

**Universidad de las Ciencias Informáticas**



**Facultad 8**

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

**Título:** Análisis y diseño del componente Gestión de Indicadores para el Cuadro de Mando Integral del Paquete de Apoyo a la Toma de Decisiones y Sistemas Inteligentes.

**Autores:**

Aramis Aguilar Jiménez  
Giraldo Yariel Aguiar García

**Tutores:**

Ing. Frank González Fernández  
Ing. Héctor Luis Reyes Zaldívar

*Ciudad de La Habana, junio de 2010*

*Año 52 de la Revolución*



“La inteligencia no es la facultad de imponerse, es el deber de ser útil a los demás”

José Martí

## DECLARACIÓN DE AUTORÍA

*Declaramos que somos los únicos autores del trabajo titulado: Análisis y diseño del componente Gestión de Indicadores para el Cuadro de Mando Integral del Paquete de Apoyo a la Toma de Decisiones y Sistemas Inteligentes (PATDSI) y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.*

*Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2010.*

Aramis Aguilar Jiménez

Giraldo Yariel Aguiar García

---

Firma del Autor

---

Firma del Autor

Frank González Fernández

Héctor Luis Reyes Zaldívar

---

Firma del Tutor

---

Firma del Tutor

*Quiero dedicarles este trabajo a todas esas personas que confiaron y creyeron en mí desde el comienzo, principalmente a mis padres Cruz Abilio Aguilar Rondón y Alina Marina Jiménez Rodríguez, por el cariño y el amor que me han brindado toda la vida, por servirme de inspiración en todo plan que me he trazado y por estar a mi lado guiándome por todos los caminos de la vida, brindándome su apoyo en todo momento. A ustedes les digo que un hijo no puede esperar mejores padres, por eso, me esforcé todo lo que pude y ya casi tengo un gran sueño realizado, el tener un título universitario, en una carrera que me gusta y sé además, que este es un sueño de ustedes también. Los quiero. Besos.*

*A mis hermanos, abuelos, tíos y primos, a mis amigos en especial Alián y Amet, a todos ustedes gracias por siempre apoyarme y confiar en mí.*

*A mi novia Yeni por estar a mi lado estos 5 años de la carrera brindándome su amor y comprensión.*

*Aramis*

*Dedico este trabajo a esa persona que me brindó su amor incondicional y me dio la fuerza que necesitaba para terminar mis estudios, a ti abuela por estar siempre conmigo.*

*A mi madre María Victoria García Lorenzo, no te comparo con nada en este mundo, sin ser perfecta me has guiado por el buen camino.*

*A mi padre Giraldo Aguiar Corzo, por ser mi punto de apoyo, mi amigo, con nada puedo pagarte el sacrificio que has hecho por mí.*

*A mi hermana Yailén Aguiar García por tu apoyo, preocupación y cariño.*

*A mis tíos Pupy y Loika por haberse comportado como mis padres todo este tiempo.*

*Yariel*

*Agradezco a:*

*La Revolución Cubana por darme la posibilidad de superarme e integrar las filas de esta tropa del futuro, en especial al comandante en Jefe Fidel Castro Ruz.*

*A mis padres que son mi razón de ser y me han dado día a día su apoyo y comprensión para que este sueño se hiciera realidad.*

*A todos mis amigos por ayudarme durante toda la carrera y brindarme su amistad.*

*A mis profesores de la UCI que con su esfuerzo me han formado como ingeniero.*

*A mis tutores por ayudarnos y estar pendientes de nuestro trabajo diario.*

*Aramis*

*Agradezco a:*

*Nuestro Comandante en Jefe Fidel por haber hecho realidad este sueño y darnos la oportunidad de estudiar en una universidad como esta.*

*A mis padres, mis hermanos, mis abuelos, mis tíos y a toda mi familia por haberme dado su apoyo y por todo lo que han hecho por mí.*

*A mis amigos por estar conmigo en las buenas y las malas, por su ayuda y sinceridad.*

*A mis Tutores por los consejos y el apoyo ofrecido durante la realización de este trabajo.*

*A mis profesores por la enseñanza y los valores que me inculcaron.*

*A todas las personas que conozco.*

*Muchas Gracias*

*Yariel*

## RESUMEN

Las empresas ante la necesidad de tener un mejor control sobre todos los procesos que se llevan a cabo dentro de las mismas, se han visto en la necesidad de gestionar el control de forma tal que vincule no solo los aspectos financieros, sino que abarque un concepto más amplio en cuanto a las actividades que le permitirían medir su desempeño y poder cumplir con la estrategia trazada. En el presente trabajo se realiza un estudio relacionado con los indicadores y su utilidad dentro de las perspectivas del Cuadro de Mando Integral como método más eficaz para controlar los procesos en la empresa.

El objetivo del trabajo es realizar el análisis y diseño de un componente de gestión de indicadores, que mejore el análisis organizacional para la toma de decisiones, que sea desarrollado sobre tecnologías libres y que se ajuste a las necesidades de las empresas cubanas.

### **PALABRAS CLAVES:**

Indicadores, gestión de indicadores, cuadro de mando integral, análisis, diseño.

## Índice de Contenido

<b>INTRODUCCIÓN</b> .....	<b>- 1 -</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN</b> .....	<b>5</b>
1. Bases conceptuales .....	5
1.1. Cuadro de Mando Integral .....	5
1.1.1. Dashboard.....	5
1.1.1.1. Key Performance Indicators .....	6
1.2. Estudio de Indicadores .....	7
1.2.1. Técnicas para elaborar indicadores .....	7
1.2.2. Características de los indicadores.....	8
1.2.3. Tipos de indicadores .....	8
1.3. Fuentes de captura de indicadores.....	11
1.4. Beneficios y riesgos de los indicadores en la implantación de un CMI .....	11
1.5. Gestión de indicadores. Contexto Actual .....	12
1.5.1 Gestión de indicadores. Tendencias mundiales .....	13
1.5.1.1 Software para la gestión de indicadores utilizando Cuadro de Mando Integral .....	13
1.5.2 Gestión de indicadores. Caso Cuba .....	16
1.6. Metodologías de desarrollo de software .....	16
1.7. Lenguaje de modelado .....	19
1.8. Herramientas CASE para el desarrollo de software .....	19
1.9. Sistema Gestor de Bases de Datos.....	21
1.10. Captura de requerimientos .....	23
1.10.1. Introspección.....	23
1.10.2. Discusiones.....	24
1.10.3. Prototipos.....	24
1.11. Validación de los requerimientos y diseño .....	24
Conclusiones del Capítulo .....	25
<b>CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL COMPONENTE PARA LA GESTIÓN DE INDICADORES.....</b>	<b>34</b>
2. Propuesta de Solución .....	34
2.1. Modelo del Dominio.....	34
2.1.1. Diagrama conceptual del modelo de dominio.....	34
2.2. Especificación de los Requisitos de Software .....	36
2.2.1. Requisitos funcionales .....	37
2.2.2. Requisitos no funcionales .....	39
2.3. Definición de los Casos de Uso .....	41
2.3.1. Definición de los actores .....	41

2.3.2. Diagrama de casos de uso del sistema.....	42
2.3.3. Descripción de los casos de uso del sistema .....	43
2.3.3.1. Descripción del CU Gestionar Indicador .....	43
2.4. Modelo de Análisis .....	45
2.4.1. Diagrama de clases del Análisis.....	45
2.4.1.1. Clases del Análisis .....	46
2.4.1.2. Diagrama de clases del análisis: Gestionar Indicador.....	47
2.4.2. Diagramas de Colaboración.....	47
2.4.2.1. Diagrama de colaboración: Caso de Uso Gestionar Indicador.....	48
2.5. Diseño del Sistema.....	49
2.5.1. Patrones utilizados en el desarrollo del sistema.....	49
2.5.1.1. Patrones de Asignación de Responsabilidades .....	50
2.5.1.2. Patrones del diseño.....	50
2.5.2. Diagrama de clases del diseño .....	51
2.5.2.1. Diagrama de Clases del Diseño: Gestionar Indicador.....	52
2.5.3. Diagramas de Secuencia .....	52
2.5.3.1. Diagramas de Secuencia del CU Gestionar Indicador .....	53
2.5.4. Diagrama de Clases Persistentes .....	54
2.5.5. Modelo de Datos.....	55
Conclusiones del Capítulo.....	56
<b>CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO.....</b>	<b>57</b>
3.1. Validación de requisitos.....	57
3.1.1. Métricas de la Calidad de Especificación .....	57
3.1.2. Matriz de trazabilidad .....	60
3.1.3. Lista de Chequeo para validar el listado de requisitos.....	62
3.2. Pruebas.....	63
3.2.1. Pruebas de sistemas.....	63
3.2.1.1. Caso de Prueba: Caso de uso Gestionar Indicadores .....	63
3.3. Métricas para la validación del diseño .....	66
Conclusiones del Capítulo.....	72
<b>CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD Y ANÁLISIS DE COSTOS.....</b>	<b>73</b>
4.1. Planificación basada en casos de uso .....	73
4.1.1. Identificar los puntos de casos de usos desajustados .....	73
4.1.2. Cálculo del factor de peso de los actores sin ajustar .....	73
4.1.3. Cálculo del factor de peso de los casos de uso sin ajustar.....	74
4.2. Ajustar los puntos de casos de uso .....	74
4.2.1. Cálculo del factor de complejidad técnica .....	75

4.2.2. Cálculo del factor de ambiente .....	76
4.3. Calcular esfuerzo de implementación .....	77
4.3.1. Cálculo del factor de conversión .....	77
4.4. Esfuerzo de todo el proyecto .....	78
4.5. Beneficios tangibles e intangibles .....	79
4.6. Análisis del costo.....	79
4.6.1. Costo del Proyecto.....	80
Conclusiones del Capítulo .....	81
<b>CONCLUSIONES GENERALES .....</b>	<b>82</b>
<b>RECOMENDACIONES:.....</b>	<b>83</b>
<b>BIBLIOGRAFÍA.....</b>	<b>84</b>
<b>ANEXOS .....</b>	<b>89</b>
Anexo 1: Descripción textual de los Casos de Uso del negocio.....	89
Anexo 2: Diagrama de clases del análisis .....	98
Anexo 3: Diagramas de colaboración .....	99
Anexo 4: Diagramas de Clases del Diseño.....	103
Anexo 5: Diagramas de Secuencia.....	106
Anexo 6: Descripción de las tablas de la Base de Datos .....	109
Anexo 7: Descripción de las clases de análisis.....	113
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>116</b>

## Índice de Figuras

Figura 1.- Representación de un CMI según Norton y Kaplan.....	6
Figura 2.- Consola del Software Delphos.....	14
Figura 3.- Modelo de Dominio.....	35
Figura 4.- Diagrama de Caso de Uso del Sistema.....	43
Figura 5.- Diagrama de clases del análisis: Gestionar Indicador.....	47
Figura 6.- Diagrama de colaboración: Adicionar indicador.....	48
Figura 7.- Diagrama de colaboración: Modificar indicador.....	48
Figura 8.- Diagrama de colaboración: Eliminar indicador.....	49
Figura 9.- Diagrama de Clases del Diseño: Gestionar Indicador.....	52
Figura 10.- Diagramas de Secuencia: Adicionar Indicador.....	53
Figura 11.- Diagramas de Secuencia: Modificar Indicador.....	53
Figura 12.- Diagramas de Secuencia: Eliminar Indicador.....	54
Figura 13.- Diagrama de Clases Persistentes.....	55
Figura 14.- Diagrama de Modelo de Datos.....	56
Figura 15.- Representación de la responsabilidad y complejidad.....	68
Figura 16.- Representación de la reutilización.....	69
Figura 17.- Representación del acoplamiento.....	71
Figura 18.- Representación de la reutilización.....	71
Figura 19.- Diagrama de clases del análisis: Capturar Datos.....	98
Figura 20.- Diagrama de clases del análisis: Configurar Captura de Datos.....	98
Figura 21.- Diagrama de clases del análisis: Asociar a objetivos.....	99
Figura 22.- Diagrama de colaboración: Capturar Datos (Fórmula).....	99

Figura 23.- Diagrama de colaboración: Capturar Datos (Archivo) .....	100
Figura 24.- Diagrama de Colaboración: Capturar Datos (Base de Datos) .....	100
Figura 25.- Diagrama de colaboración: Configurar Captura de Datos (Conexión BD) .....	101
Figura 26.- Diagrama de colaboración: Configurar Captura de Datos (Crear fórmula) .....	101
Figura 27.- Diagrama de colaboración: Configurar Captura de Datos (Carga de archivos) .....	102
Figura 28.- Diagrama de colaboración: Asociar a objetivos.....	102
Figura 29.- Diagrama de Clases del Diseño: Asociar a objetivos .....	103
Figura 30.- Diagrama de Clases del Diseño: Capturar Datos .....	104
Figura 31.- Diagrama de Clases del Diseño: Configurar Captura de Datos.....	105
Figura 32.- Diagramas de Secuencia Capturar Datos (Archivo) .....	106
Figura 33.- Diagramas de Secuencia Capturar Datos (Base de Datos).....	106
Figura 34.- Diagramas de Secuencia Capturar Datos (Fórmula).....	107
Figura 35.- Diagrama de Secuencia Configurar Captura de Datos (Archivo).....	107
Figura 36.- Diagrama de Secuencia Configurar Captura de Datos (Base de Datos) .....	108
Figura 37.- Diagrama de Secuencia Configurar Captura de Datos (Fórmula) .....	108
Figura 38.- Diagramas de Secuencia del C U Asociar a Objetivos.....	109

## Índice de Tablas

Tabla 1.- Actores del sistema.....	42
Tabla 2.- Descripción del CU Gestionar Indicador.....	45
Tabla 3.- Identificar clases del análisis.....	46
Tabla 4.- Características necesarias para la especificación de requisitos .....	60
Tabla 5.- Matriz de trazabilidad .....	61
Tabla 6.- Lista de Chequeo para validar el listado de requisitos.....	62
Tabla 7.- Secciones a probar en el Caso de Uso: Gestionar Indicador .....	63
Tabla 8.- Descripción de variable. CU Gestionar Indicador .....	64
Tabla 9.- Matriz de Datos. Adicionar indicador .....	65
Tabla 10.- Matriz de Datos. Modificar indicador .....	65
Tabla 11.- Matriz de Datos. Eliminar indicador .....	66
Tabla 12.- Procedimientos por clases .....	67
Tabla 13.- Criterio para calcular la Responsabilidad .....	67
Tabla 14.- Criterio para calcular la Complejidad de implementación .....	68
Tabla 15.- Criterio para calcular la Reutilización .....	68
Tabla 16.- Responsabilidad y Complejidad de las clases.....	68
Tabla 17.- Reutilización de las clases .....	69
Tabla 18.- Relaciones de uso por clases .....	70
Tabla 19.- Acoplamiento de las clases.....	70
Tabla 20.- Reutilización de las clases .....	71
Tabla 21.- Factor de peso de los actores sin ajustar .....	74
Tabla 22.- Factor de peso de los casos de uso sin ajustar.....	74

Tabla 23.- Escala de los factores de complejidad técnica .....	75
Tabla 24.- Peso de los factores de complejidad técnica.....	76
Tabla 25.- Peso de los factores ambientales.....	77
Tabla 26.- Esfuerzo de todo el proyecto por flujos de trabajo.....	78
Tabla 27.- Costo del Proyecto.....	80
Tabla 28.- Descripción del CU Configurar Captura de Datos .....	93
Tabla 29.- Descripción del CU Captura de Datos.....	96
Tabla 30.- Descripción CU Asociar a objetivos .....	97
Tabla 31.- Descripción de la tabla: Indicadores.....	110
Tabla 32.- Descripción de la tabla: Objetivo .....	110
Tabla 33.- Descripción de la tabla: Indicador_objetivo .....	111
Tabla 34.- Descripción de la tabla: FormaDato .....	112
Tabla 35.- Descripción de la tabla: Variable .....	112
Tabla 36.- Descripción de la tabla: TipoBD .....	112
Tabla 37.- Descripción de la tabla: TipoArchivo .....	113

## Introducción

Desde el surgimiento de la administración como ciencia a principios del siglo veinte resalta la preocupación de los administradores por el control en las empresas. Muchos han sido los aportes de las corrientes administrativas al control de la gestión, pero el desarrollo de este término no está a la par del gran desarrollo que ha tenido la ciencia administrativa en las últimas décadas. La necesidad de que el control de la gestión vincule los aspectos internos y externos de la empresa crece a medida que tratan de desenvolverse en un ambiente muy competitivo, lo que exige sacar el mayor provecho de la información interna y externa, para lograr el desarrollo de la estrategia.

En este sentido, y para mejorar la evaluación de la empresa, se presenta el Cuadro de Mando Integral (CMI) como un mecanismo para medir la validez de la estrategia corporativa y la eficacia de su implementación al utilizar indicadores en cuatro perspectivas vinculadas entre sí: Cliente, Aprendizaje y Crecimiento, Financiera y Procesos Internos. Esta herramienta junto con el apoyo de las Tecnologías de la Información y las Comunicaciones (TICs) ofrece una información sencilla, resumida y eficaz para la toma de decisiones.

En Cuba la labor del empresario es fundamental y la necesidad de hacer mucho en poco tiempo, con un sentido integrador, obliga a buscar instrumentos que alerten de las situaciones problemáticas de la entidad con un sentido estratégico y corporativo. Cuba, que se mantiene al tanto de los avances tecnológicos en el mundo con el objetivo de lograr la informatización de la sociedad, introduce las TICs. Es por ello que surgen varios proyectos informáticos dentro de los que resaltan la creación de la Universidad de las Ciencias Informáticas (UCI) como fruto de la Batalla de Ideas que lleva a cabo el país.

La UCI, desde su surgimiento en el 2002, mantiene como premisa la producción de software a partir de la vinculación docencia-producción como modelo de formación. Cuenta con una infraestructura productiva formada por Polos Productivos en las Facultades y recientemente, la creación de centros de desarrollo de software como una primera aproximación a la Industria de Desarrollo de Software Cubana.

En este marco, el Centro de Tecnologías de Gestión de Datos (DATEC) es un centro especializado en tecnologías de bases de datos que provee soluciones integrales y consultorías. Además, desarrolla nuevas tecnologías de bases de datos a través del desarrollo y la gestión de proyectos de Investigación y

Desarrollo (I + D), contribuyendo con su trabajo al cumplimiento de las misiones fundamentales de la Universidad: la formación docente de jóvenes revolucionarios vinculados a la producción de software.

DATEC está estructurado por tres líneas de desarrollo, entre las que se encuentra la Línea de Soluciones Integrales. Precisamente es en esta línea donde se lleva a cabo la creación del Paquete de Apoyo a la Toma de Decisiones y Sistemas Inteligentes (PATDSI). El mismo es concebido como una plataforma Web única de inteligencia de negocios, que integra en sí las funcionalidades más recurrentes y específicas necesarias para la toma de decisiones en diferentes contextos.

En estos momentos PATDSI no cuenta con un sistema automatizado que permita la gestión de los diferentes indicadores que conforman las cuatro perspectivas del Cuadro de Mando Integral, lo que constituye una limitante para el proceso de toma de decisiones.

Por lo anteriormente expuesto se plantea como **problema a resolver**: las insuficiencias existentes en la gestión de indicadores para el Cuadro de Mando Integral de PATDSI está afectando el análisis organizacional para la toma de decisiones a partir de este sistema.

Siendo el **objeto de estudio**: Cuadro de Mando Integral y el **campo de acción**: Gestión de indicadores.

Para darle solución al problema se propone el siguiente **objetivo general**: realizar el análisis y diseño de un componente de gestión de indicadores para PATDSI.

Del cual se desglosan los siguientes **objetivos específicos**:

- Realizar un estudio del estado del arte de los temas relacionados con la gestión de indicadores y su uso para el Análisis Organizacional.
- Realizar el análisis y diseño del componente gestión de indicadores para PATDSI.
- Validar el análisis y diseño del componente gestión de indicadores para PATDSI a partir de métricas.
- Realizar un estudio de factibilidad y análisis de costos del componente gestión de indicadores para PATDSI.

La investigación se sustenta en la siguiente **idea a defender**: la realización del análisis y diseño de un componente de gestión de indicadores para PATDSI, contribuirá a la implementación de la propuesta de componente realizada, para el monitoreo de los indicadores empresariales para la toma de decisiones en la organización.

Las **tareas de la investigación** que se llevarán a cabo para darle cumplimiento a los objetivos propuestos son:

- Valoración del enfoque de autores acerca de la gestión de indicadores y su contribución al Análisis Organizacional.
- Selección de las herramientas y metodología a usar en el análisis y diseño del componente gestión de indicadores para PATDSI.
- Levantamiento de las funcionalidades que debe tener el componente.
- Análisis y diseño del componente propuesto.
- Validación de los resultados obtenidos mediante métricas.
- Estudio de factibilidad.
- Análisis de costos.

En la realización de esta investigación se utilizan los siguientes **métodos científicos**:

- Analítico-Sintético: este método se utilizó para realizar el estudio del arte de la gestión de indicadores.
- Inductivo-Deductivo: para estudiar los fundamentos teóricos y determinar las regularidades que sustentan a la gestión de indicadores, su evolución y actualidad.
- Histórico-Lógico: este método permitió realizar un estudio detallado de la gestión de indicadores y su empleo en el mundo, además de expresar teóricamente la esencia de la trayectoria del proceso desde su desenvolvimiento hasta las conexiones históricas fundamentales.
- Modelación: se utilizó para la creación de modelos alternativos que permite una reproducción simplificada de la realidad y estudiar nuevas relaciones y cualidades de la gestión de indicadores.

Luego de la realización de las tareas antes mencionadas se espera tener el siguiente **resultado**:

- Obtener los artefactos correspondientes al análisis y diseño de un componente para la gestión de indicadores.

La tesis quedará estructurada de la siguiente forma:

**Capítulo 1: Fundamentación Teórica de la Investigación.** Se presenta un estudio del estado del arte de la gestión de indicadores. Se abordan temas y conceptos necesarios para la investigación, se brinda una descripción de las técnicas, metodologías y herramientas utilizadas para dar solución al problema.

**Capítulo 2: Análisis y diseño del componente para la gestión de indicadores.** Se define un modelo de dominio como primera aproximación a la lógica del negocio a definir en el componente. Se especifican las funcionalidades del software a través de los requisitos funcionales y no funcionales. Se hace una descripción de los casos de uso para una mejor comprensión del funcionamiento del componente a diseñar. Y por último se realizan los diagramas asociados al flujo de trabajo análisis y diseño.

**Capítulo 3: Validación de los requisitos y diseño.** Se realiza la validación de los requisitos para verificar si el componente cubre todas las funcionalidades especificadas y que el diseño realizado cumple con los patrones utilizados. Para estas validaciones se utilizan técnicas y métricas ya explicadas en el Capítulo 1.

**Capítulo 4: Estudio de Factibilidad y Análisis de Costos:** Se realiza un estudio de la factibilidad del componente diseñado, así como un análisis de los costos asociados a su diseño e integración con otros sistemas.

## Capítulo 1: Fundamentación Teórica de la Investigación

En el presente capítulo se realiza el estudio teórico que fundamenta la investigación. Se realiza una sistematización de los sistemas de gestión de la información en el mundo y las tendencias que han surgido durante su desarrollo. Este estudio es aterrizado en el ámbito mundial y nacional. Se incluye además el análisis de las tecnologías empleadas en la solución, así como, los artefactos desarrollados según la metodología seleccionada.

### 1. Bases conceptuales

Para entender la gestión de indicadores se introducen definiciones de diferentes autores vinculadas con los indicadores y su uso como parte de las perspectivas del Cuadro de Mando Integral, definiendo como guía de la investigación un conjunto de conceptos para el desarrollo de la misma.

#### 1.1. Cuadro de Mando Integral

Sistema de medición estratégico que utiliza cuatro perspectivas para controlar la implementación de la estrategia en sus factores críticos de éxito y su adecuación al entorno. Estas perspectivas son: perspectiva financiera, perspectiva del cliente, perspectiva de procesos internos y de formación y perspectiva de crecimiento. Todas ellas ligadas, contribuyen a controlar el cumplimiento de los objetivos estratégicos midiendo los factores clave y sus inductores de actuación para lograr un control anticipado y enfocado no solamente en la actuación financiera. Es la filosofía perfecta que permite a la dirección de una institución inducirla al éxito competitivo. *(Ver Figura 1.- Representación de un CMI según Norton y Kaplan)*

##### 1.1.1. Dashboard

El *Dashboard* o panel de información es una herramienta de reporte para visualizar y consolidar resúmenes de información del negocio. También son utilizados para mostrar los valores de un proceso o las cifras que genera un departamento. No tiene por qué ajustarse a una metodología de gestión. Un cuadro de mando puede ser parte de un *Dashboard*, pero se diferencian en que un cuadro de mando incluye múltiples puntos de vista y aplica una metodología de gestión, tal como el *Balanced Scorecard* (Cuadro de Mando Integral, CMI). El *Dashboard* muestra en tiempo real la comprensión global de las condiciones del negocio mediante métricas e Indicadores Clave de Rendimiento (*Key Performance Indicators - KPI*). Su característica de tiempo real otorga a los usuarios un conocimiento completo sobre la

marcha de la empresa y permite hacer análisis instantáneos e inteligencia de negocios (Díaz Hernández, et al., 2009).

## 1.1.1.1. Key Performance Indicators

Los indicadores clave de rendimiento son mediciones cuantificables que reflejan los factores críticos y de éxito de una organización. Son utilizados para evaluar los progresos realizados respecto a los objetivos previamente acordados, permitiendo que los usuarios del negocio puedan saber instantáneamente si están dentro o fuera de su plan estratégico (Imetrix, 2009).



Figura 1.- Representación de un CMI según Norton y Kaplan

Por lo anteriormente expuesto se llega a la conclusión que el uso de un CMI como sistema de control empresarial brindará un gran número de beneficios para las organizaciones, ya que les permitirá medir el progreso actual y suministrar la dirección futura de la organización, posibilitando convertir la misión en acción por medio de un conjunto coherente de objetivos, indicadores e iniciativas agrupadas en perspectivas. Su fortaleza principal está en el monitoreo de los indicadores, para ello, estos deben ser definidos y gestionados correctamente, seguidamente se hará un estudio para tener un mayor conocimiento de los mismos.

## 1.2. Estudio de Indicadores

Los indicadores surgen ante la necesidad de tener un mejor control de todos los procesos en la organización para medir, conocer y analizar los resultados de esta labor. Constituyen el mecanismo más eficaz y menos costoso para saber hacia dónde va la empresa.

Los indicadores son utilizados en la inteligencia de negocios para evaluar el estado actual del negocio y sugerir cursos de acción. La acción de monitorear estos indicadores en tiempo real se conoce como monitoreo de la actividad del negocio y es usado con frecuencia para medir o darle valor a actividades difíciles de medir (Vega Rosales, et al., 2005).

Los indicadores también son utilizados para realizar mediciones de los logros y objetivos de un determinado proceso de gestión que permiten administrar realmente un proceso, estos ayudan a establecer y administrar acciones concretas para hacer realidad las tareas y trabajos programados y planificados. Sirven como herramienta para el mejoramiento continuo de la calidad en la toma de decisiones, lo cual se traduce en una mejor calidad del producto o del servicio resultado del proceso.

Por todo lo antes estudiado se puede llegar a la conclusión de que el uso de los indicadores en el control empresarial, son la base del proceso de medición de la organización, pues a partir de ellos se puede realizar un análisis exhaustivo, con el objetivo de mejorar las posibles desviaciones que se manifiesten dentro de los principales procesos que intervienen directamente en el cumplimiento de la estrategia corporativa, de ahí la importancia de realizar un estudio más detallado de los mismos.

### 1.2.1. Técnicas para elaborar indicadores

El proceso para elaborar indicadores define tres pasos fundamentales (Liliana Cabrera, 2008):

Paso 1. Definir los atributos importantes. Mediante el uso de un diagrama de afinidad (tormenta de ideas mejorada) obtener el mayor número de ideas acerca de medidores o indicadores que puedan utilizarse para medir las actividades.

Paso 2. Evaluar si los medidores o indicadores tienen las características deseadas, es decir, si son medibles, entendibles y controlables.

Paso 3. Comparar contra el conjunto de medidores o indicadores actuales para evitar redundancia o duplicidad.

El propósito de estos pasos es tener bien identificado que es lo que se quiere medir con el indicador, para que una vez elaborados los mismos no existan ambigüedades a la hora de monitorearlos.

## 1.2.2. Características de los indicadores

Específicamente se puede describir los indicadores como medidas estándares que se usan para evaluar y comunicar los resultados obtenidos en comparación con los esperados, los mismos deben cumplir con una serie de características entre las que se encuentran (ALTAIR, 2005):

- Referirse a procesos importantes o críticos.
- Representar fielmente el objetivo a medir mediante una relación directa.
- Ser cuantificables a través de datos numéricos o un valor de clasificación.
- Ser rentables, superando el beneficio de su uso al coste de su obtención.
- Poder definir la evolución en el tiempo del objetivo siendo comparables en el tiempo.
- Ser fiables para dar confianza a los usuarios sobre su validez.
- Ser fáciles de mantener y utilizar.
- No interferir con otros indicadores siendo compatible con ellos.
- Permitir a la dirección conocer la información en tiempo real.

El diseño de los indicadores debe tener una amplia visión para no observar solamente los impactos inmediatos, sino también el proyecto futuro, para de esta forma tomar las decisiones pertinentes en caso de posibles desviaciones durante el cumplimiento de un objetivo trazado.

## 1.2.3. Tipos de indicadores

Existen diversos tipos de indicadores que son fundamentales a la hora de determinar si las empresas se dirigen hacia los objetivos trazados, a continuación se muestran los más utilizados.

### ❖ Indicadores Financieros

Las medidas financieras son inadecuadas para guiar y evaluar la trayectoria de la organización a través de los entornos competitivos; sin embargo tienen un papel esencial a la hora de alertar a los ejecutivos en torno a la mejora de la calidad, de los tiempos de respuesta, de la productividad y de los nuevos productos.

Un amplio sistema de indicadores de gestión ha de especificar la forma en que las mejoras en operaciones, servicios al cliente y nuevos productos se vinculan a una actuación financiera mejorada, a través de mayores ventas, mayores márgenes operativos, menores gastos de operativos y una rápida rotación de activos (Universidad Nacional de Colombia, 2010).

## ❖ **Indicadores Numéricos**

Un Indicador Numérico es una relación entre dos o más datos significativos que tienen una relación lógica entre ellos y que entregan información de aspectos críticos de vital importancia para la conducción de la organización. Existen distintas clases de indicadores numéricos que son obtenidos a través de razones, porcentajes y promedio.

Los Indicadores Numéricos se pueden clasificar en (Restrepo de Ocampo, Estrada Mejía, & Ramírez Aristizabal, 2007):

- **Indicadores de Resultados:** A los que también se les conoce como indicadores de efecto. Miden la consecución del objetivo estratégico.
- **Indicadores de Causa:** A los que también se les conoce como indicadores inductores. Miden el resultado de las acciones que permiten su consecución.

## ❖ **Indicadores de Gestión**

Los Indicadores de Gestión son la expresión cuantitativa del comportamiento o desempeño de toda una organización o una de sus partes, cuya magnitud al ser comparada con algún nivel de referencia, puede estar señalando una desviación sobre la cual se tomarán acciones correctivas o preventivas según el caso. Sus mediciones están relacionadas con el modo en que los servicios o productos son generados por la institución. El valor de este tipo de indicador es el resultado de la medición del mismo y constituye un valor de comparación referido a su meta asociada.

Para la definición de estos indicadores se deben identificar necesidades propias del área involucrada, clasificándose según la naturaleza de los datos y la necesidad. Su principal objetivo es poder evaluar el desempeño del área mediante parámetros establecidos en relación con las metas y observar la tendencia en un período de tiempo durante un proceso de evaluación. Con los resultados obtenidos se pueden plantear soluciones o herramientas que contribuyan al mejoramiento o correctivos que conlleven a la consecución de la meta fijada (Liliana Cabrera, 2008).

Los Indicadores de Gestión son fundamentales para:

- Poder interpretar lo que está ocurriendo.
- Tomar medidas cuando las variables se salen de los límites establecidos.
- Definir la necesidad de introducir un cambio y poder evaluar sus consecuencias.
- Planificar actividades como respuesta a nuevas necesidades.

El equipo debe definir los indicadores a partir de las siguientes preguntas:

- ¿Qué se debe medir?
- ¿Dónde es conveniente medir?
- ¿Cuándo hay que medir?, ¿En qué momento o con qué frecuencia?
- ¿Quién debe medir?
- ¿Cómo se debe medir?
- ¿Cómo se van a difundir los resultados?
- ¿Quién y con qué frecuencia va a revisar o auditar el sistema de obtención de datos?

## ❖ Indicadores Estratégicos

Los Indicadores Estratégicos son parámetros cualitativos o cuantitativos que define los aspectos relevantes sobre los cuales se lleva a cabo la evaluación para medir el grado de cumplimiento de los objetivos planteados en términos de eficiencia, eficacia y calidad, además, miden el grado de cumplimiento de los planes estratégicos o de desarrollo (Rubino, 2007).

## ❖ Indicadores de Eficacia

Eficaz tiene que ver con hacer efectivo un intento o propósito. Los indicadores de eficacia están relacionados con las razones que indican capacidad o acierto en la consecución de tareas y trabajos (Rosero, et al., 2010).

## ❖ Indicadores de Eficiencia

Los Indicadores de Eficiencia miden el nivel de ejecución del proceso, se concentran en el cómo se hicieron las cosas y miden el rendimiento de los recursos utilizados por un proceso relacionado con la productividad (Businesscol.com, 2009).

Dentro de cada una de las perspectivas de aprendizaje y crecimiento, cliente y procesos internos, se encuentran un grupo de indicadores no financieros también denominados: indicadores avanzados, porque van por delante de la rentabilidad, o sea, que si se obtiene de ellos una buena respuesta, es de esperar que se alcancen buenos resultados económicos. Esto es posible debido a que en el CMI se traza la estrategia relacionando cada una de estas perspectivas antes mencionadas de modo que converjan al final en la perspectiva financiera.

### 1.3. Fuentes de captura de indicadores

La captura de indicadores es un proceso primordial para gestionar la información de las empresas. Existen varias fuentes de almacenamiento de datos que permiten la obtención del valor de los indicadores. Dentro de estas fuentes se presentan sistemas gestores de bases de datos comerciales tales como: Microsoft SQL Server, Microsoft Access y Oracle como los más conocidos, aunque también se mencionan: IBM Informix, Interbase de CodeGear, MAGIC, FoxPro y Paradox. Sobre Sistemas de Gestión de Bases de Datos (SGBD) gratuitos se pueden mencionar: Microsoft SQL Server Compact Edition y Sybase ASE Express Edition para Linux. Por último, y más aceptados por las ventajas del uso del software libre a nivel mundial, están los SGBD libres: MySQL, SQLite, Apache Derby, Firebird y PostgreSQL.

### 1.4. Beneficios y riesgos de los indicadores en la implantación de un CMI

El CMI ofrece un método más estructurado de selección de indicadores y esto le concede más versatilidad dentro de la gestión de la organización. Con esta herramienta se integran tanto el aspecto gerencial estratégico, como la evaluación del desempeño. El conjunto de información que se obtiene de los indicadores que integran un CMI bien establecido y alineado con la visión y estrategia general, acaba transformándose en un sistema de gestión, un modo de gestión en el que se asumen los cambios del entorno y con el que se establecen mecanismos de mejora y adaptación.

No obstante, lo que ha convertido al CMI en una de las herramientas más significativas de los últimos años, es que se basa en un modelo de negocio. El éxito de su implantación radica en que el equipo de dirección se involucre y le dedique tiempo al desarrollo de su propio modelo de negocio.

A continuación se muestran una serie de beneficios y riesgos que pueden estar presente durante la implantación de un CMI (Sinnexus, 2007):

## **Beneficios de la implantación de un Cuadro de Mando Integral**

- La fuerza de explicitar un modelo de negocio y traducirlo en indicadores, facilita el consenso en toda la empresa, no sólo de la dirección, sino también de cómo alcanzarlo.
- Mediante sus indicadores clarifica cómo las acciones del día a día afectan no sólo al corto plazo, sino también al largo plazo.
- Una vez que el CMI está en marcha, se puede utilizar para comunicar los planes de la empresa, aunar los esfuerzos en una sola dirección y evitar la dispersión. En este caso, el CMI actúa como un sistema de control por excepción.
- El control de los indicadores permite detectar de forma automática desviaciones en el plan estratégico u operativo, e incluso indagar en los datos operativos de la compañía hasta descubrir la causa original que dio lugar a esas desviaciones.

## **❖ Riesgos de la implantación de un Cuadro de Mando Integral**

- Si los indicadores no se escogen con cuidado, el CMI pierde una buena parte de sus virtudes, porque no comunica el mensaje que se quiere transmitir.
- Cuando la estrategia de la empresa está todavía en evolución, es erróneo que el CMI se utilice como un sistema de control, en lugar de usarlo como una herramienta de aprendizaje.
- Existe el riesgo de que lo mejor sea enemigo de lo bueno, de que el CMI sea perfecto, pero desfasado e inútil.

Es muy importante destacar que antes de la implantación de un CMI debe hacerse un profundo análisis tanto de la organización como del entorno, con el objetivo de definir los indicadores que lleven a mejorar el control de la organización.

## **1.5. Gestión de indicadores. Contexto Actual**

La gestión de indicadores desde su aparición ha tenido una constante evolución así como una mayor aplicación por parte de las empresas, instituciones u organismos que se han dado cuenta de su

importancia para lograr el éxito. Se han creado diversos software que mediante un grupo de indicadores implementan un CMI brindando resultados favorables para aquellos que hacen uso de los mismos.

## **1.5.1 Gestión de indicadores. Tendencias mundiales**

Las empresas que en el mundo han implantado un CMI tienen hoy una economía mucho más favorable, cuentan con personal más capacitado en el trabajo y mayor cantidad de clientes, ya que midiendo su rendimiento de manera sostenida son mucho más eficientes, saben en cuál o cuáles indicadores hacer énfasis una vez observado el rendimiento de éstos y suelen ser más competitivas en un mundo bastante convulso.

### **1.5.1.1 Software para la gestión de indicadores utilizando Cuadro de Mando Integral**

Existen internacionalmente diversos software que le facilitan a las organizaciones la gestión de indicadores mediante la construcción de CMI, la mayoría de los paquetes que existen actualmente están en inglés y son propietarios. El objetivo de estos programas es proporcionar informes claros, concisos y relevantes sobre la marcha de la organización, entre ellos se encuentran:

#### **❖ Delphos**

Delphos es un software de control de gestión, desarrollado por la compañía DEINSA, que permite implementar íntegramente un CMI, un plan estratégico, un plan anual operativo o cualquier otro modelo que se requiera para controlar las operaciones de cualquier tipo de organización, e inclusive mejorar el desempeño y la productividad (Delphos, 2008). *(Ver Figura 2)*

Entre sus principales beneficios están:

- Promueve la alineación estratégica de toda la organización a partir de la transformación de la visión y la estrategia en planes concretos de acción.
- Fomenta el trabajo en equipo y por consiguiente la colaboración y la coordinación al conducir a toda la organización hacia la consecución de la estrategia definida.
- Integra y sintetiza un gran volumen de datos e indicadores que surgen de la gestión diaria de las operaciones.

- Dispone de un sistema de alertas y envío automático de correos electrónicos, que le permitirá a los usuarios designados estar informados, al momento de presentarse una situación que amerite su intervención.

Como es indicado, entre uno de sus principales beneficios se identifica la gestión de indicadores, la cual es realizada de manera íntegra por este potente sistema. Este software captura toda la información identificativa de cada indicador, además de todos los metadatos asociados a la información que los mismos contendrán, ejemplo: rangos de valores, modo de captura de datos, valores históricos, entre otros.

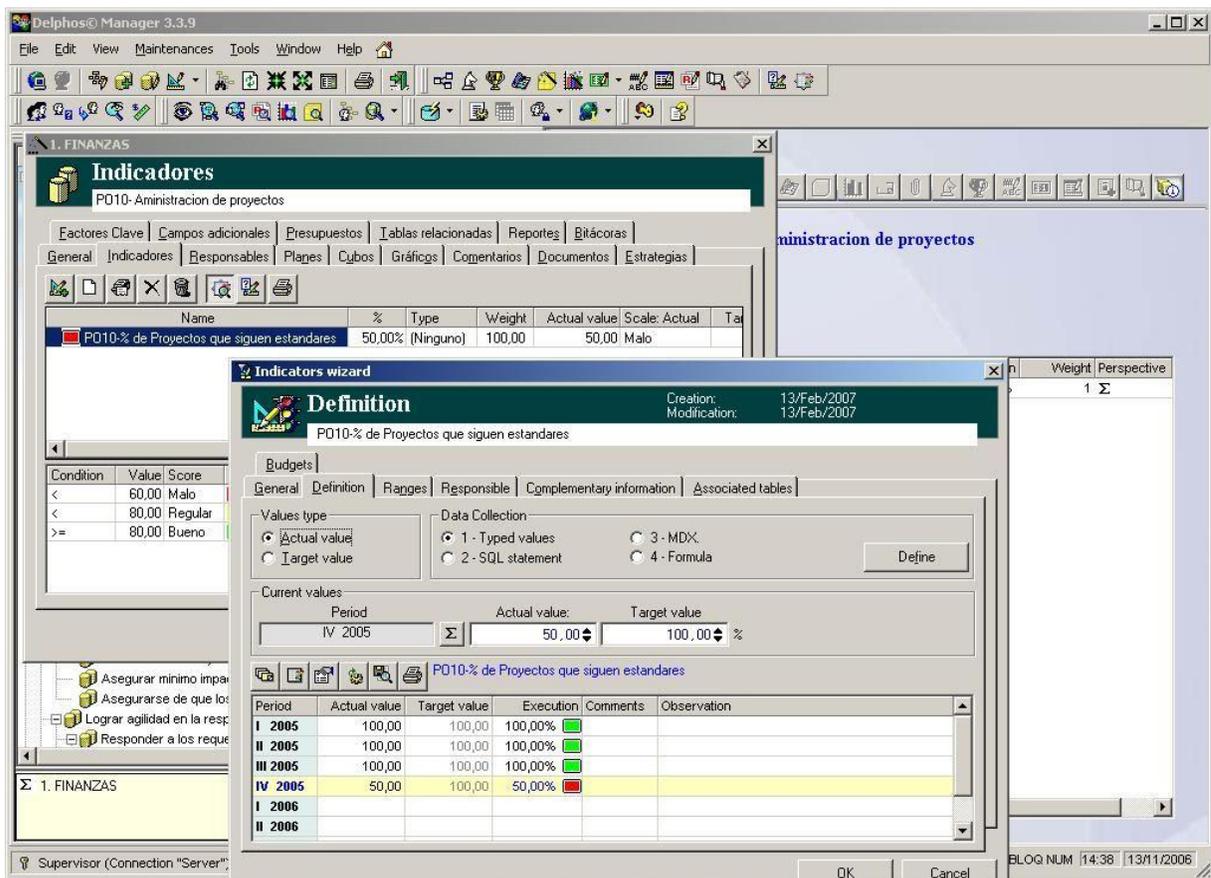


Figura 2.- Consola del Software Delphos

Se está en presencia de uno de los sistemas más completos existentes en el mundo en la actualidad que implementan un CMI. Por su parte el componente de gestión de indicadores que contiene está al nivel del mismo, con una implementación realmente completa. Dicho componente sería el ideal, de acuerdo con el software que se desea desarrollar, pero éste tiene un carácter propietario del sistema general y además

no se encuentra desacoplado como componente independiente. De cualquier modo puede ser tomado en cuenta, de acuerdo con su completitud, como prototipo guía para el diseño de un componente propio.

## **1. Dialog Strategy**

Dialog Strategy es un software de Cuadro de Mando Integral desarrollado por CACIT Group, ideal para aquellas empresas con una administración detallada necesitadas de una herramienta que permita medir su estrategia y desempeño de manera sencilla. Este software permite definir una serie de indicadores, objetivos estratégicos, enlazarlos en una relación causa-efecto, asignar las medidas apropiadas, introducir datos y comenzar a monitorizar el rendimiento de su compañía (Axsellit, 2010).

Entre sus beneficios se encuentra la fácil presentación de textos y gráficos, pero principalmente la simple, pero efectiva, manera de importar datos a través de un utilitario importador de archivos planos (DSI). Permite el ingreso de información de diferentes maneras, partiendo de un simple “cut and paste” desde una hoja de cálculo o archivos CSV, hasta la definición de fórmulas para cálculo del valor de un indicador basado en el valor de otros. Se está aquí en presencia de otro potente software que, además de ser libre, realiza una buena gestión de indicadores. Su principal desventaja está en la imposibilidad de acoplarlo como herramienta de PATDSI.

## **❖ CódigoAberto CMI**

CódigoAberto CMI está basado en una plataforma desarrollada bajo software libre y cubre amplias necesidades de análisis de datos y de informes empresariales. Las soluciones de CódigoAberto CMI están escritas en Java y tienen un entorno de implementación también basado en Java. Eso hace que CódigoAberto CMI sea una solución muy flexible para cubrir una amplia gama de necesidades empresariales tanto las típicas como las sofisticadas y específicas al negocio (Soluciones basadas en Software Libre, 2008).

Este sistema, cuenta con la ventaja de haber sido desarrollado bajo licencia de código abierto, pero la gestión de indicadores es muy básica, basada principalmente en la carga de información a partir de ETL (Extraction, Transforming & Load, según siglas en inglés), sin permitir demasiadas opciones en el trabajo sobre los indicadores, como por ejemplo escoger diferentes formas de captura de información, lo cual lo convertiría en un sistema versátil, como el que se necesitaría para el desarrollo del CMI de PATDSI.

## 1.5.2 Gestión de indicadores. Caso Cuba

Cuba no ha estado ajena al desarrollo de software de CMI existente en el mundo. A principios del siglo XXI comienzan a implementarse en las empresas cubanas diferentes sistemas informáticos para la administración de las empresas. Aunque todavía no es un movimiento popular se conoce que en el año 2003 la empresa Intermar Cienfuegos tenía un cuadro de mando para la gestión empresarial. Más tarde otras empresas como GET Varadero, SEPSA Cienfuegos, CENEX, Cubalse Cienfuegos, el Centro de Estudios Contables, Financieros y de Seguros (CECOFIS), Cubapetróleo (CUPET), Ministerio de Ciencia Tecnología y Medio Ambiente de Cuba (CITMA) y la Empresa Cubana Nacional de Software (DESOFIT) han incorporado a su sistema empresarial CMI (González, 2006).

Todos estos sistemas, que incluyen además la gestión de indicadores, han sido desarrollados bajo los requisitos específicos de las empresas que lo utilizan. Esta característica los hace exclusivos para el uso de las mismas, convirtiéndolos así en inutilizables para PATDSI, aunque sin dejar de ser una referencia a cómo debe ser desarrollado este tipo de software.

### ❖ Gestionar Indicadores. PATDSI SIGE

Precisamente, y aunque parezca contradictorio, PATDSI cuenta ya con una herramienta para la gestión de indicadores. Dentro de los módulos pertenecientes al Sistema de Gestión Estadística incluido en el PATDSI, se encuentra el módulo Registros y Clasificadores, el cual cuenta con la aplicación Gestión de Indicadores. La desventaja en este caso está en que la información gestionada por este componente del sistema está enfocada netamente al trabajo estadístico. De cualquier modo fue realizado un análisis a partir del mismo que sirvió de base para la definición de interfaces e información a gestionar.

Luego de haber sido analizados diferentes sistemas que implementan CMI y los componentes de gestión de indicadores asociados a éstos, ya sea a nivel internacional o en el caso de Cuba, se llegó a la conclusión de que los mismos no cumplían de una forma u otra con las necesidades o características para la gestión de indicadores de un CMI para PATDSI. Por esta razón se decidió el desarrollo de dicho componente, aunque el alcance de esta investigación solo incluye el análisis y diseño del mismo.

## 1.6. Metodologías de desarrollo de software

El principal objetivo de las metodologías de desarrollo es elevar la calidad del software a través de una mayor transparencia y control sobre el proceso. La competencia del mercado y el contar con la

información en tiempo y forma, requiere ser seguros, precisos y confiables creando la necesidad de trabajar con aplicaciones de software cada vez mejores, con mayor cantidad de prestaciones y que permitan a su vez un eficiente desarrollo y mantenimiento. Existen estándares y procesos de desarrollo de software que determinan buenas prácticas para el posterior desarrollo de las aplicaciones.

En el desarrollo de software es muy importante realizar un buen análisis, esta fase tiene un peso fundamental para llegar a obtener un buen software. Realizar un buen análisis constituirá una pieza fundamental para el posterior desarrollo del software. A continuación se mencionan algunas de las metodologías más usadas:

## ❖ **Proceso Unificado de Rational**

Es una infraestructura flexible de desarrollo de software que ha sido probada y a su vez proporciona prácticas recomendadas así como una arquitectura configurable. Unido al Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico proceso unificado. No es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Es una forma disciplinada de asignar tareas y responsabilidades (quién (trabajador) hace qué (artefacto), cuándo (flujo de actividades) y cómo (actividades)) (Ivar Jacobson, 2000).

## ❖ **Programación Extrema**

La Programación Extrema (XP) es una metodología de desarrollo de software ágil basada en una serie de valores y de prácticas que posibilitan comunicación, realimentación y reutilización del código con el cual se trabaja, además tiene como objetivo aumentar la productividad a la hora de desarrollar programas. XP procura la sencillez en el software.

La Programación Extrema asume que la planificación nunca será perfecta, y que variará en función de cómo varíen las necesidades del negocio. Por tanto, el valor real reside en obtener rápidamente un plan inicial y contar con mecanismos de retroalimentación que permitan conocer con precisión dónde están. En XP la planificación es iterativa por lo que un representante del negocio decide al comienzo de cada iteración qué características concretas se van a implementar (Escribano, 2002).

## ❖ **Metodología del grupo de análisis de la Línea de Soluciones Integrales para el Análisis de Datos de DATEC**

DATEC presenta una metodología centrada en la organización propuesta para la aplicación del Modelo de Líneas de Productos de Software en el Centro, la cual está basada en un grupo de metodologías de desarrollo de software, estas metodologías son: Open Up, Scrum y PMBok. A continuación se explican brevemente un conjunto de características que presentan estas metodologías.

**Open Up:** es un proceso de desarrollo iterativo del software que es (Gestión de Proyectos , 2008):

- Mínimo: solo incluye el contenido del proceso fundamental.
- Completo: puede ser manifestado como proceso entero para construir un sistema.
- Extensible: puede ser utilizado como base para agregar o para adaptar más procesos.

Open Up es apropiado para proyectos pequeños y de bajos recursos, ya que permite disminuir las probabilidades de fracaso e incrementa las probabilidades de éxito. Además, detecta errores tempranos a través de un ciclo iterativo, evitando la elaboración de documentos, diagramas e iteraciones innecesarias requeridos en la metodología RUP. Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

**Scrum:** es un proceso en el que se aplican de manera regular un conjunto de mejoras prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados rápidos, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (Ejemplos java y C/linux, 2009).

**PMBok:** es una metodología de gestión de proyectos la cual documenta y estandariza información y prácticas generalmente aceptadas para la gestión de proyectos. Esta metodología trata varias áreas de conocimiento relacionadas con la gestión de proyectos, entre las que se pueden mencionar: integración, alcance, tiempo, calidad, coste, riesgo, recursos humanos, comunicación, compras y adquisiciones. La aplicación de esta metodología permitirá llevar una buena gestión del proyecto y mantener un mayor control, permitiéndole al jefe de proyecto y a su equipo realizar proyectos de manera eficaz y eficiente, así como asegurar la calidad y transparencia a lo largo de toda la vida del proyecto (Riebeling, 2009).

Por política del Centro para el desarrollo de esta investigación fue seleccionada la *Metodología del grupo de análisis de la Línea de Soluciones Integrales para el Análisis de Datos de DATEC*.

## 1.7. Lenguaje de modelado

El Lenguaje Unificado de Modelado (UML) es un grupo de especificaciones de notación orientadas a objeto, las cuales están compuestas por distintos diagramas, que representan las diferentes etapas del desarrollo de un proyecto de software. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Este lenguaje indica cómo crear y leer los modelos, pero no dice cómo desarrollarlos, esto último es el objetivo de las metodologías de desarrollo.

Entre los principales objetivos de UML se pueden sintetizar sus funciones en (Popkin Software and Systems, 2009):

- Visualizar: permite expresar de una forma gráfica un sistema de forma que otro lo pueda entender.
- Especificar: 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.

## 1.8. Herramientas CASE para el desarrollo de software

Las herramientas CASE proporcionan un beneficio sustancial para las organizaciones al facilitar la revisión de las aplicaciones. El principal objetivo de estas herramientas consiste en representar objetos de datos de negocios y sus relaciones, además, ayuda a tener una mejor comprensión de la forma en que fluyen estos objetos de datos entre distintas zonas del negocio. El uso de estas herramientas proporciona una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información o cuando los métodos y sistemas no satisfacen las necesidades de la organización (Ingeniería del software, 2010).

## ❖ Rational Rose

Rational Rose, es una herramienta de diseño unificada, orientada al objeto del software del lenguaje de modelado UML, forma parte de un conjunto más amplio de herramientas que todas juntas abarcan el ciclo de vida del desarrollo de software. Permite completar diferentes disciplinas (flujos fundamentales) de RUP; incluye, a su vez, herramientas de ingeniería inversa y generación de código que facilitan el tránsito hasta el producto final. Rational Rose, es considerada una de las mejores herramientas para traducir requisitos de alto nivel a una arquitectura basada en componentes. Se encuentra en la avanzada en cuanto al desarrollo con UML por lo que se ha convertido en una de las mejores opciones por la notación estándar que provee para especificar, visualizar y construir productos de software (Rational Rose Enterprise Edition, 2008).

## ❖ Visual Paradigm

Es un software privativo, para modelado en UML. Esta herramienta tiene unas características gráficas muy cómodas, que facilitan la realización de los diagramas de modelado que sigue el estándar UML, los mismos son: diagramas de clases, casos de uso, comunicación, secuencia, estado, actividad y componentes.

Entre las principales ventajas de *Visual Paradigm* están:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas.
- Producto de calidad.
- Soporta aplicaciones Web.
- Varios idiomas.

- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

Visual Paradigm 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. Es una herramienta multiplataforma que se integra con varios IDEs (Ambiente Integrado de Desarrollo) y soporta múltiples usuarios trabajando sobre un mismo proyecto, permitiendo así generar la documentación del proyecto automáticamente en varios formatos (Visual Paradigm for UML (Standard Edition), 2009).

Por política del Centro y los elementos vistos anteriormente, la herramienta a utilizar para el modelado es *Visual Paradigm*.

## 1.9. Sistema Gestor de Bases de Datos

Un Sistema Gestor de Bases de Datos (SGBD) o DBMS (*DataBase Management System*) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos (Cavsi, 2008).

Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server.

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

### ❖ Oracle

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales. Es el mayor y más usado Sistema de gestión de bases de datos relacional (RDBMS) en el mundo. La Corporación Oracle lo ofrece como un producto incorporado a la línea de producción. Además, incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios. Garantiza la autenticidad apropiada de los usuarios y la privacidad e

integridad de los datos, permite manejar la asignación de privilegios, monitorear las operaciones de las bases de datos a lo largo de toda la empresa y su arquitectura ofrece escalabilidad para soportar un gran número de usuarios y cargas de trabajo de alto volumen de transacciones (Desarrollo Web, 2007).

## ❖ MySQL

MySQL es un gestor de bases de datos de fuente abierta más popular en el mundo. Su arquitectura lo hace extremadamente rápido y fácil de adaptar. Este servidor de bases de datos potente es usado por muchos programas de origen abierto incluyendo *phpBB*, *osCommerce*, y *phpNuke*.

MySQL proporciona un servidor de bases de datos SQL (*Structured Query Language*) veloz, multi-hilo, multiusuario y robusto. El servidor está proyectado tanto para sistemas críticos en producción soportando intensas cargas de trabajo como para empotrarse en sistemas de desarrollo masivo de software. El software MySQL tiene licencia dual, pudiéndose usar de forma gratuita bajo licencia GNU o bien adquiriendo licencias comerciales de MySQL AB en el caso de no desear estar sujeto a los términos de la licencia GPL (MySQL, 2010).

## ❖ PostgreSQL

Es un potente sistema de gestión de bases de datos objeto-relacional (O-RDBMS), multiusuario, centralizado y de propósito general, que ha sido desarrollado de diversas maneras desde 1977, está liberado bajo la licencia BSD (Berkeley Software Distribution). Es ampliamente considerado como el sistema gestor de bases de datos de código abierto más avanzado del mundo. Fue pionero en muchos conceptos que estuvieron disponibles en algunos sistemas de bases de datos comerciales de alto calibre como por ejemplo: (MVCC, gestión de transacciones, savepoints). Fue uno de los primeros intentos en implementar un motor de bases de datos relacional. (PostgreSQL Global Development Group, 2009)

Entre sus principales características se pueden mencionar las siguientes (Medrano, 2009):

- Organiza los datos mediante un modelo objeto-relacional.
- Capaz de manejar procedimientos, rutinas complejas y reglas.
- Soporta tablespaces, transacciones anidadas, copias de seguridad en línea y soporte para parte de los estándares SQL 92, 99, 2003 y 2008.
- Ofrece transacciones que permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.

- Es altamente extensible, soporta operadores, funciones, métodos de acceso y tipos de datos declarados por el usuario; soporta además sobrecarga de operadores, sobrecarga de procedimientos, vistas materializables, particionamiento de tablas y datos.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de la información dentro de la base de datos.
- Las restricciones y disparadores tienen la función de mantener la integridad y consistencia en las bases de datos.
- Usa una arquitectura cliente/servidor basada en un proceso por usuario. Existe un proceso maestro que se ramifica para proporcionar conexiones adicionales por cada cliente que se intenta conectar a *PostgreSQL*.

El gestor de base de datos *PostgreSQL* es robusto y por ende hoy en día es el más usado con respecto a gestores libres existentes como: *SQLite*, *MySQL* en su versión libre, *FireBird*, *DB2 Express-C* y *Apache Derby*, ya que presenta una mayor escalabilidad y rendimiento bajo grandes cargas de trabajo.

Por todo lo anteriormente visto y por política del Centro es seleccionado como Sistema de Gestión de Bases de Datos *PostgreSQL*.

## **1.10. Captura de requerimientos**

La captura de requerimientos es el proceso de identificar qué quiere el cliente del sistema propuesto. Hoy en día es una tendencia la poca comunicación entre los clientes y los desarrolladores de software, con motivo de esta situación se plantean un conjunto de técnicas que permiten mejoras en la captura de las funcionalidades del software (Ingeniería de Software, 2008).

Las técnicas que se usarán en la captura de requerimientos son las siguientes:

### **1.10.1. Introspección**

Esta técnica recomienda que el ingeniero de requerimientos se ponga en el lugar del cliente y trate de imaginar cómo desearía éste el sistema. En base a estas suposiciones comenzaría a recomendar al cliente sobre la funcionalidad que debería presentar el sistema. El problema radica en que un ingeniero no es un tipo normal de cliente, éste posee un conocimiento elevado por lo que podría recomendar más de lo que el cliente necesita.

## 1.10.2. Discusiones

Este tipo de entrevistas pretende que se sostenga una discusión con el cliente sobre su problemática para tratar de determinar en conjunto los requerimientos del sistema.

## 1.10.3. Prototipos

Es una pequeña muestra de funcionalidad limitada, de cómo sería el producto final una vez terminado. Ayudan a conocer la opinión de los usuarios y rectificar algunos aspectos antes de llegar al producto terminado.

## 1.11. Validación de los requerimientos y diseño

Los requerimientos y diseño una vez definidos necesitan ser validados. La validación de requerimientos es muy importante, pues un levantamiento de requerimientos con errores que no se detecten a tiempo, además de no conducir a resultados inesperados provocan costos excesivos y gran pérdida de tiempo (Ingeniería de Software, 2008).

Pocas son las propuestas existentes que ofrecen técnicas para la realización de la validación y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requerimientos con el usuario para detectar errores o inconsistencias. Aún así, existen algunas técnicas que pueden aplicarse, las siguientes son las que se usarán para validar los requerimientos:

### ❖ Casos de Prueba

Cada requerimiento debe ser probado, por tanto, debería asociarse un caso de prueba a cada requerimiento. Un caso de prueba para requerimientos funcionales generalmente consiste en definir las entradas, salidas del software y acciones del usuario para que se pueda completar lo que expresa el requisito (Ingeniería de Software, 2008).

### ❖ Listas de chequeo

Son frecuentemente usadas en inspecciones o revisiones de artefactos generados en el proceso de producción de software; son listas de aspectos que deben ser completados o verificados. Esta herramienta utiliza preguntas orientadas a identificar problemas por áreas y sirven para motivar posibles soluciones o la detección de oportunidades de mejora. Para identificar las oportunidades de mejora es importante realizar una revisión al documento que se desea comprobar (Bichachi, 2009).

## ❖ Métricas

Una métrica es una medida efectuada sobre algún aspecto del sistema en desarrollo o del proceso empleado que permite previa comparación con unos valores de referencia, obtener conclusiones sobre el aspecto medido con el fin de adoptar las decisiones necesarias. Las métricas examinan el modelo de análisis con el fin de predecir el tamaño del sistema resultante, en donde resulte probable que el tamaño y la complejidad del diseño estén directamente relacionadas. Estas mediciones permiten tener una visión más concisa proporcionando un mecanismo para la evaluación objetiva (Carrasco, 2008)

## ❖ Matriz de trazabilidad

La matriz de trazabilidad es una técnica empleada para validar los requerimientos identificados, esta técnica consiste en verificar que los casos de uso que tenemos satisfagan todos los requerimientos del sistema.

Una matriz de trazabilidad es creada por la asociación de necesidades con los productos de trabajo que las satisfacen. Las pruebas se asocian con los requisitos en los que se basa y el producto a ser evaluada con el requisito. La trazabilidad garantiza la integridad, que todos los requisitos de nivel inferior provienen de los requerimientos de nivel superior, y que todos los requisitos de nivel superior se asignan a los niveles inferiores. La trazabilidad también proporciona la base para la planificación de controles (Managing Requirements, 2010).

## Conclusiones del Capítulo

Este capítulo sirvió de base para la elaboración de un sistema donde las herramientas, metodologías y tecnologías seleccionadas constituyen la fuerza fundamental para su desarrollo exitoso. El estudio de todos los software existentes con funcionalidades similares a las requeridas por el producto a desarrollar, así como el análisis de los diferentes tipos de indicadores y su uso dentro de los procesos de gestión empresarial, ayudan a almacenar un conjunto de conocimientos que convergen en lo que será el correcto análisis y diseño del sistema.

### **Capítulo 2: Análisis y diseño del componente para la gestión de indicadores**

En este capítulo se presenta la lista de los requerimientos correspondientes al componente Gestión de Indicadores así como los Casos de Uso del Sistema y una descripción de cada uno de ellos, lo que permite dar paso al diseño donde se realizan los diagramas de clases del diseño, diagramas de interacción, diagrama de clases persistentes, así como el modelo de datos.

#### **2. Propuesta de Solución**

Se propone realizar el análisis y diseño de una aplicación Web que posibilite mejorar la gestión de los indicadores empresariales para la toma de decisiones en PATDSI, así como poder asociar los indicadores a los objetivos y dar la posibilidad de gestionar fórmulas para el cálculo de los indicadores. Dicha aplicación deberá permitir adicionar, modificar, mostrar y eliminar una serie de datos, enmarcados en diversas funcionalidades necesarias para de esta forma llevar a cabo un mejor proceso de gestión de indicadores en PATDSI.

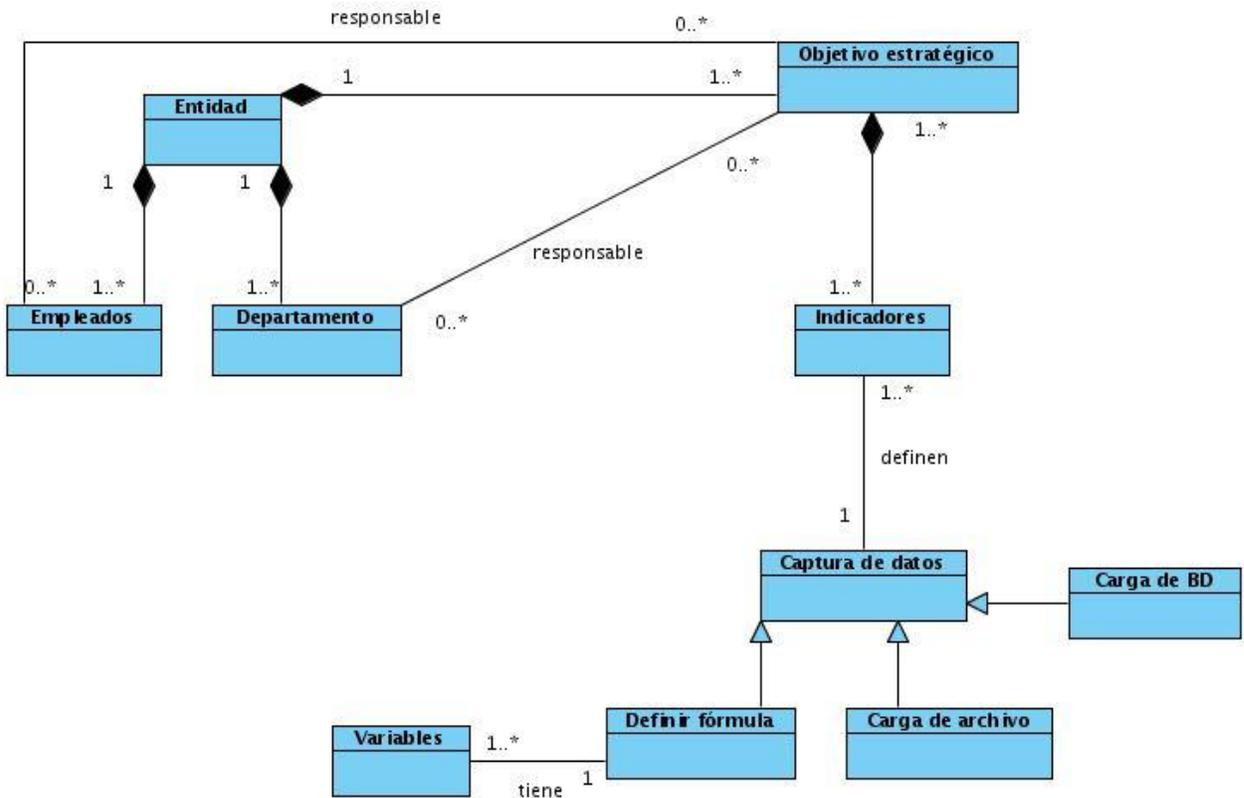
##### **2.1. Modelo del Dominio**

Un modelo de dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción. El modelo de dominio presenta uno o más diagramas de clases que no contienen conceptos propios de un sistema de software sino de la propia realidad física.

Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir (Tecnología y Synergix, 2008).

###### **2.1.1. Diagrama conceptual del modelo de dominio**

El modelo de dominio ocupa un rol protagónico en el desarrollo de software y además puede ser tomado como punto de partida para el diseño del sistema. El siguiente modelo de dominio tiene como objetivo contribuir a la comprensión de la gestión de indicadores y de los requerimientos del sistema.



*Figura 3.- Modelo de Dominio*

**Entidad:** una entidad puede ser un organismo, organizaciones, empresas, departamentos, ministerios e instituciones sociales integradas por elementos humanos, técnicos y materiales cuyo objetivo natural y principal es la obtención de utilidades.

**Empleados:** un empleado es un trabajador de la entidad, el cual puede ser responsable de objetivos estratégicos.

**Objetivos estratégicos:** son logros que la entidad quiere alcanzar en un plazo determinado para ser consciente con la orientación y propósitos estratégicos definidos en la misión.

**Indicadores:** los indicadores tienen como propósito evaluar el desempeño financiero, operacional y de producción de la entidad y están vinculados con los objetivos estratégicos.

**Departamento:** un departamento puede ser responsable de varios objetivos estratégicos.

**Captura de datos:** permite especificar la forma de obtención del valor del indicador, estas formas de captura pueden realizarse mediante: carga de BD, carga de archivo, valores digitados o definición de fórmulas.

**Carga de BD:** define a qué sistema gestor de base de datos se puede conectar y la consulta que se va a realizar para capturar los datos del indicador.

**Carga de archivo:** especifica el tipo de archivo que se debe cargar y define un formato para la captura de los datos.

**Definición de fórmulas:** permite crear fórmulas de cálculo para obtener el valor de un indicador.

**Variables:** las variables son indicadores que permiten realizar el cálculo del indicador mediante fórmulas.

### 2.2. Especificación de los Requisitos de Software

La Especificación de Requisitos Software (ERS) es una descripción completa del comportamiento del componente que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. Los casos de uso también son conocidos como requisitos funcionales. Además de los casos de uso, la ERS también contiene requisitos no funcionales. Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación. Para realizar la ERS es importante que el analista tenga la habilidad para interactuar con el cliente.

El artefacto de especificación de requisitos software debe contener una lista detallada y completa de los requisitos que debe cumplir el componente a desarrollar. El nivel de detalle de los requisitos tiene que ser claro para que el equipo de desarrollo pueda diseñar un componente que satisfaga los requisitos y los encargados de las pruebas puedan determinar si éstos se satisfacen (Agut, 2001).

En la obtención de los requerimientos se utilizaron las técnicas de introspección, discusiones y prototipos, con el fin de realizar una correcta captura de la información. Aplicando la técnica introspección a las herramientas estudiadas en el capítulo anterior se obtuvo un listado previo de 25 requerimientos, permitiendo obtener una suposición de lo que debería tener el sistema. Las discusiones del listado de requerimientos se realizaron con el equipo de desarrollo y especialistas, llegando a un acuerdo entre ambas partes de cómo realizar los procesos de estimación y presupuesto de los costos. A partir de este acuerdo se obtuvieron 6 nuevos requerimientos que se añadieron al listado actual. También se realizaron los prototipos de interfaz para mejor entendimiento de lo que se quiere obtener. Los mismos permitieron

demostrar los conceptos, así como revelar errores y omisiones que existían en los requerimientos propuestos.

### 2.2.1. Requisitos funcionales

#### *RF 1: Adicionar indicador*

El sistema debe dar la posibilidad de adicionar los indicadores, a partir de los siguientes parámetros.

➤ **Entrada:**

- Nombre de indicador.
- Código de indicador.
- Prioridad indicador (Alta, Media o Baja).
- Periodo de revisión del indicador.
- Tipo de evaluación del indicador (Cualitativo o Cuantitativo).
- Estado de disponibilidad del indicador (Activo o Inactivo).
- Unidad de medida del indicador.
- Tipo de obtención de datos del indicador.

#### *RF 2: Crear fórmula de indicador*

El sistema debe permitirle al usuario crear fórmulas de cálculo para los indicadores.

➤ **Entrada:**

- Fórmula con indicadores

#### *RF 3: Modificar indicador.*

El sistema debe dar la posibilidad de modificar los datos del indicador:

➤ **Entrada:**

- Nombre de indicador.
- Código de indicador.
- Prioridad indicador (Alta, Media o Baja).
- Periodo de revisión del indicador.
- Tipo de evaluación del indicador (Cualitativo o Cuantitativo).
- Estado de disponibilidad del indicador (Activo o Inactivo).

- Unidad de medida del indicador.
- Tipo de obtención de datos del indicador.

*RF 4: Eliminar el indicador.*

El sistema debe dar la posibilidad de eliminar el indicador.

*RF 5: Realizar consulta.*

RF 5.1 Obtener datos de los indicadores

➤ **Salida:**

- Nombre de indicador.
- Código indicador.
- Prioridad indicador (Alta, Media o Baja).
- Periodo de revisión del indicador.
- Tipo de evaluación del indicador (Cualitativo o Cuantitativo).
- Estado de disponibilidad del indicador (Activo o Inactivo).
- Unidad de medida del indicador.
- Tipo de obtención de datos del indicador.

RF 5.2 Listar indicadores

➤ **Salida:**

- Lista con indicadores

*RF 6: Configurar conexión a la base de datos*

El sistema le debe permitir al usuario configurar la conexión a la base de datos.

➤ **Entrada:**

- Datos de la conexión a la base de datos

*RF 7: Capturar datos del indicador.*

RF 7.1 Seleccionar tipo de captura de datos

➤ **Entrada:**

- Carga de archivo

RF 7.2 Seleccionar tipo de archivo que desea cargar

➤ **Entrada:**

- Archivo CSV
- Archivo XML

*RF 8: Insertar valores digitados.*

El sistema debe permitir al usuario introducir valores a los indicadores.

➤ **Entrada:**

- Datos numéricos

*RF 9: Generar reportes.*

El sistema debe permitir emitir reportes relacionados con la información almacenada en la base de datos. (Esta información se deberá obtener para cada indicador estructurado por fórmula).

➤ **Salida:**

- Reportes con fórmula del indicador

*RF 10: Asociar indicador.*

El sistema debe brindarle la posibilidad al usuario de asociar indicadores a objetivos específicos.

### 2.2.2. Requisitos no funcionales

➤ **Usabilidad**

RNF 1: Debe poseer una interfaz agradable para el cliente de acuerdo a los estándares de diseño.

RNF 2: Mostrar la información de forma lógica y correctamente estructurada.

RNF 3: Puede ser usado por cualquier persona con conocimientos básicos sobre el uso de la computadora y un ambiente Web en sentido general.

➤ **Fiabilidad**

RNF 4: El sistema se debe recuperar en un tiempo prudencial de acuerdo a la anomalía ocurrida.

RNF 5: El sistema debe hacer salvadas de la información.

➤ **Eficiencia**

RNF 6: Los tiempos de respuesta de la aplicación no deben exceder de los 20 segundos.

RNF 7: La aplicación debe soportar la conexión concurrente aproximadamente de 10 usuarios.

➤ **Hardware**

RNF 8: Se requiere de un servidor de 2 GB de RAM como mínimo y 1 GB de espacio libre en disco duro.

RNF 9: Todas las computadoras implicadas, tanto para la administración como las de los usuarios, deben estar conectadas a una red y tener al menos 1 GB de RAM.

➤ **Software**

RNF 10: Las estaciones de trabajo deben contar con:

Sistema Operativo Ubuntu 9,10

Servidor Web Apache 2

PostgreSQL 8.4

Lenguaje de script PHP 5

RNF 11: El servidor de aplicación debe contar con:

Sistema Operativo Ubuntu 9,10

Servidor Web Apache 2

PostgreSQL 8.4

Lenguaje de script PHP 5

RNF 12: El servidor de Base de Datos deberá contar como mínimo con:

Sistema Operativo Ubuntu 9,10

Servidor de base de datos PostgreSQL 8.4

➤ **Disponibilidad**

RNF 13: El sistema deberá estar disponible 24x7

➤ **Restricciones de diseño**

RNF 14: Los lenguajes de programación a utilizar son PHP y JavaScript. Para el desarrollo se utilizarán los siguientes frameworks: Symfony y ExtJs.

➤ **Requisitos para la documentación de usuarios en línea y ayuda del sistema**

RNF 15: El sistema debe contar con una ayuda que permita encontrar las funcionalidades fácilmente.

RNF 16: El sistema debe contar con un manual de usuario que permita conocer las características y las formas de funcionamiento del software.

➤ **Interfaz**

RNF 17: La aplicación contendrá distintas interfaces que servirán para el intercambio de información con la misma.

➤ **Interfaces de usuarios**

RNF 18: Las interfaces de usuario a implementar estarán orientadas a la WEB.

➤ **Interfaces de Comunicación**

RNF 19: El Sistema se comunicará con los gestores de bases de datos mediante protocolo TCP/IP, mientras que los clientes accederán al sistema vía HTTP/HTTPS.

### **2.3. Definición de los Casos de Uso**

Un CU del negocio representa a uno o varios procesos de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada CU proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. En otras palabras, un CU es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

#### **2.3.1. Definición de los actores**

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externa; con los que el negocio interactúa, un ejemplo de actor podría ser un usuario o cualquier otro sistema.

### Actores del Sistema

Actores	Descripción
Supervisor	Es el responsable de hacer el proceso de gestión con el que trabaja el sistema, además de adicionar, modificar, eliminar y visualizar los datos que se manejan desde la BD.
Digitador	Tiene acceso a un grupo menor de funcionalidades, pero cuenta con los privilegios necesarios para navegar en el sistema accediendo a los vínculos que se le definieron como accesibles, y así lograr un mayor aprovechamiento del sistema, realizando sobre el mismo una serie de operaciones básicas, como visualizar la mayor parte de la información existente y realizar reportes para ver el estado de los indicadores.
Usuario	Es un rol genérico en la aplicación que cuenta con privilegios básicos para navegar en el sistema.

*Tabla 1.- Actores del sistema*

### 2.3.2. Diagrama de casos de uso del sistema

El modelo de casos de uso está compuesto por actores, casos de uso y la relación que existe entre ellos. Representa un esquema que recoge las funcionalidades del negocio que se automatizan y determina el uso que tendrá desde el punto de vista del usuario, ya que su construcción es en base a las necesidades del mismo.

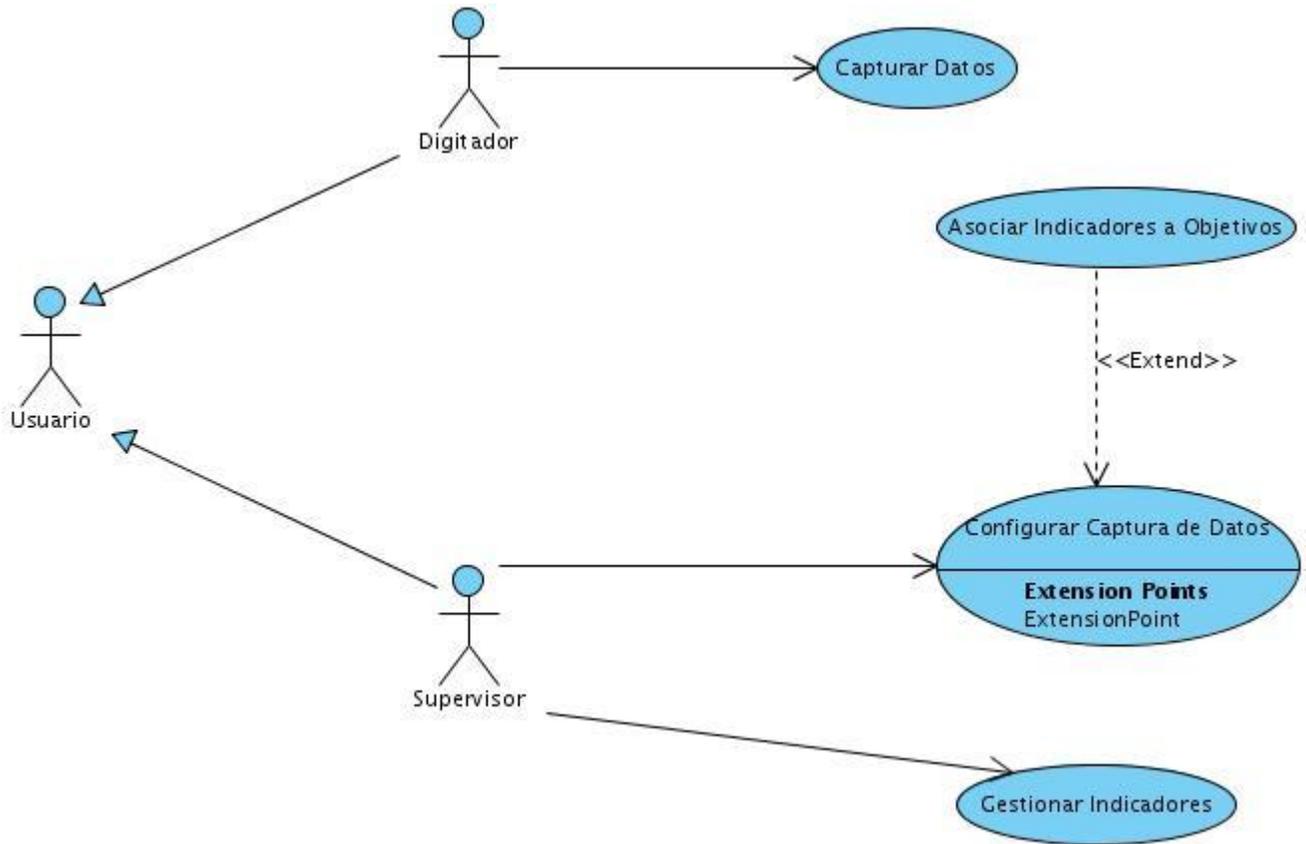


Figura 4.- Diagrama de Caso de Uso del Sistema

### 2.3.3. Descripción de los casos de uso del sistema

Los casos de uso contienen una descripción textual de todas las maneras que los actores previstos podrían trabajar con el software o el sistema.

#### 2.3.3.1. Descripción del CU Gestionar Indicador

Caso de Uso	
CU_3	Gestionar Indicadores
Propósito	El Supervisor gestionará los datos de los indicadores.
Actores	Supervisor.
Resumen	El Supervisor puede adicionar, modificar y eliminar los datos de un indicador haciendo las respectivas actualizaciones en la BD.
Pre-Condiciones	Debe existir al menos una Entidad y esta debe tener al menos un objetivo registrado en la BD.

<b>Post-Condiciones</b>	Se actualiza la BD.
<b>Referencias</b>	RF 1, RF 3, RF 4
<b>Prioridad</b>	Crítico
<b>Sección 1: Adicionar indicador</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El Supervisor selecciona la opción de adicionar Indicador.	1.1 La aplicación muestra una interfaz para que el Supervisor introduzca los datos del indicador.
2 El Supervisor introduce en el sistema los datos del Indicador.	2.1 La aplicación comprueba que los datos estén escritos correctamente mediante la validación de los campos.
3 El Supervisor selecciona el botón Aceptar.	3.1 Se actualiza la BD y se actualiza el listado de indicadores con el nuevo indicador adicionado.
<b>Flujos alternativos</b>	
<b>Acción del Actores</b>	<b>Respuesta del Sistema</b>
	2.2 El sistema valida y si los datos introducidos están escritos incorrectamente, regresa al paso 1.1. 3.2 Si el indicador ya existe en la BD el sistema muestra un mensaje de error.
<b>Sección 2: Eliminar Indicador.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
4 El Supervisor selecciona el Indicador que desea eliminar de la lista de indicadores.	4.1 El sistema carga la información del indicador seleccionado.
5 El Supervisor selecciona la opción Eliminar Indicador.	5.1 El sistema elimina el indicador seleccionado y se actualiza la BD. 5.2 El sistema muestra el listado de indicadores actualizado.
<b>Sección 3: Modificar Indicador</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
6 El Supervisor selecciona de la lista de indicadores el indicador que desea modificar.	6.1 El sistema muestra en la interfaz los datos del indicador antes de ser modificados.
7 El Supervisor modifica los datos y selecciona la opción Modificar Indicador.	7.1 El sistema actualiza el indicador con los nuevos cambios en la BD y en el listado de indicadores existentes.
<b>Flujos alternativos</b>	
<b>Acción del Actores</b>	<b>Respuesta del Sistema</b>
	7.4 El sistema valida y si los datos introducidos están escritos incorrectamente, regresa al paso 6.1.
<b>Prototipo de Interfaz de Usuario</b>	

**SIGI**

Asistente de indicadores

Nombre:

Código:

Periodicidad:

Unidad de medida:

Prioridad:

Obtención de datos:

Cargar de BD  Cargar de Archivo

Crear fórmula

Tipo:

Estado:

Aceptar

Cancelar

Listado de indicadores

Código	Nombre	UM	Periodicidad	Tipo	Estado

Modificar

Eliminar

Asociar a objetivos

Tabla 2.- Descripción del CU Gestionar Indicador

Para ver las descripciones textuales de los restantes casos de uso ver (Anexo 1: Descripción textual de los Casos de Uso del negocio).

## 2.4. Modelo de Análisis

El modelo de análisis ayuda a refinar los requisitos y permite razonar sobre los aspectos internos del sistema y sus recursos compartidos internos, también ofrece un mayor poder expresivo y una mayor formalización ayudando a estructurar los requisitos y proporcionando una arquitectura centrada, no sólo en el mantenimiento, sino que también es utilizada como entrada en las actividades de diseño y de implementación.

### 2.4.1. Diagrama de clases del Análisis

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema, esboza cómo llevar a cabo la funcionalidad dentro del sistema incluida la funcionalidad significativa para la arquitectura y sirve como una primera aproximación del diseño.

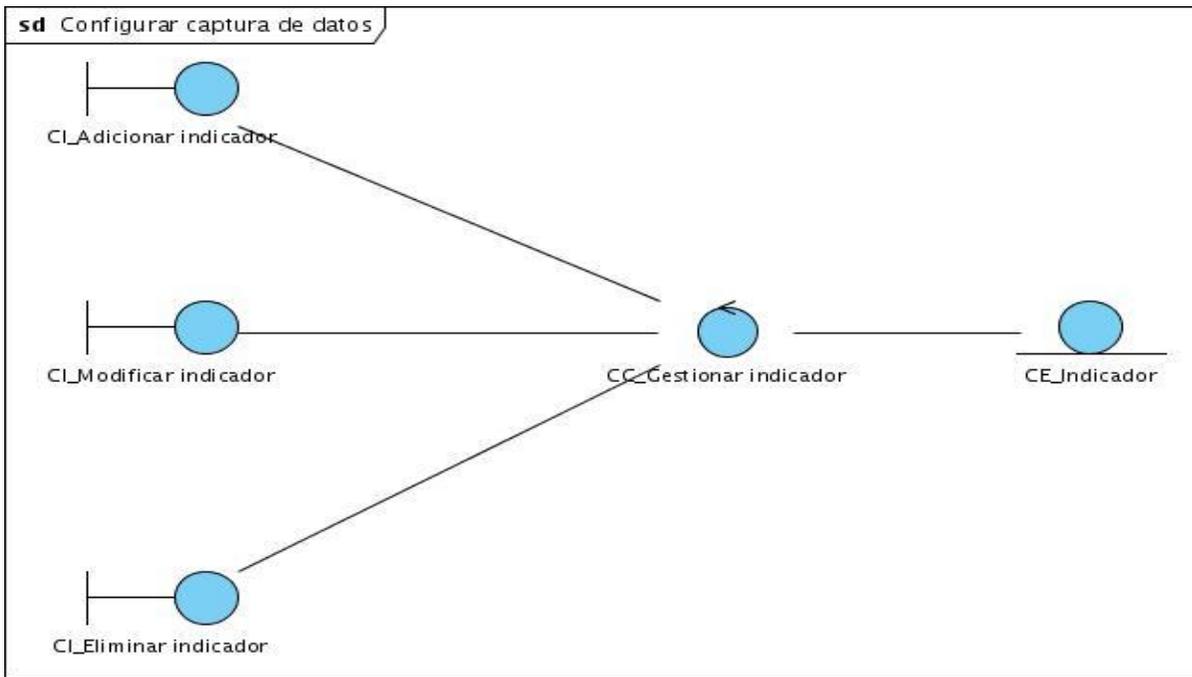
2.4.1.1. Clases del Análisis

Casos de Uso	Clase Interfaz	Clase Control	Clase Entidad
Capturar Datos	CI_Reporte CI_Fórmula CI_Carga de BD CI_Carga de Archivo	CC_Captura de Datos	CE_Indicadores
Configurar Captura de Datos	CI_Obtener Datos CI_Conexión BD CI_Crear Fórmula CI_Configurar Cargad de Archivo	CC_Configurar Captura de Datos	CE_Indicadores
Gestionar Indicador	CI_Añadir Indicador CI_Modificar Indicador CI_Eliminar Indicador	CC_Gestionar Indicador	CE_Indicadores
Asociar a objetivos	CI_Añadir indicador CI_Lista de objetivos	CC_Datos de indicador	CE_Indicadores

Tabla 3.- Identificar clases del análisis

Para ver la Descripción de las clases del análisis ver (Anexo 7: Descripción de las clases de análisis)

**2.4.1.2. Diagrama de clases del análisis: Gestionar Indicador**



*Figura 5.- Diagrama de clases del análisis: Gestionar Indicador*

Para ver los diagramas de clases del análisis correspondiente a los demás casos de uso ver (*Anexo 2: Diagrama de clases del análisis*).

**2.4.2. Diagramas de Colaboración**

Los diagramas de colaboración representan las interacciones entre objetos, mostrando el contexto de las operaciones y de los ciclos en la ejecución.

2.4.2.1. Diagrama de colaboración: Caso de Uso Gestionar Indicador

Adicionar indicador

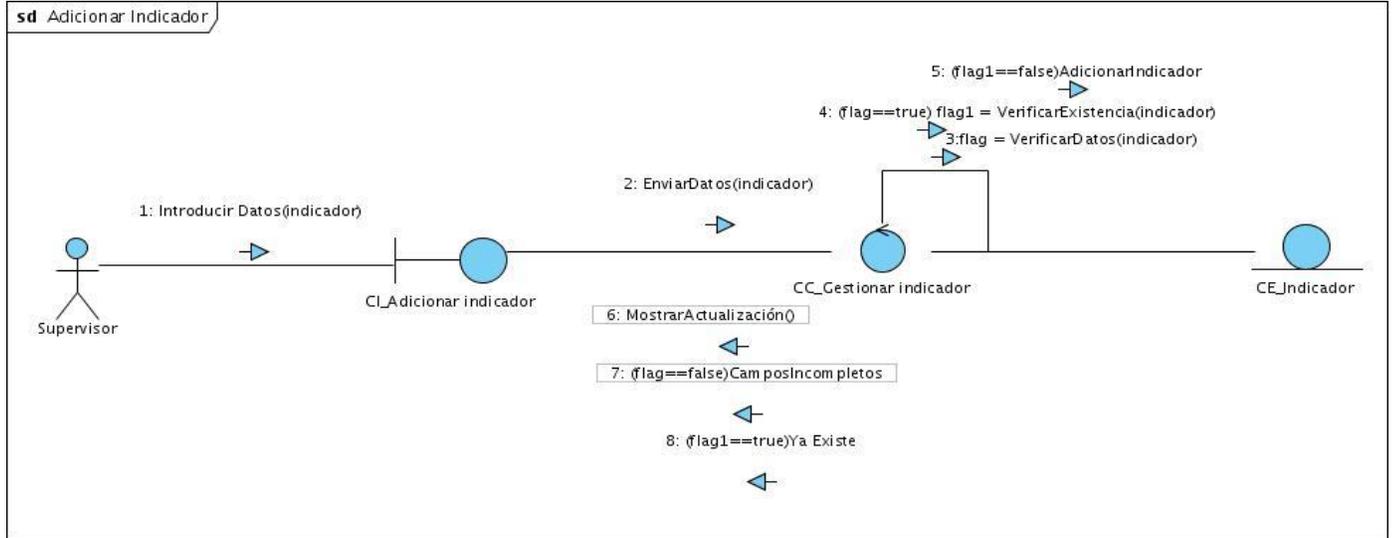


Figura 6.- Diagrama de colaboración: Adicionar indicador

Modificar indicador

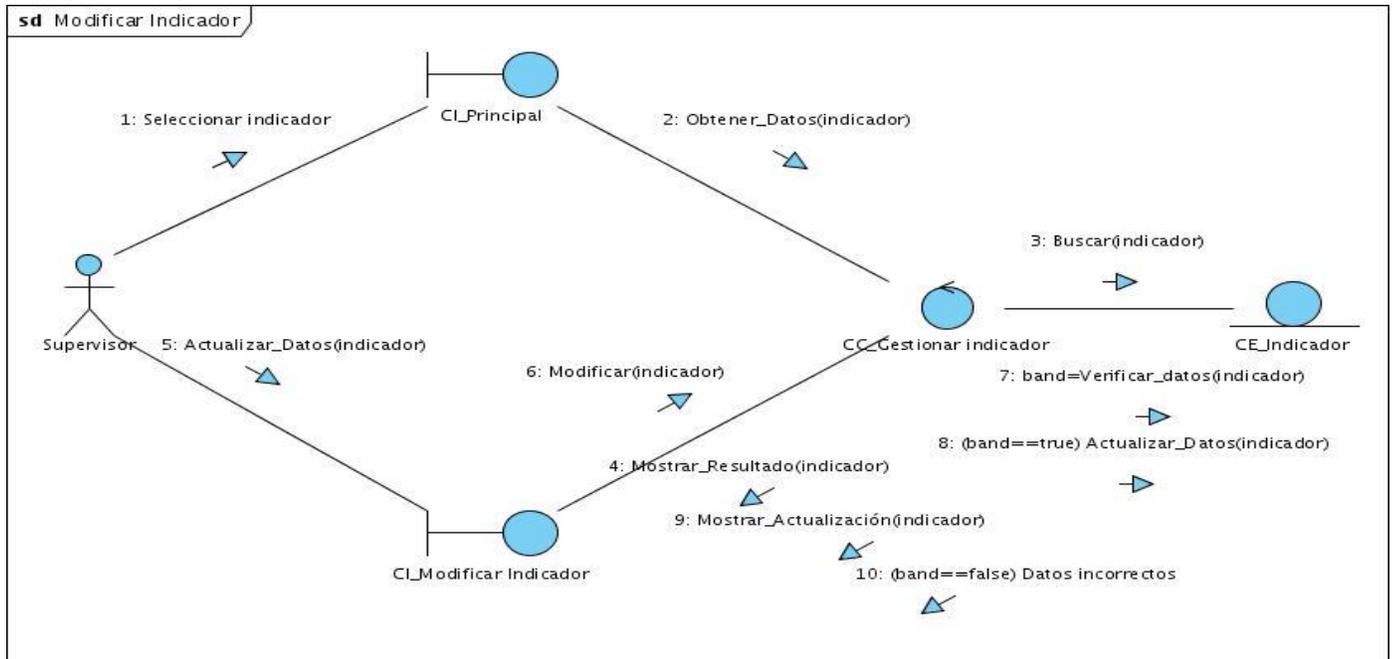


Figura 7.- Diagrama de colaboración: Modificar indicador

**Eliminar indicador**

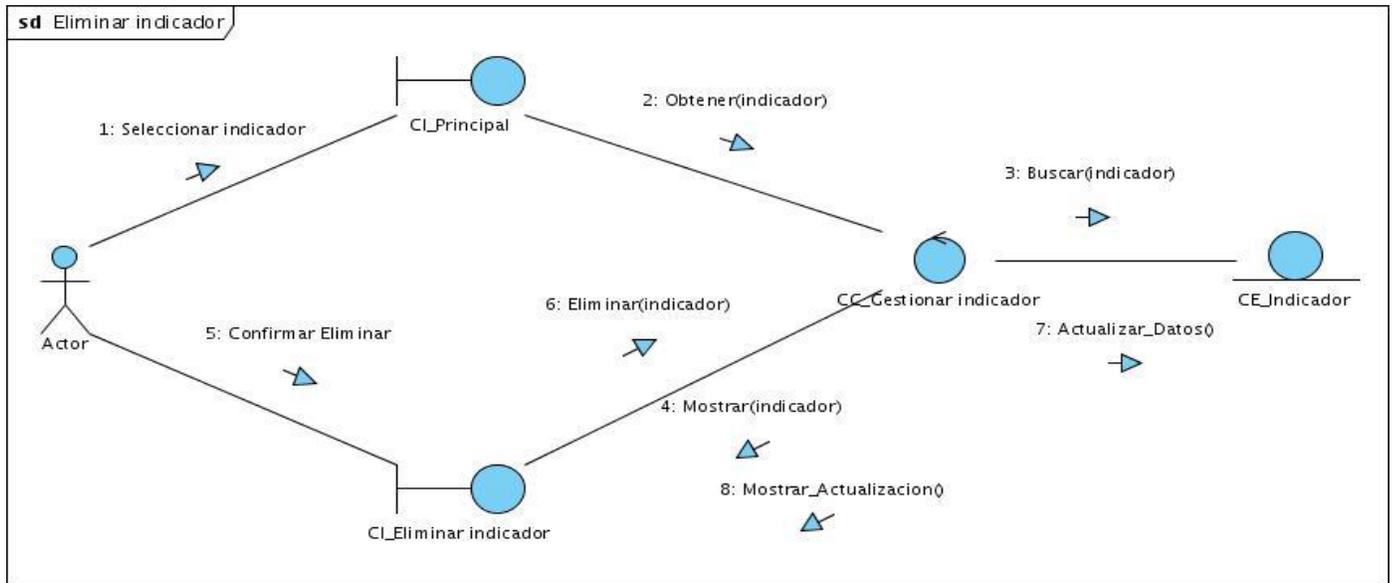


Figura 8.- Diagrama de colaboración: Eliminar indicador

Para ver los diagramas de colaboración correspondientes a los restantes casos de uso ver (Anexo 3: Diagramas de colaboración).

**2.5. Diseño del Sistema**

El diseño del sistema se encarga de desarrollar las reglas propuestas durante el análisis en función de satisfacer los objetivos planteados, tanto desde el punto de vista funcional como del no funcional.

Para hacer un diseño eficiente se tomarán en cuenta un conjunto de patrones, que permiten dar solución a problemas a través de la codificación del conocimiento y principios existentes, facilitando notablemente el trabajo posterior.

**2.5.1. Patrones utilizados en el desarrollo del sistema.**

La utilización de los patrones ha tomado gran auge a partir del desarrollo del modelo orientado a objetos. Una arquitectura orientada a objetos bien estructurada está llena de patrones y la calidad de ellos se mide por la atención que los diseñadores han prestado a las colaboraciones entre sus objetos. Los patrones conducen a arquitecturas más pequeñas, más simples y más comprensibles.

### 2.5.1.1. Patrones de Asignación de Responsabilidades

Los Patrones de Asignación de Responsabilidades (GRASP) describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades) (Departamento de Lenguajes y Sistemas Informáticos, 2008).

- Experto: asigna una responsabilidad al experto la clase que cuenta con la información necesaria para cumplir la responsabilidad.
- Creador: asigna a la clase B la responsabilidad de crear una instancia de la clase A.
- Bajo Acoplamiento: este patrón plantea que las clases deben comunicarse con un pequeño grupo de clases para disminuir las dependencias de las clases y sea más fácil el mantenimiento de las mismas.
- Patrón Alta Cohesión: una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas.
- Controlador: este patrón describe el uso de algunas clases dedicadas a capturar los mensajes de los eventos del sistema y aconseja el uso de una misma clase controladora para controlar los eventos del sistema de un mismo caso de uso.

### 2.5.1.2. Patrones del diseño

Estos patrones convierten el modelo generado en una estructura altamente flexible que presenta, entre otras ventajas, una mayor escalabilidad y portabilidad.

- Modelo-Vista-Controlador: Se usa para separar los datos de una aplicación, la interfaz de usuario, y la lógica de control en componentes más manejables, permitiendo con ello un mejor mantenimiento del sistema.
- Active Record: Usado para acceder a los datos de la base de datos. Una fila en la tabla se envuelve en una clase que implementa métodos de acceso para cada columna de la tabla, de manera que se asocian filas únicas de la base de datos con objetos del lenguaje de programación usado.

- Data Access Object (DAO): Usado para acceder a la fuente de datos y encapsular los objetos clientes, ocultando tanto la fuente como el modo de acceder a ella. Los DAOs deben implementar los métodos del interface (InterfaceDAO) que declaran. Pero además pueden implementar otros métodos que no están en la interfaz.

### 2.5.2. Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Normalmente, contiene la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de atributos.
- Navegabilidad.
- Dependencias.

A continuación se mostrarán los diagramas de clases del diseño utilizados en la realización de este trabajo.

2.5.2.1. Diagrama de Clases del Diseño: Gestionar Indicador

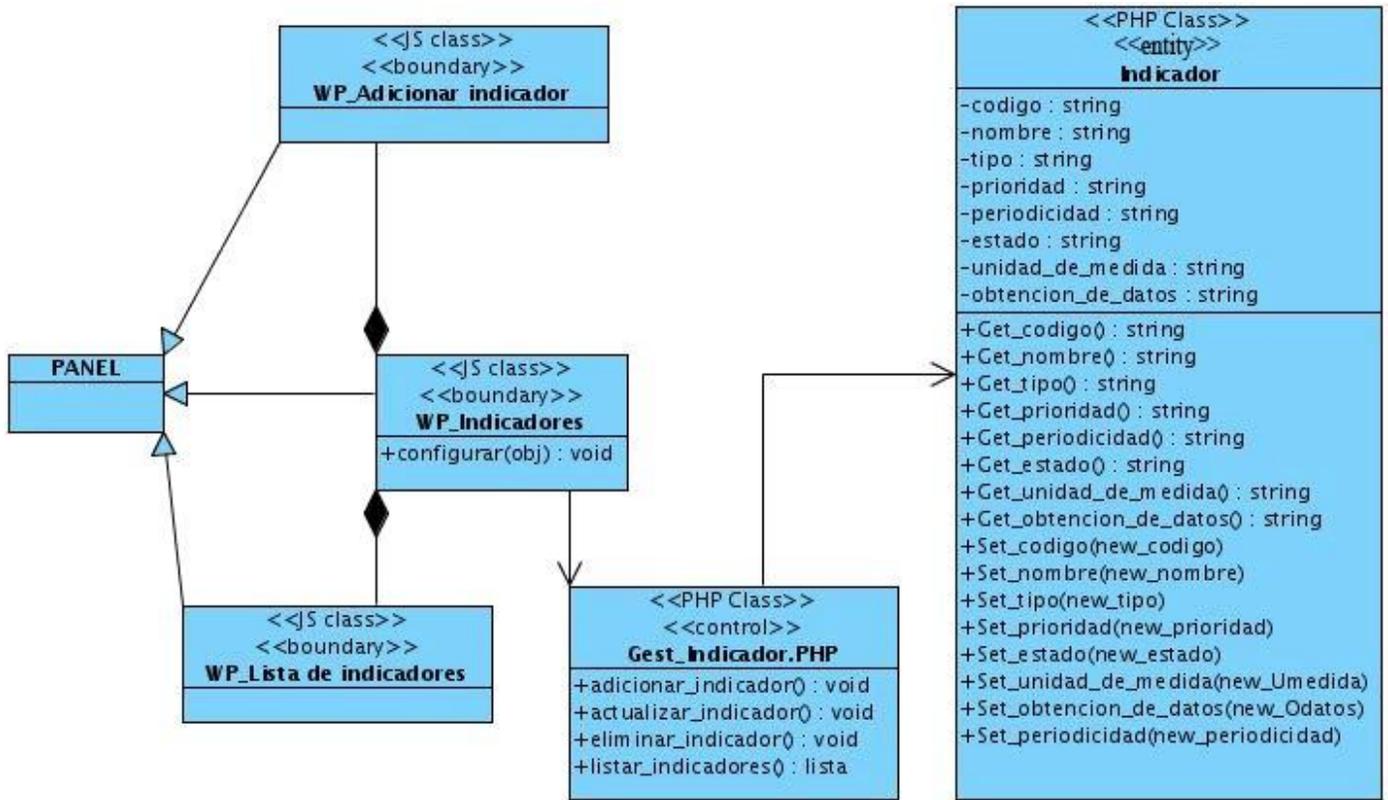


Figura 9.- Diagrama de Clases del Diseño: Gestionar Indicador

Para ver los diagramas de clases de diseño correspondientes a los restantes casos de uso ver (Anexo 4: Diagramas de Clases del Diseño).

2.5.3. Diagramas de Secuencia

Los diagramas de secuencia se utilizan para ilustrar la realización de un CU. Son particularmente importantes para los diseñadores pues aclaran los roles jugados por los objetos en un flujo, lo cual le proporciona un gran valor para la determinación de las responsabilidades de las clases. A diferencia del Diagrama de Colaboración, este incluye secuencia cronológica de los mensajes y no la relación entre los objetos, por lo que es mejor su utilización cuando el orden en el tiempo de los mensajes es de importancia.

2.5.3.1. Diagramas de Secuencia del CU Gestionar Indicador

Adicionar Indicador

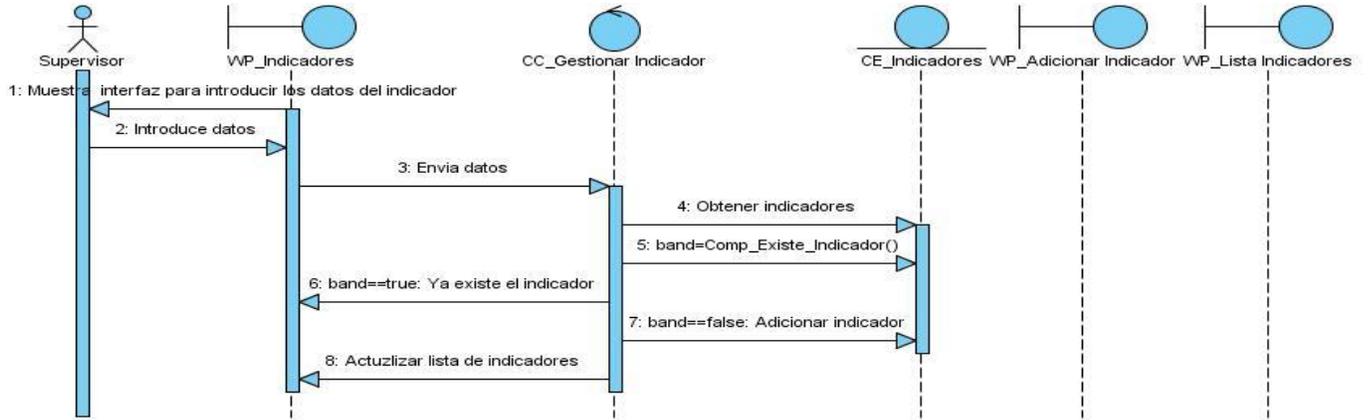


Figura 10.- Diagramas de Secuencia: Adicionar Indicador

Modificar Indicador

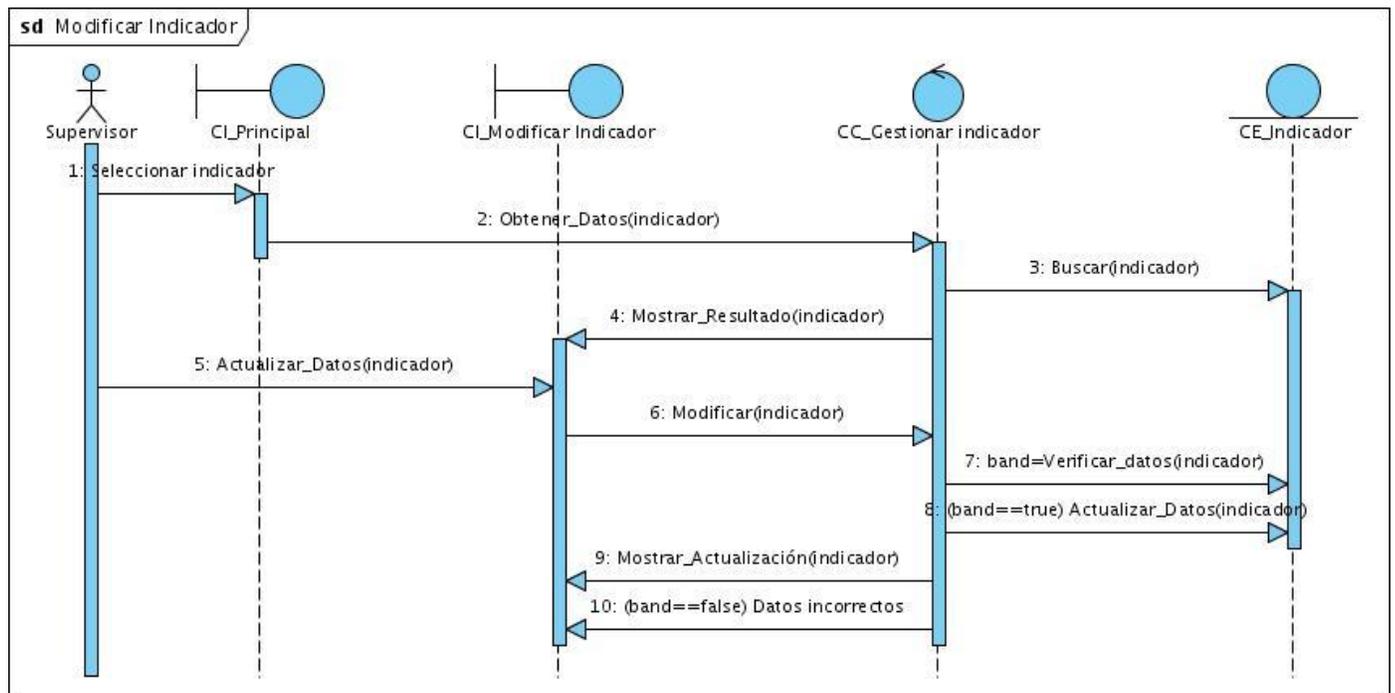


Figura 11.- Diagramas de Secuencia: Modificar Indicador

**Eliminar Indicador**

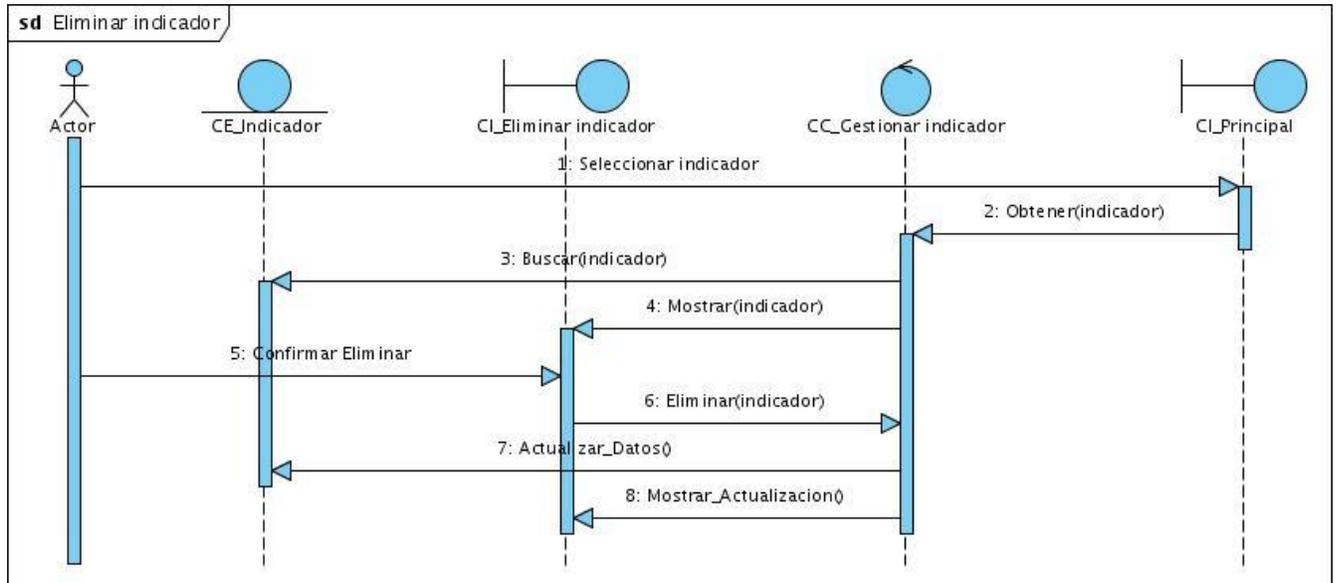


Figura 12.- Diagramas de Secuencia: Eliminar Indicador

Para ver los diagramas de secuencias correspondientes a los restantes casos de uso ver (Anexo 5: *Diagramas de Secuencia*)

**2.5.4. Diagrama de Clases Persistentes**

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. El diagrama de clases persistentes describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Generalmente, las clases persistentes tienen como origen las clases clasificadas como entidad porque ellas modelan la información del sistema y el comportamiento asociado de algún fenómeno o concepto (Pérez Castillo & Valiente Mesa, 2009).

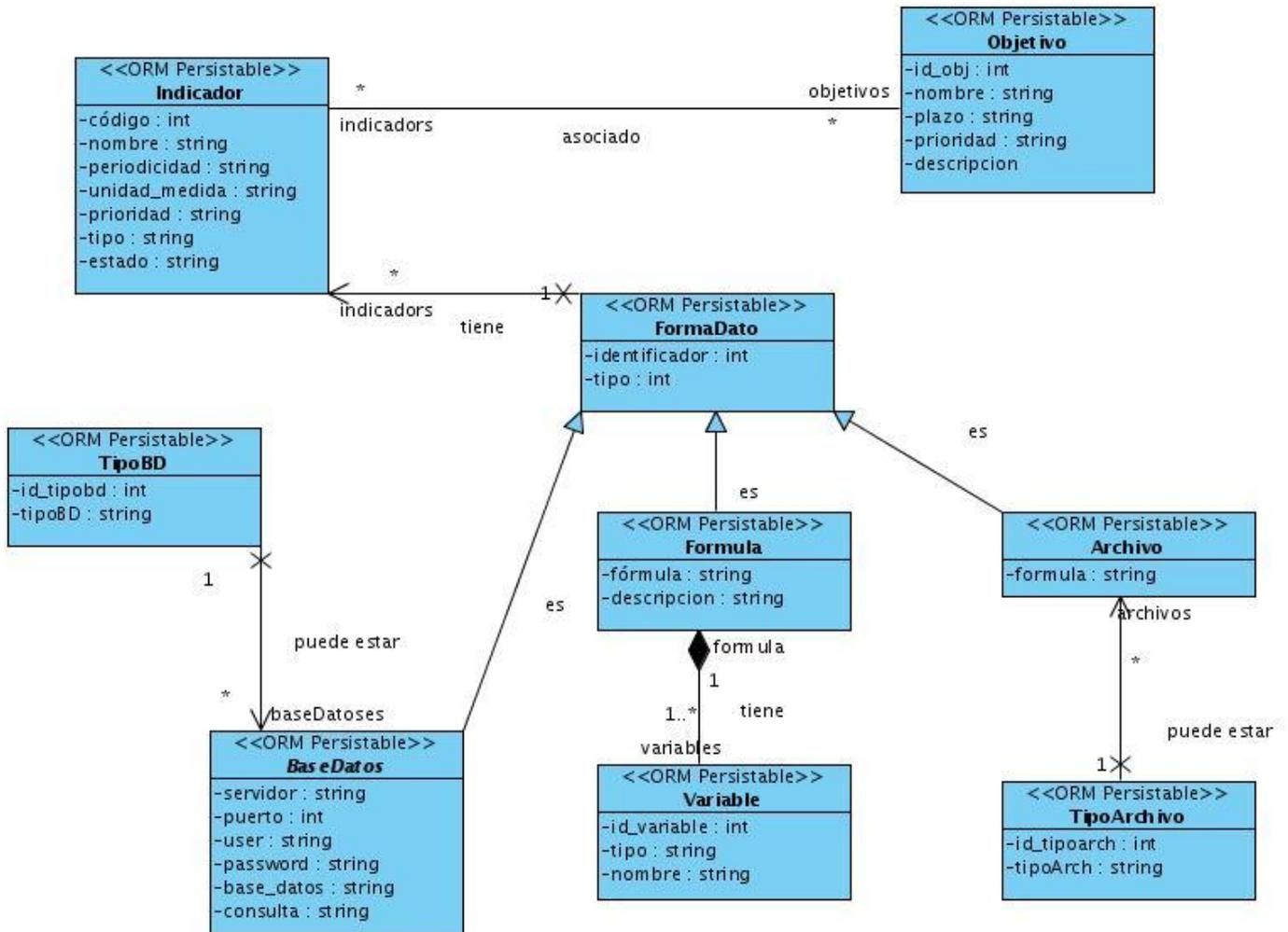


Figura 13.- Diagrama de Clases Persistentes

### 2.5.5. Modelo de Datos

El modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Por lo general, un modelo de datos permite describir el tipo de datos que incluye la base y la forma en que se relacionan, así como las restricciones de integridad y las operaciones de manipulación de los datos. El modelo de datos describe la representación lógica y física de los datos persistentes.

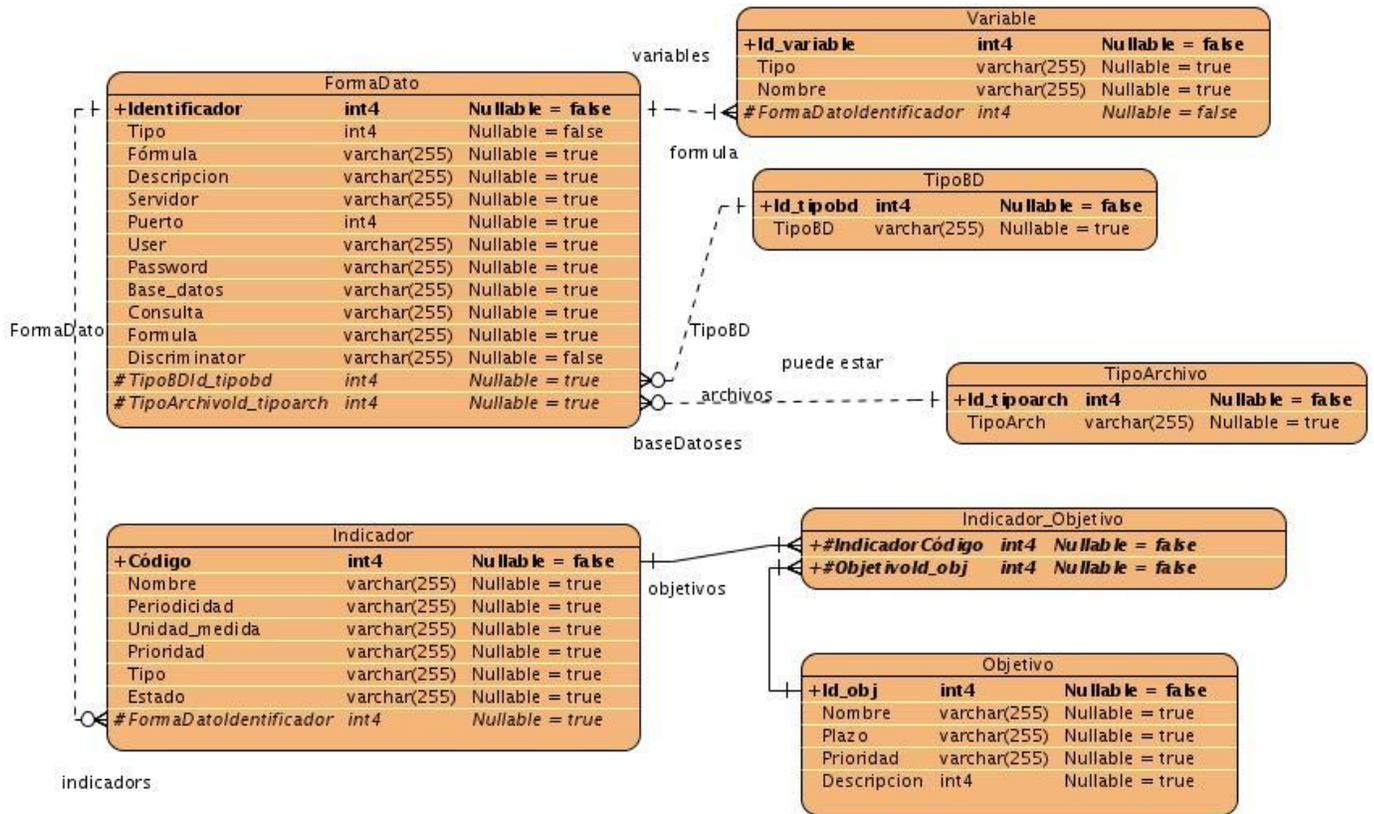


Figura 14.- Diagrama de Modelo de Datos

## Conclusiones del Capítulo

En este capítulo se obtuvo un mayor entendimiento del proceso de negocio, definiendo así los requisitos funcionales y no funcionales que permitieron identificar las funcionalidades con las que contará el sistema, las cuales darán respuesta a las necesidades del problema. Los casos de uso fueron relacionados mediante un diagrama de casos de uso del sistema y se les realizó una descripción detallada logrando así un mayor acercamiento a lo que el sistema debería cumplir. En este capítulo se generaron además, diferentes diagramas de clases del diseño, como por ejemplo los diagramas de secuencia, los cuales se realizaron para cada uno de los escenarios del sistema, brindando una panorámica de cómo el usuario interactúa con el mismo, de igual manera se describieron las tablas de la base de datos y se obtuvieron los diagramas de clases persistentes y de modelo de datos.

## Capítulo 3: Validación de los requisitos y diseño

En el presente capítulo se realizará la validación de los requisitos y el diseño, para de esta forma mostrar que los requisitos definen el sistema que realmente el usuario desea y que el diseño realizado cumple con los patrones utilizados. Para la validación de los requisitos se utilizan casos de pruebas, matriz de trazabilidad, listas de chequeo, y métricas de validación, para la validación del diseño se utilizarán métricas, todas estas definiciones fueron explicadas en el Capítulo 1.

### 3.1. Validación de requisitos

La validación es la etapa final de la ingeniería de requerimientos. Su objetivo es verificar todos los requerimientos que aparecen en el documento especificado para asegurarse que representan una descripción aceptable del sistema que se debe implementar. Esto implica verificar que los requerimientos sean consistentes y que estén completos.

#### 3.1.1. Métricas de la Calidad de Especificación

La métrica es el resultado de dividir el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas (**R<sub>i</sub>**) entre la cantidad de requisitos de software (**R<sub>t</sub>**).

Para determinar la especificidad de los requisitos, se sugiere una métrica basada en la consistencia de la interpretación de los revisores para cada requisito.

Cuanto más cerca de uno esté el valor de Q1 menor será la ambigüedad de la especificación (catarina.udlap.mx, 2008):

$$Q1 = R_i / R_t$$

$$R_t = r_f + r_{nf}$$

$$R_t = 12 + 19$$

$$R_t = 31$$

Donde  $r_f$  es: Cantidad de requisitos funcionales.

Donde  $r_{nf}$  es: Cantidad de requisitos no funcionales.

En el presente trabajo de diploma se realizaron dos revisiones para llegar a obtener un buen análisis sin ambigüedad en los requisitos, todos los revisores tuvieron una interpretación bastante parecida de los requisitos, de las cuales se obtuvieron los siguientes resultados:

## CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO

---

Revisión 1: Durante esta revisión se detectaron dos requisitos que no estaban claramente definidos. Por lo que de los 29 requisitos existentes, los revisores tuvieron interpretaciones bastante parecidas en 27.

Los resultados obtenidos al aplicar la métrica en la primera revisión se muestran a continuación:

$$\begin{aligned}R_t &= r_f + r_{nf} \\R_t &= 10 + 19 \\R_t &= 29\end{aligned}$$

$$\begin{aligned}Q_1 &= 27/29 \\Q_1 &= 0,93\end{aligned}$$

Revisión 2: En esta revisión ya se habían solucionado los problemas detectados en la revisión anterior, se adicionaron otros dos requisitos para que quedaran reflejadas todas las necesidades del cliente, por lo que ya se cuenta con 12 requisitos funcionales lo cual hace que los revisores hayan tenido interpretaciones iguales para los 31 requisitos.

$$\begin{aligned}R_t &= r_f + r_{nf} \\R_t &= 12 + 19 \\R_t &= 31\end{aligned}$$

$$\begin{aligned}Q_1 &= 31/31 \\Q_1 &= 1\end{aligned}$$

Luego de haber realizado la segunda revisión se llega a la conclusión de que los requisitos no presentan ya ningún nivel de ambigüedad, por lo que se verifica que estos cumplan con una serie de atributos o características necesarias para la especificación de requisitos.

Para que la Especificación de Requisitos sea reconocida de manera satisfactoria debe cumplirse que:

La correctitud sea menor 0,10

La completitud sea mayor que 0,90

La consistencia sea menor que 0,20

### **Correctitud**

La especificación de requisitos es correcta si todos los requisitos que figuran en ella reflejan alguna necesidad real. Esta métrica se propone determinar si la especificación de requisitos contiene todos los requisitos necesarios para satisfacer las necesidades del negocio y los interesados.

X = Correctitud

## CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO

---

D = Total de observaciones de negocio realizadas en la validación

T = Total de requisitos revisados

$$X = D/T$$

$$D = 1$$

$$T = 12$$

$$X = 1/12$$

$$\mathbf{X = 0,08}$$

### Compleitud

La métrica pretende determinar si la especificación de requisitos es completa. Una especificación de requisitos está completa si los requisitos están relacionados a la funcionalidad, el desarrollo, las restricciones del diseño, los atributos y las interfaces externas.

X = Compleitud

O = Total de observaciones de formato

S = Total de secciones del documento

$$\mathbf{X = 1 - O/S}$$

$$O = 0$$

$$S = 0$$

$$X = 1 - (0/0)$$

$$\mathbf{X = 1}$$

### Consistencia

La métrica se propone determinar si la especificación de requisitos es consistente. Una especificación de requisitos es consistente si ningún conjunto de requisitos descritos en ella son contradictorios o entran en conflicto.

X = Consistencia

C = Total de observaciones de consistencia

T = Total de requisitos revisados

$$\mathbf{X = C/T}$$

$$C = 2$$

$$T = 12$$

$$X = 2/12$$

**X = 0,17**

Atributo	Descripción	Cumplimiento (%)
Correcta	La ERS es correcta si contiene todos los requisitos que el software debe satisfacer.	El 100% ERS son correctas.
No Ambigua	La ERS es no ambigua si cada requisito tiene una única interpretación.	El 100 % de las ERS son no ambiguas.
Completa	La ERS es completa si están incluidas todas las funciones requeridas por el cliente.	El 100 % de las ERS son completas.
Consistente	La ERS es consistente si ningún subconjunto de la misma entra en contradicción con otro subconjunto.	El 100% de las ERS son consistentes.
Clara	La ERS es clara si está escrita en un lenguaje apropiado.	El 100 % de las ERS son claras.

*Tabla 4.- Características necesarias para la especificación de requisitos*

En este epígrafe se realiza un proceso de validación de requisitos mediante métricas, lo cual es bastante satisfactorio para el presente trabajo ya que ayuda a que el desarrollo del software no tenga que atrasarse por algún desperfecto en los requisitos.

Luego de obtener los resultados al aplicarle la métrica a los requisitos funcionales definidos, se puede llegar a la conclusión de que la especificación de los requisitos puede ser aceptada satisfactoriamente, ya que es clara, no presenta ambigüedades y tienen una buena Correctitud, Completitud y Consistencia.

### **3.1.2. Matriz de trazabilidad**

La matriz de trazabilidad es otra técnica empleada para validar los requerimientos identificados, esta técnica consiste en verificar que los casos de uso que se tienen satisfagan todos los requerimientos del sistema, específicamente para verificar que no existan inconsistencias en el proceso de incluir los requisitos en los Casos de Uso. A continuación se muestra una matriz donde todos los requisitos funcionales del sistema se encuentran ubicados de forma vertical y los casos de uso del sistema de forma horizontal, de manera que puedan hacerse corresponder entre ellos.

## CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO

CU RF	Gestionar Indicadores	Configurar captura de datos	Capturar datos	Asociar a objetivos
Adicionar indicadores	x			
Crear fórmula de indicador		x		
Modificar indicador	x			
Eliminar el indicador	x			
Obtener datos de los indicadores			x	
Listar Indicadores			x	
Configurar conexión a la base de datos		x		
Seleccionar tipo de archivo que desea cargar		x	x	
Seleccionar tipo de captura de datos		x		
Insertar valores digitados			x	
Generar reportes			x	
Asociar indicador				x

*Tabla 5.- Matriz de trazabilidad*

Como se puede apreciar en la tabla anterior, todos los requisitos identificados están relacionados con casos de uso del sistema, de esta forma luego de terminado el proyecto no quedará ningún requisito sin implementarse evitando así que exista inconsistencia.

## CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO

### 3.1.3. Lista de Chequeo para validar el listado de requisitos

A continuación se muestra la lista de chequeo aplicada al listado de los requisitos con el objetivo de verificar que estén correctos y no presenten inconsistencias.

Indicadores a evaluar	Nivel de importancia	Procedimiento	Respuesta
1. ¿Se refleja el seguimiento del documento?	B	Revisión de Documento	Sí
2. ¿Se especifica el número de cada requisito evaluado?	A	Revisión de Documento	Sí
3. ¿Se especifica el nombre de cada Requisito Solicitado?	A	Revisión de Documento	Sí
4. ¿Se especifica si están identificados los elementos de entrada de cada requisito?	A	Revisión de Documento	Sí
5. ¿Se especifica si están identificados los elementos de salida de cada requisito?	A	Revisión de Documento	Sí
6. ¿Se especifica si cada requisito no es ambiguo?	A	Revisión de Documento	Sí
7. ¿Se especifica si cada requisito está correcto?	A	Revisión de Documento	Sí
8. ¿El requisito puede ser verificado?	A	Revisión de Documento	Sí

Tabla 6.- Lista de Chequeo para validar el listado de requisitos

## 3.2. Pruebas

El propósito de un caso de prueba es especificar una forma de probar el sistema, incluyendo las entradas con las que se ha de probar, los resultados esperados y las condiciones bajo las que ha de probarse. Los casos de pruebas ayudan a validar y verificar las expectativas de los interesados.

### 3.2.1. Pruebas de sistemas

Son las encargadas de probar el sistema como un todo. Están basadas en los requerimientos generales y abarca todas las partes combinadas del sistema.

#### 3.2.1.1. Caso de Prueba: Caso de uso Gestionar Indicadores

##### Descripción General

Este Caso de Uso: Adiciona, modifica y elimina los datos de un indicador.

##### Condiciones de Ejecución

1. Debe Existir al menos una Entidad y esta tener al menos un Objetivo registrado en la BD.
2. A excepción del Adicionar Indicador en los restantes dos escenarios, los indicadores deberán estar previamente listados.

##### Secciones a probar en el Caso de Uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Adicionar indicador	EC 1.1: Agregar Indicador	Crea un Indicador en la Base de Datos
	EC 1.2: Ausencia o error en datos de entrada	Verifica si se introdujeron todos los datos del Indicador a Agregar o si tienen errores.
	EC 1.3: Indicador ya existe	Verifica la existencia del indicador en la Base de Datos
SC 2: Modificar Indicador	EC 2.1: Modificar Indicador	Modifica el Indicador en la Base de Datos
	EC 2.2: Ausencia o error en datos de entrada	Verifica si se introdujeron todos los datos del Indicador a modificar o si tienen errores.
SC 3: Eliminar Indicador	EC 3.1: Eliminar Indicador	Elimina un Indicador en la Base de Datos

*Tabla 7.- Secciones a probar en el Caso de Uso: Gestionar Indicador*

## CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO

### Descripción de variable

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	Nombre del Indicador	campo de texto	No.	Admite los siguientes caracteres: a-z, A-Z.
[2]	Código del Indicador	campo de texto	No.	El campo admite letras y números, las letras pueden estar escritas tanto en mayúscula como en minúscula.
[3]	Obtención de datos	radiobutton	No.	si/no

*Tabla 8.- Descripción de variable. CU Gestionar Indicador*

### Matriz de Datos

#### SC 1 Adicionar indicador

Escenario	Variable 1 <i>Nombre del Indicador</i>	Variable 2 <i>Código del Indicador</i>	Variable 3 <i>Obtención de datos</i>	Respuesta Esperada	Resultado de la Prueba	Flujo Central
EC 1.1 Agregar Indicador	V	V	V	El sistema almacena los datos en la Base de Datos. Se actualiza además el listado de indicadores existentes.		1. Seleccionar el menú "Módulos": "Gestión de Indicadores"
EC 1.2 Ausencia o error en datos de entrada	V	I	Vacío	El sistema verifica la existencia de los datos en los campos de texto y también verifica que sean correctos, muestra un mensaje notificando que debe ser especificados valores válidos.		2. Seleccionar "Gestionar Indicadores"
EC 1.3 Indicador ya existe	V	V	V	El sistema verifica la existencia del Indicador que se quiere agregar en la Base de Datos y muestra un mensaje		3. Marcar la opción "Adicionar Indicador"
						4. Se introducen los datos del nuevo indicador.
						5. Seleccionar el botón "Aceptar" para concluir la operación.

## CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO

				de error.	
--	--	--	--	-----------	--

*Tabla 9.- Matriz de Datos. Adicionar indicador*

### SC 2 Modificar Indicador

Escenario	Variable 1 <i>Nombre del Indicador</i>	Variable 3 <i>Obtención de datos</i>	Respuesta Esperada	Resultado de la Prueba	Flujo Central
EC 2.1 Modificar Indicador	V	V	El sistema almacena los nuevos datos del indicador modificado. Se actualiza además el listado de indicadores existentes.		<ol style="list-style-type: none"> <li>1. Seleccionar el menú "Módulos": "Gestión de Indicadores"</li> <li>2. Seleccionar "Gestionar Indicadores"</li> <li>3. Marcar, del listado de indicadores existentes, el indicador que se desee modificar y seleccionar el botón "Modificar".</li> <li>4. Se actualizan los datos del indicador seleccionado.</li> <li>5. Seleccionar el botón "Aceptar" para concluir la operación</li> </ol>
EC 2.2 Ausencia o error en datos de entrada	I	Vacio	El sistema verifica la existencia de los datos en los campos de texto y también verifica que sean correctos, muestra un mensaje notificando que debe ser especificados valores válidos.		

*Tabla 10.- Matriz de Datos. Modificar indicador*

### SC 3 Eliminar Indicador

Escenario	Variable 1 <i>Nombre del Indicador</i>	Variable 2 <i>Código del Indicador</i>	Variable 3 <i>Obtención de datos</i>	Respuesta Esperada	Resultado de la Prueba	Flujo Central
EC 3.1 Eliminar Indicador	V	V	V	Se actualiza la Base de Datos y el listado de indicadores existentes.		<ol style="list-style-type: none"> <li>1. Seleccionar el menú "Módulos": "Gestión de Indicadores"</li> <li>2. Seleccionar "Gestionar Indicadores"</li> </ol>

## CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO

						<p>3. Marcar, del listado de indicadores existentes, el indicador que se desee eliminar.</p> <p>4. Seleccionar el botón "Eliminar" para concluir la operación.</p>
--	--	--	--	--	--	--

*Tabla 11.- Matriz de Datos. Eliminar indicador*

### 3.3. Métricas para la validación del diseño

Para la validación de los patrones Alta Cohesión y Bajo Acoplamiento se utilizaron métricas que permitieron demostrar que los mismos fueron bien utilizados.

**Tamaño operacional de clase (TOC):** Número de métodos. Responsabilidad, aumenta para TOC+, Complejidad de implementación, aumenta TOC+, Reutilización, disminuye TOC+.

**Relaciones entre clases (bajo) (RC):** Número de relaciones de uso de una clase. Acoplamiento, aumenta RC+. Complejidad del mantenimiento, aumenta RC+. Reutilización, disminuye RC+. Cantidad de Pruebas, aumenta RC+.

Para aplicar la métrica TOC se tuvo en cuenta la cantidad de procedimientos que tenían las clases del componente, luego se realizó el cálculo del promedio de los procedimientos y mediante un criterio se obtuvo la categoría (baja, media, alta) para la Responsabilidad, Complejidad y Reutilización.

#### Cantidad de procedimientos por clases

No	Nombre de la clase	Procedimientos
1	Indicadores	16
2	Gest_Indicador	4
3	Variables	6
4	Objetivo	10
5	Reporte	4
6	Captura de Datos	2

## CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO

7	Base de Datos	6
8	Calcular	1
9	Carga Archivo	2
10	Conexion	2
11	Consulta	2
12	Formula	2
13	Obtener datos	6
14	Archivo	4
15	Captura	2
<b>Promedio de procedimientos</b>		<b>4.6</b>

*Tabla 12.- Procedimientos por clases*

Las siguientes tablas sirvieron de apoyo para obtener las gráficas de Responsabilidad, Complejidad y Reutilización.

### **Criterio para calcular la Responsabilidad**

Responsabilidad	Categoría	Criterio
	Baja	< =Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	> 2* Prom.

*Tabla 13.- Criterio para calcular la Responsabilidad*

### **Criterio para calcular la Complejidad de implementación**

Complejidad de implementación	Categoría	Criterio
	Baja	< =Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	> 2* Prom.

# CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO

Tabla 14.- Criterio para calcular la Complejidad de implementación

## Criterio para calcular la Reutilización

Reutilización	Categoría	Criterio
	Baja	> 2* Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	<= Prom.

Tabla 15.- Criterio para calcular la Reutilización

## Responsabilidad y Complejidad de las clases

Responsabilidad y Complejidad	Criterio
Baja	10
Media	4
Alta	1

Tabla 16.- Responsabilidad y Complejidad de las clases

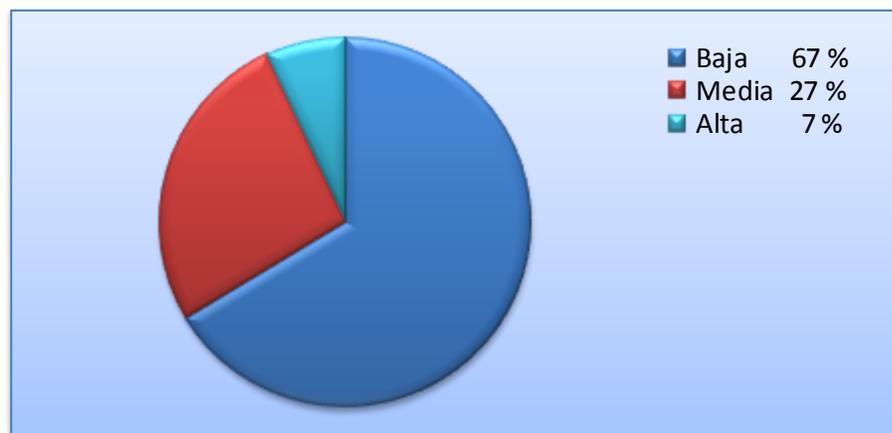
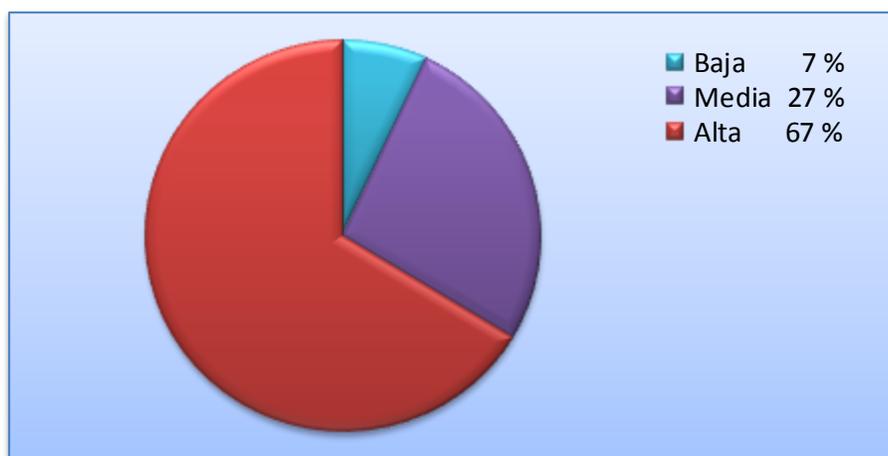


Figura 15.- Representación de la responsabilidad y complejidad

### Reutilización de las clases

Reutilización	Criterio
Baja	1
Media	4
Alta	10

*Tabla 17.- Reutilización de las clases*



*Figura 16.- Representación de la reutilización*

Cuando la suma de las clases media y baja de responsabilidad es mayor al 80% y la reutilización de ellas es alta, se puede decir que el diseño cumple con el patrón alta cohesión.

Para aplicar la métrica **RC** se tuvo en cuenta la cantidad de relaciones que tenían las clases del componente, a partir del promedio de las relaciones y mediante un criterio se obtuvo la categoría (baja, media, alta y ninguna en caso del Acoplamiento) y (baja, media, alta) para la Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas.

### Relaciones de uso por clases

No	Nombre de la clase	Relaciones
1	Indicadores	1
2	Gest_Indicador	2

## CAPÍTULO 3: VALIDACIÓN DE LOS REQUISITOS Y DISEÑO

3	VARIABLES	1
4	Objetivo	1
5	Reporte	4
6	Captura de Datos	3
7	Base de Datos	2
8	Calcular	2
9	Carga Archivo	2
10	Conexión	2
11	Consulta	2
12	Fórmula	2
13	Obtener datos	3
14	Archivo	2
15	Captura	2
<b>Promedio de asociaciones de uso</b>		<b>2.1</b>

Tabla 18.- Relaciones de uso por clases

### Acoplamiento de las clases

Acoplamiento	Cantidad de clases
Ninguno	0
Bajo	12
Medio	2
Alto	1

Tabla 19.- Acoplamiento de las clases

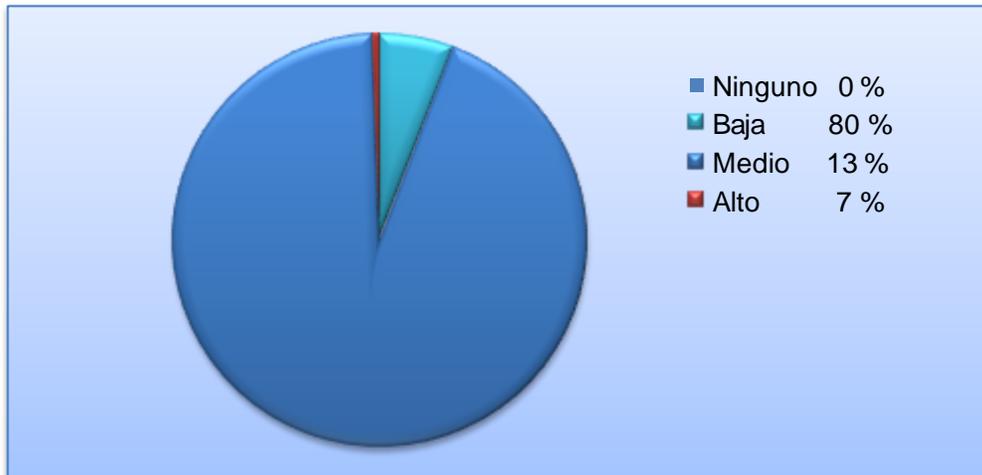


Figura 17.- Representación del acoplamiento

## Reutilización de las clases

Reutilización	Cantidad de clases
Bajo	0
Medio	3
Alto	12

Tabla 20.- Reutilización de las clases

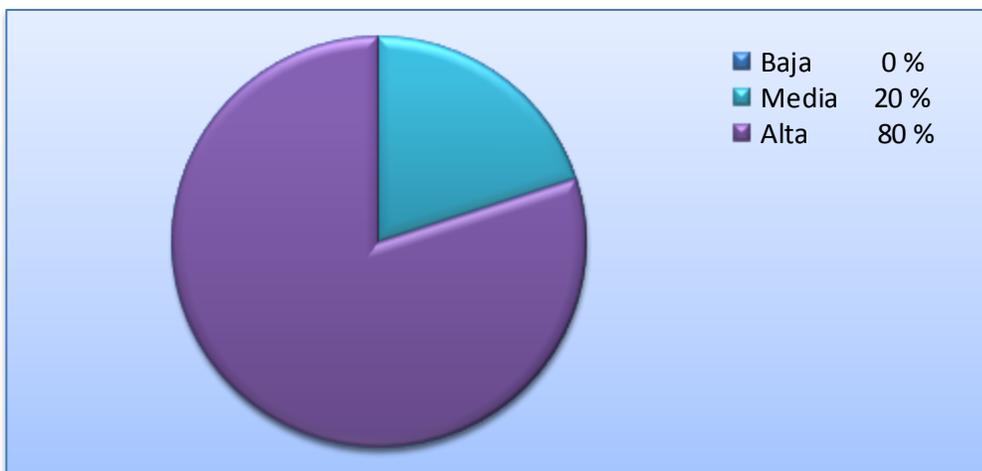


Figura 18.- Representación de la reutilización

De manera general los resultados de esta métrica son positivos, ya que el acoplamiento existente entre las clases es bajo y tienen una gran reutilización.

### **Conclusiones del Capítulo**

En este capítulo se aplicaron métricas para la validación del diseño y los requisitos obtenidos. Mediante las métricas utilizadas para la validación de los requisitos se corrigieron los posibles errores, quedando todos los requisitos con la calidad requerida para su aprobación. El diseño propuesto no presenta una alta complejidad, lo que trae consigo resultados positivos, además cumple con los patrones de bajo acoplamiento y alta cohesión. Se puede concluir que el diseño obtenido posee una calidad aceptable, facilitando la continuación eficiente del desarrollo en etapas posteriores.

## Capítulo 4: Estudio de Factibilidad y Análisis de Costos

Para la realización de un proyecto es de suma importancia el análisis del tiempo de desarrollo en meses según la cantidad de personas que se necesitan para desarrollar el proyecto, también es necesario conocer el costo total, el tiempo de duración, los gastos económicos que implicaría, así como la cantidad de hombres que se necesitan para el desarrollo del mismo. En este capítulo se hará un estudio de factibilidad y análisis de costos del componente de gestión de indicadores para el Cuadro de Mando Integral de PATDSI, donde se efectuará la estimación del mismo a través de la planificación basada en casos de uso.

### 4.1. Planificación basada en casos de uso

Aplicación de una técnica de estimación de esfuerzo y tiempo de desarrollo por Puntos de Casos de Uso.

#### 4.1.1. Identificar los puntos de casos de usos desajustados

$$UUCP = UAW + UUCW$$

Donde:

**UUCP:** Puntos de casos de uso sin ajustar

**UAW:** Factor de peso de los actores sin ajustar

**UUCW:** Factor de peso de los casos de uso sin ajustar

#### 4.1.2. Cálculo del factor de peso de los actores sin ajustar

Tipo	Descripción	Peso	Cant * peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3	1*3

## CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD Y ANÁLISIS DE COSTOS

<b>Total</b>	3
--------------	---

Tabla 21.- Factor de peso de los actores sin ajustar

$$UAW = (0*1) + (0*2) + (1*3)$$

$$UAW = 3$$

### 4.1.3. Cálculo del factor de peso de los casos de uso sin ajustar

Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	0*2
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	2*10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	2*15
<b>Total</b>			50

Tabla 22.- Factor de peso de los casos de uso sin ajustar

$$UUCW = (0*2) + (2*10) + (2*15)$$

$$UUCW = 50$$

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 50$$

$$UUCP = 53$$

### 4.2. Ajustar los puntos de casos de uso

$$UCP = UUCP * TCF * EF$$

Donde:

**UCP:** Puntos de Casos de Uso ajustados

**UUCP:** Puntos de Casos de Uso sin ajustar

**TCF:** Factor de complejidad técnica

## CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD Y ANÁLISIS DE COSTOS

**EF:** Factor de ambiente

*Escala de los factores de complejidad técnica*

Descripción	Valor
No presente o sin influencia	0
Influencia incidental o presencia incidental	1
Influencia moderada o presencia moderada	2
Influencia media o presencia media	3
Influencia significativa o presencia significativa	4
Fuerte influencia o fuerte presencia	5

*Tabla 23.- Escala de los factores de complejidad técnica*

### 4.2.1. Cálculo del factor de complejidad técnica

$$\text{TCF} = 0.6 + (0.01 * \text{TFactor})$$

$$\text{TFactor} = \sum (\text{Valor}_i * \text{Peso}_i)$$

*Peso de los factores de complejidad técnica:*

Factor	Descripción	Peso	Valor	Valor*Peso
T1	Sistema distribuido	2	5	10
T2	Objetivos de performance o tiempo de respuesta	1	5	5
T3	Eficiencia del usuario final	1	3	3
T4	Procesamiento interno complejo	1	2	2
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	4	2
T7	Facilidad de uso	0.5	4	2
T8	Portabilidad	2	0	0

## CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD Y ANÁLISIS DE COSTOS

T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	4	4
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	2	2
T13	Se requiere facilidades especiales de entrenamiento a usuario	1	1	1
<b>Total</b>				<b>42</b>

*Tabla 24.- Peso de los factores de complejidad técnica*

**TFactor = 42**

**TCF = 0.6 + (0.01\*42)**

**TCF = 1.02**

### 4.2.2. Cálculo del factor de ambiente

**EF = 1.4 + (-0.03 \* EFactor)**

**EFactor =  $\sum$  (Valor<sub>i</sub> \* Peso<sub>i</sub>)**

*Peso de los factores ambientales:*

<b>Factor</b>	<b>Descripción</b>	<b>Peso</b>	<b>Valor</b>	<b>Valor*Peso</b>
E1	Familiaridad con el modelo de proyecto utilizado	1.5	5	7.5
E2	Experiencia en la aplicación	0.5	3	1.5
T3	Experiencia en orientación a objetos	1	3	3
E4	Capacidad del analista líder	0.5	4	2
E5	Motivación	1	4	4

## CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD Y ANÁLISIS DE COSTOS

E6	Estabilidad de los requerimientos	2	2	4
E7	Personal part-time	-1	4	-4
E8	Dificultad del lenguaje de programación	-1	5	-5
<b>Total</b>				<b>13</b>

Tabla 25.- Peso de los factores ambientales

**EFactor = 13**

**EF = 1.4 + (-0.03 \*13)**

**EF = 1.01**

*Luego:*

**UCP = UUCP \* TCF \* EF**

**UCP = 53 \* 1.02 \* 1.01**

**UCP = 54.60**

### 4.3. Calcular esfuerzo de implementación

**E = UCP \* CF**

*Donde:*

**E:** esfuerzo estimado en horas-hombre

**UCP:** Puntos de Casos de Uso ajustados

**CF:** factor de conversión

**TEF:** Total de esfuerzo

#### 4.3.1. Cálculo del factor de conversión

CF = 20 horas-hombre (si TEF ≤ 2)

CF = 28 horas-hombre (si TEF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si TEF ≥ 5)

**TEF = Cant TEF < 3 (entre E1, E6) + Cant TEF > 3 (entre E7, E8)**

## CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD Y ANÁLISIS DE COSTOS

$$\text{TEF} = 2 + 0$$

$$\text{TEF} = 2$$

$$\text{CF} = 20 \text{ horas-hombre (porque TEF} \leq 2)$$

Luego:

$$\text{E} = \text{UCP} * \text{CF}$$

$$\text{E} = 54.60 * 20 \text{ horas hombre}$$

$$\text{E} = 1092 \text{ horas hombre}$$

### 4.4. Esfuerzo de todo el proyecto

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	272 horas hombre
Diseño	20%	543 horas hombre
Implementación	40%	1092 horas hombre
Prueba	15%	408 horas hombre
Sobrecarga	15%	408 horas hombre
Total	100%	2723 horas hombre

Tabla 26.- Esfuerzo de todo el proyecto por flujos de trabajo

Suponiendo que la jornada laboral del proyecto es de 7 horas, y un mes tiene como promedio 30 días; la cantidad de horas que puede trabajar una persona en 1 mes es 210 horas.

Si el esfuerzo de todo el proyecto es de **2723 horas-hombre** y por cada 210 horas se tiene 1 mes, eso daría un **TEF = 12.97 mes-hombre**.

De los resultados obtenidos se llega a la conclusión que con 2 hombres trabajando en el proyecto el mismo se desarrolla en aproximadamente 6 meses y medio.

### 4.5. Beneficios tangibles e intangibles

Con la implementación y puesta en marcha del componente automatizado para la gestión de indicadores del Cuadro de Mando Integral de PATDSI se evidencian beneficios tanto tangibles como intangibles, que se manifiestan a corto y largo plazo.

Entre los beneficios tangibles que reportará la creación del componente se pueden citar:

- Brindará un sistema flexible y de interfaz agradable para el trabajo con indicadores.
- Gestionará de forma eficiente los indicadores de las empresas, lo que se traducirá en un aumento en el rendimiento de las mismas.
- Generará la entrada necesaria para el correcto desempeño del Cuadro de Mando Integral.
- Garantizará la seguridad e integridad de la información manejada a partir del sistema.

En cuanto a los beneficios intangibles, podrían mencionarse los siguientes:

- Aumentará la experiencia del equipo de trabajo en las tecnologías empleadas en el desarrollo del sistema.
- Contribuirá con el nuevo modelo de Formación – Producción propiciando el desempeño de estudiantes en el mismo, generando además trabajos de diploma.
- Aumentará la satisfacción del cliente y la identificación con los productos desarrollados en la institución.

### 4.6. Análisis del costo

El análisis de los costos no sólo ayuda a determinar el costo del proyecto y su mantenimiento, sino que también sirve para determinar si vale o no la pena llevarlo a cabo. Todo proyecto informático lleva implícito costo monetario, recursos y esfuerzo, por eso es necesario justificar la realización de cualquier producto de este tipo. A continuación se muestra una tabla en la que aparecen los cálculos relacionados con el costo del proyecto y seguidamente se emiten las conclusiones de por qué es factible la implementación del mismo.

## CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD Y ANÁLISIS DE COSTOS

### 4.6.1. Costo del Proyecto

Se tiene que:

Asumiendo que el salario básico de un trabajador del proyecto es de \$349.00 mensual:	Asumiendo que el salario básico de un estudiante es de \$50.00 mensual:
<p><b>CH</b> = Cantidad de hombres  <b>TTP</b> = Tiempo total del proyecto  <b>SM</b> = Salario mensual  <b>TEF</b>: Total de esfuerzo  <b>CP</b> = Costo del proyecto</p> <p><b>CH</b> = 2 hombres  <b>SM</b> = \$349  <b>TEF</b> = 12.97 mes-hombre</p> <p><b>CP = SM * CH * TEF</b>  <b>CP = 349 * 2 * 12.97</b>  <b>CP = \$9053.06</b></p> <p><b>TTP = TEF/CH</b>  <b>TTP = 12.97/2</b>  <b>TTP = 6.49 meses</b></p>	<p><b>CH</b> = Cantidad de hombres  <b>TTP</b> = Tiempo total del proyecto  <b>SM</b> = Salario mensual  <b>TEF</b>: Total de esfuerzo  <b>CP</b> = Costo del proyecto</p> <p><b>CH</b> = 2 hombres  <b>SM</b> = \$50  <b>TEF</b> = 12.97 mes-hombre</p> <p><b>CP = SM * CH * TEF</b>  <b>CP = 50 * 2 * 12.97</b>  <b>CP = \$1297</b></p> <p><b>TTP = TEF/CH</b>  <b>TTP = 12.97/2</b>  <b>TTP = 6.49 meses</b></p>
<p>De los resultados obtenidos se interpreta que con 2 trabajadores construyendo el proyecto el mismo se desarrolla en aproximadamente 6 meses y medio y que su costo total se estima que sea (\$9053.06).</p>	<p>De los resultados obtenidos se interpreta que con 2 estudiantes trabajando en el proyecto el mismo se desarrolla en aproximadamente 6 meses y medio y que su costo total se estima que sea (\$1297).</p>

*Tabla 27.- Costo del Proyecto*

El componente que se propone implementar está dirigido a PATDSI y una vez implantado, contribuirá a aumentar la eficiencia de las actividades disminuyendo el tiempo en la gestión de los indicadores. Para la

## CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD Y ANÁLISIS DE COSTOS

---

realización de este componente se utilizarán tecnologías libres y de las propietarias posibles a usar ya se tienen las licencias de uso. DATEC y PATDSI cuentan con los integrantes, herramientas y tecnologías necesarias para la implementación del componente, contribuyendo esto que no se tenga que incurrir en gastos de hardware o software. Por lo anteriormente expuesto y luego de haber realizado los cálculos y análisis pertinentes de los costos del proyecto, se llega a la conclusión de que la implementación del componente es factible. Queda en manos de la dirección del proyecto la toma de decisiones enfocadas a la optimización de las variables planteadas en el estudio realizado de acuerdo con las necesidades y condiciones reales existentes para su desarrollo.

### **Conclusiones del Capítulo**

En este capítulo se describió el estudio de factibilidad correspondiente al componente propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado, para ello se efectuó un análisis de los costos del proyecto y el esfuerzo para llevarlo a cabo donde se llegó a la conclusión de que es factible implementar el componente por los beneficios que reporta, principalmente al mejorar el proceso de gestión de indicadores del Cuadro de Mando Integral de PATDSI.

### **Conclusiones Generales**

Se realizó un estudio sobre la gestión de indicadores y se analizaron los diferentes software con características similares en cuanto a las funcionalidades del sistema a desarrollar, llegando a la conclusión de que es necesario la realización del mismo por las características que presentan las empresas cubanas en particular y porque estos sistemas son cotizados a precios muy altos en el mundo convirtiéndose en inaccesibles para países como Cuba.

Se realizó el análisis y diseño de una posible solución para la futura implementación del componente de Gestión de Indicadores obteniéndose como resultado todos los artefactos necesarios y un mayor entendimiento del proceso de negocio.

Se aplicaron métricas y técnicas de validación al diseño y los requisitos del componente a desarrollar, corrigiéndose todos los errores existentes para lograr como resultado la calidad requerida de los requisitos para su aprobación y un diseño que cumple con los patrones utilizados propiciando la continuación eficiente del desarrollo en etapas posteriores.

Se realizó un estudio de factibilidad y análisis del costo del componente de Gestión de Indicadores y luego de los cálculos realizados se llegó a la conclusión de que es factible implementar el componente por los beneficios que reporta.

### **Recomendaciones:**

Se recomienda realizar la implementación del componente propuesto, así como agregar nuevas funcionalidades al sistema de acuerdo con las expectativas del cliente y desarrollar la ayuda del mismo.

## Bibliografía

**A.Durán, y. o. (1999 ).** A Requirements Elicitation Approach Based in Templates and Patterns. Sevilla, España.

**Agut, R. M. (2001).** Especificación de Requisitos Software según el estándar IEEE 830.

**Almaguer Bajuelo, Yunior Miguel, Quevedo Lussón, Juan Carlos y Rodríguez Pino, Adisleydis.** Propuesta de metodología de un grupo de análisis que utiliza un modelo de desarrollo basado en líneas de productos de software. Ciudad de la Habana : s.n.

**ALTAIR. 2005.** *El Cuadro de Mando Integral*. s.l. : Economía 3, 2005.

**Aponte Burgos, Yelenis y Azán Basallo, Yasser. 2007.** Módulo gestión de reportes y la estructura de una organización del Cuadro de Mando Integral. Ciudad de La Habana : s.n., 2007.

**Axsellit. 2010.** Axsellit. Axsellit. [En línea] 2010. [Citado el: 5 de 2 de 2010.] <http://www.axsellit.com/balancedscorecard.aspx>.

**Bichachi, D. S. (2009).** El uso de las Listas de Chequeo como herramienta para controlar la calidad de la ley.

**Businesscol.com. (2009).** Businesscol.com. Recuperado el 21 de 3 de 2010, de Businesscol.com: <http://www.businesscol.com/productos/glosarios/economico/glossary.php?word=INDICADOR%20DE%20EFICIENCIA>

**Cabrera, L. (2008).** Planeación, Estrategias Y Táctica.

**Carrasco, L. d. (2008).** Métricas del Software.

**Cavsi. 2008.** Cavsi. Cavsi. [En línea] [Citado el: 10 de 3 de 2010.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

**Ciencia y Técnica Administrativa. 2008.** HERRAMIENTAS CASE. [En línea] [Citado el: 15 de 2 de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.

**Danielpecos.com. 2010.** danielpecos.com. danielpecos.com. [En línea] 2010. [Citado el: 10 de 3 de 2010.] [http://www.netpecos.org/docs/mysql\\_postgres/x15.html](http://www.netpecos.org/docs/mysql_postgres/x15.html).

**Departamento de Lenguajes y Sistemas Informáticos. (2008).** Patrones de Asignación de Responsabilidades (GRASP). Sevilla.

**Desarrollo Web. 2007.** desarrolloweb.com. desarrolloweb.com. [En línea] 2007. [Citado el: 10 de 3 de 2010.] <http://www.desarrolloweb.com/articulos/840.php>.

**Delphos. Delphos. [En línea] 2008.** [Citado el: 1 de 2 de 2010.] <http://www.deinsa.com/cmi/principal.htm>.

**Díaz Hernández, Michel, Pacheco López, Gianni y Mesa Rodríguez, Yadier. 2009.** Desarrollo de un software para un Cuadro de Mando Integral. Ciudad de La Habana : s.n., 2009.

**Ejemplos java y C/linux. (2009).** Recuperado el 23 de 3 de 2010, de Ejemplos java y C/linux: <http://www.chuidiang.com/ood/metodologia/scrum.php>

**Escribano, G. F. (2002).** Introducción a Extreme Programming.

**Ferriol Ortiz, A., & Azahares Reyes, E. (2009).** Análisis y Diseño del Módulo Registro de Control Docente para Akademos V2.0. Ciudad de La Habana.

**Fernández., Ing. Leidy Sánchez. 2006.** "Procedimiento para el desarrollo del proceso de ingeniería de requisitos en un proyecto software (PROCIR)". 2006.

**Germán Harvey Alférez Salinas, Germán Harvey Alférez Salinas. 2008.** Establecimiento de una Metodología de Desarrollo de Software. 2008.

**Gestión de Proyectos. (2008).** Gestión de Proyectos. Recuperado el 24 de 4 de 2010, de Gestión de Proyectos: [http://kasyles.blogspot.com/2008\\_09\\_01\\_archive.html](http://kasyles.blogspot.com/2008_09_01_archive.html)

**González, Rafael Humberto Soler. 2006.** El Cuadro de Mando Integral en el CENEX. 2006.

**PMBOK. 2004.** Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®) Tercera Edición. 2004.

**Imetrix. (2009).** Imetrix. Recuperado el 20 de 1 de 2010, de Imetrix: <http://www.imetrix.biz/es/indicadores-claves-desempeno.html>

**Ingeniería de Software. (2008).** Fase de Inicio. Flujo de Análisis y Diseño. Modelo de Análisis., (pág. 16).

**Ingeniería del software. (2010).** Principales herramientas CASE del mercado y su uso. Recuperado el 6 de 3 de 2010, de Principales herramientas CASE del mercado y su uso: [http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE\\_principales.html](http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE_principales.html)

**Ivar Jacobson, James Rumbaugh, Grady Booch. 2000.** "El Proceso Unificado de Desarrollo de Software". Madrid : s.n., 2000.

**Jaramillo, A. C. (2009).** LOS INDICADORES DE GESTIÓN.

**Managing Requirements (2010).** Managing Requirements. Retrieved 4 25, 2010, from Managing Requirements: [http://www.jiludwig.com/Traceability\\_Matrix\\_Structure.html](http://www.jiludwig.com/Traceability_Matrix_Structure.html)

**Medrano, Jesus Rafael Sanchez. 2009.** Introducción a Base de Datos con PostgreSQL. 2009. pág. 69.

**Millet Lombida, Yanetsi y Piloto Arregui, Yaiset. 2007.** Módulo gestión de estrategias e indicadores del Cuadro de Mando Integral. Ciudad de La Habana : s.n., 2007.

**MySQL. (2010).** MySQL. Recuperado el 12 de 3 de 2010, de MySQL: <http://dev.mysql.com/doc/refman/5.0/es/introduction.html>

**Pérez Castillo, I., & Valiente Mesa, R. (2009).** LIMS DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA: DESARROLLO DE LA BASE DE DATOS DEL MÓDULO SECCIÓN DE MEJORAMIENTO DE LA CALIDAD. Ciudad de La Habana.

**Popkin Software and Systems. (2009).** Modelado de Sistemas con UML.

**PostgreSQL Global Development Group. 2009.** PostgreSQL 8.4.1 Documentation. California : s.n., 2009. pág. 2119.

**Pressman, R. S. (2002).** Ingeniería del Software.

- Rational Rose Enterprise Edition (2008).** Rational-Rose-Enterprise-Edition. Recuperado el 10 de 4 de 2010, de Rational-Rose-Enterprise-Edition: <http://www.taringa.net/posts/downloads/883942/Rational-Rose-Enterprise-Edition-2002.html>
- Restrepo de Ocampo, I. S., Estrada Mejía, S., & Ramirez Aristizabal, C. (2007).** Modelo de Gestión de Indicadores para una Empresa de Ventas de Vehículos. Colombia.
- Riebeling, C. (2009).** Project Management Institute. México.
- Richard Thayer, Merlin Dorfam. 2000.** Software Requeriments Engineering. Los Alamitos, California : s.n., 2000.
- Roger Oberg, Leslee y Ericsson Probasco. mayo de 2005.** Applying requirements management with use cases. mayo de 2005.
- Rubino, Marta Teresa. 2007.** Desarrollo de un Cuadro de Mando Integral aplicable a Bibliotecas: descripción de las etapas para su creación y diseño. Buenos Aires : s.n., 2007.
- Salinas, Caro. 2004.** "Unified Modeling Language.". 2004.
- Sinnexus. (2007).** Sinnexus. Recuperado el 15 de 1 de 2010, de Sinnexus: [http://www.sinnexus.com/business\\_intelligence/cuadro\\_mando\\_integral.aspx](http://www.sinnexus.com/business_intelligence/cuadro_mando_integral.aspx)
- Soft, Decide.** Decide Soft. Decide Soft. [En línea] [Citado el: 5 de 2 de 2010.] [http://www.fba-consulting.com/decidesoft/esp/cmi\\_c\\_faq.htm](http://www.fba-consulting.com/decidesoft/esp/cmi_c_faq.htm).
- Soluciones basadas en Software Libre. (2008).** CodigoAberto.es. Recuperado el 3 de 2 de 2010, de CodigoAberto.es: <http://codigoaberto.es/codigoaberto>
- Tecnología y Synergix. 2008.** Tecnología y Synergix. Tecnología y Synergix. [En línea] 10 de 7 de 2008. [Citado el: 12 de 3 de 2010.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
- Universidad Nacional de Colombia. (2010).** Dirección Nacional de Servicios Académicos Virtuales. Recuperado el 21 de 3 de 2010, de Dirección Nacional de Servicios Académicos Virtuales:

<http://www.virtual.unal.edu.co/cursos/sedes/manizales/4010059/html/Modulo%20Indicadores/Lecturas%20indicadores/LECTURA4.doc>

**Vega Rosales, Miguelina y Vega V, Raiminudo. 2005.** Diseño de un CMI para la Facultad de Ciencias de la Ingeniería de la Universidad Austral de Chile. Valdivia Chile : s.n., 2005.

**Visual Paradigm for UML (Standard Edition). (2009).** Visual Paradigm for UML (Standard Edition). Recuperado el 26 de 3 de 2010, de Visual Paradigm for UML (Standard Edition): <http://www.altacracks.com/programas-gratis/visual-paradigm-for-uml-standard-edition>

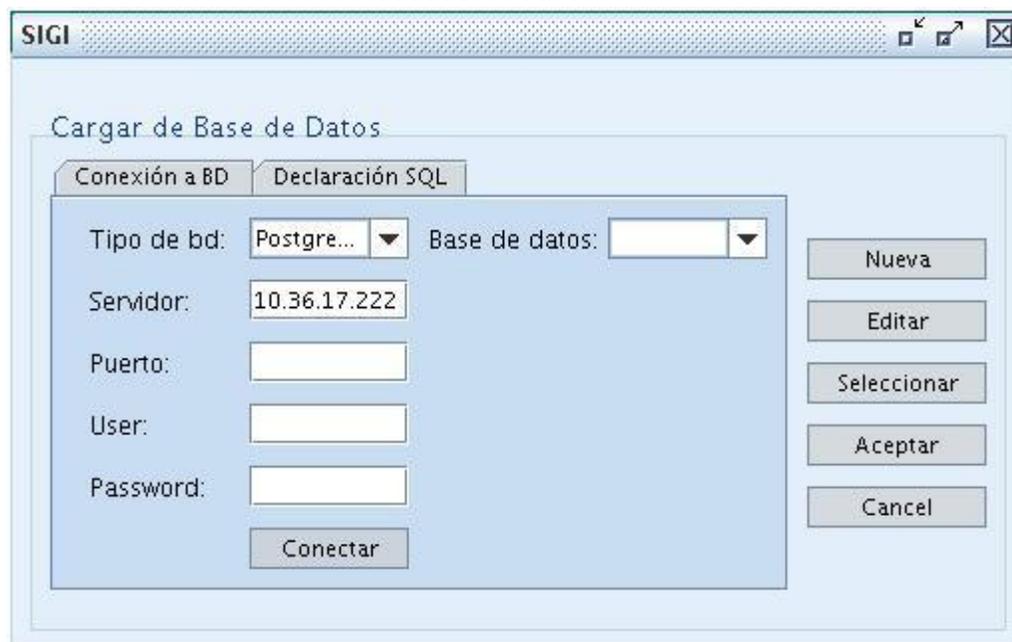
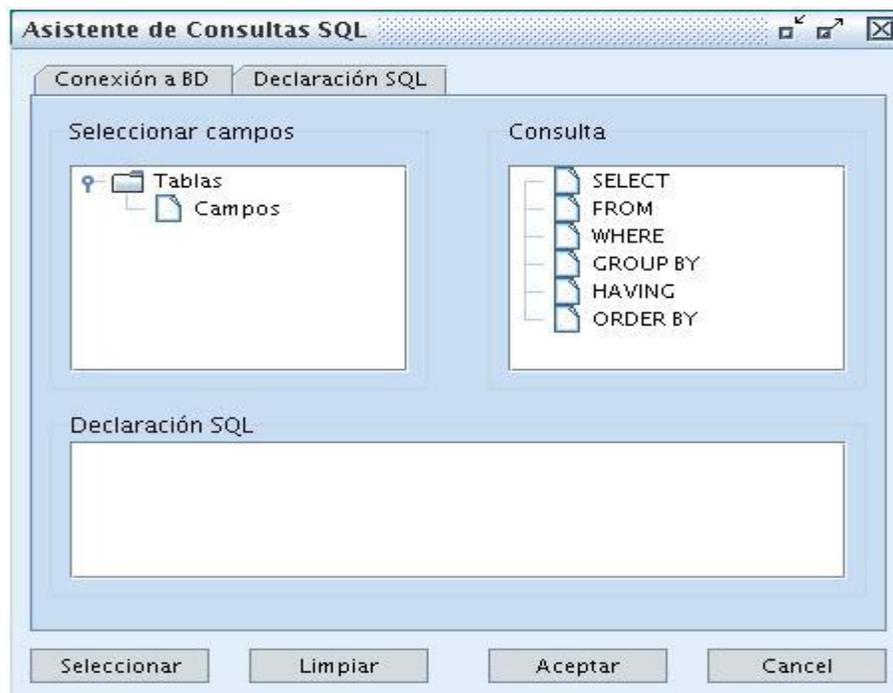
## Anexos

### Anexo 1: Descripción textual de los Casos de Uso del negocio

#### Descripción del CU Configurar Captura de Datos

Caso de Uso	
<b>CU_3</b>	Configurar Captura de Datos
<b>Propósito</b>	El Supervisor gestionará la forma en que se van a capturar los valores de los indicadores.
<b>Actores</b>	Supervisor.
<b>Resumen</b>	El Supervisor puede definir las formas de captura del valor del indicador, ya sea a través de valores digitados, carga desde una base de datos, carga desde un archivo, o crear una fórmula.
<b>Pre-Condiciones</b>	Debe existir al menos una Entidad, la misma debe de tener al menos un objetivo registrado en la BD.
<b>Post-Condiciones</b>	Se actualiza el valor del indicador en la BD.
<b>Referencias</b>	<b>RF 2, RF 5, RF 6, RF 7, RF 8, RF 9.</b>
<b>Prioridad</b>	Crítico
Acción del Actor	Respuesta del Sistema
1 El Supervisor selecciona una opción de Obtención de datos.	1.1 Si selecciona la opción cargar de BD ir a sección: "Carga de Base de Datos" 1.2 Si selecciona la opción cargar de archivo ir a sección: "Carga de Archivo" 1.3 Si selecciona la opción crear fórmula ir a sección: "Crear Fórmula".
Sección 1: Carga de Base de Datos	
	2.1 El sistema muestra la interfaz de Carga de Base de Datos.
3 El Supervisor selecciona la opción Conexión a BD.	3.1 EL sistema muestra la interfaz de Conexión a BD.
4 El Supervisor introduce los datos de configuración para conectarse a la BD.	4.1 La aplicación comprueba que los datos estén escritos correctamente mediante la validación de los campos y guarda los datos de la conexión.
5 El Supervisor selecciona la opción Declaración SQL.	5.1 El sistema muestra la interfaz del Asistente de Consultas SQL.
6 El Supervisor declara la consulta.	6.1 El sistema guarda la sentencia SQL.
Flujos alternativos	
Acción del Actores	Respuesta del Sistema
	4.2 Si el Supervisor introduce mal los datos, el sistema muestra un mensaje de error.
Sección 2: Crear Fórmula	

	7.1 El sistema muestra la interfaz de Crear Fórmula con una lista de indicadores que ya han sido agregados al sistema.
8 El Supervisor selecciona los indicadores y variables que desea asociar en su fórmula y crea la misma.	8.1 La aplicación comprueba que la fórmula esté escrita correctamente, muestra la interfaz para configurar la variable y guarda los parámetros de la fórmula.
<b>Flujos alternativos</b>	
<b>Acción del Actores</b>	<b>Respuesta del Sistema</b>
	8.2 Si el usuario introduce mal los datos, el sistema muestra un mensaje de error y vuelve al paso 7.1. 8.3 En caso de que la fórmula no contenga variables, se guardan los parámetros de la misma.
<b>Sección 3: Carga de Archivo</b>	
9	10.1 El sistema muestra la interfaz de Carga de Archivo.
11 El Supervisor selecciona la opción del tipo de archivo.	11.1 El sistema muestra la estructura del archivo en el formato especificado.
12 El Supervisor selecciona de la estructura del archivo los elementos con los que quiere alimentar al indicador y configura las operaciones que desee con esos elementos.	12.1 La aplicación comprueba que las operaciones son correctas y las guarda.
<b>Flujos alternativos</b>	
<b>Acción del Actores</b>	<b>Respuesta del Sistema</b>
13	12.2 Si las operaciones son incorrectas, el sistema muestra un mensaje de error y vuelve al paso 11.1
<b>Prototipo de Interfaz de Usuario</b>	



**SIGI**

Editor de fórmulas

+

-

/

\*

=

(

)

[

]

Crear variables

Búsqueda de indicador

Limpiar

Aceptar

Cancel

Listado de indicadores

Código	Nombre	UM	Periodicidad	Tipo	Estado

Búsqueda avanzada

Nombre:

Tipo:

Código:

Aceptar

**SIGI**

Crear variable

Nombre:

Tipo de dato:

Aceptar

Cancel

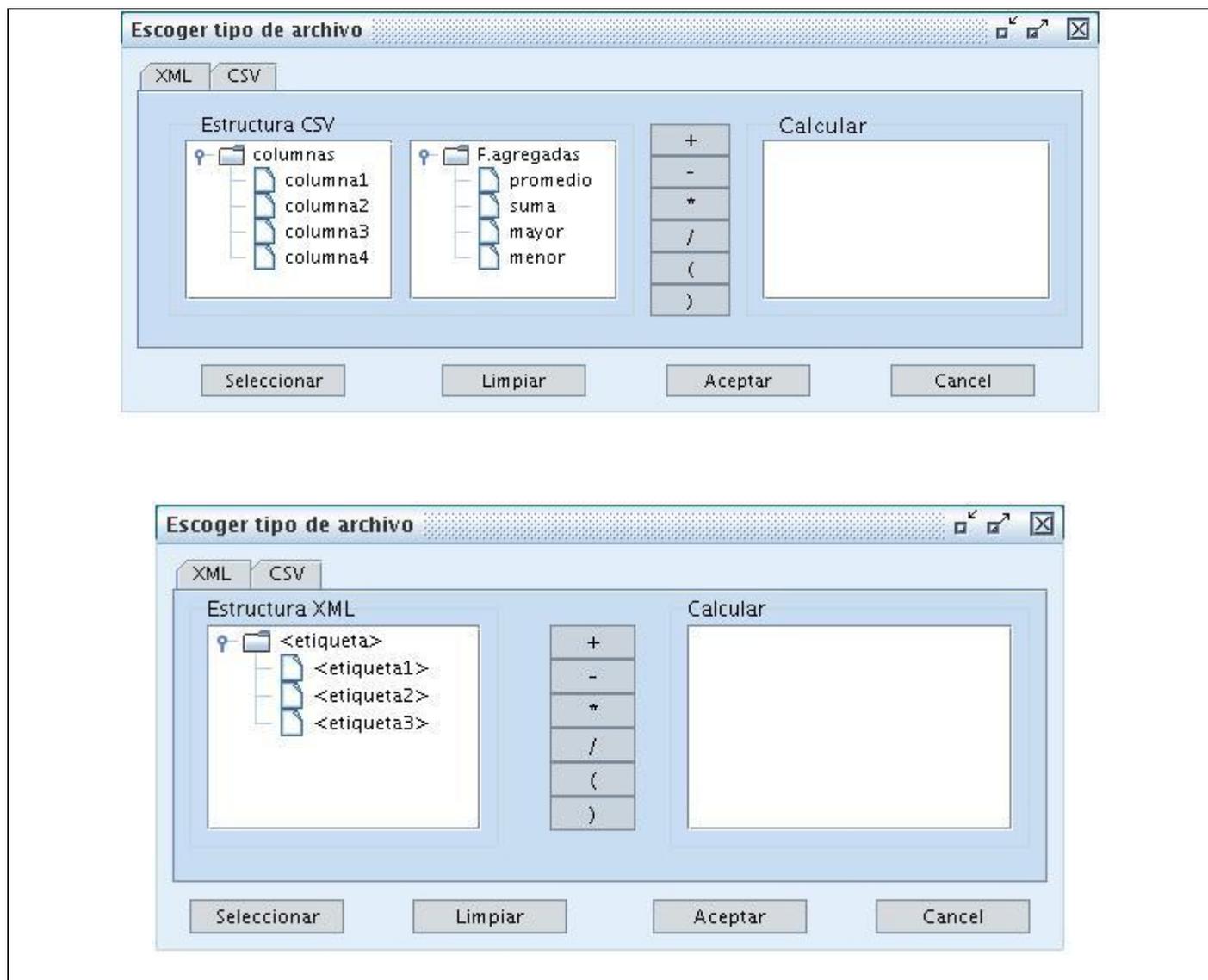


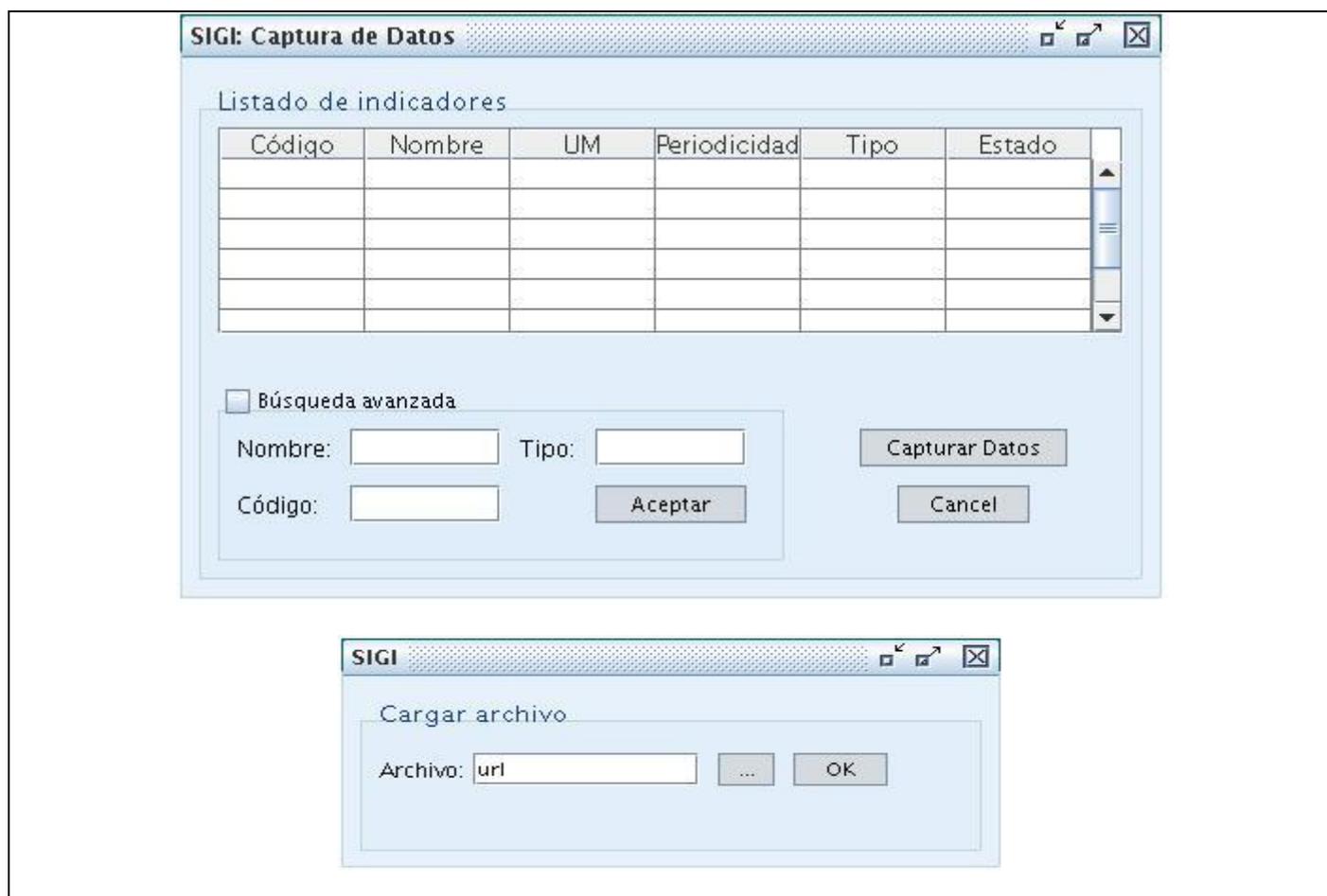
Tabla 28.- Descripción del CU Configurar Captura de Datos

### Descripción del CU Capturar de Datos

Caso de Uso	
<b>CU_2</b>	Capturar Datos
<b>Propósito</b>	Realizar la captura de los datos del indicador seleccionado.
<b>Actores</b>	Digitador
<b>Resumen</b>	El Digitador le introducirá los datos al indicador seleccionado.

<b>Pre-Condiciones</b>	Debe existir al menos una Entidad, la misma debe tener al menos un Objetivo registrado en la BD.	
<b>Post-Condiciones</b>	Se actualiza en la BD el valor del indicador.	
<b>Referencias</b>	RF 2, RF 5, RF 6, RF 7, RF 8, RF 9.	
<b>Prioridad</b>	Crítico	
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1	El Digitador selecciona el indicador al cual le hará la captura de datos.	1.1 Si el indicador obtiene los datos desde una fórmula ir a sección: "Fórmula" 1.2 Si el indicador obtiene los datos desde una BD ir a sección: "Carga de Base de Datos" 1.3 Si el indicador obtiene los datos desde la carga de archivo ir a sección: "Carga de Archivo".
<b>Sección 1: Fórmula</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
2		2.1 El sistema muestra la interfaz para introducir los valores de las variables definidas en la fórmula.
3	El Digitador introduce los valores del indicador en el sistema.	3.1 La aplicación comprueba que los datos estén escritos correctamente, ejecuta la fórmula y muestra el resultado.
<b>Flujos alternativos</b>		
<b>Acción del Actores</b>		<b>Respuesta del Sistema</b>
		3.2 Si el Digitador introduce mal los datos el sistema muestra un mensaje de error. 3.3 En caso de que la fórmula no contenga variables, el sistema muestra una interfaz con el resultado de la misma.
<b>Sección 2: Carga de Base de Datos</b>		
4		4.1 El sistema muestra la interfaz con la opción "Conectarse".
5	El Digitador selecciona esta opción.	5.1 El sistema realiza la conexión, ejecuta la consulta y devuelve los resultados.
6	El Digitador debe seleccionar el valor que desee para asignárselo al indicador.	6.1 Se actualiza la BD con el nuevo valor del indicador, mostrando además un mensaje de éxito.
<b>Flujos alternativos</b>		
<b>Acción del Actores</b>		<b>Respuesta del Sistema</b>
5		5.2 En caso de fallo de la conexión se muestra un mensaje de conexión fallida.
<b>Sección 3: Carga de Archivo</b>		
7		7.1 El sistema muestra la interfaz de Carga de Archivo.

8 El Digitador selecciona la ubicación del archivo.	8.1 El sistema extrae los datos del archivo en el formato especificado y los devuelve.
<b>Flujos alternativos</b>	
<b>Acción del Actores</b>	<b>Respuesta del Sistema</b>
	8.2 En caso de que el archivo no sea del tipo especificado se muestra un mensaje de error.
<b>Prototipo de Interfaz de Usuario</b>	



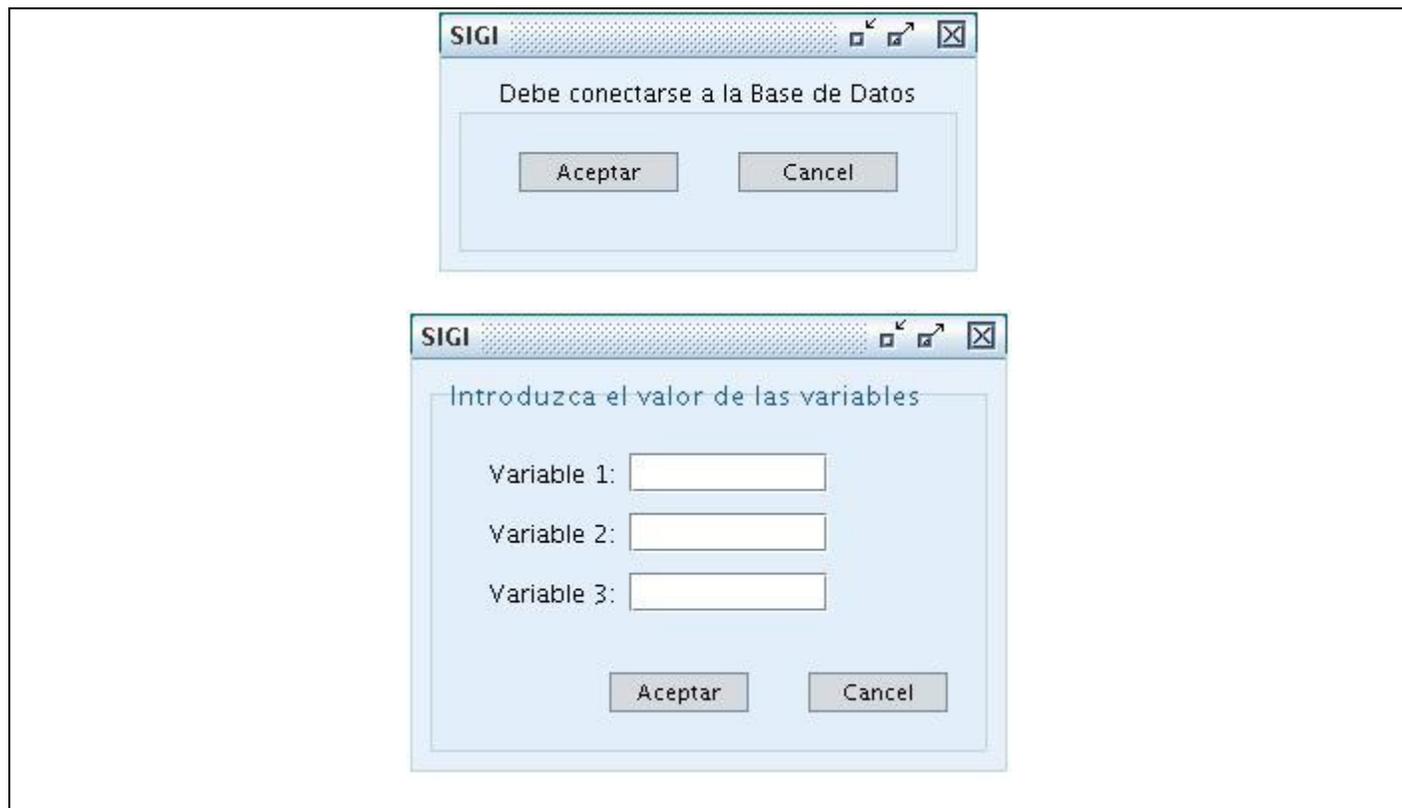


Tabla 29.- Descripción del CU Captura de Datos

### Descripción CU Asociar a objetivos

Caso de Uso	
<b>CU_4</b>	Asociar a Objetivos
<b>Propósito</b>	El Supervisor asociará un indicador a los objetivos.
<b>Actores</b>	Supervisor.
<b>Resumen</b>	El Supervisor puede asociar o no un indicador a diferentes objetivos haciendo las respectivas actualizaciones en la BD.
<b>Pre-Condiciones</b>	Debe Existir al menos una Entidad, la misma debe tener al menos un Objetivo registrado en la BD.
<b>Post-Condiciones</b>	Se actualiza la BD.
<b>Referencias</b>	<b>RF 10</b>
<b>Prioridad</b>	Crítico
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

1. El Supervisor selecciona el indicador y luego la opción de Asociar a Objetivo de la interfaz de Adicionar Indicadores.	1.1 La aplicación muestra una interfaz con la lista de objetivos para que el Supervisor seleccione los que quiere asociar con el indicador.
2 El Supervisor selecciona los objetivos.	2.1 La aplicación actualiza la BD con los cambios realizados.
<b>Flujos alternativos</b>	
<b>Prototipo de Interfaz de Usuario</b>	

**Asociar a objetivos**

Listado de objetivos

Nombre del indicador

Asociar

Quitar

Aceptar

Cancelar

Búsqueda avanzada

Nombre:  Id:

Plazo:  Aceptar

Tabla 30.- Descripción CU Asociar a objetivos

## Anexo 2: Diagrama de clases del análisis

### Diagrama de clases del análisis: Capturar Datos

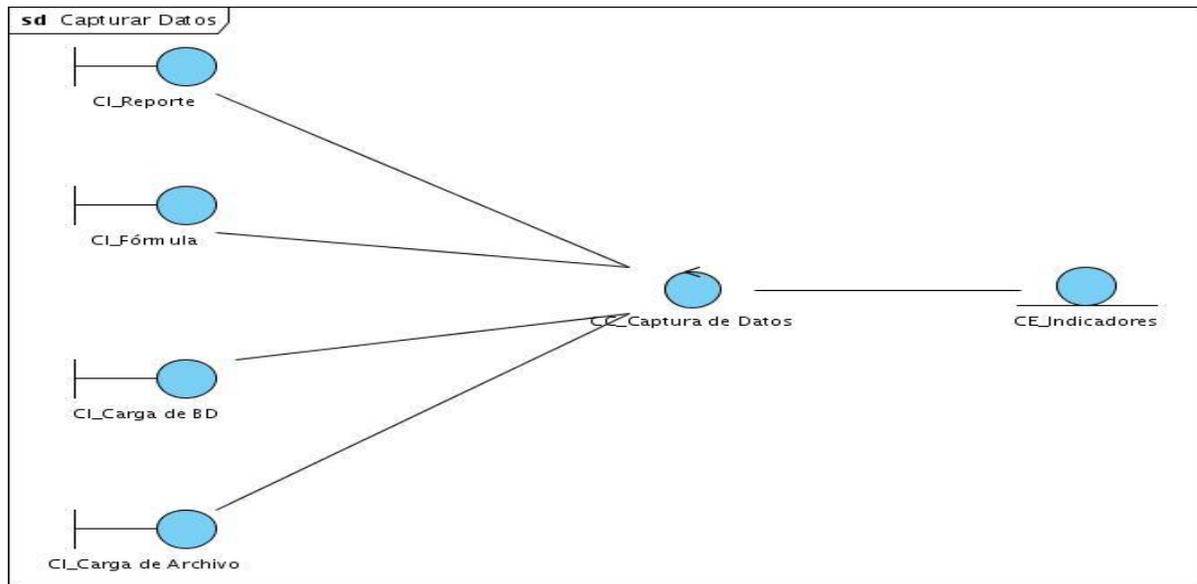


Figura 19.- Diagrama de clases del análisis: Capturar Datos

### Diagrama de clases del análisis: Configurar Captura de Datos

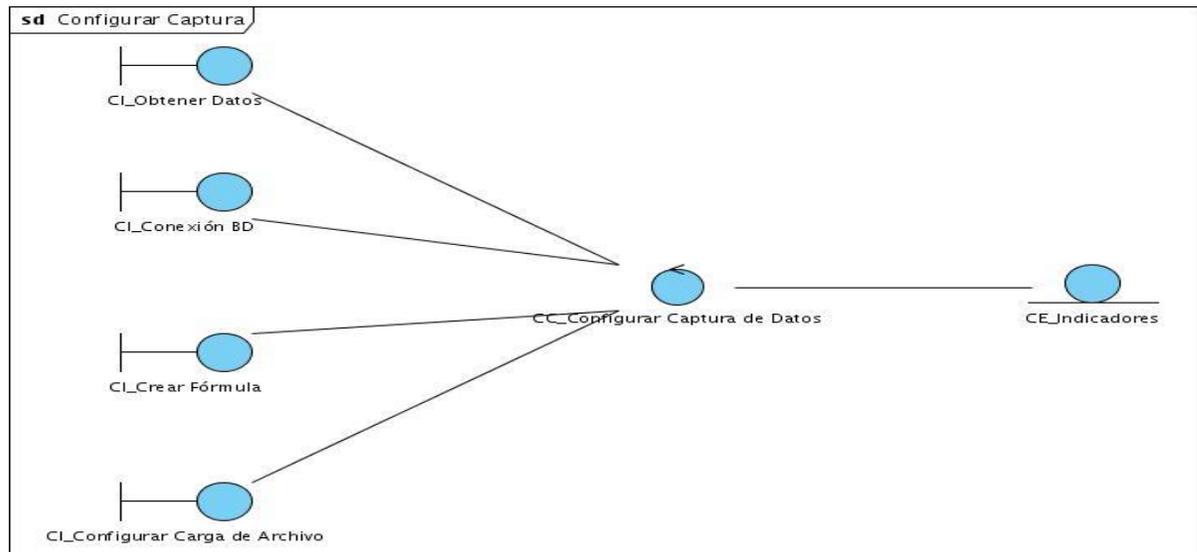


Figura 20.- Diagrama de clases del análisis: Configurar Captura de Datos

**Diagrama de clases del análisis: Asociar a objetivos**

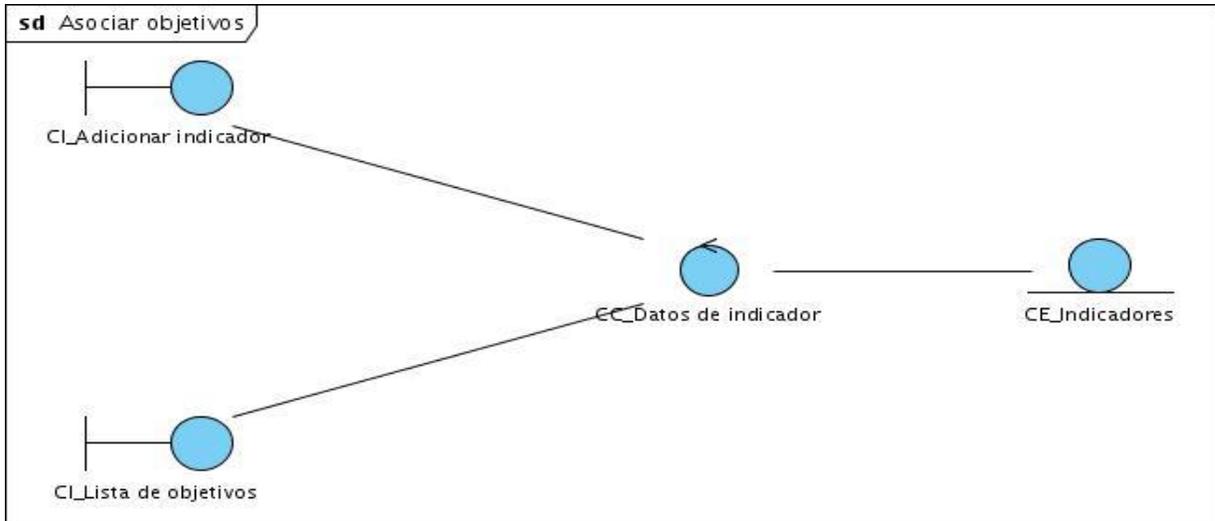


Figura 21.- Diagrama de clases del análisis: Asociar a objetivos

**Anexo 3: Diagramas de colaboración**

❖ **Diagrama de colaboración: Caso de Uso Capturar Datos**

**Capturar Datos (Fórmula)**

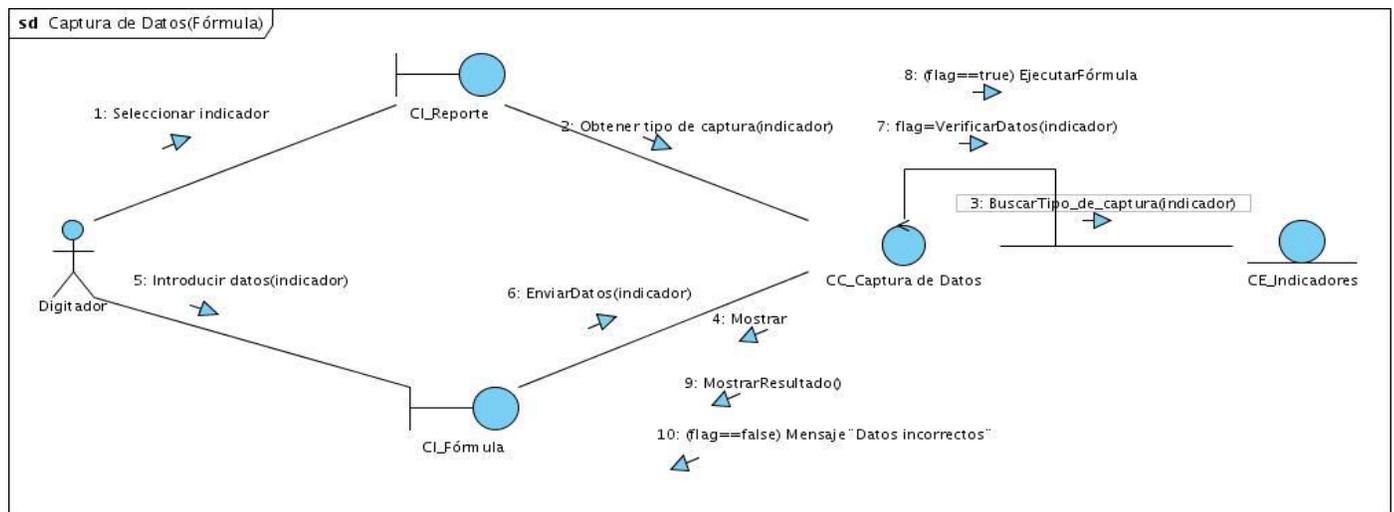


Figura 22.- Diagrama de colaboración: Capturar Datos (Fórmula)

**Capturar Datos (Archivo)**

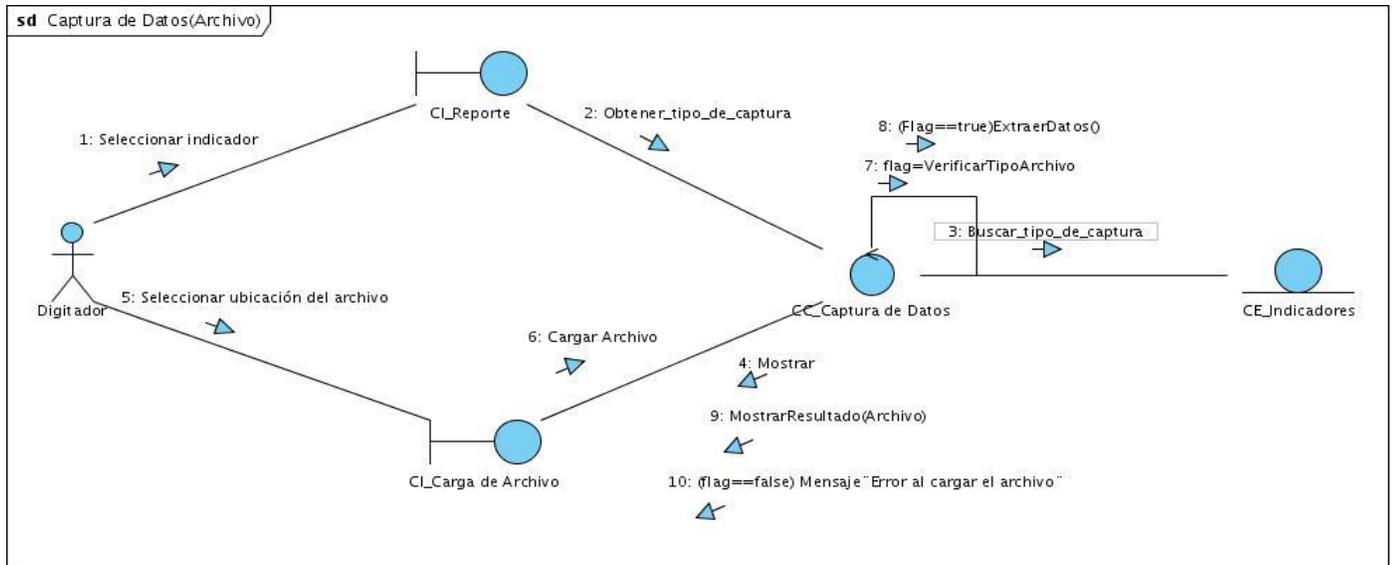


Figura 23.- Diagrama de colaboración: Capturar Datos (Archivo)

**Capturar Datos (Base de Datos)**

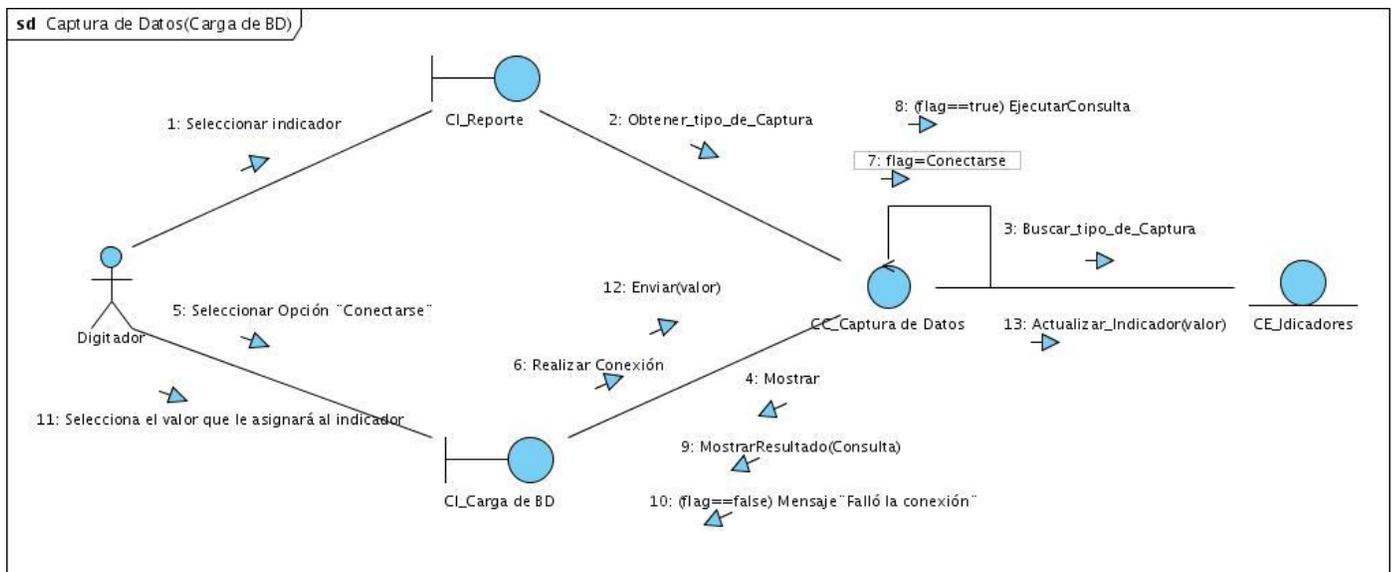


Figura 24.- Diagrama de Colaboración: Capturar Datos (Base de Datos)

❖ Diagrama de colaboración: Caso de Uso Configurar Captura de Datos

**Configurar Captura de Datos (Conexión BD)**

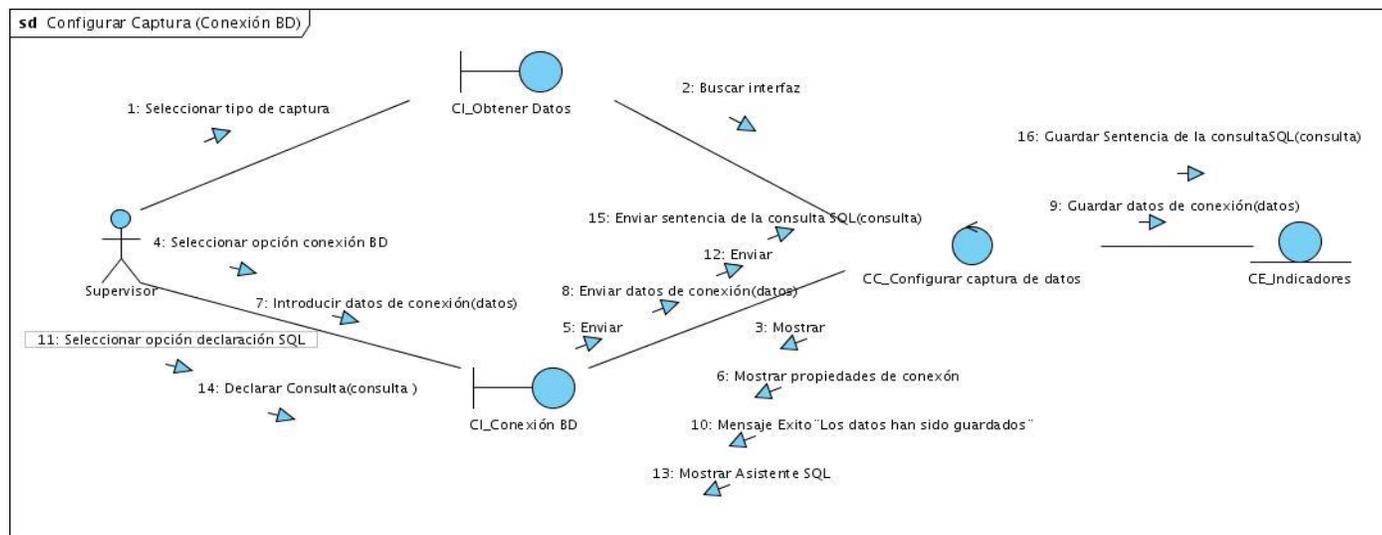


Figura 25.- Diagrama de colaboración: Configurar Captura de Datos (Conexión BD)

**Configurar Captura de Datos (Crear fórmula)**

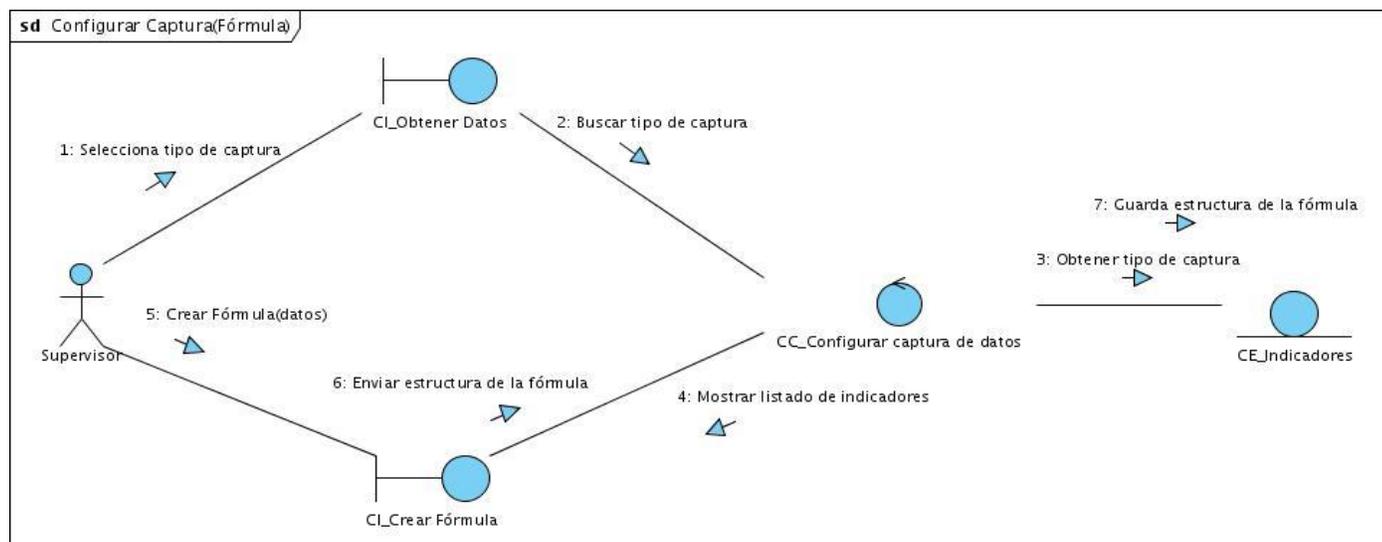


Figura 26.- Diagrama de colaboración: Configurar Captura de Datos (Crear fórmula)

**Configurar Captura de Datos (Cargas de Archivos)**

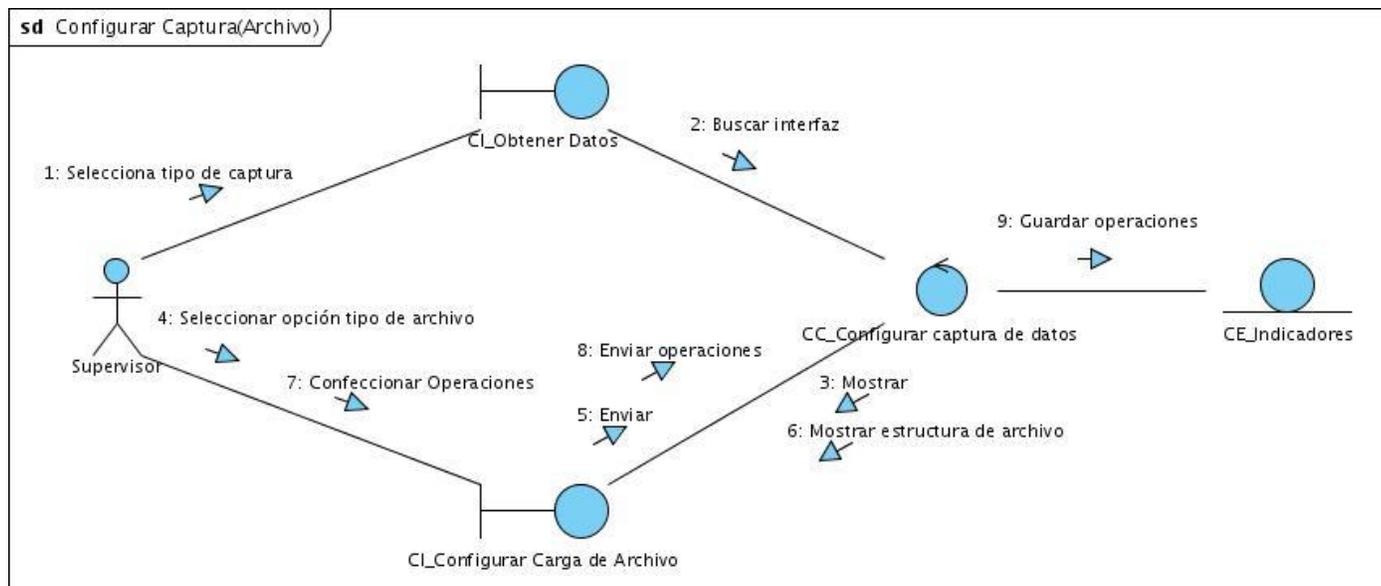


Figura 27.- Diagrama de colaboración: Configurar Captura de Datos (Carga de archivos)

❖ **Diagrama de colaboración: Caso de Uso Asociar a objetivos**

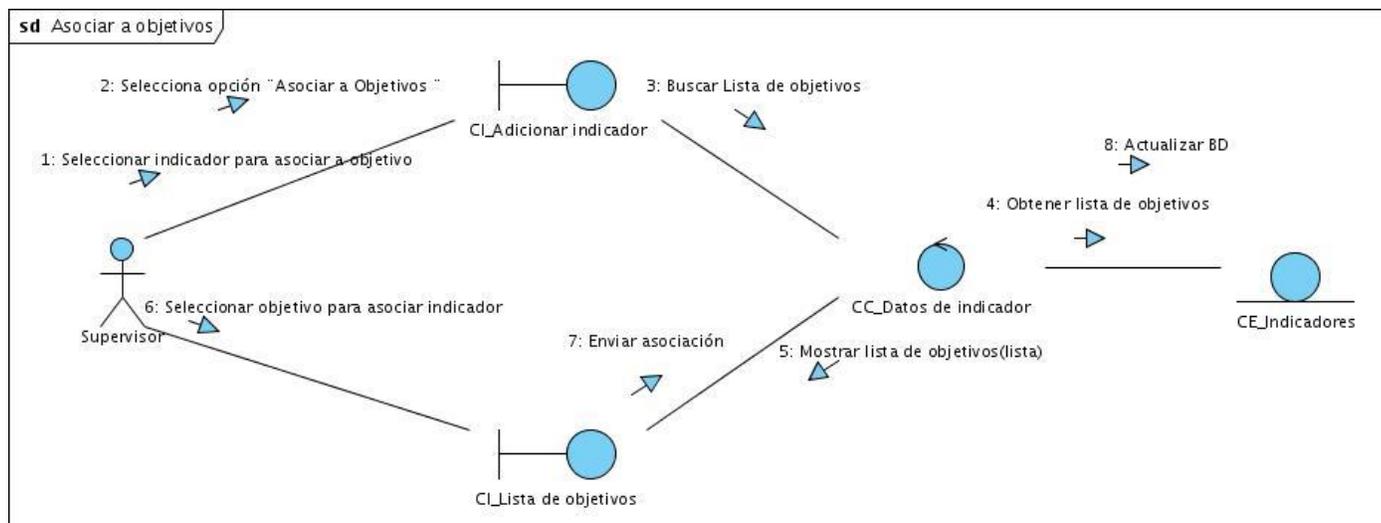


Figura 28.- Diagrama de colaboración: Asociar a objetivos

#### Anexo 4: Diagramas de Clases del Diseño

##### Diagrama de Clases del Diseño: Asociar a Objetivos

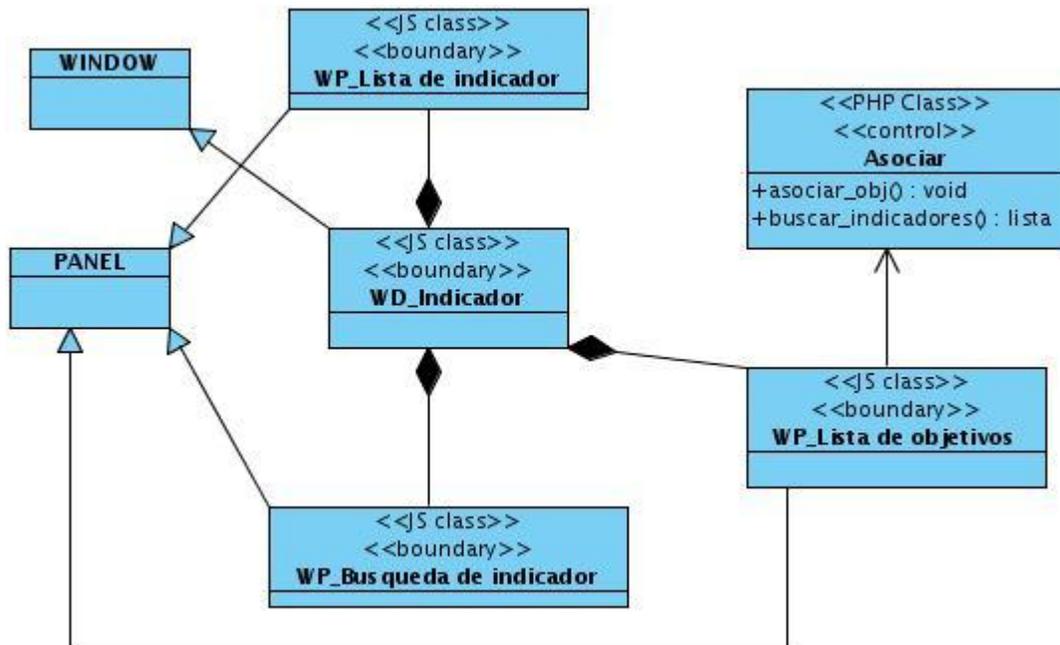


Figura 29.- Diagrama de Clases del Diseño: Asociar a objetivos

## Diagrama de Clases del Diseño: Capturar Datos

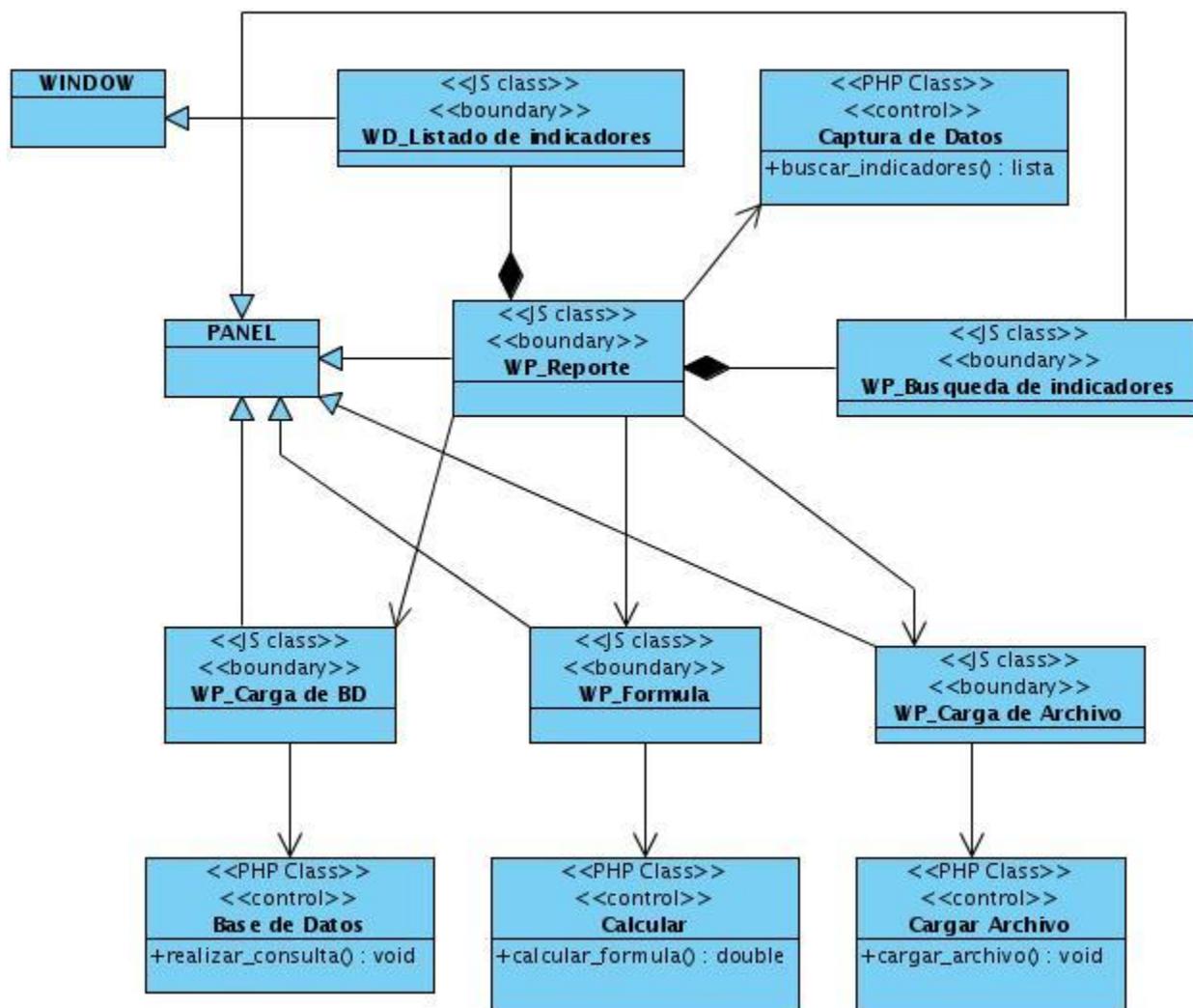


Figura 30.- Diagrama de Clases del Diseño: Capturar Datos

## Diagrama de Clases del Diseño: Configurar Captura de Datos

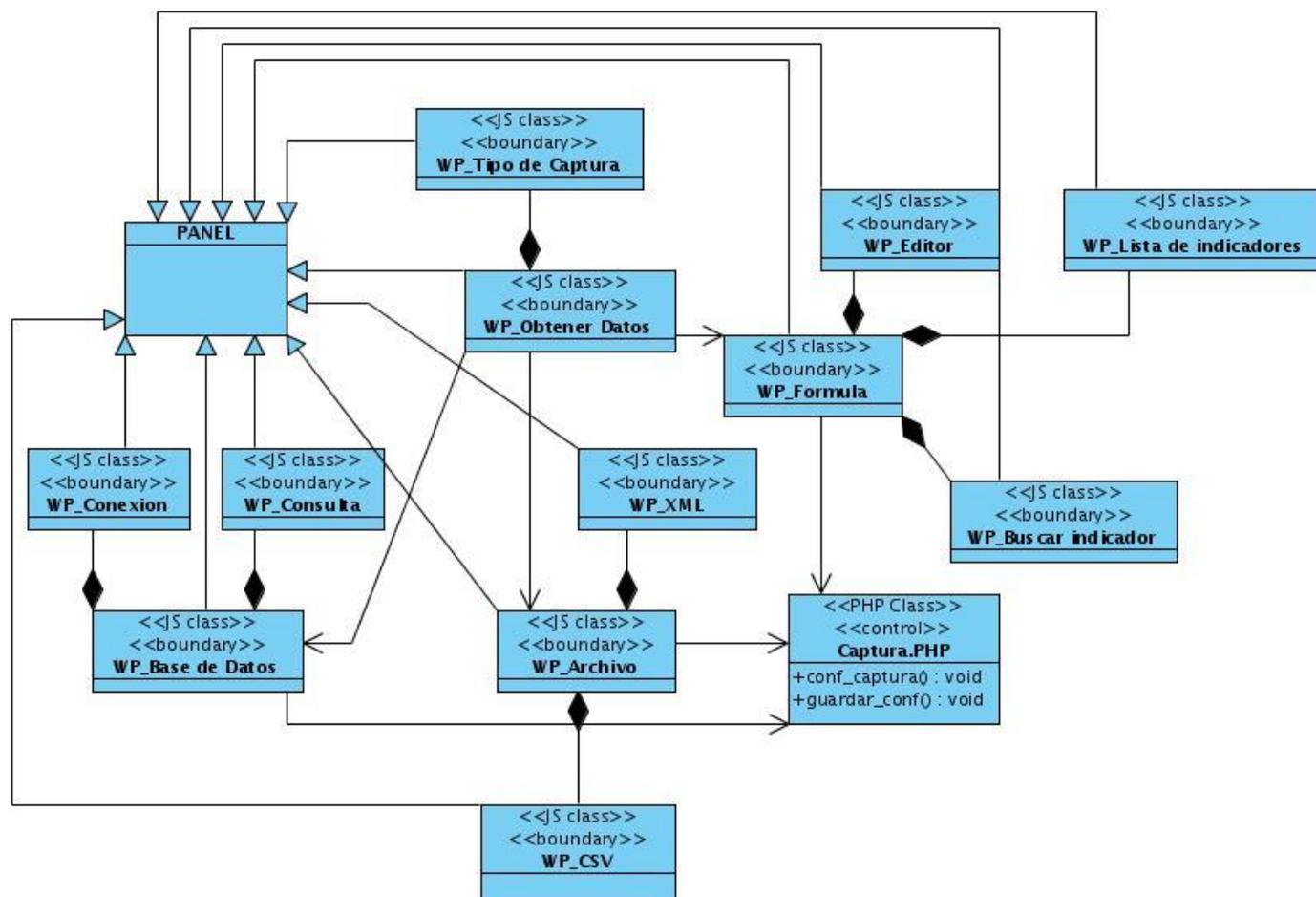


Figura 31.- Diagrama de Clases del Diseño: Configurar Captura de Datos

## Anexo 5: Diagramas de Secuencia

### ❖ Diagramas de Secuencia del CU Capturar Datos

#### Capturar Datos (Archivo)

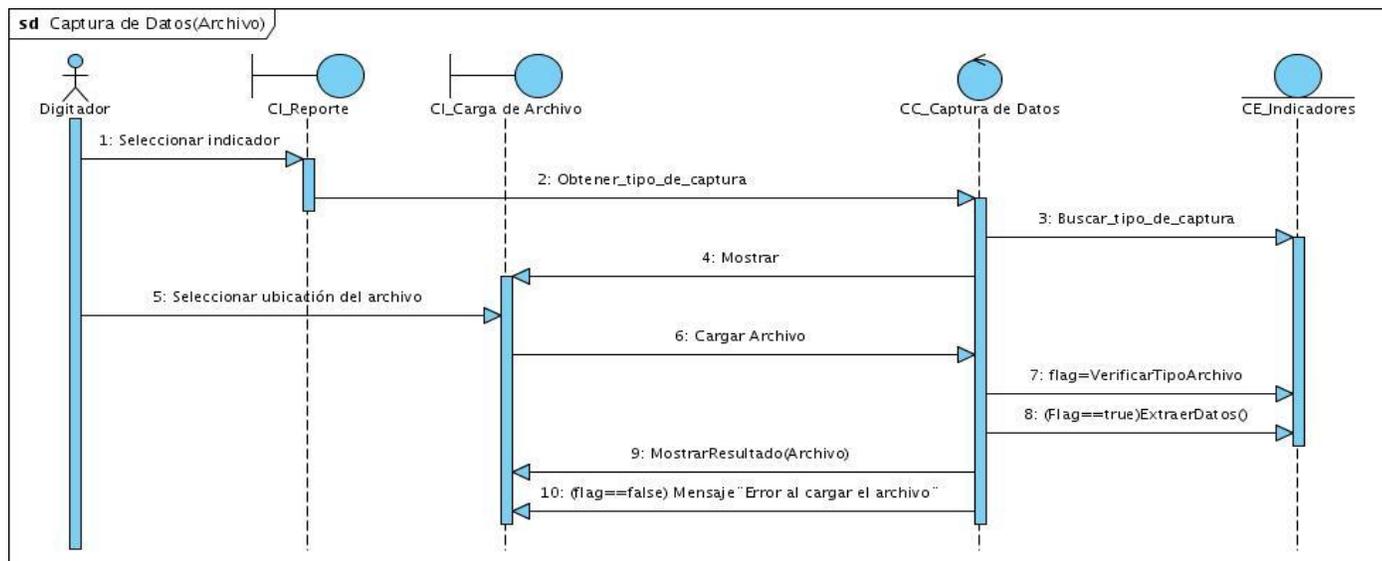


Figura 32.- Diagramas de Secuencia Capturar Datos (Archivo)

#### Capturar Datos (Base de Datos)

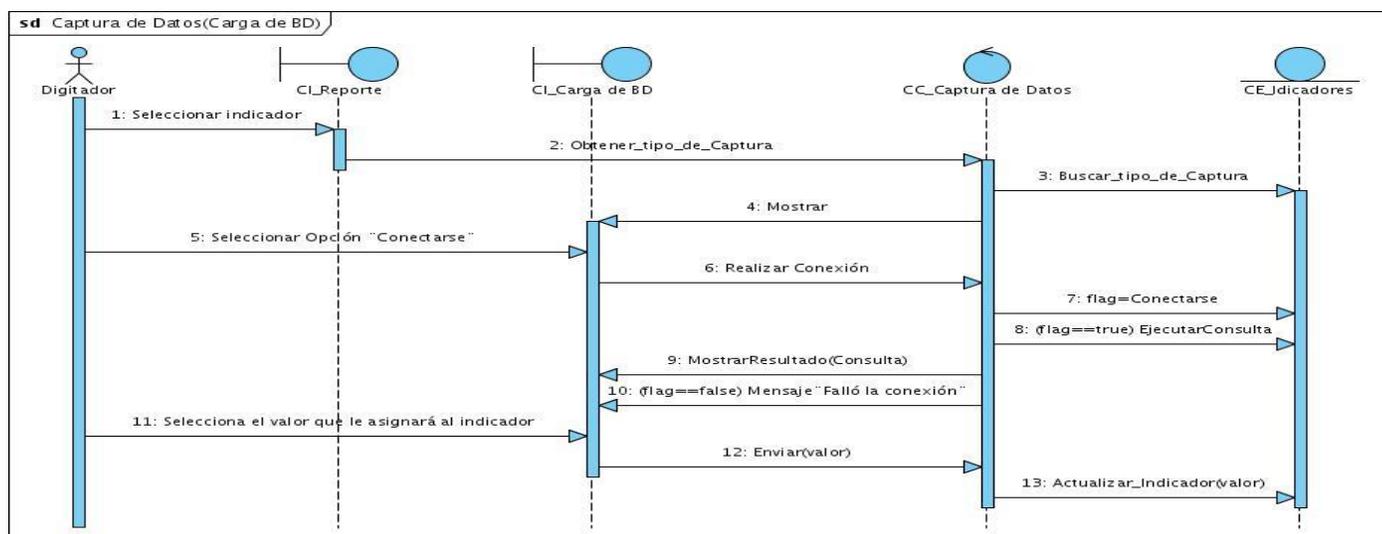


Figura 33.- Diagramas de Secuencia Capturar Datos (Base de Datos)

## Capturar Datos (Fórmula)

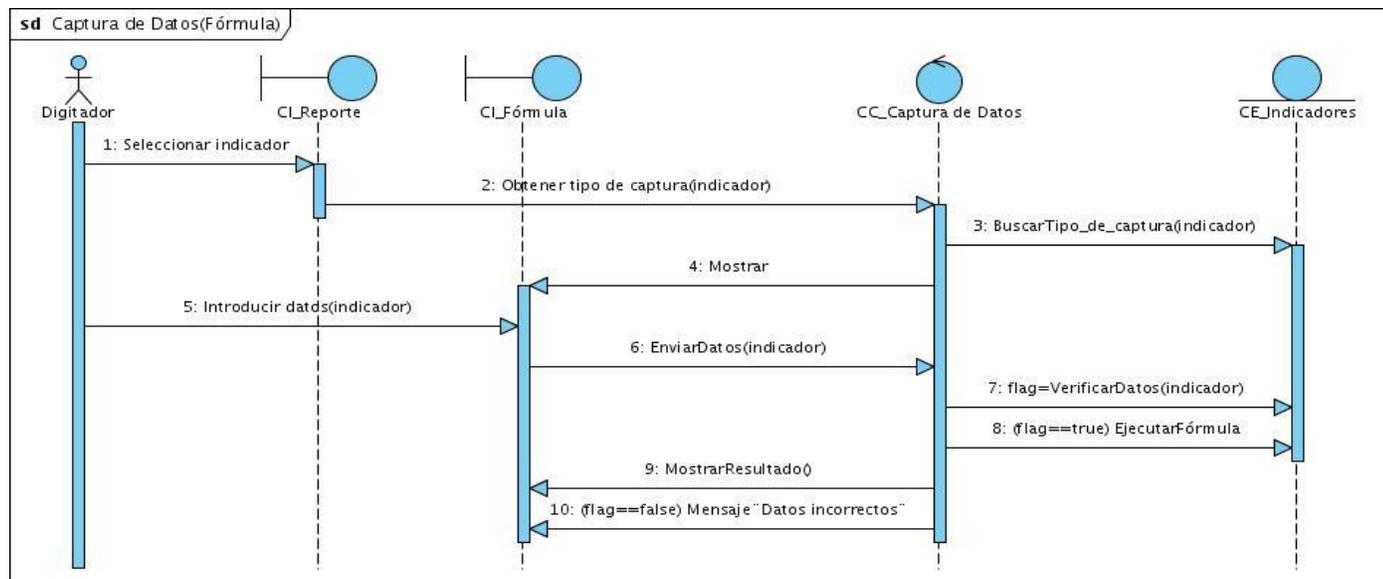


Figura 34.- Diagramas de Secuencia Capturar Datos (Fórmula)

## ❖ Diagramas de Secuencia del CU Configurar Captura de Datos

### Configurar Captura de Datos (Archivo)

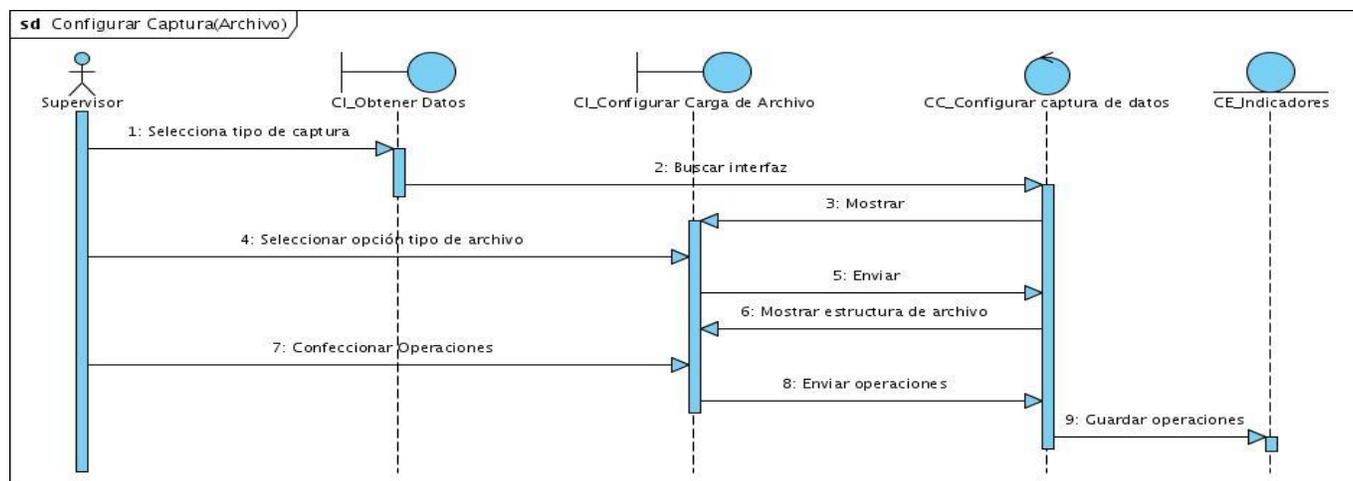


Figura 35.- Diagrama de Secuencia Configurar Captura de Datos (Archivo)

## Configurar Captura de Datos (Base de Datos)

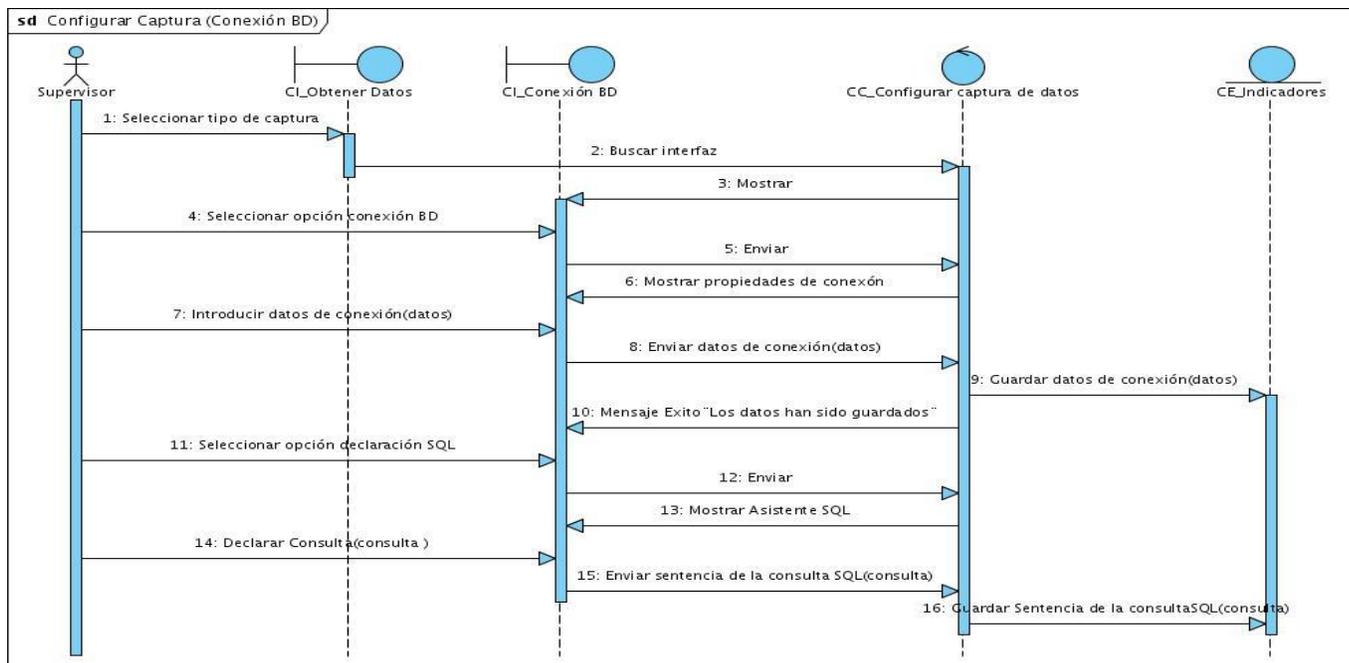


Figura 36.- Diagrama de Secuencia Configurar Captura de Datos (Base de Datos)

## Configurar Captura de Datos (Fórmula)

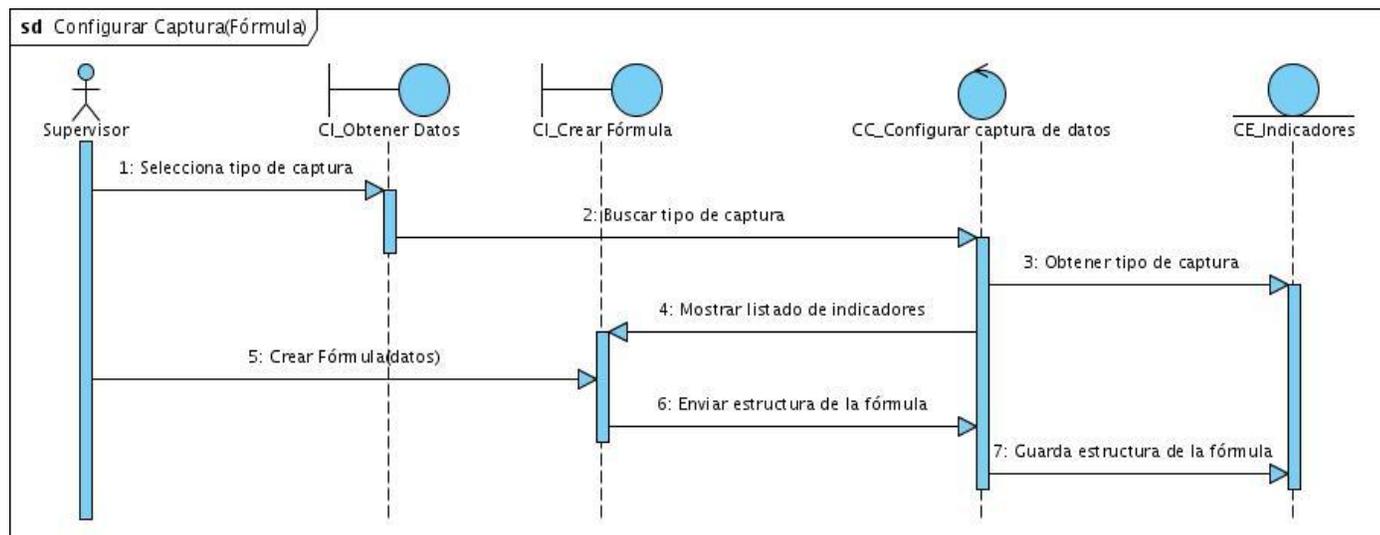


Figura 37.- Diagrama de Secuencia Configurar Captura de Datos (Fórmula)

## ❖ Diagramas de Secuencia del CU Asociar a Objetivos

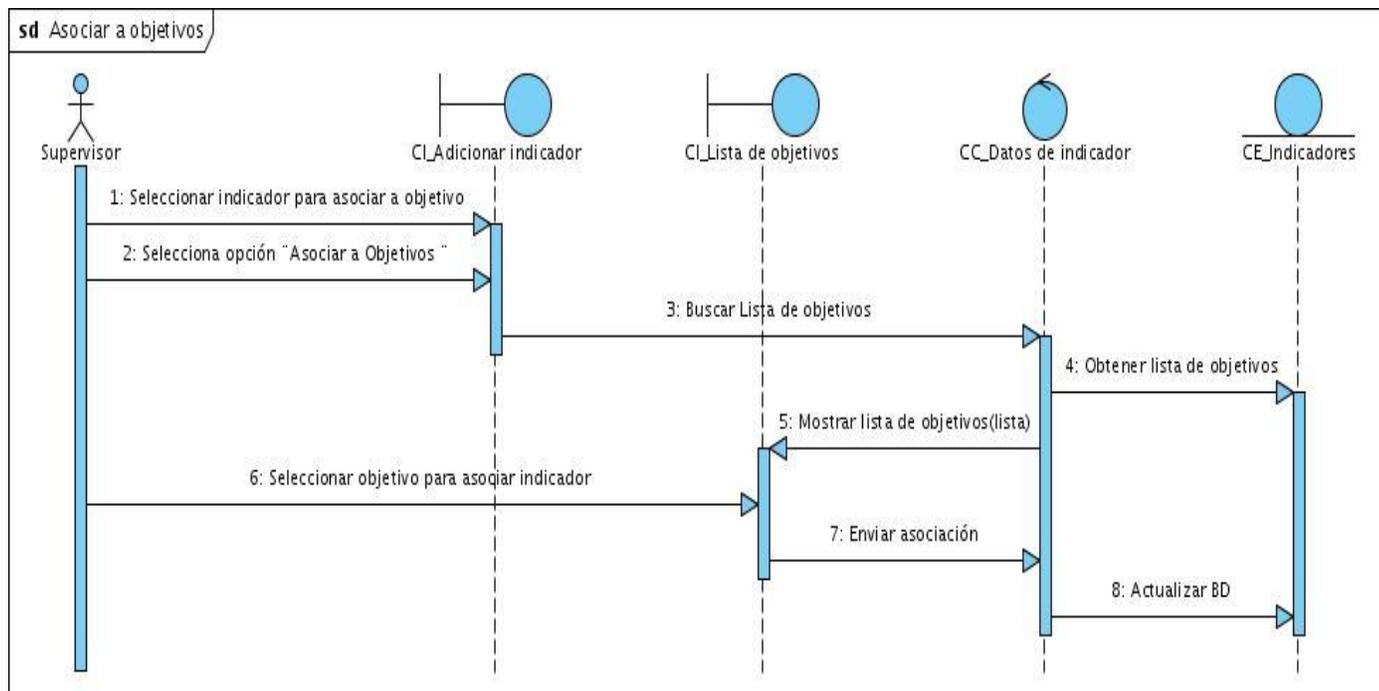


Figura 38.- Diagramas de Secuencia del CU Asociar a Objetivos

**Anexo 6: Descripción de las tablas de la Base de Datos**

## ❖ Descripción de las tablas de la Base de Datos

Las tablas de la base de datos permiten organizar y agrupar los datos según las características o los principios comunes.

**Tabla Indicadores**

Nombre: Indicador		
Descripción: Almacena todos los indicadores		
Atributo	Tipo	Descripción
código	integer	id del indicador de la tabla

nombre	varChar	nombre del indicador
tipo	varChar	almacena un tipo de indicador posible
periodicidad	varChar	frecuencia con la que se mide el indicador
unidad_medida	varChar	forma de medición que se utiliza
prioridad	varChar	prioridad que tiene el indicador
estado	varChar	estado en que se encuentra el indicador
FormaDatoidentificador	integer	id de la Forma de Dato a la que pertenece el indicador

Tabla 31.- Descripción de la tabla: Indicadores

### Tabla Objetivo

<b>Nombre: Objetivo</b>		
Descripción: Almacena todos los objetivos existentes en la base de datos		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_obj	integer	id del objetivo en cuestión
nombre	varChar	nombre del objetivo
descripción	varChar	describe la función del objetivo
plazo	varChar	tiempo para el cumplimiento del objetivo
prioridad	varChar	prioridad que tiene el objetivo

Tabla 32.- Descripción de la tabla: Objetivo

### Tabla Indicador\_objetivo

<b>Nombre: Indicador_objetivo</b>
Descripción: Esta tabla se crea por la relación muchos-muchos existentes entre la tabla indicador y la tabla objetivo.

Atributo	Tipo	Descripción
id_obj	integer	id del objetivo en cuestión
código	integer	id del indicador de la tabla

Tabla 33.- Descripción de la tabla: *Indicador\_objetivo*

### Tabla FormaDato

<b>Nombre: FormaDato</b>		
Descripción: Almacena todos los valores de las formas de obtención de datos: Archivo, Base de Datos y Fórmula.		
Atributo	Tipo	Descripción
identificador	integer	guarda el id de la forma de dato
tipo	integer	tipo de la forma de dato
fórmula	varChar	guarda la fórmula para una forma de dato del tipo: Fórmula
descripción	varChar	describe la función de la fórmula
password	varChar	guarda la contraseña de la conexión a la Base de Datos
servidor	varChar	guarda el ip donde se encuentra la base de datos
puerto	integer	guarda el puerto para la conexión a la base de datos
user	varChar	guarda el usuario de la conexión a la base de datos
base_datos	varChar	guarda el nombre de la base de datos a la que se conecta
consulta	varChar	guarda la sentencia SQL

formula	varChar	guarda la fórmula para una forma de dato del tipo: Archivo
TipoBDId_tipobd	integer	guarda el id del tipo de base de datos
TipoArchivold_tipoarch	integer	guarda el id del tipo de archivo

Tabla 34.- Descripción de la tabla: FormaDato

### Tabla Variable

Nombre: Variable		
Descripción: Tiene las variables existentes en la base de datos		
Atributo	Tipo	Descripción
id_variable	integer	id de la variable
nombre	varChar	nombre de la variable
tipo	varChar	tipo de dato de la variable
FormaDatoidentificador	integer	guarda el id de la forma de dato a la que pertenece la variable

Tabla 35.- Descripción de la tabla: Variable

### Tabla TipoBD

Nombre: TipoBD		
Descripción: Almacena todos los tipos de Base de Datos		
Atributo	Tipo	Descripción
id_tipobd	integer	id del tipo de base de datos
tipoBD	varChar	nombre del tipo de base de datos

Tabla 36.- Descripción de la tabla: TipoBD

**Tabla TipoArchivo**

<b>Nombre: TipoArchivo</b>		
Descripción: Almacena todos los tipos de Archivos		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_tipoarch	integer	id del tipo de archivo
tipoArch	varChar	guarda el nombre del tipo de archivo

*Tabla 37.- Descripción de la tabla: TipoArchivo***Anexo 7: Descripción de las clases de análisis****Caso de Uso: Capturar Datos****Clases interfaz**

CI\_Reporte: clase de interfaz que le permite al Digitador realizar la captura de datos de los indicadores seleccionados.

CI\_Fórmula: esta interfaz posee los componentes necesarios para visualizar los datos capturados luego de una selección, también posibilita introducir alguna variable de ser necesario.

CI\_Carga de BD: esta clase interfaz le mostrará al Digitador el resultado de una consulta predefinida para capturar los datos de un indicador antes seleccionado.

CI\_Carga de Archivo: clase de interfaz que le posibilitará al Digitador capturar los datos mediante la carga de un archivo XML ó CSV.

**Clase control**

CC\_Capturar datos: clase controladora que se encarga de realizar todos los procesos de captura de datos solicitados por la interfaz CI\_Reporte.

**Clase entidad**

CE\_Indicadores: clase de entidad que contiene todos los datos de los indicadores.

## Caso de Uso: **Configurar Captura de Datos**

### **Clases interfaz**

CI\_Obtener Datos: clase de interfaz que le permite al Supervisor definir el tipo de captura de datos de los indicadores.

CI\_Conexión BD: esta interfaz posee los componentes necesarios para configurar las conexiones a las base de datos y predefinir las consultas necesarias para que el Digitador pueda realizar la captura de datos.

CI\_Crear Fórmula: interfaz que le posibilita al Supervisor configurar fórmulas para calcular el valor de indicadores.

CI\_Configurar Carga de Archivo: clase de interfaz que permite configurar la lectura y carga de archivos XML y CSV.

### **Clase control**

CC\_Configurar Capturar de Datos: clase controladora que se encarga de todos los procesos relacionados con la configuración de la captura de datos.

### **Clase entidad**

CE\_Indicadores: clase de entidad que contiene todos los datos de los indicadores.

## Caso de Uso: **Gestionar Indicadores**

### **Clases interfaz**

CI\_Añadir Indicador: clase de interfaz que le permite al Supervisor añadir nuevos indicadores al sistema.

CI\_Modificar Indicador: interfaz que posee los componentes necesarios para modificar los datos de los indicadores.

CI\_Eliminar Indicador: esta interfaz le da la posibilidad al Supervisor de eliminar los indicadores no deseados.

### **Clase control**

CC\_Gestionar Indicador: clase controladora que se encarga de realizar los procesos de la gestión de indicadores solicitados por la interfaz CI\_Añadir Indicador, CI\_Modificar Indicador, CI\_Eliminar Indicador.

### **Clase entidad**

CE\_Indicadores: clase de entidad que contiene todos los datos de los indicadores.

Caso de Uso: **Asociar a objetivos**

### **Clases interfaz**

CI\_Añadir Indicador: clase de interfaz que le permite al Supervisor añadir nuevos indicadores al sistema y le brinda la opción de asociar o no estos indicadores a objetivos.

CI\_Lista de objetivos: interfaz que muestra la lista de objetivos e indicadores para poderlos asociar.

### **Clase control**

CC\_Datos de indicador: clase controladora que se encarga de realizar los procesos de asociación de indicadores.

### **Clase entidad**

CE\_Indicadores: clase de entidad que contiene todos los datos de los indicadores.

## GLOSARIO DE TÉRMINOS

**Actor:** un actor es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio y el sistema interactúan.

**Caso de uso:** es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

**Herramientas CASE:** conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software (Investigación Preliminar, Análisis, Diseño, Implementación e Instalación.).

**Modelo del Dominio:** descripción mediante un diagrama de clases UML donde se especifican las principales clases conceptuales que pueden intervenir en nuestro sistema, estos representarán los objetos que existen o eventos que suceden en el entorno en el que trabajará el sistema.

**RAM:** memoria RAM (Memoria de Acceso Aleatorio) es donde el computador guarda los datos que está utilizando en el momento presente. El almacenamiento es considerado temporal porque los datos y programas permanecen en ella mientras que la computadora este encendida o no sea reiniciada.

**Requerimientos:** condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales.

**Software:** palabra en inglés utilizada para indicar a los programas de computadoras, a las aplicaciones.

**UML:** es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.