



Universidad de las Ciencias Informáticas
Facultad 3

Título: Análisis y Diseño del Proceso de Administración y Control de la Información para la Sección Sindical Vicerrectoría Primera.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

AUTORES: Ana Lis Sánchez Freyre
Isel Campo Valls

TUTOR: Ing. Geiser Arcio Pérez Rivas

Ciudad de la Habana, junio de 2009.

Declaración de Autoría

Declaramos ser las únicas autoras de este trabajo y autorizamos a la Sección Sindical Vicerrectoría Primera de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ___ días del mes de Junio del año 2009.

Autor: Ana Lis Sánchez Freyre.

Autor: Isel Campo Valls.

Tutor: Ing. Geiser Arecio Pérez Rivas.

Dedicatoria

A mi hermana Ivet, por ser la luz de mi vida.

A mi papá por su infinito amor y comprensión.

A mi mami linda por su fe inquebrantable y por darme tanto amor.

A mi Buey, por ser tan linda y especial.

A todos mis amigos, por darme tanto apoyo.

Lis

A mi mamá y hermano, por ser ambos tan especiales.

A mis vecinos, familia y amigos, por su infinito apoyo.

A las personas que han sido importantes en mi vida y que ya no se encuentran físicamente presente.

Lis

Agradecimientos

En especial a mis padres, que me han dado el apoyo, el amor y la fe para poder llegar hasta aquí.

A mi familia, por ser tan unida y especial.

A mi hermana Ivet, que es lo más importante en mi vida.

A mi Buelly, por su infinito amor.

A mis amigos Dary, Lis, May, Yoa, Noi, TPT, Zaily, Dayris, Eve, Yandy, Yami, uds forman parte de mí, una parte muy importante... los quiero, de veras que sí.

A mi novio René, por darme tanto amor y tener tanta paciencia.

A mi tutor, a la Vicerrectoría Primera y a todas aquellas personas que contribuyeron de una forma u otra a que este sueño se hiciera realidad.

A Dios, por estar siempre conmigo.

150

A mi mamá por su preocupación y amor y por ser lo más grande para mí.

A mi hermano Maykell por su apoyo y ayuda.

A toda mi extensa familia, a Félix Mario Marrero por su ayuda, a mi familia de amigos por soportarme.

A mis profesores de la UCI y profesores de las anteriores etapas escolares por las que he pasado.

A nuestro tutor por asignarnos este tema.

Lis.

Resumen

Actualmente la Dirección de la Sección Sindical Vicerrectoría Primera se ha trazado nuevos objetivos para mejorar el funcionamiento de sus nueve comités sindicales.

Esta organización maneja constantemente grandes volúmenes de información vitales para su funcionamiento. La documentación de esta no se ha digitalizado, lo cual incide en la imposibilidad de una recuperación inmediata y eficaz de la misma, haciendo más difícil el proceso de control de la información generada.

La presente investigación tiene como objetivo el análisis y diseño de un sistema automatizado como punto de partida para su posterior implementación, viabilizando toda la información referida a los procesos de administración y control que se llevan a cabo en esta organización.

Para cumplimentar el propósito de esta investigación, a partir de la indagación con personal dirigente de esta organización se pudo delimitar y jerarquizar las problemáticas de mayor relevancia y frecuencia que incidían en la labor eficiente de esta. Se realizó un estudio de los trabajos existentes similares a dicha investigación, estudiándose con detalle las herramientas, lenguajes y metodologías a utilizar. Se investigó sobre la Ingeniería de requisitos y las técnicas y clasificaciones propuestas por esta, así como las técnicas y pasos del diseño de sistemas, describiéndose previamente los roles de analista y diseñador de sistemas. Se definió el modelo de negocio de las actividades que se llevan a cabo actualmente en la organización y se obtuvieron los requisitos que debe cumplir el sistema así como el modelamiento del mismo, generando todos los artefactos pertenecientes al análisis y diseño de sistemas. La validación de los resultados se realizó mediante diferentes métricas midiendo de manera eficiente la calidad de los artefactos obtenidos durante el desarrollo de esta investigación.

Palabras claves:

Análisis, Diseño, Sección Sindical, Proceso de Administración de la información, Proceso de Control de la información.

Tabla de Contenidos

DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA	III
AGRADECIMIENTOS.....	IV
RESUMEN	V
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS.....	XI
INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	6
<i>Introducción del Capítulo 1.....</i>	<i>6</i>
1.1. ¿Qué es un Sistema de Gestión de Contenidos?	6
1.1.1. Sistemas Existentes.	6
1.2. Ingeniería de Software.....	7
1.3. Metodologías para el desarrollo de software	8
1.3.1. El Proceso Unificado de Desarrollo de Software.	9
1.3.2. Extreme Programming	11
1.3.3. Feature -Driven Development	13
1.4. Lenguajes de Modelado	14
1.4.1. Lenguaje de Modelado Unificado (UML).....	14
1.4.2. BPMN.....	17
1.4.3. IDEF0	18
1.5. Herramientas CASE.	19
1.5.1. Rational Rose.	20
1.5.2. Visual Paradigm.....	20
1.6. Lenguajes de Programación	22
1.6.1. Java.....	22
1.6.2. C#	23
1.7. Sistemas Gestores de Bases de Datos (SGBD)	23
1.7.1. PostgreSQL.....	24
1.7.2. MySQL	25
1.8. Patrones de Caso de Uso.....	26

1.9.	Patrones de diseño	27
1.10.	Descripción de los roles de Analista y Diseñador del Sistema.....	30
1.11.	Ingeniería de Requisitos	31
1.11.1.	Clasificación de los requisitos:	32
1.11.2.	Técnicas de obtención de requisitos	32
1.11.3.	Actividades de la ingeniería de requisitos.	34
1.12.	Diseño de Sistemas	35
1.13.	Métodos de validación de los resultados.....	36
	<i>Conclusiones Parciales del Capítulo 1</i>	38
CAPÍTULO 2.	MODELAMIENTO DEL NEGOCIO	40
	<i>Introducción del Capítulo 2</i>	40
2.1.	Modelo de negocio	40
2.1.1.	Descripción actual de la organización.	40
2.1.2.	Reglas del Negocio	41
2.1.3.	Actores del negocio.....	41
2.1.4.	Diagrama de Casos de Uso del Negocio	43
2.1.5.	Descripción de los Casos de Usos del Negocio	44
2.1.6.	Diagrama de Objetos.....	47
2.1.7.	Diagrama de Actividades	49
2.2.	Especificación de los requisitos de software.....	53
2.2.1.	Requisitos funcionales.....	53
2.2.2.	Requerimientos no funcionales	57
	<i>Conclusiones Parciales del Capítulo 2</i>	58
CAPÍTULO 3.	ANÁLISIS Y DISEÑO DEL SISTEMA	60
	<i>Introducción del Capítulo 3</i>	60
3.1.	Definición de los actores	60
3.2.	Diagrama de casos de uso del sistema.....	61
3.3.	Descripción de los casos de uso del sistema.	62
3.4.	Modelo de clases de análisis	73
3.4.1.	Diagramas de clases del análisis.	73
3.4.2.	Diagramas de interacción (colaboración).....	75

3.5. Modelo de diseño	82
3.5.1. Patrones de diseño.....	83
3.5.2. Diagrama de clases del diseño.....	83
3.5.3. Diagrama de interacciones (Secuencia).....	87
<i>Conclusiones Parciales del Capítulo 3.....</i>	<i>91</i>
CAPÍTULO 4. VALIDACIÓN DE LOS RESULTADOS	92
<i>Introducción del Capítulo 4.....</i>	<i>92</i>
4.1. Métrica de la Calidad de Especificación de los Requisitos.	92
4.2. Métrica para determinar el número de requisitos que no son considerados en ningún CU.....	92
4.3. Métricas Orientadas a Clases.	93
4.3.1. Tamaño de clases	93
4.3.2. Relaciones entre clases (RC).....	96
<i>Conclusiones Parciales del Capítulo 4.....</i>	<i>99</i>
CONCLUSIONES.....	100
RECOMENDACIONES	101
BIBLIOGRAFÍA.....	102
ANEXOS	105
ANEXO # 1 DESCRIPCIÓN TEXTUAL DE LOS CU DEL SISTEMA	105
ANEXO # 2 DIAGRAMAS DE CLASES DEL ANÁLISIS DE LOS CU DEL SISTEMA.....	127
ANEXO # 3 DIAGRAMAS DE COLABORACIÓN.....	132
ANEXO # 4 DIAGRAMA DE CLASES DEL DISEÑO	144
ANEXO # 5 DIAGRAMAS DE SECUENCIA	151
ANEXO # 6 MATRIZ DE QDF.....	169
ANEXO # 7 PROTOTIPOS DE INTERFAZ DE USUARIO.	172
ANEXO # 8 AVALES DE PROTOTIPOS.	183
GLOSARIO DE TÉRMINOS:	187

Índice de Figuras

Figura 1-1 Logo del Sindicato Médico	6
Figura 1-2 La ingeniería de software, tecnología multicapas	7
Figura 1-3 Fases y flujos de RUP	10
Figura 1-4 Fases de la Programación extrema.....	13
Figura 1-5 Vocabulario de U.M.L. (Jacobson, 2000).	15
Figura 2-1 Descripción de Casos de Uso del Negocio	44
Figura 2-2 Diagrama de Objetos Realizar Emulación Individual	47
Figura 2-3 Diagrama de Objetos Realizar Emulación por Comité Sindical.....	48
Figura 2-4 Diagrama de Objetos Recibir Información de Afiliado.	48
Figura 2-5 Diagrama de Objetos Recibir Información del Comité Sindical.	49
Figura 2-6 Diagrama de Actividades Recibir Información de Afiliado.....	50
Figura 2-7 Diagrama de actividades Realizar Emulación Individual.....	51
Figura 2-8 Diagrama de Actividades Recibir Información del Comité Sindical.....	52
Figura 2-9 Diagrama de Actividades Realizar Emulación por Comité Sindical.	53
Figura 3-1 Diagrama de casos de uso del sistema.	61
Figura 3-2 DCA del CU Autenticar Usuario	74
Figura 3-3 DCA del CU Gestionar noticia.....	74
Figura 3-4 DCA del CU Gestionar usuario.....	74
Figura 3-5 DCA del CU Gestionar afiliado más integral.....	75
Figura 3-6 DCA del CU Gestionar c/s más integral.....	75
Figura 3-7 Diag. Interacción del CU Autenticar usuario	76
Figura 3-8 Diag. Interacción del CU Adicionar noticia.....	76
Figura 3-9 Diag. Interacción del CU Eliminar noticia.....	76
Figura 3-10 Diag. Interacción del CU Modificar noticia	77
Figura 3-11 Diag. Interacción del CU Mostrar Noticia.	77
Figura 3-12 Diag. Interacción del CU Adicionar Usuario.	78
Figura 3-13 Diag. Interacción del CU Eliminar Usuario.	78
Figura 3-14 Diag. Interacción del CU Modificar Usuario.	78
Figura 3-15 Diag. Interacción del CU Mostrar Usuario.	79
Figura 3-16 Diag. Interacción del CU Adicionar c/s más integral.....	79

Figura 3-17 Diag. Interacción del CU Eliminar c/s más integral.....	80
Figura 3-18 Diag. Interacción del CU Modificar c/s más integral.	80
Figura 3-19 Diag. Interacción del CU Mostrar c/s más integral.....	81
Figura 3-20 Diag. Interacción del CU Adicionar afiliado más integral.....	81
Figura 3-21 Diag. Interacción del CU Eliminar afiliado más integral.....	81
Figura 3-22 Diag. Interacción del CU DC Modificar afiliado más integral.	82
Figura 3-23 Diag. Interacción del CU Mostrar afiliado más integral.....	82
Figura 3-24 DCD del CU Autenticar Usuario.....	84
Figura 3-25 DCD del CU Gestionar noticia.....	84
Figura 3-26 Diag. Interacción del CU Gestionar afiliado más integral	85
Figura 3-27 DCD del CU Gestionar comité sindical más integral.....	86
Figura 3-28 DCD del CU Gestionar usuario.....	87
Figura 3-29 DS del CU Eliminar noticia.....	88
Figura 3-30 DS del CU Adicionar noticia.....	89
Figura 3-31 DS del CU Modificar noticia	90
Figura 3-32 DS del CU Mostrar noticia	91

Índice de Tablas

Tabla 2-1: Justificación de los actores del negocio	42
Tabla 2-2: Justificación de los Trabajadores del Negocio.....	42
Tabla 2-3: Descripción del Caso de Uso del Negocio Recibir información de afiliado.	44
Tabla 2-4: Descripción del Caso de Uso del Negocio Recibir información del Comité Sindical.....	45
Tabla 2-5: Descripción del Caso de Uso del Negocio Realizar Emulación Individual.	46
Tabla 2-6: Descripción del Caso de Uso del Negocio Realizar Emulación por Comité Sindical.	46
Tabla 3-1: Actores del Sistema	60
Tabla 4-1: Valores de los umbrales para TC.	93
Tabla 4-2: Tamaño de las Clases.	93
Tabla 4-3: Cantidad de clases por clasificación.	96
Tabla 4-4: Resultados de la Métrica TC.	96
Tabla 4-5: Cantidad de relaciones de uso entre las clases	96
Tabla 4-6: Acoplamiento	98
Tabla 4-7: Cantidad de pruebas y complejidad de mantenimiento.....	98
Tabla 4-8: Reutilización	98

Introducción

La informática es una de las ciencias más jóvenes que está en constante desarrollo, esta se encuentra presente en prácticamente todas las esferas del mundo actual. La automatización es imprescindible en las organizaciones para lograr una mayor eficiencia en los procesos, para facilitar el trabajo a todo el personal implicado, y para lograr su correcto y óptimo funcionamiento.

En la Universidad de las Ciencias Informáticas (UCI) como en todos los centros educativos y de trabajo existe el movimiento sindical, representado por la Sección de Trabajadores de Cuba, organización de masas constituida por todos los sindicatos y sus afiliados, en la que todos sus esfuerzos están centrados principalmente en cumplir cabalmente con las tareas encomendadas por el Comandante Fidel.

La Sección Sindical Vicerrectoría Primera está dirigida por un secretario general y cuatro miembros que atienden los frentes de finanzas, emulación, asuntos laborales y sociales, cultura, deporte y recreación. Asimismo esta Sección Sindical esta compuesta por 460 afiliados, separados por comités sindicales (dónde el número de comités varía debido a la inestabilidad estructural existente), con un secretario general y activistas para los principales frentes.

En la Sección Sindical Vicerrectoría Primera el trabajo relacionado con la gestión del personal que integran los comités sindicales existentes se realiza de forma manual, por lo cual no existe suficiente tiempo para entregar toda la documentación requerida. Los documentos e información que se manipulan se archivan en formato duro o en computadoras de bajas prestaciones y en diferentes formatos. No existe un espacio donde publicar noticias y avisos para los sindicalistas. El personal trabajador que no tiene cuenta de correo, carece de una vía de información sobre las fechas de las reuniones, actividades y tareas que se realizan. La estructura es muy inestable, pues se producen constantes cambios en el personal de las secciones sindicales y los afiliados no cuentan con una aplicación mediante la cual se informen dichos cambios de forma más eficiente y rápida que el correo electrónico. Dada la gran cantidad de información a consultar es muy difícil llevar a cabo la emulación, causando insatisfacción en las personas responsables de esta tarea. El control de la asistencia a las actividades se toma en formato duro, el cual en la mayoría de las ocasiones no es digitalizado y guardado para su seguimiento, de

esta forma no se puede ofrecer la opción a los sindicalistas de saber en cuales actividades ha participado y su estado en relación con otros trabajadores.

El proceso de dar altas y bajas a los sindicalistas es muy complejo y no esta automatizado. No existe un medio informático adecuado donde sea posible evaluar el desempeño de los afiliados, ni un mecanismo para saber si se le dio seguimiento a los acuerdos y tareas pactadas en las reuniones.

El tema relacionado con deporte, cultura y recreación en dicha Sección Sindical no se desarrolla satisfactoriamente, puesto que se dificulta la gestión de los participantes en estas actividades. Los trabajadores no conocen la fecha ni los horarios de los juegos deportivos, y como se reestructura su día laboral debido a estos.

Las fechas significativas durante el año a menudo pasan desapercibidas por la mayoría de los sindicalistas ya que no se organizan actividades para su celebración, en esto influye la carencia de un medio informativo adecuado.

Por todo lo anteriormente expuesto se plantea como **problema científico** a resolver:

¿Cómo transformar las necesidades de los afiliados de la Sección Sindical de la Vicerrectoría Primera a un lenguaje entendible por los desarrolladores que facilite su posterior automatización?

Dado lo anteriormente expuesto la **hipótesis** se ha definido de la siguiente forma:

Si se realiza un correcto análisis y diseño del proceso de administración y control de la información, entonces se podrán transformar las necesidades de los afiliados a un lenguaje entendible por los desarrolladores.

Por consiguiente el **objeto de estudio** del presente trabajo de diploma será el Proceso de Desarrollo de Software.

El **campo de acción** en el cual se enmarca este trabajo es la modelación del análisis y diseño de los procesos de administración y control de la información de la Sección Sindical Vicerrectoría Primera.

El **objetivo general** es realizar el análisis y diseño del proceso de administración y control de la información para la Sección Sindical de la Vicerrectoría Primera.

Como **tareas a desarrollar** asociadas al objetivo general se tienen:

- Analizar bibliografía sobre el tema.
- Realizar entrevistas para determinar la situación problemática.
- Definir las metodologías, herramientas y lenguajes de modelado a utilizar.
- Procesar y analizar los resultados obtenidos.
- Modelar del Negocio.
- Especificar de los Requisitos del software.
- Modelar el Sistema.
- Realizar Análisis y Diseño.

Este trabajo se sustenta en los siguientes métodos de investigación:

Métodos Teóricos

- Analítico – Sintético: Permitió el análisis de documentos, teorías, acerca del proceso de desarrollo de software y la extracción de los elementos más importantes relacionados a este proceso.
- Inducción – Deducción: Permitió que a partir de un estudio profundo de los procesos de desarrollo de software arribar a una propuesta de proceso de desarrollo de software para alcanzar el objetivo de este trabajo.
- Histórico – Lógico: Se utilizó para constatar teóricamente como ha evolucionado el proceso de desarrollo de software con el objetivo de obtener información necesaria sobre este proceso.

Métodos Empíricos

- Observación: Permitió percibir los problemas existentes en la Sección Sindical Vicerrectoría Primera.
- Entrevista: Permitió la exploración en los sindicalistas de como se lleva a cabo los procesos de administración y control de la información, a fin de evaluar que problemas existen.

- Encuesta: Se utilizó para obtener información sobre los problemas que existen en la Sección Sindical Vicerrectoría Primera.

El presente trabajo se estructura en cuatro capítulos:

Capítulo 1: Fundamentación Teórica.

En este capítulo se hace un estudio del estado del arte sobre el proceso de desarrollo de software. Se hace un análisis a otros trabajos existentes semejantes al trabajo en desarrollo, se muestran los principales conceptos manipulados en el transcurso de la investigación, se hace referencia a las metodologías de desarrollo, los lenguajes de modelado y herramientas, seleccionando las que serán utilizadas en el desarrollo del presente trabajo. Se resalta la importancia que tiene la ingeniería de requisitos para obtener productos que satisfagan las necesidades de los clientes y se hace énfasis en el papel desarrollado por el analista y diseñador de sistemas durante todo el proceso.

Capítulo 2: Modelamiento del Negocio.

Este capítulo cuenta con una amplia descripción del negocio, presentando los diagramas y casos de uso correspondientes a dicho modelado de negocio para de esta manera identificar las mejoras potenciales en la organización, y comprender el contexto donde será implantado el sistema. Luego se capturan los requisitos funcionales y no funcionales para la realización del modelo de casos de uso del sistema.

Capítulo 3: Análisis y Diseño del Sistema.

En este capítulo se refinan y estructuran los requisitos obtenidos profundizando en el dominio de la aplicación, lo que permitirá una mayor comprensión del problema para modelar la solución. Se realiza el modelado del diseño para lograr una arquitectura estable y sólida, de manera que contribuya a crear la base de lo que será implementado posteriormente.

Capítulo 4: Validación de los Resultados.

En este capítulo se expone el proceso de la validación de los resultados obtenidos durante el desarrollo del trabajo, evaluando mediante mediciones de calidad la especificación de los requisitos y del diseño realizado.

Capítulo 1. Fundamentación teórica.

Introducción del Capítulo 1.

En este capítulo se hace referencia al concepto de Sistema de Gestión de Contenidos, así como ejemplificar en la actualidad las tendencias de este tipo de sistemas a nivel mundial. Se hace un análisis de los trabajos existentes que son similares al trabajo a desarrollar, se dan a conocer elementos del proceso de desarrollo de software, se realiza un análisis comparativo de las principales metodologías existentes, además de hacer un estudio detallado de las herramientas CASE, los lenguajes de modelado, sistemas gestores de base de datos y lenguajes de programación, escogiendo cuales serán utilizados durante el desarrollo de este trabajo.

1.1. ¿Qué es un Sistema de Gestión de Contenidos?

Un Sistema de gestión de contenidos (Content Management System en inglés, abreviado CMS) es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. (2009)

1.1.1. Sistemas Existentes.



Figura 1-1 Logo del Sindicato Médico

El Sindicato Medico del Uruguay es un sistema donde se publica información para los afiliados, como documentos, resoluciones, efemérides, comunicados, estatutos, etc. Cumple con diversas funcionalidades como la de gestionar información gremial, socio-cultural y científica para médicos. (2009.Sindicato Médico del Uruguay) <http://www.smu.org.uy>

En Cuba existen sitios web informativos de organizaciones políticas y de masas como el Partido Comunista Cubano y la Central de Trabajadores de Cuba. Portales Web como el de la Federación Estudiantil Universitaria y la Unión de Jóvenes Comunistas constituyen en la UCI sitios de obligatoria consulta por constituir un espacio de referencia a la historia, noticias de última hora, reglamentos, estatutos, planes de trabajo, actividades y eventos a realizar en el ámbito universitario. Estos portales brindan algunas funcionalidades como la descarga de archivos de interés para reuniones y ofrecen el formato en que deben ser entregadas las actas, pero no permiten gestionar la información de los sindicalistas, o mostrar los acuerdos tomados en las reuniones, el seguimiento de las actividades y el desarrollo de la emulación.

En la Vicerrectoría Primera no existe ningún sistema que facilite la gestión de la información o brinde que brinde algunas de las funcionalidades antes mencionadas.

1.2. Ingeniería de Software.

"La Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos" (Cota, 2005)

La ingeniería de software se define según Pressman como una tecnología multicapa en la que se pueden identificar: los métodos, el proceso y las herramientas.



Figura 1-2 La ingeniería de software, tecnología multicapas.

Los métodos de la ingeniería del software indican como construir técnicamente el software, las herramientas de la ingeniería del software proporcionan un enfoque automático o semi-automático para el proceso y para los métodos. Cuando se integran

herramientas para que la información creada pueda ser utilizada por otra, se establece un sistema de soporte para el desarrollo del software llamado Ingeniería del Software Asistida por Computador (CASE). (Pressman, 2005)

1.3. Metodologías para el desarrollo de software

A medida que fue evolucionando el desarrollo de la Informática y la exigencia de los clientes fue cada vez mayor, los especialistas se dieron a la tarea de desarrollar diferentes metodologías para organizar el trabajo, basándose en métodos comunes para el desarrollo y diseñadas bajo un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para lograr la calidad, que es el principal objetivo estratégico en las organizaciones. Para el desarrollo del sistema informático propuesto, se realizó un estudio de las metodologías pesadas (tradicionales) y metodologías ligeras (ágiles).

Existen dos grupos dentro de los cuales se pueden clasificar las metodologías:

Metodologías tradicionales, en las cuales lo principal es el control del proceso, a través de una planificación exhaustiva, donde se controlan las actividades que se realizarán, los artefactos que se generarán, además de las herramientas y notaciones que serán usadas. Se caracterizan por comenzar con la obtención y análisis de los requerimientos solicitados por el usuario, y luego se definen los requerimientos funcionales y no funcionales del futuro sistema. Están basadas en la producción de proyectos de larga duración, lo que permite estructurar un amplio equipo de trabajo en roles que cumplirían diferentes funciones dentro de la producción. Dentro de estas metodologías se encuentra Rational Unified Process (RUP), la cual es muy usada en la actualidad pues goza de excelente prestigio dentro del mundo informático por los éxitos alcanzados.

Metodologías ágiles: surgidas por la incapacidad que tienen las tradicionales de operar en condiciones volátiles, se caracterizan por una gran reducción del tiempo de desarrollo del software, planteando que es más importante la producción de un software funcional que procesar una documentación excesiva. Además no se sigue un plan estricto por lo que se posee una alta capacidad de respuesta a un cambio.

Todo esto las hace más flexibles que las anteriores. El cliente es un trabajador más del equipo de trabajo, pues colabora de forma ininterrumpida con la realización del software.

Estas metodologías han presentado éxito en proyectos de poco alcance, donde el equipo de desarrollo no necesita ser amplio y se requiere la entrega inmediata del producto. Una metodología visible aquí es Extreme Programming (XP). (2009)

1.3.1. El Proceso Unificado de Desarrollo de Software.

El Proceso Unificado de Desarrollo (RUP) es el resultado de tres décadas de desarrollo y trabajo práctico. Transitó desde el Proceso Objectory con su primera publicación en 1987 hasta el Proceso Objectory del Rational en 1997, para luego en 1998, tomar el nombre de Proceso Unificado de Rational. Ofrece un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, es más que un simple proceso: es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. (Jacobson, 2000)

Esta metodología es usada en proyectos de gran envergadura que impliquen elevados tiempos de desarrollo y que posean un equipo de desarrollo grande.

RUP se basa en componentes y como aspectos definitorios se señala que es dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental.

Dirigido por casos de uso. Los casos de uso guían el proceso de desarrollo pues los modelos que se obtienen representan la realización de los mismos.

Centrado en arquitectura. La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo.

Iterativo e Incremental. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla algunos más que otros. En el caso de una iteración de la fase elaboración, se centra la atención en el análisis y diseño, a la vez que se refinan los requerimientos y se obtiene un producto con un determinado nivel, que irá creciendo incrementalmente en cada iteración.

RUP se describe normalmente desde tres perspectivas:

1. Una perspectiva dinámica que muestra las fases del modelo sobre el tiempo, las cuales son:
 - Inicio o Conceptualización: es donde se determina la visión del proyecto, o sea se comprende el entorno y se determina el alcance del producto.
 - Elaboración: en esta etapa se determinan los cimientos de la arquitectura y se analiza el dominio del problema.
 - Construcción: en esta fase se obtiene la capacidad operacional inicial del producto.
 - Transición: se obtiene el release o liberación del producto y se pone en manos de los usuarios finales.

2. Una perspectiva estática que muestra los flujos de trabajo del proceso. Existen seis flujos de trabajo de proceso y tres flujos de trabajo de soporte.

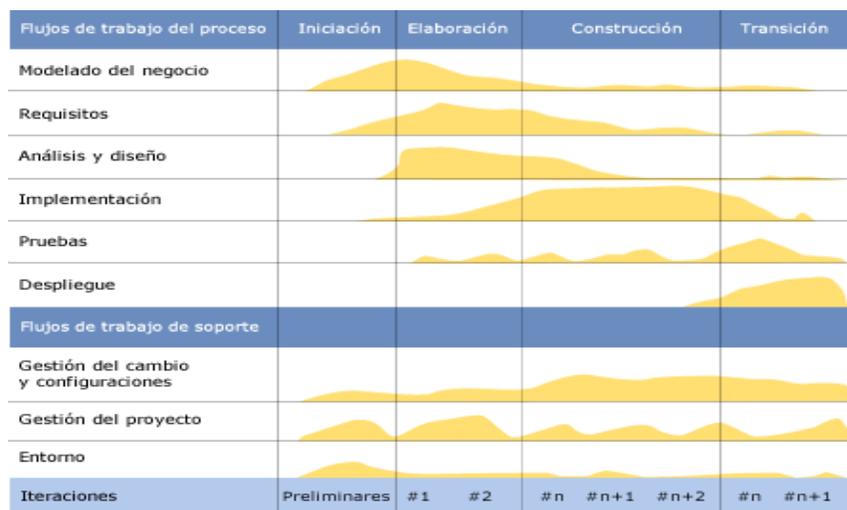


Figura 1-3 Fases y flujos de RUP

3. Una perspectiva práctica, que sugiere 6 buenas prácticas a utilizar durante el proceso.
 - Desarrolle el software de forma iterativa. Planifique incrementos del sistema basado en las prioridades del usuario y desarrolle y entregue las características del sistema de más alta prioridad al inicio del proceso de desarrollo.
 - Gestione los requerimientos. Documente explícitamente los requerimientos del cliente y manténgase al tanto de los cambios de estos requerimientos. Analice el impacto de los cambios en el sistema antes de aceptarlos.

- Utilice arquitecturas basadas en componentes. Estructure la arquitectura del sistema en componentes.
- Modele el software visualmente. Utilice modelos gráficos UML para presentar vistas estáticas y dinámicas del software.
- Verifique la calidad del software. Asegure que el software cumple los estándares de calidad organizacionales.
- Controle los cambios del software. Gestione los cambios del software usando un sistema de gestión de cambios y procedimientos y herramientas de gestión de configuraciones. (Rational Software Corporation, 2003)

1.3.2. Extreme Programming

Extreme Programming es una metodología ágil para equipos de desarrollo de software pequeños o medianos, que pone mayor énfasis en la adaptabilidad que en la previsibilidad. Esta centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Le da prioridad a las tareas que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación. Es adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y responsabilidad a la hora de afrontar los cambios, además de en:

- Pruebas Unitarias: son pruebas realizadas a los principales procesos, de manera que podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- Re fabricación: es la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa. (Marquetti, 1997)

X.P destaca cuatro variables para cualquier proyecto software: costo, tiempo, calidad, y alcance. Sus objetivos son la satisfacción del cliente, dándole al cliente el software que necesita y cuando lo necesita, y el desarrollo al máximo del trabajo en equipo, estando involucrados en el desarrollo del software tanto clientes como desarrolladores. (Calabria, 2003)

Prácticas básicas de la programación extrema. (Marquetti, 1997)

1. Equipo completo: forman parte del equipo todas aquellas personas que tienen algo que ver con el proyecto.
2. Planificación: se planifica en qué orden se van a hacer las cosas y las mini-versiones.
3. Test del cliente: el cliente propone sus propias pruebas con el objetivo de evaluar las mini-versiones.
4. Versiones pequeñas: las versiones deben ser pequeñas y deben ofrecer algo útil al usuario final.
5. Diseño simple: mantener siempre sencillo el código.
6. Pareja de programadores: los programadores trabajan por parejas y éstas se intercambian con frecuencia.
7. Desarrollo guiado por las pruebas automáticas: se deben realizar programas de prueba automática.
8. Mejora del diseño: extraer funcionalidades comunes y eliminar líneas de código innecesarias.
9. Integración continua: las funcionalidades deben de recompilarse y probarse a medida que sean desarrolladas.
10. El código es de todos: cualquiera puede y debe tocar y conocer cualquier parte del código.
11. Normas de codificación: debe haber un estilo común de codificación.
12. Metáforas: buscar frases o nombres que definan cómo funcionan las distintas partes del programa.
13. Ritmo sostenible: se debe trabajar a un ritmo que se pueda mantener durante todo el proceso de desarrollo del software.

Características fundamentales de la programación extrema:

1. Desarrollo iterativo e incremental.
2. Pruebas unitarias continuas, frecuentemente repetidas y automatizadas.
3. Programación en parejas.
4. Frecuente interacción del equipo de programación con el cliente o usuario.
5. Corrección de todos los errores antes de añadir una nueva funcionalidad.
6. Propiedad del código compartido.
7. Simplicidad en el código.

Fases de la programación extrema:

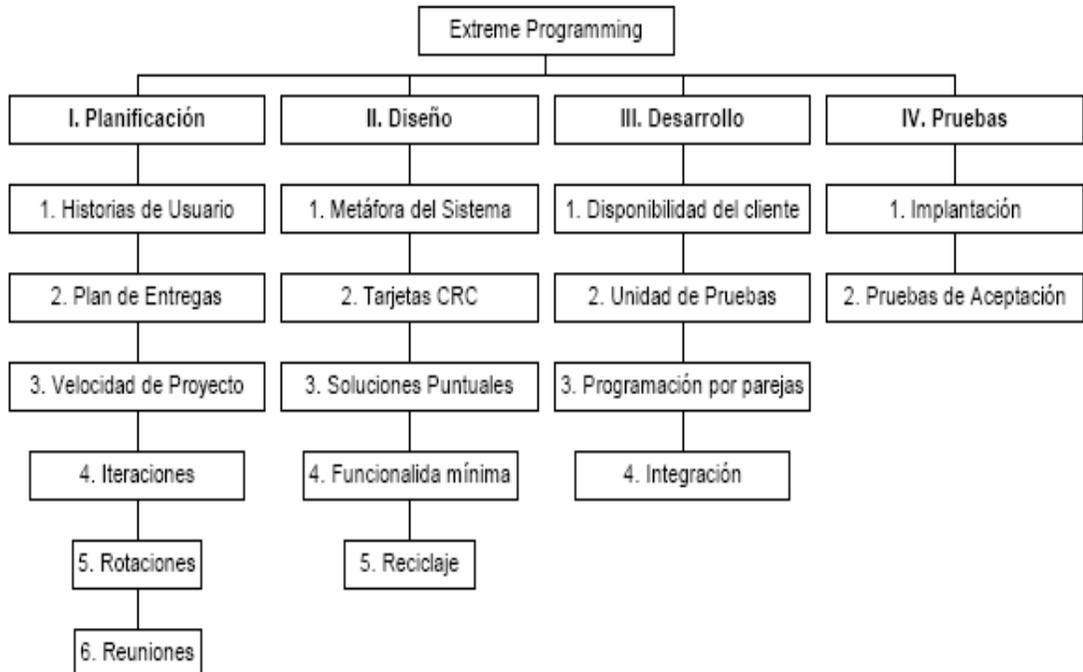


Figura 1-4 Fases de la Programación extrema.

1.3.3. Feature -Driven Development

F.D.D o Desarrollo Basado en Funcionalidades es un enfoque ágil para el desarrollo de sistemas. No hace énfasis en la obtención de los requerimientos sino en como se realizan las fases de diseño y construcción. Se enfoca en la calidad durante todo el proceso

monitorizando el avance del proyecto. Se basa en iteraciones cortas, aproximadamente de dos semanas, obteniéndose resultados periódicos y tangibles. Ayuda a contrarrestar el exceso de presupuesto, las fallas en la planificación o el hecho de entregar menos de lo que espera el cliente. El proceso consiste en cinco fases secuenciales durante las cuales se desarrolla el diseño y la construcción del sistema:

- Desarrollo de un modelo general.
- Construcción de una lista de funcionalidades.
- Planeación por funcionalidad.
- Diseño por funcionalidad.
- Construcción por funcionalidad

F.D.D necesita tener un equipo compuesto con miembros de experiencia, que marquen el camino a seguir desde el principio, con la elaboración del modelo a seguir, puesto que no es tan ágil como podría serlo X.P. (Calabria, 2003)

1.4. Lenguajes de Modelado

El lenguaje de modelado es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software orientado a objetos. Se usan en combinación de una metodología de desarrollo de software. (2009)

1.4.1. Lenguaje de Modelado Unificado (UML)

El lenguaje Unificado de Modelado es ante todo un lenguaje que proporciona un vocabulario y reglas para permitir una comunicación; en este caso este lenguaje se centra en la representación gráfica de un sistema desde diferentes perspectivas.

U.M.L surge a raíz de la necesidad de comunicar las ideas a otros desarrolladores además para servir de apoyo en los procesos de análisis de un problema. U.M.L se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y diseño; nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Las funciones principales de U.M.L se pueden apreciar como siguen:

- Visualizar: U.M.L permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Aunque este lenguaje está pensado para modelar sistemas complejos, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo en una empresa.

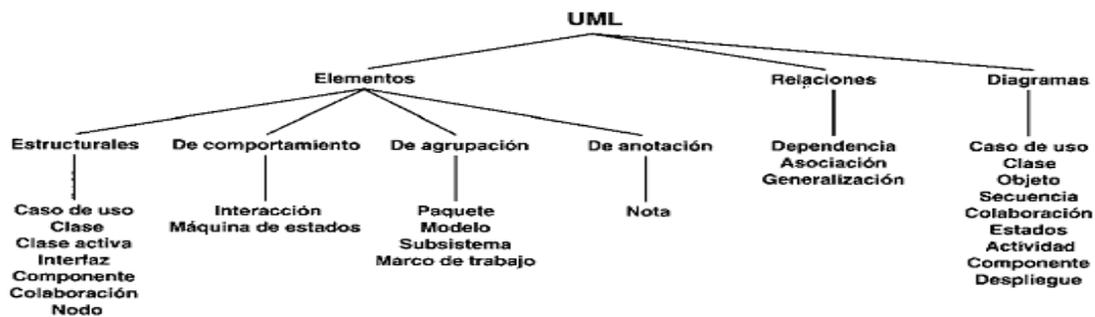


Figura 1-5 Vocabulario de U.M.L. (Jacobson, 2000).

U.M.L es una especificación de notación orientada a objetos. Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto:

- Vista casos de uso: Se forma con los diagramas de casos de uso, colaboración, estados y actividades.
- Vista lógica: Se forma con los diagramas de clases, objetos, colaboración, estados y actividades.
- Vista de procesos: Se forma con los diagramas de la vista de diseño, recalando las clases y objetos referentes a procesos.
- Vista de implementación: Se forma con los diagramas de componentes, colaboración, estados y actividades.

- Vista de despliegue: Se forma con los diagramas de despliegue, interacción, estados y actividades.

La esencia del U.M.L son sus diagramas, que crean las diferentes vistas antes mencionadas. Existen dos tipos diferentes de diagramas los que dan una vista estática del sistema y los que dan una visión dinámica:

Diagramas de estructura estática: Describen las propiedades estructurales del sistema.

- Diagrama de clases: Conjunto de clases, interfaces y colaboraciones.
- Diagrama de objetos: Conjunto de objetos y sus relaciones.
- Diagrama de casos de uso: Conjunto de casos de uso y actores y sus relaciones.

Diagramas de comportamiento:

- Diagramas de interacción (secuencia y colaboración): Objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos.
- Diagrama de estados: Muestra una máquina de estado que consta de estados, transiciones, eventos y actividades.
- Diagrama de actividad: Es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.

Diagramas de implementación:

- Diagrama de componentes: Organización y las dependencias entre un conjunto de componentes.
- Diagrama de despliegue: Configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.

U.M.L aporta las siguientes ventajas:

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos.
- Es un lenguaje de uso universal.

- Prácticamente todas las herramientas C.A.S.E y de desarrollo la han adaptado como lenguaje de modelado.
- El uso de lenguajes visuales facilitan su asimilación y entendimiento por parte del equipo de desarrollo.
- El tiempo invertido en el desarrollo de la arquitectura se minimiza.
- La detección y resolución de errores se agiliza siempre y cuando se haga uso de herramientas adecuadas de diagnóstico y depuración
- La trazabilidad y documentación del proyecto se realiza de una forma ordenada y guiada por los casos de uso.
- Notable efectividad y productividad en labores de diseño arquitectónico y mantenimiento.

En definitiva, implantar metodologías de desarrollo con U.M.L en la organización es una práctica que convierte a los desarrolladores en profesionales más competitivos y eficaces en sus proyectos Software.

1.4.2. BPMN

Notación para el Modelado de Procesos del Negocio (B.P.M.N) es un lenguaje gráfico para representar el negocio, facilitando a las organizaciones la habilidad para comunicar esos procedimientos de una manera estándar.

B.P.M.N está planeado para dar soporte únicamente a aquellos procesos que sean aplicables a procesos de negocios. Esto significa que cualquier otro tipo de modelado realizado por una organización con fines distintos a los del negocio no estará en el ámbito de BPMN.

El principal objetivo de BPMN es proveer una notación estándar que sea de fácil lectura y entendible por parte de todos los involucrados e interesados del negocio (stakeholders). Entre estos interesados están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores técnicos (responsables de implementar los procesos) y los gerentes y administradores del negocio (quienes monitorean y gestionan los procesos). En síntesis BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. (2009)

1.4.3. IDEF0

Definición de la integración para la modelización de las funciones (IDEF0). Son normas que definen la metodología para la representación de funciones modeladas. Facilita el trabajo en situaciones de mayor complejidad de problemas y de mayores exigencias de precisión en el tratamiento. Utiliza unos grafos de visualización de sus elementos, no sólo para facilitar la aplicación del método, sino para diferenciar claramente las magnitudes a tratar en aplicaciones de software. Este va introduciendo niveles de detalle a través de la estructura del modelo. De esta manera, la comunicación se produce dando al lector un tema bien definido con una cantidad de información detallada disponible para profundizar en el modelo. Es muy utilizado para describir procesos de negocio y existen numerosas aplicaciones de software que apoyan su desarrollo.

El IDEF0 guía en la descripción de cada proceso o actividad considerada como combinación de cinco magnitudes básicas que se representan gráficamente como:

- Procesos o actividades.
- Inputs (insumos).
- Controles.
- Mecanismos o recursos para la realización de tareas.
- Outputs o resultados conseguidos en el proceso.

Las principales ventajas que presenta son:

- Es una forma unificada de representar funciones o sistemas.
- Su lenguaje es simple pero riguroso y preciso.
- Permite establecer unos límites de representación de detalle establecido universalmente.
- Puede ser representada con diversos paquetes informáticos como es el iGraff Proccess.

Se recomienda el uso del IDEF0:

- Cuando hay que preparar un modelo de proceso que facilite exactitud, detalle, y claridad en la descripción.

- Cuando el proceso posee cierta complejidad y los otros métodos de descripción darían lugar a diagramas confusos.
- Cuando se trata de modelar una amplia gama de procesos distintos en un P.D.L (Process Description Language) consistente.
- Cuando se posee cierto tiempo para trabajar y desarrollar una descripción/modelo completo y correcto del proceso.
- En los puntos de identificación/definición, en la comprensión y delimitación de aspectos de problemas que se plantean en el proceso, en la presentación de soluciones, y en la estandarización de las mejoras/cambios. (Elm)

1.5. Herramientas CASE.

Las herramientas CASE constituyen productos de software que apoyan al desarrollo de software, las mismas permiten: automatizar procesos repetitivos, mantener la información bien estructurada, gestionar grandes cantidades de información, además de permitir un aumento de la productividad y la calidad.

Tienen como objetivo:

- Permitir la aplicación práctica de metodologías estructuradas.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes del software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones.

Existen varios elementos que componen a una herramienta CASE, como son:

- Repositorio (diccionario)
- Meta modelo (no siempre visible)
- Carga o descarga de datos
- Comprobación de errores
- Interfaz de usuario

Existen varias herramientas de modelado UML que posibilitan el modelado visual de proyectos de software, viabilizando la comunicación entre los integrantes del equipo de desarrollo. Algunos ejemplos son Rational Rose y Visual Paradigm.

1.5.1. Rational Rose.

Es la herramienta CASE desarrollada por los creadores de UML, que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables; ofrece la capacidad de modelar y visualizar sus procesos de negocios y destacar oportunidades para aumentar la eficiencia. Permite el chequeo de la sintaxis UML y realizar ingeniería inversa. Posibilita la generación de documentación automáticamente y la generación de código a partir de los modelos. El navegador UML de Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable.

Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información, pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción. (Rat1)

Unifica a los analistas de negocios, sistemas y datos al permitirles crear y administrar modelos en una herramienta con un solo lenguaje de modelado. Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas.

1.5.2. Visual Paradigm.

Es una herramienta con un diseño centrado en casos de uso y enfocado al negocio que genera un software de calidad, tiene la particularidad de ser un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación. Posee capacidades de ingeniería directa (versión profesional) e inversa, modelo y código que permanece sincronizado en todo el ciclo de desarrollo, disponibilidad de múltiples versiones, disponibilidad en múltiples plataformas.

Existen varias versiones de Visual Paradigm para U.M.L. La gratuita no permite realizar ingeniería inversa, pero permite crear diagramas y generar código a partir de ellos. Admite

la importación y exportación de X.M.L e imágenes, la administración de requerimientos, la creación de esquemas de clases a partir de una base de datos y viceversa, y cuenta con un soporte que facilita el trabajo simultáneo sobre un mismo diagrama entre dos desarrolladores en un tiempo real.

Visual Paradigm para U.M.L es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado U.M.L ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar 13 tipos de diagramas diferentes a través de un intuitivo modelado visual. La herramienta U.M.L CASE también proporciona abundantes tutoriales de U.M.L, demostraciones interactivas de U.M.L y proyectos.

Ventajas del Visual Paradigm:

- Soporta UML versión 2.1.
- Interoperabilidad con modelos UML2 (meta modelos UML 2.x para plataforma Eclipse) a través de XML.
- Presenta un Editor de Detalles de Casos de Uso.
- Permite realizar el Diagramas de flujo de datos.
- Soporta la Generación de objetos Java desde la base de datos.
- Ofrece un Generador de informes para generación de documentación.
- Importación y exportación de ficheros XML.

Tipos de Diagramas en Visual Paradigm:

- Diagramas de Caso de Uso: Representan la funcionalidad del sistema.
- Diagramas de Clase: Describen la estructura (estática) del sistema.
- Diagrama de Actividad: Describen como se desarrolla un flujo de actividades entre elementos del sistema o del dominio.
- Diagramas de Secuencia: Describen la interacción entre elementos del sistema en el tiempo.
- Diagrama de Estado: Describen el estado, condiciones y respuesta de los elementos del sistema.

- Diagrama de Componentes: Describen la organización de los elementos físicos que implementan sistema.
- Diagrama de Despliegue: Describen la configuración del entorno de máquinas y redes sobre el que se distribuyen componentes y procesos del sistema.

1.6. Lenguajes de Programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

Aunque muchas veces se usan los términos 'lenguaje de programación' y 'lenguaje informático' como si fuesen sinónimos, no tiene por qué ser así, ya que los lenguajes informáticos engloban a los lenguajes de programación y a otros más. Un lenguaje de programación permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. (2009)

1.6.1. Java

Java es un lenguaje de programación orientado a objetos, toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Los programas escritos en el lenguaje Java pueden ejecutarse en cualquier tipo de hardware. La recolección de basura de Java es un proceso prácticamente invisible al desarrollador. Es decir, el programador no tiene conciencia de cuándo la recolección de basura tendrá lugar, ya que ésta no tiene necesariamente que guardar relación con las acciones que realiza el código fuente.

Java logra la inclusión de sonidos, animación e interactividad del teclado y el ratón para lograr efectos especiales si se dispone de un programa que convierta la PC en una máquina virtual, estos programas se encuentran incluidos en los siguientes navegadores: HotJava de SUN, en el Netscape a partir de la versión 2.0 y en el Internet Explorer a partir de la versión 3.0. La mayor ventaja de esta tecnología es que la amenaza de cualquier virus al descargar estos programas de la red es casi nula puesto que incluyeron una doble

protección en los applets de JAVA haciendo casi imposible que puedan ser infectados por virus.

La protección consiste en que cada archivo de clases contiene información adicional, permitiendo conocer si los archivos son originales o han sido alterados y por otra parte, tras cargar un applet desde la red y posteriormente ser ejecutado, se realiza una comprobación de funciones que son accedidas por el programa, y solo se permite su ejecución si se cumplen con los criterios de seguridad. (www)

Se selecciona Java para el desarrollo de este trabajo pues los programas escritos en este lenguaje pueden ejecutarse en cualquier tipo de hardware. Es un lenguaje portable, robusto, estable y sencillo de aprender.

1.6.2. C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET el cual es similar al de Java. C# fue diseñado para combinar el control de lenguajes de bajo nivel como C y la velocidad de programación de lenguajes de alto nivel como Visual Basic. C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma.

C# contiene dos categorías generales de tipos de datos integrados: tipos de valor y tipos de referencia. El término tipo de valor indica que esos tipos contienen directamente sus valores. Los caracteres en C# no son cantidades de 8 bits como en otros muchos lenguajes de programación. Por el contrario, C# usa un tipo de caracteres de 16 bits llamado Unicode al cual se le llama char. No existen conversiones automáticas de tipo entero a char.

C# tiene cuatro clases generales de operadores: aritméticos, a nivel de bit, relacionales y lógicos. (CSh)

1.7. Sistemas Gestores de Bases de Datos (SGBD)

Los SGBD son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. (2009)

1.7.1. PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.
- Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Otras características:

- Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (foreign keys).

- Disparadores (triggers): Un disparador o trigger se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:
 - El nombre del disparador o trigger
 - El momento en que el disparador debe arrancar
 - El evento del disparador deberá activarse sobre...
 - La tabla donde el disparador se activará
 - La frecuencia de la ejecución
 - La función que podría ser llamada

Entonces combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (triggers).

- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas. (2009)

1.7.2. MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. Es un sistema de administración relacional de base de datos lo que significa que archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Es un software de fuente abierta por lo que se puede usar y modificar su código fuente.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones, pero poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. (2009)

1.8. Patrones de Caso de Uso

Los patrones de casos de uso modelan reglas que hacen más comprensibles y correctos los modelos. Algunos de estos patrones y los que se seleccionaron a la hora de modelar los casos de uso fueron los siguientes:

1. El nombre revela la intención. (IntentionRevealingName)

Este patrón se utilizó en la confección de los casos de uso ya que plantea como problema que los nombres genéricos sin sentido, pueden no ubicar las expectativas del lector. Es por esto que es bueno utilizar nombres descriptivos para los casos de uso, ya que revelan la intención de cada caso de uso. Para esto propone como solución nombrar los casos de uso utilizando un verbo o frase que represente la meta del actor primario.

Escenario más Fragmentos. (ScenarioPlusFragments)

Este patrón fue puesto en práctica a la hora de realizar la confección de la descripción de los casos de uso, ya que plantea como problema que el lector debe ser capaz de seguir el camino a través del flujo específico o historia en que ellos están interesados, en caso contrario probablemente se frustrarán o extrañarán la información importante. Es por esto que plantea como solución que se debe escribir los eventos del flujo principal como un escenario simple, sin considerar posibles fallos. Debajo ubicar los fragmentos del flujo que muestran la condición alternativa que puede ocurrir.

2. Equipo Pequeño Escribiendo. (SmallWritingTeam)

Este patrón fue puesto en práctica a la hora de distribuir responsabilidades dentro del grupo del trabajo, ya que plantea como problema que utilizar muchas personas para escribir un caso de uso es ineficaz, y los compromisos hechos para alinear los diversos puntos de visión pueden dar lugar a un sistema menos satisfactorio. Para dicho problema plantea la solución de que se tiene que restringir el número de un equipo de trabajo, a solo dos o tres personas. Ya que para comunicarse y llegar a un acuerdo general es más factible un equipo pequeño.

3. Audiencia Participativa. (ParticipatingAudience)

Este patrón se utilizó en todo momento, para facilitar el trabajo con el cliente, ya que como problema plantea que no se puede satisfacer la necesidad de los Stakeholders, y como

solución plantea que se debe implicar activamente a los clientes y a los Stakeholders en el proceso de desarrollo de los casos de uso, ya que se conoce mejor las verdaderas necesidades.

4. Extensión o Inclusión Concreta

Extensión: este patrón consiste en dos casos de uso y la relación de extensión entre ellos; es aplicable cuando un flujo puede extender el flujo de otro caso de uso. Tipo: Patrón de estructura.

5. Extensión o Inclusión Concreta

Inclusión: este patrón consiste en la relación de incluido entre el caso de uso base y el caso de uso incluido. Es aplicable cuando un flujo puede incluir el flujo de otro caso de uso al igual que la realización del mismo. Tipo: Patrón de estructura.

6. CRUD: Completo

Este patrón consiste en un caso de uso base llamado Información CRUD (o Administrador de información), el cual modela los distintos tipos de operaciones que pueden realizarse en un segmento de información de cierto tipo como: Crear, Eliminar, Actualizar, Modificar etc. Tipo: Patrón de estructura.

7. Múltiples actores

Distintos Roles: este patrón consiste en un caso de uso y dos actores, es aplicable cuando dos actores interactúan con un mismo caso de uso pero con roles distintos e interacciones diferentes. Tipo: Patrón de estructura.

8. Múltiples actores

Roles comunes: este patrón es un patrón alternativo en el que dos actores juegan el mismo rol en un caso de uso. Este patrón es aplicable cuando desde el punto de vista del caso de uso existe una entidad externa interactuando con cada instancia del caso de uso. Tipo: Patrón de estructura. (Overgaard, 2004)

1.9. Patrones de diseño

Los patrones de diseño son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema.

Ventajas del uso de patrones de diseño:

- Facilitan la localización de los objetos que formarán el sistema.
- Facilitan la determinación de la granularidad adecuada.
- Especifican interfaces para las clases.
- Especifican implementaciones.
- Facilitan el aprendizaje y la comunicación entre programadores y diseñadores. (Larman, 2003)

Gang o Four:

Gang o Four o Pandilla de los cuatro (G o F) son una herramienta fundamental para cualquier programador que se precie. Recopilaron y documentaron 23 patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos. Estos patrones de diseño son una descripción de clases y objetos que se comunican entre si, adaptada para resolver un problema general de diseño en un contexto particular. (Larman, 2003)

Estos se dividen en:

Patrones Creacionales:

- **Fábrica Abstracta:** Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando.
- **Constructor virtual:** Abstrae el proceso de creación de un objeto complejo, centralizando dicho proceso en un único punto.
- **Método de fabricación:** Centraliza en una clase constructora la creación de objetos de un subtipo de un tipo determinado, ocultando al usuario la casuística para elegir el subtipo que crear.
- **Prototipo:** Crea nuevos objetos clonándolos de una instancia ya existente.
- **Instancia única:** Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. (Larman, 2003)

Patrones Estructurales:

- Adaptador: Adapta una interfaz para que pueda ser utilizada por una clase que de otro modo no podría utilizarla.
- Puente: Desacopla una abstracción de su implementación.
- Objeto compuesto: Permite tratar objetos compuestos como si de uno simple se tratase.
- Envoltorio: Añade funcionalidad a una clase dinámicamente.
- Fachada: Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema.
- Peso ligero: Reduce la redundancia cuando gran cantidad de objetos poseen idéntica información.
- Proxy: Mantiene un representante de un objeto. (Larman, 2003)

Patrones de Comportamiento:

- Cadena de responsabilidad: Permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada.
- Orden: Encapsula una operación en un objeto, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma.
- Intérprete: Dado un lenguaje, define una gramática para dicho lenguaje, así como las herramientas necesarias para interpretarlo.
- Iterador: Permite realizar recorridos sobre objetos compuestos independientemente de la implementación de estos.
- Mediador: Define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto.
- Recuerdo: Permite volver a estados anteriores del sistema.
- Observador: Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él.
- Estado: Permite que un objeto modifique su comportamiento cada vez que cambie su estado interno.
- Estrategia: Permite disponer de varios métodos para resolver un problema y elegir cuál utilizar en tiempo de ejecución.

- Método plantilla: Define en una operación el esqueleto de un algoritmo, delegando en las subclases algunos de sus pasos, esto permite que las subclases redefinan ciertos pasos de un algoritmo sin cambiar su estructura.
- Visitante: Permite definir nuevas operaciones sobre una jerarquía de clases sin modificar las clases sobre las que opera. (Larman, 2003)

Patrones de Software para la asignación General de Responsabilidad

General Responsibility Assignment Software Patterns o Patrones de Software para la asignación General de Responsabilidad (GRASP). Son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. (Larman, 2003)

Estos se dividen en:

- Experto: Asigna una responsabilidad a la clase que tiene la información necesaria para cumplirla. Es un patrón que se usa más que cualquier otro al asignar responsabilidades.
- Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos.
- Bajo Acoplamiento: Asigna una responsabilidad para mantener bajo acoplamiento.
- Alta Cohesión: Asigna una responsabilidad de modo que la cohesión siga siendo alta.
- Controlador: Asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. (Larman, 2003)

1.10. Descripción de los roles de Analista y Diseñador del Sistema

El analista del sistema define el alcance del sistema identifica a los actores y casos de uso que permiten modelar y completar consistentemente el sistema, estructura el modelo de casos de uso y tiene como cometido analizar un problema y describirlo con el propósito de ser solucionado mediante un sistema informático. (2009).

El diseñador de sistemas se encarga de resolver problemas y construir una solución durante el diseño de un sistema. Al tomar decisiones de alto nivel que se apliquen a todo el sistema, el diseñador desglosa el problema en subsistemas, de tal manera que sea posible realizar más trabajo por parte de varios diseñadores que trabajarán independientemente en distintos subsistemas. Se responsabiliza con la asignación de subsistemas a componentes hardware y software. (Dis)

1.11. Ingeniería de Requisitos

Son muchos los proyectos de software que hoy en día superan el presupuesto inicial y no satisfacen las necesidades del cliente, que no son entregados en el tiempo pactado, y el producto resultante es difícil de utilizar, algunos incluso no llegan a terminarse, por existir desacuerdos entre los usuarios y el equipo de desarrolladores en cuanto a lo que debe hacer el sistema. Por esta razón en los últimos años se le ha dado gran importancia a la identificación de requerimientos dentro del proceso de desarrollo de software.

Se hace difícil decidir en la construcción de sistemas que es lo que se va a construir, por ello es imprescindible establecer correctamente los requerimientos de software, pues ninguna otra parte del trabajo puede perjudicar tanto el resultado final si se realiza de forma errónea.

El proceso de Ingeniería de Requisitos tiene el objetivo de descubrir, modelar, validar y mantener un documento de requisitos mediante un conjunto de actividades usando una combinación de técnicas, herramientas y personas, aunque es un proceso centrado en las personas ya que acontece en un universo de discurso con actores reales. Este proceso se adapta a los diferentes modelos de procesos de Ingeniería de Software como pueden ser, de cascada, espiral, prototipado, etc.

El proceso donde se especifican y validan los servicios que debe proporcionar el sistema así como las restricciones sobre las que se deberá operar recibe el nombre de tratamiento de requisitos, y consiste en un proceso iterativo y cooperativo de análisis del problema, en el que se documentan los resultados en distintos formatos y probando la exactitud del conocimiento adquirido.

Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a

requisitos, siendo estos el enlace entre las necesidades reales de los clientes, usuarios y otros participantes vinculados al sistema para resolver o alcanzar un objetivo, necesidades que pueden cambiar. Son atributos necesarios, declaraciones que identifican las capacidades o características de un sistema a fin de que tengan una utilidad a un cliente o usuario.

Los requisitos poseen propiedades deseables, teniendo estos que ser comprensibles para los usuarios y desarrolladores pues estos no son más que el resultado de la comunicación de las necesidades de los clientes. Deben ser verificables, de manera que se pueda comprobar si el sistema cumple o no con determinado requisito, deben ser consistentes de manera que no hallan dos o más requisitos que especifiquen conductas diferentes del sistema para las mismas condiciones, rastreables, de forma tal que se conozca el origen de dicho requisito y sea trazable hacia atrás y hacia delante, y por ultimo que no sean ambiguos, teniendo estos una sola interpretación.

1.11.1. Clasificación de los requisitos:

Los requisitos pueden dividirse en: funcionales y no funcionales.

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer. Estos requisitos dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requisitos. (Sommerville, 2007)

Los requisitos no funcionales son aquellos que se refieren a las propiedades emergentes de un sistema como la fiabilidad, el tiempo de respuesta, la disponibilidad, rendimiento del sistema y la capacidad de almacenamiento, siendo a menudo más críticos que los requisitos funcionales.

1.11.2. Técnicas de obtención de requisitos

La obtención de requisitos es el proceso mediante el cual los interesados en un sistema de software descubren, revelan, articulan y entienden sus requisitos.

Una buena captura de requisitos implica clientes más satisfechos, y soluciona la mayor parte de los problemas que existen en el desarrollo del software, que son los que están relacionados con la obtención de los requisitos. La captura de requisitos no es un proceso

sencillo, pues la identificación de los requisitos ha sido un gran problema que los analistas han tenido que enfrentar debido a que los usuarios en la mayoría de las ocasiones no saben lo que realmente quieren, o puede que se omita información por parecer obvia a clientes o usuarios, que exista incomprensión por parte de los desarrolladores de las necesidades del usuario, que varias personas describan el mismo proceso de forma diferente, o bien porque los analistas no hacen las preguntas adecuadas a las personas adecuadas.

Por ello se han propuesto técnicas que orientan al analista en la obtención de requerimientos. Entre ellas están:

Las entrevistas: La más tradicional de las técnicas, que requiere gran participación del analista, y consiste en reuniones entre el analista y el cliente en la cuales se suceden preguntas y respuestas. Las entrevistas se realizan en conjunto con otras técnicas. (Koch, 2002)

La Tormenta de ideas: basada en reuniones de grupo con el objetivo de generar ideas en un ambiente libre de críticas o juicios. Ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de captura, cuando los requisitos no están todavía muy claros. (Zapata, 2004)

El Juego de roles: consiste en cada uno de los miembros del equipo de desarrollo del software toman el lugar del interesado y ejecutan la actividad de trabajo que éste desempeña, experimentando las inexactitudes y problemas ligados con el sistema que se está especificando.

El Desarrollo conjunto de aplicaciones (JAD): técnica que se basa en la realización de reuniones en grupo durante un período de 2 a 4 días, con el objetivo de ayudar a los clientes y usuarios a formular problemas y explorar posibles soluciones. (Robertson, 2006)

Casos de uso: técnica que incluye las descripciones de actores, eventos, operaciones y objetivos de esas operaciones, generalmente ligados con el funcionamiento de una solución informática, y que facilitan la obtención de requisitos y son fácilmente comprensibles por clientes y usuarios.

Existen otras como las TFEA (Técnicas para facilitar las especificaciones en una aplicación) basadas en la creación de un equipo conjunto de clientes y desarrolladores que trabajan juntos para identificar el problema, proponer soluciones y especificar un conjunto de requisitos de la solución.

La técnica DFC (Despliegue de la función de calidad) incluye entrevistas y encuestas, datos históricos, que se traducen a una tabla de requisitos, usando luego métodos de evaluación, diagramas y matrices para extraer los requisitos.

La Introspección: recomienda que el analista se ponga en lugar del interesado e imagine como desearía esta la aplicación de software, entregando luego suposiciones sobre la funcionalidad que debería tener dicha aplicación al interesado. El problema radica en que un ingeniero no es un tipo normal de cliente, posee un conocimiento técnico más elevado por lo que se podrían recomendar cosas que el cliente no necesite.

1.11.3. Actividades de la ingeniería de requisitos.

La IR provee los mecanismos necesarios para comprender lo que el cliente necesita. Estas actividades se describen a continuación siendo las más importantes.

- Elicitación de requisitos: el objetivo es hacer explícito el conocimiento oculto sobre las necesidades de los clientes y usuarios y el sistema a desarrollar de forma que todos los participantes en el problema sean capaces de entenderlo.
- Análisis de requisitos y negociación: los requisitos obtenidos se agrupan por categorías, se estudia cada requisito en relación con el resto, se examinan los requisitos en su consistencia, completitud y ambigüedad. Es usual que los clientes y usuarios propongan requisitos contradictorios, estos conflictos se resuelven a través de un proceso de negociación. Los riesgos asociados a cada requerimiento son identificados y analizados, además se realizan estimaciones del esfuerzo de desarrollo que se utilizan para valorar el impacto de cada requisito en el coste del proyecto.
- Especificación de requisitos: "La Especificación del Sistema es el producto final sobre los requisitos del sistema obtenidos por el ingeniero. Sirve como fundamento para la ingeniería del hardware, ingeniería del software, la ingeniería de bases de datos y la ingeniería humana. Describe la función y características de un sistema

de computación y las restricciones que gobiernan su desarrollo. La especificación delimita cada elemento del sistema. La Especificación del Sistema describe la información (datos y control) que entra y sale del sistema. (Pressman, 2005)

- Validación de requisitos: se deben comprobar los documentos de requisitos para detectar omisiones, conflictos y ambigüedades no detectadas en el análisis, además se debe comprobar que los requisitos siguen las normas de calidad establecidas.
- Gestión de requisitos: se realiza durante todas las actividades de la IR. Su objetivo es controlar los cambios y el mantenimiento de los requisitos para que representen el sistema que se va a desarrollar o que se ha desarrollado utilizando procedimientos para controlar los cambios y el uso de técnicas para la gestión de la configuración, como factor importante esta la rastreabilidad para poder realizar el análisis de cuanto puede impactar un cambio.

1.12. Diseño de Sistemas

El Diseño de Sistemas es el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, proceso o Sistema, con suficientes detalles como para permitir su interpretación y realización física. Es definir la arquitectura de hardware y software, componentes, módulos y datos de un sistema de cómputo para satisfacer ciertos requerimientos. Es la etapa posterior al análisis de sistemas. Se ocupa de desarrollar las directrices propuestas durante el análisis en función de aquella configuración que tenga más posibilidades de satisfacer los objetivos planteados tanto desde el punto de vista funcional como del no funcional.

Roger Pressman en su libro: "Ingeniería del Software. Un enfoque práctico" plantea una serie de principios básicos del diseño que hacen posible que el ingeniero de software navegue por el proceso de diseño, y que hay que tener en cuenta a la hora de diseñar ya que ellos garantizan el correcto funcionamiento del sistema.

Los principios de diseño son aplicables a todos los niveles del diseño del software, y todos se relacionan entre si:

- En el proceso deben tomarse enfoques alternativos.
- El diseño deberá poderse rastrear hasta el modelo de análisis.

- El diseño no deberá inventar nada que ya esté inventado.
- El diseño deberá minimizar la distancia intelectual entre el software y el problema como si de la vida real se tratara.
- El diseño deberá presentar uniformidad e integración.
- El diseño deberá estructurarse para admitir cambios.
- El diseño deberá estructurarse para degradarse poco a poco, incluso cuando se enfrenta con datos, sucesos o condiciones de operación aberrantes.
- El diseño no es escribir código y escribir código no es diseñar.
- El diseño deberá evaluarse en función de la calidad mientras se va creando, no después de terminado.
- El diseño deberá revisarse para minimizar los errores conceptuales (semánticos).

El proceso de Diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del sistema a construir, se presenta en un nivel de abstracción alto y cada vez se refina más, evaluando la calidad de su evaluación mediante revisiones técnicas formales. Guía para la evaluación de un buen diseño:

- Deberá implementar todos los requerimientos.
- Ser legible y comprensible para poder codificar.
- Proporcionar imagen completa de todos los elementos de un Diseño de Software.

1.13. Métodos de validación de los resultados

En toda investigación es imprescindible realizar el análisis de los resultados de la propuesta de solución al problema en cuestión, para de esta forma saber hasta qué punto satisface realmente al favorecido. En la creación de un software también pasa lo mismo, se debe realizar la validación, la misma ayuda a saber al desarrollador si el sistema que ellos están construyendo es realmente lo que espera el cliente.

Las métricas constituyen un método de evaluación de los productos y procesos de software, las cuales suelen ser aplicadas a muchas organizaciones, procesos y productos. (Edith, 2008)

Por ello se estudiaron varios métodos de validación de manera que fuesen validados todos los artefactos generados durante esta investigación.

Los requisitos son una parte importante de toda investigación, y la validación de los mismos tiene como misión demostrar que la definición de estos determina el sistema que quiere el usuario, si no se realiza una adecuada validación los errores se propagarán a las fases siguientes y precisamente el precio de eliminación de un error en esas fases es muy alto. Por ello es necesario utilizar métodos para su validación.

La métrica de la Calidad de Especificación de los Requisitos mide la especificidad de los requisitos, haciendo que la parte interesada pueda entenderlos de una manera fácil y se puedan probar. (Pressman, 2005)

Para llevar a cabo este proceso se tiene que: N_r representa el número de requisitos del sistema: $N_r = N_f + N_{nf}$

Donde N_f es el número de requisitos funcionales y N_{nf} es el número de requisitos no funcionales. Luego se procede a medir la especificidad de los requisitos con la siguiente fórmula:

$$Q = \frac{N_{ui}}{N_r}$$

Donde N_{ui} es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas. El valor de Q a medida que se acerca a uno, se va disminuyendo la ambigüedad de la especificación. (Edith, 2008)

Otro método de validar los requisitos es con la técnica del Esquema QFD (Quality Function Deployment), su propósito es determinar el número de requisitos que no son considerados en ningún caso de y confirmar que cada uno de los requisitos funcionales han sido registrados en al menos un caso de uso que representará su funcionalidad. Para esto se ha tomado la especificación de los requisitos y el modelo de casos de uso del sistema, verificándose que todo requisito podrá ser implementado a través de algún caso de uso, y que todo caso de uso satisface algún requisito. El esquema QFD es una matriz que representa las casas de calidad, en las cuales las filas representan los “qué”, o sea, la lista de requerimientos, mientras que las columnas representan los “cómo”, es decir, como se llevan a cabo los requerimientos en casos de uso.

Fueron estudiadas también las métricas orientadas a clases teniéndose en cuenta que para el equipo de desarrollo es importante conocer la complejidad del proceso de implementación y las características del diseño desde el punto de vista del nivel de

dependencia entre las clases, la cantidad de pruebas que será necesario realizar, la flexibilidad del código y qué clases podrán ser reutilizables.

Dentro de las métricas orientadas a clases se encuentra la métrica Tamaño de clase (TC) en la que se tienen en cuenta los siguientes aspectos:

- Total de operaciones, ya sean las propias o las heredadas de las clases padres e interfaces que implementen.
- Cantidad de atributos, tanto los de ella, como lo de los padres.
- Promedio general de los dos anteriores para el sistema completo. (Edith, 2008)

Relaciones entre clases (RC) es otra de las métricas pertenecientes al grupo de métricas orientadas a clases. Esta métrica está dada por la cantidad de relaciones de uso que existe entre las distintas clases que forman el diseño propuesto. Se aplica a las mismas clases en las que fue aplicada la métrica TC. Los aspectos de calidad que se miden son: Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas.

Conclusiones Parciales del Capítulo 1.

Luego de haber realizado un minucioso análisis de las metodologías candidatas se decidió seleccionar la metodología R.U.P porque esta realiza una intensa etapa de análisis y diseño previa a la fase de implementación, generándose de esta manera todos los artefactos necesarios para brindarle a los desarrolladores una documentación detallada y abundante que les sirva para realizar una implementación con calidad. El hecho de utilizar la Programación Orientada a Objetos permite obtener un sistema escalable en el tiempo que no necesitará grandes inversiones de recursos en sus modificaciones posteriores, se basa en U.M.L para la modelación de sistemas, Implementa muy buenas prácticas de desarrollo de software debido a que es iterativa e incremental, dirigida por casos de usos y basada en la arquitectura.

Se ha decidido utilizar el lenguaje de modelado U.M.L debido a que está consolidado como el lenguaje estándar en el análisis y diseño de sistemas informáticos. Sus características visuales ofrecen a los integrantes de un equipo una fácil comunicación e intercomunicación que tributa a la común comprensión de lo que se quiere hacer. A través de él se logran robustas aplicaciones, bien diseñadas y de fácil mantenimiento como resultados de ser desarrolladas también bajo la tecnología orientada objetos y la

metodología de desarrollo R.U.P. Además es un lenguaje de modelado estándar de baja complejidad.

Como herramienta de modelado se seleccionó el Visual Paradigm debido a su intuitivo modelado, es multiplataforma y cuenta con amplias funcionalidades que satisfacen las necesidades de los desarrolladores. Además utiliza U.M.L como lenguaje de modelado y soporta el ciclo completo de desarrollo de software. Con ella se agiliza el modelado del software, aumenta la productividad y viabiliza el trabajo en equipo.

Como lenguaje de programación se propone el uso de JAVA debido a su robustez, seguridad, portabilidad, dinamismo e independencia de la arquitectura.

El sistema permitirá realizar peticiones simultáneas por los usuarios para recibir los servicios que se brindan, por lo que se necesita un gestor de base de datos con capacidad ilimitada y debe permitir el trabajo con procedimientos almacenados, consultas e integridad referencial para poder desarrollar un sistema robusto y consistente. El motor de base de datos debe estar basado en las licencias de software libre, y presentar una estabilidad muy alta, y una gran seguridad de los datos. Como consecuencia de todos estos requisitos que debe cumplir la plataforma, se seleccionó PostgreSQL como gestor de base de datos a utilizar.

Capítulo 2. Modelamiento del Negocio

Introducción del Capítulo 2

En este capítulo se realiza una descripción detallada de algunos procesos que se desarrollan en la Vicerrectoría Primera. Se hace un análisis del negocio que se plantea automatizar y se seleccionan los actores que se beneficiarán de la realización de los casos de uso y los trabajadores que los ejecutarán como posibles candidatos a ser actores del sistema. Para una mejor descripción del sistema propuesto se enumeran los requisitos, tanto funcionales como no funcionales, identificando a partir de los primeros los Casos de Uso para de esta forma construir el Diagrama de Casos de Uso del Sistema.

2.1. Modelo de negocio

El Modelo de Negocio, también llamado diseño de negocio es el mecanismo por el cual un negocio trata de generar ingresos y beneficios. Es el primer flujo de trabajo ingenieril que propone la metodología RUP. Es importante destacar que es una técnica para la especificación de los requisitos más importantes del sistema, que da soporte al negocio, con lo cual se refuerza la idea de que sea el propio negocio lo que determine los requisitos. La modelación del negocio se hace con el principal objetivo de comprender la estructura y dinámica de la organización para la cual se implementa el sistema, comprender sus problemas actuales y además asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la misma para lograr obtener los requerimientos del sistema.

La finalidad del modelo de negocio es describir cada proceso de negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio. Una vez identificados los procesos de negocio y los objetivos estratégicos de la empresa, es preciso encontrar los agentes involucrados en su realización. Cada uno de estos agentes o actores del negocio desempeña cierto papel (juega un rol) cuando colabora con otros para llevar a cabo las actividades que conforman dicho caso de uso del negocio. (Pressman, 2001)

2.1.1. Descripción actual de la organización.

Actualmente la Vicerrectoría Primera se compone por nueve Comités Sindicales integrados por diversos afiliados, los cuales son dirigidos por los secretarios generales y estos se encargan de evaluar el desempeño de cada afiliado y velar por el correcto

funcionamiento de su comité sindical. Todos ellos tienen al frente a un secretario general de la Sección Sindical y cuatro activistas para los principales frentes: emulación, finanzas, asuntos laborales y sociales, y cultura deporte y recreación. Los secretarios y activistas necesitan llevar un control de la información que les concierne. El proceso de emulación se desarrolla tomando como base los indicadores y parámetros seleccionados de forma objetiva y cumplible. La información se transmite a través de reuniones, matutinos, murales o vía correo.

2.1.2. Reglas del Negocio

En una organización, la información y los procesos que se manipulan y desarrollan, están restringidos por las reglas del negocio, las cuales aseguran que las actividades se lleven a cabo de acuerdo a las normas impuestas dentro de la propia organización.

Reglas del negocio identificadas:

1. Para acceder a la información los trabajadores deberán ser afiliados de la Sección Sindical VRP.
2. El Secretario del Comité Sindical debe dar a conocer las actividades sindicales mediante un mural informativo.
3. El Secretario del Comité Sindical deberá publicar con anticipación la fecha, hora y lugar de cada una de las reuniones planificadas por el comité sindical, en un mural.
4. El Secretario de Deporte, Cultura y Recreación debe publicar la fecha y el lugar de las actividades.
5. El Secretario de la Sección Sindical deberá mostrar los deberes y derechos de un sindicalista, así como los tipos de sanciones en el mural del sindicato.
6. Los indicadores de la emulación solo serán gestionados por el responsable de la emulación en cada comité sindical y en la Sección Sindical VRP.

2.1.3. Actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Los actores del negocio interactúan con el negocio enviando y recibiendo mensajes, y para conocer el papel del actor se debe precisar en qué procesos se involucra el actor. (2007-2008)

Tabla 2-1: Justificación de los actores del negocio

Actores del Negocio	Justificación
Afiliado	Es el personal que recibe todos los beneficios de los procesos del negocio. Inicia los casos de uso Informar Afiliado y Realizar emulación individual.
Dirigente Comité Sindical	Representa a los directivos del Comité Sindical. Inicializa los casos de uso Informar Comité Sindical y Realizar emulación por Comité Sindical.

Un trabajador del negocio es una persona, grupo de personas, organización, sistema (software) o hardware dentro del negocio que realiza las actividades que están comprendidas dentro de un caso de uso. Los trabajadores del negocio siempre permanecen dentro de las fronteras del mismo y posteriormente se convertirán en usuarios del sistema a desarrollar. Cada trabajador del negocio debe definirse brevemente con su responsabilidad dentro del negocio. Los aspectos fundamentales a tener en cuenta durante la definición de los trabajadores del negocio son los que se relacionan a continuación:

- Un Trabajador del Negocio representa a un ser humano, organización, software o hardware que desempeña un rol dentro de las Realizaciones del Caso de Uso del Negocio.
- El trabajador interactúa con entidades y otros trabajadores para que el negocio funcione.
- Los trabajadores de negocio son roles y no posiciones organizacionales, ya que una persona puede desempeñar varios roles pero solo tiene una posición en la organización. (2008)

Tabla 2-2: Justificación de los Trabajadores del Negocio

Trabajadores del Negocio	Justificación
Secretario de Comité Sindical	Tiene el deber de controlar y custodiar los documentos y la información que se genera en el

	sindicato y lleva el registro de todas las actividades que se llevan a cabo.
Activista de emulación	Es el encargado de establecer los indicadores para la realización satisfactoria de la emulación. Selecciona los trabajadores cumplidores y desarrolla las vías de estimulación a los mismos. A nivel de comité sindical
Secretario de Sección Sindical	Máximo representante de todos los afiliados y responsable de las orientaciones que se imparten a los trabajadores que representa. Responsable de convocar reuniones del sindicato y controlar e informar las tareas que se realizan.
Secretario de Emulación	Es el encargado de establecer los indicadores para la realización satisfactoria de la emulación. Selecciona los trabajadores cumplidores y desarrolla las vías de estimulación a los mismos. A nivel de sección sindical.

2.1.4. Diagrama de Casos de Uso del Negocio

El Diagrama de Casos de Uso del negocio describe los procesos del negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como clientes y socios.

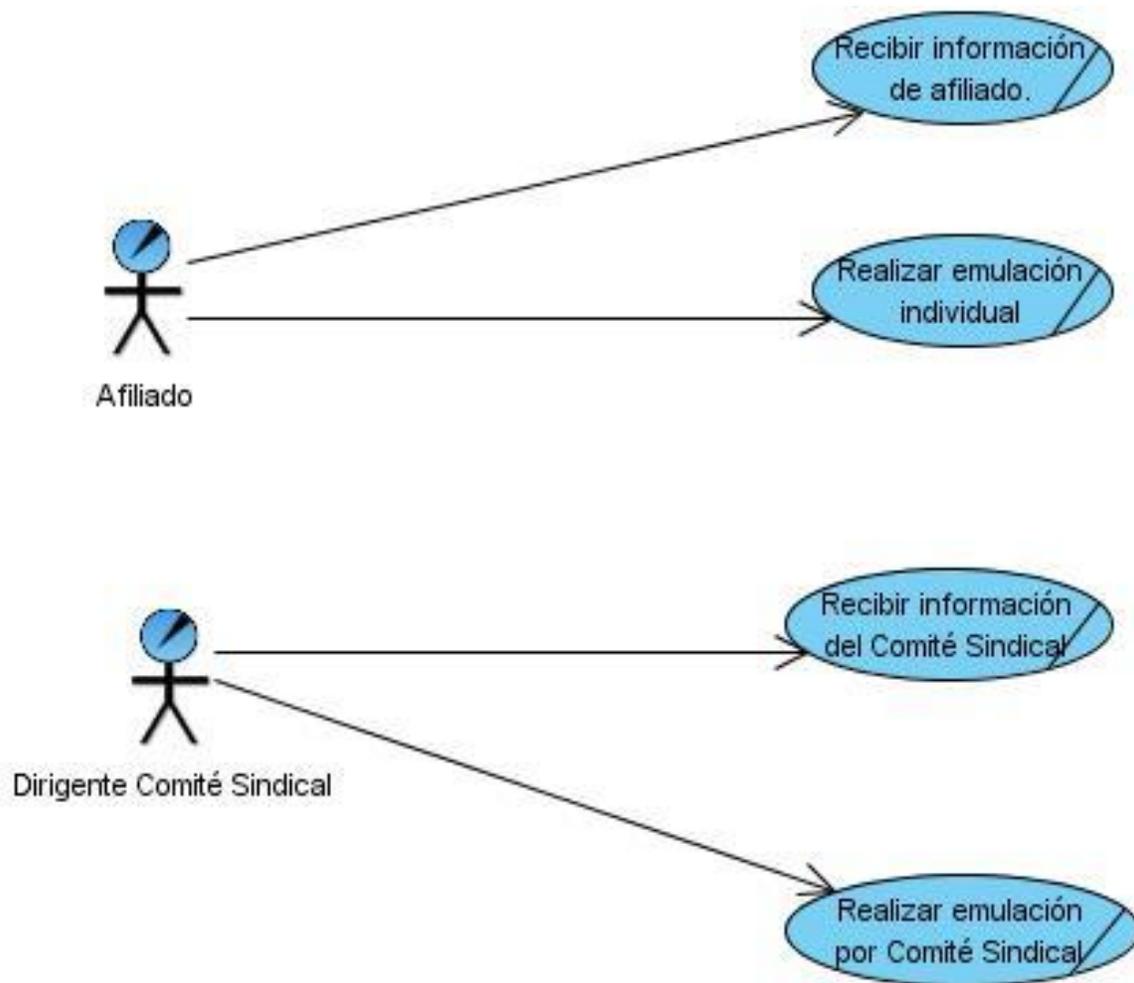


Figura 2-1 Descripción de Casos de Uso del Negocio

2.1.5. Descripción de los Casos de Usos del Negocio

Un caso de uso del negocio es una secuencia de actividades o acciones dentro de un proceso de negocio que produce un resultado observable para ciertos actores del negocio. (2007-2008)

Tabla 2-3: Descripción del Caso de Uso del Negocio Recibir información de afiliado.

Caso de Uso del Negocio	Recibir información de afiliado
Actor	Afiliado(inicia)
Trabajadores	Secretario del Comité Sindical
Propósito	Que el afiliado consulte información.
Resumen:	El caso de uso inicia cuando el afiliado solicita información, el secretario del comité sindical le facilita información política, de noticias nacionales e internacionales, efemérides relevantes y da las orientaciones sindicales.

Capítulo II: Modelamiento del Negocio

Flujo normal de los eventos	
Acción del actor	Respuesta del proceso de negocio
1-El afiliado solicita información	2-El secretario del comité sindical convoca a un encuentro.
3-El afiliado asiste al encuentro	4-El secretario del comité sindical proporciona información política, noticias y efemérides.
	5-El secretario da orientaciones sindicales y aborda las tareas y actividades a cumplir por los afiliados.
	6-El secretario del comité controla la asistencia.
	7-El secretario envía la información y las orientaciones sindicales al afiliado.
8-Recibe y consulta la información.	
Flujos Alternos	
Prioridad: Crítico	
Precondiciones: La existencia de la información actualizada.	

Tabla 2-4: Descripción del Caso de Uso del Negocio Recibir información del Comité Sindical.

Caso de Uso del Negocio	Recibir información del Comité Sindical
Actor	Dirigente Comité Sindical(inicia)
Trabajadores	Secretario de sección sindical
Propósito	Facilitar información del comité sindical
Resumen: El caso de uso inicia cuando el dirigente del Comité Sindical solicita, el secretario de la sección sindical le facilita información política, de noticias nacionales e internacionales, efemérides relevantes e informa las orientaciones sindicales y las actividades a realizar.	
Flujo normal de los eventos	
Acción del actor	Respuesta del proceso de negocio
1-El dirigente del comité sindical solicita información.	2-El secretario general de la sección sindical convoca a encuentro.
3-El dirigente del comité Sindical asiste a la reunión.	4-El secretario de la sección sindical proporciona información política, noticias y efemérides.
	5-El secretario de la sección sindical informa las orientaciones sindicales y las actividades a realizar.
	6-El secretario de la sección sindical controla la asistencia.
	7- El secretario de la sección sindical envía la información y orientaciones sindicales al dirigente del comité.
8-Recibe y consulta la información.	
Flujos Alternos	
Prioridad: Crítico	
Precondiciones: La existencia de la información actualizada.	

Tabla 2-5: Descripción del Caso de Uso del Negocio Realizar Emulación Individual.

Caso de Uso del Negocio	Realizar Emulación Individual
Actor	Afiliado(inicia)
Trabajadores	Secretario del Comité Sindical Secretario de sección sindical Activista de Emulación
Propósito	Posibilita la emulación entre afiliados mediante parámetros e indicadores a cumplir.
Resumen: El caso de uso inicia cuando el afiliado solicita los indicadores de emulación, el activista de emulación se los envía y el afiliado actualiza el estado de su desempeño. El secretario del comité sindical controla el desempeño del afiliado y lo estimula en caso de ser el más destacado, y el secretario de la sección sindical lo estimula si es el afiliado más integral de la sección sindical.	
Flujo normal de los eventos	
Acción del actor	Respuesta del proceso de negocio
1-Solicita los indicadores de emulación.	2-El activista de emulación selecciona los parámetros de emulación y los envía al afiliado.
3-Recibe los indicadores y actualiza su desempeño.	
4-Informa su desempeño.	5-El secretario del comité sindical controla el desempeño del afiliado.
	6-El activista de emulación evalúa el desempeño del afiliado.
	7-El activista de emulación selecciona a los afiliados cumplidores y los envía al secretario del comité sindical.
	8-El secretario del comité sindical estimula al afiliado más destacado por comité y le envía los destacados al secretario de la sección sindical.
	9-El secretario de la sección sindical selecciona al afiliado más integral de la sección sindical y lo estimula.
Flujos Alternos	
Prioridad: Crítico	
Precondiciones: La existencia de los parámetros e indicadores para la emulación.	
Pos condiciones : El mejor trabajador de la sección sindical se registra en el expediente del mejor trabajador	

Tabla 2-6: Descripción del Caso de Uso del Negocio Realizar Emulación por Comité Sindical.

Caso de Uso del Negocio	Realizar Emulación por Comité Sindical
Actor	Dirigente Comité Sindical(inicia)

Trabajadores	Secretario de la Sección Sindical Secretario de Emulación
Propósito	Realizar la emulación entre comités sindicales mediante parámetros e indicadores a cumplir.
Resumen:	El caso de uso inicia cuando el activista de emulación muestra los indicadores de emulación al dirigente del comité sindical y el mismo actualiza el estado de su desempeño. El secretario del comité sindical controla el desempeño del comité sindical y lo estimula en caso de ser el más destacado.
Flujo normal de los eventos	
Acción del actor	Respuesta del proceso de negocio
1-El dirigente del comité sindical solicita los parámetros de emulación	2-El activista de emulación muestra los parámetros de emulación
3-Actualiza desempeño del Comité Sindical.	4-El secretario de la sección sindical controla el desempeño del comité sindical.
	5-El activista de emulación evalúa el desempeño del comité sindical.
	6-El activista de emulación selecciona a los comités sindicales cumplidores.
	7-El secretario del comité sindical estimula al comité sindical más destacado.
Flujos Alternos	
Prioridad: Crítico	
Precondiciones: La existencia de la parámetros e indicadores para la emulación.	

2.1.6. Diagrama de Objetos

El diagrama de objetos muestra como se relacionan los trabajadores del negocio con las entidades.

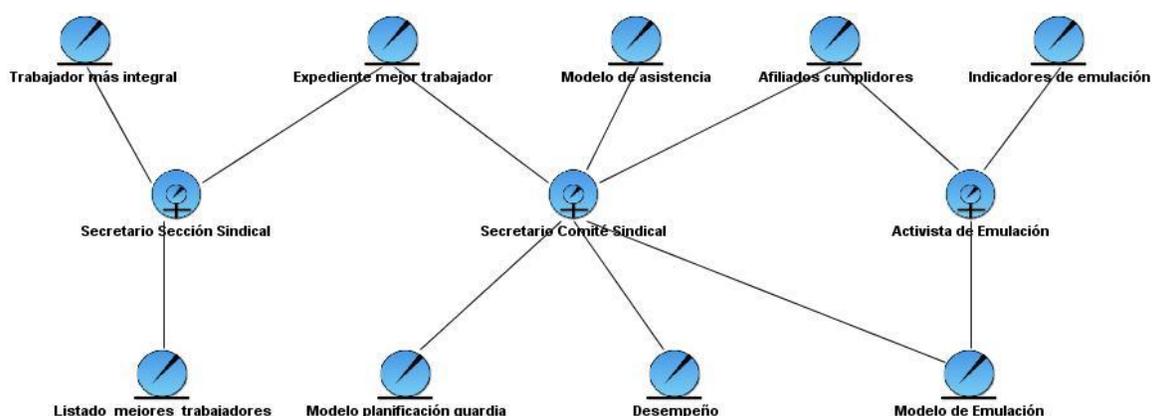


Figura 2-2 Diagrama de Objetos Realizar Emulación Individual

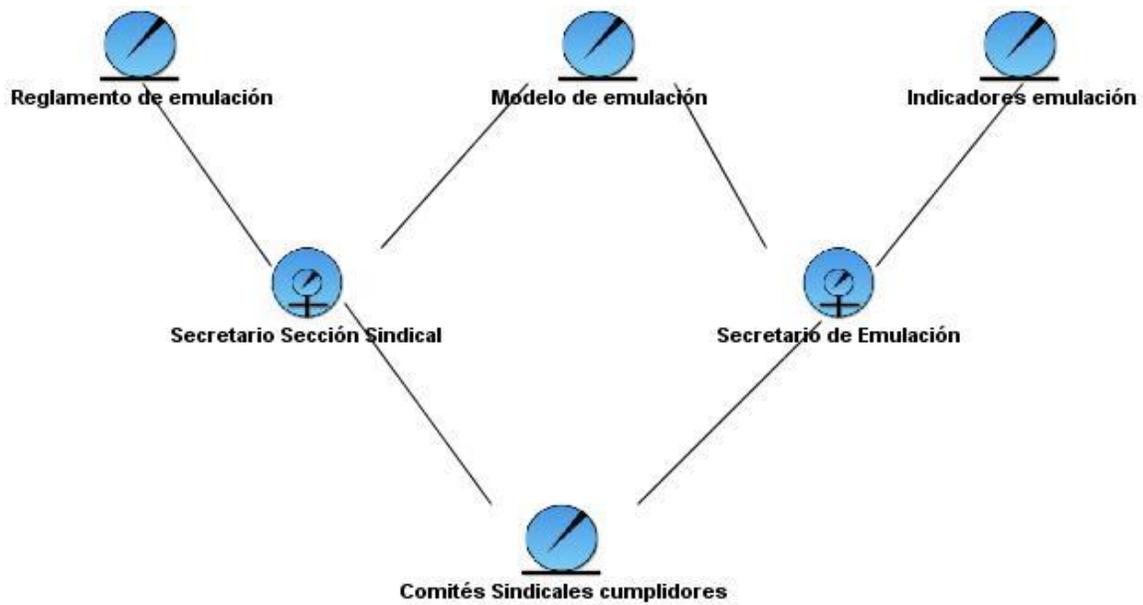


Figura 2-3 Diagrama de Objetos Realizar Emulación por Comité Sindical.

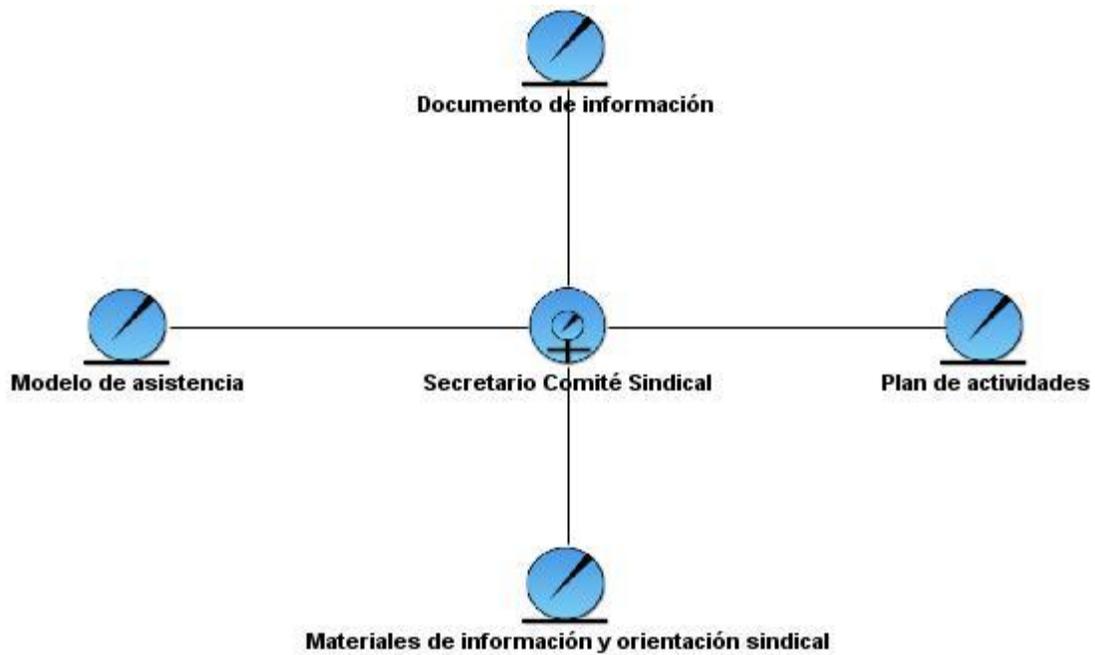


Figura 2-4 Diagrama de Objetos Recibir Información de Afiliado.

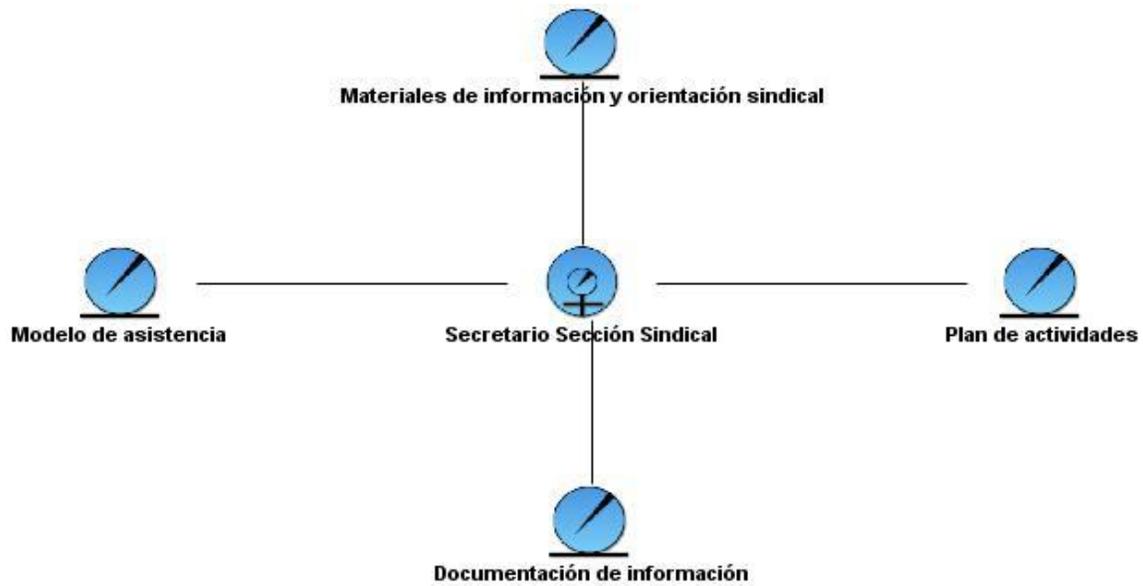


Figura 2-5 Diagrama de Objetos Recibir Información del Comité Sindical.

2.1.7. Diagrama de Actividades

Los Diagramas de Actividades ayudan a describir el detalle de lo que pasa dentro del negocio, y para ello examinamos los roles específicos de las personas (trabajadores del negocio) y las actividades que realizan. Ayudan también a identificar las funciones a cumplir por el producto del software, y quiénes serán los actores del futuro sistema.

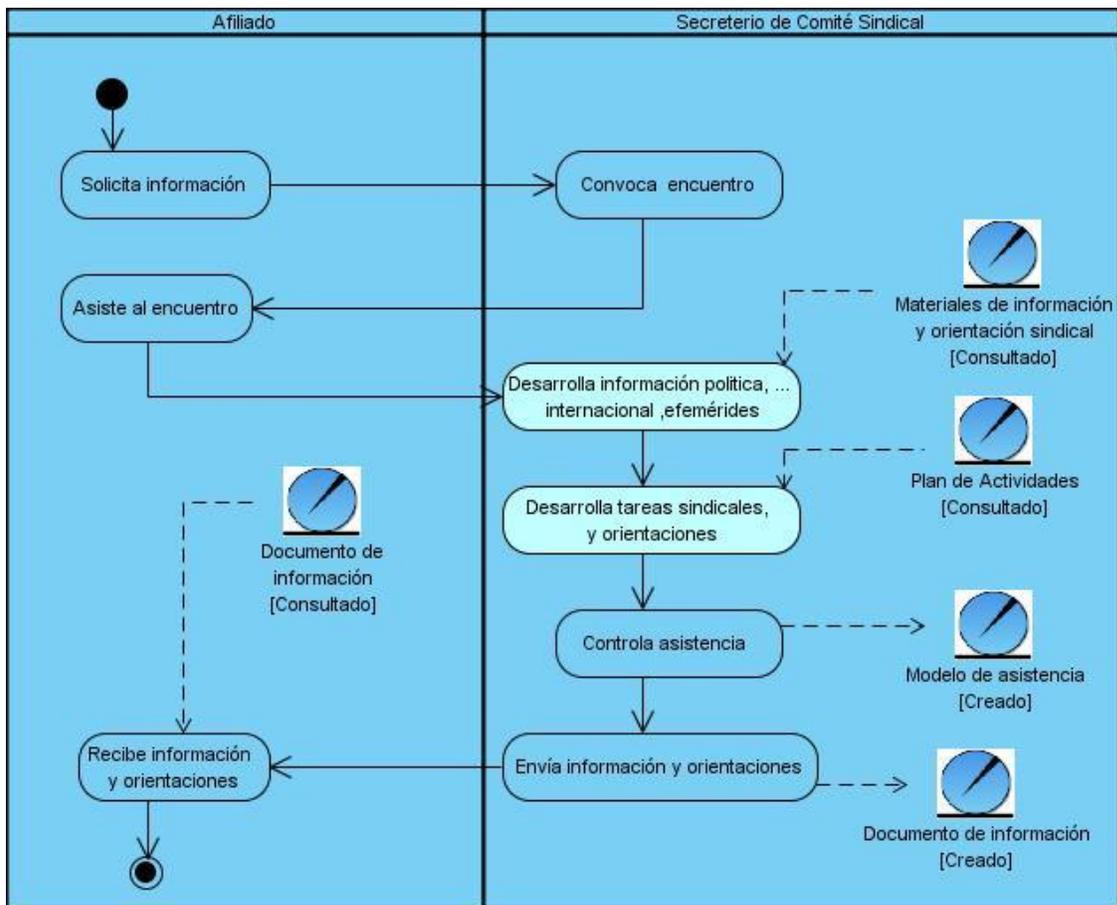


Figura 2-6 Diagrama de Actividades Recibir Información de Afiliado.

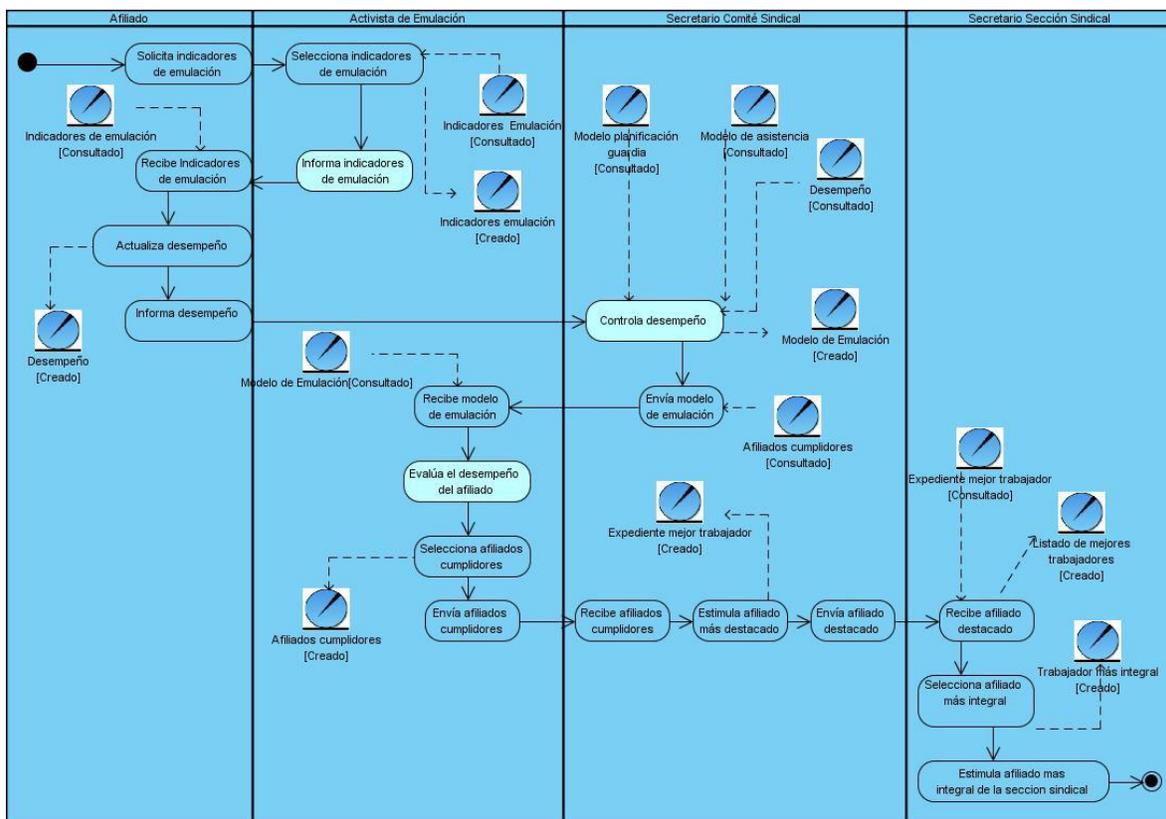


Figura 2-7 Diagrama de actividades Realizar Emulación Individual.

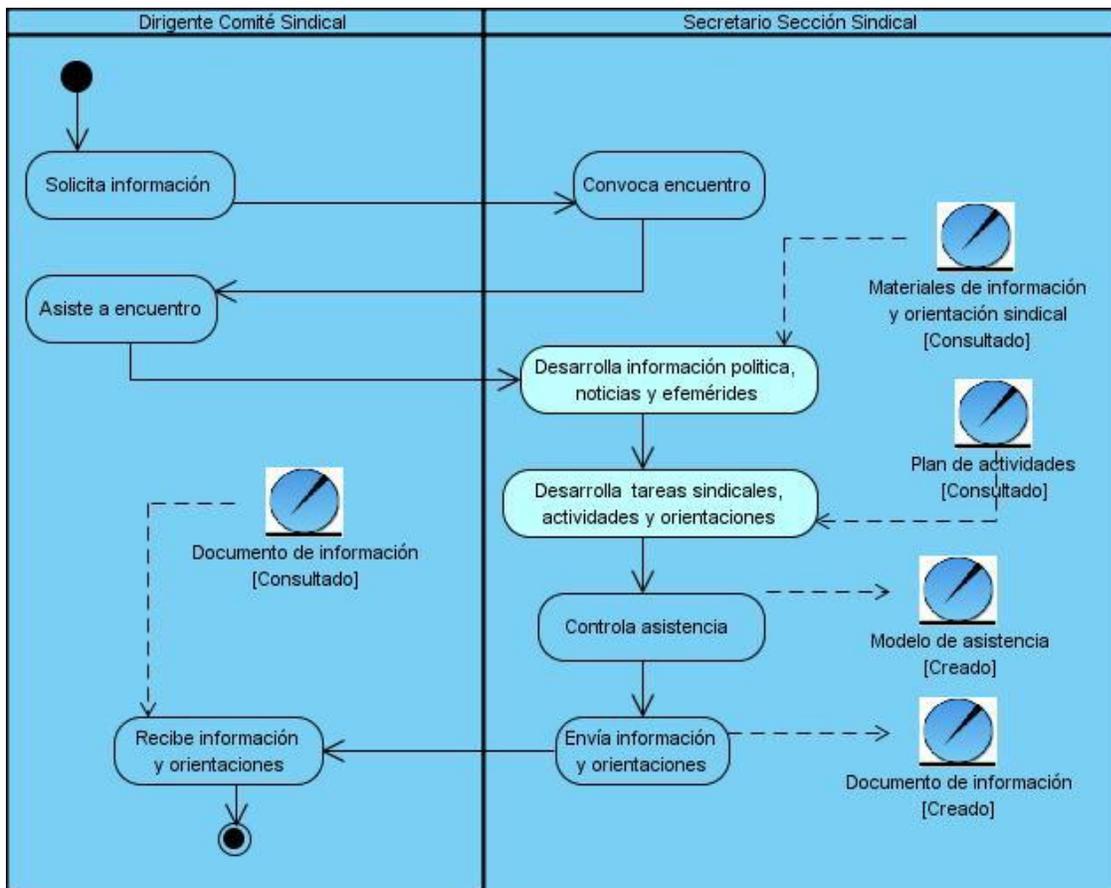


Figura 2-8 Diagrama de Actividades Recibir Información del Comité Sindical.

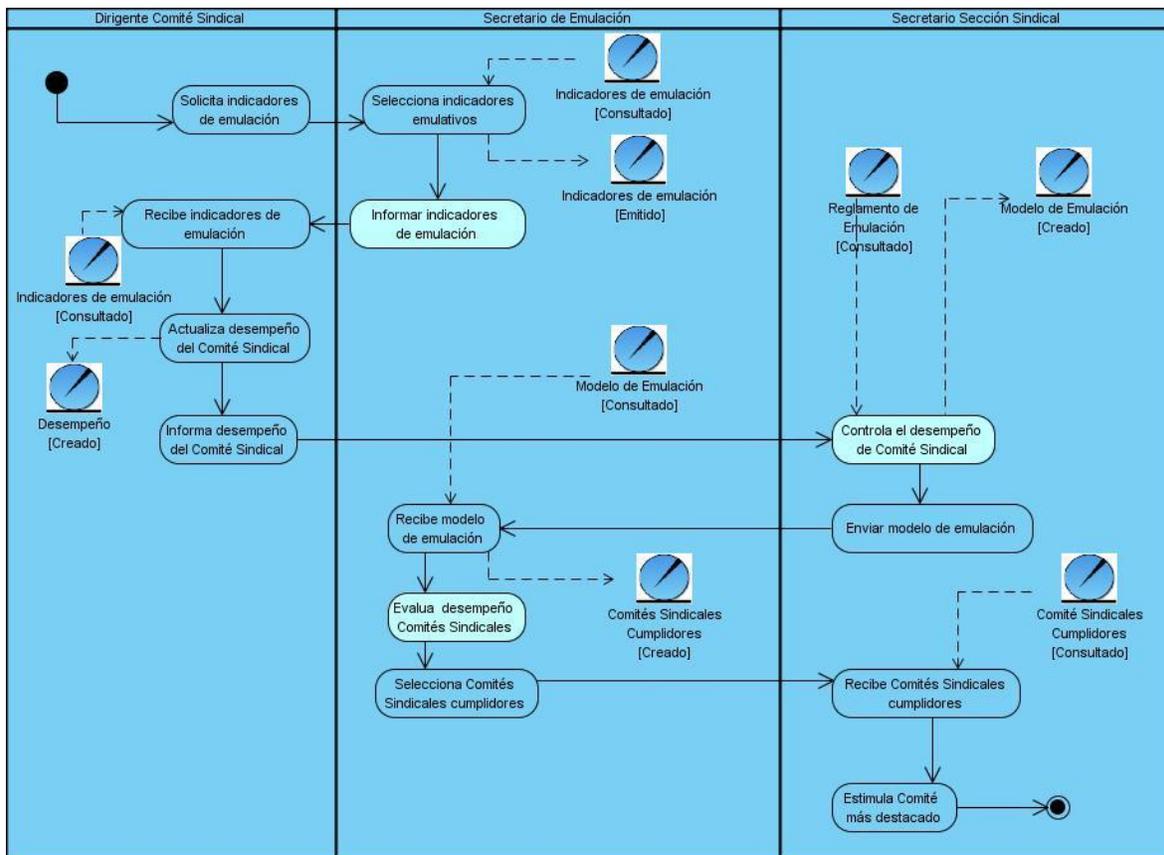


Figura 2-9 Diagrama de Actividades Realizar Emulación por Comité Sindical.

2.2. Especificación de los requisitos de software.

2.2.1. Requisitos funcionales

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas se pueden definir los siguientes requisitos funcionales:

1. Modificar reglamento.
El sistema tiene que permitir al usuario modificar el reglamento ya existente.
2. Mostrar reglamento.
El sistema tiene que permitir mostrar al usuario los reglamentos.
3. Adicionar reglamento.

- El sistema debe permitir al usuario adicionar nuevo reglamento.
4. Eliminar parámetro.
El sistema debe permitir eliminar parámetros.
 5. Modificar parámetro.
El sistema debe permitir al usuario modificar parámetros ya existentes.
 6. Adicionar parámetro.
El sistema debe permitir al usuario adicionar parámetros.
 7. Mostrar parámetros
El sistema debe permitirle al usuario ver los parámetros existentes.
 8. Adicionar evaluación de afiliados.
El sistema debe permitir adicionar la evaluación de cada uno de los afiliados de la sección sindical.
 9. Mostrar evaluación de afiliado.
El sistema debe permitir mostrar al usuario la evaluación de un afiliado.
 10. Eliminar evaluación de afiliado.
El sistema debe permitir eliminar la información referente a la evaluación de un afiliado.
 11. Modificar evaluación de afiliado.
El sistema debe permitir modificar la información referente a la evaluación de un afiliado.
 12. Mostrar afiliado más integral.
El sistema debe mostrar al usuario el afiliado más integral de la sección sindical.
 13. Eliminar afiliado más integral.
El sistema debe permitir eliminar el afiliado más integral.
 14. Adicionar afiliado más integral.
El sistema debe permitir adicionar afiliado más integral de la sección sindical a la planilla de "Afiliado más integral".
 15. Modificar afiliado más integral.
El sistema debe permitir modificar la información referente a la evaluación de un afiliado.
 16. Mostrar comité sindical integral.
El sistema debe mostrar al usuario el comité sindical integral.
 17. Adicionar comité sindical integral.

El sistema debe permitir al usuario adicionar el comité sindical integral a la planilla de “Comité sindicales cumplidores”.

18. Eliminar comité sindical integral.

El sistema debe permitir al usuario eliminar un comité sindical.

19. Modificar comité sindical integral.

El sistema debe permitir al usuario modificar la información referente al comité sindical integral.

20. Mostrar reporte.

El sistema debe permitir mostrar cualquier documento que el usuario desee consultar.

21. Imprimir reporte.

El sistema debe permitir al usuario imprimir el reporte.

22. Exportar reporte.

El sistema debe permitir al usuario exportar el reporte a formatos .pdf, .xls, .doc.

23. Mostrar noticias.

El sistema debe mostrarle al usuario las noticias tanto nacionales como internacionales.

24. Adicionar noticias.

El sistema debe permitir al usuario adicionar noticias tanto nacionales como internacionales.

25. Eliminar noticias.

El sistema debe permitir al usuario eliminar noticias tanto nacionales como internacionales.

26. Modificar noticias.

El sistema debe permitir al usuario modificar noticias tanto nacionales como internacionales.

27. Mostrar efemérides.

El sistema debe mostrarle al usuario las efemérides del mes.

28. Adicionar efemérides.

El sistema debe permitir al usuario adicionar las efemérides del mes.

29. Eliminar efemérides.

El sistema debe permitir al usuario eliminar las efemérides del mes.

30. Modificar efemérides.

El sistema debe permitir al usuario modificar las efemérides.

31. Mostrar actividades extra laborales.

El sistema debe mostrar al usuario todas las actividades extra laborales planificadas.

32. Adicionar actividades extra laborales.

El sistema debe permitir al usuario adicionar actividades extra laborales

33. Eliminar actividades extra laborales.

El sistema debe permitir al usuario eliminar actividades.

34. Modificar actividades extra laborales.

El sistema debe permitir al usuario modificar las actividades planificadas.

35. Autenticar usuario.

El sistema debe permitir la autenticación de los usuarios.

36. Adicionar usuario.

El sistema debe permitir adicionar usuarios.

37. Eliminar usuario.

El sistema debe permitir eliminar usuarios.

38. Mostrar usuario.

El sistema debe permitir mostrarle al usuario otros usuarios.

39. Modificar usuario.

El sistema debe permitir modificar usuarios.

40. Buscar información.

El sistema debe permitir al usuario buscar información referente a cualquier tema relacionado con la sección sindical dependiendo de criterios de búsqueda.

41. Registrar participación en los eventos.

El sistema debe permitir registrar la participación de los afiliados en los eventos.

42. Registrar resultados.

El sistema debe permitir registrar los resultados obtenidos por un afiliado en un evento determinado.

43. Registrar participación en las actividades.

El sistema debe permitir registrar la participación de los afiliados en las actividades.

44. Mostrar actividad sindical.

El sistema debe permitir mostrarle al usuario las actividades sindicales planificadas en ese momento por la organización sindical.

45. Modificar actividad sindical.

El sistema debe permitir modificar la información de una actividad sindical planificada.

46. Eliminar actividad sindical.

El sistema debe permitir al usuario eliminar una actividad sindical.

47. Adicionar actividad sindical.

El sistema debe permitir al usuario adicionar una actividad sindical.

2.2.2. Requerimientos no funcionales

Teniendo en cuenta que los requisitos no funcionales (RNF) son propiedades o cualidades que el producto software debe tener, y que estos tienen varias clasificaciones, en este trabajo los requisitos no funcionales fueron agrupados de la siguiente manera:

1. Requerimientos de Usabilidad

- La aplicación Web será amigable y de fácil acceso para todo usuario del dominio UCI que lo visite.
- Las habilidades requeridas para el manejo de la aplicación consistirán sólo en conocimientos informáticos básicos (navegabilidad Web, inserción de datos).
- La aplicación debe permitir a los usuarios un acceso rápido a las diferentes funcionalidades que brinde, el manejo del mismo debe ser fácil para que su utilización no se tome tediosa para los usuarios que tengan muy poca información sobre computación.

2. Requerimientos de apariencia o interfaz externa

- El diseño del sistema debe ser sencillo y fácil de utilizar de manera que no haya dificultades en el entendimiento y manejo del mismo por parte de los usuarios que no son expertos en la rama de la informática.
- La interfaz será amigable y no sobre cargada de información por página.
- Deben utilizarse plantillas con un mismo estilo para que el usuario no se sienta perdido dentro de la aplicación.

- Por el uso diario y constante que tendrá el software, la interfaz debe ser agradable, que favorezca el estado de ánimo del cliente y que combine correctamente los colores, tipo de letra y tamaño.
3. Requerimientos de Seguridad
- Confiabilidad: La información manejada por el sistema está protegida de acceso no autorizado, y se especifican los diferentes roles existentes para establecer el acceso de los usuarios a determinadas acciones.
 - Integridad: La información será protegida contra corrupción y estados inconsistentes
 - Disponibilidad: Los usuarios del sistema podrán acceder a la información cada vez que lo necesiten.
4. Restricciones de diseño e implementación
- El sistema será desarrollado utilizando como lenguaje de programación Java y como gestor de BD Postgre SQL.
5. Requerimientos de hardware
- Se requiere que las computadoras tengan tarjeta de red.
 - Se requiere tengan al menos 256 MB de memoria RAM.
 - Se requiere al menos 4 GB de disco duro.
 - Procesador 512 MHz como mínimo.
6. Requerimientos de Software
- Se requiere Sistema Operativo Windows 95 o Superior
7. Requerimientos de rendimiento
- El tiempo de respuesta de una petición al servidor y la velocidad de procesamiento de la información deben ser rápidas para la toma de decisiones.
 - La aplicación debe estar concebida para el consumo mínimo de recursos.

Conclusiones Parciales del Capítulo 2

En este capítulo se comenzó a desarrollar la propuesta de solución a partir del estudio realizado de los procesos del negocio. Los diagramas de actividades, la interacción entre los actores y casos de uso, el modelo de objeto con los trabajadores y las clases entidades con las que se relacionan, así como la captura y especificación de los requisitos

funcionales y no funcionales constituyen la base para la realización del flujo de análisis y diseño del sistema.

Capítulo 3. Análisis y diseño del sistema

Introducción del Capítulo3

Este capítulo esta conformado por el diagrama de casos de uso del sistema; la descripción de algunos de estos y la modelación de los diagramas de clases del análisis correspondiente; donde se aprecia la interacción del usuario con las diferentes interfaces del sistema y las clases entidades involucradas. Seguidamente se muestran los diagramas de clases y los diagramas de secuencia, grupo de artefactos pertenecientes a la parte del diseño, dando paso a la propuesta del sistema.

3.1. Definición de los actores

Los actores representan los usuarios del sistema y otras aplicaciones que interactúan con él, suelen corresponderse con trabajadores (o actores del negocio) en un negocio. También se representa mediante uno o más actores cada sistema externo con el que interactúa el sistema. (Jacobson I. , 1999)

Tabla 3-1: Actores del Sistema

Actores	Justificación
Afiliado	El afiliado consulta su desempeño laboral, las actividades a realizar, revisa los avisos e información que se emiten y desarrolla diversas tareas según el plan de trabajo en la Sección Sindical VRP.
Secretario de Comité Sindical	El secretario de comité sindical controla el desempeño de los afiliados de su comité y gestiona la información que emitirá a sus afiliados. Es responsable del buen funcionamiento del comité sindical.
Secretario de Sección Sindical	Quien controla el desempeño de todos los comités sindicales y gestiona la información que emitirá a los mismos es el secretario de sección sindical, máximo responsable de la sección sindical.
Secretario de Emulación	El secretario de emulación gestiona los parámetros de emulación que emitirá a los afiliados y se encarga de la emulación sindical.
Secretario de deporte cultura y recreación	El secretario de cultura ,deporte y recreación gestiona las actividades extra laborales que se realizan, como eventos deportivos, festivales culturales etc.
Administrador	El administrador vela por la seguridad de la información a través de los permisos de acceso.

3.2. Diagrama de casos de uso del sistema.

“El modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones. Representa un esquema donde se recogen las funcionalidades del negocio que se automatizan y determina como será utilizado desde el punto de vista del usuario (Actor), pues se construye sobre la base de sus necesidades”. (Jacobson I. , 1999)

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. (Jacobson I. , 1999)

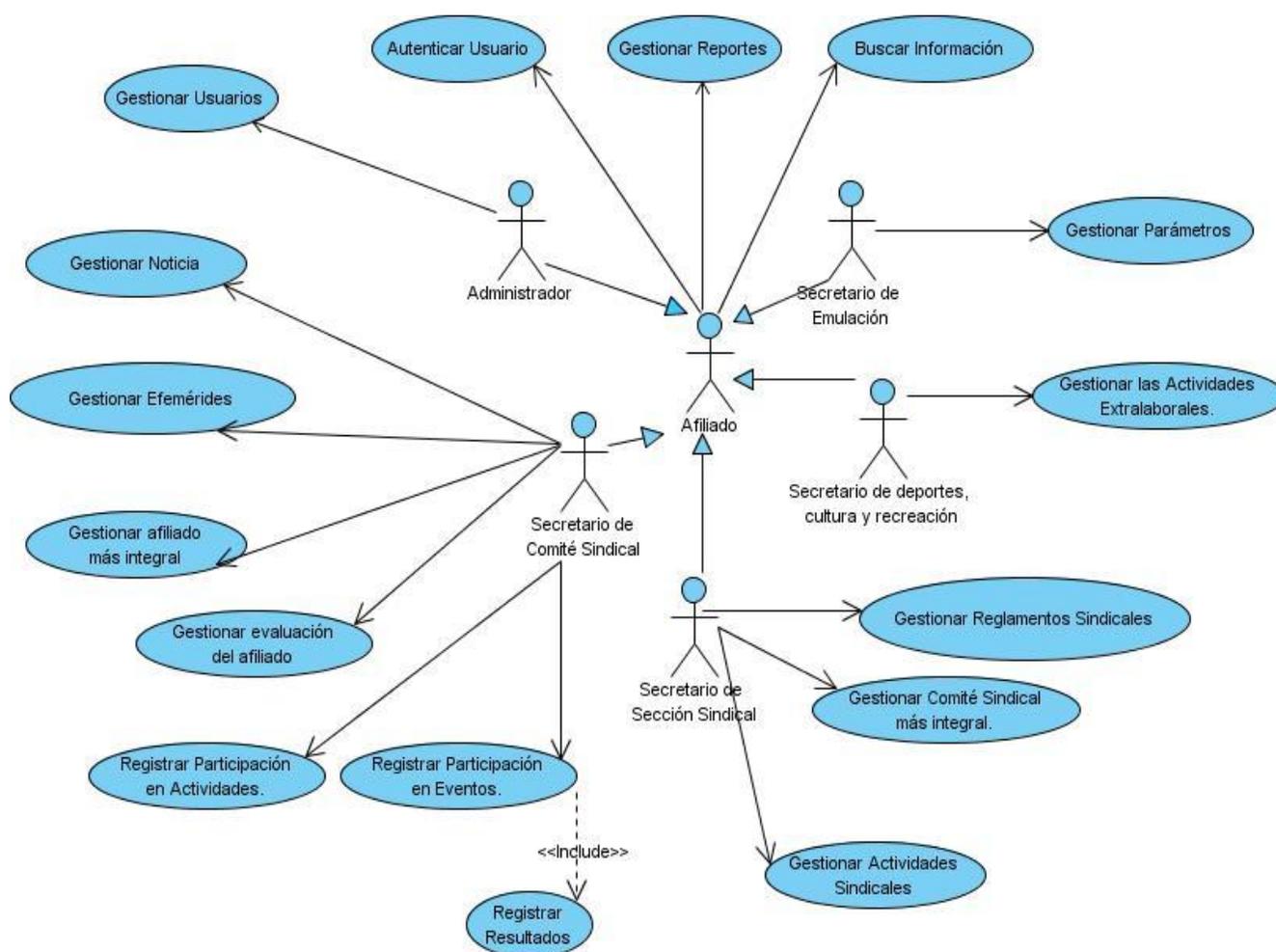


Figura 3-1 Diagrama de casos de uso del sistema.

3.3. Descripción de los casos de uso del sistema.

Mediante las descripciones de los casos de uso se especifica la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema.

A continuación se mostrarán las descripciones de los casos de uso del sistema: Autenticar Usuario, Gestionar Noticia, Gestionar usuario y Gestionar Comité sindical más cumplidor. En el Anexo 1 se podrán encontrar las descripciones del resto de los casos de uso del sistema.

Tabla 3-2: Descripción del CU Autenticar Usuario

Caso de uso	Autenticar usuario
Actor	Afiliado(inicia)
Propósito	Autenticar usuario al sistema
Resumen	El actor introduce su usuario y contraseña. El sistema le permite la posibilidad de acceder. EL caso de uso termina cuando el usuario accede al sistema satisfactoriamente.
Referencias	RF35
Precondiciones	El actor debe estar registrado previamente.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1- Introduce los datos necesarios y presiona aceptar.	2- El sistema valida los datos
	3- El sistema muestra interfaz principal según el rol del usuario.
	4- Termina el caso de uso.
Curso Alterno	
Línea 2	
2.1 El sistema detecta errores en los datos insertados y los notifica al usuario solicitando que sean rectificadas.	
2.2 Usuario arregla los errores y solicita aceptar.	
Regresa la línea 2 de la sección 1.	
El usuario selecciona cancelar y termina el caso de uso	
Pos condiciones	El usuario se ha autenticado satisfactoriamente.
Prioridad	Crítico

Tabla 3-3: Descripción del CU Gestionar Noticia

Caso de uso	Gestionar noticia
Actor	Secretario de la sección sindical
Propósito	Gestionar las noticias nacionales e internacionales.
Resumen	El caso de uso inicia cuando el Secretario de la sección sindical desea mostrar, adicionar, modificar o eliminar las noticias.
Referencias	RF23, RF24, RF25, RF26
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema. El usuario debe tener los permisos necesarios para hacer la operación.
Curso Normal de Eventos	
Sección 1. Mostrar Noticias	
Acción del actor	Respuesta del sistema
	1- El sistema muestra un formulario con un listado de noticias a seleccionar.
2-El secretario de la sección sindical selecciona noticia.	3-El sistema muestra la noticia seleccionados.
Curso Alterno	
Línea 2 2.1- Si no hay un noticia seleccionada el sistema mostrara mensaje “seleccione noticia que desea ver” Regresa a la línea 2 de la sección 1. * El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.	
Sección 2. Adicionar Noticias	
Acción del actor	Respuesta del sistema
	4- El sistema muestra listado de noticias.
5-El secretario de la sección sindical presiona el botón de adicionar nueva noticia.	6-El sistema muestra formulario con los campos a llenar.
7-El secretario de la sección sindical llena los campos de formulario y presiona el botón aceptar.	8-El sistema valida y registra los nuevos datos.
Curso Alterno	

<p>Línea 8. 8.1-El sistema detecta errores en los datos adicionados y lo notifica al secretario de la sección sindical solicitando que sean rectificadas. 8.2-El secretario de la s/s arregla los errores y solicita guardar los datos rectificadas. Regresar a la línea 8 de la sección 2. * El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos. Termina el caso de uso.</p>	
Sección 3. Eliminar Noticias	
Acción del actor	Respuesta del sistema
	9- El sistema muestra formulario con el listado de noticias a seleccionar.
10-El secretario de la sección sindical selecciona el noticia y presiona el botón eliminar.	11-El sistema elimina la noticia seleccionada.
Curso Alterno	
<p>Línea 13 10.1-Si no hay una noticia seleccionada el sistema muestra el mensaje: Debe seleccionar una noticia. Regresa a la línea 10 de la sección 3. * El usuario selecciona Cancelar en cualquier momento y el sistema no elimina los datos. Termina el caso de uso.</p>	
Sección 4. Modificar Noticias	
Acción del actor	Respuesta del sistema
	12- El sistema permite seleccionar noticias a modificar.
13-El secretario de la s/s selecciona noticia y presiona botón modificar.	14-El sistema muestra formulario con noticia a modificar. 15-El sistema registra y valida los nuevos datos entrados.
Curso alterno	
<p>Línea 15 15.1-Si los datos no son válidos el sistema muestra un mensaje Datos no válidos. Regresa a la línea 15 de la sección 4. * El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.</p>	
Pos condiciones	Las noticias han sido gestionadas (adicionadas, modificadas, eliminadas o mostradas) correctamente.
Prioridad	Crítico.

Tabla 3-4 Descripción del CU Gestionar afiliado más integral.

Caso de uso		Gestionar afiliado más integral	
Actor	Secretario de la sección sindical.		
Propósito	Gestionar afiliado más integral de la sección sindical.		
Resumen	El caso de uso inicia cuando el Secretario de la sección sindical desea mostrar, adicionar, modificar o eliminar afiliado más integral.		
Referencias	RF12, RF13, RF14, RF15		
Precondiciones	<p>El actor se ha identificado y autenticado correctamente ante el sistema.</p> <p>El usuario debe tener los permisos necesarios para hacer la operación.</p> <p>Se deben haber seleccionado con anterioridad y de manera correcta por los secretarios de los comités sindicales los afiliados más destacados por comité sindical.</p>		
Curso Normal de Eventos			
Sección 1. Mostrar afiliado más integral			
Acción del actor	Respuesta del sistema		
	1- El sistema muestra un formulario con afiliado más integral a seleccionar.		
2-El secretario selecciona afiliado más integral	3-El sistema muestra afiliado más integral seleccionado.		
Curso Alterno			
Línea 2			
2.1- Si no hay afiliado más integral seleccionado el sistema mostrara mensaje “seleccione afiliado más integral que desea ver”			
Regresa a la línea 2 de la sección 1.			
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.			
Sección 2. Adicionar afiliado más integral			

Acción del actor	Respuesta del sistema
	4- El sistema muestra listado con el afiliado más integral
5-El secretario presiona el botón de adicionar nuevo afiliado más integral.	6-El sistema muestra formulario con los campos a llenar.
7-El secretario llena los campos de formulario y presiona el botón aceptar.	8-El sistema valida y registra los nuevos datos.
Curso Alterno	
<p>Línea 8.</p> <p>8.1-El sistema detecta errores en los datos adicionados y lo notifica al secretario solicitando que sean rectificadas.</p> <p>8.2-El secretario arregla los errores y solicita guardar los datos rectificadas.</p> <p>Regresar a la línea 8 de la sección 2.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos. Termina el caso de uso.</p>	
Sección 3. Modificar afiliado más integral	
Acción del actor	Respuesta del sistema
	9- El sistema muestra formulario con afiliado más integral a seleccionar.
10-El secretario selecciona afiliado más integral y presiona el botón eliminar.	11-El sistema elimina afiliado más integral.
Curso Alterno	
<p>Línea 10</p> <p>10.1-Si no hay afiliado más integral seleccionado el sistema muestra el mensaje:"Debe seleccionar afiliado más integral".</p> <p>Regresa a la línea 10 de la sección 3.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no elimina los datos. Termina el caso de uso.</p>	

Sección 4. Eliminar afiliado más integral	
Acción del actor	Respuesta del sistema
	12-El sistema permite seleccionar actividades extra laborales a modificar.
13- El secretario selecciona actividad y presiona botón modificar	14- El sistema muestra formulario con actividad a modificar 15- El sistema registra y valida los nuevos datos entrados.
Curso Alterno	
Línea 15	
15.1-Si los datos no son válidos el sistema muestra un mensaje “Datos no válidos”.	
Regresa a la línea 15 de la sección 4.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos .Termina el caso de uso.	
Pos condiciones	El afiliado más integral ha sido gestionado (adicionado, eliminado, mostrado o modificado) correctamente.
Prioridad	Crítico.

Tabla 3-5 Descripción del CU Gestionar comité sindical más integral

Caso de uso	Gestionar comité sindical integral
Actor	Secretario de la sección sindical.
Propósito	Gestionar comité sindical integral de la sección sindical.
Resumen	El caso de uso inicia cuando el Secretario de la sección sindical desea mostrar, adicionar, modificar o eliminar un comité sindical integral.
Referencias	RF16, RF17, RF18,RF19
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema. El usuario debe tener los permisos necesarios para hacer la operación.

Se deben haber seleccionado con anterioridad los comités sindicales cumplidores.	
Curso Normal de Eventos	
Sección 1. Mostrar comité sindical	
Acción del actor	Respuesta del sistema
	1- El sistema muestra un formulario con los comités sindicales a seleccionar.
2-El secretario selecciona comité sindical.	3-El sistema muestra comité sindical seleccionado.
Curso Alterno	
Línea 2	
2.1- Si no hay comité sindical seleccionado el sistema mostrara mensaje “seleccione comité sindical que desea ver”	
Regresa a la línea 2 de la sección 1.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.	
Sección 2. Adicionar comité sindical	
Acción del actor	Respuesta del sistema
	4- El sistema muestra formulario con opción de adicionar comité sindical más integral.
5-El secretario presiona el botón de adicionar nuevo comité sindical.	6-El sistema muestra formulario con los campos a llenar.
7-El secretario llena los campos de formulario y presiona el botón aceptar.	8-El sistema valida y registra los nuevos datos.
Curso Alterno	
Línea 8.	
8.1-El sistema detecta errores en los datos adicionados y lo notifica al secretario solicitando que sean rectificadas.	
8.2-El secretario arregla los errores y solicita guardar los datos rectificadas.	

<p>Regresar a la línea 8 de la sección 2.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos. Termina el caso de uso.</p>	
<p>Sección 3. Eliminar comité sindical</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
	<p>9- El sistema muestra formulario con la opción de eliminar comité sindical integral</p>
<p>10-El secretario selecciona comité sindical y presiona el botón eliminar.</p>	<p>11-El sistema elimina comité sindical.</p>
<p>Curso Alterno</p>	
<p>Línea 10</p> <p>10.1-Si no hay comité sindical seleccionado el sistema muestra el mensaje:”Debe seleccionar comité sindical”.</p> <p>Regresa a la línea 10 de la sección 3.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no elimina los datos. Termina el caso de uso.</p>	
<p>Sección 4. Modificar comité sindical</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
	<p>12- El sistema permite seleccionar comité sindical a modificar.</p>
<p>13-El secretario de la s/s selecciona comité sindical y presiona botón modificar.</p>	<p>14-El sistema muestra formulario con la información del comité sindical a modificar.</p> <p>15-El sistema registra y valida los nuevos datos entrados</p>
<p>Curso alternativo</p>	
<p>Línea 15</p> <p>15.1-Si los datos no son válidos el sistema muestra un mensaje Datos no válidos.</p> <p>Regresa a la línea 15 de la sección 4.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.</p>	
<p>Pos condiciones</p>	<p>Ha sido gestionado (adicionado, eliminado,</p>

	modificado o mostrado) correctamente el comité sindical más integral.
Prioridad	Crítico.

Tabla 3-6 Descripción del CU Gestionar usuario

Caso de uso	Gestionar usuario
Actor	Administrador.
Propósito	Permitir al actor gestionar los usuarios que van a interactuar con el sistema, así como encontrar el usuario deseado producto de una búsqueda.
Resumen	El caso de uso se inicia cuando el administrador necesita gestionar una cuenta de usuario (adicionar nueva cuenta, eliminarla, modificarla, o mostrarla para que otro usuario pueda utilizarla con el objetivo de trabajar con el sistema. El caso de uso concluye cuando finalizan cualquiera de los escenarios de la gestión de la cuenta escogida por el actor.
Referencias	RF36, RF37, RF38,RF39
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema. El usuario debe tener los permisos necesarios para hacer la operación.
Curso Normal de Eventos	
Sección 1. Mostrar usuario	
Acción del actor	Respuesta del sistema
1. Llena los criterios de búsqueda y ordena buscar	2- Muestra el listado de los resultados según los criterios especificados, permitiendo así eliminar cualquiera de las respuestas obtenidas.
Curso Alterno	
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.	

Sección 2. Adicionar usuario	
Acción del actor	Respuesta del sistema
3. Escoge la opción de adicionar una nueva cuenta.	4. Habilita el formulario para llenar los datos de la cuenta, brindando la posibilidad de escoger el grupo al que va a pertenecer.
5. Llena el formulario. 7. Escoge el grupo al que va a pertenecer el usuario.	6. Muestra la lista de las accesibilidades asociadas al grupo que el usuario seleccionó.
8-El secretario llena los campos de formulario y presiona el botón aceptar.	9-El sistema valida y registra los nuevos datos.
10. Acepta el usuario.	11-. Guarda los datos y habilita la cuenta.
Curso Alterno	
<p>Línea 9.</p> <p>9.1-El sistema detecta errores en los datos adicionados y lo notifica al administrador solicitando que sean rectificadas.</p> <p>9.2-El administrador arregla los errores y solicita guardar los datos rectificadas.</p> <p>Regresar a la línea 9 de la sección 2.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos. Termina el caso de uso.</p>	
Sección 3. Eliminar usuario	
Acción del actor	Respuesta del sistema
12-Selecciona en los elementos del resultado de la búsqueda la opción de eliminar una cuenta de usuario específica.	13-. Muestra un cartel anunciando que la cuenta será borrada.
14-Ordena aceptar la eliminación.	15-Borra la cuenta de la base de datos y muestra el resultado.
Curso Alterno	

<p>Línea 12</p> <p>10.1-Si no hay una cuenta de usuario seleccionada el sistema muestra el mensaje:”Debe seleccionar cuenta de usuario”.</p> <p>Regresa a la línea 12 de la sección 3.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no elimina los datos. Termina el caso de uso.</p>	
<p>Sección 4. Modificar usuario</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>16- Selecciona en los elementos del resultado de la búsqueda la opción de modificar una cuenta de usuario específica.</p>	<p>17-Muestra un formulario con la cuenta a modificar.</p>
<p>18-El usuario modifica los datos y presiona el botón aceptar.</p>	<p>19-El sistema guarda y valida los datos.</p>
<p>Curso alterno</p>	
<p>Línea 16</p> <p>16.1-Si no hay cuenta de usuario seleccionada el sistema muestra el mensaje:”Debe seleccionar cuenta de usuario”.</p> <p>Regresa a la línea 16 de la sección 4.</p>	
<p>Línea 19</p> <p>19.1-El sistema detecta errores en los datos modificados y lo notifica al administrador solicitando que sean rectificadas.</p> <p>19.2-El administrador arregla los errores y solicita guardar los datos rectificadas.</p> <p>Regresar a la línea 19 de la sección 4.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no elimina los datos. Termina el caso de uso.</p>	
<p>Pos condiciones</p>	<p>Los usuarios han sido gestionados (adicionados, eliminados o mostrados) correctamente.</p>
<p>Prioridad</p>	<p>Secundario</p>

3.4. Modelo de clases de análisis

El modelo de análisis se centra en el tratamiento de requisitos funcionales y pospone los no funcionales para el diseño. Esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluidas las funcionalidades significativas para la arquitectura, sirve como una primera aproximación del diseño y es el resultado de la actividad de analizar los casos de uso.

En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el diagrama de clases del análisis, estas representan los conceptos en un dominio del problema, sin llegar al nivel de detalle necesario que se persigue en el diseño, constituyendo una primera aproximación a este. (Kendall, 2005)

3.4.1. Diagramas de clases del análisis.

Los diagramas de clases del análisis representan la relación entre las clases, las cuales se centran en los requisitos funcionales, tienen atributos y entre ellas se establecen relaciones de asociación, agregación/composición, generalización/especialización, y tipos asociativos. RUP propone clasificar las clases en: interfaz, controladora y entidad.

- Las clases interfaz modelan la interacción entre el sistema y sus actores.
- Las clases controladoras coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.
- Las clases de entidad modelan información que posee larga vida y que es a menudo persistente. (Kendall, 2005)

A continuación se muestran los diagramas de clases del análisis: Autenticar usuario, Gestionar noticia, Gestionar usuario, Gestionar afiliado más integral y Gestionar comité sindical más integral. En el Anexo 2 podrán encontrarse otros de los restantes diagramas de clases del análisis.

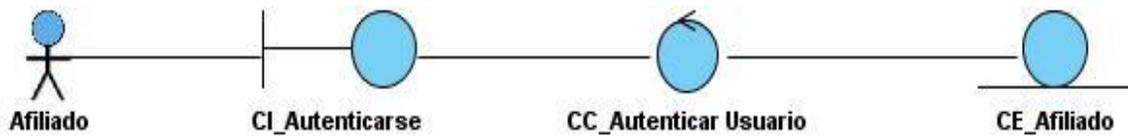


Figura 3-2 DCA del CU Autenticar Usuario.

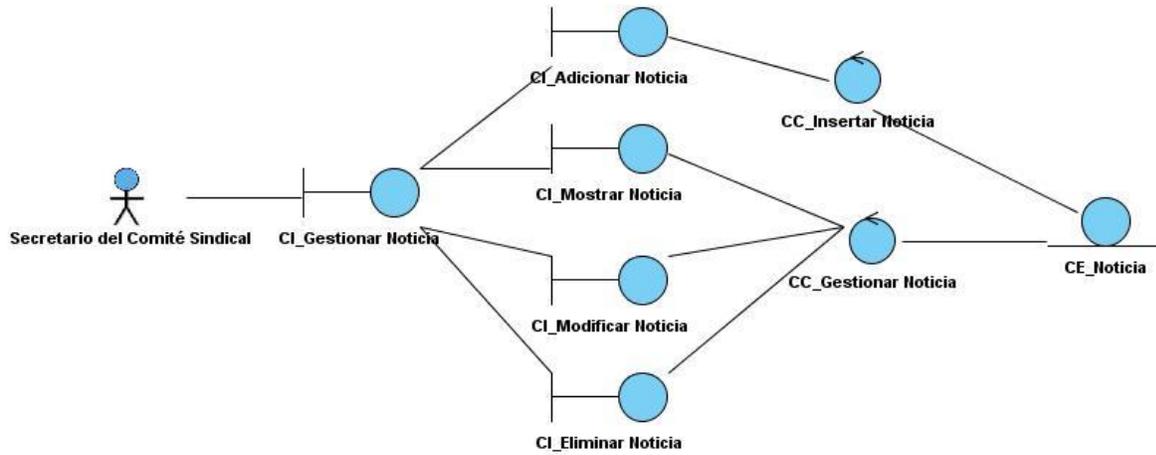


Figura 3-3 DCA del CU Gestionar noticia

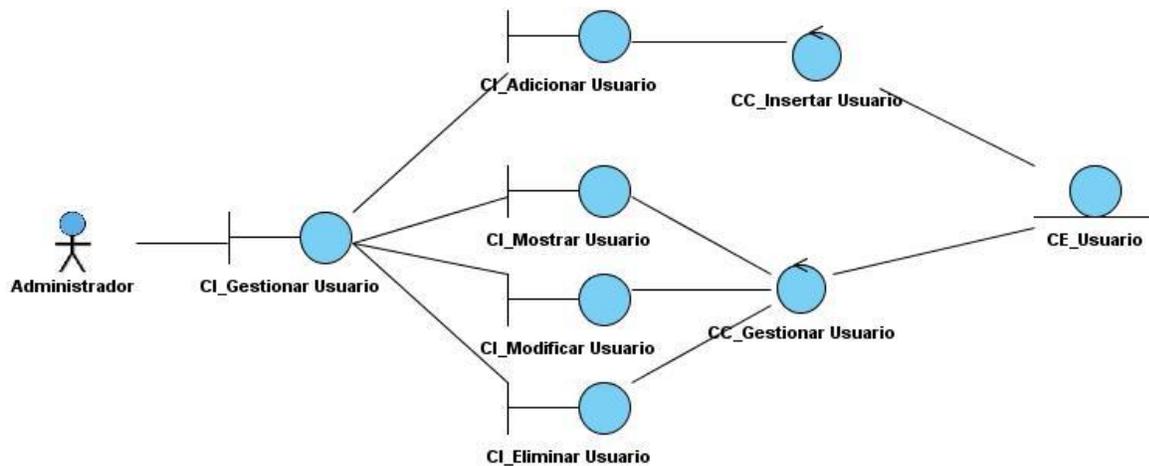


Figura 3-4 DCA del CU Gestionar usuario.

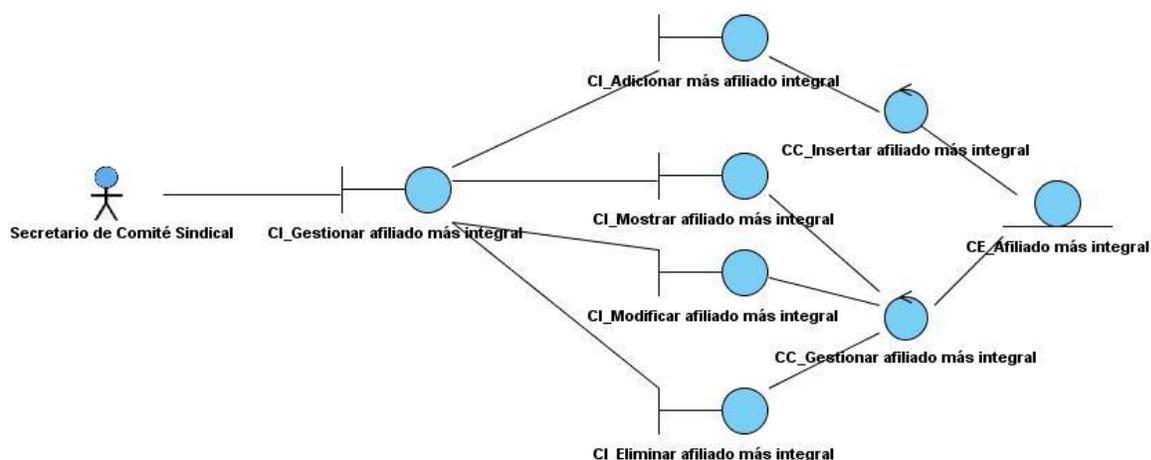


Figura 3-5 DCA del CU Gestionar afiliado más integral.

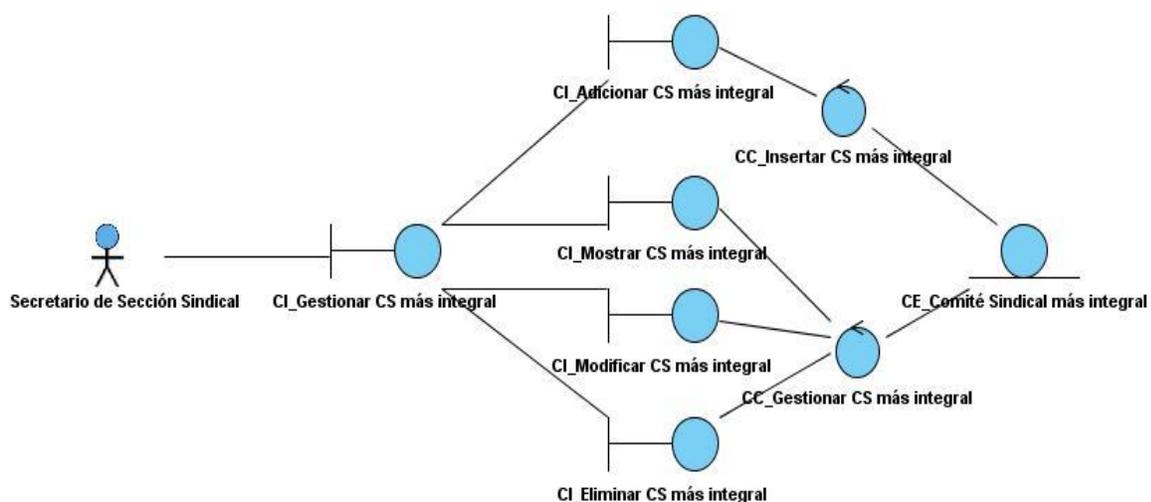


Figura 3-6 DCA del CU Gestionar c/s más integral.

3.4.2. Diagramas de interacción (colaboración).

Se presentarán los diagramas de interacción de los casos de uso Autenticar usuario, Gestionar noticia, Gestionar usuario, Gestionar comité sindical más integral y Gestionar afiliado más integral, divididos por escenarios, lo cual permitirá una mejor comprensión de los mismos. En el Anexo 3 se podrán encontrar otros de los restantes diagramas de interacción.

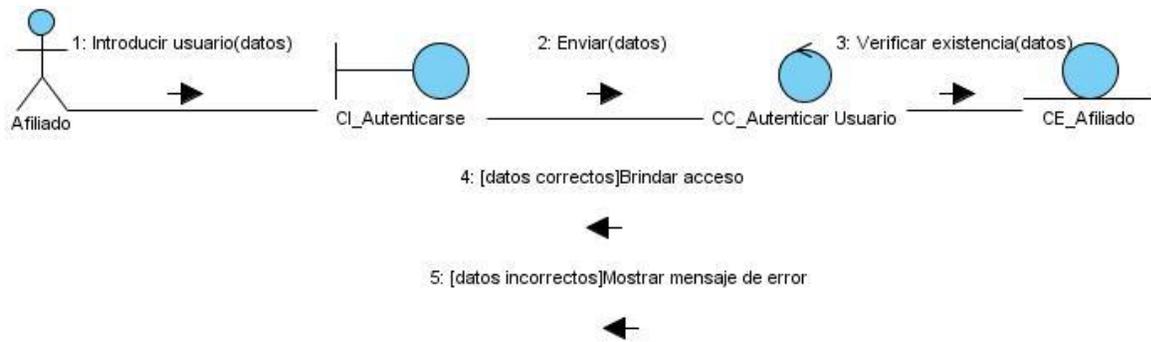


Figura 3-7 Diag. Interacción del CU Autenticar usuario

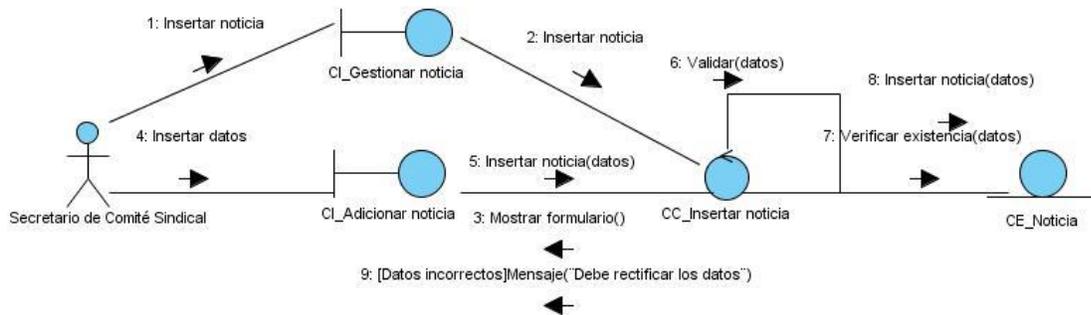


Figura 3-8 Diag. Interacción del CU Adicionar noticia

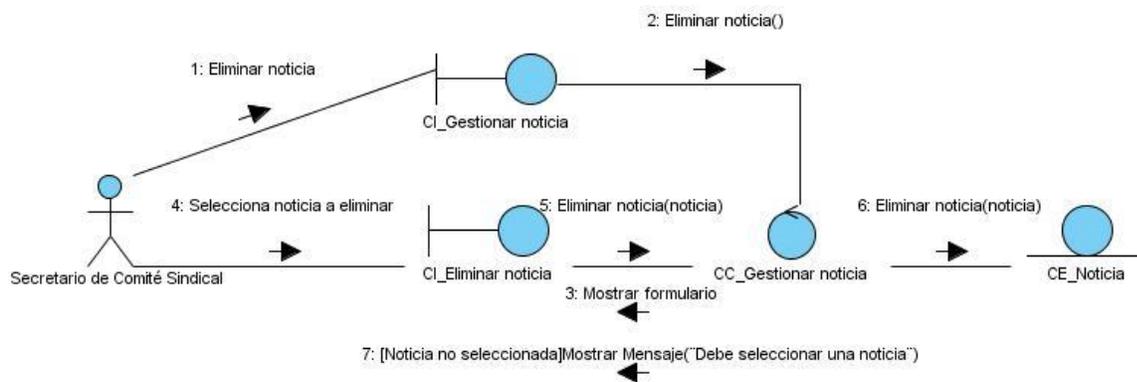


Figura 3-9 Diag. Interacción del CU Eliminar noticia

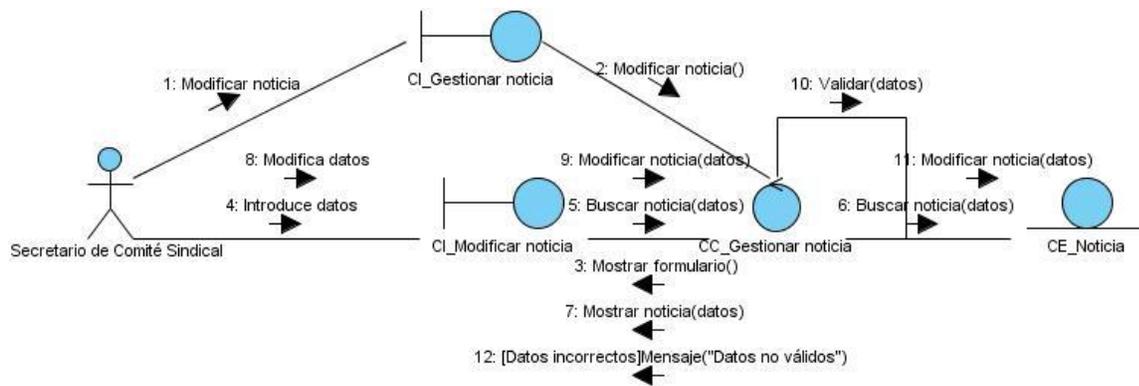


Figura 3-10 Diag. Interacción del CU Modificar noticia

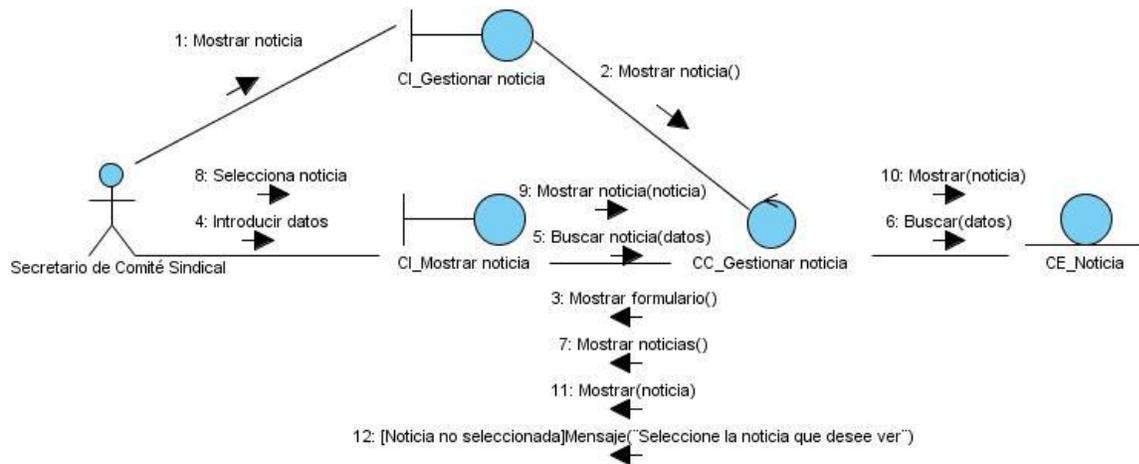


Figura 3-11 Diag. Interacción del CU Mostrar Noticia.

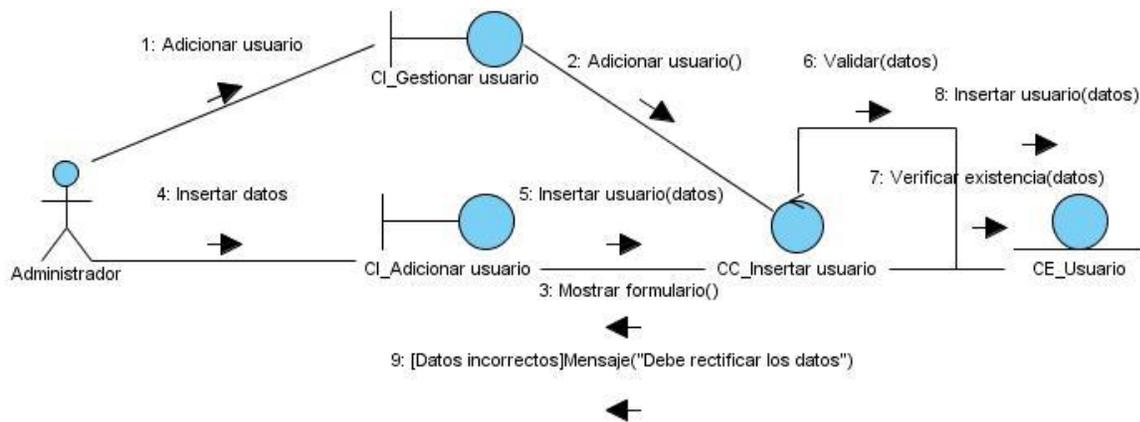


Figura 3-12 Diag. Interacción del CU Adicionar Usuario.

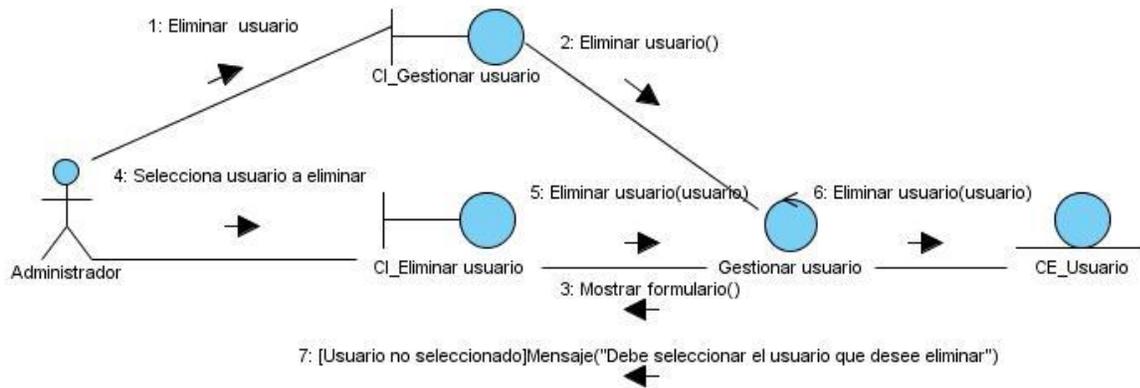


Figura 3-13 Diag. Interacción del CU Eliminar Usuario.

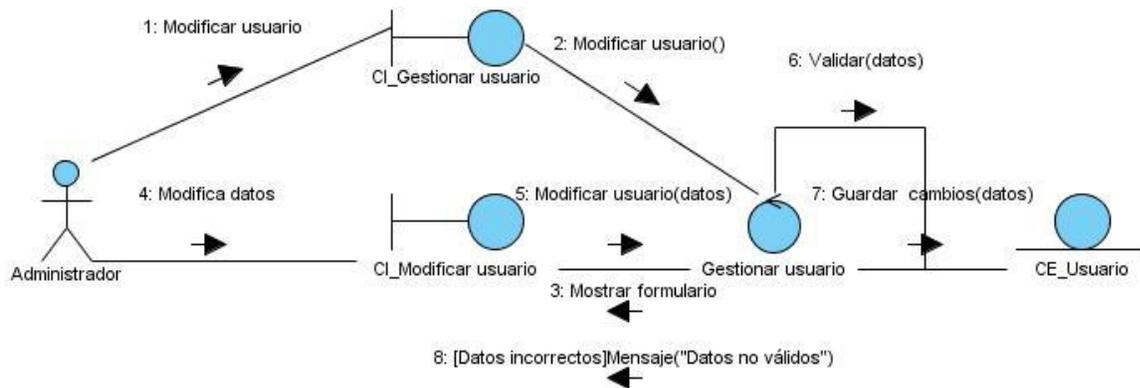


Figura 3-14 Diag. Interacción del CU Modificar Usuario.

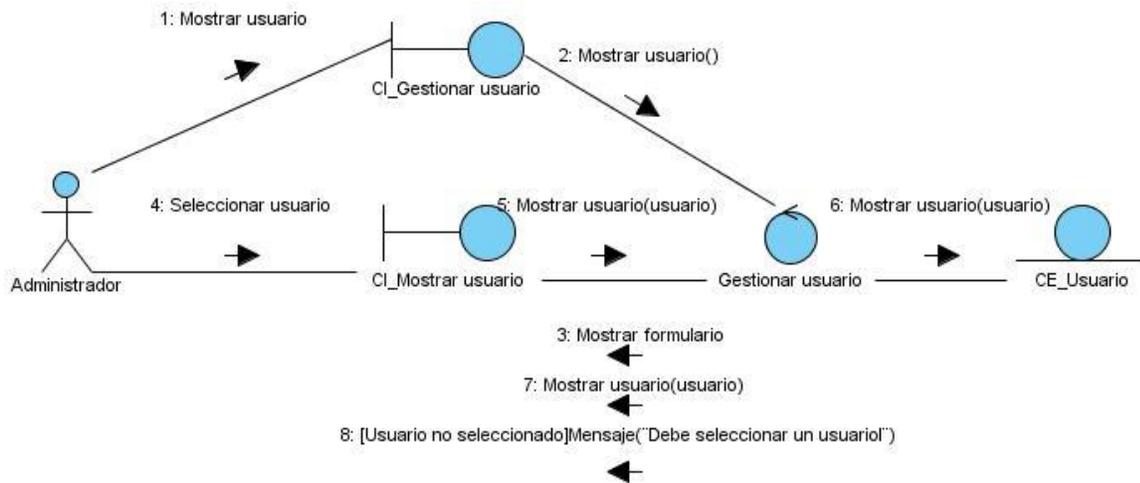


Figura 3-15 Diag. Interacción del CU Mostrar Usuario.

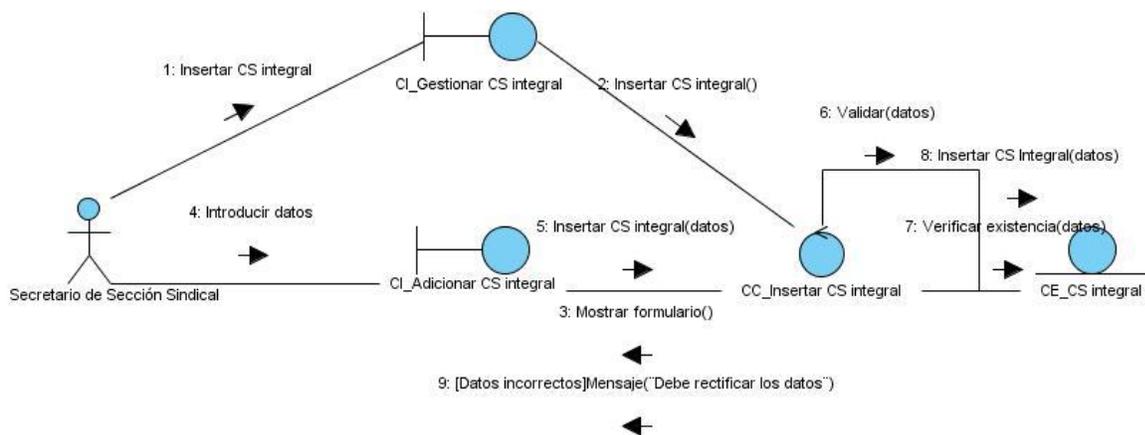


Figura 3-16 Diag. Interacción del CU Adicionar c/s más integral.

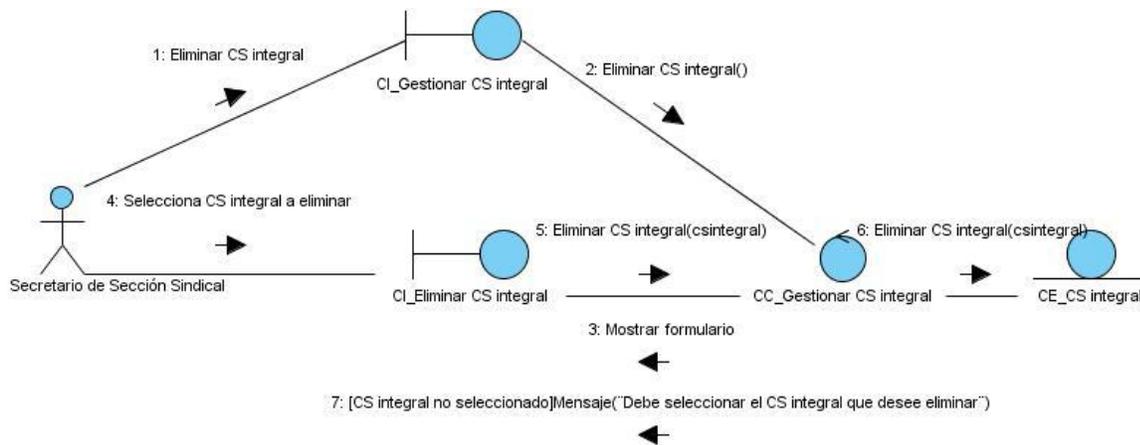


Figura 3-17 Diag. Interacción del CU Eliminar c/s más integral.

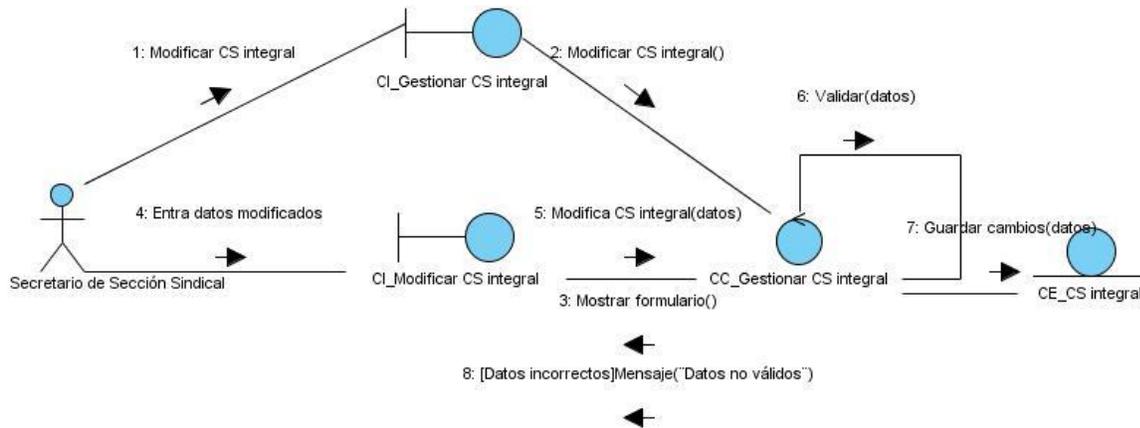


Figura 3-18 Diag. Interacción del CU Modificar c/s más integral.

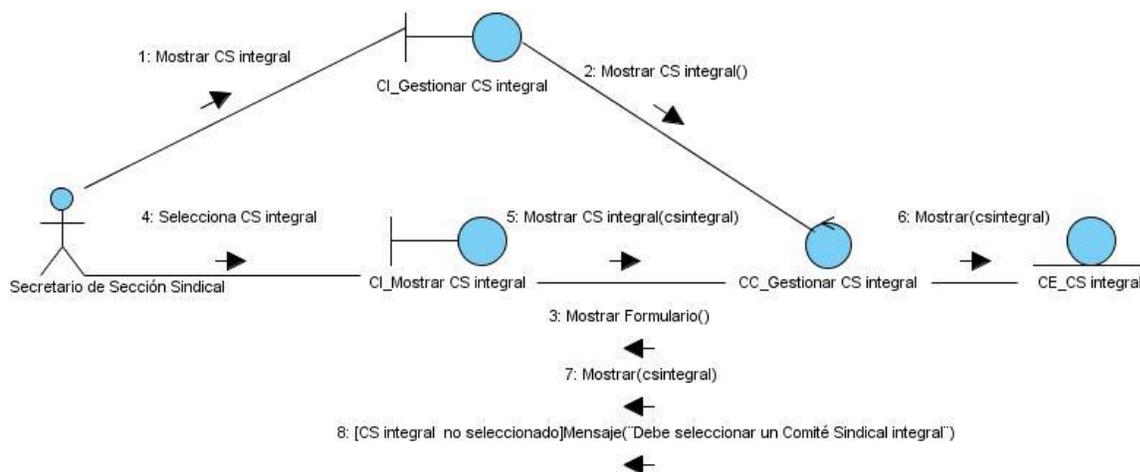


Figura 3-19 Diag. Interacción del CU Mostrar c/s más integral.

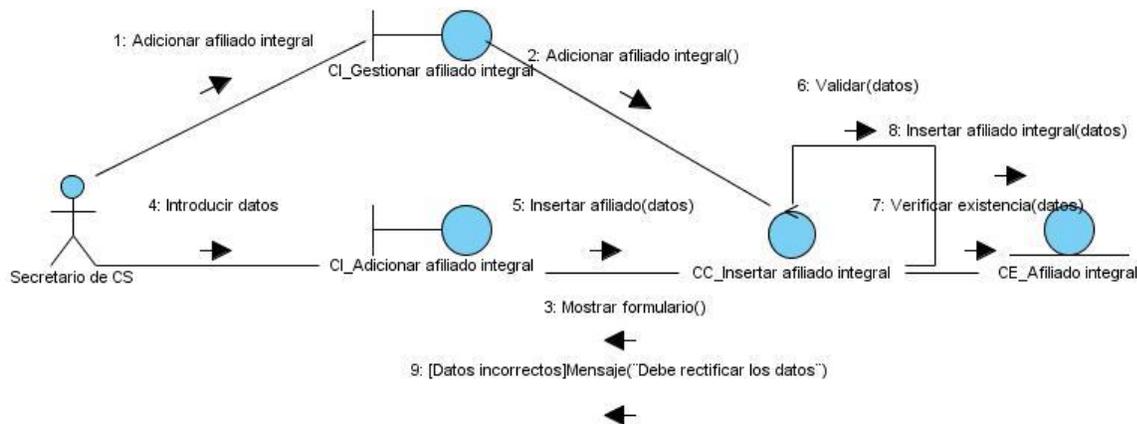


Figura 3-20 Diag. Interacción del CU Adicionar afiliado más integral.

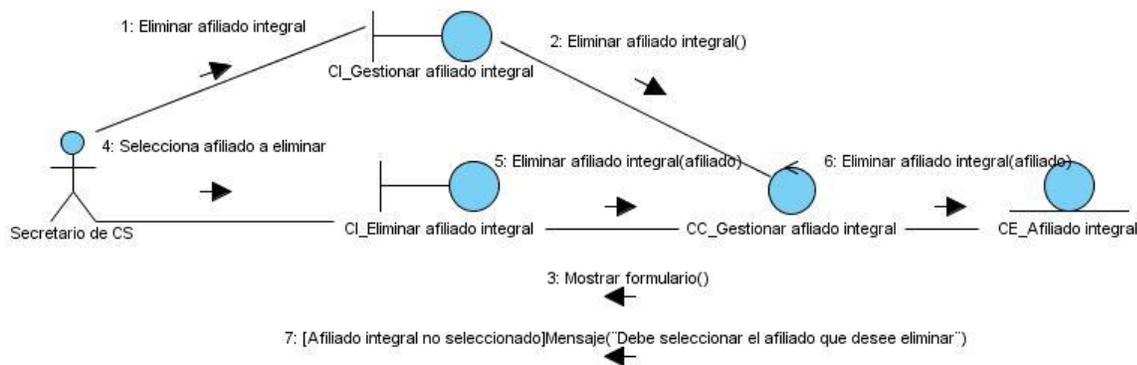


Figura 3-21 Diag. Interacción del CU Eliminar afiliado más integral

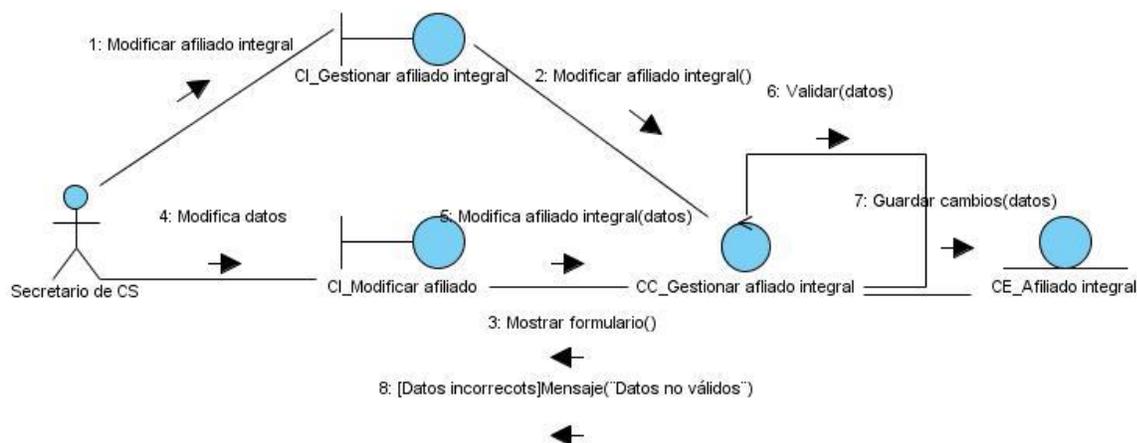


Figura 3-22 Diag. Interacción del CU DC Modificar afiliado más integral.

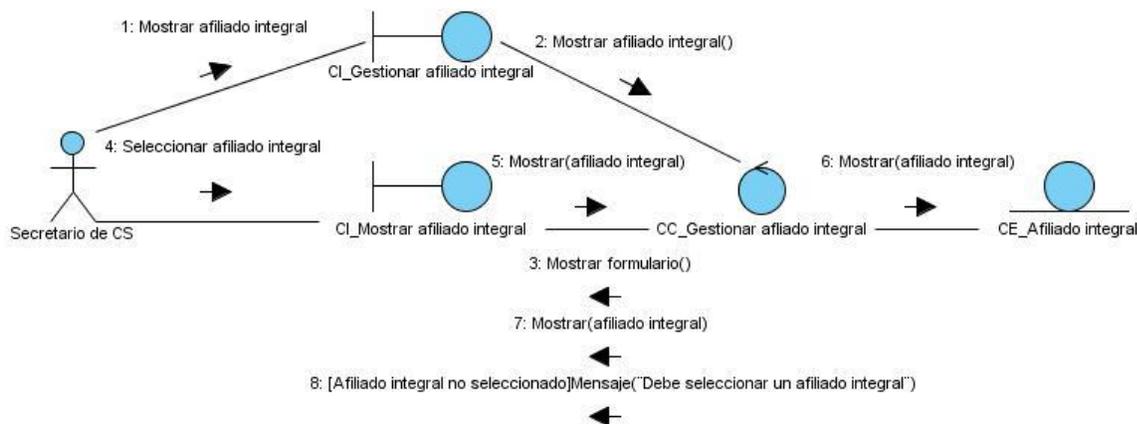


Figura 3-23 Diag. Interacción del CU Mostrar afiliado más integral.

3.5. Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso. Se centra en como los requisitos funcionales y no funcionales tienen impacto en el sistema a desarrollar. Dentro de sus propósitos están: crear una entrada apropiada y un punto de partida para la implementación, descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo y adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones vinculadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia.

3.5.1. Patrones de diseño

Los Patrones de Diseño posibilitan la reutilización del conocimiento, su utilización como guía para el desarrollo de software, garantiza un producto robusto y acabado.

Por ello, durante el diseño del Proceso de Administración y Control de la Información para la Sección Sindical VP se proponen utilizar patrones generales para la asignación de responsabilidades o patrones GRASP (Edith, 2008), necesarios para construir eficazmente un software orientado a objetos pues estos constituyen principios básicos a tener en cuenta. En el diseño propuesto se utilizaron los patrones: Experto, Creador, Controlador, Bajo Acoplamiento y Alta Cohesión.

Se realizó un estudio de los patrones de la pandilla de los cuatro o GOF, de manera que con la utilización de dichos patrones se contribuya de manera eficiente con la posterior implementación, de ellos fue seleccionado: Singleton.

Con este patrón se garantiza una única instancia de aquellas clases que se desee tener una sola en toda la aplicación, proporcionando un punto de acceso global a dichas clases. Tiene como ventajas que reduce el espacio de nombres y es una mejora sobre las variables globales. Es usado debido a la necesidad de trabajar con el mismo objeto en distintos momentos y distintos subsistemas. (Edith, 2008)

3.5.2. Diagrama de clases del diseño.

En el diagrama de clases de diseño se muestran los atributos y métodos de cada clase y se representa de una forma sencilla la colaboración y las responsabilidades de las distintas clases que forman el sistema.

Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases también son la base para un par de diagramas relacionados: los diagramas de componentes y los diagramas de despliegue. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

Se presentarán los diagramas de clases del diseño de los casos de uso: Gestionar Usuario, Gestionar Comité Sindical más integral, Gestionar Noticia Gestionar afiliado más integral y Autenticar Usuario. En el Anexo 4 podrán encontrarse otros de los restantes diagramas de clases del diseño.

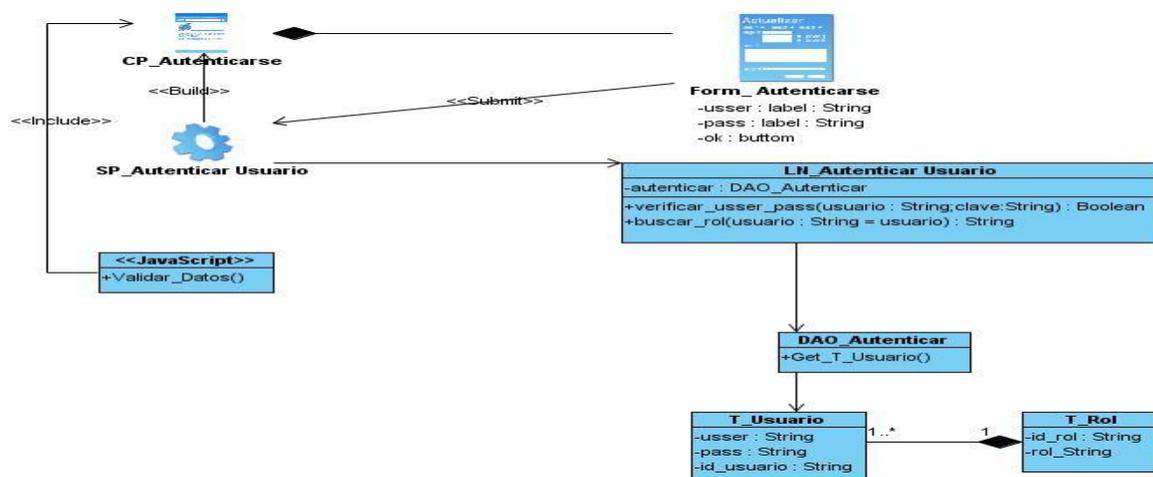


Figura 3-24 DCD del CU Autenticar Usuario

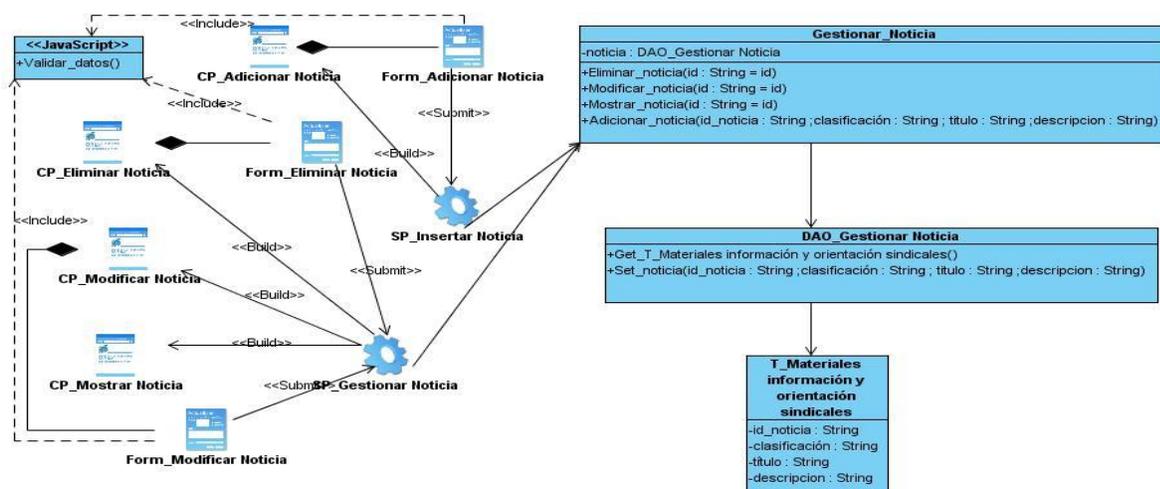


Figura 3-25 DCD del CU Gestionar noticia

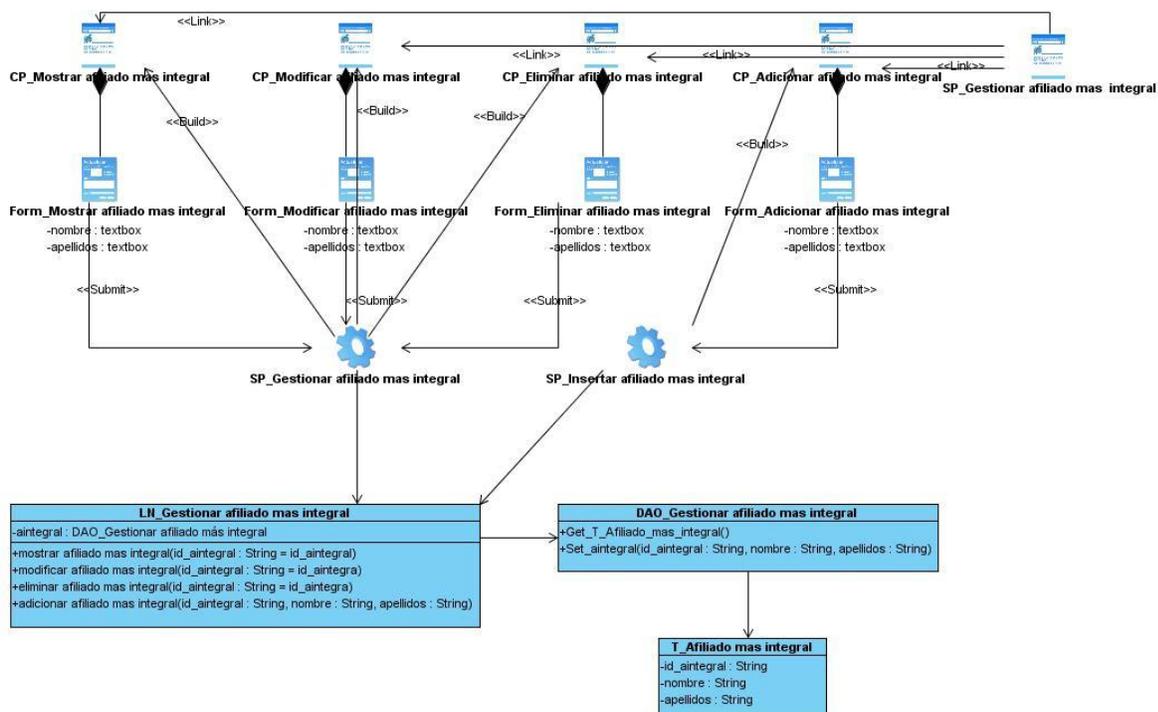


Figura 3-26 Diag. Interacción del CU Gestionar afiliado más integral

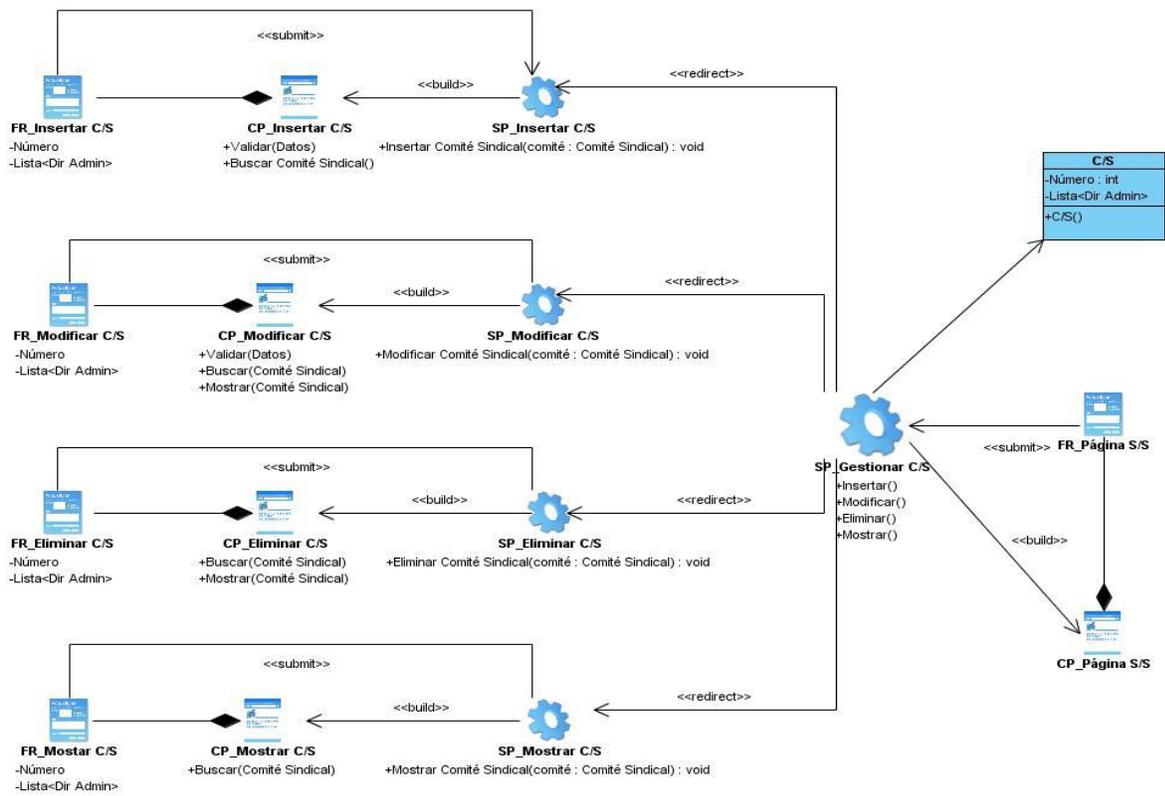


Figura 3-27 DCD del CU Gestionar comité sindical más integral.

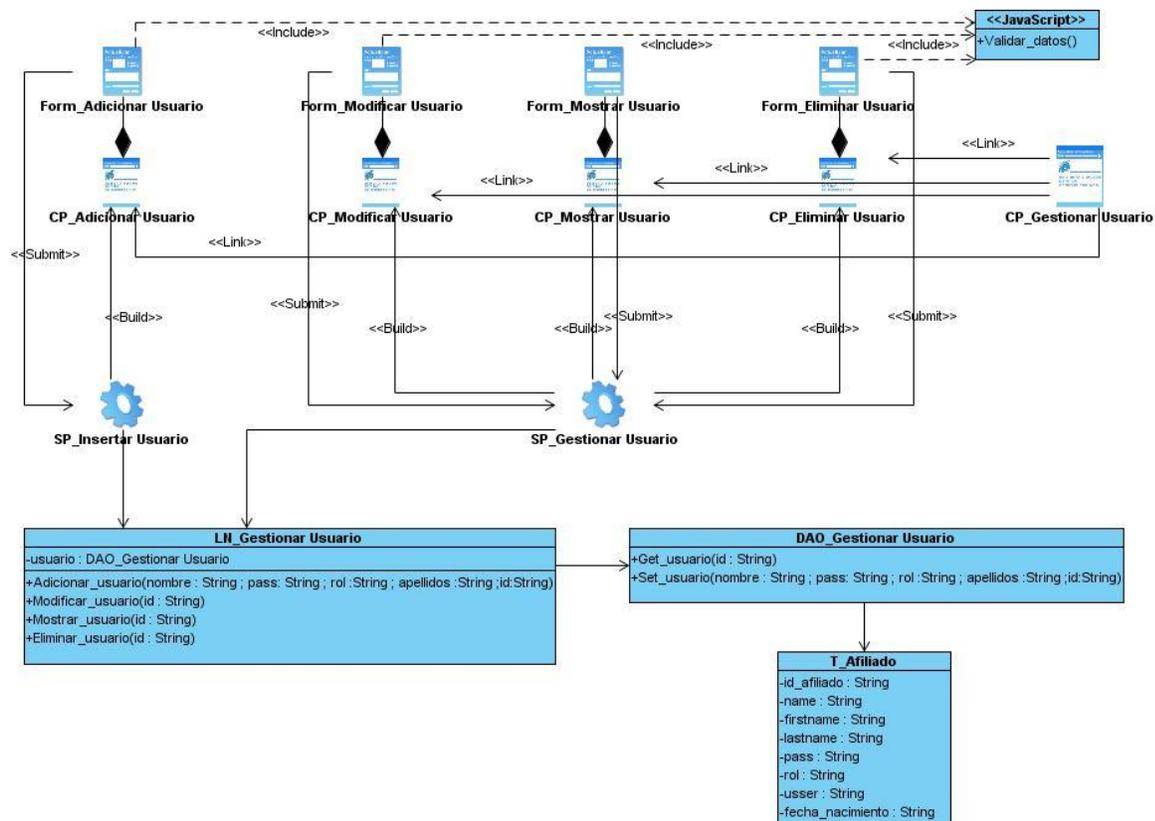


Figura 3-28 DCD del CU Gestionar usuario

3.5.3. Diagrama de interacciones (Secuencia).

Los Diagramas de Interacción se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento.

Los Diagramas de Interacción pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de una sociedad particular de objetos, o se pueden utilizar para modelar un flujo de control particular de un Caso de Uso. (Jacobson, 2004)

Estos diagramas son de dos tipos Diagrama de Secuencia o Colaboración.

Un Diagrama de Secuencia representa las interacciones entre objetos ordenadas en secuencia temporal. Los clientes entienden fácilmente este tipo de diagramas. Contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos. En particular muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados. (Socorro Suárez, y otros, 2007)

Se presentarán los diagramas de interacción del caso de uso: Gestionar Noticia, dividido por escenarios, para lograr una mejor comprensión del mismo. En el Anexo 5 podrán encontrarse otros de los restantes diagramas de interacción.

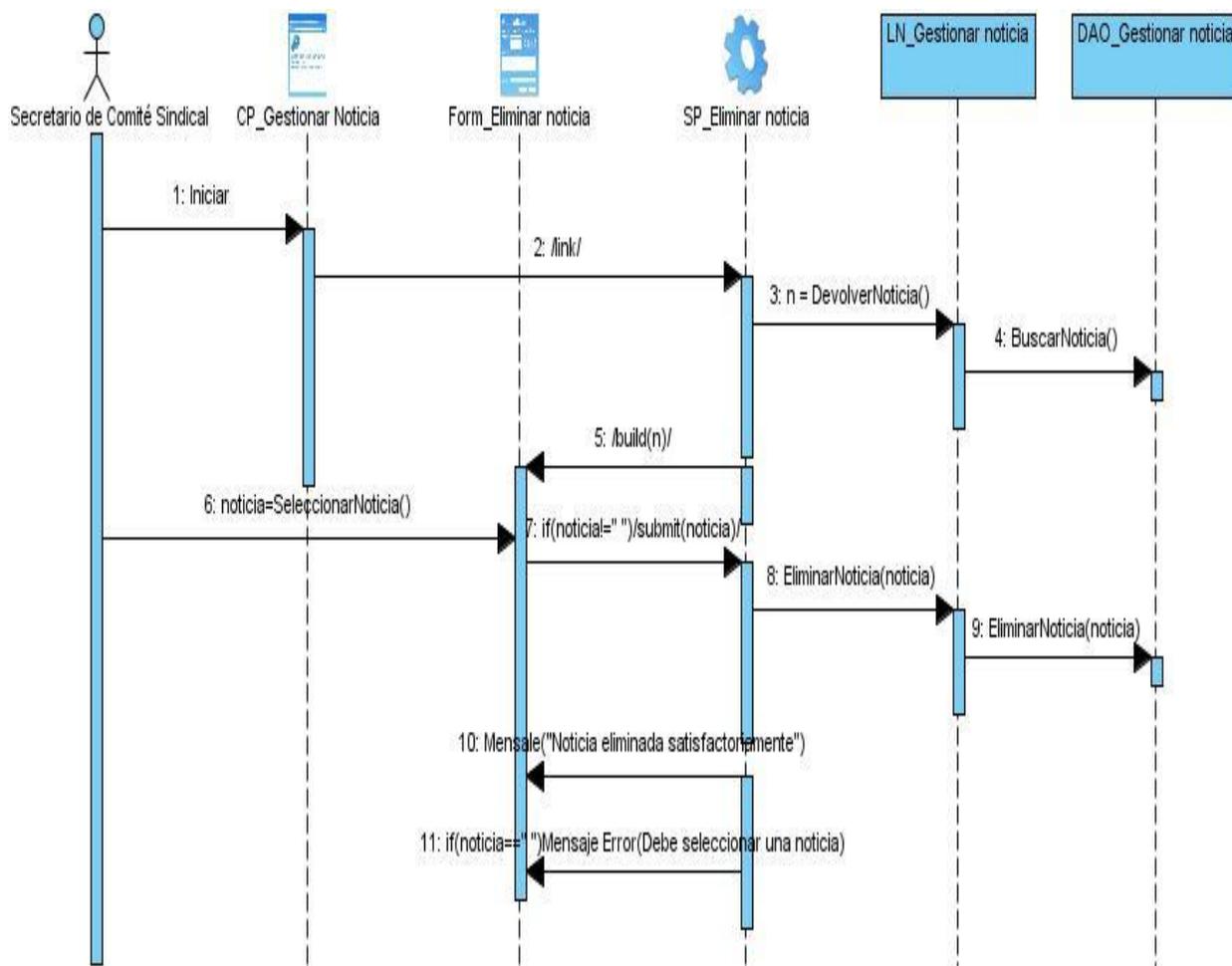


Figura 3-29 DS del CU Eliminar noticia.

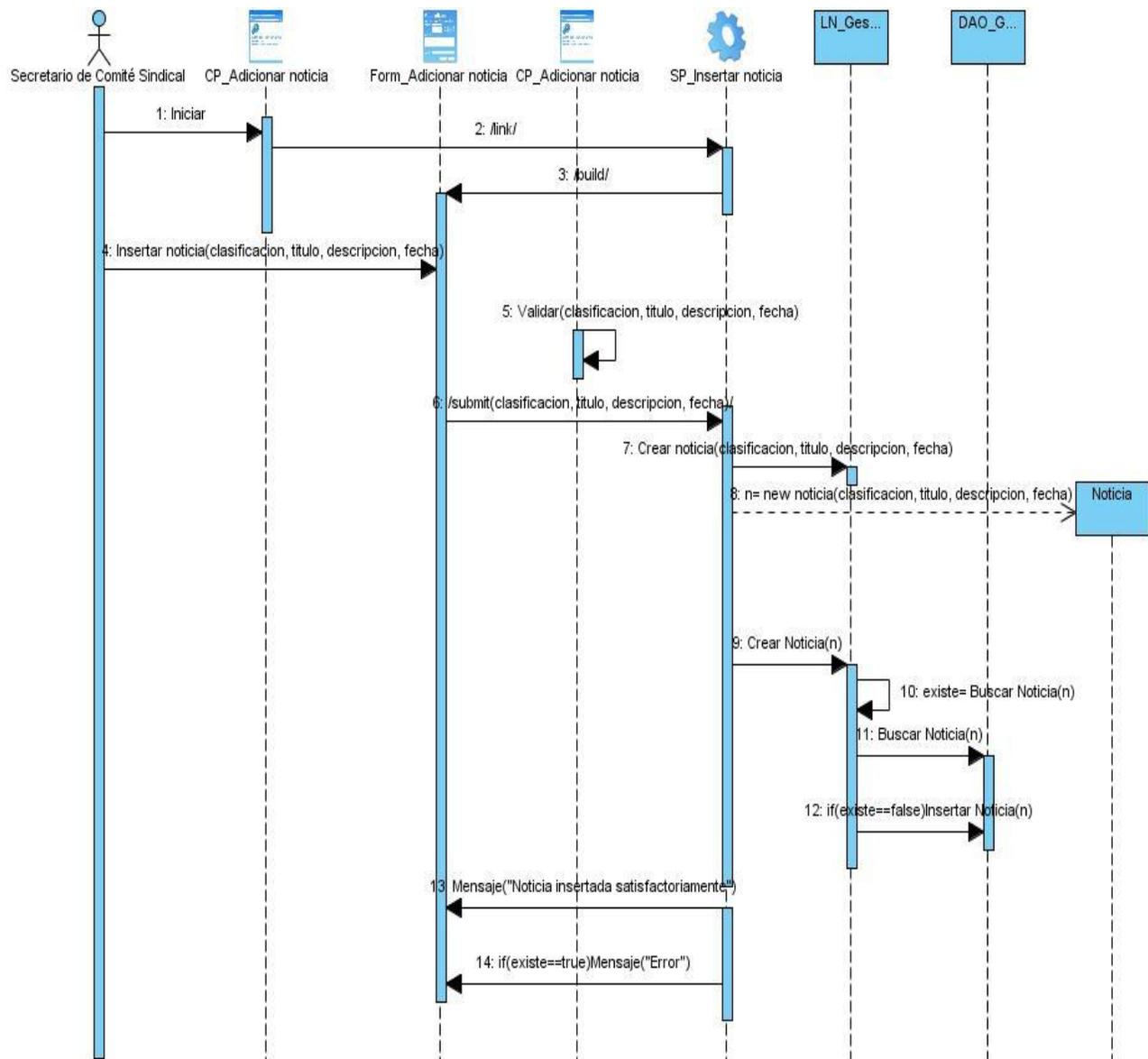


Figura 3-30 DS del CU Adicionar noticia

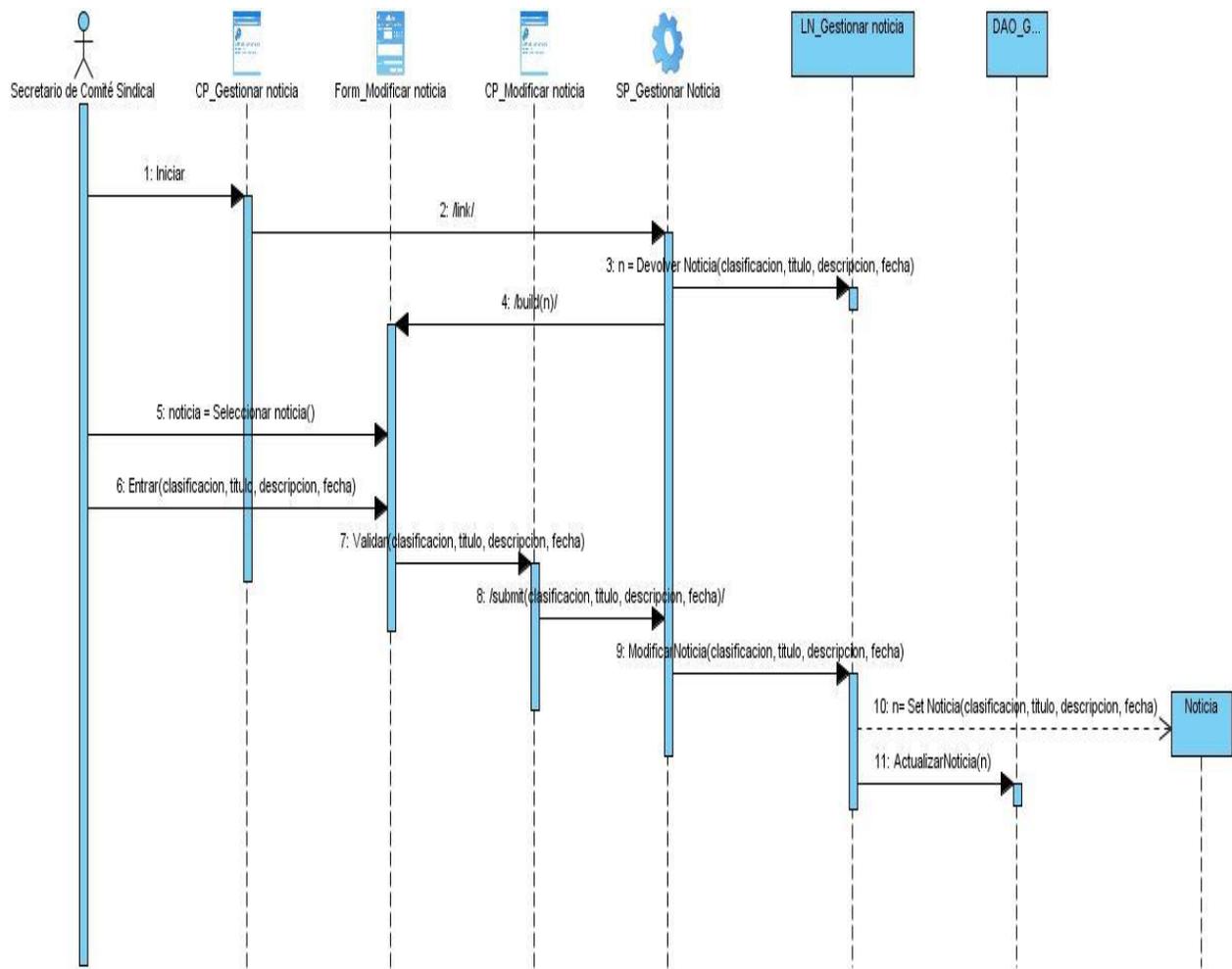


Figura 3-31 DS del CU Modificar noticia

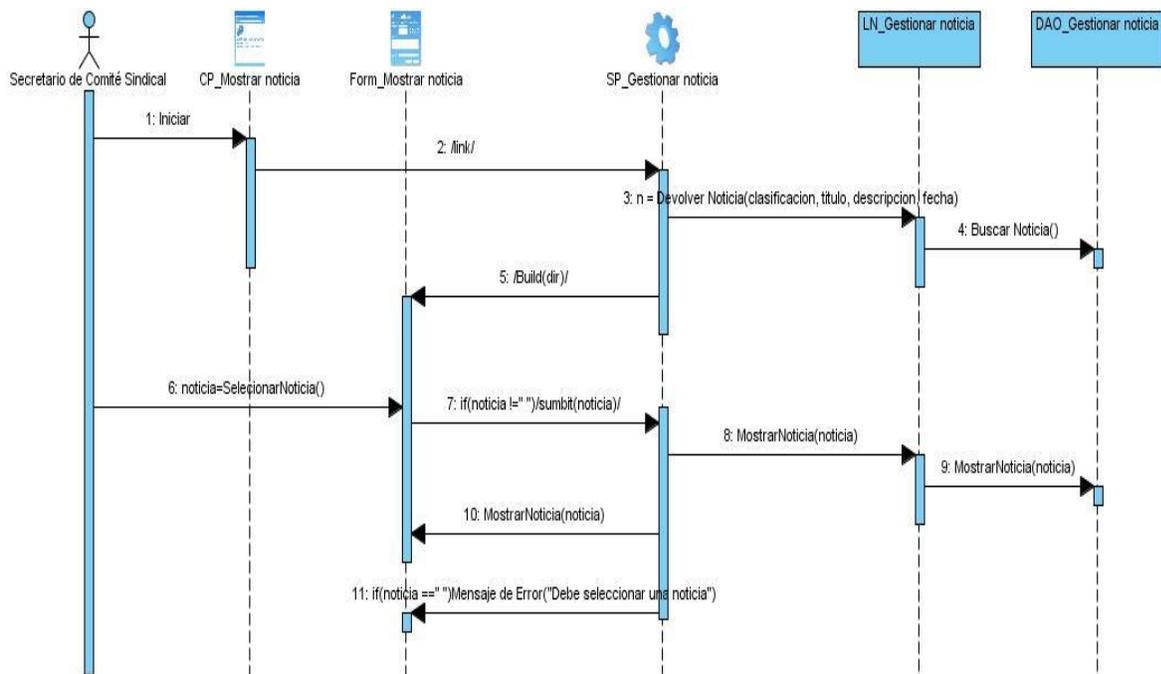


Figura 3-32 DS del CU Mostrar noticia

Conclusiones Parciales del Capítulo 3

En este capítulo se realizó un modelo detallado de la solución propuesta, a través del modelado de los diagramas de clases del análisis y los de interacción por cada realización de caso de uso del sistema que facilitaron una primera aproximación al modelo del diseño, se definieron los actores del sistema y se describieron los casos de uso pertenecientes al sistema. Se modelaron las clases del diseño y los diagramas de interacción pertenecientes al flujo de trabajo de diseño.

Capítulo 4. Validación de los Resultados

Introducción del Capítulo 4

A raíz del trabajo realizado para el análisis y diseño del Proceso de Administración y Control de la Información para la Sección Sindical VP se obtuvo una gran cantidad de artefactos por parte del equipo de trabajo. Para comprobar la efectividad de las actividades y validar dichos artefactos generados se empleará la Métrica de la Calidad de Especificación de los Requisitos, que propone Pressman, las Métricas Orientadas a Clases, El Esquema QFD(Quality Function Deployment), los prototipos de interfaz de usuario, y las Métricas Orientadas a Clases.

La validación por prototipos de interfaz se encuentra en el Anexo # 7.

4.1. Métrica de la Calidad de Especificación de los Requisitos.

Debido a la cantidad de requisitos capturados y las características de los mismos, se va a aplicar la métrica que mide su especificidad.

Para realizar el proceso de medición anteriormente planteado en el capítulo 1, y así proceder a la evaluación se llevó a cabo una revisión y se consultaron revisores, con el objetivo de eliminar ambigüedades y especificar detalladamente los requisitos capturados. Se capturaron cuarenta y siete requisitos funcionales y diecinueve requisitos no funcionales, sesenta y seis en total. Los resultados de la revisión se muestran a continuación:

Después de haber sido aplicada la métrica de especificación de los requisitos, se redujo el nivel de ambigüedad a un 0.01%, por lo que casi la totalidad de los requisitos fueron interpretados de la misma forma: $Q = 65 / 66$, $Q = 0.99$. Se puede concluir que se logró un grado bajo de ambigüedad de forma general, por lo tanto, el nivel de calidad del proceso fue bastante alto. Realizar este tipo de evaluaciones contribuyó a obtener un análisis y un diseño en coherencia con lo que desea la parte interesada.

4.2. Métrica para determinar el número de requisitos que no son considerados en ningún CU.

Al aplicar el esquema QFD en cada caso de uso del sistema, se comprobó que todos los requerimientos tienen al menos una cruz marcada, entonces se pudo comprobar que todos los requerimientos están incluidos en algún caso de uso. Además, después de construir la matriz se concluyó que el error calculado de acuerdo a la métrica y teniendo en cuenta la

cantidad de requerimientos totales y la cantidad de casos de uso del sistema que satisfacen los requerimientos es del 0%.

Para consultar resultados de la métrica ver Anexo 6.

4.3. Métricas Orientadas a Clases.

4.3.1. Tamaño de clases

Para evaluar las métricas son necesarios los umbrales. En este caso las clases se clasifican en tres grupos según su tamaño, los que se representan en la siguiente tabla junto con los umbrales seleccionados para su clasificación. (Edith, 2008)

Tabla 4-1: Valores de los umbrales para TC.

Clasificación	Valores de los umbrales
Pequeño	≤ 20
Medio	> 20 y ≤ 30
Grande	> 30

Tabla 4-2: Tamaño de las Clases.

No.	Nombre	Cantidad Atributos	Cantidad Operaciones	Tamaño
1	L_N Autenticar usuario	1	2	pequeño
2	DAO_Autenticar	0	1	pequeño
3	T_Usuario	3	0	pequeño
4	T_Rol	2	0	pequeño
5	L_N Gestionar actividades extra laborales.	1	4	pequeño
6	DAO _ Gestionar actividades extra laborales.	0	2	pequeño
7	T_Actividad	4	0	pequeño
8	L_N Gestionar actividades sindicales.	1	4	pequeño
9	DAO _ Gestionar	0	2	pequeño

Capítulo IV: Validación de los Resultados

	actividades sindicales.			
10	T_Actividad	4	0	pequeño
11	L_N Gestionar afiliado más integral.	1	4	pequeño
12	DAO _ Gestionar afiliado más integral.	0	2	pequeño
13	T_Afiliado más integral	3	0	pequeño
14	L_N Gestionar C/S más integral.	1	4	pequeño
15	DAO _ Gestionar C/S más integral.	0	2	pequeño
16	T_C/S más integral	3	0	pequeño
17	L_N Gestionar efemérides	1	3	pequeño
18	DAO _Gestionar efemérides	0	2	pequeño
19	T_ efeméride	3	0	pequeño
20	L_N Gestionar evaluación	1	3	pequeño
21	DAO _Gestionar evaluación	0	2	pequeño
22	T_ evaluación	3	0	pequeño
23	T_ afiliado	8	0	pequeño
24	Gestionar noticia	1	4	pequeño
25	DAO _Gestionar noticia	0	2	pequeño
26	T_ noticia	5	0	pequeño
27	L_N Gestionar parámetros	1	4	pequeño
28	DAO _Gestionar parámetros	0	2	pequeño
29	T _parámetros de emulación	3	0	pequeño

Capítulo IV: Validación de los Resultados

30	L_N Gestionar reglamentos	1	3	pequeño
31	DAO _Gestionar reglamentos	0	2	pequeño
32	T_reglamentos	3	0	pequeño
33	L_N Gestionar usuario	1	4	pequeño
34	DAO _Gestionar usuario	0	2	pequeño
35	T_afiliado	8	0	pequeño
36	L_N Registrar participación en actividades	1	1	pequeño
37	DAO _ Registrar actividades	0	2	pequeño
38	T_Actividad	4	0	pequeño
39	T_afiliado	8	0	pequeño
40	L_N Registrar participación en actividades	1	1	pequeño
41	DAO _ Registrar evento	0	2	pequeño
42	T_evento	3	0	pequeño
43	T_afiliado	8	0	pequeño
44	L_N Registrar resultados	1	1	pequeño
45	DAO _ Registrar resultados	0	2	pequeño
46	T_afiliado	8	0	pequeño
47	T_resultado	3	0	pequeño

Quando existe un TC grande se afectan los parámetros de calidad definidos por esta métrica. Se reduce la reutilización de las clases, la implementación se hace más compleja, las pruebas son difíciles de realizar y aumenta la responsabilidad de las clases. (Edith, 2008)

Todas de las clases que conforman el diseño que será la entrada para el flujo de trabajo de implementación están dentro de la categoría de pequeñas, lo que demuestra que la construcción del Proceso de Administración y Control de la Información de la Sección Sindical VP no será compleja. Los resultados obtenidos son positivos según esta métrica, como se puede ver en las siguientes tablas.

Tabla 4-3: Cantidad de clases por clasificación.

Clasificación	Cantidad de Clases
Pequeño	47
Medio	0
Grande	0

Tabla 4-4: Resultados de la Métrica TC.

Cantidad de Clases	Promedio de Atributos	Promedio de Operaciones
47	2.13	1.5

4.3.2. Relaciones entre clases (RC)

Tabla 4-5: Cantidad de relaciones de uso entre las clases

No.	Nombre	Relaciones de uso
1	L_N Autenticar usuario	1
2	DAO _Autenticar	1
3	T _Usuario	0
4	T _Rol	1
5	L_N Gestionar actividades extra laborales.	1
6	DAO _ Gestionar actividades extra laborales.	1
7	T _Actividad	0
8	L_N Gestionar actividades sindicales.	1
9	DAO _ Gestionar actividades sindicales.	1
10	T _Actividad	0
11	L_N Gestionar afiliado más integral.	1
12	DAO _ Gestionar afiliado más integral.	1
13	T_ Afiliado más integral	0

Capítulo IV: Validación de los Resultados

14	L_N Gestionar C/S más integral.	1
15	DAO _ Gestionar C/S más integral.	1
16	T_ C/S más integral	0
17	L_N Gestionar efemérides	1
18	DAO _ Gestionar efemérides	1
19	T_ efeméride	0
20	L_N Gestionar evaluación	1
21	DAO _ Gestionar evaluación	1
22	T_ evaluación	1
23	T_ afiliado	0
24	Gestionar noticia	1
25	DAO _ Gestionar noticia	1
26	T_ noticia	0
27	L_N Gestionar parámetros	1
28	DAO _ Gestionar parámetros	1
29	T _parámetros de emulación	0
30	L_N Gestionar reglamentos	1
31	DAO _ Gestionar reglamentos	1
32	T _reglamentos	0
33	L_N Gestionar usuario	1
34	DAO _ Gestionar usuario	1
35	T _afiliado	0
36	L_N Registrar participación en actividades	1
37	DAO _ Registrar actividades	1
38	T_Actividad	1
39	T_ afiliado	0
40	L_N Registrar participación en actividades	1
41	DAO _ Registrar evento	1
42	T_evento	1
43	T_ afiliado	0
44	L_N Registrar resultados	1
45	DAO _ Registrar resultados	1

46	T_ afiliado	1
47	T_ resultado	0

Para medir el acoplamiento según los resultados de esta métrica, se plantean los siguientes valores.

Tabla 4-6: Acoplamiento

Categoría	Relaciones de uso	Cantidad de clases
Ninguno	0	14
Bajo	1	33
Medio	2	0
Alto	>2	0

Los otros parámetros de calidad que mide esta métrica dependen del valor promedio de las relaciones de uso de todas las clases, en el caso del Análisis y Diseño del Proceso de Administración y Control de la Información para la sección sindical VP dicho promedio es de 0.7.

Tabla 4-7: Cantidad de pruebas y complejidad de mantenimiento.

Categoría	Criterio	Cantidad de clases
Baja	\leq Prom.	14
Media	$>$ Prom. Y ≤ 2 *Prom.	33
Alta	> 2 *Prom.	0

Tabla 4-8: Reutilización

Categoría	Criterio	Cantidad de clases
Baja	> 2 *Prom.	0
Media	$>$ Prom. Y ≤ 2 *Prom.	33
Alta	\leq Prom.	14

Resumiendo el uso de esta métrica, los resultados pueden calificarse como positivos. El acoplamiento existente entre las clases es bajo, es decir, el necesario para contribuir a una alta cohesión, el 29.8% no tiene ningún acoplamiento, y el 70.2% es de bajo acoplamiento. El nivel de reutilización de las clases es del 29.8% de las clases pueden ser altamente

reutilizables, teniendo el 70.2% de las clases en el nivel medio de reutilización. Sucede de la misma manera con la cantidad de pruebas y la complejidad de mantenimiento, el 29.8% de las clases son fáciles de reparar, teniendo mediana complejidad el 70.2% de las clases y la cantidad de pruebas a realizar es relativamente pequeña.

Conclusiones Parciales del Capítulo 4

Luego de validar los resultados obtenidos durante la investigación con la utilización de métricas para evaluar la especificación de los requisitos, el tamaño de clases y las relaciones entre ellas y determinar el número de requisitos que no son considerados en ningún caso de uso se concluye que:

El nivel de ambigüedad de los requisitos es bajo, logrando así el nivel de especificidad requerido y el entendimiento casi total por parte de los revisores.

El 100% de las clases que conforman el diseño están dentro de la categoría de pequeñas, lo que demuestra que el proceso de construcción y pruebas del Proceso de Administración y Control de la Información para la Sección Sindical VP no será complejo.

El 70.2% de las clases presenta clasificaciones de bajo acoplamiento, contribuyendo a una alta cohesión. El 29.8% de las clases es altamente reutilizable, con el 70.2% de mediana reutilización.

Las métricas aplicadas a los artefactos obtenidos durante la investigación validan de forma general la calidad del análisis y diseño del Proceso de Administración y Control de la Información para la Sección Sindical VP.

Conclusiones

En el presente trabajo de diploma se demostró la necesidad de realizar el análisis y diseño de una aplicación de escritorio para mejorar los procesos de administración y control de la información de la sección sindical Vicerrectoría Primera perteneciente a la Universidad de las Ciencias Informáticas.

A modo de conclusión se puede destacar que:

Para llevar a cabo el desarrollo de este trabajo se realizó el análisis de algunas soluciones existentes en el mundo con características similares al propuesto.

Se realizó un estudio detallado de la gestión de la información generada en la sección sindical Vicerrectoría Primera de la Universidad de las Ciencias Informáticas.

Se logró escoger la metodología correcta para el mayor entendimiento del negocio actual y las herramientas que garantizaron un buen diseño.

El sistema se diseñó siguiendo la metodología planteada en el estado del arte y se utilizaron representaciones para la modelación de la fase del proyecto análisis y diseño.

Se propuso el prototipo no funcional del sistema cumpliendo con las expectativas del cliente.

Se desarrolló el flujo de análisis y diseño que les permita a los programadores tener una idea clara y precisa sobre el contenido a implementar.

Con el sistema que se propone se le darán soluciones a muchos de los problemas que existen en la Sección Sindical Vicerrectoría Primera, y de esta forma se facilitará el trabajo de los secretarios a nivel de comité sindical y de sección sindical. De esta forma se cumplen con los objetivos trazados para la elaboración del trabajo de diploma.

Recomendaciones

En este trabajo se realizó una primera versión que brindó solución a los problemas referentes a los procesos de administración y control de la información de la sección sindical Vicerrectoría Primera ubicada en la Universidad de las Ciencias Informáticas. Por lo que se recomienda:

Integrar el sistema propuesto con las diferentes aplicaciones de la Universidad de las Ciencias Informáticas.

Implementar la solución propuesta.

Bibliografía

[En línea] // Gobierno Bolivariano de Venezuela. Me rinde. Me rinde. - 2008. - http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=473&Itemid=29 1.

[En línea]. - www.angelfire.com.

Alineación entre Metas Organizacionales y Elicitación de Requisitos del Software.

[Libro] / aut. Zapata Carlos y Arango, J.. - 2004.

Análisis y Diseño de Sistemas. [Libro] / aut. Kendall Kendall &. - 2005.

Análisis y Diseño de un Nodo Virtual de Procesos. [Libro] / aut. Edith Maylen y Ortiz, Leydis A.. - Habana, UCI : [s.n.], 2008.

Aprendiendo UML en 24 horas. [Libro] / aut. Schumler J. - Mexico : Pearson Education, 2000.

Business Process Modeling Notation. Wikipedia, la enciclopedia libre. [En línea]. - 2009.

C Sharp. Wikipedia, la enciclopedia libre. [En línea].

Conferencia 4, Ingeniería de software 1 [Libro]. - 2009.

Desarrollo ágil de software. Wikipedia, la enciclopedia libre [En línea]. - 2009.

Diseño de Sistemas. Monografías.com [En línea].

Diseño de una aplicación para el análisis de imágenes en la caracterización genética de microorganismos. [Libro] / aut. Socorro Suárez Sianny y Alvarez Pérez Rigoberto Miguel.. - Ciudad de la Habana, UCI : [s.n.], 2007.

El modelo IDEF0 [En línea]. - http://sunwc.cepade.es/jrivera/org_temas/metodos/model_simulac/modelo_idef0.htm.

El mundo informático [Libro] / aut. Larman C. - 2008.

El proceso Unificado de Desarrollo de Software [Libro] / aut. Jacobson I. & Booch, G. & Rumbaugh, J.. - La Habana. : [s.n.], 2004. - Vol. I y II.

El proceso unificado de desarrollo de software. [Libro] / aut. Jacobson I.,Booch,G. % Rumbaugh,J.. - Madrid : Addison Wesley Object Technology Series., 2000.

Fase de inicio. Modelo de negocio [Conferencia]// Conferencia 2. Ingeniería de Software I. - Ciudad de la Habana : [s.n.], 2007-2008.

Ingeniería de requisitos en Aplicaciones para la web - Un estudio comparativo. [Libro] / aut. Koch NOra y María, José.. - España: Universidad de Sevilla Lenguajes y Sistemas Informáticos : [s.n.], 2002.

Ingeniería de ría del Software. Un enfoque práctico. [Libro] / aut. Cota A.. - [s.l.] : Felix Varela, 2005.

Ingeniería de software [Libro] / aut. Sommerville Ian. - España : Pearson Education s.a, 2005.

Ingeniería de Software [Libro] / aut. Sommerville Ian. - 2007.

Ingeniería de Software, Un enfoque práctico. [Libro] / aut. Pressman Roger S.. - Madrid : [s.n.], 2001. - Vol. I.

Ingeniería de Software. Un enfoque práctico. [Libro] / aut. Pressman R.S.. - 2005.

Ingeniería de Software. Un enfoque Práctico. [Libro] / aut. Pressman R.S. - 1998.

Lenguaje de Modelado de objetos. Wikipedia, la enciclopedia libre. [En línea]. - 2009. - [http://es.wikipedia.org/wiki/Lenguaje de modelado de objetos..](http://es.wikipedia.org/wiki/Lenguaje_de_modelado_de_objetos..)

Lenguaje de Programación. Wikipedia, la enciclopedia libre. [En línea]. - 2009.

Marquetti Asociados [Libro] / aut. Marquetti H.,& Otros. - 1997.

Mastering the Requirements Process.2. [Libro] / aut. Robertson Suzanne y Robertson, James. - Portland, Oregon : Addison-Wesley Professional, 2006.

Metodología FDD [Libro] / aut. Calabria Luis. - 2003.

MySQL. Wikipedia, la enciclopedia libre [En línea]. - 2009.

PostgreSQL. Wikipedia, la enciclopedia libre [En línea]. - 2009.

Rational Rose: Procesos básicos para desarrollar un proyecto con UML. [En línea]. - <http://www.vico.org/TallerRationalRose.pdf>.

Rational Unified Process 1.0 [Libro] / aut. Rational Software Corporation. - 2003.

Sistema de gestión de base de datos. Wikipedia, la enciclopedia libre. [En línea]. - 2009.

Sistema de Gestión de Contenidos. Wikipedia, la enciclopedia libre. [En línea]. - 2009. - <http://es.wikipedia.org/wiki/CMS>.

Sistemas de Información [Libro] / aut. Peralta Manuel. - 2003.

UML y Patrones [Libro] / aut. Larman C.. - 2003.

Use Cases Patterns and Blueprints [Libro] / aut. Overgaard Gunnar, Palmkvist Karin. - 2004.

Anexos

Anexo # 1 Descripción textual de los CU del sistema

Tabla A 1.1 Descripción CU Registrar participación en actividades.

Caso de uso	Registrar participación en actividades.
Actor	Secretario del comité sindical
Propósito	Registrar las actividades en que han participado los afiliados del comité sindical
Resumen	El Secretario del comité sindical solicita insertar la actividad. El sistema brinda la posibilidad de insertar la actividad en que han participado los afiliados de su comité sindical. El CU termina cuando el actor guarda la actividad o selecciona la opción Cancelar.
Referencias	RF43
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema. El actor ha buscado y seleccionado el documento a imprimir.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
.	1. Muestra una interfaz con el modelo de Control de Participación en Actividades.
2- Selecciona el afiliado que desea insertarle una actividad en la que ha participado.	3-Permite insertar la actividad.
4- Inserta los datos referentes a la actividad y solicita guardarlos.	5- El sistema valida los datos del modelo. 6- El sistema registra los datos del modelo 7- El sistema notifica al usuario que los datos se guardaron correctamente. 8- Termina el caso de uso.
Curso Alterno	
Línea 5	

<p>5.1 El sistema detecta errores en los datos insertados y los notifica al Secretario del comité sindical solicitando que sean rectificadas.</p> <p>5.2 El Secretario arregla los errores y solicita guardar datos rectificadas. Regresar a la línea 5.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos en el Modelo de Control de Participación en Actividades. Termina el caso de uso.</p>	
Pos condiciones	La actividad ha sido guardada correctamente en el modelo de Control de Participación en Actividades
Prioridad	Secundario.

Tabla A 1.2 Descripción CU Gestionar Parámetros

Caso de uso	Gestionar parámetros
Actor	Secretario de Emulación
Propósito	Gestionar los parámetros de emulación.
Resumen	El caso de uso inicia cuando el Secretario de emulación desea mostrar, adicionar, modificar o eliminar los parámetros de emulación.
Referencias	RF4,RF5, RF6, RF7
Precondiciones	<p>El actor se ha identificado y autenticado correctamente ante el sistema.</p> <p>El usuario debe tener los permisos necesarios para hacer la operación.</p>
Curso Normal de Eventos	
Sección 1. Mostrar parámetros.	
Acción del actor	Respuesta del sistema

1- El sistema muestra un formulario con un listado de parámetros a seleccionar.	
2-El secretario de emulación selecciona parámetros.	3-El sistema muestra los parámetros seleccionados.
Curso Alterno	
Línea 2	
2.1- Si no hay un parámetro seleccionado el sistema mostrara mensaje “seleccione parámetros que desea ver”	
Regresa a la línea 2 de la sección 1.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.	
Sección 2. Adicionar parámetros.	
Acción del actor	Respuesta del sistema
	4- El sistema muestra listado de parámetros.
5-El secretario de emulación presiona el botón de adicionar nuevo parámetro.	6-El sistema muestra formulario con los campos a llenar.
7-El secretario llena los campos de formulario y presiona el botón aceptar.	8-El sistema valida y registra los nuevos datos.
Curso Alterno	
Línea 8.	
8.1-El sistema detecta errores en los datos adicionados y lo notifica al secretario de emulación solicitando que sean rectificadas.	

8.2-El secretario de emulación arregla los errores y solicita guardar los datos rectificadas.

Regresar a la línea 8 de la sección 2.

* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos. Termina el caso de uso.

Sección 3. Eliminar parámetros.

Acción del actor

Respuesta del sistema

9- El sistema muestra formulario con el listado de parámetros a seleccionar.

10-El secretario de emulación selecciona el parámetro y presiona el botón eliminar. 11-El sistema elimina el parámetro seleccionado.

Curso Alterno

Línea 10

10.1-Si no hay un parámetro seleccionado el sistema muestra el mensaje:”Debe seleccionar un parámetro”.

Regresa a la línea 10 de la sección 3.

* El usuario selecciona Cancelar en cualquier momento y el sistema no elimina los datos. Termina el caso de uso.

Sección 4. Modificar parámetros.

Acción del actor

Respuesta del sistema

12- El sistema permite seleccionar parámetros a modificar.

13-El secretario de emulación selecciona parámetros y presiona modificar. 14-El sistema muestra formulario con parámetros a modificar.

botón modificar.	15-El sistema registra y valida los nuevos datos entrados.
Curso alterno	
<p>Línea 15</p> <p>15.1-Si los datos no son válidos el sistema muestra un mensaje “Datos no válidos”.</p> <p>Regresa a la línea 15 de la sección 4.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.</p>	
Pos condiciones	Los parámetros de emulación han sido gestionados (adicionados, modificados, eliminados o mostrados) correctamente.
Prioridad	Crítico.

Tabla A1.3 Descripción del CU Gestionar reglamentos.

Caso de uso	Gestionar reglamentos
Actor	Secretario de la sección sindical
Propósito	Gestionar los reglamentos sindicales.
Resumen	El caso de uso inicia cuando el Secretario de la sección sindical desea mostrar, adicionar, modificar los reglamentos.
Referencias	RF1, RF2, RF3
Precondiciones	<p>El actor se ha identificado y autenticado correctamente ante el sistema.</p> <p>El usuario debe tener los permisos necesarios para</p>

hacer la operación.	
Curso Normal de Eventos	
Sección 1. Mostrar reglamentos.	
Acción del actor	Respuesta del sistema
	1- El sistema muestra un formulario con reglamento a seleccionar.
2-El secretario de la sección sindical selecciona reglamento.	3-El sistema muestra el reglamento seleccionado.
Curso Alterno	
Línea 2	
2.1- Si no hay un reglamento seleccionado el sistema mostrara mensaje “seleccione reglamento que desea ver”	
Regresa a la línea 2 de la sección 1.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.	
Sección 2. Adicionar reglamentos	
Acción del actor	Respuesta del sistema
	4- El sistema muestra reglamento.
5-El secretario de la sección sindical presiona el botón de adicionar nuevo reglamento.	6-El sistema muestra formulario con los campos a llenar.
7-El secretario de la sección sindical llena los campos de formulario y presiona el botón	8-El sistema valida y registra los nuevos datos.

aceptar.

Curso Alterno

Línea 8

8.1-El sistema detecta errores en los datos adicionados y lo notifica al secretario de la sección sindical solicitando que sean rectificadas.

8.2-El secretario de la s/s arregla los errores y solicita guardar los datos rectificadas.

Regresar a la línea 8 de la sección 2.

* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos. Termina el caso de uso.

Sección 3. Modificar reglamentos

Acción del actor

Respuesta del sistema

9-El sistema permite seleccionar reglamento a modificar

10-El secretario de la s/s selecciona reglamento y presiona botón modificar.

11-El sistema muestra formulario con reglamento a modificar

12-El sistema registra y valida los nuevos datos entrados.

Curso Alterno

Línea 12

12.1-Si los datos no son válidos el sistema muestra un mensaje "Datos no válidos".

Regresa a la línea 12 de la sección 3.

* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.

Pos condiciones	El reglamento ha sido gestionado (adicionado, modificado o mostrado) correctamente.
Prioridad	Crítico.

Tabla A1.4 Descripción del CU Gestionar actividades sindicales.

Caso de uso	Gestionar actividades sindicales
Actor	Secretario de la sección sindical
Propósito	Gestionar las actividades sindicales.
Resumen	El caso de uso inicia cuando el Secretario de la sección sindical desea mostrar, adicionar, modificar o eliminar las actividades sindicales.
Referencias	RF44, RF45, RF46, RF47
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema. El usuario debe tener los permisos necesarios para hacer la operación.
Curso Normal de Eventos	
Sección 1. Mostrar actividades sindicales	
Acción del actor	Respuesta del sistema
	1- El sistema muestra un formulario con actividades sindicales a seleccionar.
2-El secretario de la sección sindical selecciona actividades sindicales.	3-El sistema muestra las actividades sindicales seleccionadas.
Curso Alterno	
Línea 2	
2.1- Si no hay actividades sindicales seleccionadas el sistema mostrara mensaje “seleccione actividades sindicales que desea ver”	
Regresa a la línea 2 de la sección 1.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.	

Sección 2. Adicionar actividades sindicales	
Acción del actor	Respuesta del sistema
	4- El sistema muestra actividades sindicales.
5-El secretario de la sección sindical presiona el botón de adicionar nueva actividad sindical.	6-El sistema muestra formulario con los campos a llenar.
7-El secretario de la sección sindical llena los campos de formulario y presiona el botón aceptar.	8-El sistema valida y registra los nuevos datos.
Curso Alterno	
Línea 8.	
8.1-El sistema detecta errores en los datos adicionados y lo notifica al secretario de la sección sindical solicitando que sean rectificadas.	
8.2-El secretario de la s/s arregla los errores y solicita guardar los datos rectificadas.	
Regresar a la línea 8 de la sección 2.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos. Termina el caso de uso.	
Sección 3. Eliminar actividades sindicales	
Acción del actor	Respuesta del sistema
	9- El sistema muestra formulario con las actividades sindicales a seleccionar.
10-El secretario de la sección sindical selecciona la actividad sindical y presiona el botón eliminar.	11-El sistema elimina la actividad seleccionada.
Curso Alterno	

<p>Línea 10</p> <p>10.1-Si no hay actividad seleccionada el sistema muestra el mensaje:”Debe seleccionar actividad”.</p> <p>Regresa a la línea 10 de la sección 3.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no elimina los datos. Termina el caso de uso.</p>	
<p>Sección 4. Modificar actividades sindicales</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
	<p>12- El sistema permite seleccionar actividad a modificar.</p>
<p>13- El secretario de la sección sindical selecciona actividad y presiona botón modificar.</p>	<p>14- El sistema muestra formulario con actividad a modificar. El sistema registra y valida los nuevos datos entrados.</p>
<p>Curso alterno</p>	
<p>Línea 14</p> <p>14.1-Si los datos no son válidos el sistema muestra un mensaje “Datos no válidos”.</p> <p>Regresa a la línea 14 de la sección 4.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos .Termina el caso de uso.</p>	
<p>Pos condiciones</p>	<p>Las actividades sindicales han sido gestionadas correctamente.</p>
<p>Prioridad</p>	<p>Critico</p>

Tabla A1.5 Descripción del CU Gestionar efemérides.

Caso de uso	Gestionar efemérides
<p>Actor</p>	<p>Secretario de la sección sindical</p>
<p>Propósito</p>	<p>Gestionar las efemérides.</p>
<p>Resumen</p>	<p>El caso de uso inicia cuando el Secretario de la sección sindical desea mostrar, adicionar, modificar o</p>

	eliminar las efemérides
Referencias	RF27, RF28, Rf29, RF30.
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema. El usuario debe tener los permisos necesarios para hacer la operación.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
Sección 1. Mostrar efemérides.	
Acción del actor	Respuesta del sistema
	1- El sistema muestra un formulario con efemérides a seleccionar.
2-El secretario de la sección sindical selecciona efemérides.	3-El sistema muestra las efemérides seleccionadas.
Curso Alterno	
Línea 2	
2.1- Si no hay efemérides seleccionadas el sistema mostrara mensaje “seleccione efemérides que desea ver”	
Regresa a la línea 2 de la sección 1.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.	
Sección 2. Adicionar efemérides.	
Acción del actor	Respuesta del sistema
	4- El sistema muestra efemérides.
5-El secretario de la sección sindical presiona el botón de adicionar nueva efeméride.	6-El sistema muestra formulario con los campos a llenar.
7-El secretario de la sección sindical llena los campos de formulario y presiona el botón	8-El sistema valida y registra los nuevos datos.

aceptar.	
Curso Alterno	
<p>Línea 8.</p> <p>8.1-El sistema detecta errores en los datos adicionados y lo notifica al secretario de la sección sindical solicitando que sean rectificadas.</p> <p>8.2-El secretario de la s/s arregla los errores y solicita guardar los datos rectificadas.</p> <p>Regresar a la línea 8 de la sección 2.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos. Termina el caso de uso.</p>	
Sección 3. Eliminar efemérides.	
Acción del actor	Respuesta del sistema
	9- El sistema muestra formulario con las efemérides a seleccionar.
10-El secretario de la sección sindical selecciona la efeméride y presiona el botón eliminar.	11-El sistema elimina la efeméride.
Curso Alterno	
<p>Línea 10</p> <p>10.1-Si no hay efeméride seleccionada el sistema muestra el mensaje:"Debe seleccionar efeméride".</p> <p>Regresa a la línea 10 de la sección 3.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no elimina los datos. Termina el caso de uso.</p>	
Sección 4. Modificar efemérides.	
Acción del actor	Respuesta del sistema
	12- El sistema permite seleccionar efeméride a modificar.
13-El secretario de la s/s selecciona efeméride y	14-El sistema muestra formulario con efeméride a modificar.

presiona botón modificar.	15-El sistema registra y valida los nuevos datos entrados.
Curso alterno	
Línea 15	
15.1-Si los datos no son válidos el sistema muestra un mensaje “Datos no válidos”. Regresa a la línea 15 de la sección 4.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos .Termina el caso de uso.	
Pos condiciones	Las efemérides han sido gestionadas (adicionadas, modificadas, eliminadas o mostradas) correctamente.
Prioridad	Crítico.

Tabla A1.6 Descripción del CU Gestionar actividades extra laborales

Caso de uso	Gestionar actividades extra laborales
Actor	Secretario de deporte, cultura y recreación.
Propósito	Gestionar las actividades extra laborales.
Resumen	El caso de uso inicia cuando el Secretario de deporte, cultura y recreación (D/C/R) desea mostrar, adicionar, modificar o eliminar las actividades extra laborales.
Referencias	RF31, RF32, RF33, RF34
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema. El usuario debe tener los permisos necesarios para hacer la operación.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
Sección 1. Mostrar actividades extra laborales	
Acción del actor	Respuesta del sistema
	1- El sistema muestra un formulario con actividades extra laborales a seleccionar.
2-EI secretario selecciona	3-El sistema muestra las actividades extra laborales

actividades extra laborales.	seleccionadas.
Curso Alterno	
Línea 2	
2.1- Si no hay actividades extra laborales seleccionadas el sistema mostrara mensaje “seleccione actividades extra laborales que desea ver”	
Regresa a la línea 2 de la sección 1.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.	
Sección 2. Adicionar actividades extra laborales.	
Acción del actor	Respuesta del sistema
	4- El sistema muestra actividades extra laborales.
5-El secretario presiona el botón de adicionar nueva actividad.	6-El sistema muestra formulario con los campos a llenar.
7-El secretario llena los campos de formulario y presiona el botón aceptar.	8-El sistema valida y registra los nuevos datos.
Curso Alterno	
Línea 8.	
8.1-El sistema detecta errores en los datos adicionados y lo notifica al secretario solicitando que sean rectificadas.	
8.2-El secretario arregla los errores y solicita guardar los datos rectificadas.	
Regresar a la línea 8 de la sección 2.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos. Termina el caso de uso.	
Sección 3. Eliminar actividades extra laborales.	
Acción del actor	Respuesta del sistema
	9- El sistema muestra formulario con las actividades extra laborales a seleccionar.
10-El secretario selecciona la	11-El sistema elimina la actividad.

actividad y presiona el botón eliminar.	
Curso Alterno	
<p>Línea 10</p> <p>10.1-Si no hay actividad seleccionada el sistema muestra el mensaje:”Debe seleccionar actividad”.</p> <p>Regresa a la línea 10 de la sección 3.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no elimina los datos. Termina el caso de uso.</p>	
Sección 4. Modificar actividades extra laborales.	
Acción del actor	Respuesta del sistema
	12- El sistema permite seleccionar actividades extra laborales a modificar.
13-El secretario selecciona actividad y presiona botón modificar.	14-El sistema muestra formulario con actividad a modificar. 15-El sistema registra y valida los nuevos datos entrados.
Curso alterno	
<p>Línea 15</p> <p>15.1-Si los datos no son válidos el sistema muestra un mensaje “Datos no válidos”.</p> <p>Regresa a la línea 15 de la sección 4.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos .Termina el caso de uso.</p>	
Pos condiciones	Las actividades extra laborales han sido gestionadas (adicionadas, modificadas, eliminadas o mostradas) correctamente.
Prioridad	Crítico.

Tabla A1.7 Descripción del CU Buscar Información.

Caso de uso	Buscar información
--------------------	---------------------------

Actor	Usuario(inicia)
Propósito	Permite al usuario realizar búsquedas de información de un equipo determinado.
Resumen	El caso de uso se inicia cuando un usuario solicita una búsqueda de información, inmediatamente se muestra un formulario con los parámetros de búsqueda, se llenan los campos de los cuales se desea conocer información, ésta se muestra en un listado y finaliza así el caso de uso.
Referencias	RF40
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción de Buscar Información.	2-Muestra inmediatamente los parámetros de búsqueda.
3. Selecciona y llena los campos de los cuales desea obtener la información.	4. Busca la información solicitada en la base de datos y muestra todos los datos según los parámetros de búsqueda especificados, finalizando así el caso de uso.
Pos condiciones	Se ha realizado la búsqueda correctamente.
Prioridad	Crítico.

Tabla A1.8 Descripción del CU Registrar participación en los eventos.

Caso de uso	Registrar participación en los eventos
Actor	Secretario del comité sindical (inicia).
Propósito	Registrar los eventos en que ha participado un afiliado determinado perteneciente a un comité sindical.
Resumen	El Secretario del comité sindical solicita insertar el

	evento. El sistema brinda la posibilidad de insertar el evento en que ha participado el afiliado. El CU termina cuando el actor guarda el evento o selecciona la opción Cancelar.
Referencias	RF41
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
.	1. Muestra una interfaz con el modelo de Control de Participación en Eventos. 2. Brinda la posibilidad de insertar un evento.
3. Inserta los datos referentes al evento en que haya participado el afiliado previamente buscado.	
4-En caso de que el afiliado haya obtenido resultados en el evento se invoca al CU Registrar Resultados.	
5-Solicita guardar los datos sobre el evento.	
	6-El sistema valida los datos del modelo de Control de Participación en Eventos. 7-El sistema registra los datos del modelo de Control de Participación en Eventos. 8- El sistema notifica al usuario que los datos del modelo se guardaron correctamente. 9. Termina el caso de uso.
Curso Alterno	
Línea 4	
En caso de que el afiliado no haya obtenido resultados en el evento no se invoca el	

CU	
Registrar Resultados y el actor solicita guardar.	
Línea 6	
6.1 El sistema detecta errores en los datos insertados y los notifica al Secretario del comité sindical solicitando que sean rectificadas.	
6.2 El Secretario del comité sindical arregla los errores y solicita guardar datos rectificadas. Regresar a la línea 6.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos en el modelo de Control de Participación en Eventos. Termina el caso de uso.	
Pos condiciones	El evento ha sido guardado correctamente en el modelo de Control de Participación en Eventos.
Prioridad	Secundario.

Tabla A1.9 Descripción del CU Registrar resultados.

Caso de uso	Registrar resultados
Actor	Secretario del comité sindical (inicia).
Propósito	Registrar los resultados que haya tenido un afiliado tras haber participado en un evento y haber obtenido algún reconocimiento.
Resumen	El Secretario del comité sindical solicita insertar el resultado que haya obtenido un afiliado tras participar en un evento y obtener algún reconocimiento. El sistema brinda la posibilidad de insertar el resultado del afiliado. El CU termina cuando el actor guarda el resultado del afiliado en el evento o selecciona la opción Cancelar.
Referencias	RF42
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema. .
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
.	1. Muestra una interfaz con el modelo de Control de

	<p>Participación en Eventos.</p> <p>2. Brinda la posibilidad de modificar un evento para insertarle los resultados.</p>
<p>3. Selecciona el afiliado que haya obtenido resultados en un evento determinado por su destacada participación.</p>	<p>4. Muestra de manera editable el evento al que es necesario agregarle los resultados.</p>
<p>5-Inserta los resultados y solicita guardarlos.</p>	<p>6-El sistema valida los datos del modelo de Control de Participación en Eventos.</p> <p>7-El sistema registra los resultados del modelo de Control de Participación en Eventos.</p> <p>8- El sistema notifica al usuario que los datos del modelo se guardaron correctamente.</p> <p>9. Termina el caso de uso.</p>
<p>Curso Alterno</p>	
<p>Línea 6</p> <p>6.1 El sistema detecta errores en los datos insertados y los notifica al Secretario del comité sindical solicitando que sean rectificadas.</p> <p>6.2 El Secretario del comité sindical arregla los errores y solicita guardar datos rectificadas. Regresar a la línea 6.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los resultados en el modelo de Control de Participación en Eventos. Termina el caso de uso.</p>	
<p>Pos condiciones</p>	<p>El resultado del evento obtenido por el militante ha sido guardado correctamente en el modelo de Control de Participación en Eventos.</p>
<p>Prioridad</p>	<p>Secundario.</p>

Tabla A1.10 Descripción del CU Gestionar evaluación de afiliado.

Caso de uso	Gestionar evaluación de afiliado
-------------	----------------------------------

Actor	Secretario del comité sindical (inicia).
Propósito	Llenar el modelo de evaluación de un afiliado en el comité sindical o modificarlo en caso de ser necesario.
Resumen	El secretario del comité sindical solicita adicionar, eliminar, mostrar o modificar datos en el modelo de evaluación de un afiliado. El sistema brinda la posibilidad de mostrar los datos en un modelo de evaluación, insertar nuevos datos, modificar los datos que ya estén insertados o eliminarlos en caso de ser necesario. El CU termina cuando el actor guarda el modelo luego de modificarlo, lo elimina o selecciona la opción Cancelar.
Referencias	RF8, RF9, RF10, RF11
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema.
Curso Normal de Eventos	
Sección 1. Mostrar evaluación de afiliado	
Acción del actor	Respuesta del sistema
	1- El sistema muestra un formulario con un listado de afiliados a seleccionar.
2-El secretario del comité sindical selecciona un afiliado.	3-El sistema muestra en otro formulario la evaluación del afiliado seleccionado.
Curso Alterno	
Línea 2	
2.1- Si no hay un afiliado seleccionado el sistema mostrara mensaje “seleccione afiliado para ver su evaluación”	
Regresa a la línea 2 de la sección 1.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no muestra los datos. Termina el caso de uso.	
Sección 2. Modificar evaluación de afiliado	
Acción del actor	Respuesta del sistema

	4-El sistema muestra formulario a modificar con evaluación de afiliado seleccionado previamente por el actor.
5-El secretario del comité sindical introduce nuevos datos y presiona el botón modificar.	6-El sistema valida datos. 7-El sistema guarda nuevos datos insertados.
<p>Línea 6</p> <p>6.1-Si los datos no son válidos el sistema muestra un mensaje “Datos no válidos”. Regresa a la línea 6 de la sección 2.</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no modifica los datos. Termina el caso de uso.</p>	
Sección 3. Eliminar evaluación de afiliado	
Acción del actor	Respuesta del sistema
	8-El sistema muestra formulario con afiliados y su evaluación en el comité sindical.
9-El secretario de la sección sindical selecciona un afiliado y presiona el botón eliminar.	10-El sistema elimina afiliado seleccionado conjuntamente con su evaluación en el comité sindical. 11-El sistema registra los cambios efectuados en el formulario.
Curso Alterno	
<p>Línea 9</p> <p>9.1-Si el secretario de la sección sindical no selecciona un afiliado el sistema muestra el mensaje “Debe seleccionar afiliado y evaluación a eliminar”</p> <p>* El usuario selecciona Cancelar en cualquier momento y el sistema no elimina los datos. Termina el caso de uso.</p>	
Sección 4. Adicionar evaluación de afiliado	
Acción del actor	Respuesta del sistema
	12-El sistema muestra listado de afiliados.
13-El secretario de la sección sindical	14-El sistema muestra formulario con

selecciona afiliado para introducirle los datos de su evaluación.	campos a llenar.
15-El secretario de la sección sindical llena los campos y presiona el botón guardar.	16-El sistema valida y guarda los nuevos datos.
Curso Alterno	
Línea 16	
16.1-El sistema detecta errores en los datos adicionados y lo notifica al secretario solicitando que sean rectificadas.	
16.2-El secretario arregla los errores y solicita guardar los datos rectificadas.	
Regresar a la línea 16 de la sección 4.	
* El usuario selecciona Cancelar en cualquier momento y el sistema no inserta los datos. Termina el caso de uso.	
Pos condiciones	Se ha gestionado la evaluación del afiliado de manera correcta.
Prioridad	Critico

Tabla A1.11 Descripción del CU Gestionar reporte.

Caso de uso	Gestionar reporte
Actor	-
Propósito	Exportar los documentos a formatos .pdf, .xls, .doc.
Resumen	Este caso de uso es iniciado por aquellos casos de uso que llevan la realización de algún reporte, consistiendo en los procedimientos de imprimir o exportar el reporte.
Referencias	RF20, RF21, RF22
Precondiciones	El actor se ha identificado y autenticado correctamente ante el sistema. El actor ha buscado y seleccionado un modelo de Control de Participación en Actividades para un afiliado determinado.
Curso Normal de Eventos	

Sección 1. Imprimir reporte	
Acción del actor	Respuesta del sistema
1-El actor ordena imprimir el reporte.	2-El sistema imprime el reporte.
Sección 2. Exportar reporte	
Acción del actor	Respuesta del sistema
3-El actor ordena exportar el reporte	4- El sistema exporta el reporte
Pos condiciones	Quedó mostrado, impreso y exportado el reporte de manera correcta
Prioridad	Secundario

Anexo # 2 Diagramas de clases del análisis de los CU del sistema.

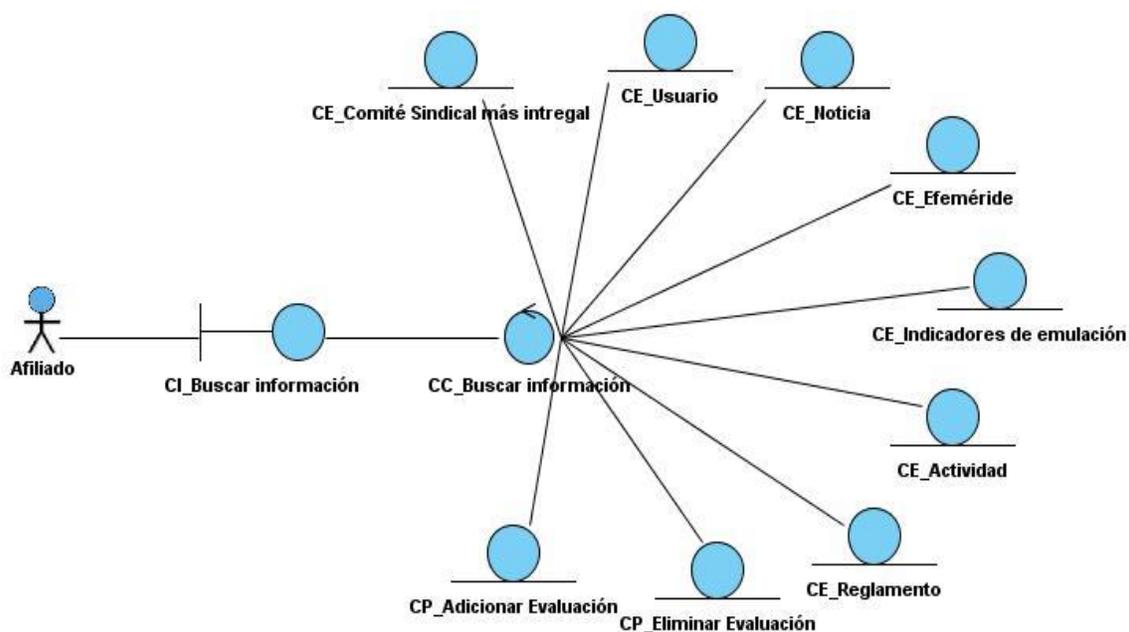


Fig. A 2.1 DCA Buscar información

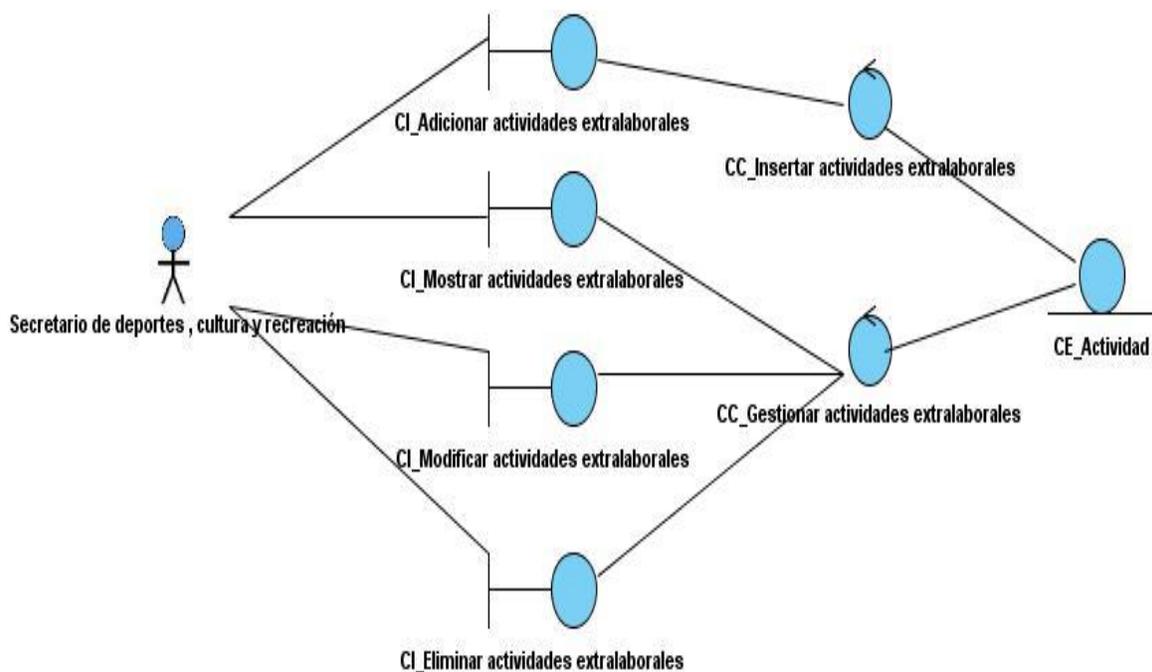


Fig. A 2.2 DCA Gestionar actividades extra laborales.

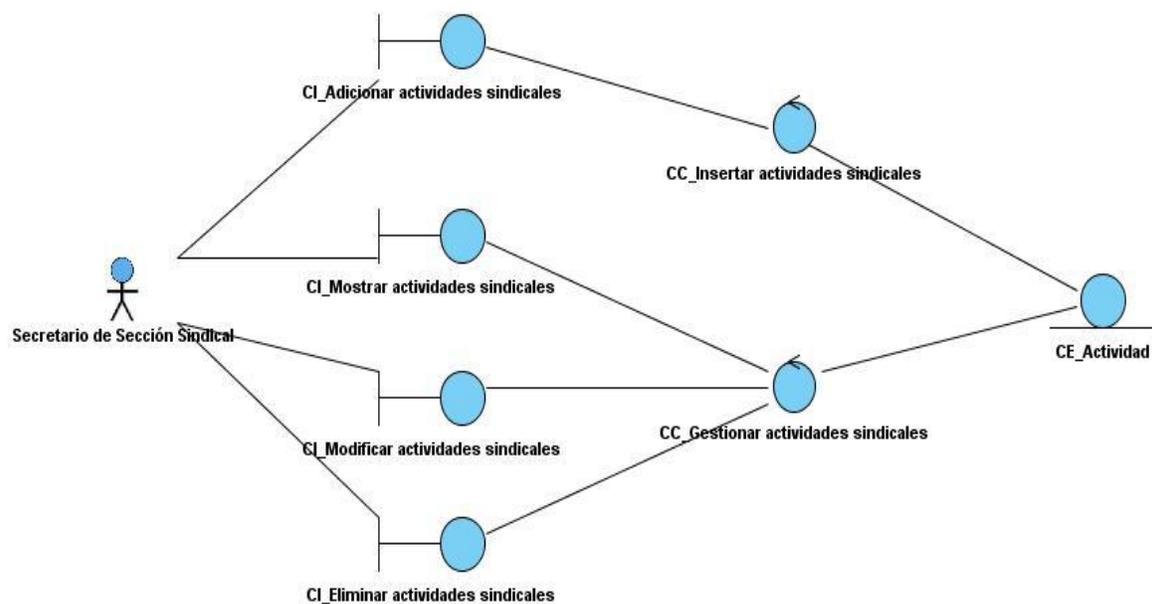


Fig. A 2.3 DCA Gestionar actividades sindicales

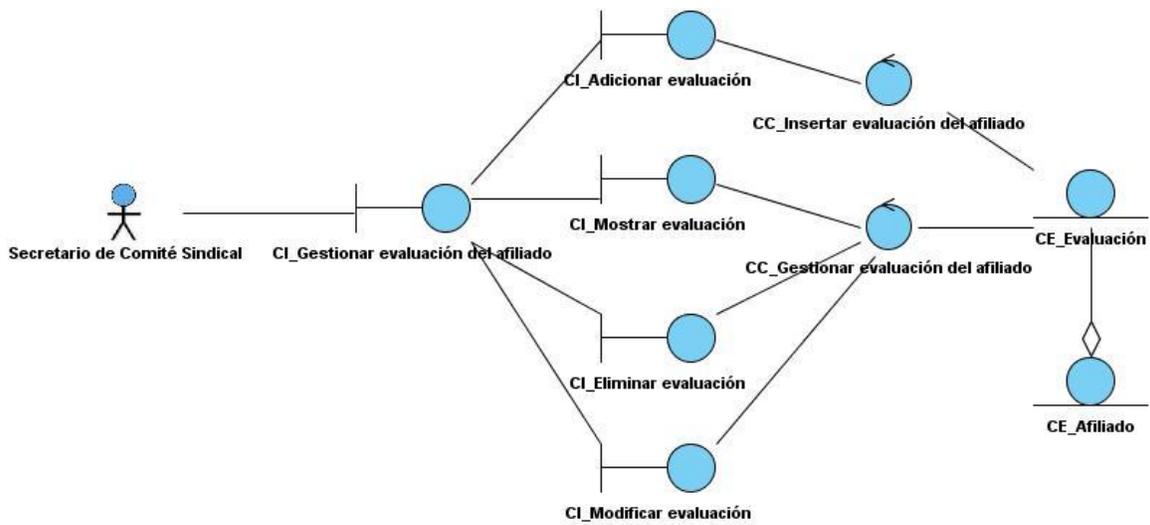


Fig. A 2.4 DCA Gestionar evaluación de afiliado.

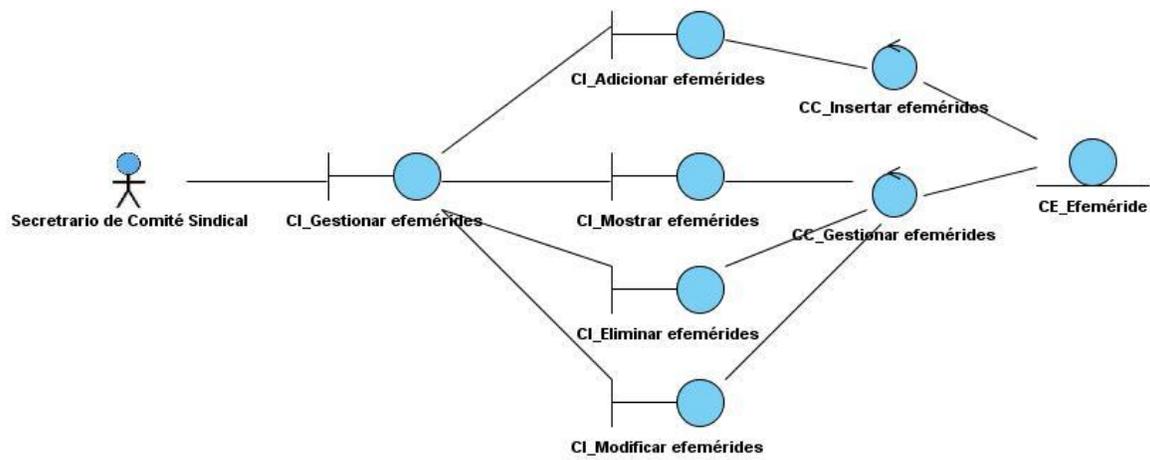


Fig. A 2.5 DCA Gestionar efemérides

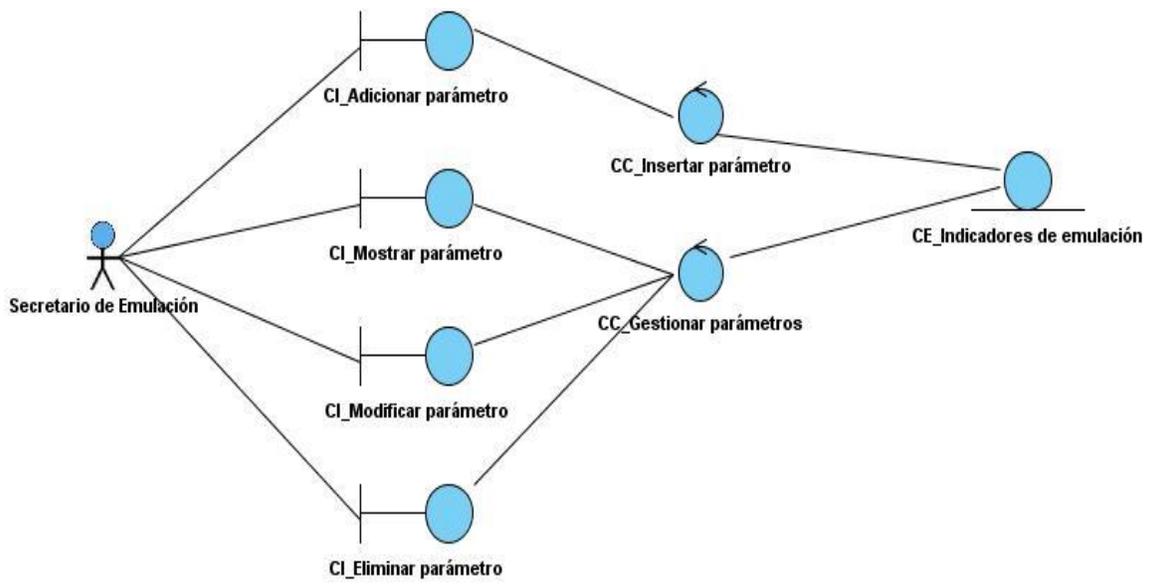


Fig. A 2.6 DCA Gestionar parámetros

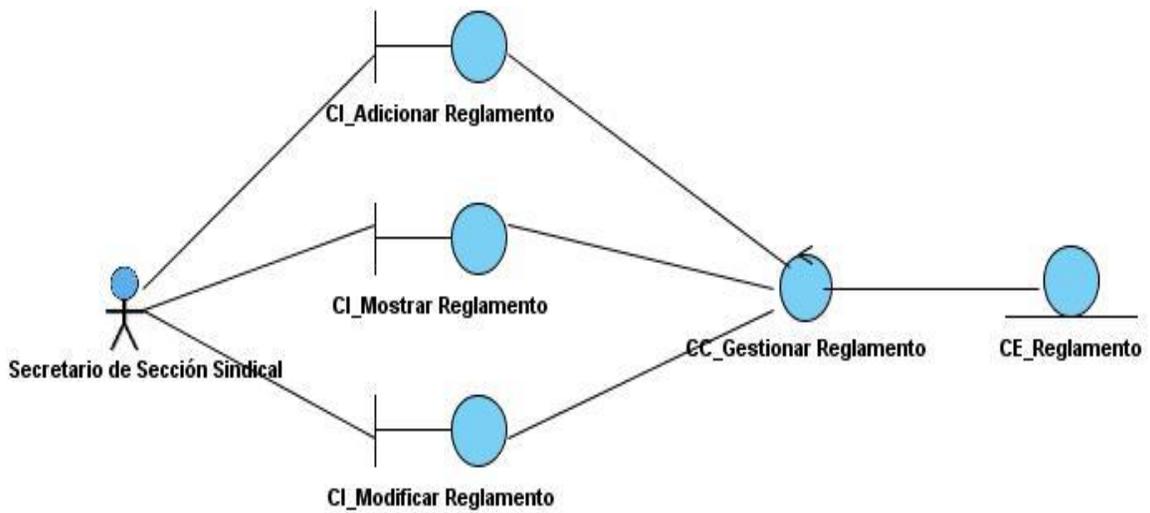


Fig. A 2.7 DCA Gestionar reglamentos

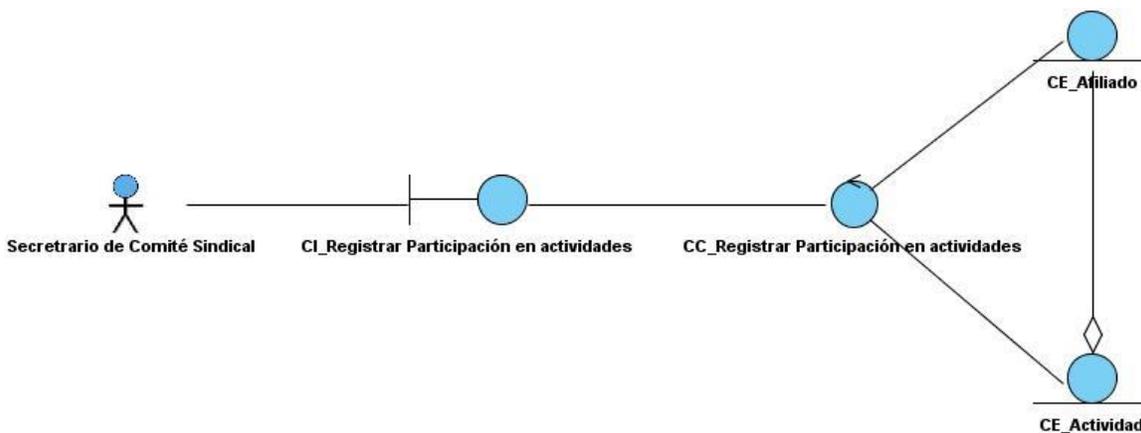


Fig. A 2.8 DCA Registrar participación en actividades

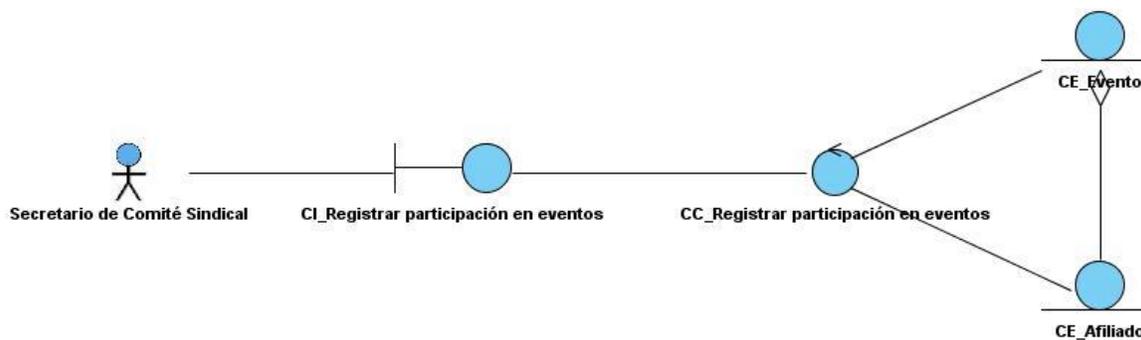


Fig. A 2.9 DCA Registrar participación en evento.

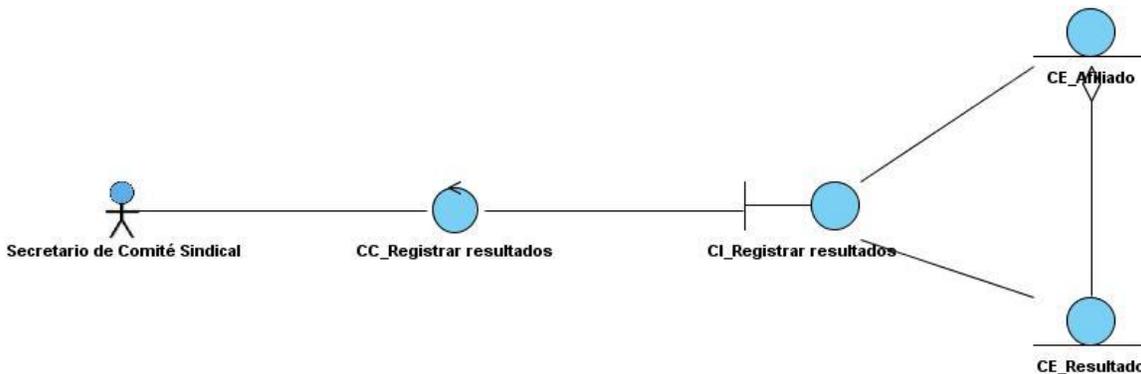


Fig. A 2.10 DCA Registrar resultados.

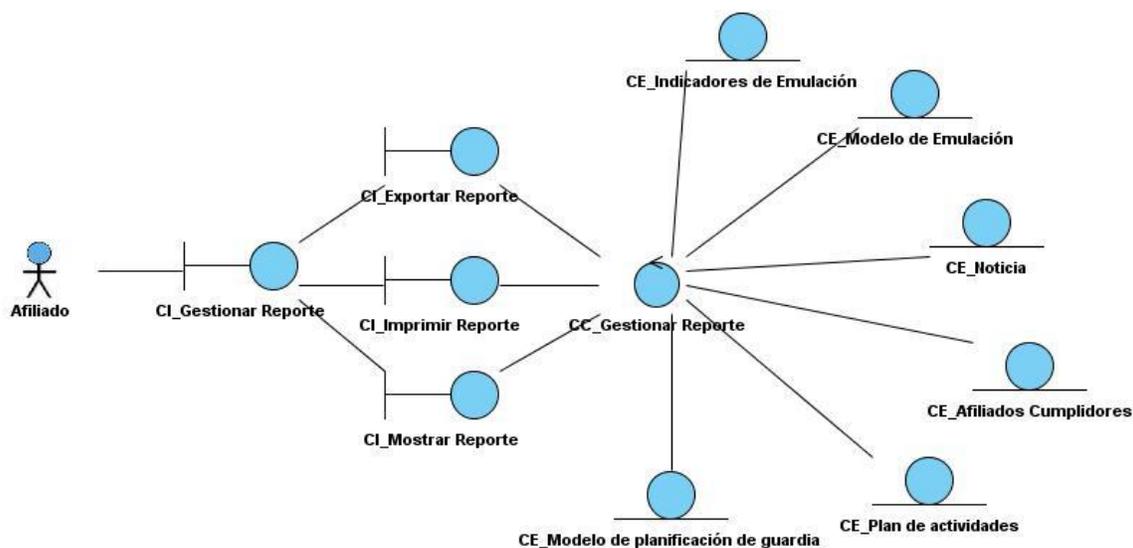


Fig. A 2.11 DCA Gestionar reporte.

Anexo # 3 Diagramas de Colaboración.

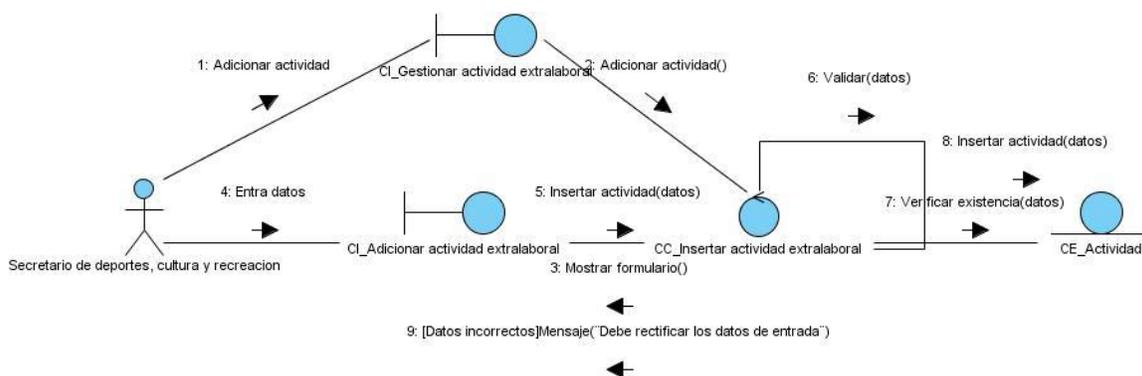


Fig.A3.1 DC Adicionar actividades extra laborales

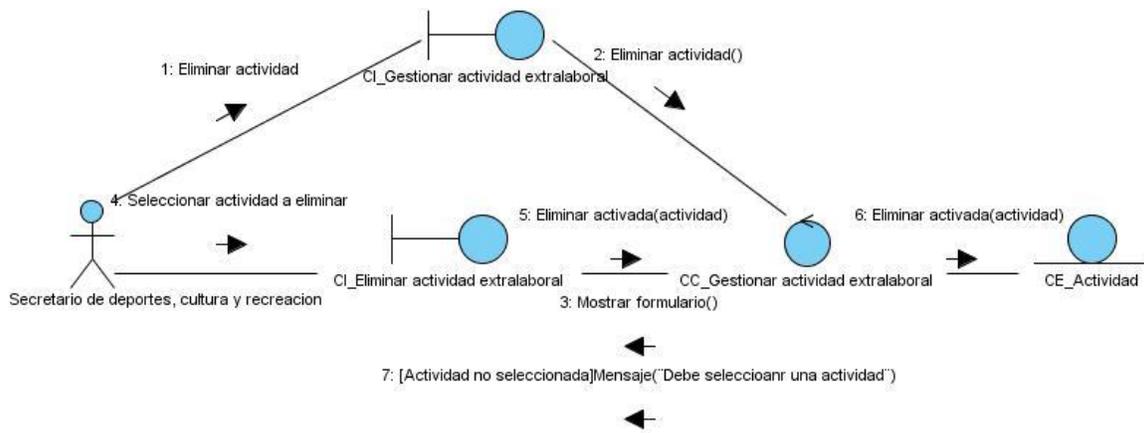


Fig.A3.2 DC Eliminar actividades extra laborales

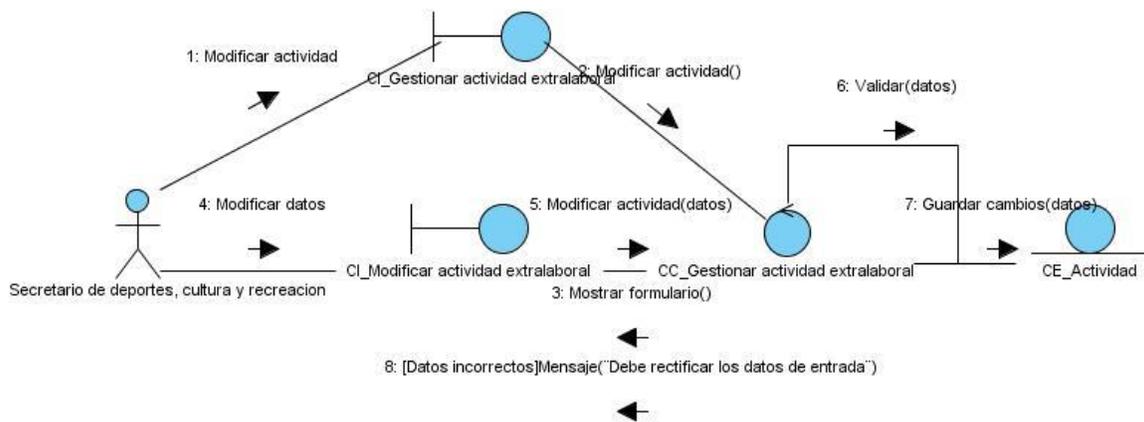


Fig.A3.3 DC Modificar actividades extra laborales.

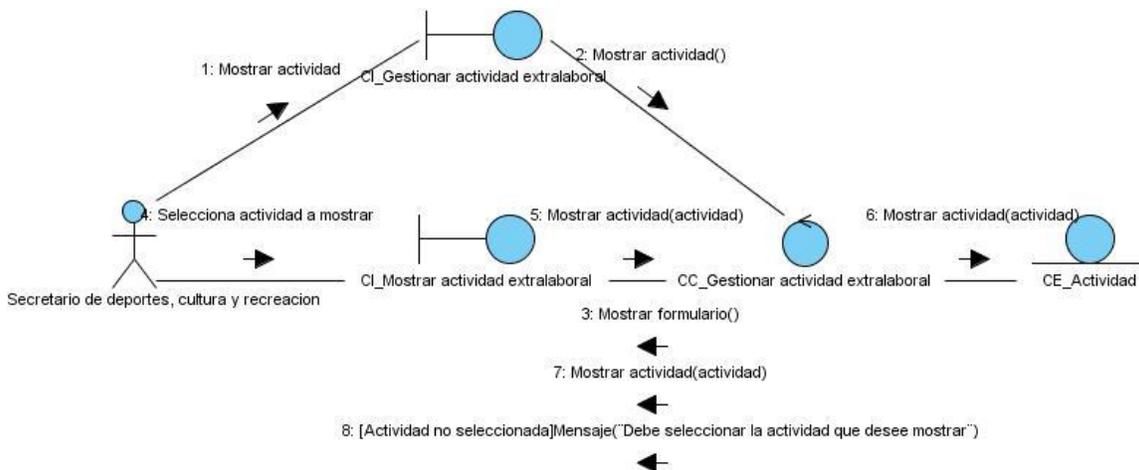


Fig.A3.4 DC Mostrar actividades extra laborales.

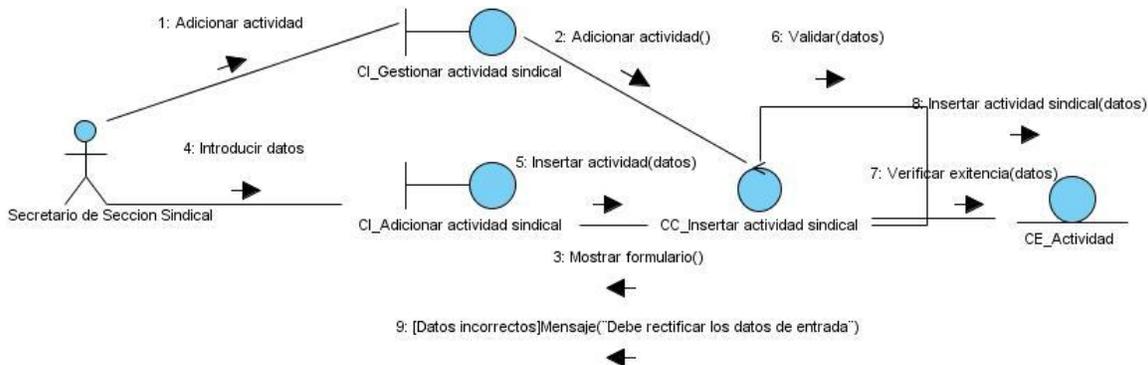


Fig.A3.5 DC Adicionar actividades sindicales.

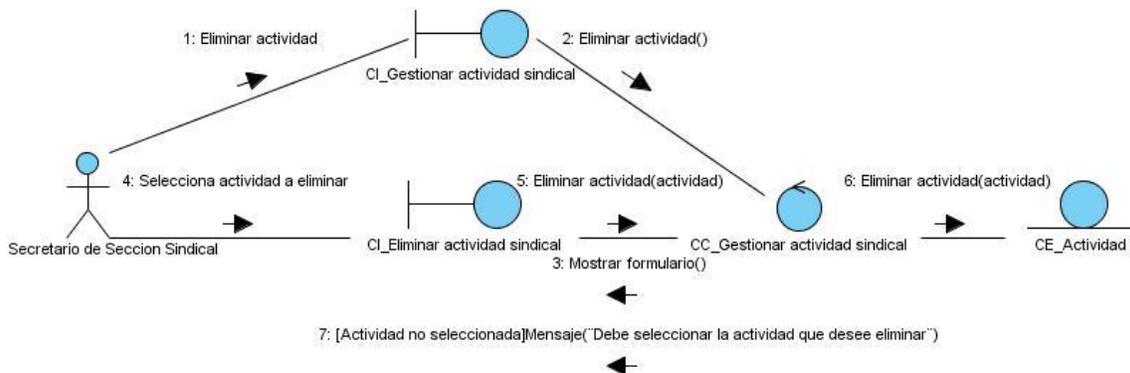


Fig.A3.6 DC Eliminar actividades sindicales.

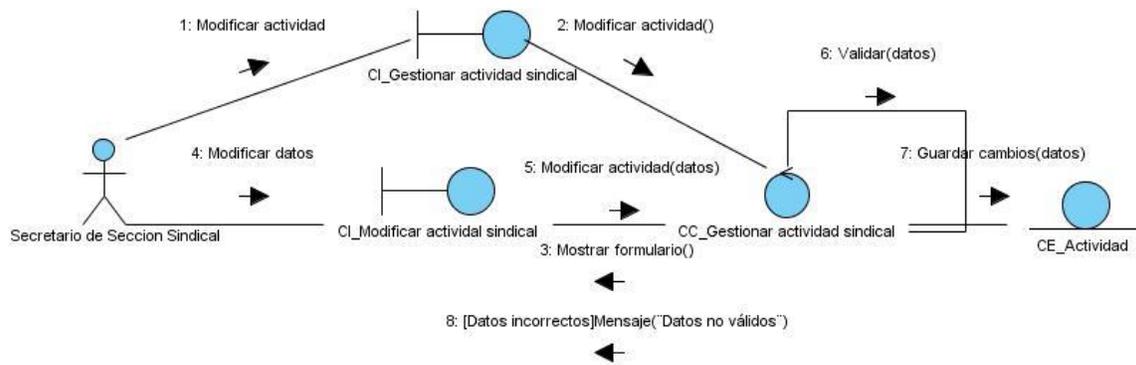


Fig.A3.7 DC Modificar actividades sindicales.

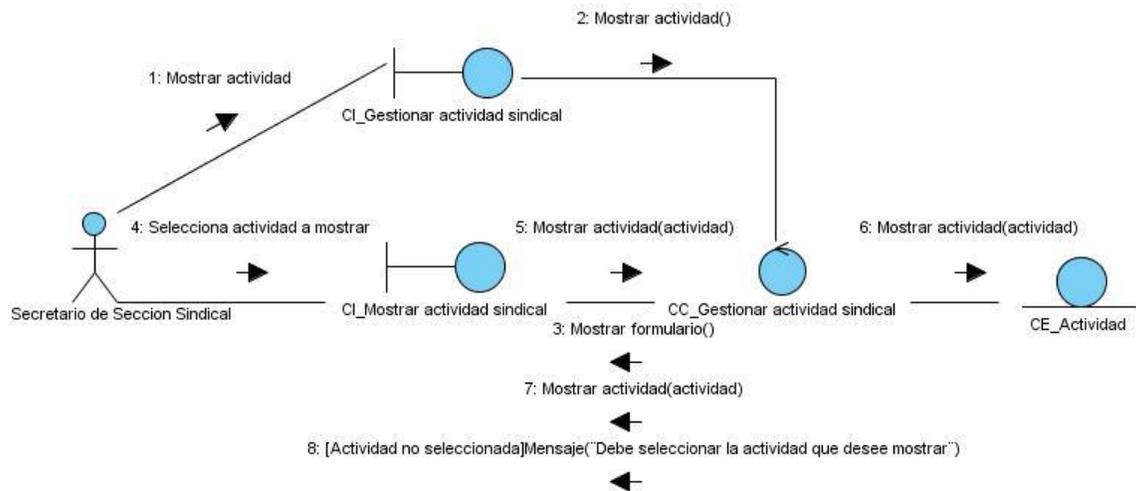


Fig.A3.8 DC Mostrar actividades sindicales.

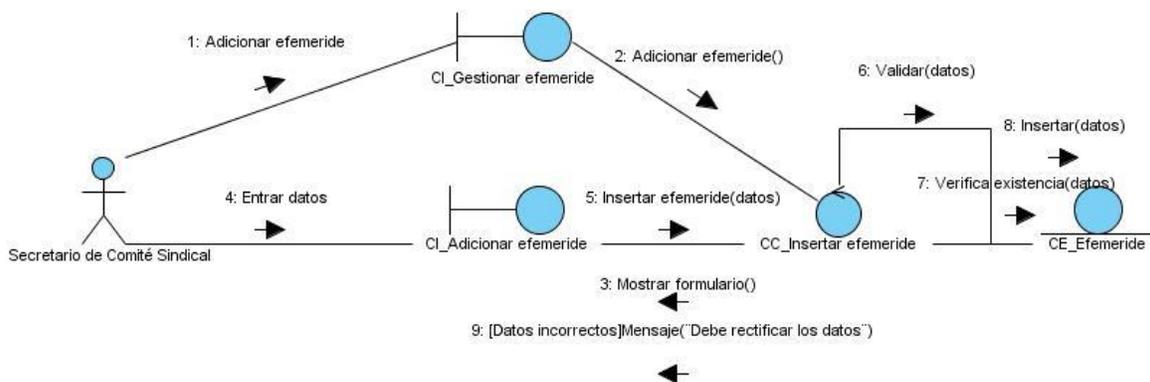


Fig.A3.9 DC Adicionar efemérides.

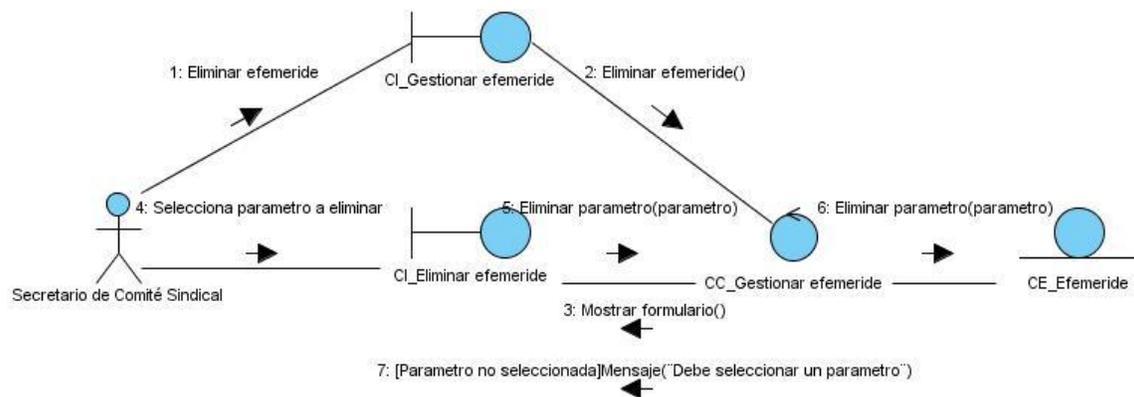


Fig.A3.10 DC Eliminar efemérides.

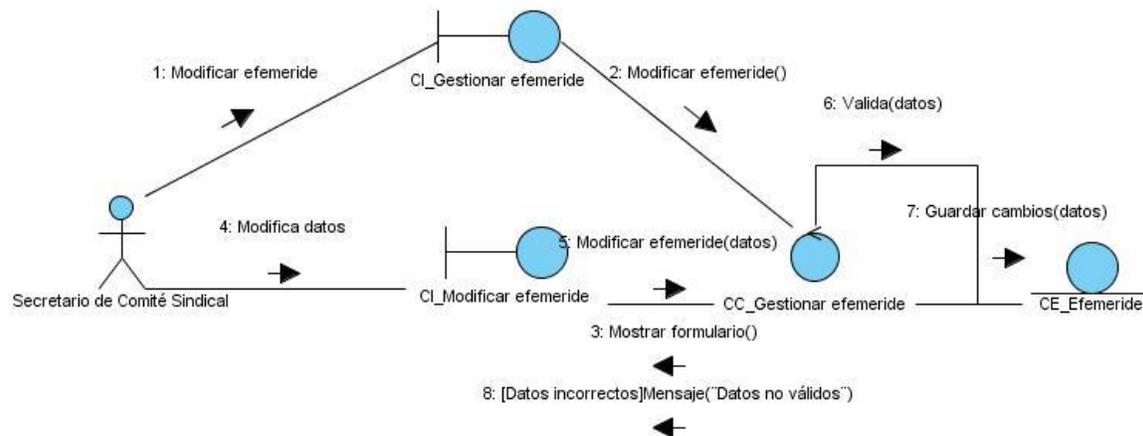


Fig.A3.11 DC Modificar efemérides.

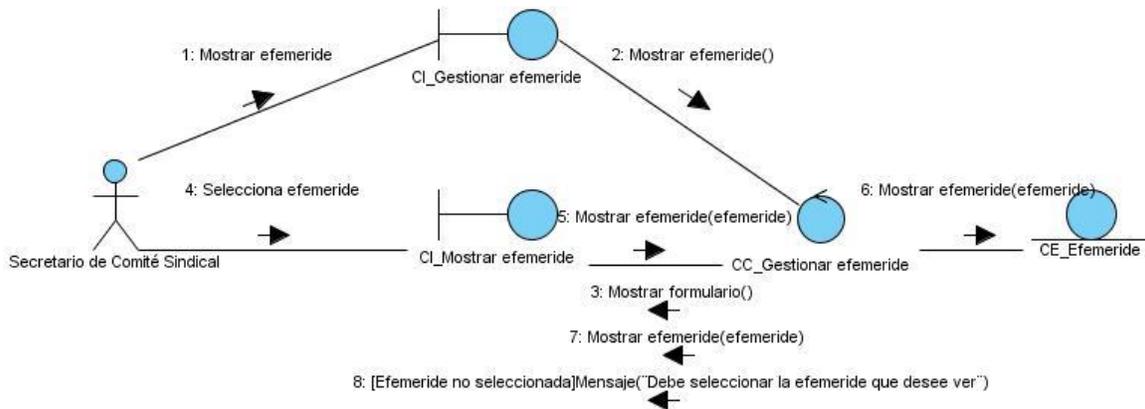


Fig.A3.12 DC Mostrar efemérides.

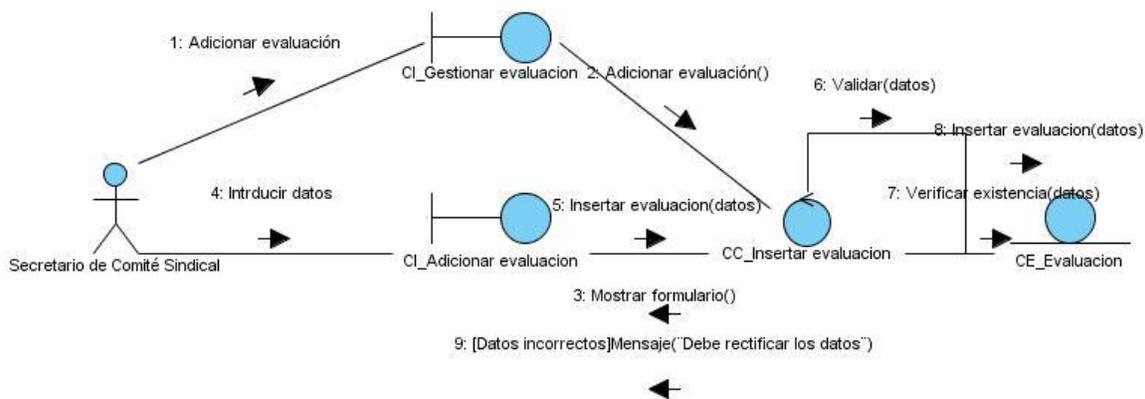


Fig.A3.13 DC Adicionar evaluación de afiliado.

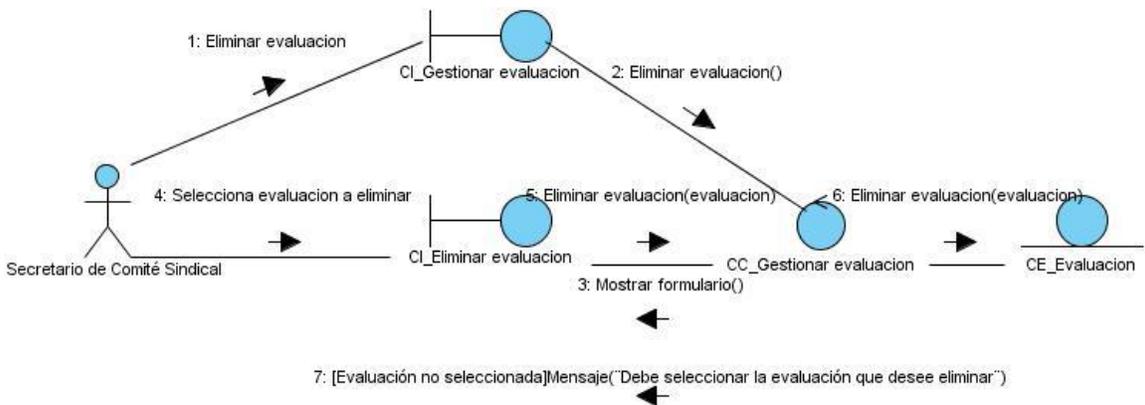


Fig.A3.14 DC Eliminar evaluación de afiliado.

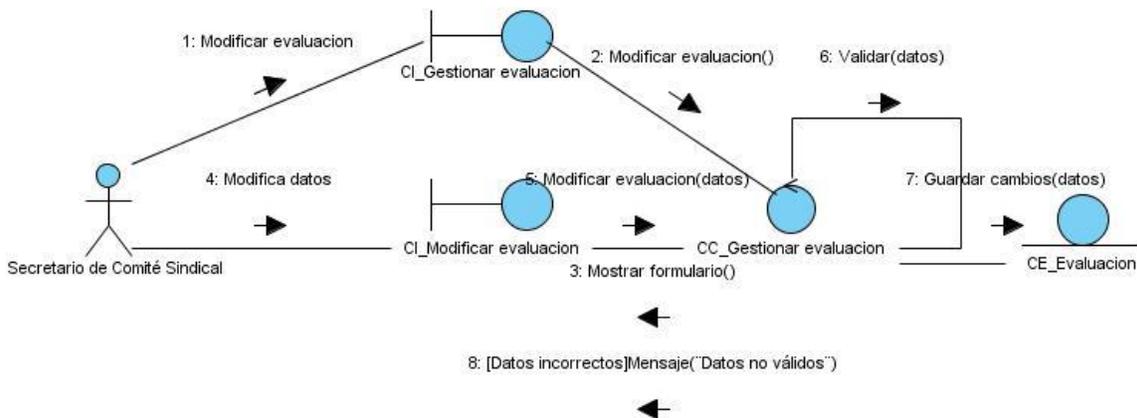


Fig.A3.15 DC Modificar evaluación de afiliado.

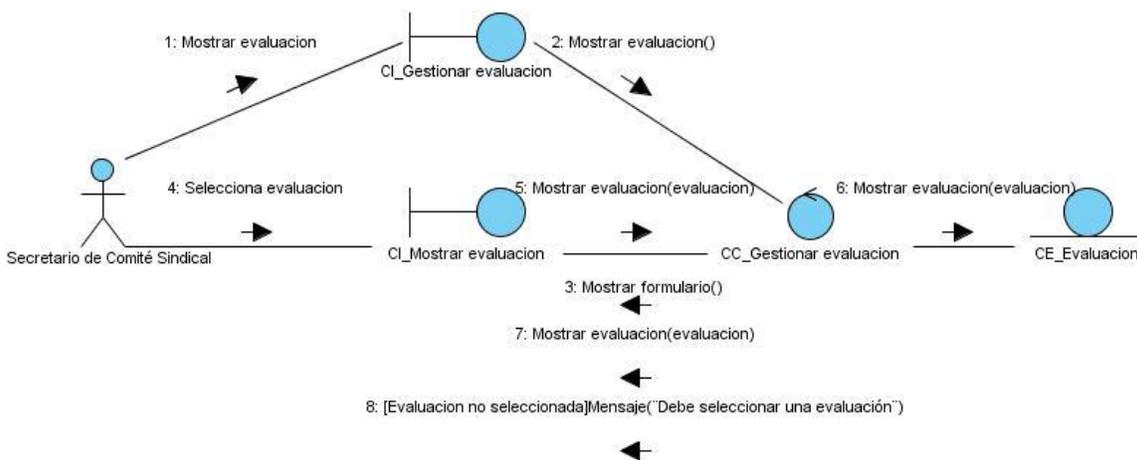


Fig.A3.16 DC Mostrar evaluación de afiliado.

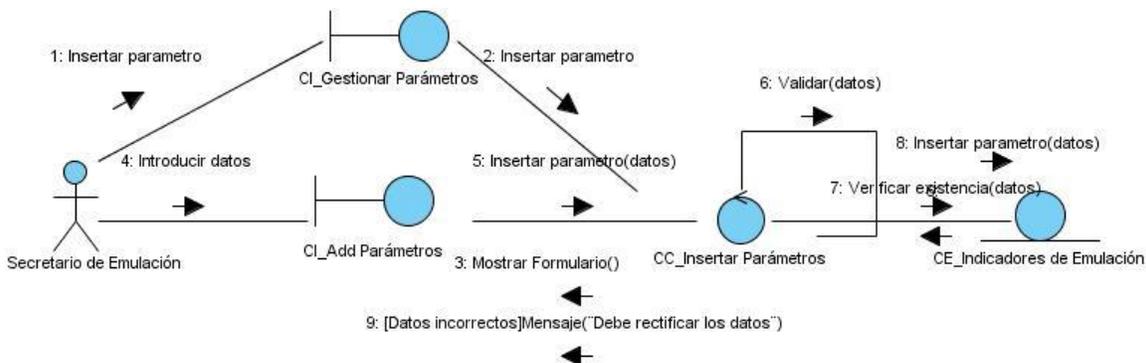


Fig.A3.17 DC Adicionar parámetros.

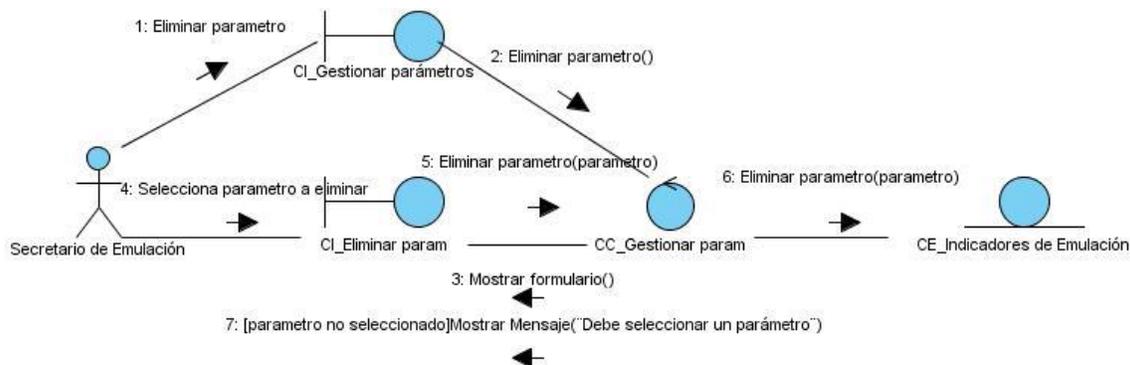


Fig.A3.18 DC Eliminar parámetros.

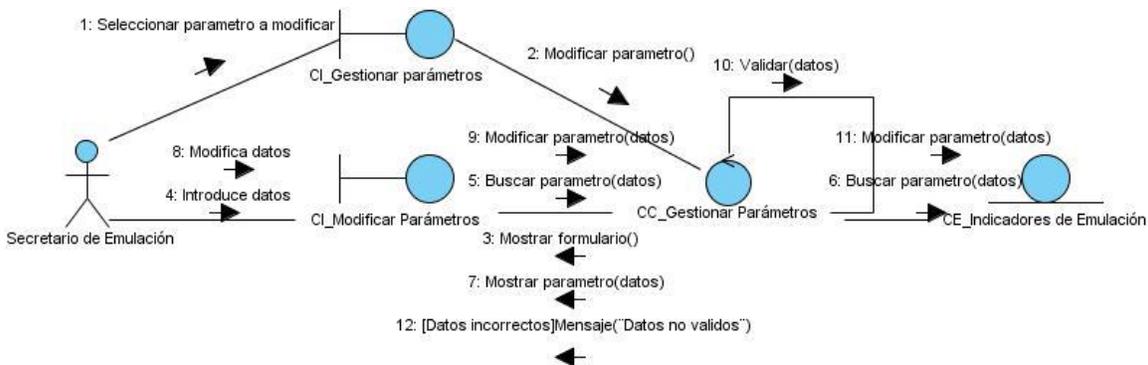


Fig.A3.19 DC Modificar parámetros.

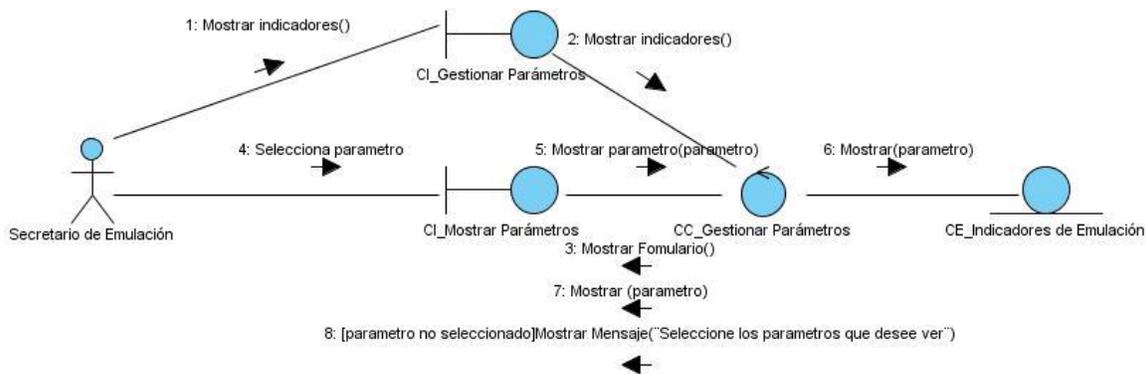


Fig.A3.20 DC Mostrar parámetros.

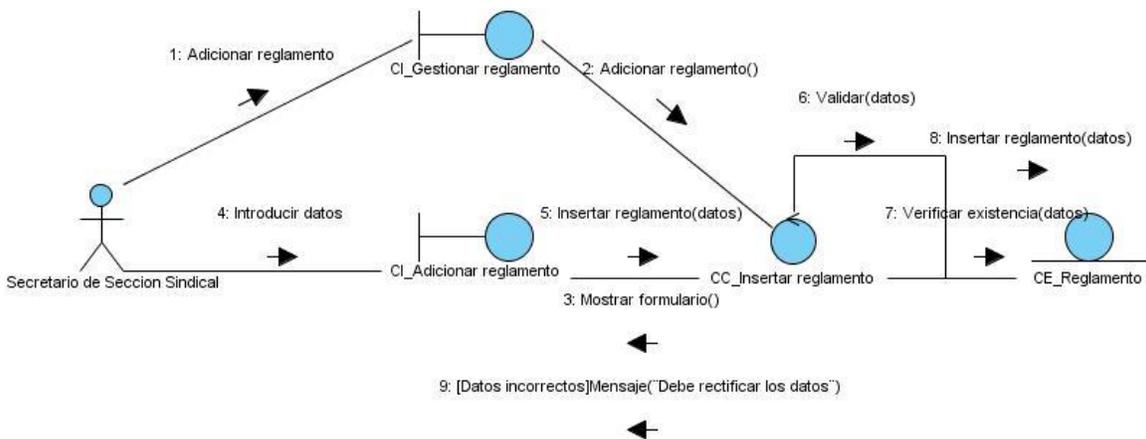


Fig.A3.21 DC Adicionar reglamento.

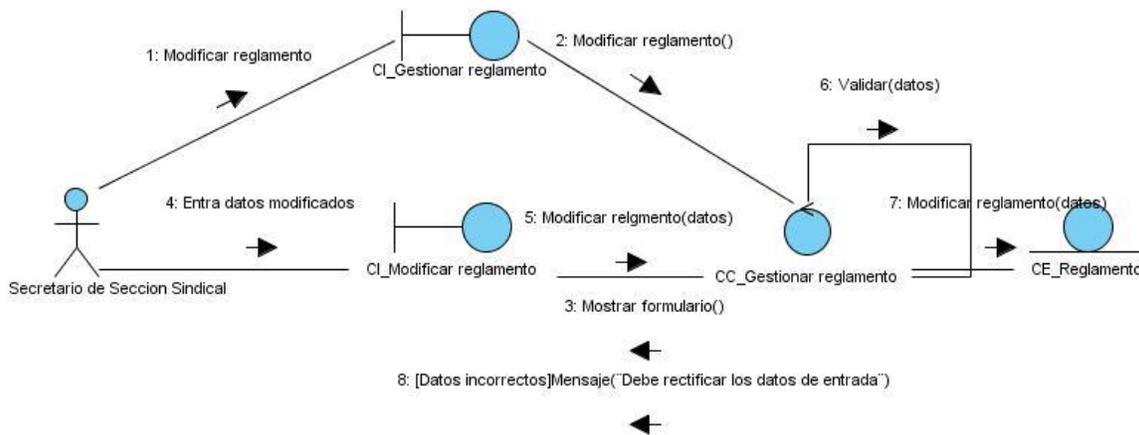


Fig.A3.22 DC Modificar reglamento.

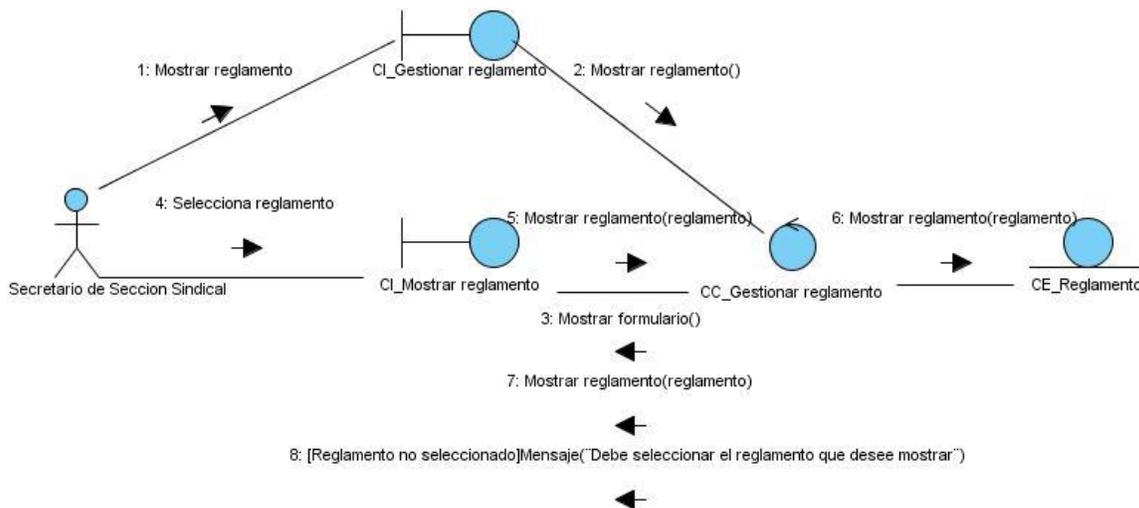


Fig.A3.23 DC Mostrar reglamento.

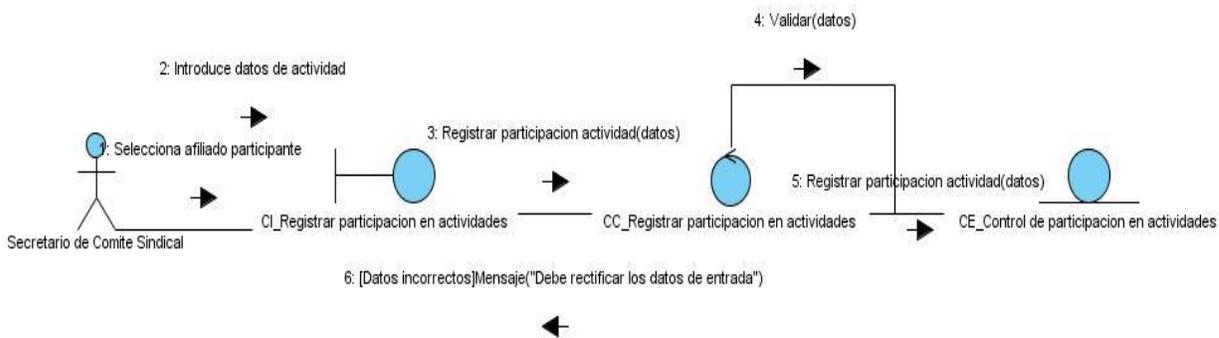


Fig.A3.24 DC Registrar participación en actividades.

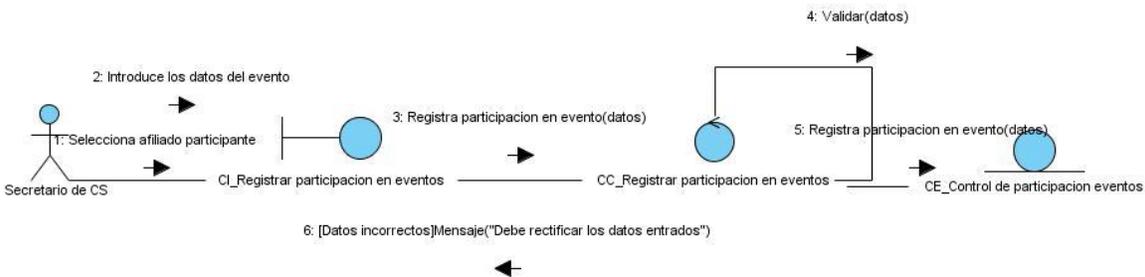


Fig.A3.25 DC Registrar participación en eventos.

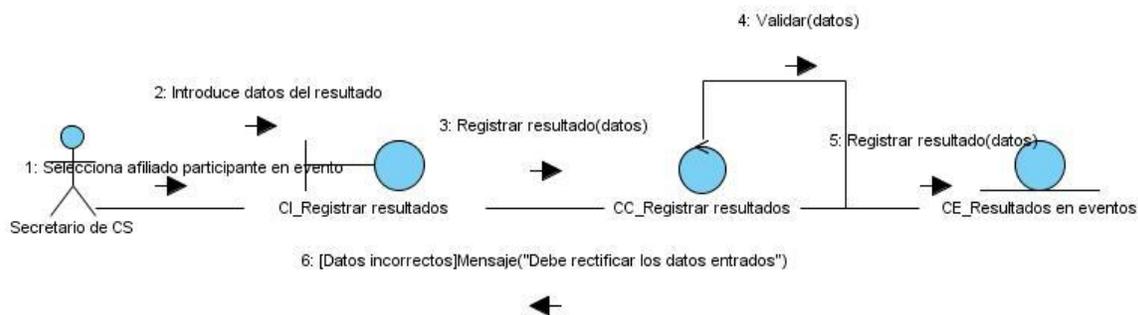


Fig.A3.26 DC Registrar resultados.

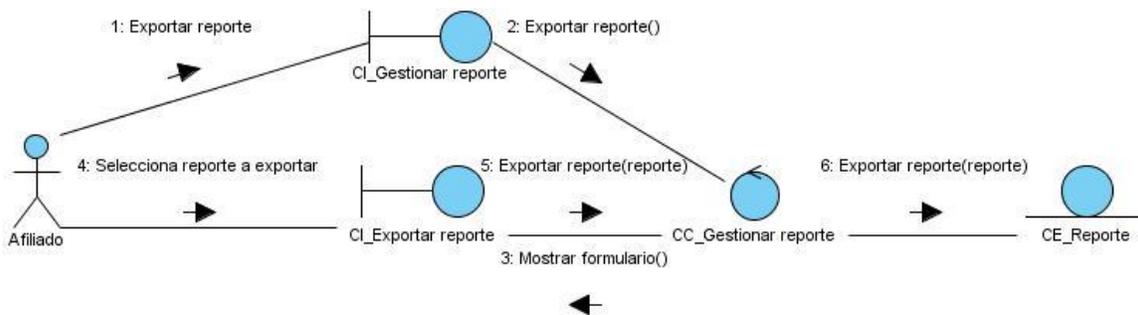


Fig.A3.27 DC Exportar reporte.

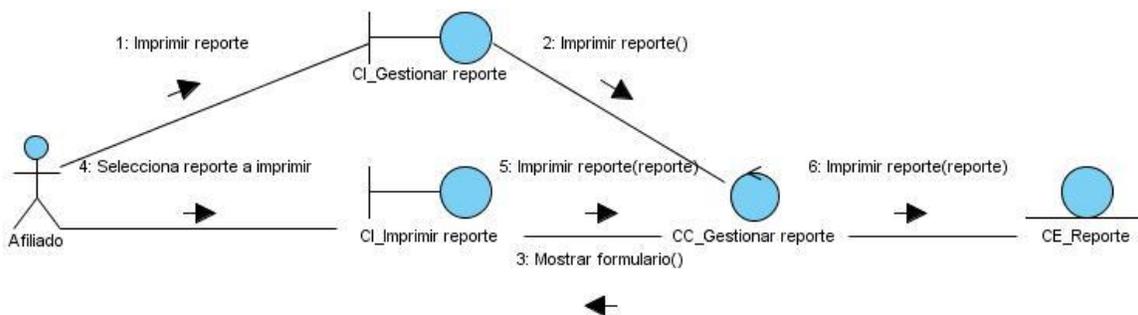


Fig.A3.28 DC Imprimir reporte.

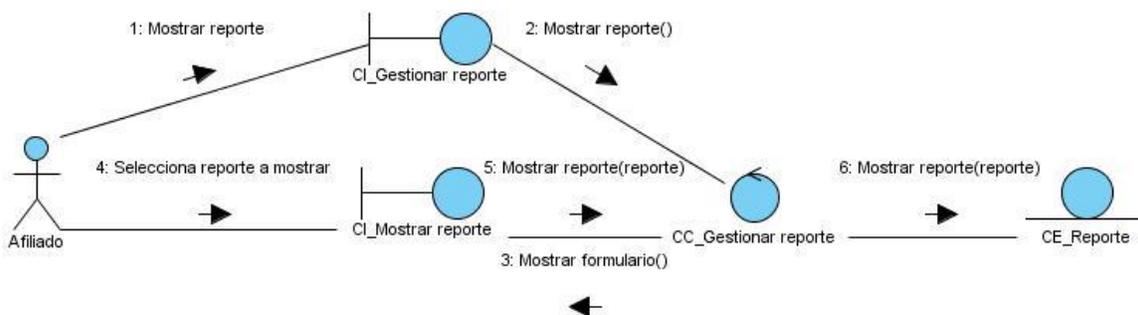


Fig.A3.29 DC Mostrar reporte.

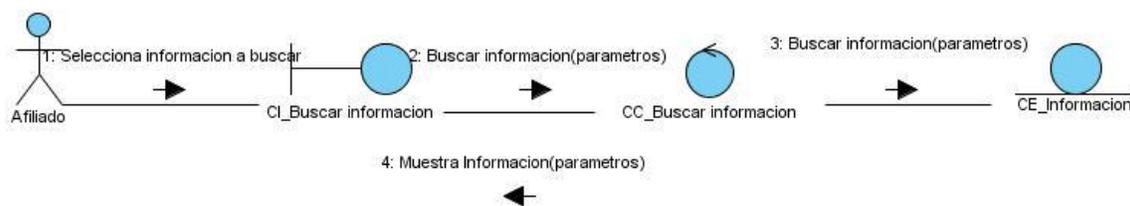


Fig.A3.30 DC Buscar información.

Anexo # 4 Diagrama de clases del diseño

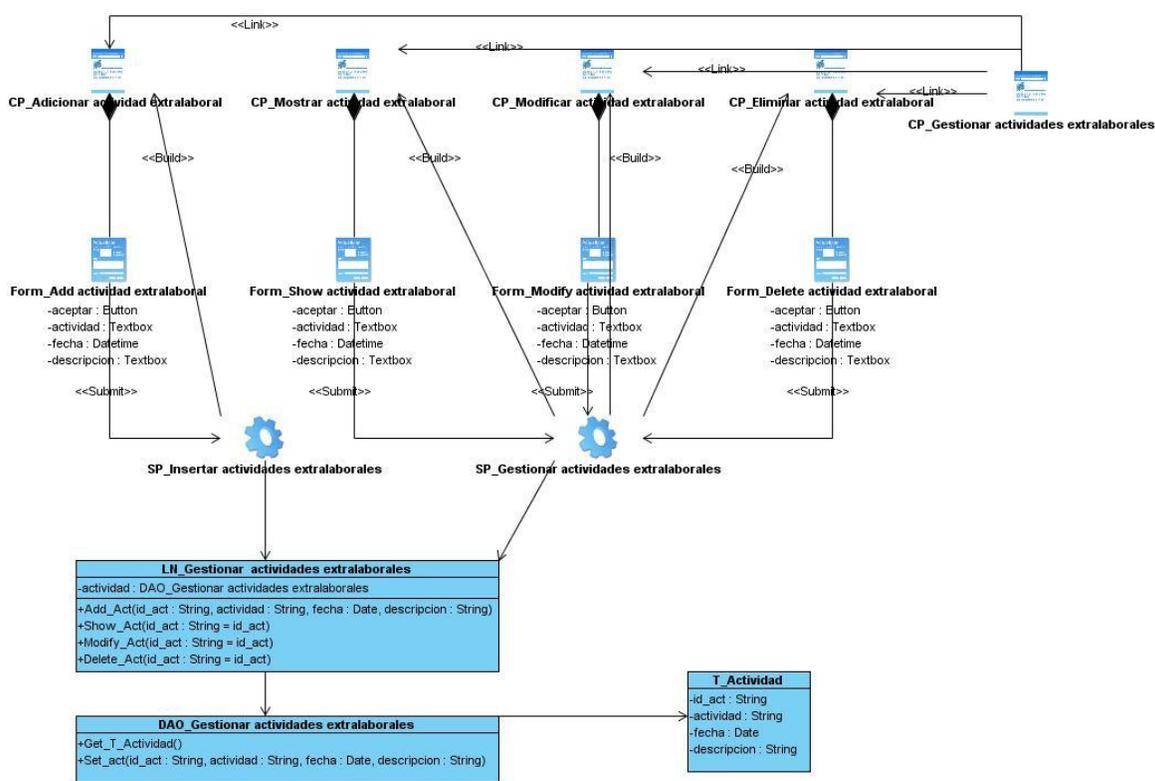


Fig.A4.1 DCD Gestionar actividades extra laborales.

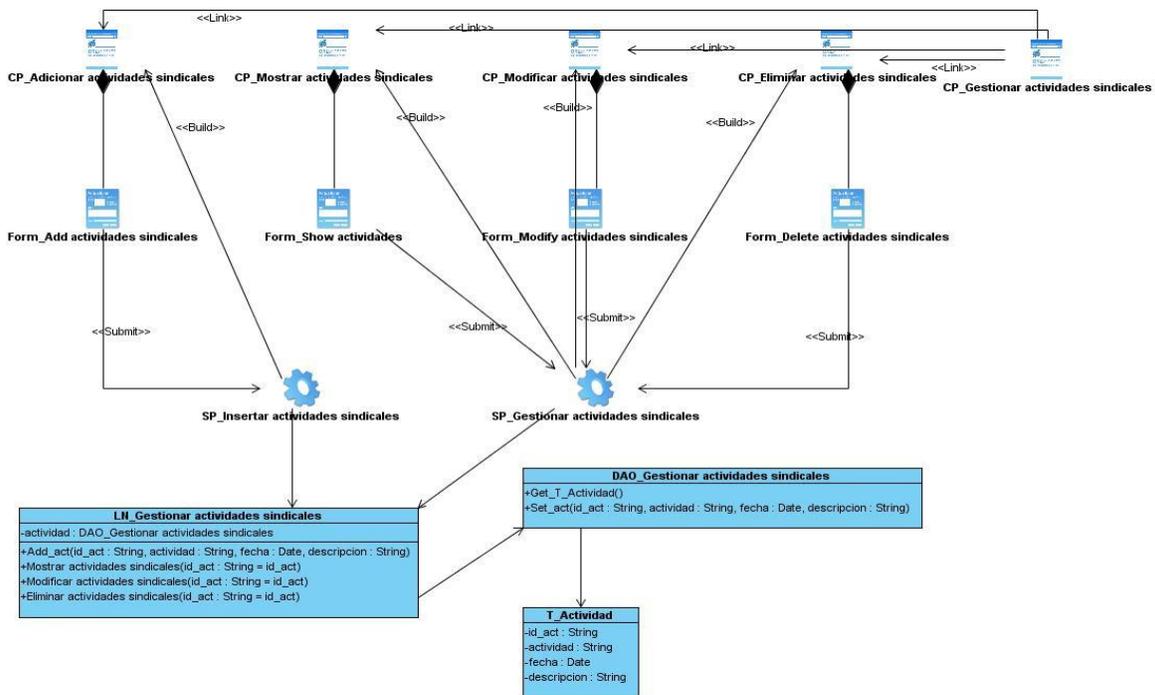


Fig.A4.2 DCD Gestionar actividades sindicales.

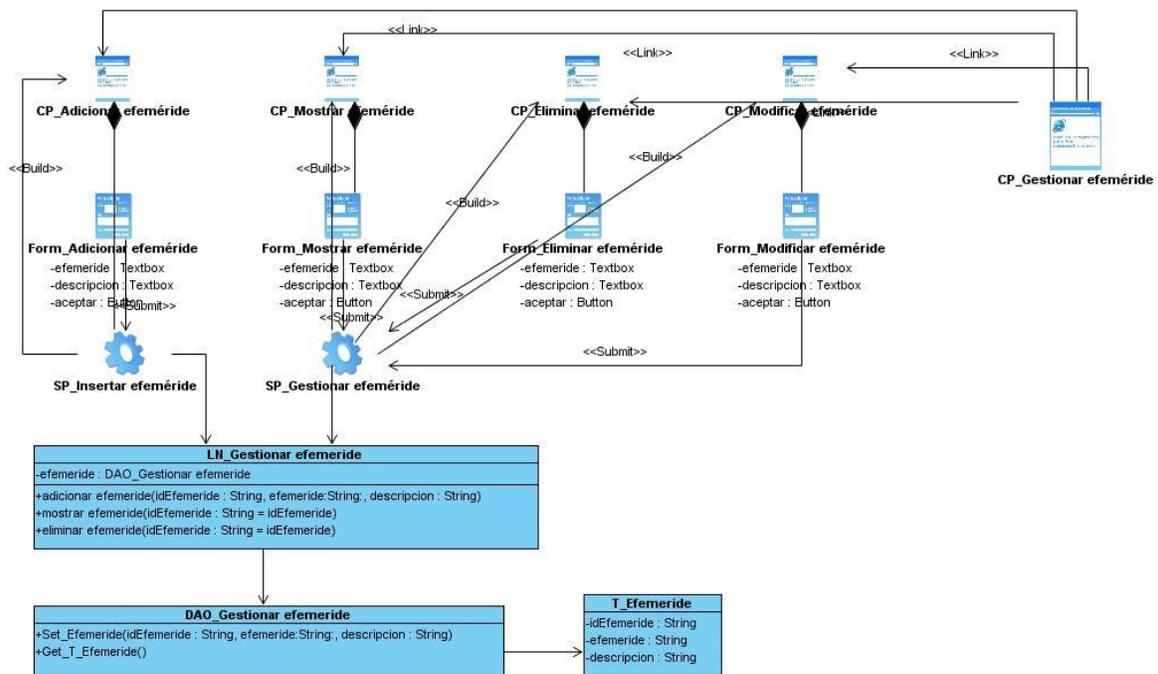


Fig.A4.3 DCD Gestionar efemérides.

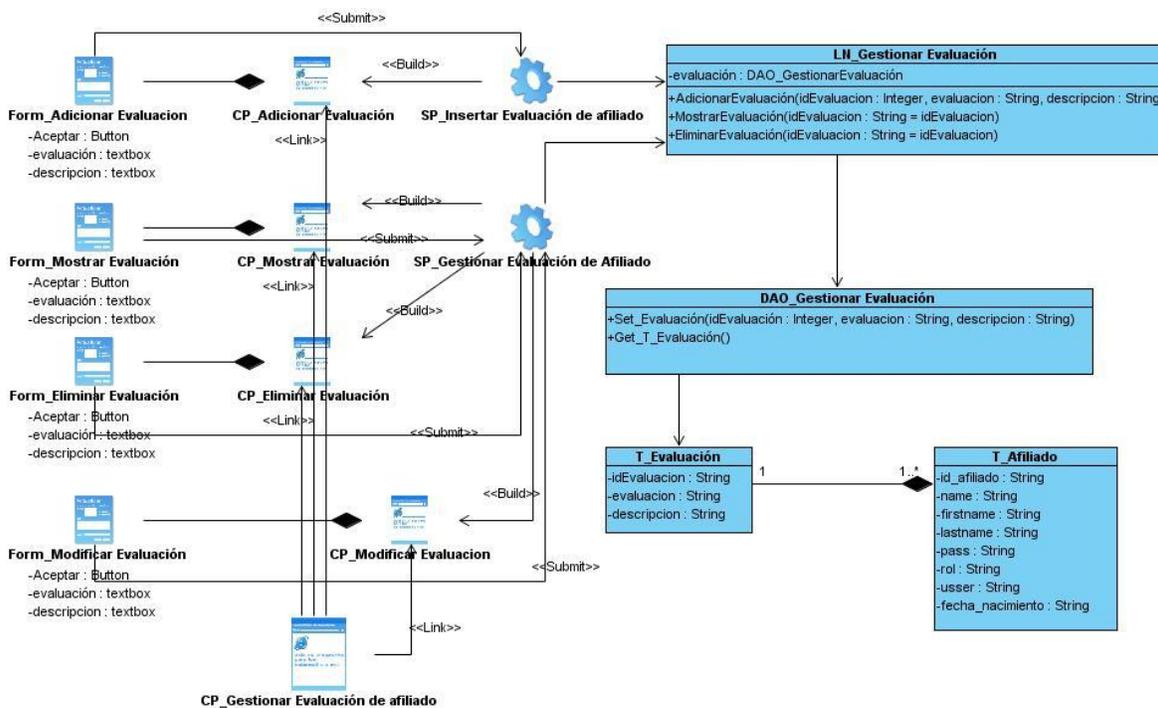


Fig.A4.4 DCD Gestionar evaluación de afiliado.

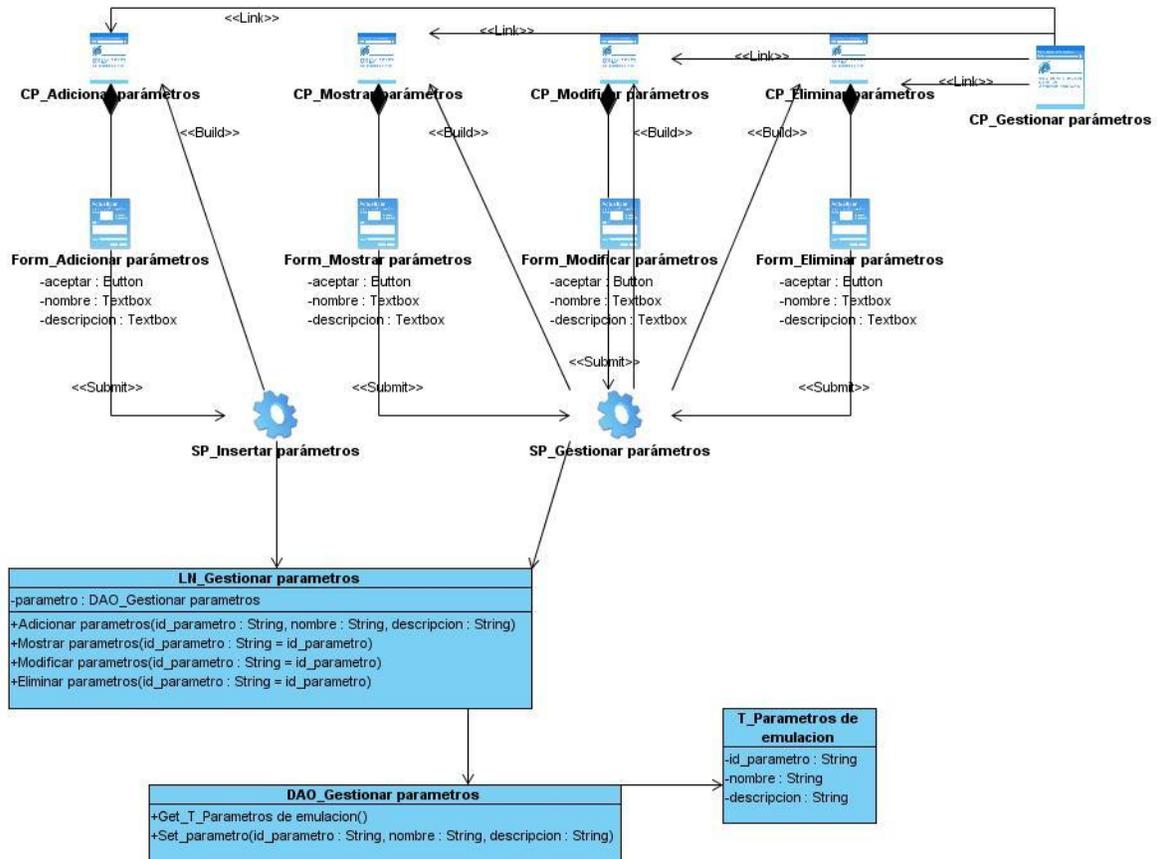


Fig.A4.5 DCD Gestionar parámetros.

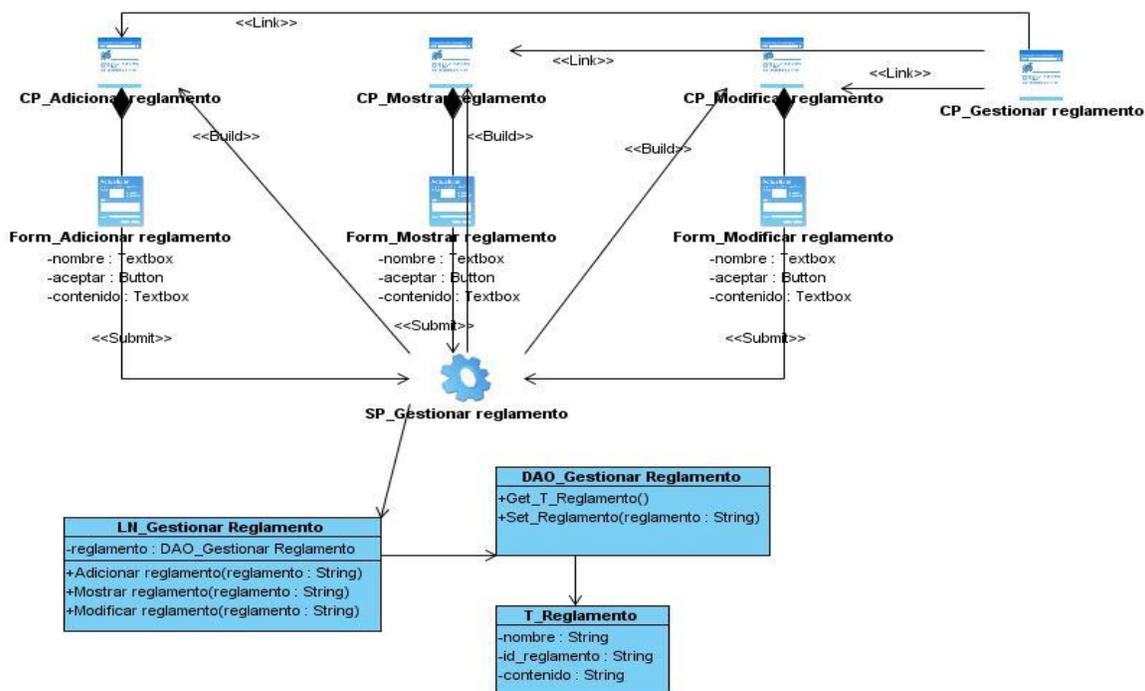


Fig.A4.6 DCD Gestionar reglamentos.

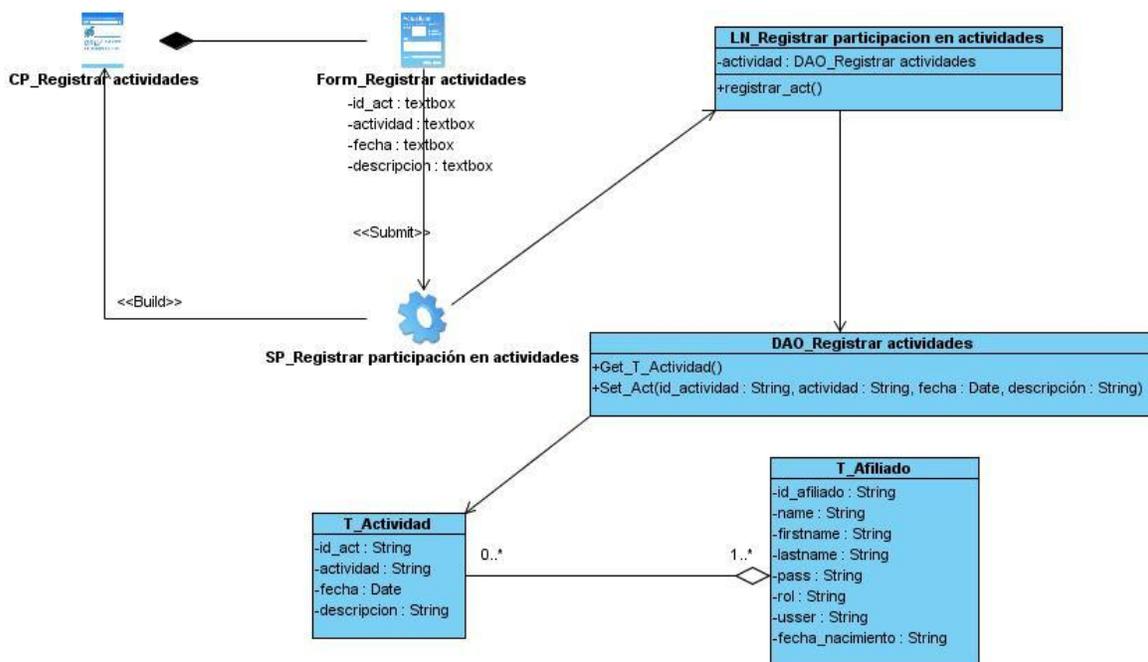


Fig.A4.7 DCD Registrar participación en actividades.

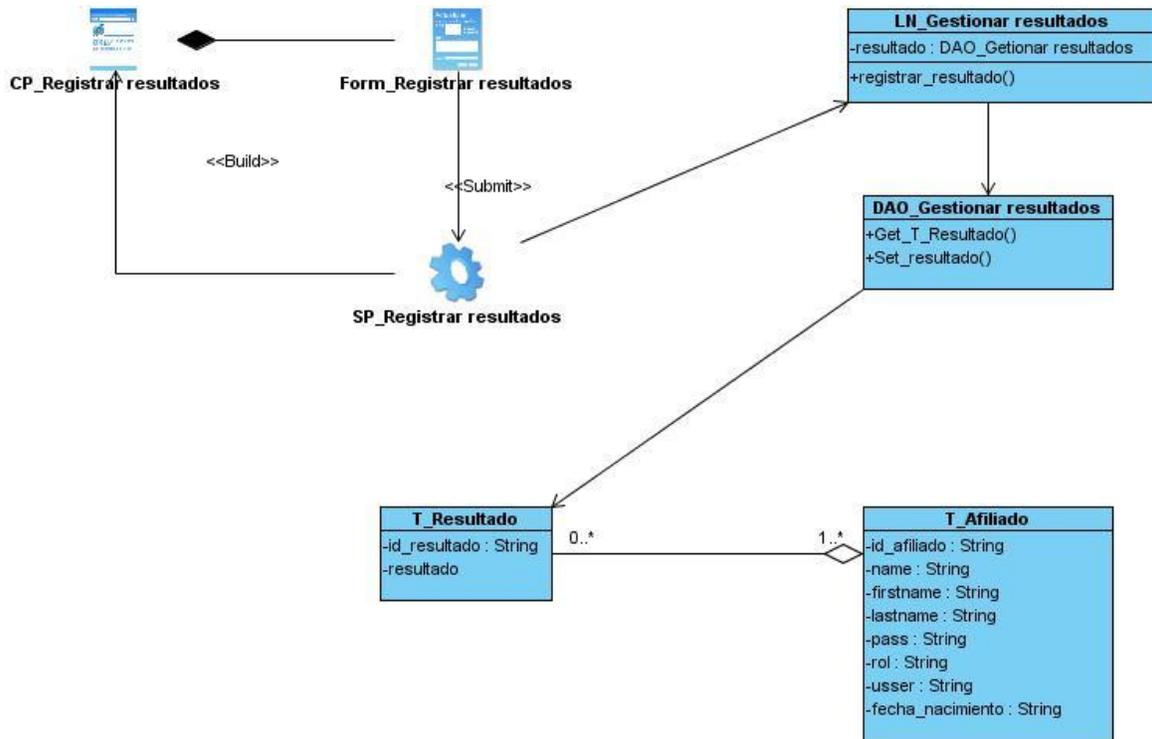


Fig.A4.10 DCD Registrar resultados.

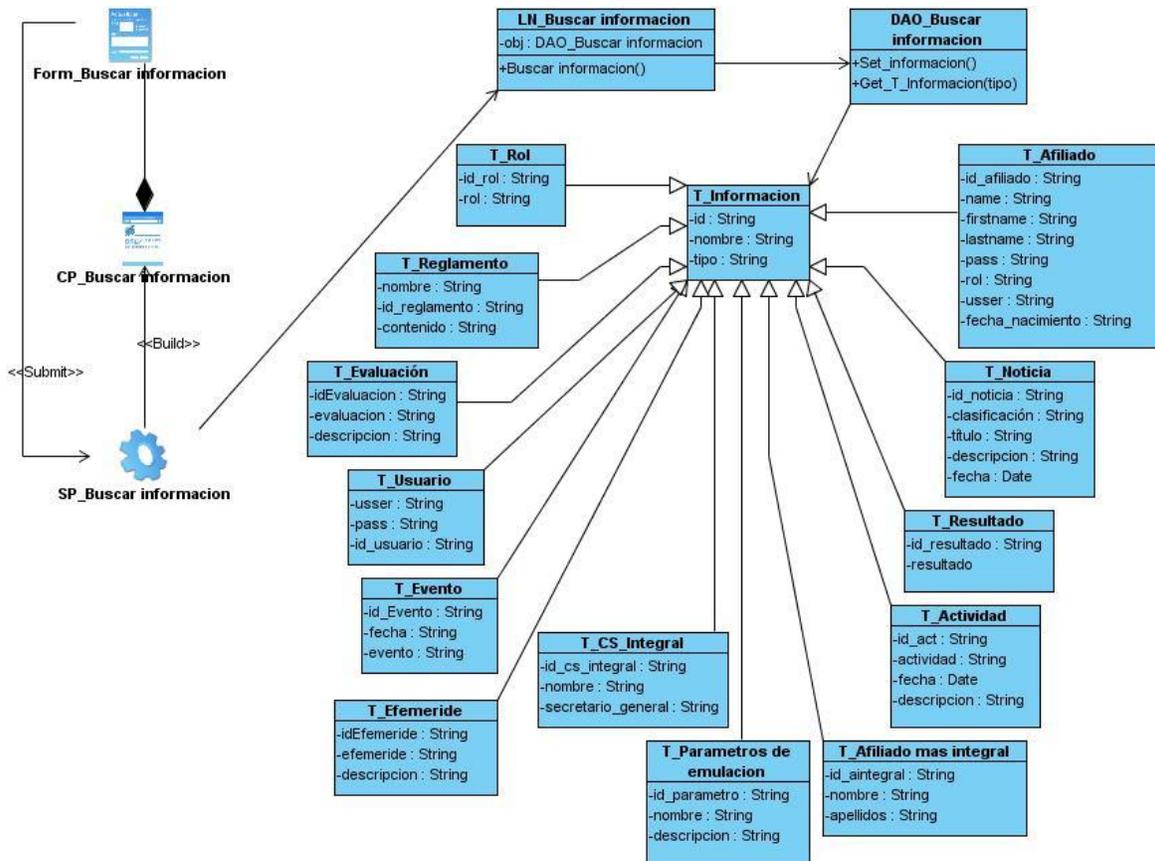


Fig.A4.11 DCD Buscar información.

Anexo # 5 Diagramas de secuencia

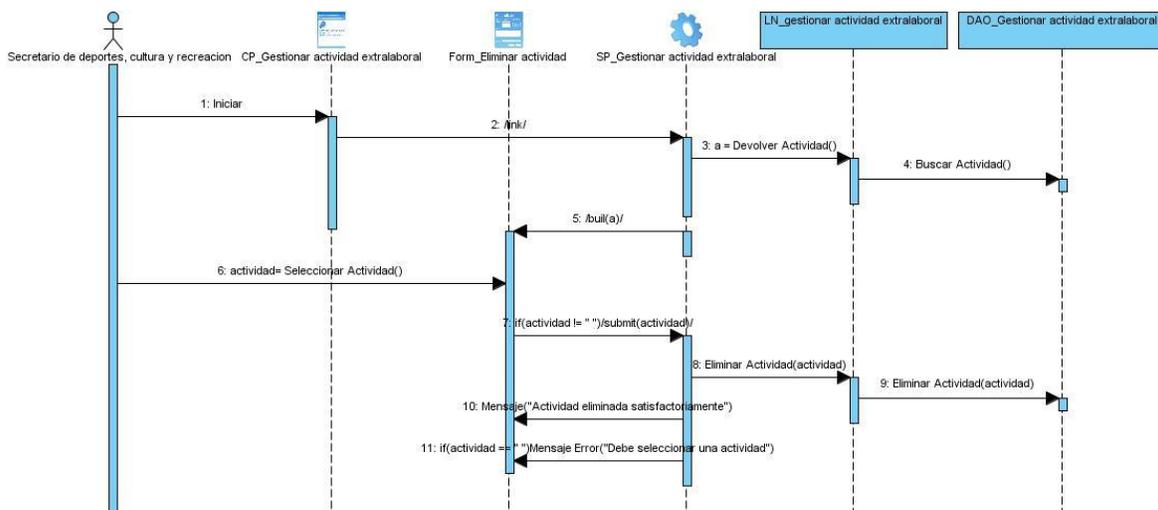


Fig.A5.1 DS Eliminar actividad extra laboral.

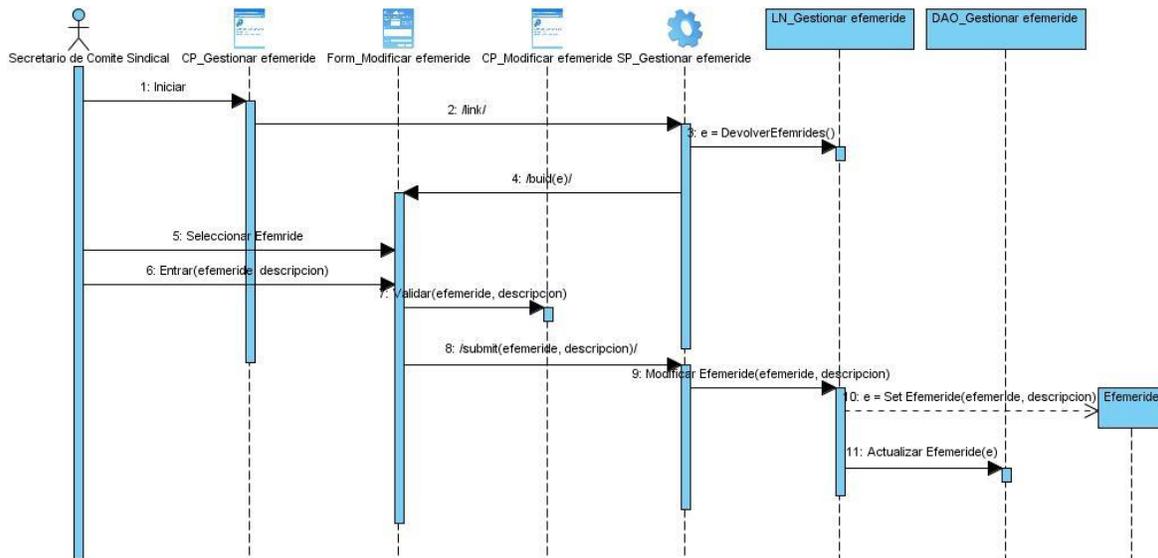


Fig.A5.2 DS Modificar efemerides.

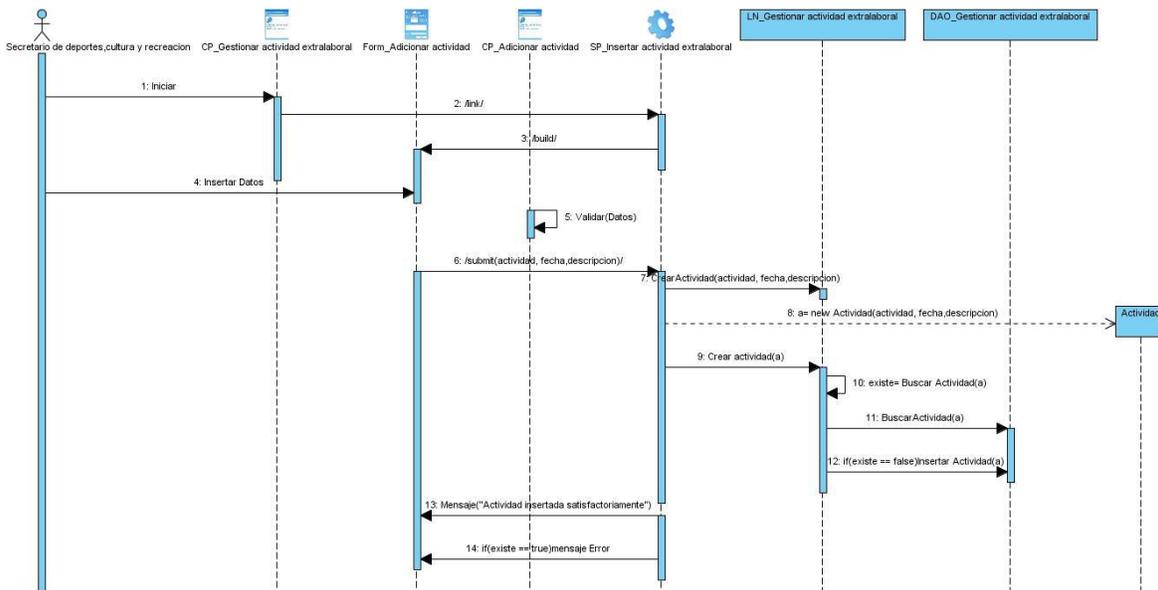


Fig.A5.3 DS Adicionar actividad extra laboral.

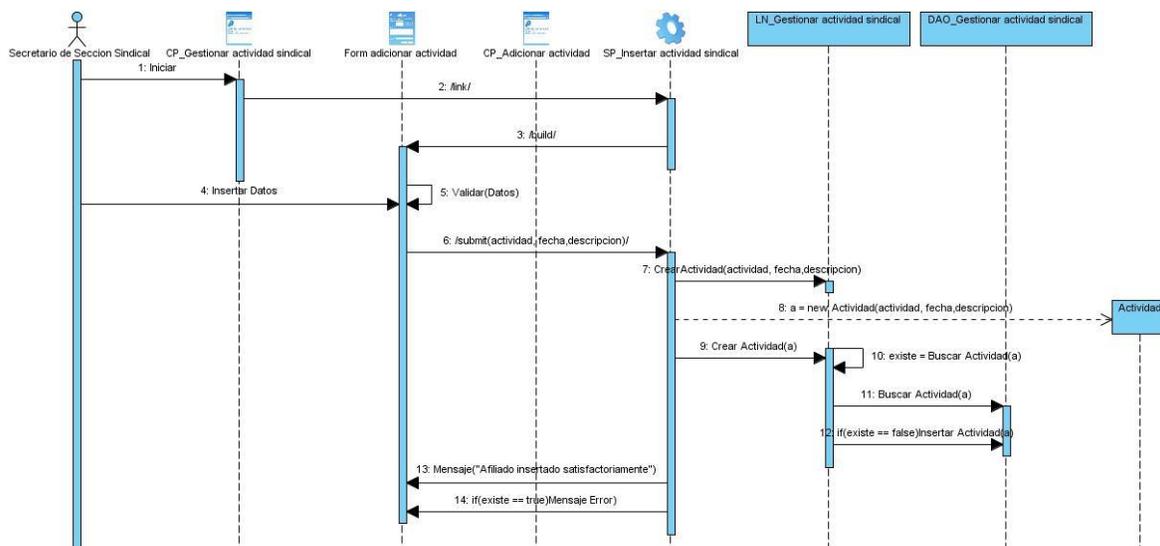


Fig.A5.4 DS Adicionar actividad sindical.

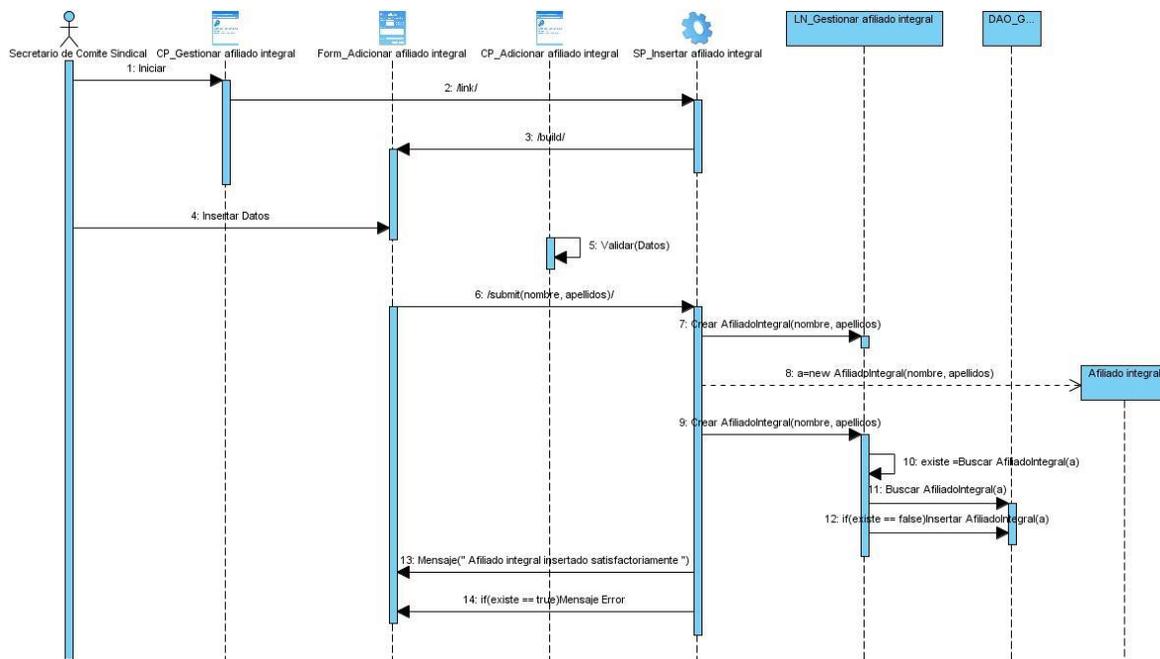


Fig.A5.5 DS Adicionar afiliado más integral.

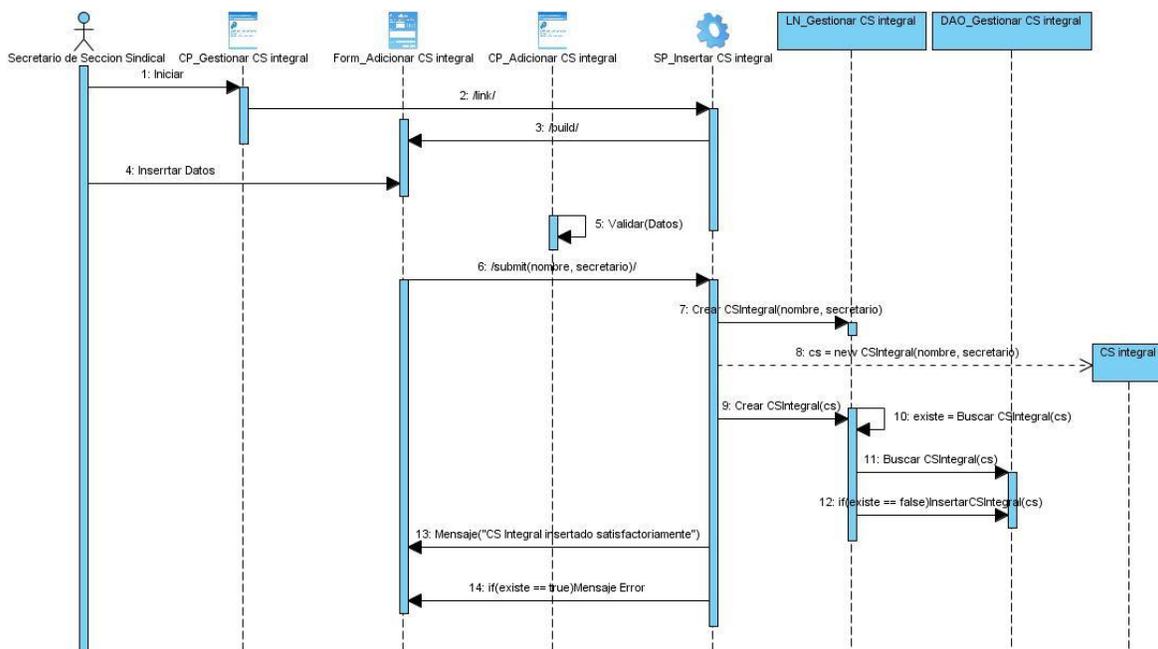


Fig.A5.6 DS Adicionar c/s más integral.

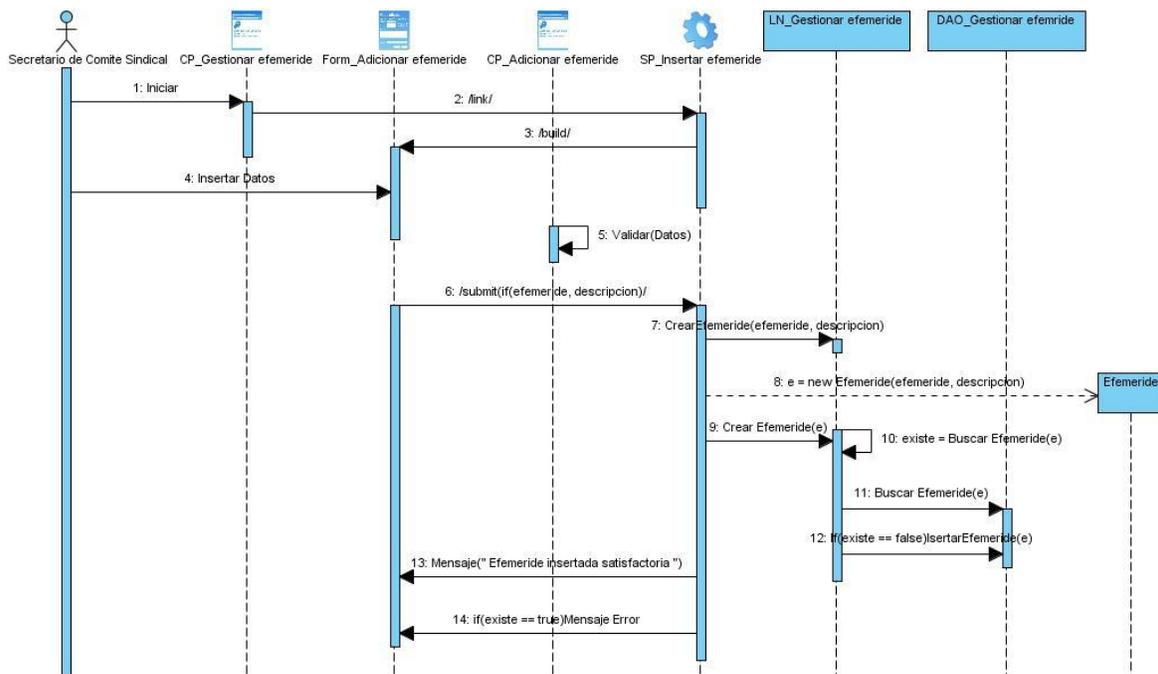


Fig.A5.7 DS Adicionar efemérides.

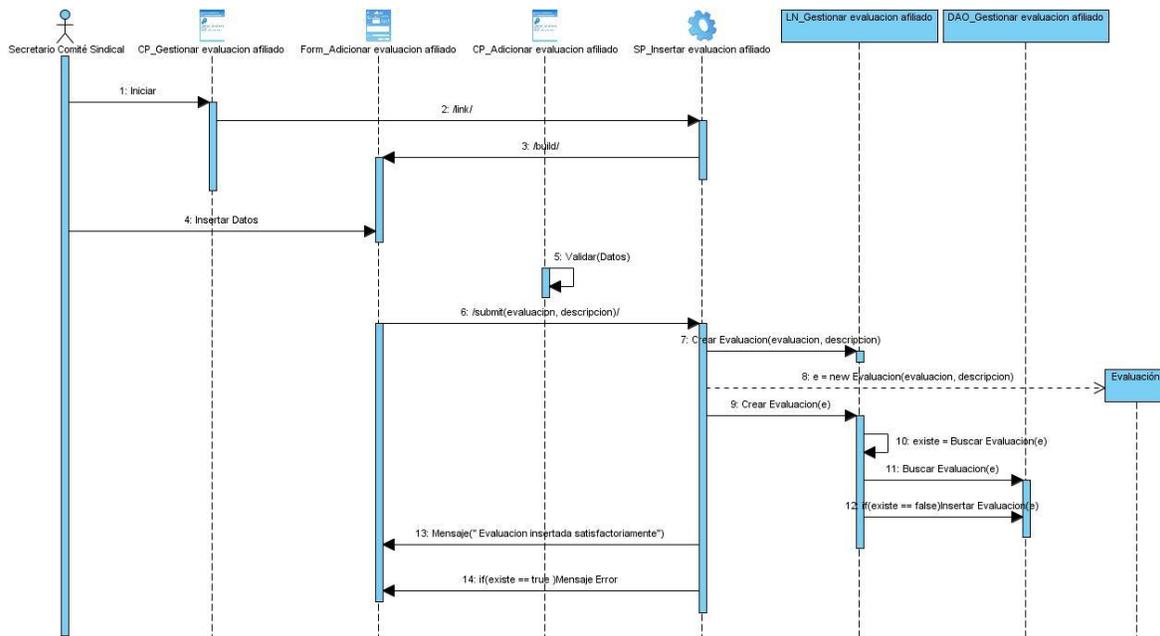


Fig.A5.8 DS Adicionar evaluación de afiliado.

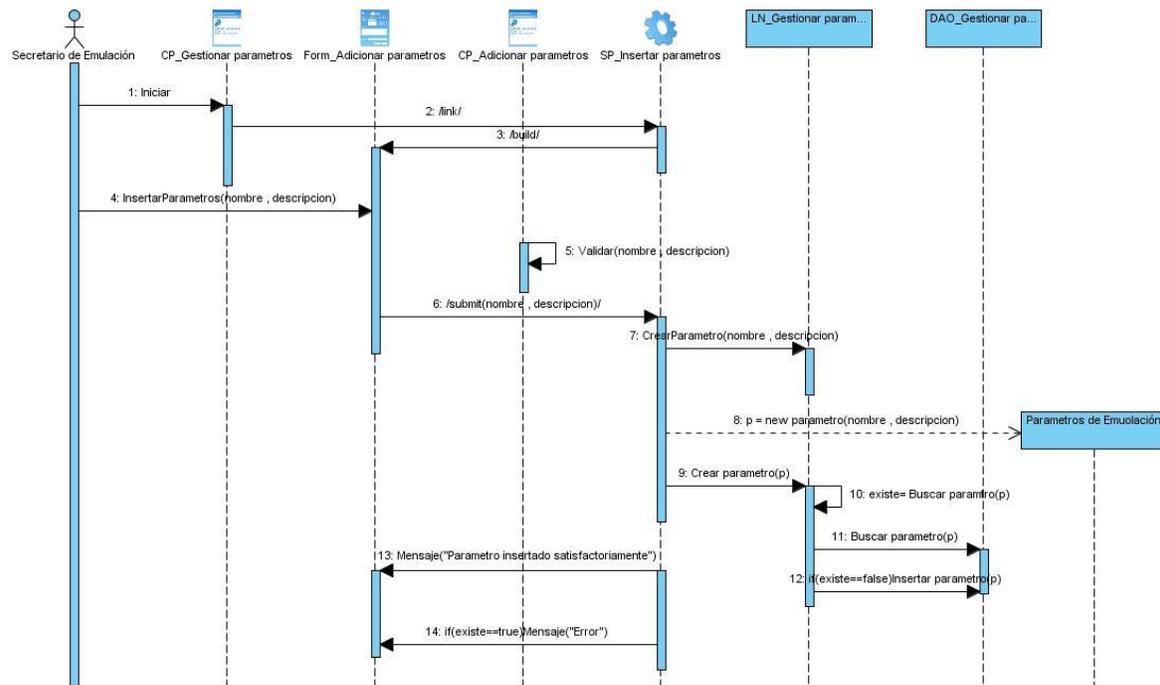


Fig.A5.9 DS Adicionar parámetros.

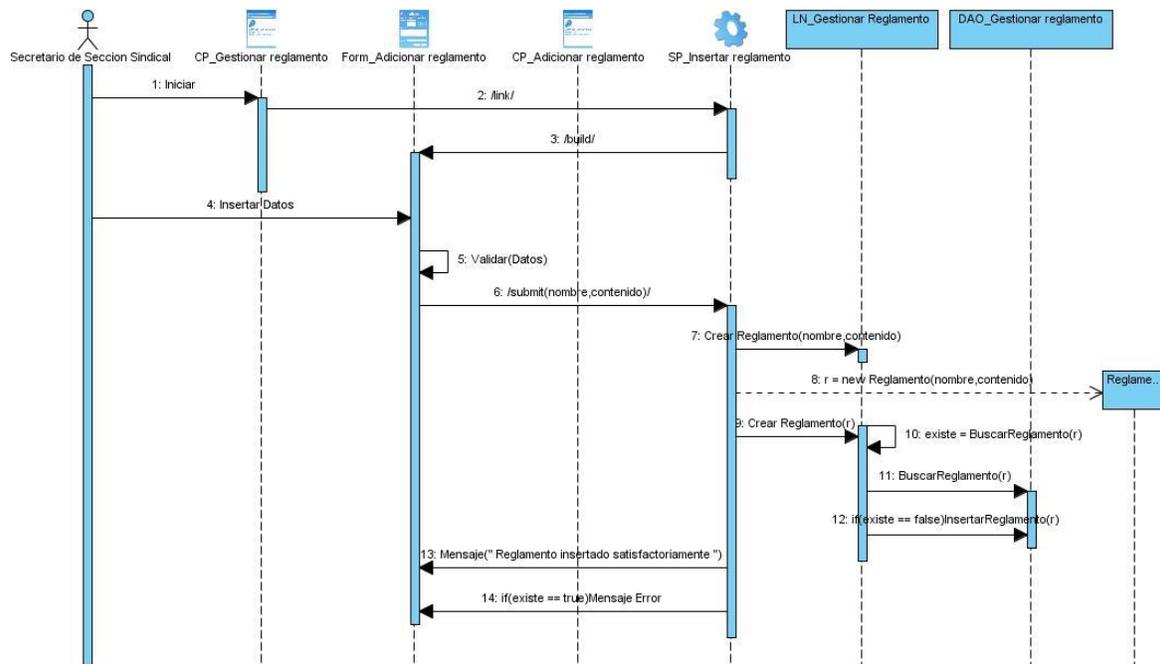


Fig.A5.10 DS Adicionar reglamento sindical.

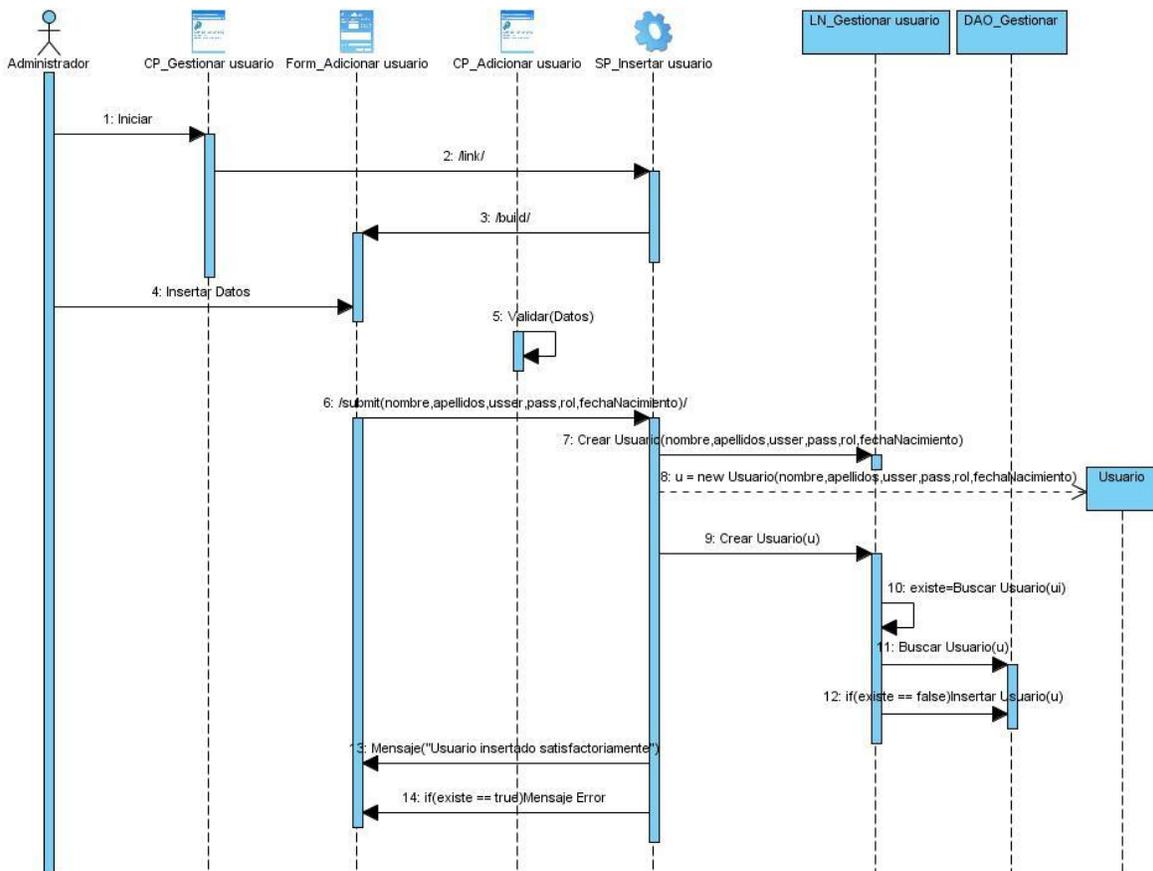


Fig.A5.11 DS Adicionar usuario.

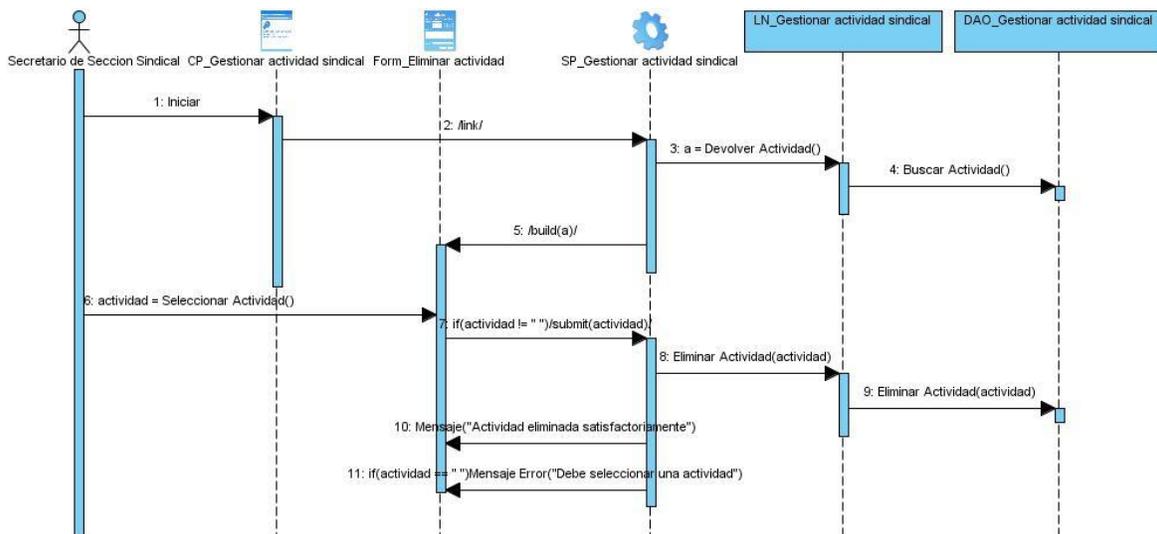


Fig.A5.12 DS Eliminar actividad sindical.

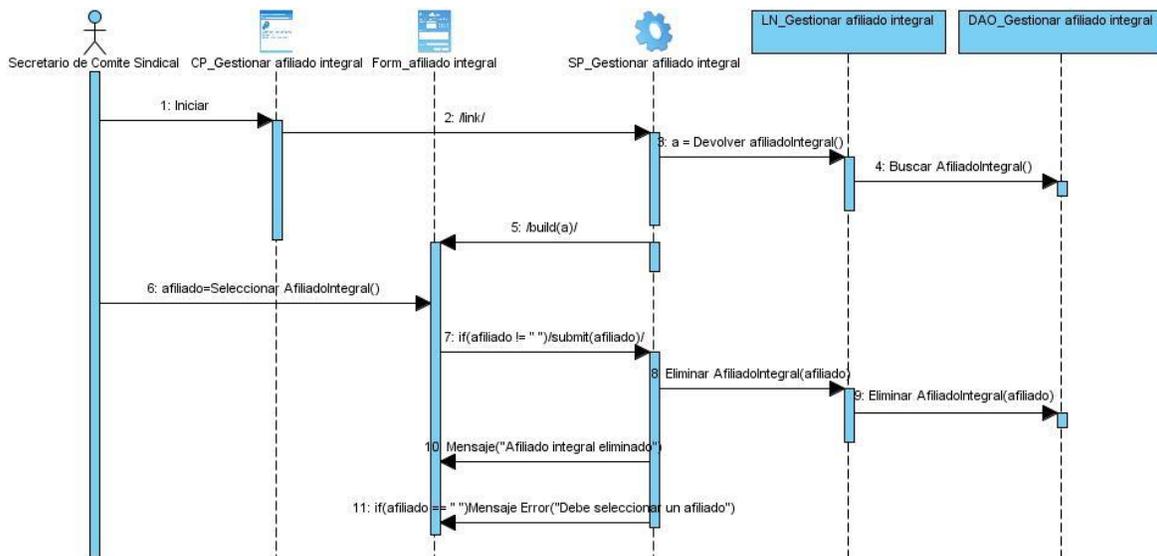


Fig.A5.13 DS Eliminar afiliado más integral.

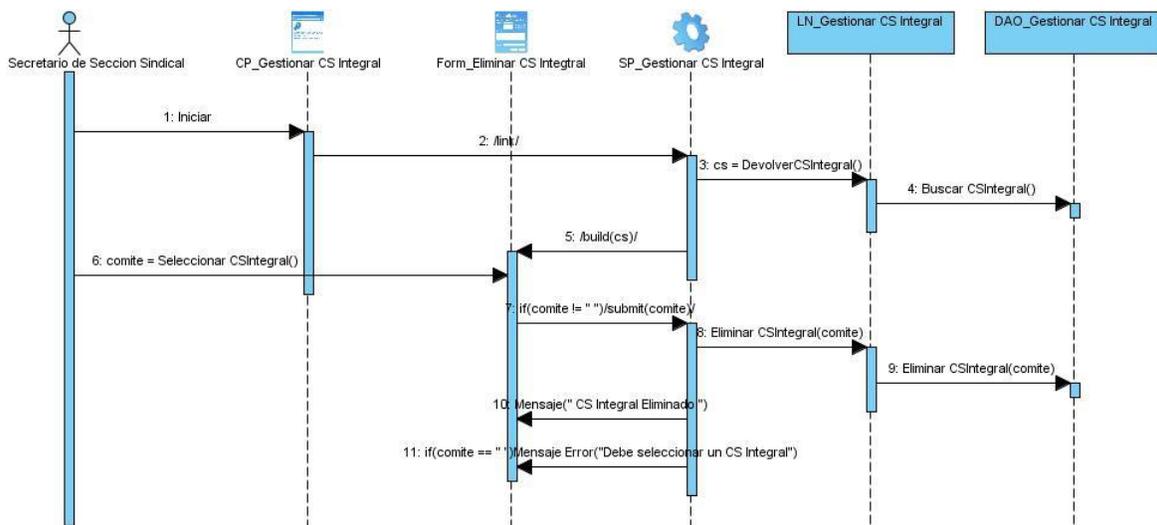


Fig.A5.14 DS Eliminar c/s más integral.

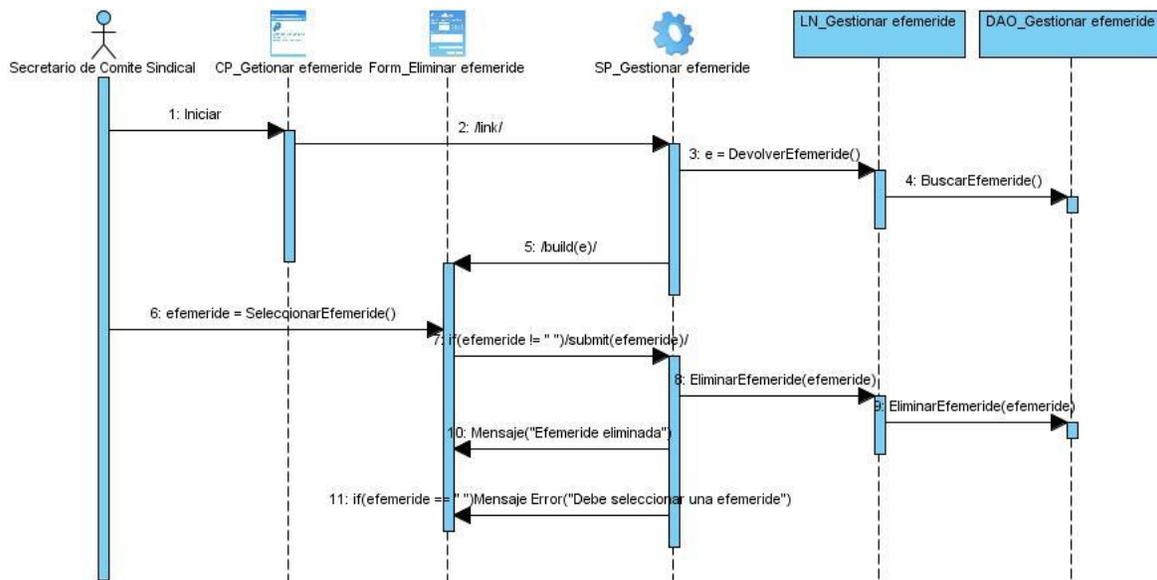


Fig.A5.15 DS Eliminar efemérides.

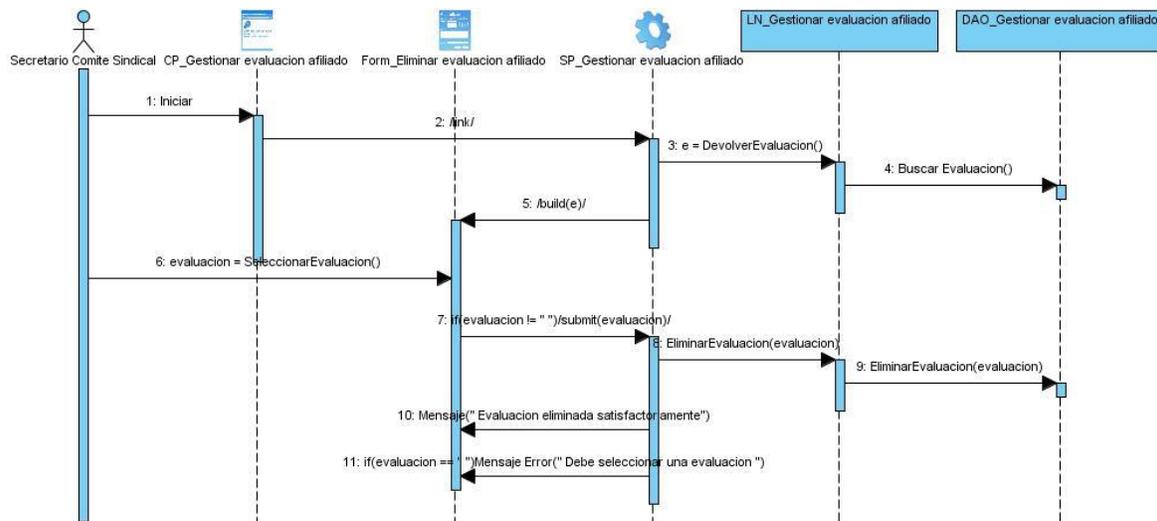


Fig.A5.16 DS Eliminar evaluación de afiliado.

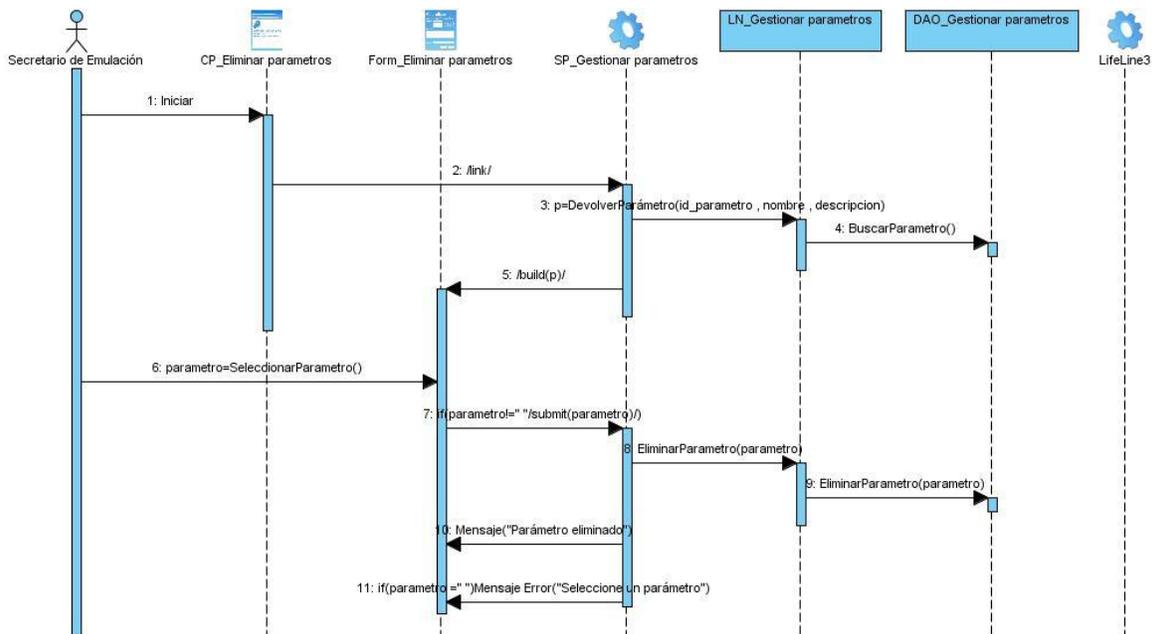


Fig.A5.17 DS Eliminar parámetros.

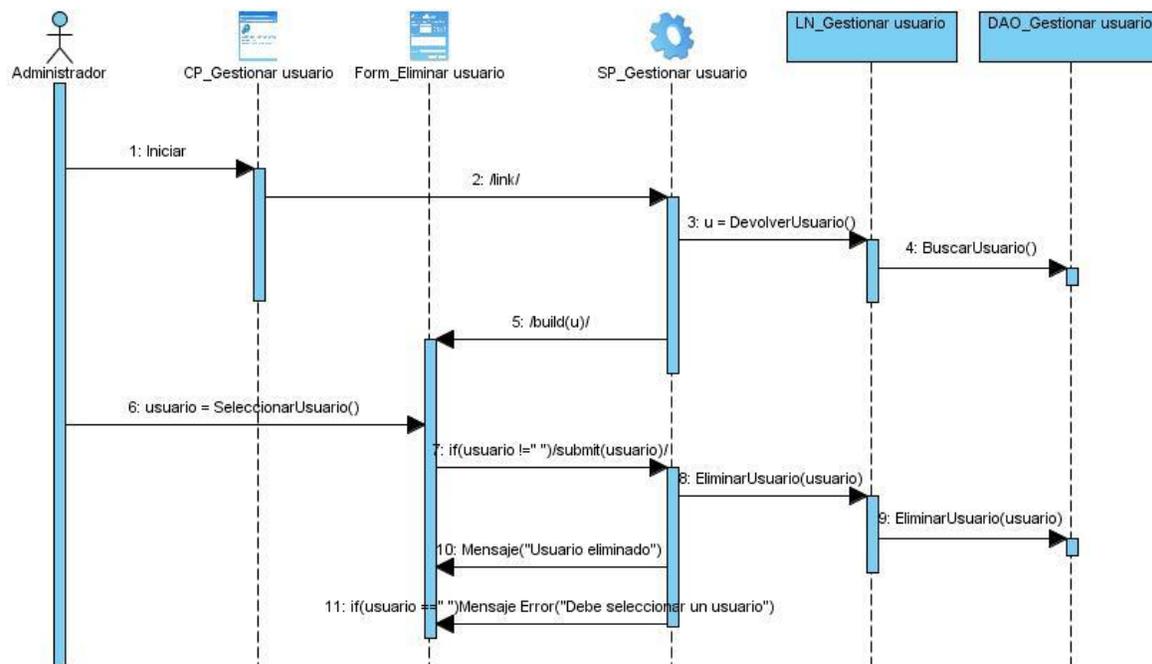


Fig.A5.18 DS Eliminar usuario.

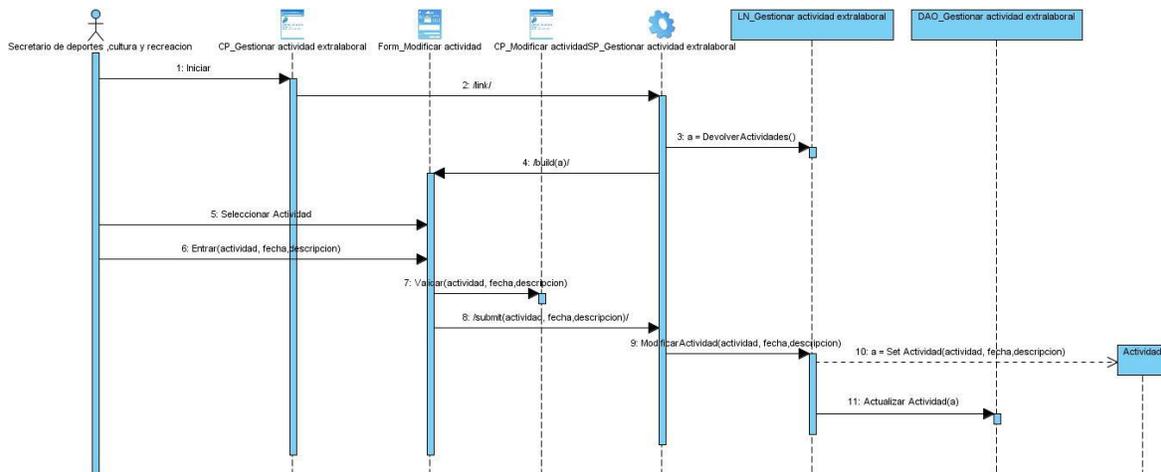


Fig.A5.19 DS Modificar actividad extra laboral.

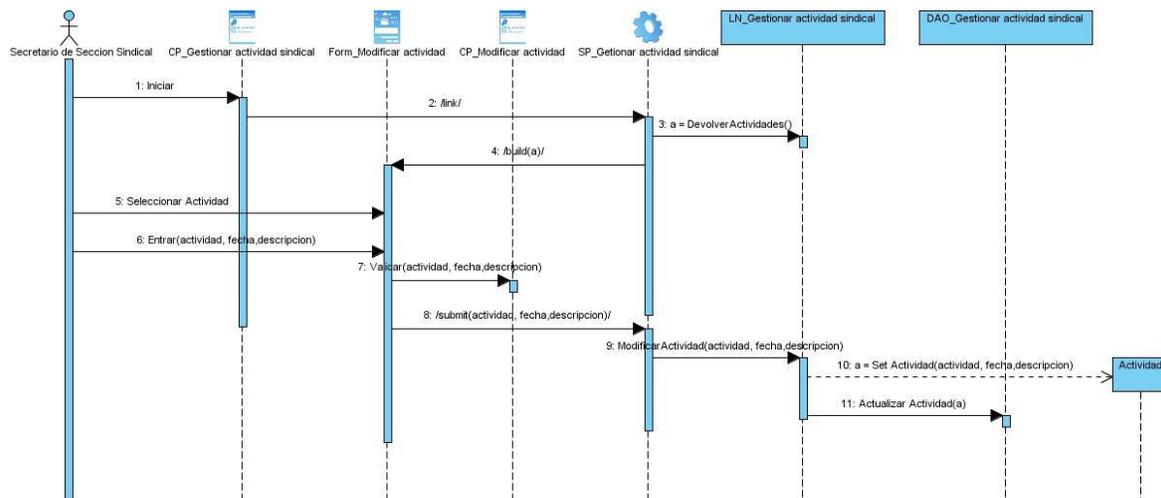


Fig.A5.20 DS Modificar actividad sindical.

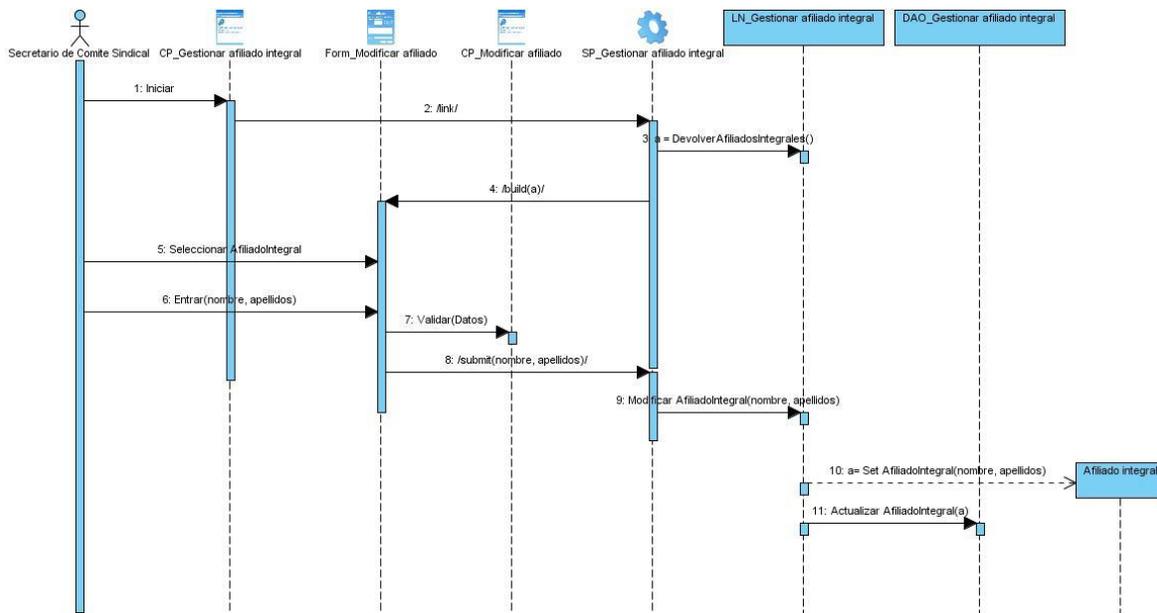


Fig.A5.21 DS Modificar afiliado más integral.

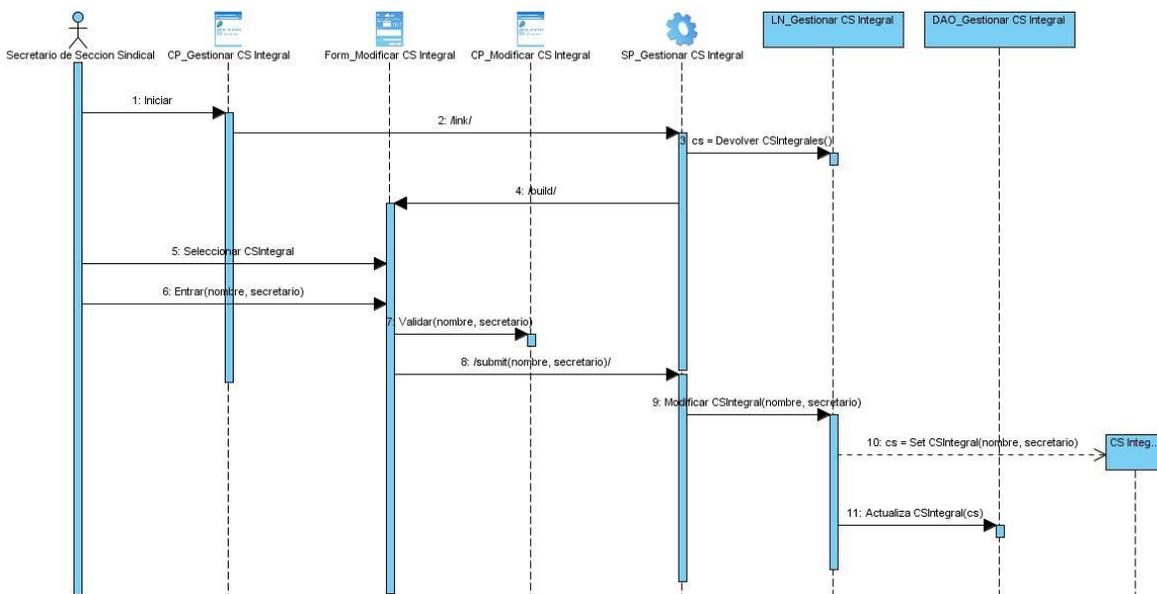


Fig.A5.22 DS Modificar c/s más integral.

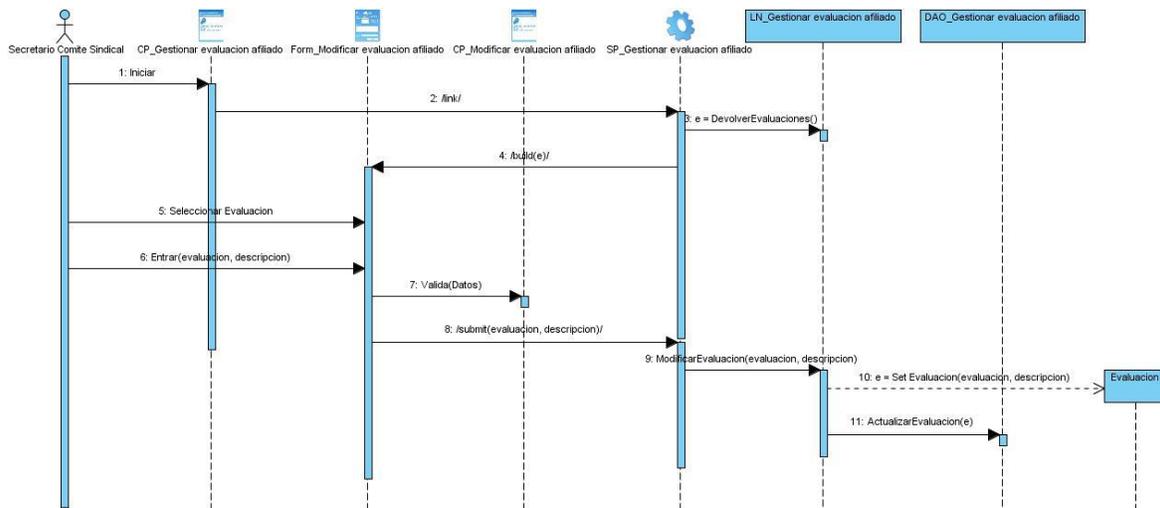


Fig.A5.23 DS Modificar evaluación de afiliado.

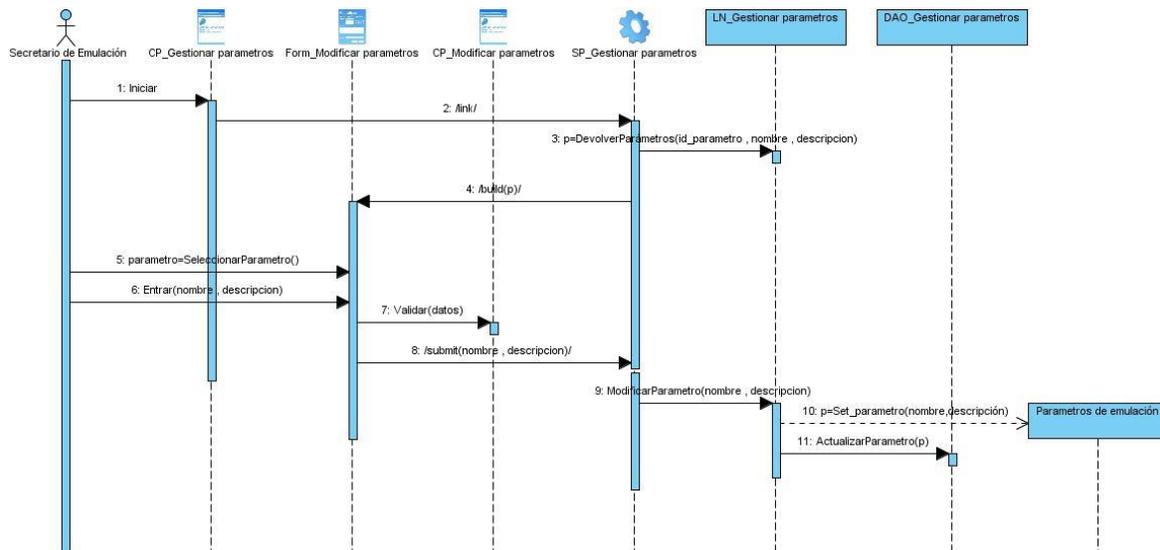


Fig.A5.24 DS Modificar parámetro.

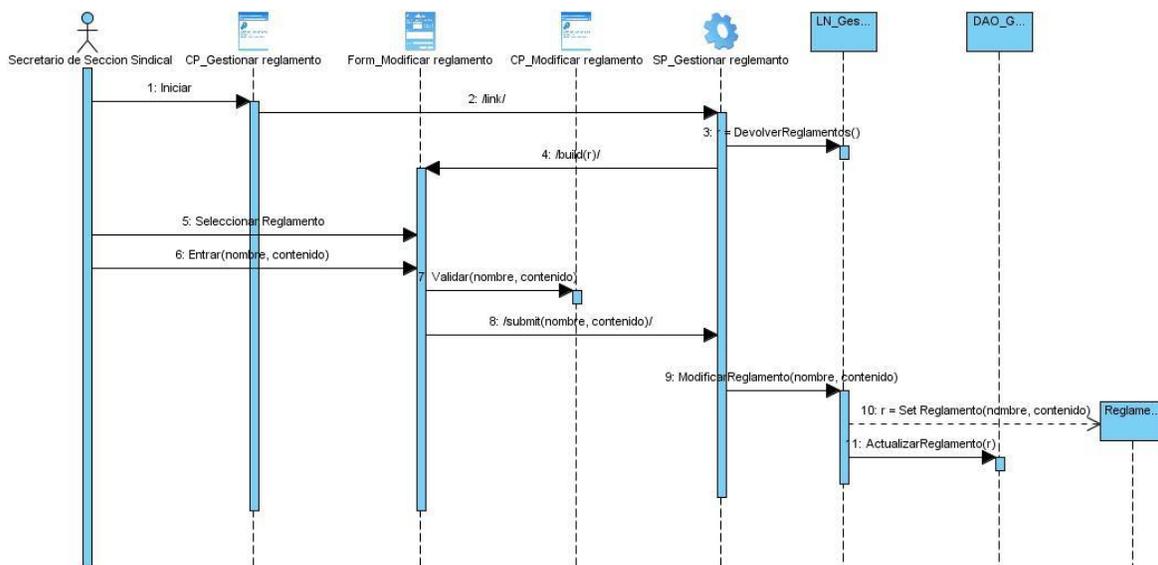


Fig.A5.25 DS Modificar reglamento.

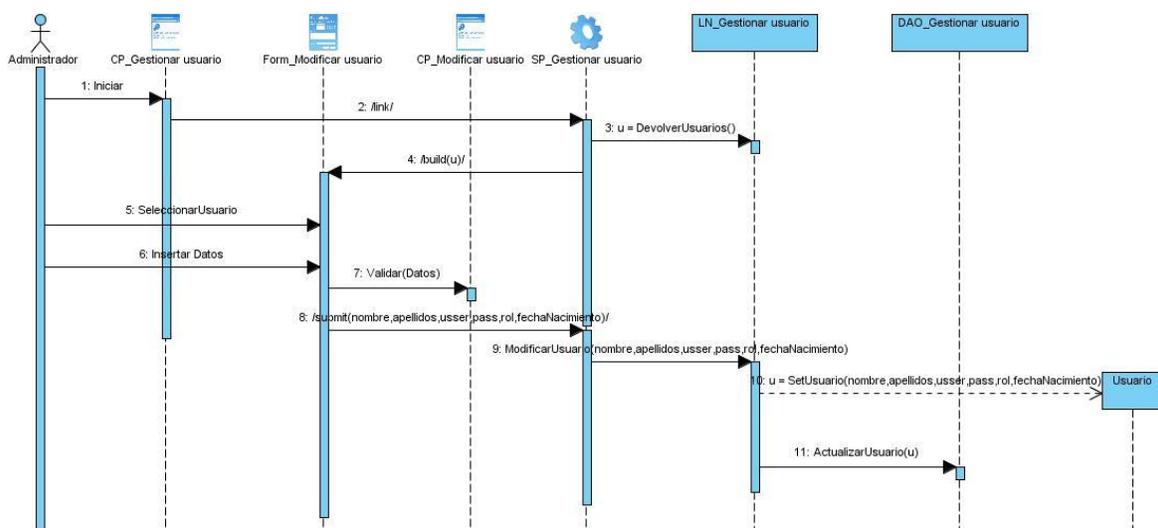


Fig.A5.26 DS Modificar usuario.

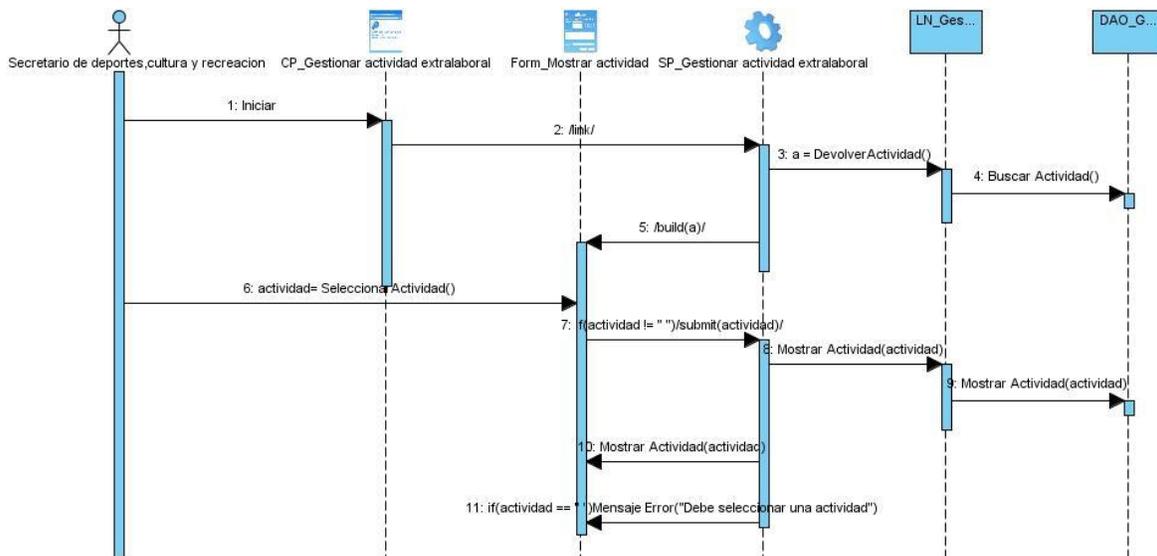


Fig.A5.27 DS Mostrar actividad extra laboral.

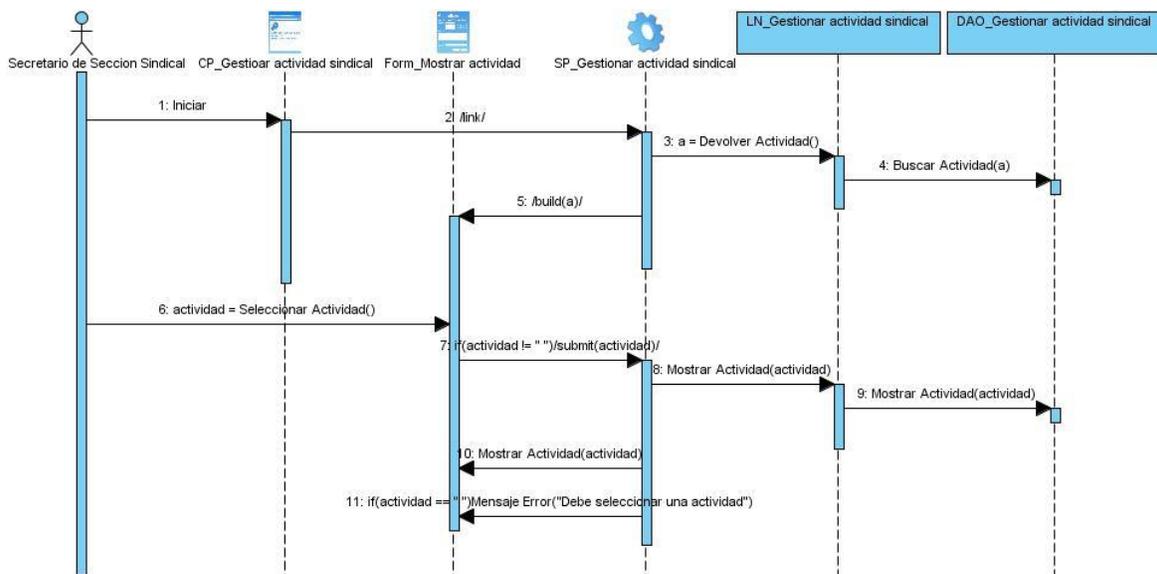


Fig.A5.28 DS Mostrar actividad sindical.

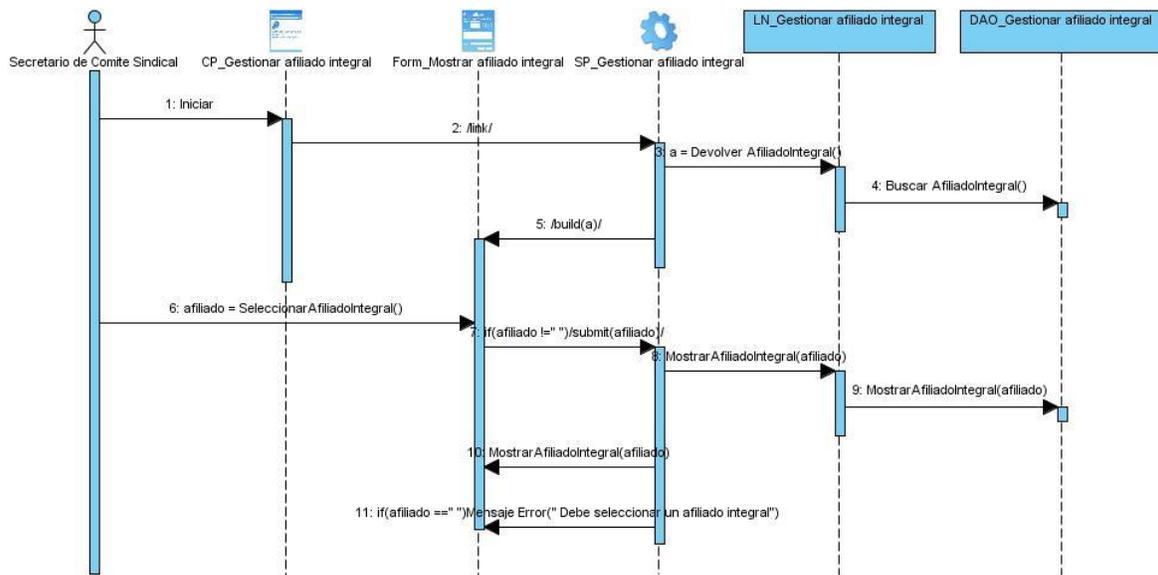


Fig.A5.29 DS Mostrar afiliado más integral.

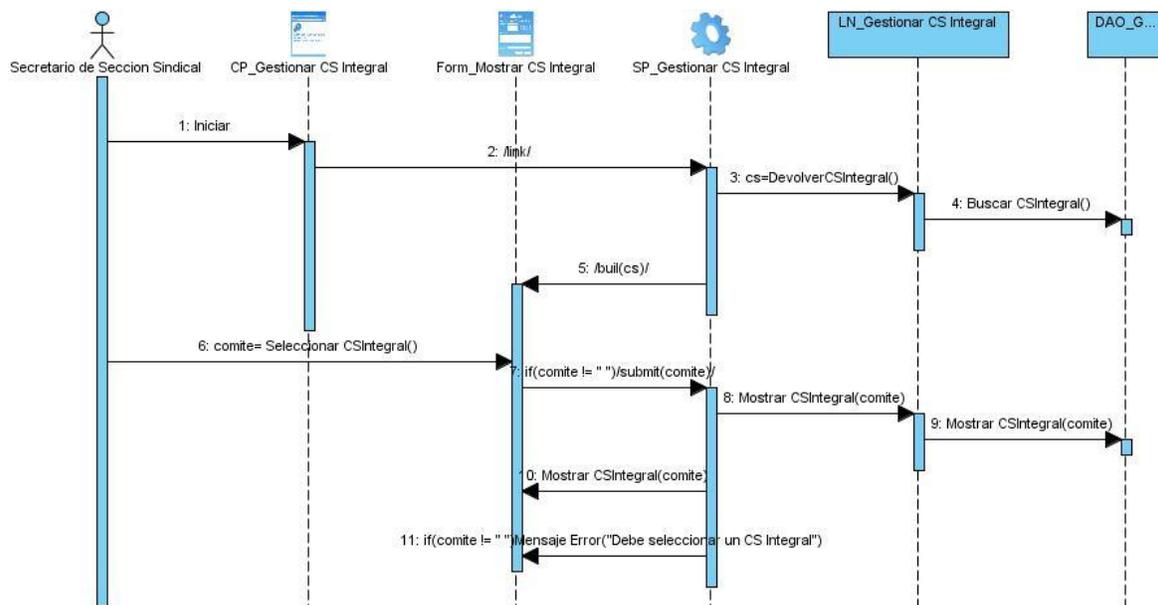


Fig.A5.30 DS Mostrar c/s más integral.

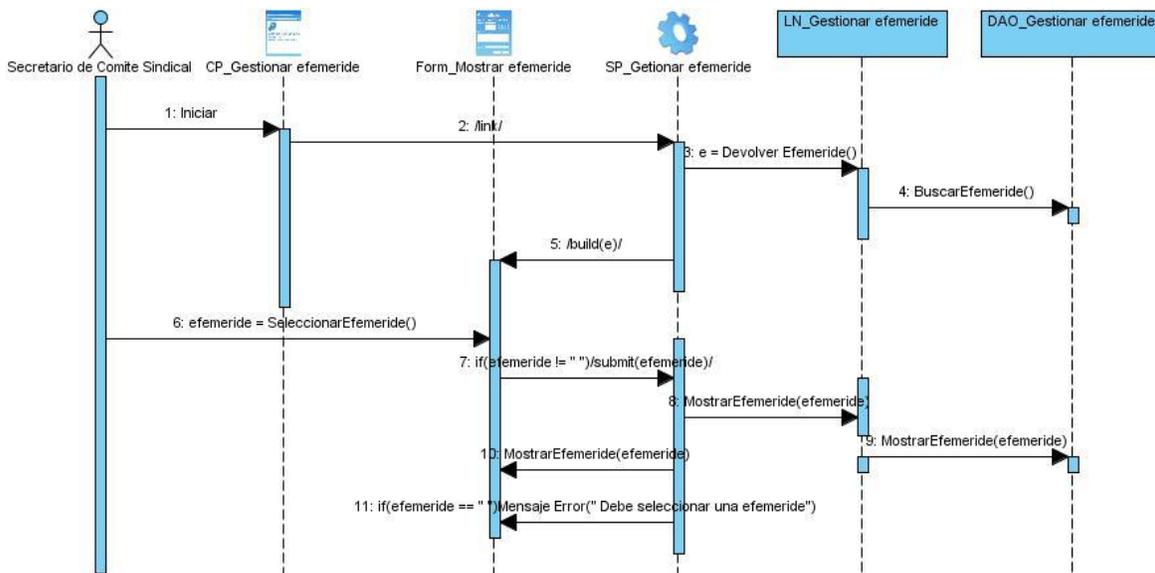


Fig.A5.31 DS Mostrar efemérides.

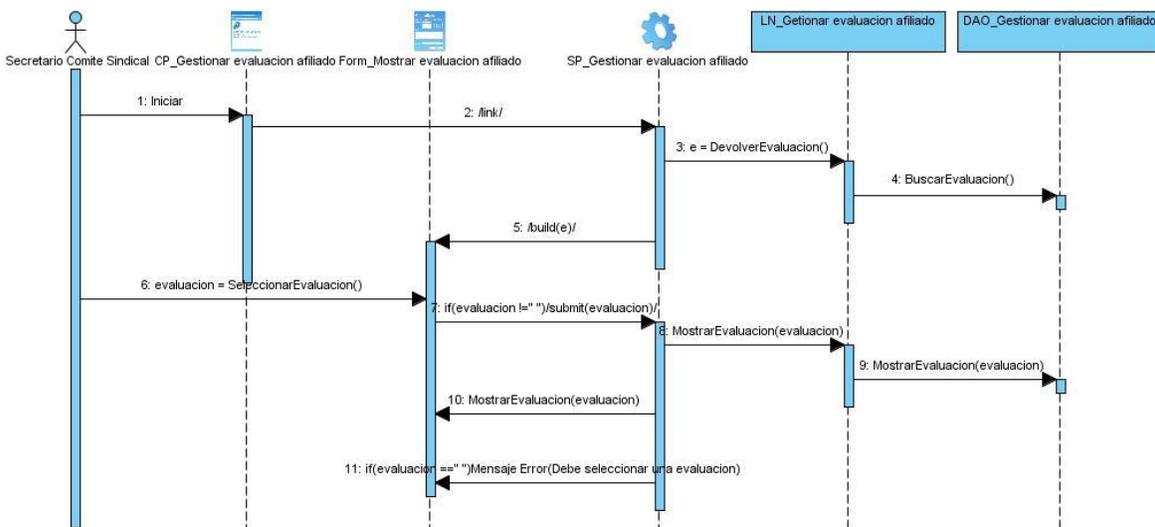


Fig.A5.32 DS Mostrar evaluación de afiliado.

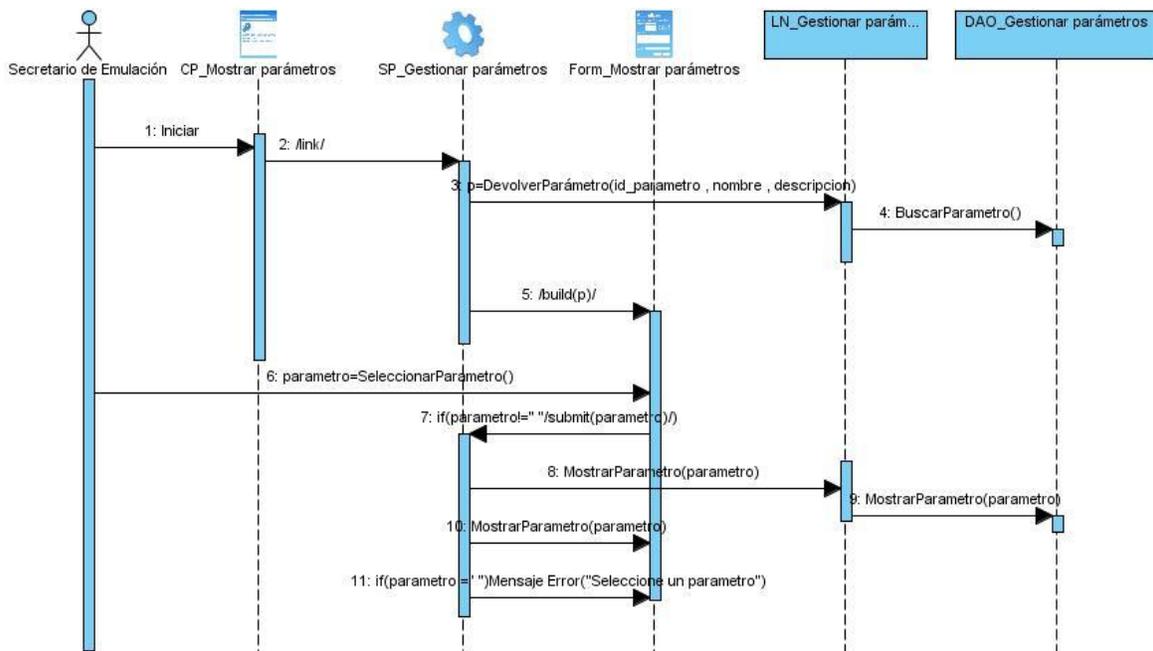


Fig.A5.33 DS Mostrar parámetros.

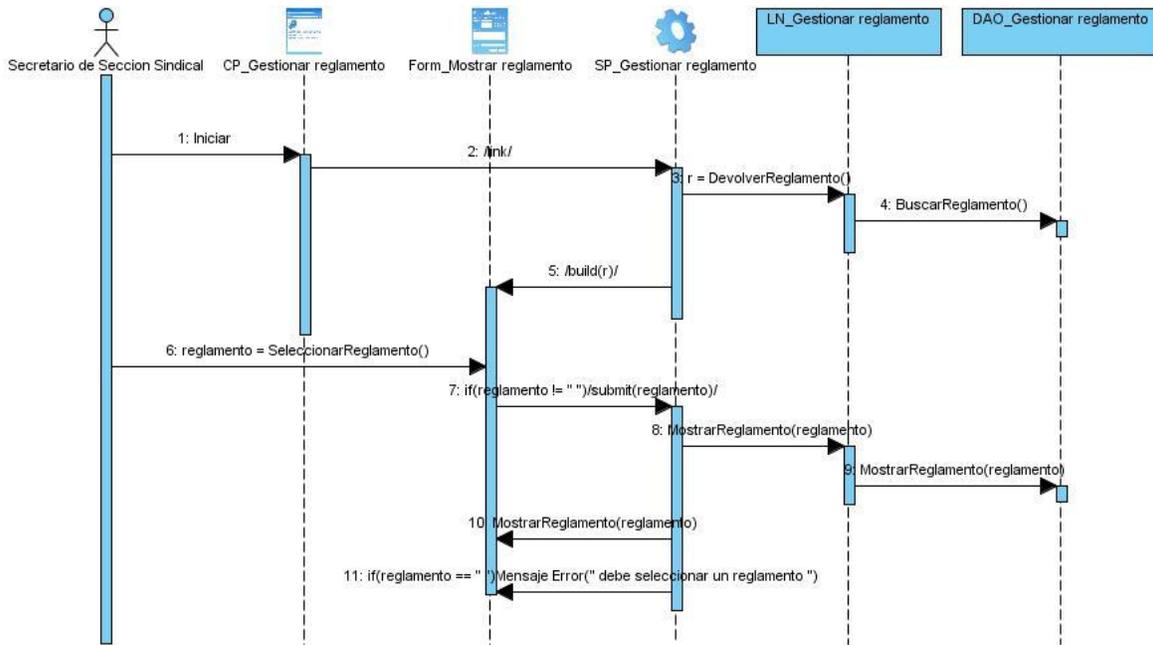


Fig.A5.34 DS Mostrar reglamento.

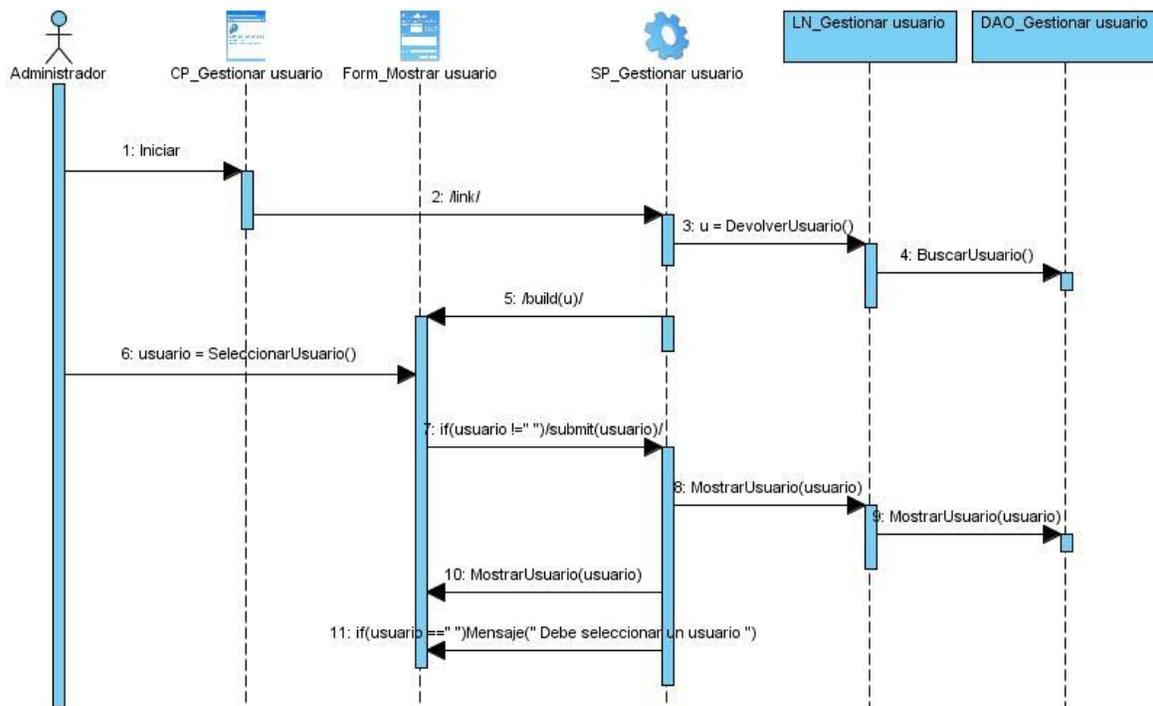


Fig.A5.35 DS Mostrar usuario.

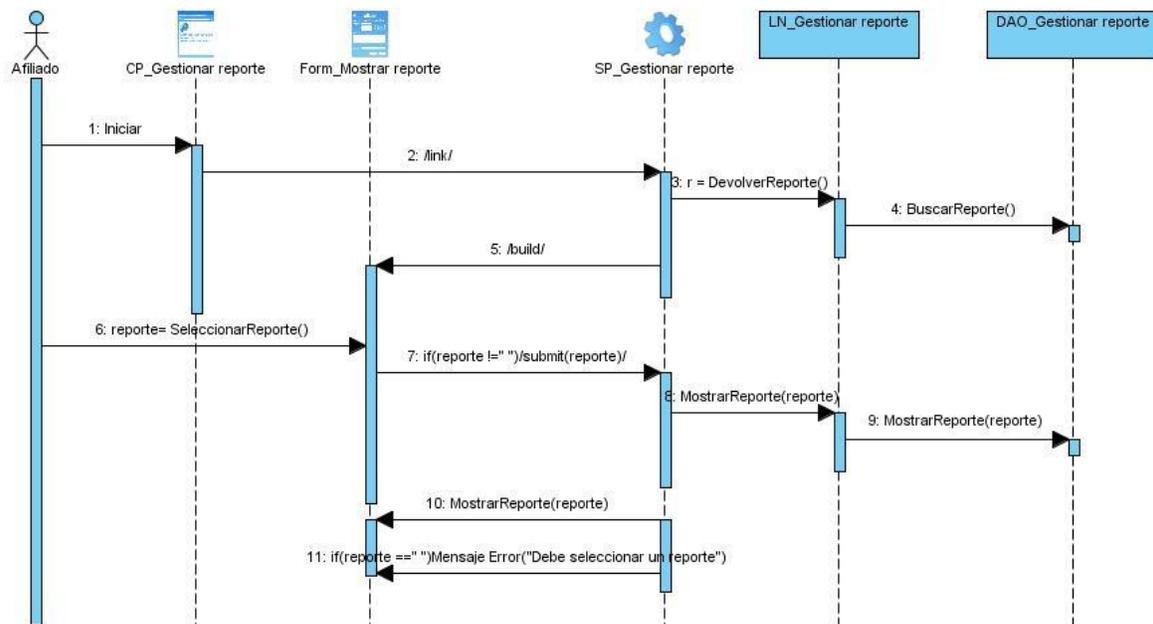


Fig.A5.36 DS Mostrar reporte.

Anexo # 6 Matriz de QDF.

Para mayor comprensión de la matriz QFD de enumeraron los casos de uso como sigue:

- CU#1 Autenticar usuario
- CU#2 Gestionar usuario
- CU#3 Gestionar noticia
- CU#4 Gestionar efemérides
- CU#5 Gestionar afiliado más integral
- CU#6 Gestionar evaluación de afiliado
- CU#7 Gestionar comité sindical más integral
- CU#8 Gestionar actividades sindicales
- CU#9 Gestionar reglamentos sindicales
- CU#10 Gestionar actividades extra laborales
- CU#11 Gestionar parámetros
- CU#12 Gestionar reportes
- CU#13 Buscar información
- CU#14 Registrar participación en actividades
- CU#15 Registrar participación en eventos
- CU#16 Registrar resultados

Figure A6.1 Esquema QFD

47							*									
46							*									
45							*									
44							*									
43														*		
42															*	
41													*			
40											*					
39		*														
38		*														
37		*														
36		*														
35	*															

34										*							
33										*							
32										*							
31										*							
30				*													
29				*													
28				*													
27				*													
26			*														
25			*														
24			*														
23			*														
22																*	
21																*	
20																*	
19								*									
18								*									
17								*									
16								*									
15				*													
14				*													
13				*													
12				*													
11					*												
10					*												
9					*												
8					*												
7																*	
6																*	
5																*	
4																*	
3									*								

2									*							
1									*							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

* Intercepción Requisitos –Casos de Uso

Anexo # 7 Prototipos de Interfaz de usuario.

The image shows a user interface form with a light green background. At the top, the title "ACTIVIDAD EXTRA LABORAL" is displayed in green. Below the title, there are several input fields: "Fecha:" followed by three dropdown menus for "Día", "Mes", and "Año"; "Hora:" followed by a text input field; "Lugar:" followed by a text input field; and "Nombre:" followed by a text input field. At the bottom of the form, there are two buttons: "Aceptar" and "Cancelar".

Fig A7. 1 PIU Actividad extra laboral.

GESTIONAR ACTIVIDADES EXTRA LABORALES

Fecha: Día: Mes: Año:

Hora: Lugar:

Nombre:

Descripción:

No.	Fecha	Hora:	Lugar:

Fig A7. 2 PIU Gestionar actividad extra laboral.

ACTIVIDAD SINDICAL

Fecha:

Día: Mes: Año:

Hora:

Lugar:

Nombre:

Fig A7. 3 PIU Actividad sindical.

GESTIONAR ACTIVIDAD SINDICAL

Nombre:

Fecha: Día: Mes: Año:

Hora:

Lugar:

Descripción:

No.	Fecha	Hora	Lugar

Fig A7. 4 PIU Gestionar actividad sindical.

AFILIADO MAS INTEGRAL

Usuario:

C/S

Descripción:

Fig A7. 5 PIU Afiliado más integral.

GESTIONAR AFILIADO MAS INTEGRAL

Usuario

Descripción:

No	Nombre	Apellido	C/S	Usuario

Fig A7. 6 PIU Gestionar Afiliado más integral.

C/S MAS INTEGRAL

C/S

Descripción

Fig A7. 7 PIU C/S más integral.

GESTIONAR C/S MAS INTEGRAL

C/S

Descripción

Fig A7. 8 PIU Gestionar C/S más integral.

EFEMERIDE

Nombre:

Fecha: Día: Mes: Año

Descripción:

Fig A7. 9 PIU Efeméride.

GESTIONAR EFEMERIDES

Fecha: Día Mes Año

Nombre:

Contenido:

No.	Nombre	Clasificación	Descripción	Fecha

Fig A7. 10 PIU Gestionar Efemérides.

PARAMETROS

parámetro:

Descripción:

Fig A7. 11 PIU Parámetros.

GESTIONAR PARAMETROS DE EMULACION

Tipo de emulación:

Tipo de parámetro

Indicadores de emulación:

No.	Indicador	Ptos por indicador	Acumulado de Ptos.	Cumplidor

Fig A7. 12 PIU Gestionar Parámetros.

GESTIONAR REGLAMENTOS

Tipo de reglamento

Descripción:
enter text...

Fig A7. 13 PIU Gestionar reglamentos.

REGLAMENTO

Tipo de reglamento

Nombre: enter text...

Descripción:
enter text...

Fig A7. 14 PIU Reglamentos.

GESTIONAR REPORTE

Buscar Imprimir Exportar

Descripción de reporte
enter text...

Mostrar Cancelar

Fig A7. 15 PIU Gestionar reporte.

REPORTE

Nombre enter text...

Descripción
enter text...

Aceptar Cancelar

Fig A7. 16 PIU Reporte.

GESTIONAR USUARIO

Usuario:

Descripción:

No	Usuario	Nombre	Apellidos	Rol	Contraseña

Fig A7. 17 PIU Gestionar usuario.

USUARIO

Usuario:

Contraseña:

Nombre:

Apellido:

Rol:

Fig A7. 18 PIU Usuario.

REGISTRAR PARTICIPACION EN ACTIVIDADES

Actividad

Fecha Día Mes Año

Usuario

Descripción

No.	Usuario	Actividad	Fecha

Fig A7. 19 PIU Registrar participación en actividades.

REGISTRAR PARTICIPACION EN EVENTOS

Evento **Obtuvo resultados?**

Fecha Día Mes Año

Usuario

Descripción

No.	Usuario	Evento	Fecha

Fig A7. 20 PIU Registrar participación en eventos.

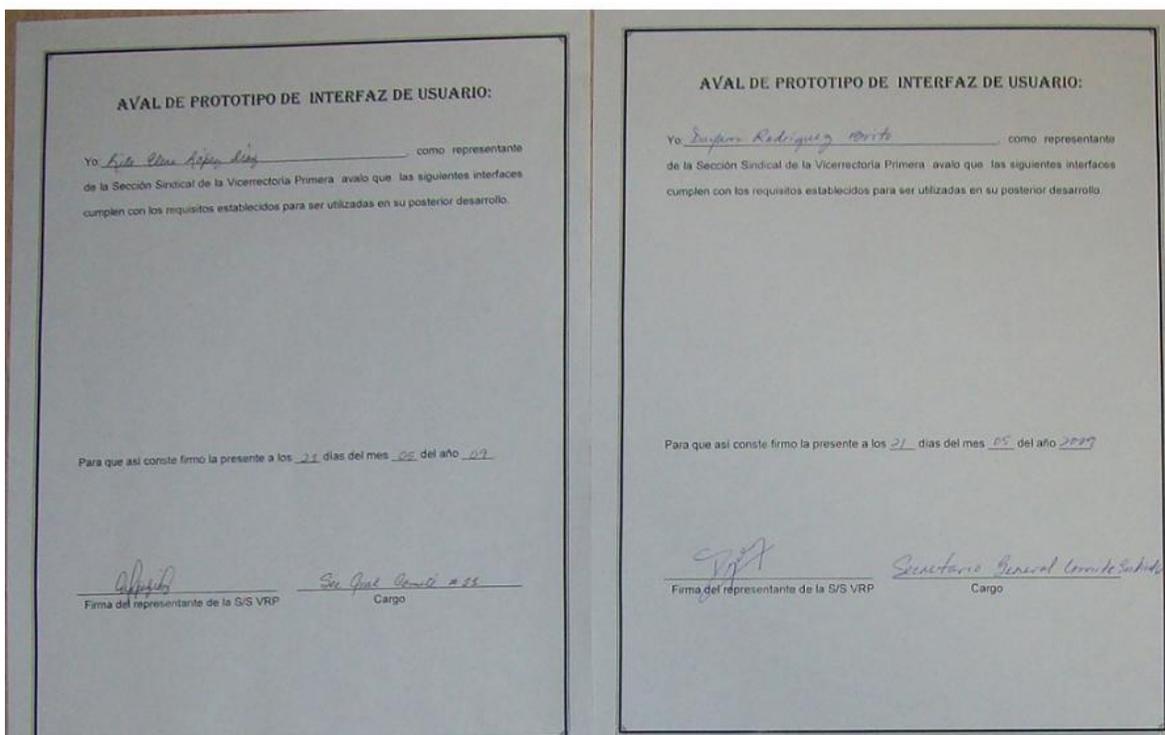
Anexo # 8 Avals de Prototipos.

AVAL DE PROTOTIPO DE INTERFAZ DE USUARIO:

Yo, Aleandra Olivares Hernández como representante de la Sección Sindical de la Vicerrectoría Primera avalo que las siguientes interfaces cumplen con los requisitos establecidos para ser utilizadas en su posterior desarrollo.

Para que así conste firmo la presente a los 21 días del mes 5 del año 2007.

[Firma] Firma del representante de la S/S VRP Sec. Gral. Comité # 3
Cargo



AVAL DE PROTOTIPO DE INTERFAZ DE USUARIO:

Yo: Guillermo de Cero Velasco como representante de la Sección Sindical de la Vicerrectoría Primera avalo que las siguientes interfaces cumplen con los requisitos establecidos para ser utilizadas en su posterior desarrollo.

Para que así conste firmo la presente a los 21 días del mes 15 del año 2009

[Firma] S.S.S
Firma del representante de la S/S VRP Cargo

<p>AVAL DE PROTOTIPO DE INTERFAZ DE USUARIO:</p> <p>Yo <u>Juan Carlos Lopez Gonzalez</u> como representante de la Sección Sindical de la Vicerrectoría Primera avalo que las siguientes interfaces cumplen con los requisitos establecidos para ser utilizadas en su posterior desarrollo.</p> <p>Para que así conste firmo la presente a los <u>21</u> días del mes <u>05</u> del año <u>2009</u>.</p> <p><u>Juan Carlos Lopez Gonzalez</u> Firma del representante de la S/S VRP</p> <p><u>Secretaria General Sección Sindical</u> Cargo</p>	<p>AVAL DE PROTOTIPO DE INTERFAZ DE USUARIO:</p> <p>Yo <u>Michael Gonzalez Jarama</u> como representante de la Sección Sindical de la Vicerrectoría Primera avalo que las siguientes interfaces cumplen con los requisitos establecidos para ser utilizadas en su posterior desarrollo.</p> <p>Para que así conste firmo la presente a los <u>21</u> días del mes <u>5</u> del año <u>2009</u>.</p> <p><u>Michael Gonzalez Jarama</u> Firma del representante de la S/S VRP</p> <p><u>Secretaria General Sección Sindical</u> Cargo</p>
---	--

Glosario de términos:

Actividad: Unidad tangible de trabajo realizada por un trabajador en un flujo de trabajo.

Actor: Alguien o algo, fuera del sistema o negocio que interactúa con el sistema o negocio.

Base de datos: Conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo.

Clases de control: Representan coordinación, secuencia, transacciones, y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto.

Clases de entidad: Se utilizan para modelar información que posee una vida larga y que es a menudo persistente.

Clases de interfaz: Se utilizan para modelar la interacción entre el sistema y sus actores (es decir, usuarios y sistemas externos).

Cliente: Persona, organización o grupo de personas que encarga la construcción de un sistema, ya sea empezando desde cero, o mediante el refinamiento de versiones sucesivas.

Flujo de trabajo: Realización de un caso de uso de negocio o parte de él. Puede describirse en términos de diagramas de actividad que incluyen a los trabajadores participantes, las actividades que realizan y los artefactos que producen.

Prototipo de interfaz de usuario: Fundamentalmente, un prototipo ejecutable de una interfaz de usuario, pero que puede, en los momentos iniciales del desarrollo, consistir únicamente en dibujos en papel, diseños de plantillas, etc.

Rol: Papel, cometido o función que tiene o desempeña que interpreta un actor.

Sección Sindical: Está integrada por todos los trabajadores, jubilados y pensionados afiliados al Sindicato en un centro de trabajo, departamento, sección, turno y brigada.

Trabajador: Puesto que puede ser asignado a una persona o equipo, y que requiere responsabilidades y habilidades como realizar determinadas actividades o desarrollar determinados artefactos.

Usuario: Humano que interactúa con un sistema.