcional
- 1 Arquitecto de base de datos
- 1 programador Delphi
- 1 programador J2EE
- 1 programador ANSI C / C++
- Ambiente de trabajo para 6 personas que incluya muebles de oficina.
- 6 computadoras con acceso a Internet y aplicativos de oficina

**Costo Estimado del Proyecto:**

- El costo estimado del sistema utilizando la metodología COCOMO es de US\$ 38,219.00

**Beneficios Estimados:**

- Reducir el esfuerzo de codificación de aplicaciones para atender los requerimientos de integración de la empresa.
- Reducir el tiempo de puesta en producción de los proyectos de integración de sistemas.
- Independencia del proveedor del sistema, lo que se traduce en menores costos de mantenimiento. El personal de la propia empresa podrá realizar los cambios que se requieran en sus procesos de negocios.
- Menores costos de desarrollo de los proyectos, por la reducción en el uso de recursos de la empresa y del tiempo de desarrollo de los proyectos. Este ahorro puede ser trasladado al cliente dándole a la empresa una ventaja económica u otorgarle a la empresa una mayor rentabilidad en los proyectos.

- Mejorar la competitividad de la empresa al brindar la flexibilidad para realizar las modificaciones y adopciones de los procesos de negocio.
- Automatizar la interacción entre los sistemas de una empresa y sistemas exteriores a ella como sus proveedores.
- Mejorar el posicionamiento de Novatronic como empresa especializada en integración de sistemas, al poder ofrecer menores tiempos de desarrollo de proyectos a los clientes y flexibilidad en el mantenimiento de las soluciones.

**Estimación de fechas a programar:**

Fecha de inicio : 30/04/2008

Fecha de término: 10/09/2008

**G. Autoridad del Proyecto**

- **Autorización**

Gerente de Investigación y Desarrollo

- **Gerente del proyecto**

Nombre: Julio Zelada B.

- **Comité de Seguimiento (Dirección)**

- Gerencia General
- Gerencia Investigación y desarrollo
- Jefe de QA
- Gerente del proyecto

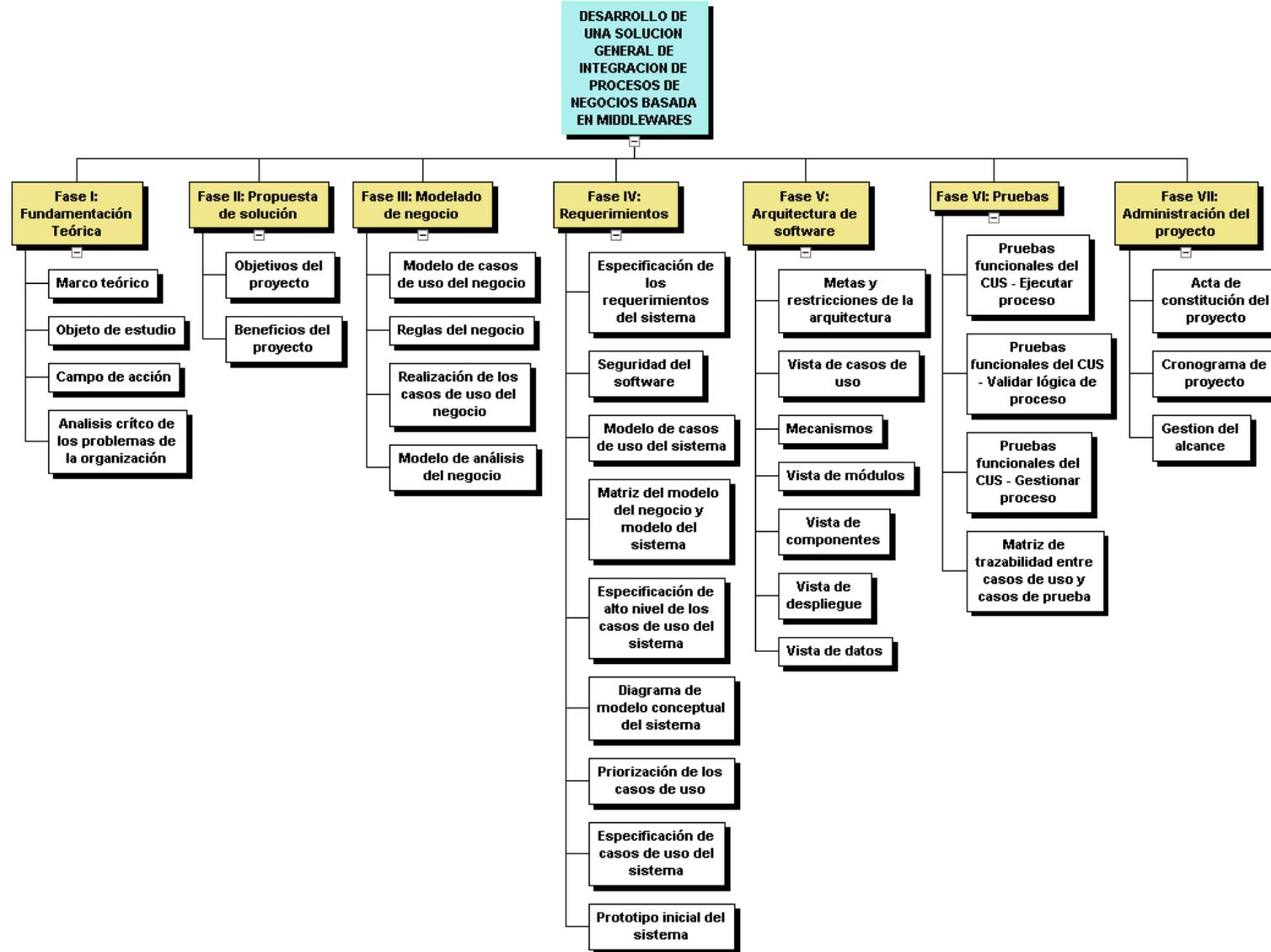
**H. Integrantes del equipo del proyecto, Roles y Responsabilidades**

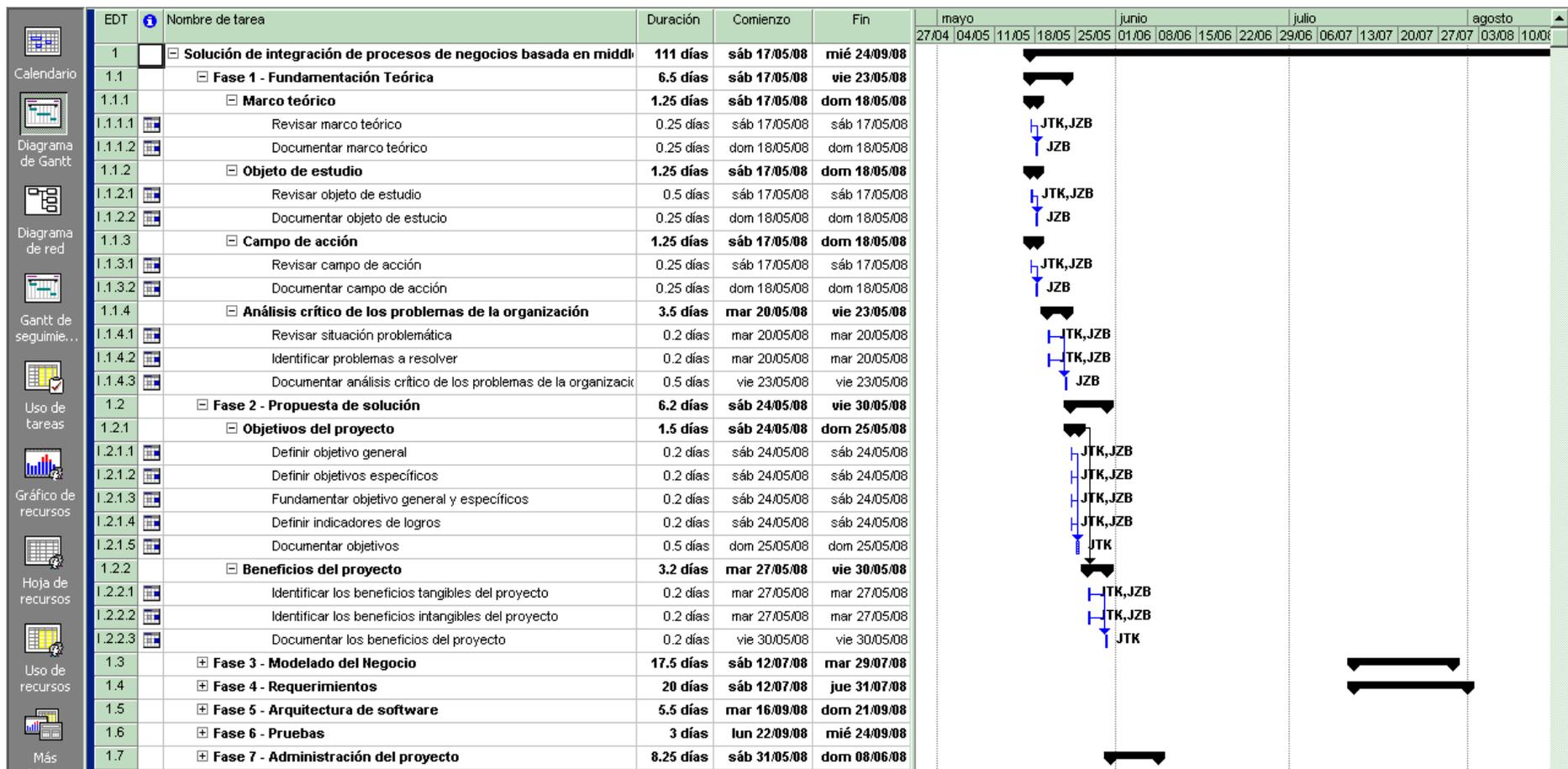
- Gerente de proyecto – Gestionar el proyecto
- Analistas funcionales – Realizar el análisis y diseño del sistema
- Arquitecto de Base de Datos – Modelar la base de datos
- Programadores – Encargados del desarrollo y pruebas unitarias de los módulos que conforman el sistema

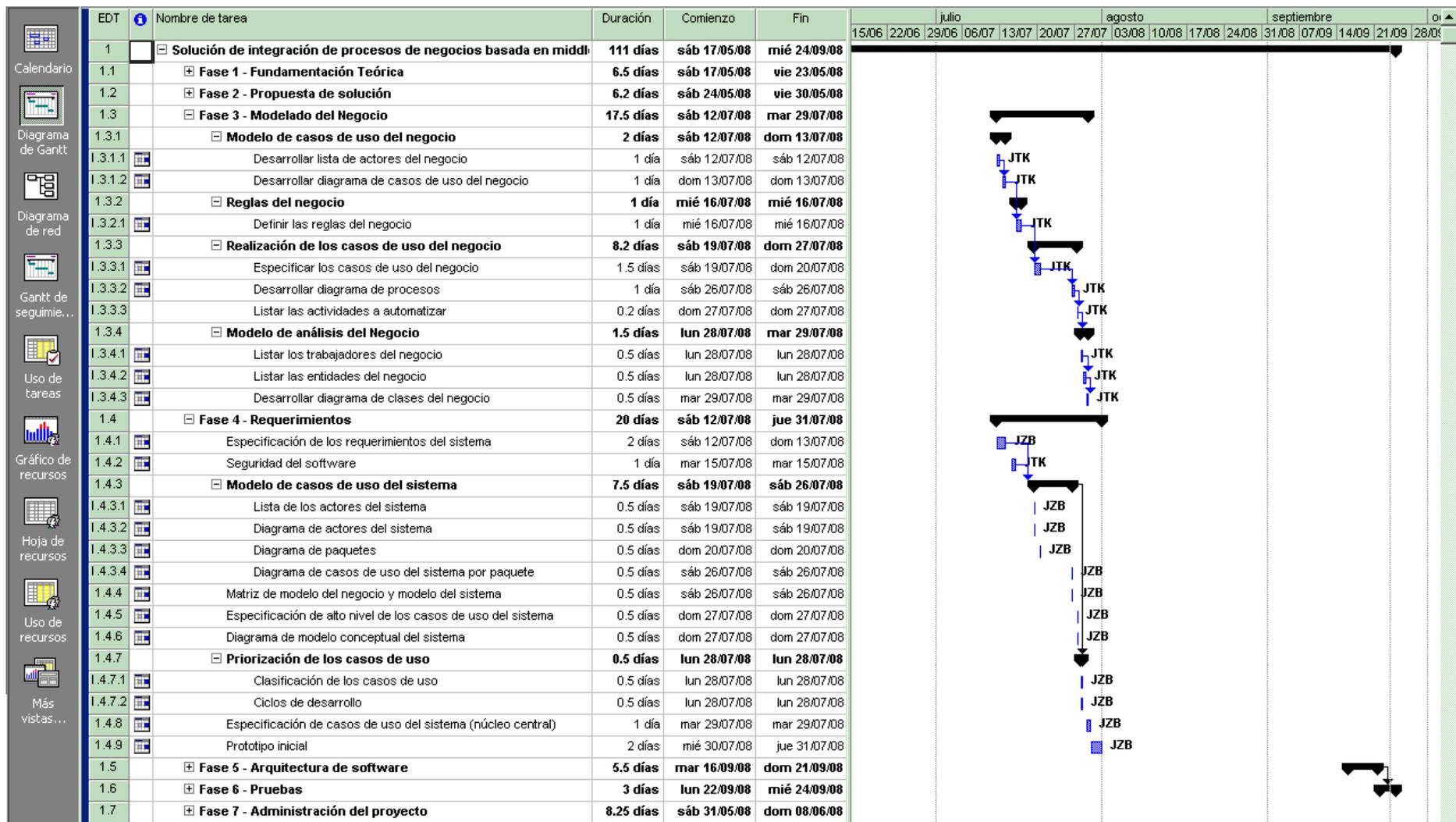
**I. Firmas**

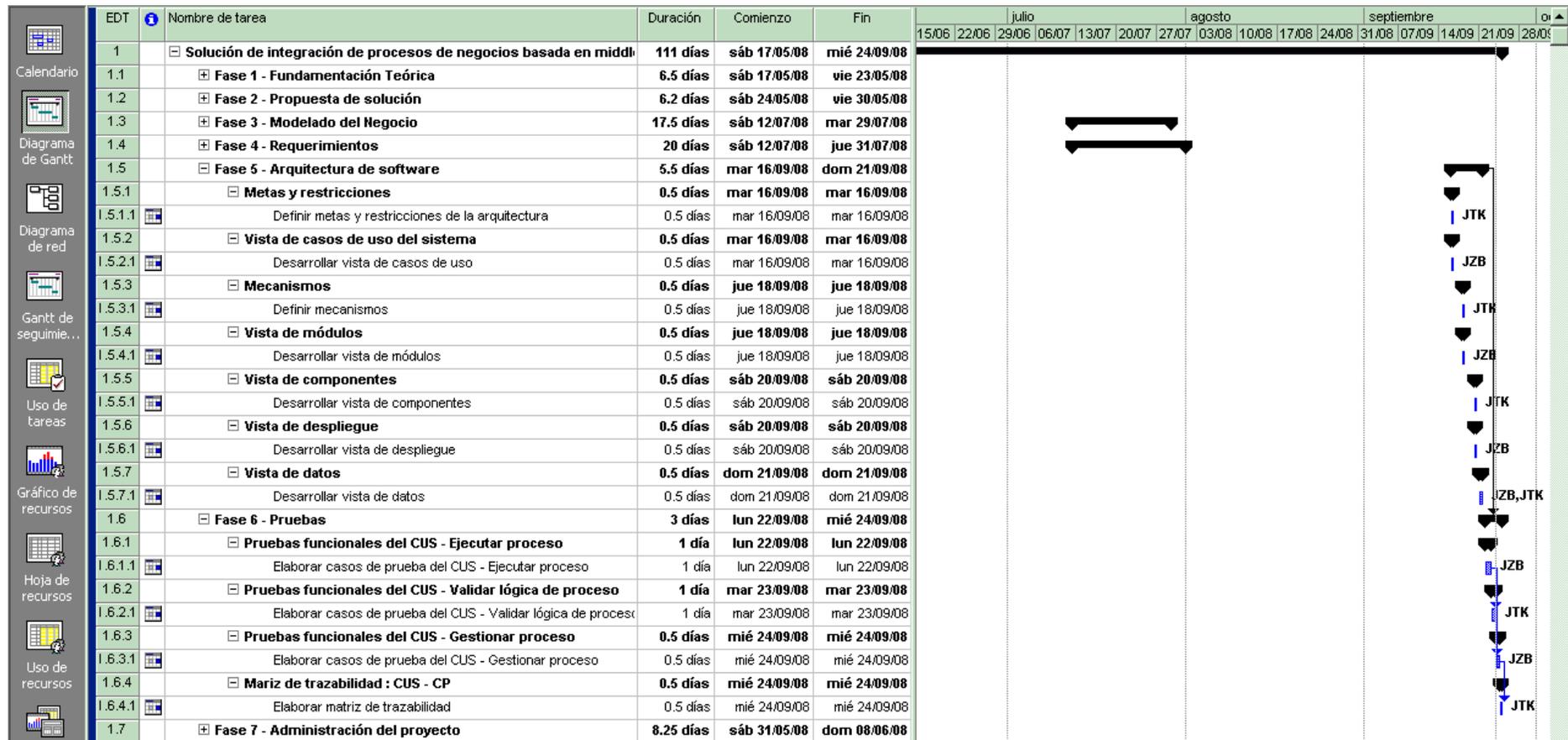
<b>Nombre/Función</b>	<b>Firma</b>	<b>Fecha</b>

**7.3. Cronograma de ejecución del proyecto**









**7.4. Gestión del alcance**

<b>Nombre del Proyecto:</b>	Solución general de integración de procesos de negocios basada en middlewares
<b>Preparado por:</b>	Toma K, Juan / Zelada, Julio
<b>Fecha:</b>	06/06/2008
<p><b>1. Describir cómo será administrado el alcance del Proyecto:</b></p> <p>Cualquier integrante del Comité de Seguimiento podrá solicitar un cambio en el alcance del proyecto, el cual será canalizado por el Gerente del Proyecto utilizando la plantilla de solicitud de cambios en el alcance.</p> <p>El Comité de Seguimiento será el responsable de aprobar las solicitudes de modificación a los entregables del producto. Los integrantes del Comité del Seguimiento son :</p> <ul style="list-style-type: none"> <li>• Gerente General</li> <li>• Gerente de investigación y desarrollo</li> <li>• Gerente del Proyecto</li> </ul>	
<p><b>2. Evaluar la estabilidad del alcance del proyecto (cómo manejar los cambios, la frecuencia e impacto de los mismos):</b></p> <p>a. El Gerente de Proyecto informará al Comité de Seguimiento sobre el impacto en el proyecto de las variaciones en el alcance, debido a los cambios solicitados. El impacto se informará en términos de variación del plazo, presupuesto o alcance.</p> <p>b. El Comité de Seguimiento de Proyecto se reunirá para medir los cambios o impacto en el proyecto y será responsable de aprobar o rechazar los cambios.</p> <p>c. Las reuniones se llevarán a cabo todos los viernes. Si no hubiera solicitudes de cambios o requerimientos, el Gerente de Proyecto lo comunicará al Comité de Seguimiento de Proyecto. Excepcionalmente algún miembro del Comité podrá solicitar la asistencia de otra persona que no pertenece al Comité para apoyar la justificación de un cambio.</p>	
<p><b>3. ¿Cómo los cambios al alcance, serán identificados y clasificados?</b></p> <p>a. Análisis Preliminar: El Gerente de Proyecto efectuará un análisis preliminar para calificar el cambio. Las calificaciones posibles son las siguientes:</p> <ul style="list-style-type: none"> <li>• Mejora, solo si el requerimiento está enmarcado en el alcance del proyecto y no afecta los costos y/o los cronogramas o su efecto es manejable a criterio del Gerente del Proyecto.</li> <li>• Adecuación, solo si el requerimiento está enmarcado en el alcance del proyecto afecta los costos y/o los cronogramas.</li> <li>• Modificación, solo si el requerimiento no está enmarcado en el contexto de los alcances definidos en el proyecto y/o afecta sustancialmente los costos y/o los cronogramas del proyecto.</li> </ul> <p>b. Asignar Responsable de Evaluación: El Gerente de Proyecto asignará a un miembro del equipo del proyecto la tarea de evaluar el cambio.</p> <p>c. Evaluar el Cambio: El responsable de la evaluación debe tener en cuenta el nivel de impacto, tiempo, riesgo, descripción de la solución y costo estimado (horas hombre).</p> <p>d. Declaración de Viabilidad: La solicitud de cambio se presenta al Comité de Seguimiento para su aprobación o rechazo.</p>	
<p><b>4. Describir cómo los cambios del alcance serán integrados al proyecto:</b></p> <p>Si el cambio es aprobado se deberán actualizar todos los entregables afectados. El responsable de la actualización de la Estructura Detallada de Trabajo (EDT) efectuará la actualización respectiva</p>	
<p><b>5. Comentarios adicionales:</b></p> <p>Ninguno</p>	

### **7.5. Conclusiones**

En este capítulo se ha definido el acta de constitución del proyecto y el enunciado del alcance del proyecto. También se ha establecido el plan del alcance el cual nos permitirá completar el proyecto satisfactoriamente.

### **CONCLUSIONES**

- El propósito de los capítulos expuestos ha sido presentar un alcance detallado de la solución informática que se plantea para la integración de procesos de negocios basada en un middleware.
- La solución que se propone es un complemento al middleware elegido como base de comunicación, no implica un rediseño del middleware sino que hace uso de las funcionalidades actuales del programa para la integración de diversos tipos de sistemas.
- El diseño de la base de datos donde se modelarán los procesos de negocios y los mecanismos de interpretación de dicho modelo son las tareas críticas de la solución que se propone
- La solución que se plantea podrá ser utilizada por aquellas empresas que necesiten integrar sistemas bajo plataforma Unix, Windows y Linux.
- Como consecuencia de la investigación realizada, se ha llegado a la conclusión de que el proyecto es económicamente viable y tecnológicamente factible.
- El trabajo propuesto tiene actualidad, pues se basa en los últimos conceptos en integración de negocios y además es un trabajo original en el ámbito nacional porque actualmente no existe un producto peruano de estas características

## **ANEXO 1**

### **TIPOS ESPECIALIZADOS DE MIDDLEWARE**

Este anexo presenta una lista de los tipos de middlewares que existen en el mercado, se pueden dividir en seis (6) categorías [MIDDLEW:2008:1]:

1. **Message Oriented Middleware.** Esta es una extensa categoría e incluye aplicaciones asincrónicas con capacidad de stored&forward así como brokers integradores que realizan transformación de mensajes y ruteo de eventos
2. **Object Middleware.** Basados en ORB (Object Request Brokers).
3. **RPC Middleware.** A diferencia de los MOM, RPC representa interacciones sincrónicas entre sistemas y es comúnmente usada dentro de una aplicación.
4. **Database Middleware.** Permite acceso directo a las estructuras de datos y provee interacción directa con las bases de datos. Existen numerosos gateways de base de datos y opciones de conectividad a las mismas. Los paquetes ETL (Extract, Transform, and Load) están incluidos en esta categoría.
5. **Transaction Middleware.** transaction processing monitors (TPM) y web application servers..

6. **Portals.** Se ha incluido a los Portales porque facilitan la integracion “front end”.  
Permiten la interaccion el “desktop” del usuario y los sistemas y servicios “back end”.

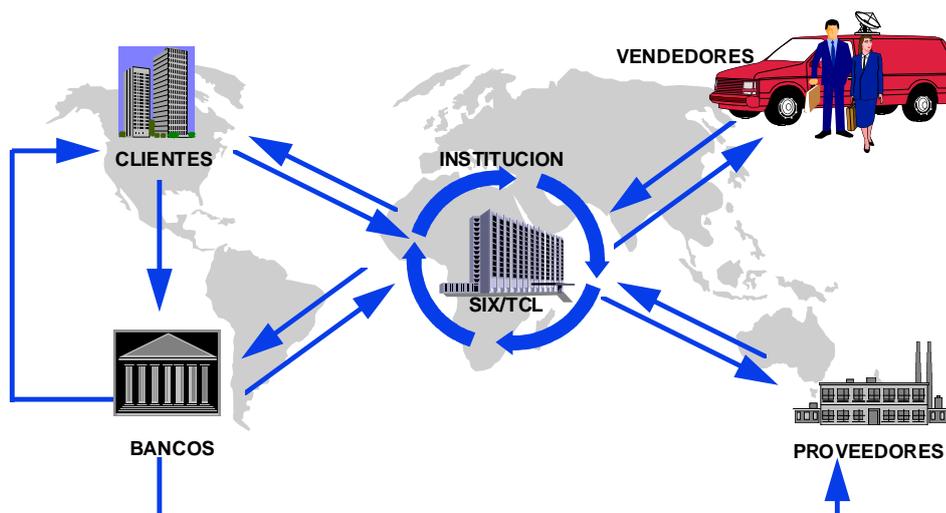
## **ANEXO 2**

### **DESCRIPCIÓN DEL MIDDLEWARE SIX/TCL**

Las empresas, día a día enfrentan el reto de mejorar y ampliar su gama de servicios, ya sea por la competencia, demanda de información de los usuarios o el cambiante entorno empresarial. Con la globalización de los mercados, el establecimiento y desarrollo de las super carreteras de información - tal es el caso de INTERNET e INFOVIA - y la mayor difusión del concepto de comercio electrónico, se ha hecho imprescindible contar con herramientas que permitan a las instituciones brindar servicios de acceso e intercambio de información, aprovechando las ventajas de la tecnología disponible.

En este sentido el servicio de teleacceso de clientes ha venido siendo utilizado por algunas instituciones para brindar a sus clientes corporativos acceso a su información, aceptar operaciones en línea, así como transferir e intercambiar información necesaria para la marcha normal de la empresa.

## SIX/TCL Y EL COMERCIO ELECTRONICO



Sin embargo, las soluciones que se tienen en la actualidad, necesitan de una renovación que les permita explotar los avances tecnológicos recientes, facilitando la modernización y ampliación de los nuevos servicios que se demandan.

**SIX/TCL** es un producto diseñado para satisfacer estos requerimientos. Su arquitectura, basada en el esquema Cliente/Servidor, facilita la rápida implantación de estos servicios y encapsula la complejidad técnica, permitiendo que los esfuerzos de la institución se concentren en definir la información que deberá desplegarse y el tipo de operaciones que podrán realizar sus clientes, proveedores y empleados.

**SIX/TCL** se encarga del manejo, monitoreo y administración de las comunicaciones, así como del control de los aspectos de seguridad, enrutamiento de las operaciones y atención de las solicitudes de ejecución de procesos remotos.

El empleo del **SIX/TCL** asegura a la institución una fácil migración a la nueva tecnología Cliente/Servidor y una rápida implementación de sus nuevos servicios, así como lograr bajos costos de inversión y una mejor relación de beneficio/costo.

### **¿Quiénes requieren SIX/TCL?**

El **SIX/TCL** cubre una gran variedad de aplicaciones en el ámbito empresarial, entre las cuales se pueden destacar :

- ❖ Las instituciones que desean brindar servicios de acceso remoto a su información, tal es el caso de Bancos, Compañías de seguros, Centrales de Información, Centrales de órdenes de pedido.
- ❖ Empresas que desean integrar sus operaciones con sus clientes y proveedores en un esquema de comercio electrónico interempresarial (EDI).
- ❖ Empresas que deseen establecer redes privadas de intercambio de información (transferencia electrónica de archivos), en la forma de mensajes, datos o imágenes.
- ❖ Cualquier otra institución que requiera proveer información, atender transacciones y/o intercambiar información (archivos) con sus clientes, proveedores, vendedores, representantes, así como su propio personal.

### **¿Qué beneficios brinda el SIX/TCL?**

El **SIX/TCL** permite que las instituciones que quieran implementar servicios de acceso e intercambio de información centren sus esfuerzos en los aspectos de negocio, encargándose el **SIX/TCL** por completo de resolver la complejidad técnica de este tipo de servicio. Sus funciones cubren los siguientes aspectos:

- ❖ Seguridad del acceso y cifrado de la información
- ❖ Disponibilidad del servicio
- ❖ Confiabilidad de las operaciones
- ❖ Eficiencia en el uso de los recursos de cómputo
- ❖ Posibilidad de crecimiento gradual del sistema (escalabilidad)
- ❖ Facilidad de implementación, de administración, y de desarrollo de nuevos servicios
- ❖ Cobertura geográfica ilimitada
- ❖ Bajos costos de inversión y de operación

### ¿Cómo opera el SIX/TCL?

El **SIX/TCL** opera en el esquema cliente/servidor y provee de una serie de funciones que permiten que el programa aplicativo cliente pueda solicitar una determinada gama de servicios (transferencia de archivos, transacciones, entre otros) al **SIX/TCL** server. Los servicios a ser brindados en el sistema son atendidos por programas servidores que pueden convivir con el **SIX/TCL** o residir en el host de la institución o en otros computadores, actuando el **SIX/TCL** como un gateway.

Tanto el acceso a los servicios como la disponibilidad de los servidores aplicativos están bajo el control del **SIX/TCL**, siendo transparente a la aplicación cliente la ubicación física del servidor aplicativo.

La arquitectura del **SIX/TCL** permite realizar transacciones en línea, intercambio de archivos, envío de jobs remotos, incluso la operación del sistema en modalidad off-host de no estar el computador principal disponible.

### **SIX/TCL Servidor**

Es el componente principal del **SIX/TCL**, se ejecuta en un equipo estándar o PC bajo el sistema operativo UNIX, lo que permite tener costos reducidos de equipamiento y crecer de acuerdo a los requerimientos del negocio. El **SIX/TCL Server** se encarga del monitoreo y control del sistema de comunicaciones, de las sesiones activas de los clientes, y de enrutar los requerimientos de servicios hacia los computadores correspondientes. Una de sus funciones principales es la de controlar el acceso mediante el requerimiento de identificación y validación de claves de las estaciones de los clientes. Asimismo el **SIX/TCL Server** provee la transparencia de los protocolos de comunicación entre las estaciones y el computador servidor destino; por ejemplo una estación puede estar comunicándose en protocolo asíncrono, mientras que el host que atiende los servicios solicitados puede estar comunicándose en protocolo SNA LU 6.2.

Esta independencia del protocolo de comunicaciones, asimismo, permite que el **SIX/TCL Server** pueda atender a un grupo de estaciones que utilizan comunicación asincrónica y a otras comunicadas bajo el protocolo SLIP TCP/IP o CDPD en forma independiente al funcionamiento de las aplicativos servidores.

### **SIX/TCL Cliente**

Las estaciones de trabajo, representan el lado del cliente en el esquema C/S del **SIX/TCL**. Es en la estación donde se generan: la llamada de conexión al **SIX/TCL Server**, el envío de una transacción y recepción de su respuesta correspondiente, el envío de un archivo o de un job, entre otros. Toda esta funcionalidad está encapsulada en la librería de funciones provista por el **SIX/TCL Cliente**, las cuales son accedidas mediante un API (interfase aplicativa) de fácil manejo. Esto permite que la institución o el responsable pueda

concentrarse en desarrollar el programa aplicativo del cliente sin tener que involucrarse en los aspectos de índole especializada. Las funciones provistas operan en forma transparente al lenguaje de programación bajo WINDOWS (Visual Basic, Delphi, C++, entre otros), permitiendo aprovechar las ventajas de un ambiente de programación visual y ofreciendo una funcionalidad variada, entre las que podemos destacar:

- La identificación de la estación mediante un botón electrónico de seguridad.
- La búsqueda automática de rutas alternantes para acceder al servidor incluso bajo diferentes protocolos. Por ejemplo un SIX/TCL Server puede atender diferentes huntings de líneas, e incluso protocolos diferentes, y las estaciones cliente pueden configurarse a fin de que automáticamente encuentren rutas alternas en el caso de que exista congestión o avería de línea. Incluso la búsqueda de rutas alternas puede ser definida a nivel de cada cliente/estación, estableciendo diferentes niveles de disponibilidad de servicio. Toda esta funcionalidad es totalmente transparente al programa aplicativo ya que son servicios provistos por el API del SIX/TCL cliente.
- La información es transmitida en forma cifrada bajo estándares internacionales lo que garantiza la máxima seguridad.
- Compresión de los datos a ser enviados
- Si durante la transferencia de información (archivos) se produce una interrupción, el sistema intenta recuperarse en forma automática y reinicia la transmisión desde el punto de corte, garantizando la integridad de los datos.
- Existe un log de actividad donde se deja un registro de lo actuado en el día por una determinada estación.

### **¿Cómo se invoca a las funciones del SIX/TCL ?**

La interfase aplicativa (API) provista permite que los programas aplicativos de la institución, escritos en algún lenguaje de entorno gráfico, tal como Visual Basic, Visual C++, Delphi, Power Builder, y otros que invoquen DLLs, puedan solicitar a través de llamadas sencillas los servicios del SIX/TCL Server.

Adicionalmente se tiene disponible la versión cliente del **SIX/TCL** para ambiente UNIX, Windows, MVS, VSE, AS/400, VMS, entre otros.

### **Plataforma tecnológica**

La configuración base del SIX/TCL requiere como mínimo de una Pentium IV con sistema operativo UNÍX o LINUX, con 128 MB de memoria y un disco duro de 2 GB. El SIX/TCL Servidor está disponible para plataformas UNÍX: AIX (RS/6000), DG/UX, Sun Solaris y Linux para versiones Red Hat y Suse.

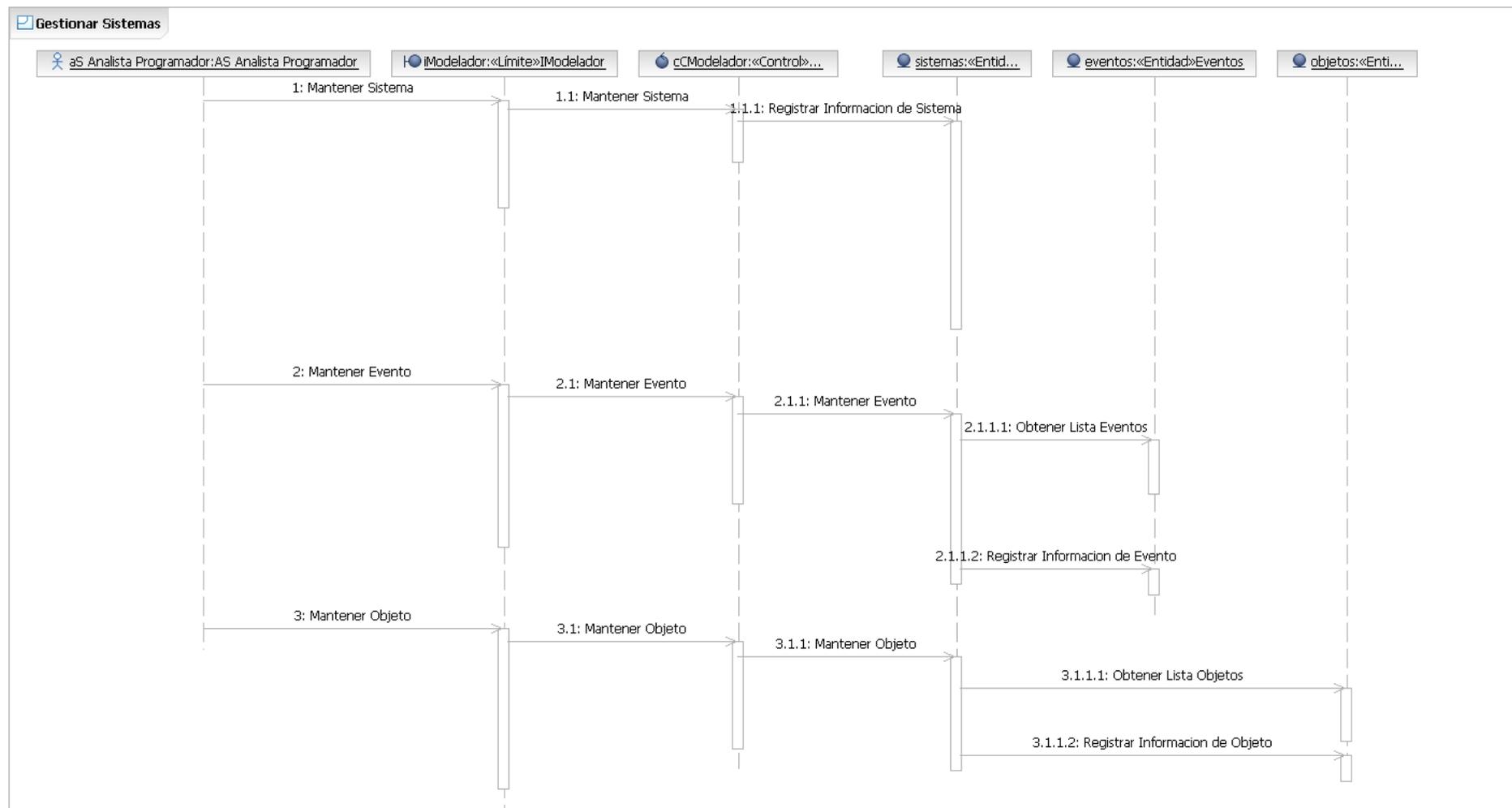
Las facilidades de comunicación variarán de acuerdo a los protocolos de comunicación requeridos, así como al número de estaciones clientes que se quieran operar y el nivel de actividad transaccional que se prevea.

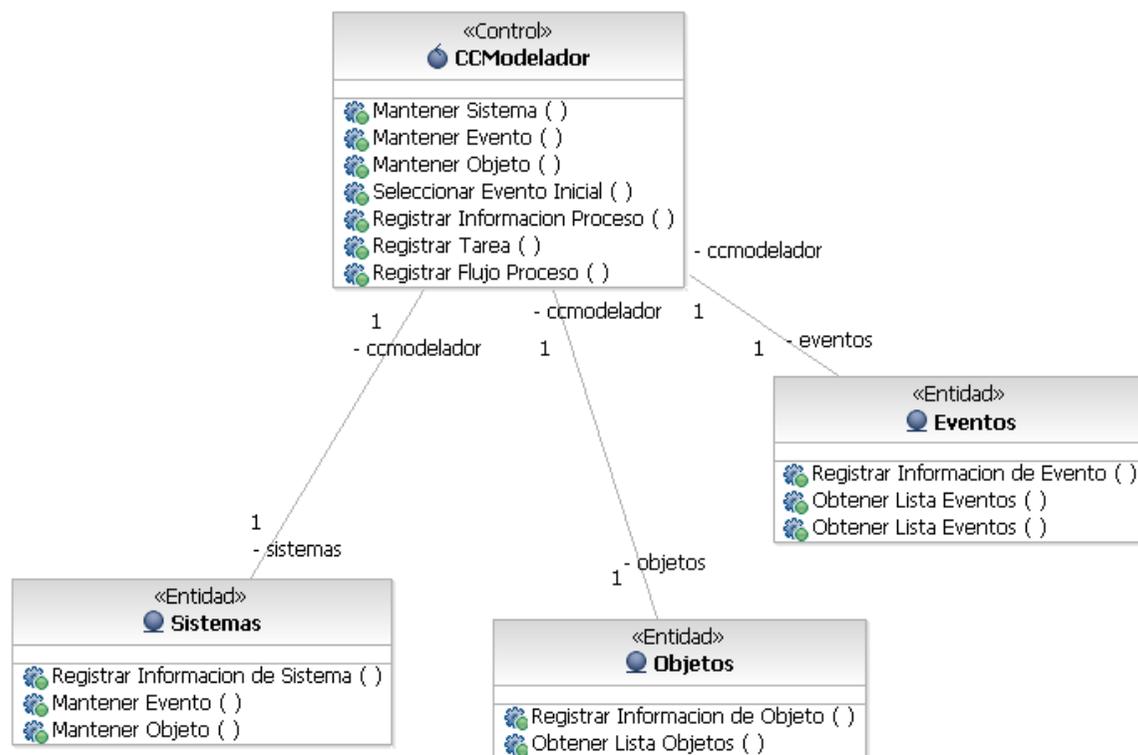
La configuración típica, considera el uso de una tarjeta multipuerto para las comunicaciones con los clientes o un Terminal Server. Para la conexión con el Host se requiere de una tarjeta de red o una tarjeta de comunicación sincrónica y el software base correspondiente.

**ANEXO 3**

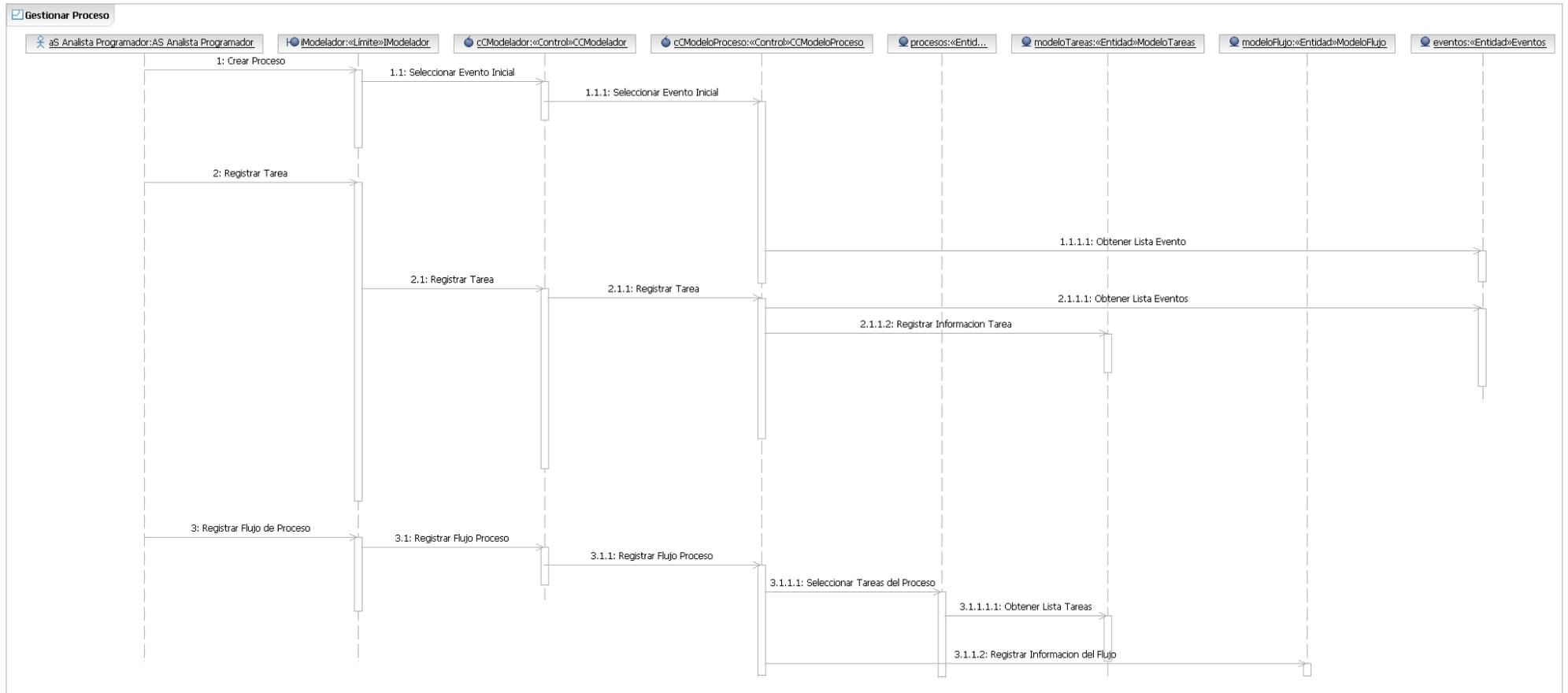
**ANÁLISIS**

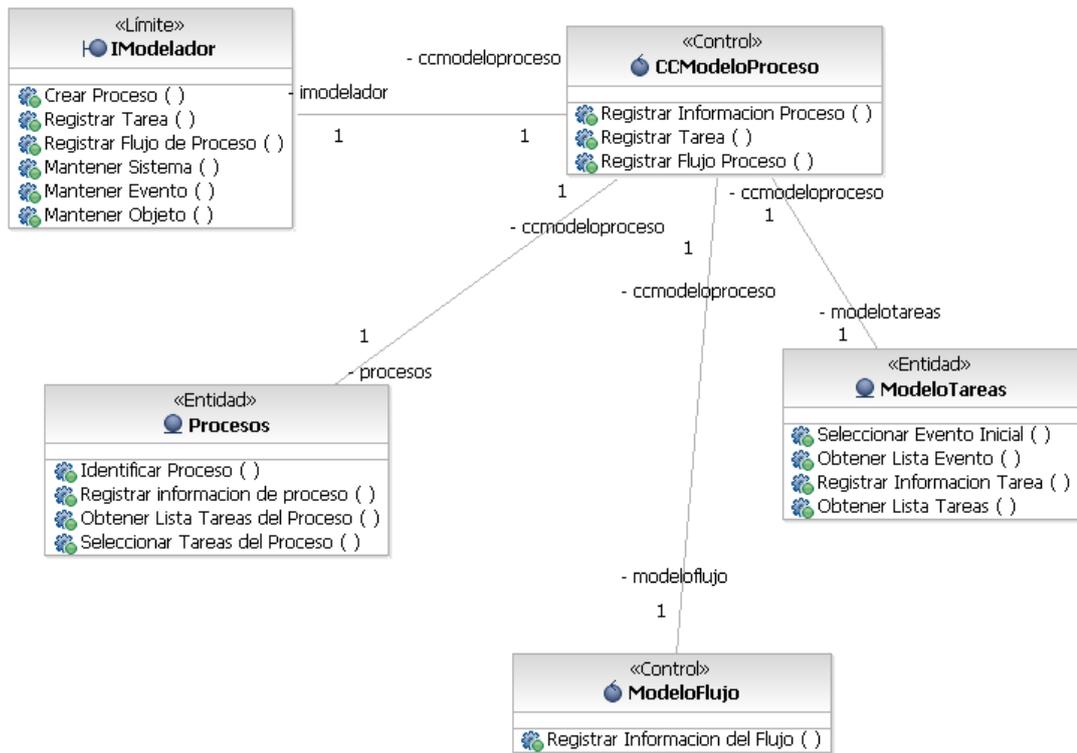
### DIAGRAMA DE SECUENCIA DEL CASO DE USO: GESTIONAR SISTEMA



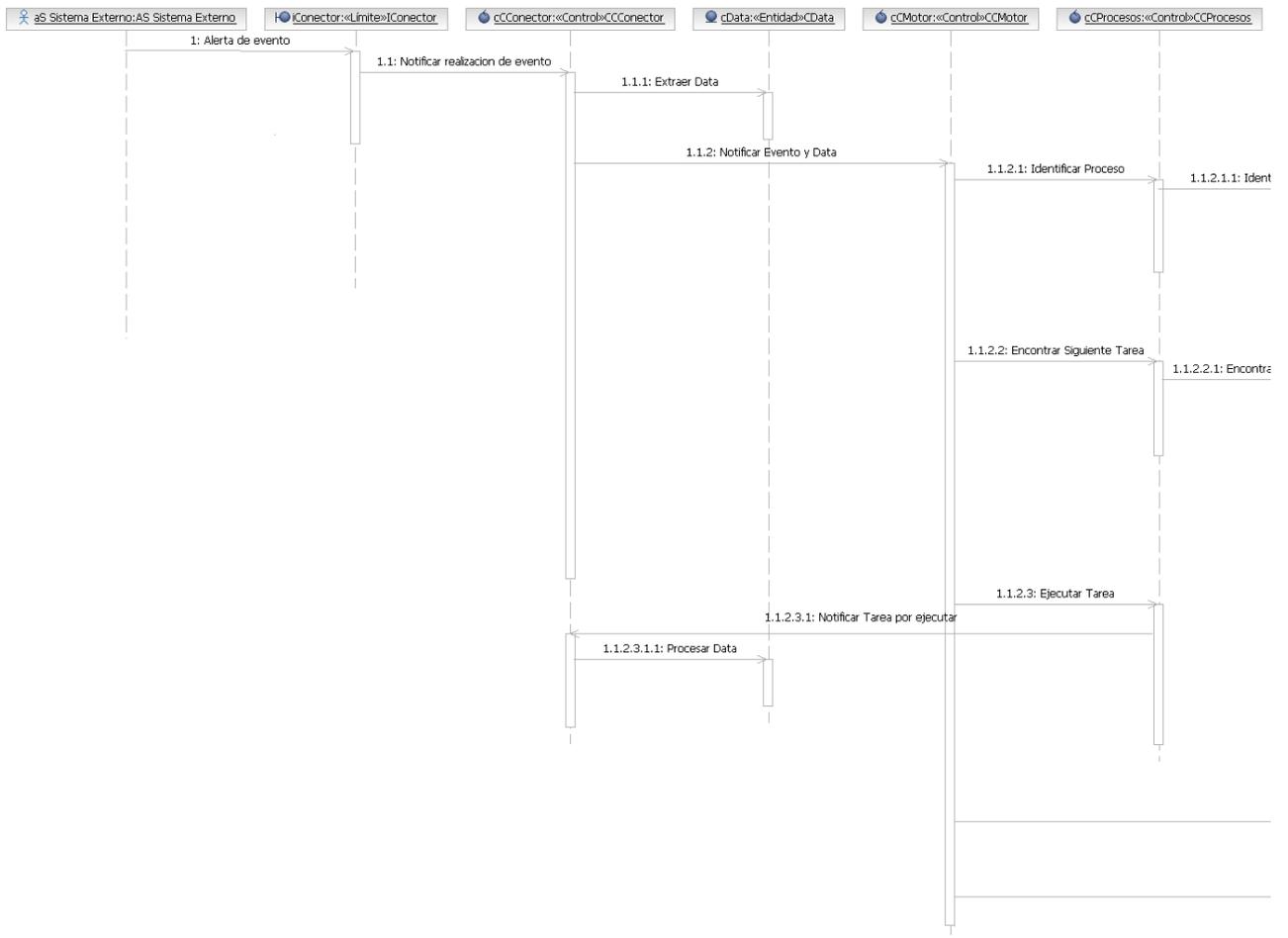


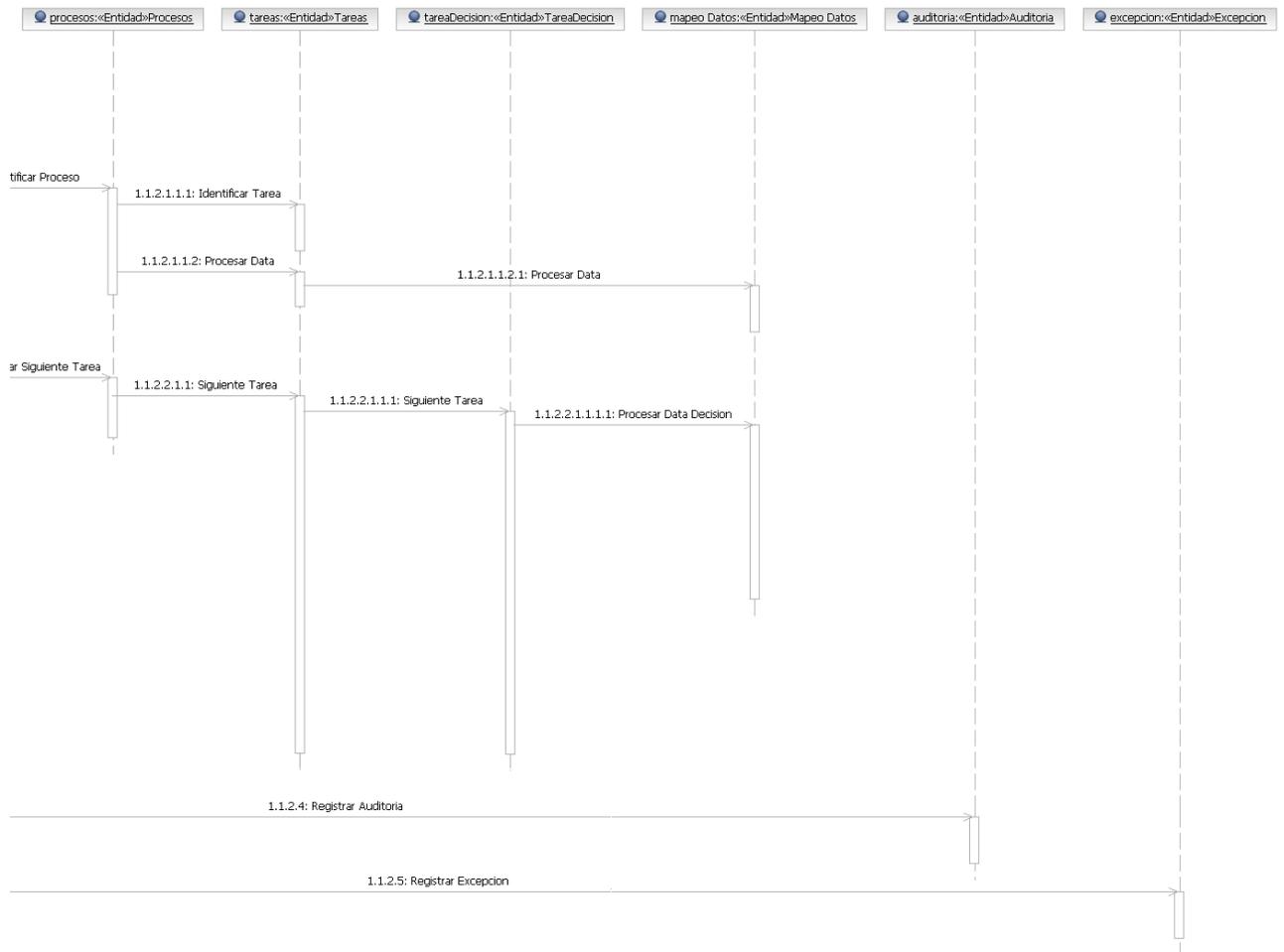
**DIAGRAMA DE SECUENCIA DEL CASO DE USO: GESTIONAR PROCESO**

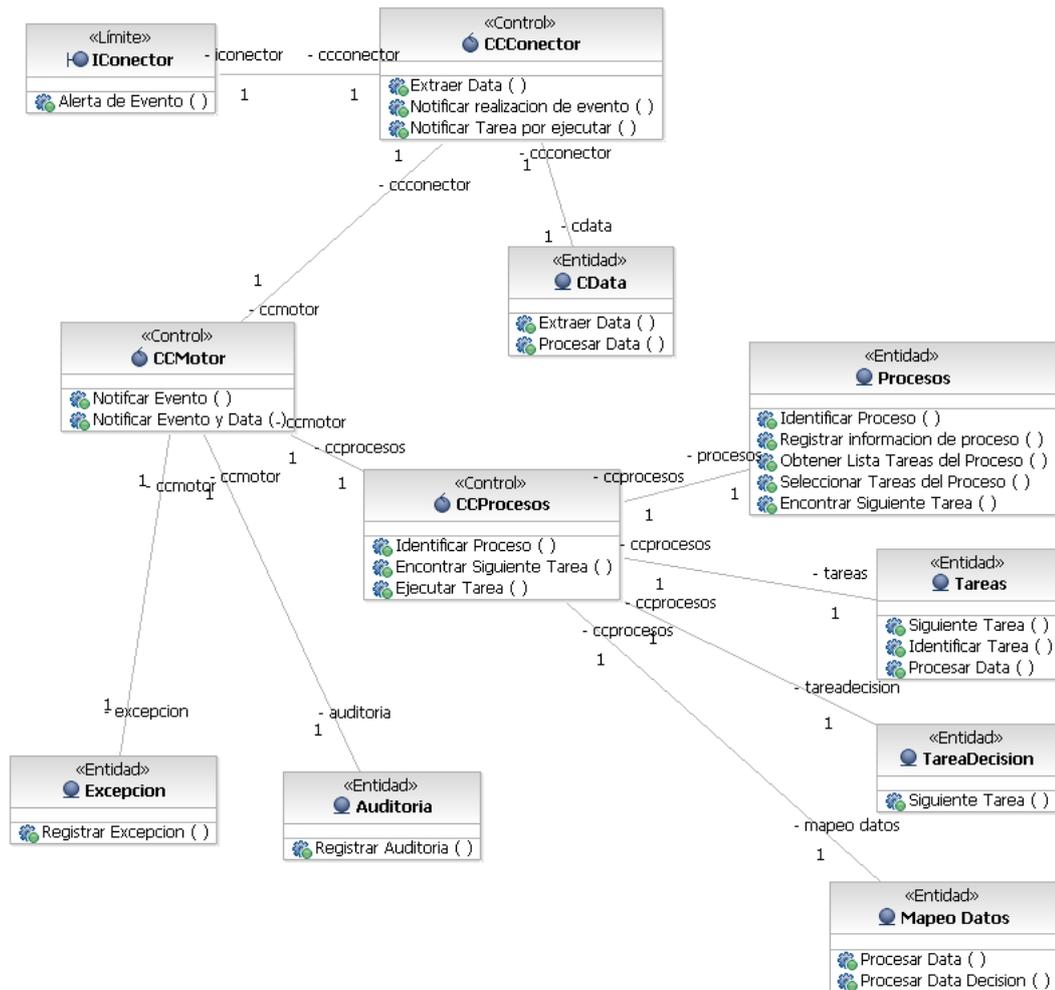




**DIAGRAMA DE SECUENCIA DEL CASO DE USO: EJECUTAR PROCESO**









## **GLOSARIO DE TÉRMINOS**

### **Aplicación Stand-Alone**

Aplicación que en tiempo de ejecución no tiene interfaz con el usuario.

### **CSV (Comma Separated Value)**

Archivo de texto donde los campos se separan por el carácter ‘,’.

### **DES**

Data Encryption Standard (DES) es un método de encriptación ampliamente utilizado, el cual utiliza una llave única tanto para el cifrado como para el descifrado de datos. Existen 72,000,000,000,000,000 posibilidades de llaves que se pueden generar. DES fue creado por IBM en 1977 y fue adoptado por el departamento de defensa de los Estados Unidos. Su uso está especificado en la norma ANSI X3.92

### **ANI**

Es un servicio que provee al receptor de una llamada telefónica el número del teléfono que le llama. El método de proveer esta información, depende del proveedor del servicio.

**Eventos y Objetos de negocio**

Los eventos de negocio son el resultado de actividades específicas que guían la ejecución de las tareas.

Los objetos de negocio son entidades lógicas, los cuales son manipulados en el proceso de ejecución de una función (tarea). Los objetos son definidos por sus atributos.

**IVR ( Interactive Voice Response)**

Es un software aplicativo encargado de dar respuesta por voz a través de sistemas telefónicos.

**MAC (Message Authentication Code)**

Es un código que se genera a partir de un mensaje de longitud arbitraria y de una clave secreta compartida entre remitente y destinatario, y que sirve para autenticar el mensaje.

El valor MAC del mensaje garantiza la integridad del mensaje (si el cálculo del valor MAC no coincide con el valor MAC enviado, el mensaje fue modificado) y lo autentica (el mensaje debe provenir del remitente, pues es el único que conoce la clave privada). A diferencia de un algoritmo de cifrado, el algoritmo de cálculo de un valor MAC no necesita ser reversible.

**Mapeo de datos**

Para integrar las aplicaciones, los datos deben ser pasados desde las aplicaciones fuentes hacia las aplicaciones destinos. La aplicación destino debe reconocer y entender los datos que está recibiendo de la aplicación fuente. Por ejemplo, mientras el sistema A registra un usuario utilizando su nombre, apellido paterno y apellido materno, el sistema B solo

requiere el nombre, por lo tanto los campos que identifican al usuario en el sistema A deben ser “transformados” (mapeados) para que el sistema B lo entienda.

### **Proceso asíncrono**

Es aquel proceso en el que la aplicación que envió el mensaje procede con su propio procesamiento sin esperar una respuesta. La aplicación que recibe el mensaje no necesita construir una respuesta para la aplicación origen.

### **Proceso síncrono**

Es aquel proceso en el que la aplicación que envió el mensaje queda esperando una respuesta del receptor del mensaje antes de continuar el procesamiento.

### **Proceso de negocio**

Un proceso de negocio define y describe las diferentes formas en que las actividades son ejecutadas en una organización.

### **Proceso de negocio distribuido**

Tradicionalmente, las organizaciones se dividen en departamentos. Las tareas dentro de estos departamentos están automatizadas con algún software aplicativo. Sin embargo muchos procesos de negocios trascienden estos departamentos.

Aunque ciertas tareas son ejecutadas dentro de una sola aplicación, existen otras que son ejecutadas en diferentes departamentos. Mediante el uso del sistema que se propone, un proceso de negocio distribuido se convierte en un proceso donde diferentes tareas son completadas en forma conjunta a través de diferentes aplicaciones.

**Reglas de Negocio**

Una regla de negocio encapsula una política de la empresa y define la secuencia de las tareas a ejecutar para completar un proceso de negocio.

**Solución intrusiva**

Se define como una solución intrusiva a aquella en la que la aplicación que se desea integrar debe ser modificada para añadirle las funcionalidades de comunicación con el middleware. Esta modificación implica una recompilación de la aplicación

**Solución no Intrusiva**

Una solución intrusiva es aquella en la que la aplicación que se desea integrar no necesita ser modificada. El intercambio de información entre la aplicación y el middleware se hace a través del almacén de datos de la aplicación (base de datos o archivos).

**Tareas**

Un proceso de negocios consta de un conjunto de funciones de negocios. Estas funciones se dividen en una serie de pasos (unidades de proceso) llamadas Tareas, las cuales describen como estas funciones de negocios son realizadas.

**Tupla**

Las tuplas se emplean para describir objetos matemáticos que tienen estructura, es decir que son capaces de ser descompuestos en un cierto número de componentes. Por ejemplo, un Grafo dirigido se puede definir como una tupla de  $(V, E)$  donde  $V$  es el conjunto de nodos y  $E$  es el subconjunto de  $V \times V$  que denota los vértices del grafo.

### **REFERENCIAS BIBLIOGRÁFICAS**

[FOLDOC:2008:1]

Free OnLine Dictionary of Computing.  
<http://foldoc.org/>

[MIDDLEW:2008:1]

Pagina oficial del Middleware Resource Center  
<http://www.middleware.org/whatis.html>

## **BIBLIOGRAFÍA**

- 1 Process Innovation, Boston, U.S.A.  
Thomas H. Davenport
- 2 1999 - Professional Java Server Programming, Birmingham, Inglaterra.  
Varios. Wrox Press Ltd.
- 3 1999- Microsoft Visual C++ Programación avanzada en WIN32  
Francisco Javier Ceballos  
Editorial Rama
- 4 La importancia de un Middleware Robusto y escalable en las soluciones empresariales  
cliente/servidor.  
Marzo 1996 Revista Sistemas  
Universidad de los Andes - Colombia

### Direcciones Web consultadas

Tibco, USA.  
Microsoft Corporation, USA  
Object Management Group

[www.tibco.com](http://www.tibco.com)  
[www.microsoft.com](http://www.microsoft.com)  
[www.cgi.omg.org](http://www.cgi.omg.org)