

Universidad de las Ciencias Informáticas
Facultad 2



Título: “Módulo gestión de estrategias e indicadores del Cuadro de Mando Integral.”

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

Autor(es):

Yanetsi Millet Lombida
Yaiset Piloto Arregui

Tutor(es):

Lic. Alain Pérez Balart
Dr. Guillermo Ronda Pupo

Ciudad de la Habana
Julio del 2007

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

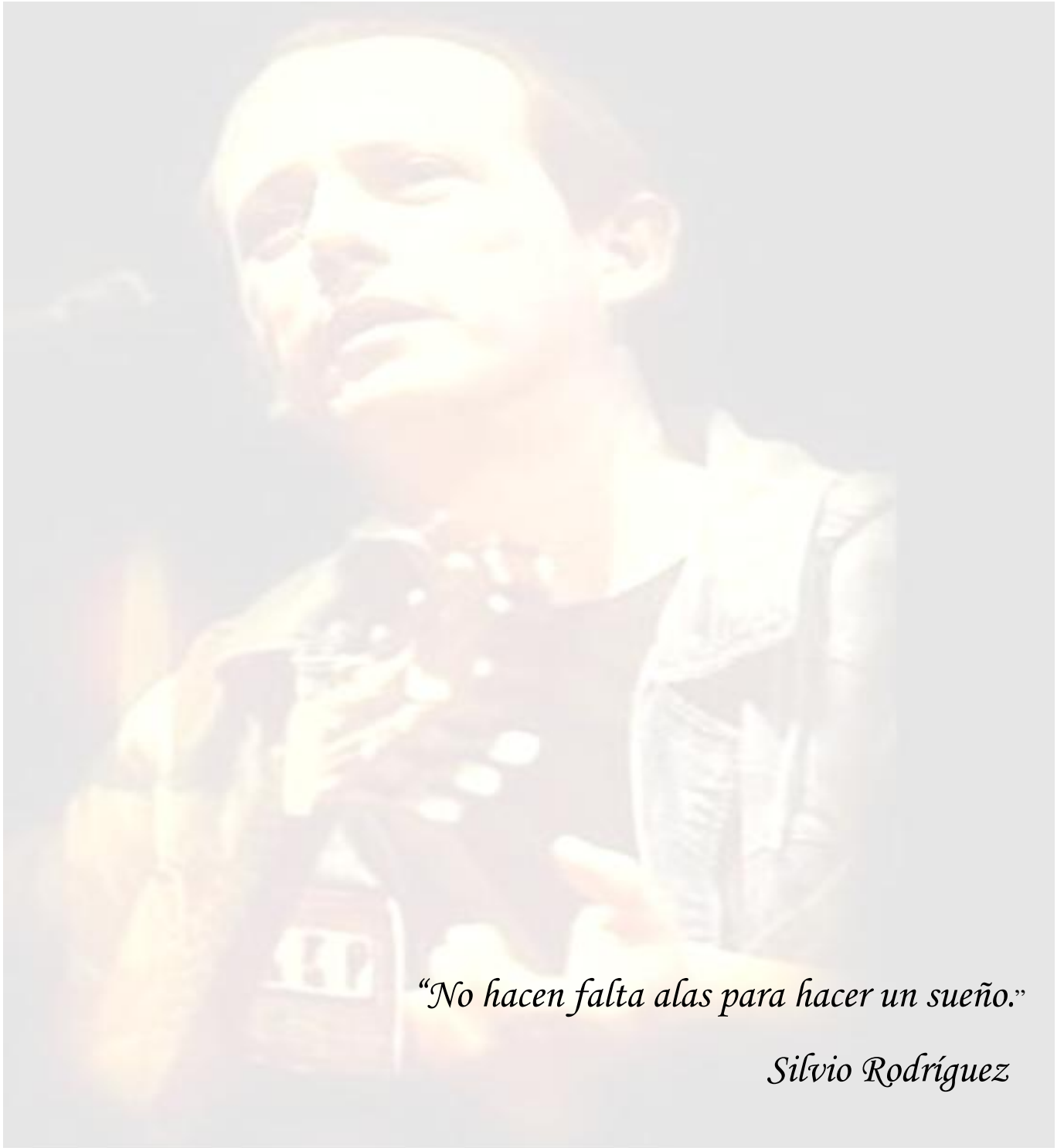
Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Firma del Autor
Yaiset Piloto Arregui

Firma del Autor
Yanetsi Millet Lombida

Firma del Tutor
Lic. Alain Pérez Balart

Firma del tutor
Dr. Guillermo Ronda Pupo



“No hacen falta alas para hacer un sueño.”

Silvio Rodríguez

Agradecimientos

De forma sencilla, pero muy sincera, hoy quiero agradecer, en 1º lugar a mis padres y hermano, pues por ellos, hoy soy todo cuanto soy.

A mi hermana, por el apoyo incondicional de tantos años.

También merecen estar aquí mi familia, incluyendo a mi abuelo (papi), que en el lugar que se encuentre sabe que este triunfo es suyo.

A mi novio, Reinier (TATI), por la comprensión y apoyo incondicional que me ha brindado siempre, por haber esperado tanto tiempo por este resultado. Gracias por amarme por encima de todas las cosas.

A mi padrino, Yosdado (Diosda) por sus consejos, confianza y todo lo que ha sembrado en mí.

Gracias a mis compañeros de carrera, fundamentalmente a Adi y Yadi, por soportarme y apoyarme siempre.

A mis compañeros de Tesis, por sobre todos a mi compañera Yaiset y con signos de admiración para Yasser (sin él hubiera sido casi imposible).

A Susana y Wilfredo cuya experiencia fue esencial para poder comenzar y continuar este trabajo.

A mi tutor Alain por habernos dedicado parte de su tiempo y a Iván por la inspiración y fuerza que nos brindó.

A mis profesores por su donación de paciencia, conocimientos y virtudes.

Agradezco a nuestro Comandante y a la Revolución por esta oportunidad que nos dieron.

A todos, ¡MUCHAS GRACIAS!

Yanetsi Millet Lombida

Agradecimientos

A mi mamá por apoyarme cada vez que lo he necesitado. Por escucharme llorar el día antes de cada examen. Gracias mami, confiaste en mi y aquí me tienes, soy ingeniera.

A mi papá por haber seguido cada paso de mi carrera, por enseñarme a confiar en mi misma. Por resultar ser motivo de inspiración tantas veces en mi vida. Gracias de corazón papi.

A mis queridos abuelitos Dalia y Alfredo que han sido como mis propios padres y que nunca han dejado de quererme como a una hija.

A mis hermanitos Yinet, Adrián y Dayán gracias por soportarme siempre. Ustedes saben que los quiero.

A mi novio, Lisvan, por quererme tanto, por su apoyo incondicional, por estar siempre en el lugar exacto, donde me hizo falta que estuviese. Mi corazón es tuyo: ¡Te Amo!

A mis compañeros de grupo con los que he vivido cinco años de momentos malos y buenos, sin ustedes llegar aquí no hubiese sido tan lindo. Nunca los olvidaré.

A mis compañeros de proyecto con los que he compartido todo este tiempo y que tanto me han ayudado. Especialmente a mi compañera de tesis Millet y a Yasser.

A mi tutor Alain por habernos apoyado, a Iván por conversar tanto con nosotros y darnos tanta confianza.

A mi amiga Elizabeth, a Tere, a Gilbe, por estar entre las personas que me quieren, y que me han enseñado que en la vida cuando se quiere con esfuerzo y sacrificio, se puede. Por desear que cada día crezca como persona, como profesional, como amiga.

A todos, muchas gracias. Los quiero mucho.

Yaiset Piloto Arregui

Dedicatoria

*A mis padres
y a mi hermano,
por ser mis seres más queridos.*

Yanetsi Millet Lombida

*A mis maravillosos padres, por ser mis maestros en el camino de la vida.
A la Revolución y a Fidel.*

Yaiset Piloto Arregui

RESUMEN

El desarrollo de la informática y el aumento de su impacto social, optan por incorporar aplicaciones que gestionen su información cada vez más en las instituciones u organizaciones, logrando una mayor dinámica en sus procesos de negocio.

Este trabajo de diploma brinda una solución automatizada, flexible y única, a todo el flujo de gestión de estrategias e indicadores de un Cuadro de Mando Integral (CMI) que anteriormente se hacía manual.

Para darle solución a estos problemas se decidió desarrollar una aplicación Web, basada en: tecnología PHP 5.1, con gestor de base de datos MySQL 5.0, metodología de desarrollo RUP, lenguaje de modelado visual UML, y como herramienta Case Rational Rose Enterprise Edition 2003.

El sistema propuesto contribuye a que toda la información sea accesible, disminuyendo el tiempo en las búsquedas de información. Además permite mejorar las condiciones de trabajo del personal de dicha empresa, evitándoles el agotamiento y demora que produce el procesamiento manual de la información al contribuir positivamente en el almacenamiento y control de esta.

TABLA DE CONTENIDOS

RESUMEN.....	IX
INTRODUCCIÓN	1
Capítulo 1: FUNDAMENTACIÓN TEÓRICA	3
1.1 Introducción	3
1.2 Cuadro de Mando Integral	3
1.2.1 Surgimiento del Cuadro de Mando Integral.....	4
1.2.2.1 Perspectiva Financiera	6
1.2.2.2 Perspectiva Cliente.....	7
1.2.2.3 Perspectiva Procesos Internos.....	8
1.2.2.4 Perspectiva Aprendizaje y Crecimiento	9
1.3 Sistemas automatizados existentes.....	10
1.4 Necesidades del Cuadro de Mando Integral en Cuba	12
1.5 Diseño metodológico de la investigación	13
1.6 Tendencias y Tecnologías actuales	13
1.6.1 Las aplicaciones Web.	14
1.6.2 Lenguajes de programación para la Web.....	15
1.6.2.1 PHP 5.1	15
1.6.2.2 JavaScript.....	17
1.6.3 Sistemas de Gestión de Base de Datos.....	17
1.6.3.1 MySQL 5.0	18
1.6.4 Servidor Web Apache 2.2	19
1.6.5 Patrones de arquitectura.....	20
1.6.5.1 Modelo Vista Controlador	20
1.6.6 Proceso de Desarrollo.	21
1.6.6.1 RUP	21
1.6.6.2 UML.....	22
1.6.7 Herramienta utilizadas	23
1.6.7.1 Rational Rose	23
1.6.7.2 Diseño de interfaz: Dreamweaver 8.0.....	24
1.6.7.3 Zend Studio 5.1	25
1.6.7.4 SQL Manager for MySQL 2005 versión 3.7.0.1	26
1.6.7.5 Framework Code Igniter 1.5	26
1.6.7.6 Framework AJAX: YUI.....	28
1.7 Conclusiones	28
Capítulo 2: CARACTERÍSTICAS DEL SISTEMA	29
2.1 Introducción	29
2.2 Objeto de estudio	29
2.2.1 Objeto de automatización.....	29

2.2.2 Información que se maneja	29
2.2.3 Propuesta del sistema	30
2.3 Modelo del negocio.....	30
2.3.1 Proceso de Negocio.....	30
2.3.2 Reglas del negocio a considerar.....	30
2.3.3 Actores del negocio.....	31
2.3.4 Diagrama de Casos de Uso del Negocio.....	31
2.3.5 Trabajadores del negocio	32
2.3.6 Descripción textual de los casos de uso del negocio. (VER ANEXO 1).....	33
2.3.7 Diagrama de actividades. (VER ANEXO 2).	33
2.3.8 Diagrama de clases del Modelo de Objeto. (VER ANEXO 3).....	33
2.4 Especificación de los requisitos de software.....	33
2.4.1 Definición de los requerimientos funcionales.....	33
2.4.2 Definición de los requerimientos no funcionales.....	34
2.5 Modelo del Sistema.....	35
2.5.1 Actores del sistema	35
2.5.2 Diagrama de Casos de Uso del Sistema.....	36
2.5.3 Descripción textual de los casos de uso del sistema. (VER ANEXO 4).	37
2.6 Conclusiones	37
Capítulo 3: ANÁLISIS Y DISEÑO DEL SISTEMA.	38
3.1 Introducción	38
3.2 Modelo de Análisis	38
3.2.1 Diagrama de clases del análisis.....	38
3.3 Modelo de Diseño	41
3.3.1 Diagrama de clases del diseño	41
3.4 Diagramas de interacción.....	47
3.4.1 Diagramas de colaboración. (VER ANEXO 5).	47
3.5 Descripción de las clases del diseño	47
3.6 Modelo de datos	55
3.7 Descripción de las tablas de la Base de Datos	57
3.8 Definiciones de diseño	60
3.8.1 Patrones	60
3.8.1.1 Patrones GRASP	60
3.8.1.2 Patrones de diseño Web	61
3.9 Tratamiento de errores	62
3.10 Seguridad	62
3.11 Interfaz de usuario.....	62
3.12 Concepción de la ayuda	63
3.13 Conclusiones	63
Capítulo 4: IMPLEMENTACIÓN	64
4.1 Introducción	64

4.2 Diagrama de despliegue	64
4.3 Diagrama de componentes	65
4.4 Conclusiones	66
Capítulo 5: ESTUDIO DE FACTIBILIDAD	67
5.1 Introducción	67
5.2 Planificación basada en casos de uso.....	67
5.2.1 Identificar los puntos de casos de usos desajustados	67
5.2.2 Cálculo de UAV	67
5.2.3 Cálculo de UUCW	68
5.3 Ajustar los puntos de casos de uso.....	68
5.3.1 Cálculo de TCF	68
5.3.2 Cálculo de EF.....	69
5.4 Calcular esfuerzo FT implementación	71
5.4.1 Cálculo de CF.....	71
5.5 Esfuerzo de todo el proyecto.....	72
5.6 Beneficios tangibles e intangibles.....	72
5.7 Análisis del costo	73
5.8 Conclusiones	73
CONCLUSIONES.....	75
RECOMENDACIONES.....	76
REFERENCIAS BIBLIOGRÁFICAS.....	77
BIBLIOGRAFÍA.....	78
ANEXOS	81
Anexo 1. Descripción textual de los casos de uso del negocio.....	81
Anexo 2. Diagrama de actividades.	83
Anexo 3. Diagrama de clases del Modelo de Objetos.	85
Anexo 4. Descripción textual de los casos de uso del sistema.	86
2.5.3.1 Caso de uso “Gestionar Estrategia”	86
2.5.3.2 Caso de uso “Gestionar Indicadores”	89
2.5.3.3 Caso de uso “Calcular indicador”	92
2.5.3.3 Caso de uso “Gestionar Fórmula”	94
2.5.3.4 Caso de uso “Gestionar Unidad de Medida”	98
Anexo 5. Diagrama de clases del Modelo de Objetos.	101
GLOSARIO DE TÉRMINOS.....	106

INTRODUCCIÓN

La administración surge como ciencia a principio del siglo pasado y desde su inicio está marcada por una gran preocupación de los administradores por el control en las empresas. Muchos han sido los criterios de los teóricos de la administración sobre lo que esta es, su objetivo y sus funciones, pero ninguna de las teorías ha pasado por alto la importancia del control empresarial, aunque los diferentes conceptos que de él se han desarrollado han estado muy vinculados a las concepciones sobre las empresas y el papel de sus administradores en el momento en que han desarrollado sus teorías.

Aunque son evidentes los aportes de las corrientes administrativas al control de gestión, el desarrollo de este término no ha ido aparejado al gran desarrollo que ha tenido en las últimas décadas la ciencia administrativa, siendo así que Kaplan y Johnson, (1987) plantean que *“el control de gestión tradicional detuvo su desarrollo en 1925, momento en el cual todos los procedimientos contables, aún en uso ya existían: presupuestos, costos estándar, el modelo Du Pont, etc.”*

A medida que se han tratado de adaptar los sistemas de control a las necesidades de los directivos y de las empresas en la actualidad, se evidencia una mayor preocupación por la descentralización, la participación de todos los miembros de la organización y por concebir al control como un proceso íntimamente ligado a la estrategia y a los objetivos en el corto y largo plazo.

La necesidad de que el control de gestión evolucione hacia una concepción proactiva y que conjugue los aspectos internos y externos de la empresa aumenta a medida que estas tratan de desenvolverse en un entorno en el que las condiciones competitivas que le son impuestas, exigen sacar el mayor provecho posible de la información interna y externa para lograr el desarrollo de la estrategia. Con el objetivo de resolver estas cuestiones de vital importancia para la empresa, surge en la década de los 90, el Cuadro de Mando Integral (CMI) de Kaplan y Northon.

Actualmente, Cuba posee muy pocos CMI implantados como herramienta para la gestión empresarial, es por eso, que esta investigación surge como necesidad de dar solución a la insuficiencia de un software que permita aplicar las tecnologías de la informática y las comunicaciones para agilizar la generación y el flujo de información en un Cuadro de Mando Integral (CMI), por lo que el **problema científico** es ¿cómo apoyar el servicio empresarial cubano mediante un sistema automatizado desde el módulo gestión de estrategias e indicadores del Cuadro de Mando Integral (CMI) para contribuir a elevar la efectividad de los mismos?

El **objeto de la investigación** se definió como la gestión de estrategias e indicadores de un Cuadro de Mando Integral (CMI) y su **campo de acción** se enmarcó en la gestión de estrategias e indicadores en las empresas cubanas.

Se trazó como **objetivo general** automatizar el módulo gestión de estrategias e indicadores de las perspectivas del Cuadro de Mando Integral (CMI) para apoyar el servicio empresarial cubano.

Para cumplir los objetivos se desarrollaron las siguientes **tareas**:

- Investigación profunda de las perspectivas de un Cuadro de Mando Integral.
- Realización de un estudio detallado de sistemas automatizados de un Cuadro de Mando Integral tanto a nivel internacional como nacional.
- Estudio de las tendencias y tecnologías actuales que mejor se ajusta al desarrollo de este sistema automatizado.

Aportes prácticos

- Se diseña, fundamenta e implementa un sistema automatizado del Cuadro de Mando Integral con carácter sistémico y funcional en organizaciones en el país.
- Posibilidad de generalización de su aplicación en diversas organizaciones, lo cual contribuyó a elevar el nivel de efectividad de la implantación de la dirección estratégica conllevando al mejoramiento del desempeño de las mismas.
- Rapidez en las búsquedas de información por parte del personal de apoyo disminuyendo su carga de trabajo.

Novedad Científica de la tesis

- Implementación de un modelo específico de dirección estratégica, que considera la integración de los niveles estratégicos, tácticos y operativos, a partir de la fusión de varias herramientas de dirección.
- Determinación de las regularidades, principios y premisas que rigen el proceso de implantación y ejecución de la dirección estratégica de una organización de seguridad y protección.

Capítulo 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Este capítulo contiene los principales problemas que fundamentan la propuesta de solución, y los objetivos generales que se persiguen. Brinda un enfoque general de sistemas automatizados y el análisis comparativo de las soluciones existentes con la propuesta dada en este trabajo. Se describen las tecnologías actuales de desarrollo utilizadas para el análisis, diseño e implementación del sistema sobre las cuales se apoya la propuesta.

1.2 Cuadro de Mando Integral

El Cuadro de Mando Integral (CMI) es una herramienta muy útil para describir, aplicar y dirigir la visión y estrategia futura de una organización; es un sistema de medición que va más allá de la perspectiva financiera con la que los gerentes acostumbran a evaluar la marcha de una empresa. Muestra continuamente cuando una compañía y sus empleados alcanzan los resultados perseguidos por la estrategia.

Componentes básicos de un buen CMI:

- **Una cadena de relaciones de causa efecto:** Expresa el conjunto de hipótesis de la estrategia a través de objetivos estratégicos y su logro mediante indicadores de desempeño.
- **Un enlace a los resultados financieros:** Los objetivos del negocio y sus respectivos indicadores, deben reflejar la composición sistémica de la estrategia, a través de cuatro perspectivas: Financiera, Clientes, Procesos Internos, Aprendizaje y Crecimiento. Los resultados deben traducirse finalmente en logros financieros que conlleven a la maximización del valor creado por el negocio para sus accionistas.
- **Un balance de indicadores de resultados (efecto) e indicadores guía (causa):** Además de los indicadores que reflejan el desempeño final del negocio, se requiere un conjunto de indicadores que reflejen las cosas que se necesitan "hacer bien" para cumplir con el objetivo (asociados a las palancas de valor e indicadores guía- causa). Estos miden el progreso de las acciones que nos acercan o que propician el logro del objetivo. El propósito es canalizar acciones y esfuerzos orientados hacia la estrategia del negocio.
- **Mediciones que generen e impulsen el cambio:** Una de las premisas a las que hacen mención Kaplan y Norton es: La medición motiva determinados comportamientos, asociados tanto al logro

como a la comunicación de los resultados organizacionales, de equipo e individuales. De allí que un componente fundamental es el de definir indicadores que generen los comportamientos esperados, particularmente aquellos que orienten a la organización a la adaptabilidad ante un entorno en permanente y acelerado cambio.

- **Alineación de iniciativas o proyectos con la estrategia a través de los objetivos estratégicos:** Cada proyecto que exista en la empresa debe relacionarse directamente con el apalancamiento de los logros esperados para los diversos objetivos expresados a través de sus indicadores.
- **Consenso del equipo directivo de la empresa u organización:** El Cuadro de Mando Integral permite el resultado del diálogo entre los miembros del equipo directivo, para lograr reflejar la estrategia del negocio, y de un acuerdo sobre cómo medir y respaldar lo que es importante para el logro de dicha estrategia.

Resumiendo, el CMI consiste en:

- Formular una estrategia consistente y transparente.
- Comunicar la estrategia a través de la organización.
- Coordinar los objetivos de las diversas unidades organizativas.
- Conectar los objetivos con la planificación financiera y presupuestaria.
- Identificar y coordinar las iniciativas estratégicas.
- Medir de un modo sistemático la realización, proponiendo acciones correctivas oportunas.

Ricardo Martínez Rivadeneira (profesor consultor de Gestión) plantea: “El Cuadro de Mando Integral es una forma integrada, balanceada y estratégica de medir el progreso actual y suministrar la dirección futura de la compañía que le permitirá convertir la visión en acción, por medio de un conjunto coherente de indicadores agrupados en cuatro diferentes perspectivas, a través de las cuales es posible ver el negocio en conjunto”. [1]

1.2.1 Surgimiento del Cuadro de Mando Integral

Los orígenes del Cuadro de Mando Integral (CMI) o Balanced Scorecard (BSC), data de 1990, cuando el Instituto Nolan Norton, patrocinó un estudio de un año de duración sobre múltiples empresas. El estudio

fue motivado por la creencia de que los enfoques existentes sobre la medición de la actuación se estaban volviendo obsoletos.

En los inicios del proyecto se examinaron estudios recientes sobre casos de sistemas innovadores de medición de la actuación. Una de las empresas-fabricantes describió un enfoque para medir la tasa de progreso de actividades de mejora continua. El caso mostró la forma en que esta entidad estaba utilizando un “Cuadro de mando corporativo”, de nueva creación, que además de varios indicadores financieros tradicionales, contenía mediciones de actuación relacionadas con los plazos de entrega a los clientes, la calidad y los tiempos de los ciclos de los procesos de fabricación, y la eficacia de los avances de los nuevos productos.

Durante la primera mitad del estudio se presentó una gran variedad de ideas, incluyendo el valor del accionista, mediciones de productividad y calidad, y nuevos planes de compensación, centrándose los participantes en un Cuadro de Mando multidimensional, ya que, parecía ser lo más prometedor para sus necesidades.

Las discusiones del grupo condujeron a una expansión del Cuadro de Mando hasta llegar a lo que se denominó como un “Cuadro de Mando Integral”. Los líderes empresariales lo vieron como algo más que un sistema de mediciones, querían utilizar el nuevo sistema para comunicar y alinear sus organizaciones con las nuevas estrategias; lejos del enfoque histórico y a corto plazo de reducción de costos y competencia a bajo precio, y hacia la generación de crecientes oportunidades, ofreciendo a los clientes productos y servicios con valor añadido y a medida.

Norton y Kaplan describieron la importancia de elegir indicadores basados en el éxito estratégico, publicado en septiembre- octubre de 1993 en el artículo “Cómo poner a trabajar al Cuadro de Mando Integral”, donde veían al CMI como un vehículo para ayudar a la empresa a traducir y poner en práctica la estrategia. Por tanto, el Cuadro de Mando Integral había evolucionado de un sistema de indicadores mejorado, para convertirse en un sistema de gestión central.

1.2.2 Perspectivas del Cuadro de Mando Integral

El Cuadro de Mando Integral usa un modelo de negocio basado y estructurado en indicadores y objetivos que giran en torno a cuatro perspectivas: financiera, clientes, procesos internos, aprendizaje y crecimiento. Estas perspectivas son las más comunes, porque son aplicables en un gran número de empresas. (Ver figura 1.1).

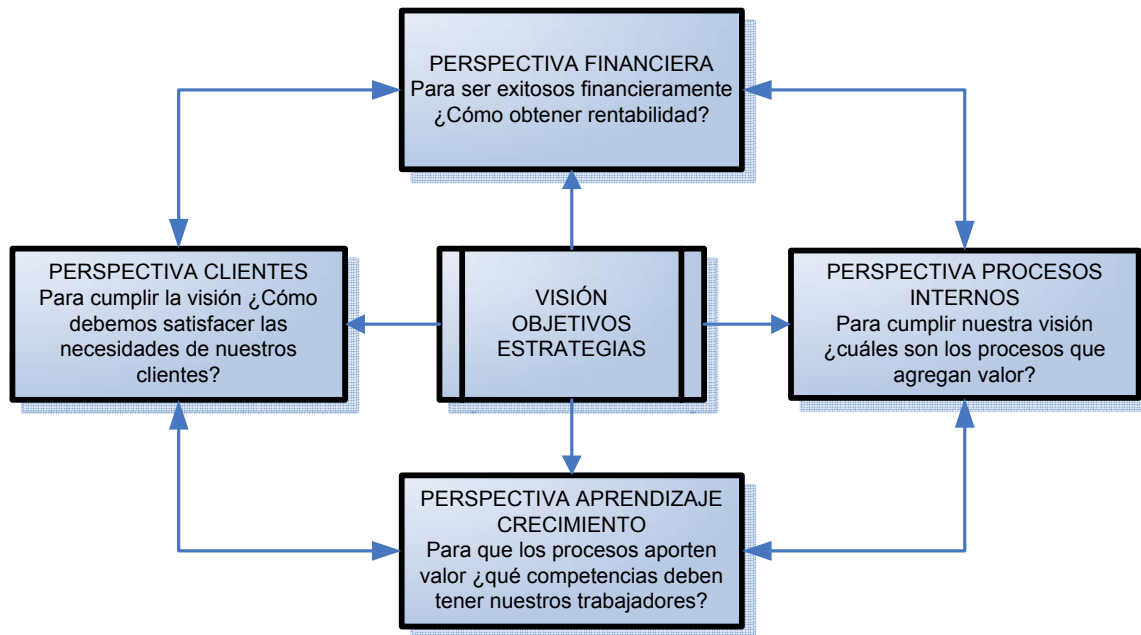


Figura 1.1: Las perspectivas del Cuadro de Mando Integral

1.2.2.1 Perspectiva Financiera

El Cuadro de Mando Integral puede hacer que los objetivos financieros sean explícitos, y que se adapten a la medida de las unidades de negocio, en las diferentes fases de su crecimiento y ciclo de vida. El cuadro de mando utiliza los objetivos financieros en relación al aumento e intensificación de la rentabilidad, los rendimientos de los activos y los ingresos.

Este permite que la alta dirección de las unidades de negocio especifique, no solo como se evaluará el éxito de la empresa a largo plazo, sino las variables que se consideran más importantes para la creación y el impulso de los objetivos de resultados a largo plazo. Además, las empresas pueden elegir estrategias financieras entre los temas relacionados con el crecimiento de los ingresos, la reducción de costos y la utilización de los activos.

Fases del ciclo de vida de la perspectiva financiera.

Se ha descubierto que, para cada uno de los tres objetivos de crecimiento, sostenimiento y recolección, existen tres estrategias financieras:

- Crecimiento y diversificación de los ingresos.

- Reducción de costes / mejora de la productividad.
- Utilización de los activos / estrategia de inversión.

El crecimiento y la diversificación de los ingresos implican la expansión de la oferta de producción y servicios para que se conviertan en una oferta de mayor valor añadido, y cambien los precios de dichos productos y servicios. El objetivo de reducción de los costos y mejora de la productividad supone esfuerzos para rebajar los costos directos de la producción y los servicios, reducir los precios indirectos y compartir los recursos comunes con otras unidades de negocios. Los directivos intentan reducir los niveles de capital circulante que se necesitan para apoyar a un volumen o una diversidad del negocio dada.

1.2.2.2 Perspectiva Cliente

Para lograr el desempeño financiero que una empresa desea, es fundamental que posea clientes leales y satisfechos, con ese objetivo en esta perspectiva se miden las relaciones con los clientes y las expectativas que los mismos tienen sobre los negocios. Además, en esta perspectiva se toman en cuenta los principales elementos que generan valor para los clientes, para poder así centrarse en los procesos que para ellos son más importantes y que más los satisfacen.

El conocimiento de los clientes y de los procesos que más valor generan es muy importante para lograr que el panorama financiero sea próspero. Sin el estudio de las peculiaridades del mercado al que está enfocada la empresa no podrá existir un desarrollo sostenible en la perspectiva financiera, ya que en gran medida el éxito financiero proviene del aumento de las ventas, situación que es el efecto de clientes que repiten sus compras porque prefieren los productos que la empresa desarrolla teniendo en cuenta sus preferencias.

Los directivos también han de identificar que clientes en los segmentos seleccionados, valoran y eligen la propuesta de valor que ellos entregarán a estos clientes. Pueden seleccionar objetivos y medidas entre las tres clases de atributos que, si se satisfacen, permitan que la empresa retenga y amplíe sus negocios con estos clientes seleccionados. Las tres clases de atributos son:

- Atributos de producto y servicio: funcionalidad, calidad y precio.
- Relaciones con los clientes: calidad de la experiencia de compra y relaciones personales.
- Imagen y reputación.

Al seleccionar objetivos e indicadores concretos pertenecientes a estas tres categorías, los directivos pueden encarrilar a su organización hacia la entrega a sus segmentos de clientes seleccionados de una propuesta de valor superior.

Existen cinco tipos de estrategias que son los más eficaces, según Kaplan y Norton, para la medición de los objetivos genéricos de esta perspectiva.

- 1 La cuota de mercado
- 2 El incremento de clientes
- 3 La satisfacción del cliente
- 4 La rentabilidad del cliente
- 5 El indicador de las propiedades valor del cliente

Estos indicadores por ser tan generales tendrán que ser adaptados a cada entidad en cuestión con respecto a las expectativas que ellas tengan hacia sus segmentos de mercado.

1.2.2.3 Perspectiva Procesos Internos

En la perspectiva del proceso interno, los directivos identifican los procesos críticos en los que deben sobresalir con excelencia si es que quieren satisfacer los objetivos de los segmentos de accionistas y de clientes seleccionados.

Un avance reciente ha sido la incorporación del proceso de innovación como un componente vital de la perspectiva del proceso interno. El proceso de innovación subraya la importancia de la identificación de las características de los segmentos de mercado que la organización desea satisfacer con sus productos y servicios futuros, y de diseñar y desarrollar los productos y los servicios que en el futuro satisfagan a esos segmentos seleccionados. Este enfoque permite que la organización ponga un énfasis considerable en los procesos de investigación, diseño y desarrollo que dan como resultado nuevos productos, servicios y mercados.

El proceso operativo sigue siendo importante, y las organizaciones deben identificar las características de costo, calidad, tiempo y actuación que permitirán entregar productos y servicios superiores a sus clientes existentes. El proceso de servicio posventa permite a la empresa presentar aspectos importantes del servicio que se ofrecen después de que el producto o servicio adquirido ha sido entregado al cliente.

Además esta perspectiva identifica aquellos pocos procesos que se espera que tengan el mayor impacto sobre la estrategia. Por ejemplo una empresa puede aumentar sus inversiones en Investigación y Desarrollo (I+D) y re estructurar sus procesos de desarrollo de producto de manera que pueda obtener productos innovadores y de alto rendimiento para sus clientes.

Cada negocio tiene un conjunto único de procesos para crear valor para los clientes y producir resultados financieros. Sin embargo, se ha descubierto que un modelo genérico de cadena de valor proporciona una plantilla que las empresas pueden hacer a su medida, al preparar su perspectiva del proceso interno. (Ver figura 1.2).



Figura 1.2: La cadena de valor de la perspectiva del proceso interno

1.2.2.4 Perspectiva Aprendizaje y Crecimiento

La capacidad de alcanzar las ambiciosas metas de los objetivos financieros, del cliente y de los procesos internos depende de la capacidad de crecimiento y aprendizaje de la organización. Esta perspectiva es el motor impulsor de las anteriores perspectivas del cuadro de mando y refleja los conocimientos y habilidades que la empresa posee tanto para desarrollar sus productos como para cambiar y aprender. Los inductores de crecimiento y aprendizaje provienen primordialmente de tres fuentes: *los empleados, los sistemas y la equiparación de la organización*. Las estrategias para una actuación superior exigirán, en general, unas inversiones importantes en personal, sistemas y procesos que construyen capacidades para la organización.

Los objetivos financieros, de clientes y de procesos internos del CMI revelarán grandes vacíos entre las capacidades existentes de las personas, los sistemas y procedimientos; al mismo tiempo mostrarán que será necesario para alcanzar una actuación que represente un gran adelanto. Para llenar estos vacíos, las

empresas tendrán que invertir en la recalificación de empleados, potenciar los sistemas y tecnología y coordinar los procedimientos y rutinas de la organización. La perspectiva de formación (o aprendizaje) y crecimiento permite que la empresa se asegure su capacidad de renovación a largo plazo, un requisito previo para una existencia duradera.

La cuarta y última perspectiva del Cuadro de Mando Integral desarrolla objetivos e indicadores para impulsar el aprendizaje y crecimiento de la organización. Los objetivos establecidos en las perspectivas financiera, del cliente y de los procesos internos identifican los puntos en que la entidad ha de ser excelente. Los objetivos de la perspectiva de aprendizaje y crecimiento proporciona una infraestructura que permite que se alcancen objetivos ambiciosos en las restantes tres perspectivas.

1.3 Sistemas automatizados existentes

En el ámbito internacional existen varios tipos de software que ayudan a informar y evaluar la implantación del CMI en una organización, no así en lo nacional. Debido a esta problemática, se realizó un estudio previo a distintos software.

DECIDE SOFT: Primer software en español para la completa realización de Cuadros de Mando Integral.

Decide Soft es un producto creado, diseñado y comercializado por dos empresas compostelanas:

Dimensiona Consultoría y Tecnológica: empresa dedicada al desarrollo de aplicaciones informáticas punteras (sistemas de traducción automática y sistemas de publicación Web).

Ferreiro/Boullón & Asociados: empresa declarada IEBTS (de base tecnológica) dedicada a la provisión de soluciones para la toma de decisiones empresariales, especialmente en el área sanitaria.

Delphos: Es un software de Control de Gestión que permite implementar íntegramente un Cuadro de Mando Integral (Balanced ScoreCard), un Plan Estratégico, un Plan Anual Operativo o cualquier otro modelo que se requiera para controlar las operaciones de cualquier tipo de organización, mejorar el desempeño y la productividad.

Fue implantado en Costa Rica en la década de los 90, e incluye algunos países de: Centroamérica, Suramérica y Norteamérica. Su principal beneficio es permitir desarrollar una administración orientada a resultados, mejorando y controlando completamente el ámbito de acción de cada uno de los miembros de la organización, sea pública o privada.

Delphos integra en una sola aplicación:

- Administración estratégica (objetivos, indicadores y responsables).
- Administración de Proyectos.
- Visualización de información mediante cubos o modelos multidimensionales.
- Graficador, Reporteador, Presupuesto Gerencial.

Los cuales tienen conectividad universal, por lo que pueden ser alimentados automáticamente desde cualquier base de datos. Delphos está diseñado para profesionales NO informáticos, esto garantiza que con una pequeña capacitación de su uso y aplicación, el cliente obtendrá el máximo provecho de su inversión, contando con soporte certificado las 24 horas al día los 365 días del año.

Dialogy Strategy: Es el software de Cuadro de Mando Integral o Balanced Scorecard ideal para aquellas empresas que quieren crecer, que han entendido que a una administración detallada deben agregarle herramientas que les permitan medir su estrategia y desempeño de manera sencilla.

Este software permite definir una serie de indicadores, objetivos estratégicos, enlazarlos en una relación de causa-efecto, asignar las medidas apropiadas, introducir datos y comenzar a monitorizar el rendimiento de su compañía.

Dialog Strategy es un software gratuito para el desarrollo de Cuadro de Mando Integral. Sus versiones están disponibles en MS SQL Server y Access 2000. Posee la posibilidad de ser entregado en inglés, español o noruego, según el deseo de los clientes. Es multiusuario, multizona, multidivisión y multiproducto. Interfases intuitiva y amigable.

Requiere de estos mínimos privilegios:

- Windows 98, ME, 2000, NT or XP.
- Min. 64MB RAM.
- 25MB de espacio de disco libre.

Algunos de los ejemplos del uso de Cuadro de Mando Integral en **Cuba** lo podemos encontrar en:

Caso GET Varadero: el diseño e implementación del Cuadro de Mando Integral en el GET Varadero, unido a la aplicación de otras herramientas complementarias, le permitió al equipo de dirección observar globalmente la marcha de los aspectos clave de la gestión de la empresa, lo que ayudó a corregir oportunamente las desviaciones con tendencia a alterar las metas u objetivos previstos, mediante un

control por excepción a ciertas áreas clave seleccionadas cuidadosamente, las que por medio de señales rápidas de alerta permiten identificar las desviaciones, con sus causas y responsables, ligadas al nivel inmediato inferior de su propia función directiva. La información debe obtenerse con rapidez para poder actuar oportunamente.

La empresa de Servicios de Protección de Cienfuegos, SEPSA: En su primer año, el CMI se ha utilizado como una herramienta de control, comunicación y retroalimentación. Ha creado una auténtica intranet corporativa utilizando técnicas del Cuadro de Mando Integral, así como condiciones para efectuar el control estratégico de la misma. En esta empresa predomina el análisis holístico gracias a esta herramienta. La comunicación dentro de la organización ha mejorado ostensiblemente, al igual que la toma de decisiones. Mediante su control operativo y corto efectivo el CMI ha ayudado indiscutiblemente a impulsar la gestión del aprendizaje en SEPSA Cienfuegos.

CMI en ETECSA: La empresa telefónica ha usado el CMI en la unidad de negocio internacional (UNI). Tiene implementada una aplicación que automatiza el control de las llamadas internacionales en cuanto a la facturación en minutos al mes o la duración de las llamadas en total, el importe percibido, los gastos incurridos y la cantidad de llamadas.

1.4 Necesidades del Cuadro de Mando Integral en Cuba

El Cuadro de Mando Integral se está expandiendo a las empresas cubanas, las cuales se han ido adaptando a los cambios ocurridos en el contexto internacional. Muchas de estas organizaciones se han apropiado de las normas de calidad ISO 9000 y la están aplicando con resultados satisfactorios.

En Cuba se ha reflejado un despertar en este sentido. El mismo ha sido declarado en la Resolución Económica del V Congreso del Partido Comunista de Cuba, el cual plantea lo siguiente:

"El empleo de técnicas modernas de dirección empresarial, adecuadas a nuestra características y basadas en las mejores y más avanzadas prácticas contemporáneas..."[2]

Todo este proceso demuestra una brecha abierta hacia el reconocimiento de la necesidad de un cambio en nuestro sistema empresarial hacia una búsqueda de mejores desempeño.

El modelo de Perfeccionamiento Empresarial ha contribuido al desarrollo de las organizaciones. El mismo se encamina a potenciar de manera continua los niveles de eficiencia, autoridad y ejecutividad de la empresa estatal cubana como eslabón fundamental de la economía.

Este proceso hace que el Cuadro de Mando Integral sea un camino importante en el logro de los objetivos organizacionales porque presupone un cambio en la concepción de la forma que puede llevar a nuestras organizaciones a un nuevo estado.

El objetivo de las organizaciones tanto lucrativas como sin fines de lucro es obtener beneficio, estableciendo un ambiente en el que las personas puedan sentirse útiles a la comunidad y contribuir con sus capacidades a la obtención de bienes y servicios que conducen al bienestar social.

Una organización, tanto con fines de lucro como no lucrativos, es un sistema coordinado conscientemente que tiene el fin de conseguir cada vez más beneficios sobre la base de aportar bienes y servicios a la sociedad con el fin de satisfacer las necesidades, deseos de un conjunto de personas que integran el mercado objetivo.

Cada vez es más creciente las entidades de la isla que muestran interés por la herramienta de gestión empresarial, dada a conocer por Kaplan y Norton en 1992.

1.5 Diseño metodológico de la investigación

Se utilizaron los siguientes métodos teóricos para el estudio de las características del objeto de investigación que no son observables directamente, ya que, facilitan la construcción de modelos de investigación y crean las condiciones para ir más allá de las particularidades fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis.

Analítico – sintético.

El análisis de fuentes nos ayuda en la búsqueda de la esencia, rasgos que caracterizan y distinguen al Cuadro de Mando Integral, y establece las tendencias e insuficiencias en el campo de acción y el objeto de estudio.

Modelación.

El diseño del sistema que se propone, utiliza símbolos para designar las propiedades del sistema real de un CMI y tiene la capacidad de representar las características y relaciones fundamentales de dicho sistema.

1.6 Tendencias y Tecnologías actuales

Se hizo un análisis de las tendencias y tecnologías actuales posibles a emplear, considerando las situaciones del trabajo.

1.6.1 Las aplicaciones Web.

El desarrollo de aplicaciones Web está siendo utilizado en muchas organizaciones, esta situación va ir creciendo indefinidamente. Es por ello que día a día se requieren más programadores capacitados para desarrollos basados en el World Wide Web (WWW).

Una aplicación web es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador Web como cliente ligero. La habilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Aplicaciones como los webmails, wikis, weblogs, MMORPGs y tiendas en línea son ejemplos bien conocidos de aplicaciones web.

Las aplicaciones Web son una especialización de las aplicaciones cliente-servidor, o sea, su arquitectura general es la de un sistema cliente/servidor, donde tanto el cliente (el navegador) como el servidor (el servidor Web), y el protocolo mediante el que se comunican (el HTTP: HyperText Transfer Protocol) son estándar, y no han de ser creados por el desarrollador.

La parte del cliente de las aplicaciones Web está formada por el código HTML (HyperText Markup Language) que forma la página Web, con opción a código ejecutable mediante los lenguajes script de los navegadores (JavaScript, VBScript, PerlScript) o mediante pequeños programas (applets) en Java. La parte del servidor está formada por un programa o script que es ejecutado por el servidor Web, y cuya salida se envía al navegador del cliente.

Las aplicaciones Web ofrecen grandes ventajas que pueden ser aprovechadas por muchas organizaciones. Entre ellas se pueden mencionar:

- No requieren instalación, pues usan tecnología Web, lo cual nos permite el aprovechamiento de todas las características del Internet.
- Son fáciles de usar (no requieren conocimientos avanzados de computación).
- Alta disponibilidad, ya que puede realizar consultas en cualquier parte del mundo donde tenga acceso a Internet y a cualquier hora.

No obstante a la serie de ventajas que presenta tiene además algunas desventajas, las cuales son:

- Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- La interactividad no se produce en tiempo real, en las aplicaciones web cada acción del usuario conlleva un tiempo de espera excesivo hasta que se obtiene la reacción del sistema.
- Elementos de interacción muy limitados. En comparación con el software de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones web (mediante formularios principalmente) son muy escasas.
- Diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.

1.6.2 Lenguajes de programación para la Web.

Uno de los ejes fundamentales que diferencian a Internet de otros medios de comunicación es la interacción y personalización de la información con el usuario. Esto se logra por medio de lenguajes para programación Web que existen hoy en día. Los cuales se encuentran tanto del lado del servidor como del lado del cliente

Entre los lenguajes del lado del servidor podemos encontrar a PERL, ASP, PHP como los más sobresalientes por el auge que han tenido. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la Información, entre otras funciones.

Del lado del cliente se encuentran principalmente el JavaScript (JScript) y el Visual Basic Script (VBScript), que son los encargados de aportar dinamismo a la aplicación en los navegadores. En el caso del VBScript este es prácticamente usado a la hora de programar en ASP del lado del servidor, ya que su mayor desventaja radica en que solo es soportado por el Navegador Web de su fabricante, Microsoft.

1.6.2.1 PHP 5.1

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. Además es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Ventajas de PHP:

- Es un lenguaje multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server).
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destacando su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la Web.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Los diseñadores de PHP5 han realizado un cambio radical en el tratamiento de las variables objeto: en PHP5 todas las variables que nombran objetos son en realidad referencias. No hay que usar el operador '&' ni en las asignaciones, ni en el paso de parámetros que son objetos, ahorrándose con ello gran cantidad de potenciales errores. El mismo introduce tres palabras clave (public, private y protected) que sustituyen a var en la definición de variables miembro --atributos-- de la clase, y que preceden a la definición de funciones miembro --métodos--. Para el uso de este lenguaje se cuenta con un framework, llamado Code Igniter, que ya nos brinda todas las funcionalidades necesarias para crear la aplicación Web esperada y la posibilidad de construir en base al patrón de arquitectura MVC (Modelo Vista Controlador), lo más usual en este tipo de aplicación.

Otros lenguajes como Perl (Practical Extraction and Report Language), ASP (Active Server Pages) y JSP (Java Server Pages) tienen características similares al PHP aunque poseen rasgos que los marcan y por ello los distingue, entre ellos podemos encontrar:

- *Características multiplataformas:* Menos el ASP, que es solamente soportado por la plataforma Windows, los demás lenguajes están soportados en múltiples plataformas.
- *Velocidad de ejecución:* La velocidad es mayor en PHP, seguidos por PERL y JSP.

- *Disponibilidad de recursos:* Actualmente los más utilizados en la Internet son el PHP y el JSP, siendo más utilizado en la publicación de artículos y códigos de ejemplos. PHP tiene una de las comunidades más grandes en Internet, al igual que la de Java.
- *Familiaridad con el lenguaje:* En la universidad los lenguajes más utilizados por los programadores es el ASP y el PHP.

De acuerdo a las anteriores comparaciones, más las posibilidades multiplataforma, rapidez y su interactividad con el sistema gestor de base de datos MySQL, el PHP resultó ser un lenguaje etiquetado para el desarrollo de la aplicación, particularmente en la versión PHP 5.1.

1.6.2.2 JavaScript

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. No es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

La mayoría de los navegadores interpretan el código JavaScript integrado dentro de las páginas Web, ya que es muy sencillo.

1.6.3 Sistemas de Gestión de Base de Datos.

Entre los SGBD comúnmente utilizados en el mundo tenemos Oracle, Microsoft SQL Server, PostgreSQL, MySQL, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

Los sistemas mencionados anteriormente facilitan el trabajo con la base de datos y tienen características que los diferencian, por ejemplo:

- Oracle: Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias). Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios.[3] Como se mencionó anteriormente, su gran desventaja consiste en adquirir licencia para utilizarlo, mientras que MySQL es gratis.

- Microsoft SQL Server: Puede ser utilizado solamente con el sistema operativo Windows que está patrocinado por la compañía Microsoft, por lo que no es multiplataforma y posee licencias con costos muy elevados. Todo lo contrario de MySQL, que es multiplataforma y gratuito.
- PostgreSQL: Es un sistema de gestión de bases de datos Objeto-Relacionales (ORDBMS) libre. PostgreSQL tiene transacciones, integridad referencial, vistas, y multitud de funcionalidades, pero es lento y pesado. Por otro lado, la velocidad de respuesta que ofrece este gestor parece un poco deficiente. También resulta engorroso las pequeñas variaciones que presenta en algunos de los tipos de datos que maneja, siendo el problema más comentado el referente al tipo "serial". Sin embargo, MySQL en la versión 5.0 soporta transacciones e integridad referencial, con la diferencia de que este último es mucho más rápido. Por tanto, es este gestor de base de datos una buena opción.

1.6.3.1 MySQL 5.0

MySQL es la base de datos de código abierto (Open Source) más popular de Internet. Es un sistema de gestión de base de datos, multihilo, multiplataforma y multiusuario con más de seis millones de instalaciones. Este es fiable, fácil de usar y muy robusto.

MySQL está desarrollado como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso.

MySQL es muy utilizado en aplicaciones Web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

Las principales características de este gestor de bases de datos son las siguientes:[4]

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones, es por esto que como SGBD se seleccionó MYSQL.

1.6.4 Servidor Web Apache 2.2

Es un software de código abierto que ofrece una perfecta combinación entre estabilidad y sencillez, por ser un servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Por supuesto, se distribuye prácticamente con todas las implementaciones de Linux.

Tiene capacidad para servir páginas tanto de contenido estático, para lo que nos serviría sencillamente un viejo ordenador 486, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información.

Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios Web elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.

Las principales características de Apache son:[5]

- Funcionalidad en múltiples plataformas.
- Soporte del último protocolo http.
- Sencilla administración basada en la configuración de un único archivo.
- Soporte para CGI (Common Gateway Interface) y FastCGI.

Beneficios

Apache puede soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos. El servidor puede personalizarse mejor para las necesidades de cada sitio Web. Por ejemplo, los sitios Web que necesitan más que nada escalabilidad pueden usar un MPM hebrado como worker, mientras que los sitios Web que requieran por encima de otras cosas estabilidad o compatibilidad con software antiguo pueden usar prefork. Además, se pueden configurar funcionalidades especiales como servir diferentes hosts con diferentes identificadores de usuario (perchild).

1.6.5 Patrones de arquitectura

Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Los patrones de arquitectura representan el nivel más alto en el sistema de patrones propuesto en *Pattern Oriented Software*. [6]

1.6.5.1 Modelo Vista Controlador

Un propósito común en numerosos sistemas es el de tomar datos de un almacenamiento y mostrarlos al usuario. Luego que el usuario introduce modificaciones, las mismas se reflejan en el almacenamiento. Dado que el flujo de información ocurre entre el almacenamiento y la interfaz, una tentación común, un impulso espontáneo (hoy se llamaría un anti-patrón) es unir ambas piezas para reducir la cantidad de código y optimizar el rendimiento.



Figura 1.3: Modelo - Vista - Controlador

El patrón conocido como Modelo-Vista-Controlador (MVC), figura 1.3, separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

Modelo: Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: Maneja la visualización de la información.

Controlador: Controla el flujo entre la vista y el modelo (los datos).

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual.

Entre las ventajas del estilo Modelo-Vista-Controlador están las siguientes:

- Soporte de múltiples vistas: Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación Web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes.
- Adaptación al cambio: Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares o PDAs. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Una desventaja que tiene este modelo es el costo de actualizaciones frecuentes: Si el modelo experimenta cambios frecuentes, por ejemplo, podría desbordar las vistas con una lluvia de requerimientos de actualización.

1.6.6 Proceso de Desarrollo.

La producción de software busca adecuarse cada día más a las necesidades del usuario, esto trae como consecuencia que la misma aumente en tamaño y complejidad. Por estas razones, se hizo necesario definir la metodología de ingeniería del software que guiara el proceso de automatización de nuestro trabajo, la cuál fue seleccionada el Proceso Unificado de Desarrollo de Software (RUP), junto con el Lenguaje Unificado de Modelado (UML).

1.6.6.1 RUP

La primera versión de RUP, fue puesta en el mercado en 1998 después de haberse realizado varias prácticas.

El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

Principales características de RUP durante su ciclo de vida:

- Iterativo e incremental
- Centrado en la arquitectura
- Guiado por los casos de uso.

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al concluir cada ciclo; estos se dividen en 4 fases, las cuales establecen oportunidad y alcance, identifican las entidades externas o actores con las que se trata, e identifica también los casos de uso. Así como contiene 9 flujos de trabajos: los 6 primeros son conocidos como flujos de Ingeniería y los restantes 3 como de apoyo. (Ver figura 1.4).

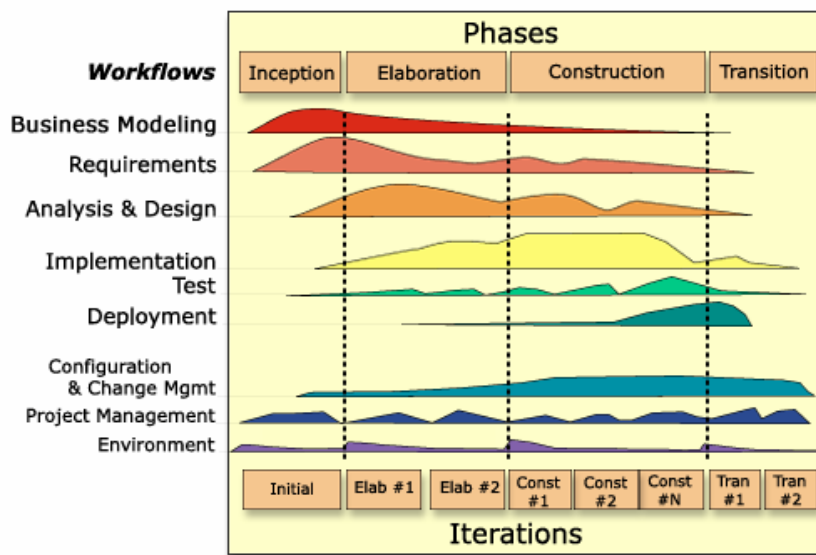


Figura 1.4 Proceso Racional Unificado (RUP)

1.6.6.2 UML

UML (Unified Modeling Language) es un lenguaje que permite modelar, especificar, construir y documentar los elementos que forman un sistema software orientado a objetos; no es un método o un proceso. Fue impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

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.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

- Diagramas de estructura estática:
 - Diagrama de clases
 - Diagrama de objetos
 - Diagrama de casos de uso
- Diagramas de comportamiento:
 - Diagramas de interacción (secuencia y colaboración)
 - Diagrama de estados
 - Diagrama de actividad
- Diagramas de implementación:
 - Diagrama de componentes
 - Diagrama de despliegue

De forma general las principales características son: [7]

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

1.6.7 Herramienta utilizadas

1.6.7.1 Rational Rose

Se decidió que se utilizaría el Rational Rose Enterprise Edition 2003, para sustentar la documentación, como modelador visual de la notación UML (Unified Modeling Language) para la confección de los

diagramas que se ilustran en este documento. Esta herramienta es muy completa y ofrece amplias potencialidades.

Rational Rose es una herramienta CASE que da soporte al modelado visual mediante UML ofreciendo distintas perspectivas del sistema. Da soporte al Proceso Unificado de Rational (RUP). Permite especificar, analizar, y diseñar el sistema antes de codificarlo.

Características del Rational Rose:

- Mantiene la consistencia de los modelos del sistema software.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Chequeo de la sintaxis UML.
- Generación de documentación automáticamente.
- Generación de código a partir de los modelos.
- Capacidades de ingeniería inversa (crear modelo a partir código).
- Disponibilidad en múltiples plataformas.

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML.

Esta herramienta de software propone la utilización de cuatro tipos de vistas para realizar un diseño del sistema: vista de Caso de Uso, vista Lógica, vista de Componentes y la vista de Despliegue; permitiendo crear y refinar de esta forma un modelo completo que represente el dominio del problema y el sistema de software.

1.6.7.2 Diseño de interfaz: Dreamweaver 8.0

El diseño de interfaz es uno de los elementos "clave" en la realización del programa. Podemos definir el interfaz como: "el conjunto de trabajos y pasos que seguirá el usuario, durante todo el tiempo que se relacione con el programa, detallando lo que verá y escuchará en cada momento, y las acciones que realizará, así como las respuestas que el sistema le dará". [8]

Dreamweaver es la herramienta de diseño de páginas Web más avanzada. Es un programa que se puede actualizar con componentes, que fabrica tanto Macromedia como otras compañías, para realizar otras

acciones más avanzadas. Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías muy fáciles de usar:

- Hojas de estilo y capas.
- JavaScript para crear efectos e interactividades.
- Inserción de archivos multimedia.

Su última versión, Dreamweaver 8.0, es la herramienta de desarrollo Web líder del mercado y permite a sus usuarios diseñar, desarrollar y mantener de forma eficaz sitios y aplicaciones Web basadas en normas. Con esta los desarrolladores Web lo abarcan todo, desde la creación y mantenimiento de sitios Web básicos hasta aplicaciones avanzadas compatibles con las mejores prácticas y las tecnologías más recientes.

Además incluye herramientas para trabajar aplicaciones que manejan XML, así como mejoras a su manejo de hojas de estilo (CSS). Para el trabajo con CSS han simplificado la creación y manejo de diferentes estilos, promoviendo los estándares para nuevos usuarios y facilitando su aplicación para usuarios avanzados. Manifiesta el potencial que representa para la comunidad Web el desarrollo abierto basado en estándares y mejora, al mismo tiempo, las herramientas básicas que necesitan los profesionales. Con esta, los desarrolladores Web lo abarcan todo, desde la creación y mantenimiento de sitios Web básicos hasta aplicaciones avanzadas compatibles con todas las principales tecnologías de servidor como, por ejemplo, ColdFusion, PHP, ASP, ASP.NET y JSP, para que los desarrolladores, más expertos o menos expertos, puedan dar vida a sus diseños. También han buscado facilitar la difusión de Flash Video, con herramientas que permiten incluir este formato muy fácilmente en páginas Web.

1.6.7.3 Zend Studio 5.1

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones Web, en lenguaje PHP. Además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. Es muy útil para el trabajo en equipo porque implementa opciones de trabajo conocido como CVS (sistema de controlador de versiones). La depuración hace la vida más fácil al programador, agregando que permite corregir los errores sintácticos en tiempo real.

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.

El mismo es uno de los ambientes de desarrollo integrado o Integrated Development Environment (IDE) que a través de un comprensivo conjunto de herramientas de edición, depurado, análisis, optimización y bases de datos acelera los ciclos de desarrollo y simplifica los proyectos complejos.

1.6.7.4 SQL Manager for MySQL 2005 versión 3.7.0.1

Características:

1. SQL Manager para MySQL 2005 es una herramienta de alto aprovechamiento para la administración y desarrollo de un servidor de Base de Datos MySQL
2. Soporta cualquier versión del MySQL.
3. Ofrece incluso las vistas, funciones y procedimientos almacenados, las llaves extranjeras con el motor de almacenamiento InnoDB.
4. El Diseñador de la Base de Datos Visual completamente rehecho.
5. Fácil de usar asistentes que realizan los servicios de MySQL.
6. Capacidad de importar y exportar datos.
7. Poderosa administración de seguridad.
8. Posibilidad de acceso al servidor MySQL a través del protocolo HTTP.
9. Múltiples conexiones a diferentes máquinas servidoras de base de datos MySQL

1.6.7.5 Framework Code Igniter 1.5

En el desarrollo de software, un framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de

programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.[9]

Lista de Framework en PHP

1. Symfony - Framework MVC.
2. Akelos Framework - MVC multilingüe desarrollado en España. (port de Rails a PHP)
3. BIF - framework hecho en PHP
4. CakePHP - framework MVC (inspirado en Ruby on Rails)
5. Code Igniter - framework.
6. Kumbia - Framework MVC (inspirado en Ruby on Rails)
7. Prado.
8. Zend FrameWork.
9. Zheta FrameWork - con lenguaje propio
10. ChiRoTuPHP - framework MVC (realizada por chichi, Tuco y Roberto Q. de la UAGRM)

“Code Igniter es una framework no abrumador, es un potente framework, construida para los programadores de PHP que necesitan una caja de herramientas simple y elegante para crear sus propios códigos. Es muy sencillo, y fácil de adaptar a cualquier sistema que deseemos realizar.”[10]

“Está pensado para ofrecer un alto rendimiento, ser ligero y fácilmente instalable (puede usarse en un alojamiento compartido y no es necesario tener acceso a la línea de comando). Además, gracias a la documentación y los foros (en inglés ambos), el aprendizaje es muy rápido y en un par de horas se puede empezar a trabajar con él. Por supuesto, será necesario más tiempo para conocer su funcionamiento completo, pero enseguida pueden empezar a hacerse cosas muy interesantes.

Viene con varias librerías para gestionar el acceso a datos, sesiones de usuarios, formularios, la seguridad, etc.... Además la comunidad de usuarios ha creado una serie de plugins, clases y librerías para extenderlo que lo hacen aún más interesante.” [11]

1.6.7.6 Framework AJAX: YUI

La librería YUI es libre. Está bajo la licencia BSD [12] que permite comercial o no la proliferación entre los usuarios. El código YUI es usado y refinado en millones de páginas vistas cada día. YUI se ha incrementado comprensiblemente, bien documentada y soportada por la comunidad en YUI forum.

1.7 Conclusiones

En el capítulo se expuso las condiciones y problemas referentes al objeto de estudio, a través de los diferentes conceptos y definiciones. Quedó definido de forma clara las herramientas, proceso de desarrollo y tecnologías a utilizar en todo el proceso para llevar a cabo la implementación de un Cuadro de Mando Integral.

Capítulo 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se realiza un estudio previo del negocio, el cual permite obtener una visión más clara del proceso en cuestión, por ello se muestra diversas reglas del negocio asociadas al campo de acción. Además se describen los actores y trabajadores del negocio, así como el diagrama de actividades y el modelo de objetos.

Se identifican los requisitos funcionales y no funcionales del sistema que dará solución al problema planteado; quiénes interactuarán con él (actores del sistema) y las distintas funcionalidades que ofrecerá a cada uno de los actores.

2.2 Objeto de estudio

El objeto de estudio trazado en este trabajo es la gestión de estrategias e indicadores de un Cuadro de Mando Integral (CMI), que de manera eficiente resuelva los problemas planteados anteriormente.

2.2.1 Objeto de automatización

Los procesos que son objeto de automatización de dicho sistema son los siguientes:

- Gestionar Estrategia: es un conjunto de acciones que se llevan a cabo para dar cumplimiento a un objetivo determinado. Permite la mejor distribución de los recursos y medios disponibles a efectos de poder obtener aquellas metas trazadas.
- Gestionar Indicadores: permiten evaluar, de manera sencilla y fiable, los aspectos de un sistema, así como constituyen elementos esenciales y forman parte de las actividades de seguimiento y evaluación, logrando una mejor toma de decisiones y efectividad de una empresa.

2.2.2 Información que se maneja

La información que se maneja es la referente al documento de la estrategia y la planilla de los indicadores para los procesos que tienen lugar en las empresas del país. En el documento de la estrategia se almacenan el nombre y la descripción de la misma. Mientras que la planilla de los indicadores contiene, su nombre, la perspectiva a la que pertenece, y la fórmula por la que debe ser calculado.

2.2.3 Propuesta del sistema

Con el objetivo de mejorar la administración y calidad de trabajo de las distintas empresas cubanas se decidió desarrollar un sistema que se centralizara dentro de este proceso, que es hoy en día muy significativo por la escasez de sistemas automatizados que existen en Cuba.

Para ello se llevó a cabo la agilización y desarrollo de las estrategias, así como los indicadores y sus métodos de cálculos que son muy importantes en el seguimiento de los objetivos y reportes generados para la toma de decisiones de la organización.

2.3 Modelo del negocio

Comprende la estructura, la dinámica, los problemas actuales de la organización e identificar las mejoras potenciales en la cual se va a implantar un sistema, constituye unos de los objetivos principales del modelamiento del negocio. Por tanto, teniendo en cuenta las deficiencias detectadas en el mismo y bajo un análisis profundo de las fuentes de problemas, se ha elaborado una propuesta de negocio que mantiene invariable el flujo de los procesos, que incurre en cambios de la ejecución de los mismos.

2.3.1 Proceso de Negocio

El proceso de solicitud de las estrategias tiene como objetivos definirlas según las necesidades del directivo. Cuando el directivo solicita la estrategia, el jefe de área la recibe y la define. Si no se encuentra la crea y entrega el documento de la estrategia al directivo. De estar creada solamente la entregaría.

En el caso de la solicitud del cálculo del indicador, el directivo solicita el valor del indicador, el jefe de área recibe la solicitud de la planilla y la identifica. De existir el indicador, este lo calcula y entrega el resultado al directivo. De lo contrario el jefe de área crea el indicador y lo calcula para luego entregar e informar los resultados obtenidos al directivo.

2.3.2 Reglas del negocio a considerar

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. En este trabajo se definió varias reglas, las mismas son las siguientes:

- El jefe de área es el encargado de definir, crear y entregar las estrategias según las necesidades requeridas.

- El jefe de área es la única persona autorizada para identificar, crear.
- El jefe de área es la única persona autorizada a calcular y entregar el resultado del indicador.
- El jefe de área debe crear el indicador, si no está identificado, antes de calcularlo.

2.3.3 Actores del negocio

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

A continuación se muestra en la tabla 2.1 los actores del negocio y su correspondiente justificación:

Actores del negocio	Justificación
Directivo	El directivo es el que inicia las acciones que tienen lugar en la solicitud de la estrategia e identificación de los indicadores, y al mismo tiempo es el que se beneficia totalmente con el resultado de dicho proceso.

Tabla 2.1: Descripción de los actores del negocio.

2.3.4 Diagrama de Casos de Uso del Negocio

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio. Este trabajo consta de su correspondiente Diagrama de Casos de Uso del Negocio. (Ver figura 2.1).

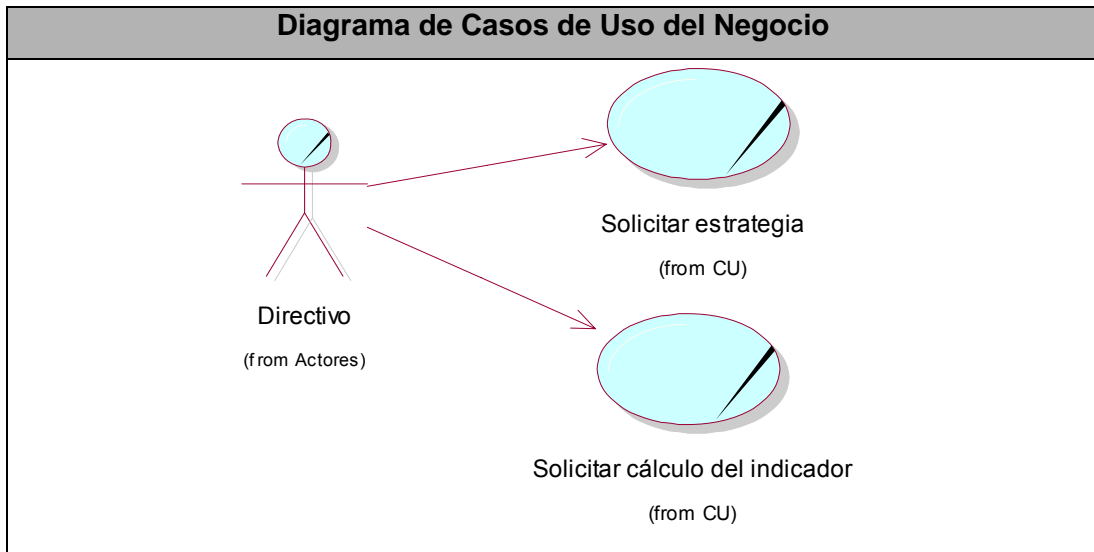


Figura 2.1 Diagrama de Casos de Uso del Negocio

2.3.5 Trabajadores del negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol.

A continuación se muestra en la tabla 2.2 los trabajadores del negocio y su correspondiente justificación:

Trabajadores del negocio	Justificación
Jefe de Área	Es el que realiza las distintas acciones que se desea analizar. Luego de esto define los documentos referidos a estrategias e indicadores. Toma las decisiones que cree necesaria y no se beneficia en ningún momento de las acciones realizadas en los procesos de negocio que tienen lugar, sino que se limita a ejecutar dichas acciones.

Tabla 2.2: Descripción de los trabajadores del negocio.

2.3.6 Descripción textual de los casos de uso del negocio. (VER ANEXO 1).

2.3.7 Diagrama de actividades. (VER ANEXO 2).

El diagrama de actividades es un grafo que contiene estados en que puede hallarse una actividad. Además, describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio. Así como se divide en varias calles, donde cada una de ella representa el actor y trabajador que está llevando a cabo en el proceso y muestra cómo se utilizan las entidades del negocio.

2.3.8 Diagrama de clases del Modelo de Objeto. (VER ANEXO 3).

El diagrama de clases, como artefacto que se construye para describir el modelo de objetos del negocio, muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos.

2.4 Especificación de los requisitos de software

Los requisitos del software son condiciones o capacidades que necesita un usuario para resolver un problema o lograr un objetivo, dichas capacidades o condiciones tienen que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.

2.4.1 Definición de los requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Estos no alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importarle con que propiedades o cualidades se relacionen.

Para cumplir con los objetivos propuestos se previó que el sistema tuviese las siguientes funcionalidades:

R.1 Gestionar Estrategia

R.1.1 Crear estrategia

R.1.2 Editar estrategia

R.1.3 Eliminar estrategia

R.2 Gestionar Indicadores

- R.2.1 Insertar indicadores
- R.2.2 Editar indicadores
- R.2.3 Eliminar indicadores

R.3 Gestionar Fórmula

- R.3.1 Crear fórmula
- R.3.2 Editar fórmula
- R.3.3 Eliminar fórmula

R.4 Gestionar Unidad de Medida

- R.4.1 Crear unidad de medida
- R.4.2 Editar unidad de medida
- R.4.3 Eliminar unidad de medida

R.5 Calcular Indicador

- R.5.1 Calcular por el método de BD
- R.5.2 Calcular por el Método manual

2.4.2 Definición de los requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta:

- **Apariencia o interfaz externa:** Un ambiente Web con banner que represente el tema del software. Colores serios y pocos. La mayor parte de la interfaz debe tener apariencia de aplicación de escritorio.
- **Usabilidad:** Debe permitir al usuario asimilar el uso del sistema de una manera rápida e intuitiva, sin necesidad de cursos ni consultas previas. Se añade al software una ayuda en línea que explica los principales conceptos y brinda una guía de instrucciones en los servicios automatizados del mismo.

Esta ayuda permite que usuarios con poca o ninguna experiencia en el uso de sistemas informáticos puedan utilizarlo.

- **Rendimiento:** El sistema debe dar posibilidades de usarse en computadores de mediano y bajo rendimiento, teniendo en cuenta las características del hardware que posee Cuba, las cuales permitan procesar la información y dar respuestas en plazos no mayores de 6 segundos.
- **Soporte:** Minimizar las necesidades de soporte. En caso de posible trabajo de mantenimiento, la Universidad de las Ciencias Informáticas dará este tipo de aseguramiento.
- **Software:** Como la aplicación tiene la arquitectura cliente/servidor es necesario que en la máquina servidor esté disponible:
 - El Apache Web Server 2.x
 - Un servidor de MySQL 5.0 o superior.
 - Un navegador para visualizar e interactuar con la aplicación, puede ser Internet Explorer desde la versión 5. Otro candidato es el Mozilla Firefox.
- **Portabilidad:** El sistema debe brindar posibilidades de uso en varias plataformas, como mínimas en Windows 98, 2000, 2003, XP y las distintas distribuciones de Linux.
- **Seguridad:** El control de la entrada de datos será regulada a través de usuarios autenticados con roles determinados. Otro tipo de usuario solo tendrá acceso a leer el contenido. Además el sistema será puesto en redes locales, solo teniendo acceso los usuarios de la red.
- **Políticos-culturales:** La aplicación servirá a los intereses de la Universidad de las Ciencias Informáticas y de Cuba.
- **Legales:** No poseer compromiso de pago por el uso de alguna reutilización de código empleado, no perteneciente a la Universidad de las Ciencias Informáticas.
- **Ayuda:** Implementar un manual de usuario competente.

2.5 Modelo del Sistema

2.5.1 Actores del sistema

Los actores del sistema pueden representar el rol que juega una o varias personas, equipo o un sistema automatizado. Cada trabajador del negocio que tiene actividades a automatizar es un candidato a actor del sistema; y si algún actor del negocio va a interactuar con el sistema, entonces también podría ser un actor del mismo.

A continuación se muestra en la tabla 2.3 el actor del sistema y su correspondiente justificación:

Actores del sistema	Justificación
Directivo	Es el actor del sistema que se va a beneficiar con las distintas actividades que se van a automatizar en el sistema, como son: gestionar estrategias, gestionar indicadores, calcular los mismos, gestionar unidad de medidas y fórmulas.

Tabla 2.3: Descripción de los actores del sistema.

2.5.2 Diagrama de Casos de Uso del Sistema

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores. (Ver figura 2.2).

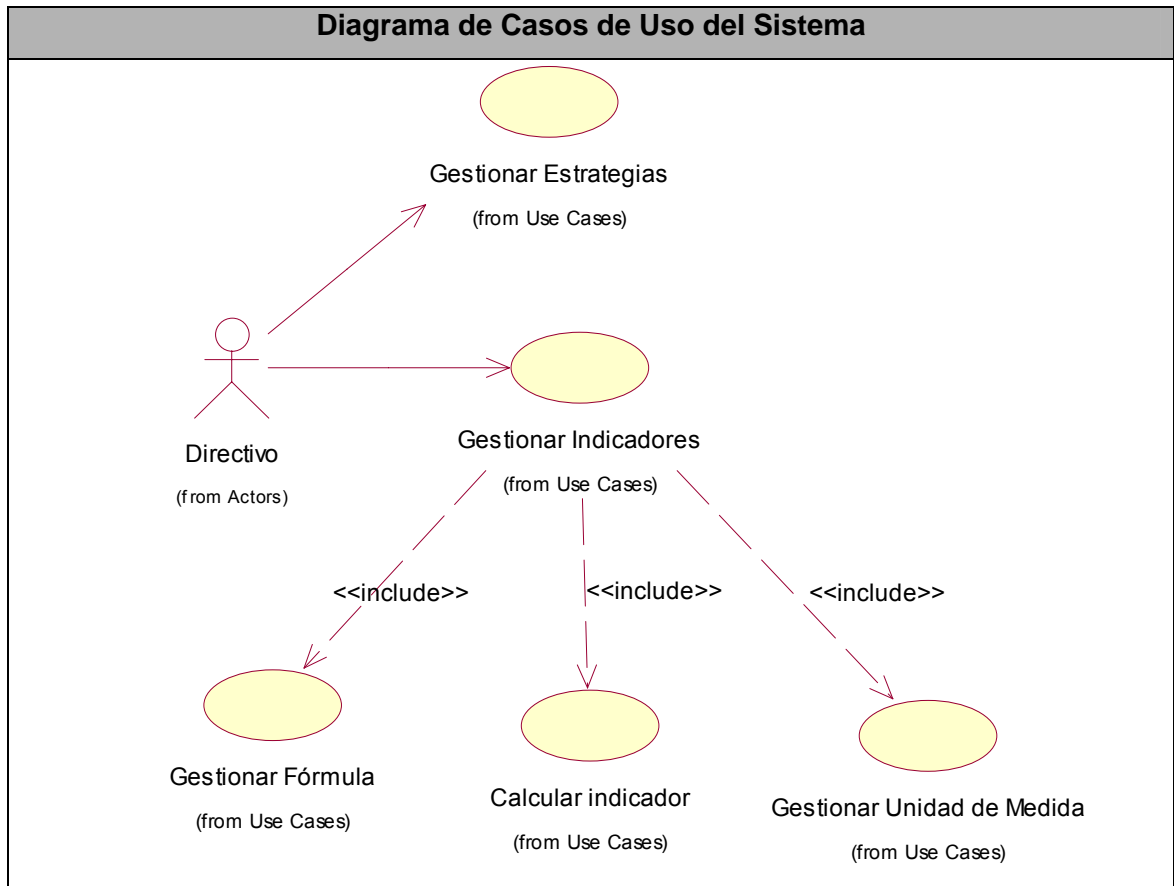


Figura 2.2: Diagrama de Casos de Uso del Sistema

2.5.3 Descripción textual de los casos de uso del sistema. (VER ANEXO 4).

2.6 Conclusiones

Este capítulo ha descrito el proceso de la identificación de los procesos internos de un Cuadro de Mando Integral mediante los distintos diagramas del modelo del negocio. Asimismo se comenzó a desarrollar una propuesta de solución, obteniéndose las principales funcionalidades que deben tener el sistema y sus características. Todo esto trajo consigo un mejor entendimiento para darle respuesta al problema.

Capítulo 3: ANÁLISIS Y DISEÑO DEL SISTEMA.

3.1 Introducción

Después de haber definido y descrito en el capítulo anterior las funcionalidades deseadas y necesarias del sistema propuesto; se hace preciso la definición del desarrollo de esta aplicación.

El objetivo que trae consigo este capítulo, es la presentación de los diagramas de clases de análisis, diagramas de clases Web y diagramas de interacción del diseño. Se plantea además, la concepción general de la ayuda del sistema propuesto al igual que los patrones de diseño, tratamiento de errores, interfaz de usuario y seguridad seguida.

3.2 Modelo de Análisis

En la construcción del modelo de Análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes.

3.2.1 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. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas.

A continuación se muestran los diagramas de clases del análisis para los distintos Casos de Uso. (Ver figuras 3.1- 3.5).

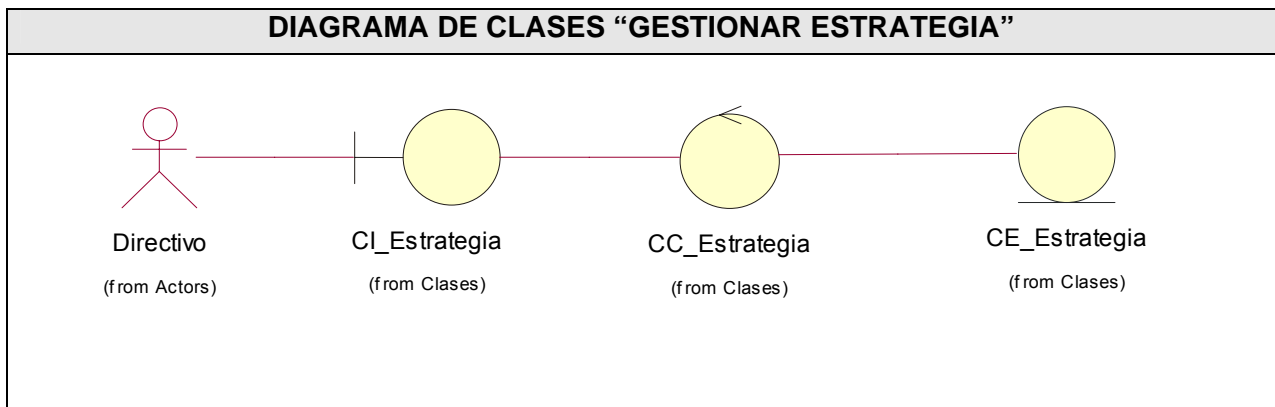


Figura 3.1: Diagrama de clases "Gestionar Estrategia".

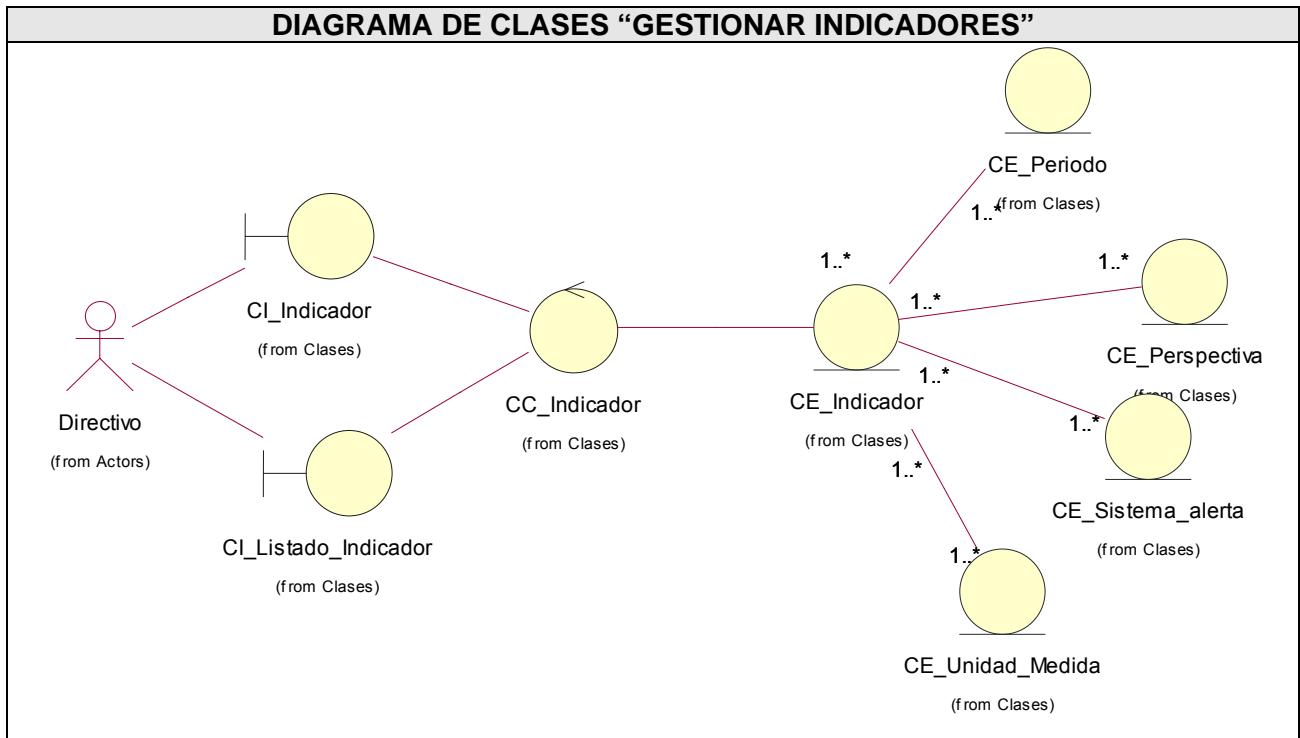


Figura 3.2: Diagrama de clases “Gestionar Indicadores”.

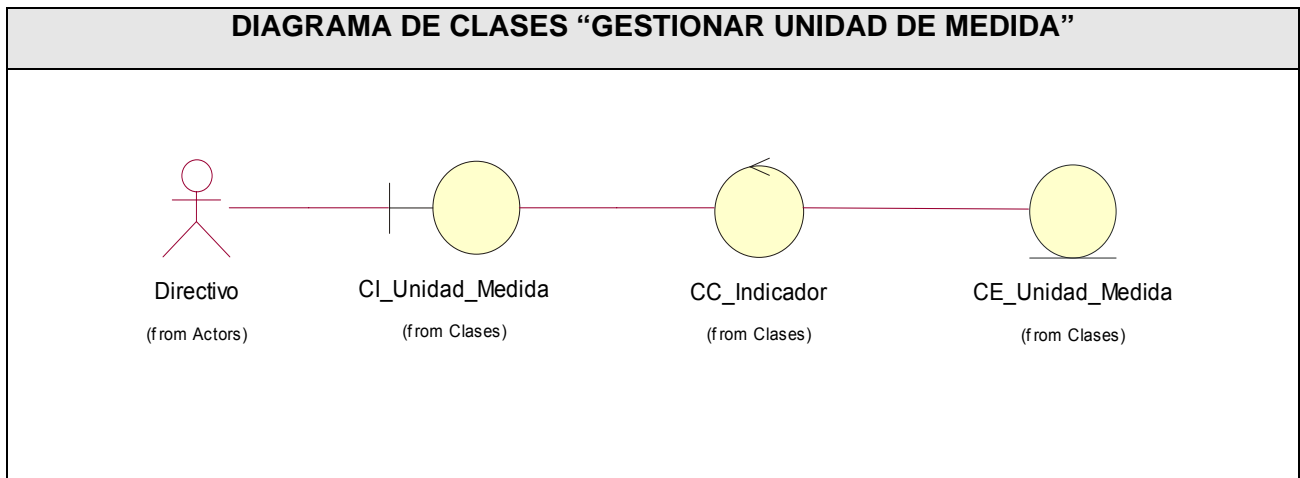


Figura 3.3: Diagrama de clases “Gestionar Unidad de Medida”.

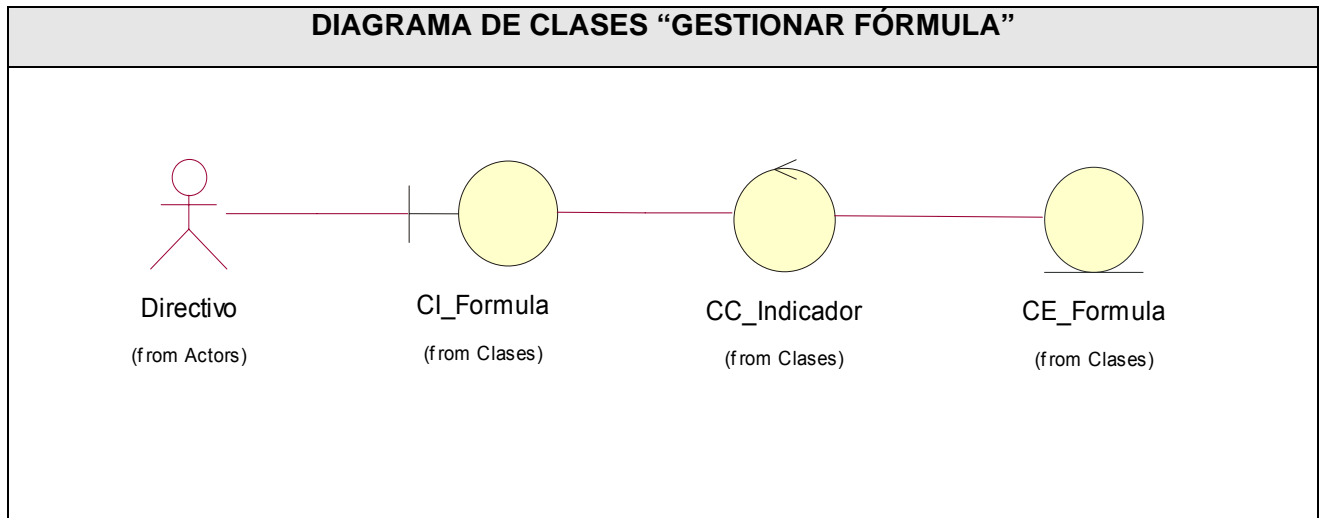


Figura 3.4: Diagrama de clases “Gestionar Fórmula”.

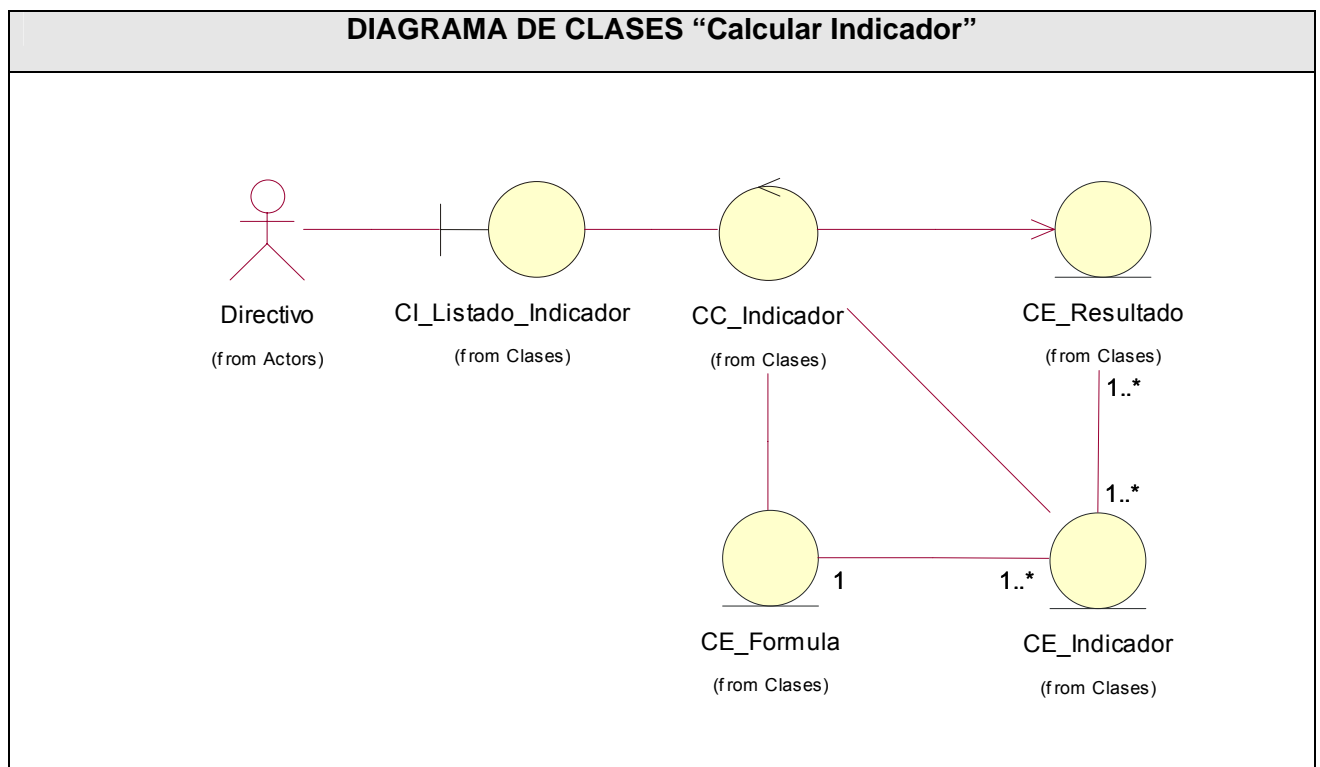


Figura 3.5: Diagrama de clases “Calcular Indicador”.

3.3 Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que describe la realización de los casos de uso y al mismo tiempo constituye una abstracción del modelo de implementación y del código fuente, constituye una entrada esencial a las actividades de implementación y prueba.

3.3.1 Diagrama de clases del diseño

Un diagrama de clases muestra las relaciones entre un conjunto de clases, interfaces, colaboraciones. En las aplicaciones Web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase, es muy importante pues estos son los artefactos que se necesitan modelar para que el desarrollador los implemente y obtenga de esta forma el producto final con la calidad requerida.

El diagrama de clases Web, fue definido, a partir de los diferentes casos de uso del sistema y empleando las extensiones de UML para Web.

A continuación se muestran los diagramas de clases Web del diseño para los distintos Casos de Uso. (Ver figuras 3.6 – 3.10).

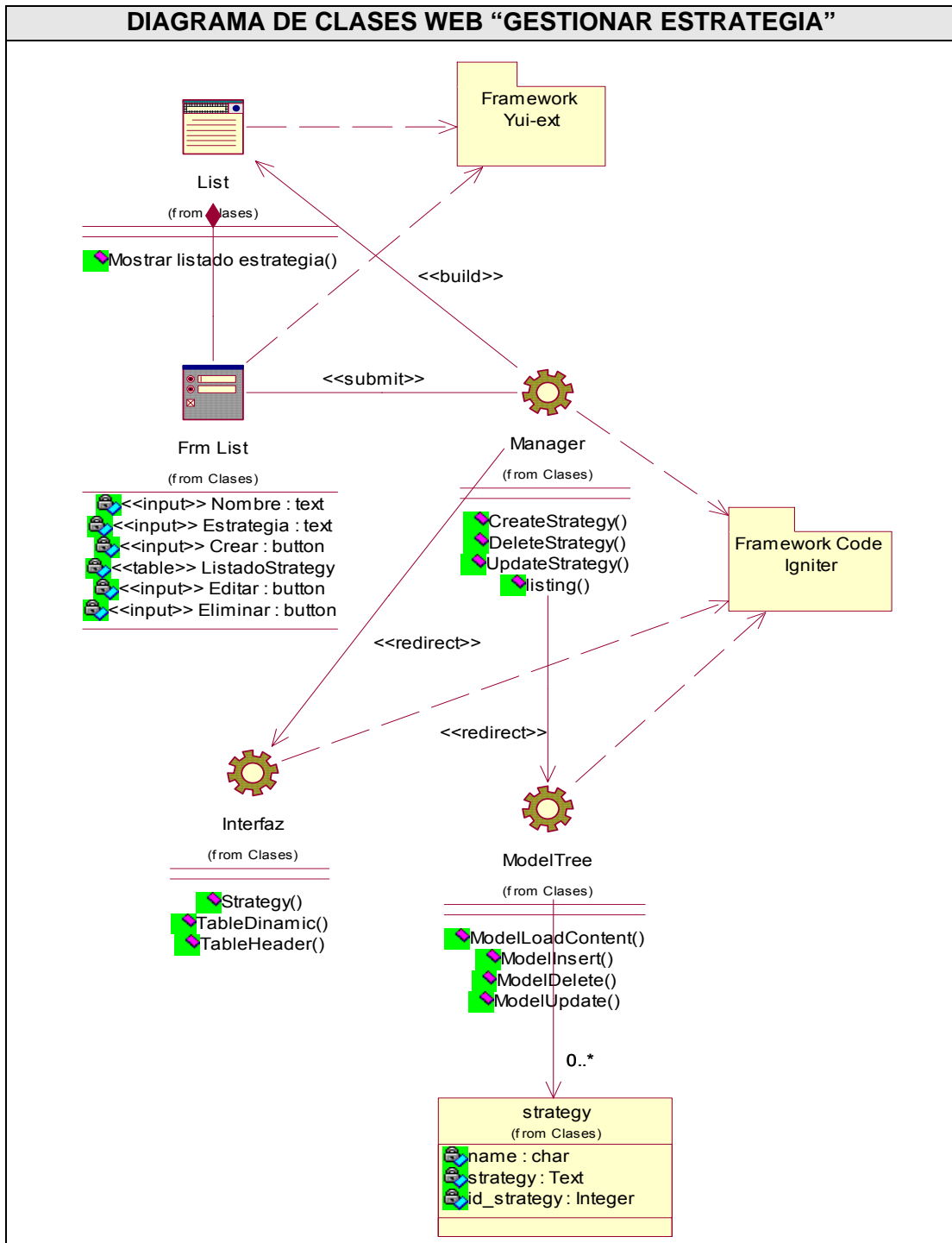


Figura 3.6: Diagrama de clases Web “Gestionar Estrategia”.

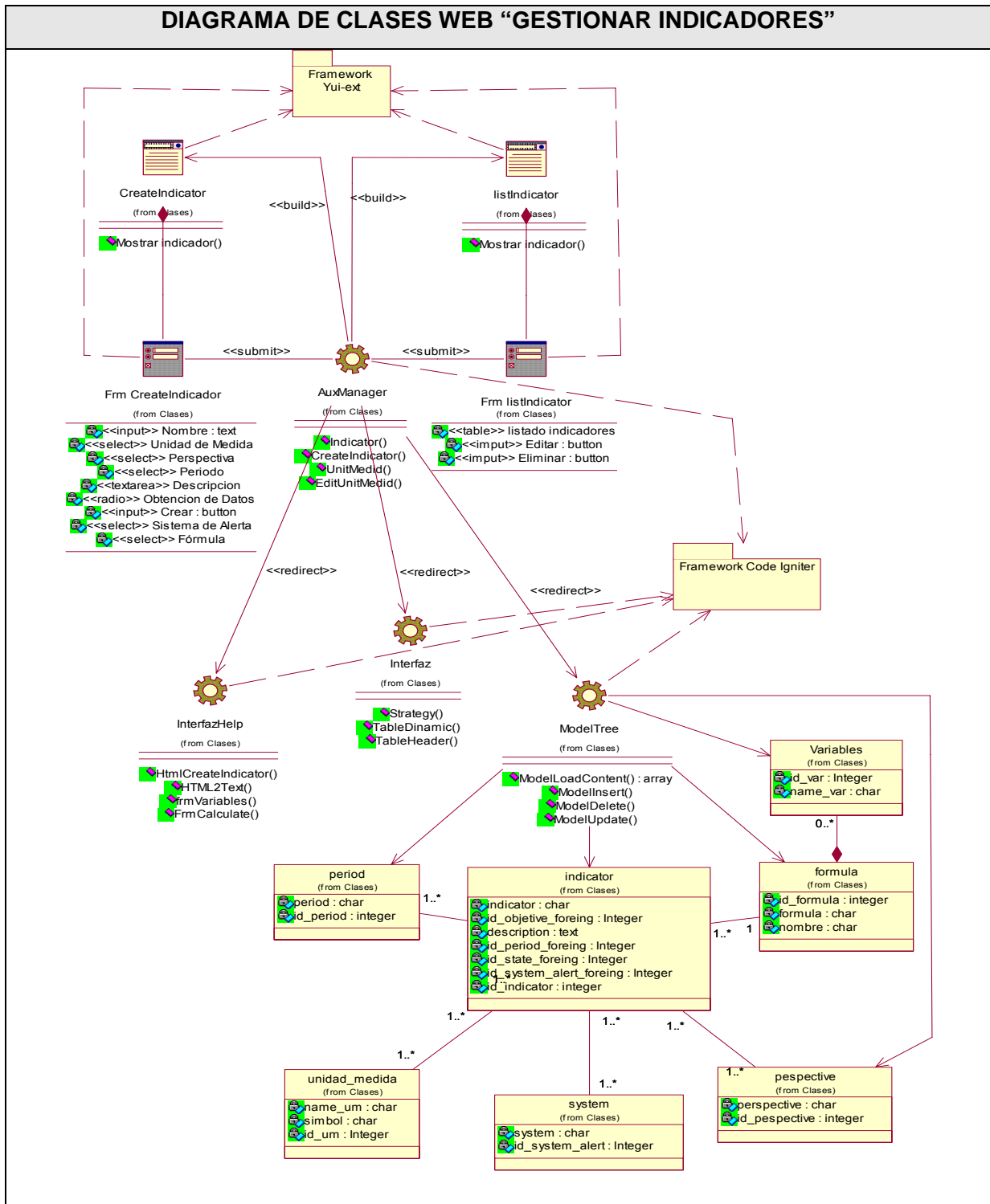


Figura 3.7: Diagrama de clases Web “Gestionar Indicadores”.

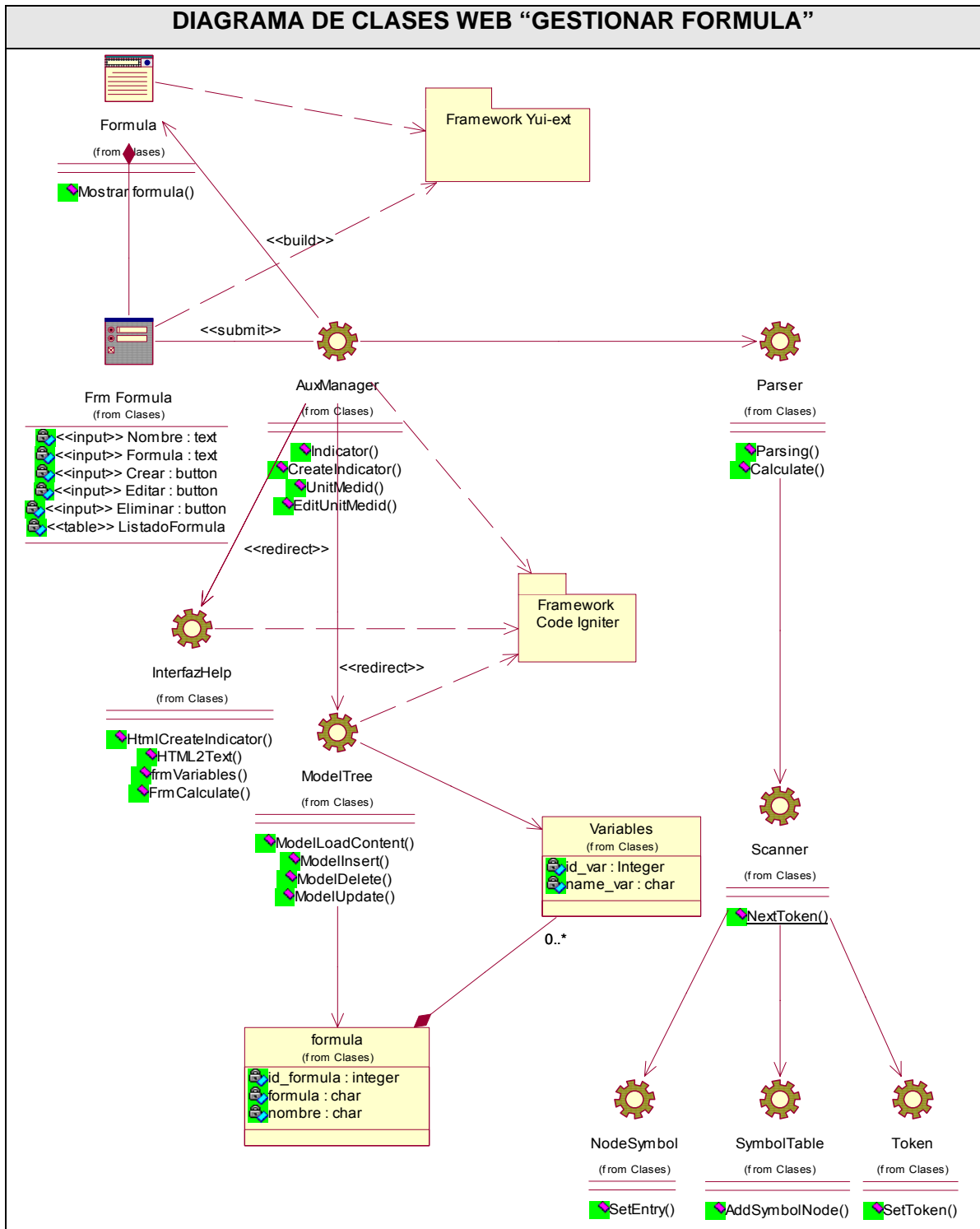


Figura 3.8: Diagrama de clases Web “Gestionar Fórmula”.

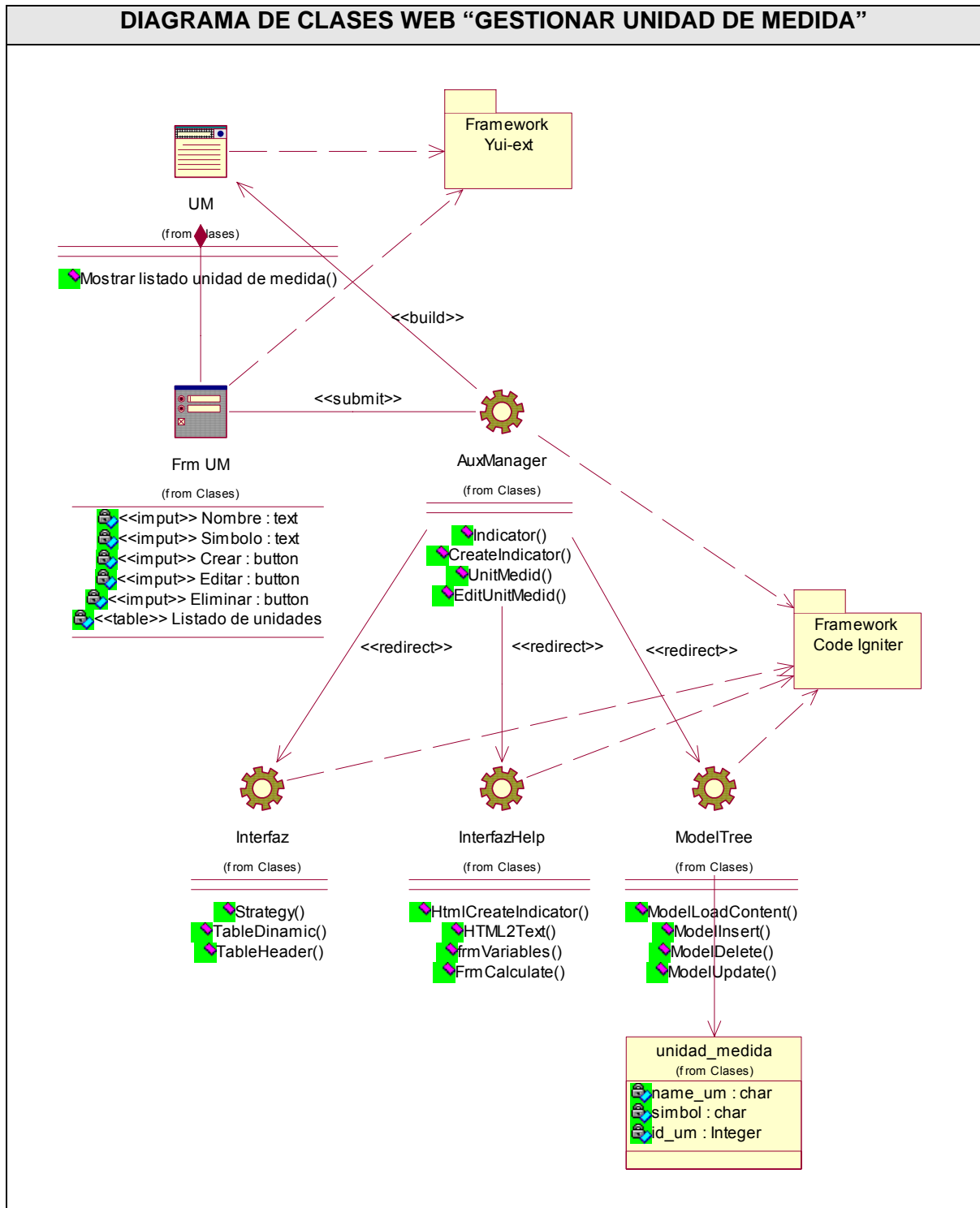


Figura 3.9: Diagrama de clases Web “Gestionar Unidad de Medida”.

3.4 Diagramas de interacción

Los diagramas de secuencia y los diagramas de colaboración (ambos llamados diagramas de interacción) son dos de los cinco tipos de diagramas de UML que se utilizan para modelar los aspectos dinámicos de los sistemas. Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

3.4.1 Diagramas de colaboración. (VER ANEXO 5).

Un diagrama de colaboración destaca la organización de los objetos que participan en una interacción. Presenta dos características principales que lo distingue de los diagramas de secuencia:

- el camino: para indicar cómo se enlaza un objeto con otro.
- el número de secuencia: para indicar el orden temporal de los mensajes; se precede de un número (comenzando con el mensaje número 1), que se incrementa secuencialmente por cada nuevo mensaje en el flujo de control (2, 3, etc.).

3.5 Descripción de las clases del diseño

A continuación se muestran las descripciones de las clases del diseño. (Ver tablas 3.1 – 3.14)

Nombre: Manager	
Tipo de clase: Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CreateStrategy()
Descripción:	Se encarga de crear una estrategia.
Nombre:	DeleteStrategy()
Descripción:	Se encarga de eliminar una estrategia.
Nombre:	UpdateStrategy()
Descripción:	Se encarga de actualizar una estrategia.
Nombre:	Listing(\$table="t_strategy",\$name_field="Estrategias",\$show="name",\$id="id_s

	trategy",\$sid_delete='url5',\$sid_edit='url3',\$sid_container='container_table2',\$sid_co ntainer_edit='insert)
Descripción:	Une el código de tableDinamic y table header y devuelve el html de la tabla.

Tabla 3.1: Descripción de la clase del diseño “Manager”.

Nombre: AuxManager	
Tipo de clase: Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	Indicator()
Descripción:	Elimina un indicador.
Nombre:	CreateIndicator()
Descripción:	Crea un indicador.
Nombre:	UnitMedid()
Descripción:	Crea y elimina una unidad de medida.
Nombre:	EditUnitMedid()
Descripción:	Edita una unidad de medida.

Tabla 3.2: Descripción de la clase del diseño “AuxManager”.

Nombre: Token	
Tipo de clase: Controladora	
Atributo:	Tipo
\$position	private
\$type	private
Para cada responsabilidad:	
Nombre:	SetToken(\$type_lex,\$position_lex)
Descripción:	Carga el token.
Nombre:	GetType()

Descripción:	Devuelve la variable \$type.
Nombre:	GetPosition()
Descripción:	Devuelve la variable \$position.

Tabla 3.3: Descripción de la clase del diseño "Token".

Nombre: Parser	
Tipo de clase: Controladora	
Atributo:	Tipo
\$scanner	private
\$notpolaca	private
Para cada responsabilidad:	
Nombre:	__autoload(\$class_name)
Descripción:	Carga automáticamente un fichero según el nombre de la clase que se le pase.
Nombre:	SetFormula(\$formula)
Descripción:	Carga la fórmula.
Nombre:	Parsing()
Descripción:	Método principal del analizador sintáctico.
Nombre:	Init()
Descripción:	Devuelve verdadero si la formula comenzó correctamente.
Nombre:	Oper_Asig()
Descripción:	Devuelve verdadero si encontró operador de asignación.
Nombre:	Body()
Descripción:	Devuelve verdadero si el cuerpo de la fórmula es correcto.
Nombre:	Exp1()
Descripción:	Identifica si detrás de un operador se encuentra un número o una variable.
Nombre:	End()
Descripción:	Encuentra el fin de la fórmula.
Nombre:	ExpGeneral()

Descripción:	Devuelve si una expresión hallada esta correcta.
Nombre:	Number()
Descripción:	Devuelve verdadero si encontró un número.
Nombre:	Key_Close()
Descripción:	Verifica si hay una llave que cierra.
Nombre:	Bracket_Close()
Descripción:	Verifica si hay un corchete que cierra.
Nombre:	Parenth_Close()
Descripción:	Verifica si hay un paréntesis que cierra.
Nombre:	Oper_Arit()
Descripción:	Verifica si hay operador aritmético.
Nombre:	Letter()
Descripción:	Verifica si hay una variable.
Nombre:	AddNotPolaca(\$var)
Descripción:	Realiza la notación polaca.
Nombre:	Priority(\$token,\$oper="")
Descripción:	Prioridad de los token operador de asignación y llaves.

Tabla 3.4: Descripción de la clase del diseño "Parser".

Nombre: SymbolTable	
Tipo de clase: Controladora	
Atributo:	Tipo
\$array_symbolNode	private
\$length	private
Para cada responsabilidad:	
Nombre:	AddSymbolNode(\$node)
Descripción:	Agrega un nodo a la tabla de símbolos.
Nombre:	GetToken(\$position)
Descripción:	Devuelve un token según una posición.

Nombre:	GetLexeme(\$position)
Descripción:	Devuelve el lexema del token según la posición.
Nombre:	GetArraySymbolNode()
Descripción:	Devuelve la tabla de símbolos.
Nombre:	GetLengthSymbolTable()
Descripción:	Devuelve la longitud de la tabla de símbolos.

Tabla 3.5: Descripción de la clase del diseño “SymbolTable”.

Nombre: SymbolNode	
Tipo de clase: Controladora	
Atributo:	Tipo
\$lexeme	private
\$type	private
Para cada responsabilidad:	
Nombre:	SetEntry(\$new_type,\$lex)
Descripción:	Cargar las variables privadas con los valores que se pasan como parámetro.
Nombre:	GetLexeme()
Descripción:	Devuelve la variable \$lexeme.
Nombre:	GetType()
Descripción:	Devuelve la variable \$type.

Tabla 3.6: Descripción de la clase del diseño “SymbolNode”.

Nombre: Scanner	
Tipo de clase: Controladora	
Atributo:	Tipo
\$formula	private
\$start	private
\$symbolTable	private
Para cada responsabilidad:	

Nombre:	__autoload(\$class_name)
Descripción:	Carga automáticamente un fichero según el nombre de la clase que se le pase.
Nombre:	GetSymbolTable()
Descripción:	Devuelve la tabla de símbolos.
Nombre:	GetLexeme(\$position)
Descripción:	Devuelve el lexema de un token según la posición.
Nombre:	SetFormula(\$formula1)
Descripción:	Carga el atributo \$formula.
Nombre:	isLetter(\$str)
Descripción:	Verifica que el carácter de entrada sea letra.
Nombre:	Number(\$str)
Descripción:	Verifica si el carácter de entrada es un número.
Nombre:	Retract()
Descripción:	Posición anterior de lectura de la fórmula.
Nombre:	NextChar()
Descripción:	Próxima posición de lectura de la fórmula.
Nombre:	AddToSymbolTable(\$type,\$char)
Descripción:	Añade un símbolo a la tabla de símbolo.
Nombre:	ReturnToken(\$type,\$char)
Descripción:	Devuelve un token.
Nombre:	NextToken()
Descripción:	Realiza un análisis léxico de la fórmula.

Tabla 3.7: Descripción de la clase del diseño "Scanner".

Nombre: Interfaz	
Tipo de clase: Controladora	
Atributo:	Tipo
Para cada responsabilidad:	

Nombre:	Strategy(\$accion_strategy,\$array)
Descripción:	Devuelve el HTML de un formulario que hace falta para editar.
Nombre:	TableDinamic(\$array,\$name,\$id,\$urlId_delete='url5',\$urlId_edit='url3',\$id_container='container_table2',\$idEdit='insert',\$style="Nofixed2a")
Descripción:	Devuelve el HTML de una tabla con datos.
Nombre:	TableHeader(\$table,\$name=",\$container='container_table2',\$Idurl='url2')
Descripción:	Devuelve el HTML de una tabla que hace función de cabecera en la tabla dinámica.

Tabla 3.8: Descripción de la clase del diseño “Interfaz”.

Nombre: InterfazHelp	
Tipo de clase: Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	HtmlCreateIndicator(\$array_um,\$array_perspective,\$array_sa)
Descripción:	Devuelve parte de un formulario para crear un indicador.
Nombre:	HTML2Text(\$array,\$field='name_um',\$id_var='id_um',\$text='simbol',\$accion='EditUnitMedid')
Descripción:	Envía el formulario para editar una unidad de medida.

Tabla 3.9: Descripción de la clase del diseño “InterfazHelp”.

Nombre: ModelTree	
Tipo de clase: Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	ModelLoadContent(\$field,\$table,\$condition='2=2')
Descripción:	Devuelve un arreglo de datos necesarios.
Nombre:	ModelInsert(\$object,\$table)

Descripción:	Se encarga de insertar duplas a una tabla especificada.
Nombre:	ModelDelete(\$id,name_id,\$table)
Descripción:	Elimina duplas en una tabla especificada según un identificador pasado como parámetro.
Nombre:	ModelUpdate(\$id_name,\$id_content,\$table,\$data)
Descripción:	Actualiza duplas en una tabla especificada.

Tabla 3.10: Descripción de la clase del diseño "ModelTree".

Nombre: List	
Tipo de clase: Interfaz	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	Mostrar listado estrategia.
Descripción:	Muestra el listado de las estrategias con sus resultados.

Tabla 3.11: Descripción de la clase del diseño "List".

Nombre: CreateIndicator	
Tipo de clase: Interfaz	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	Mostrar indicador()
Descripción:	Muestra el listado de los indicadores con sus resultados.

Tabla 3.12: Descripción de la clase del diseño "CreateIndicator".

Nombre: ListIndicator	
Tipo de clase: Interfaz	
Atributo:	Tipo

Para cada responsabilidad:	
Nombre:	Mostrar indicador()
Descripción:	Muestra el listado de los indicadores con sus resultados.

Tabla 3.13: Descripción de la clase del diseño "ListIndicador".

Nombre: UM	
Tipo de clase: Interfaz	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	Mostrar listado unidad de medida()
Descripción:	Muestra el listado de las unidades de medidas con sus resultados.

Tabla 3.14: Descripción de la clase del diseño "UM".

3.6 Modelo de datos

El modelo de los datos describe la representación lógica y física de datos persistentes en el sistema. (Ver figura 3.11).

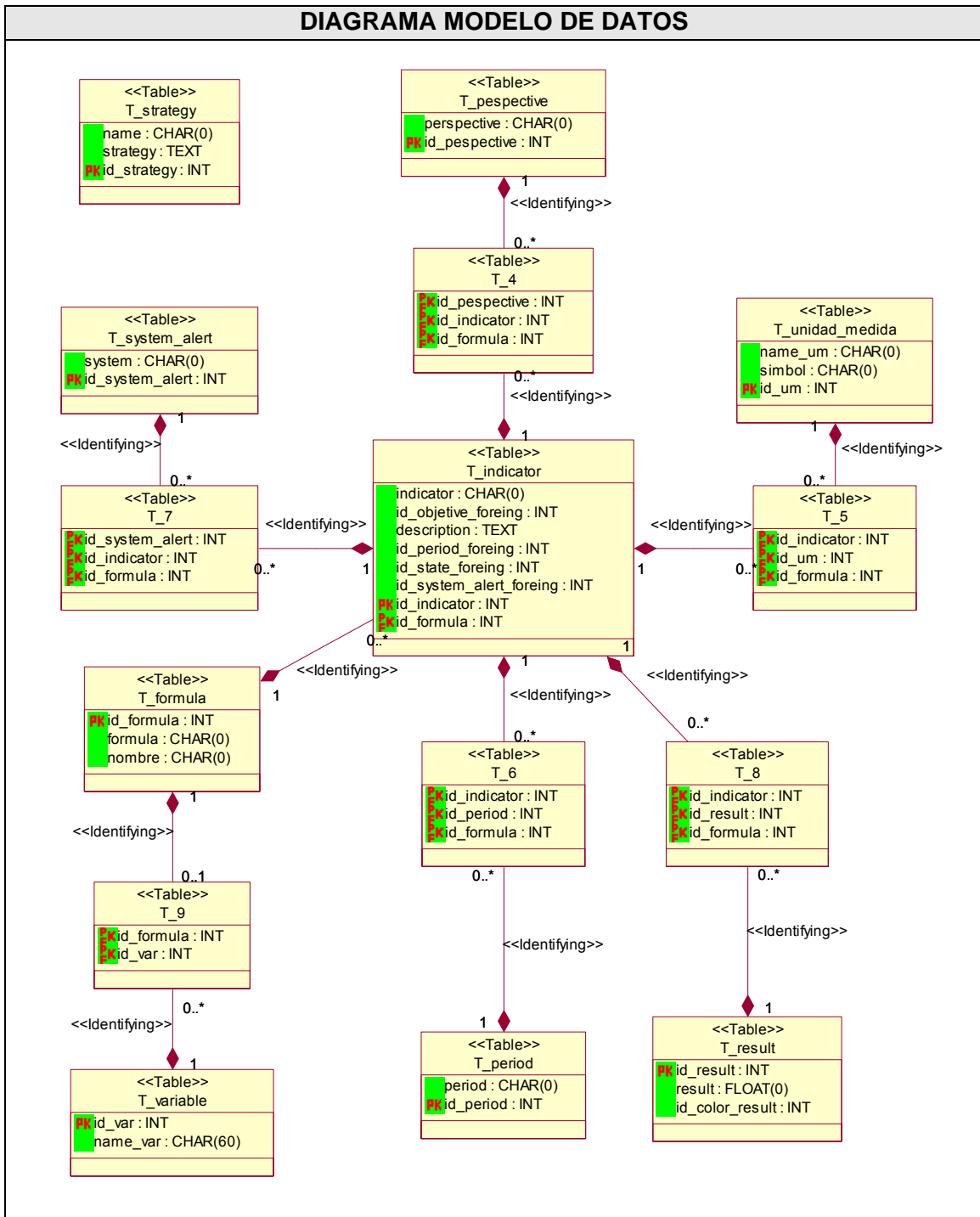


Figura 3.11: Diagrama Modelo de Datos

3.7 Descripción de las tablas de la Base de Datos

A continuación se muestran las descripciones de las tablas de la base de datos. (Ver tablas 3.15 – 3.29)

Nombre: T_strategy		
Descripción: Almacena los datos referentes a estrategias.		
Atributo	Tipo	Descripción
id_strategy	Integer	Llave primaria de la tabla.
name	Char(60)	Nombre de la estrategia.
strategy	Text	Estrategia.

Tabla 3.15: Descripción de la tabla de la base de datos “T_strategy”.

Nombre: T_indicator		
Descripción: Almacena los datos referentes a indicadores.		
Atributo	Tipo	Descripción
Id_indicator	Integer	Llave primaria de la tabla.
Indicator	Char(60)	Nombre del indicador.
id_objective_foreing	Integer	Llave foránea.
description	Text	Descripción del indicador.
id_period_foreing	Integer	Llave foránea.
id_system_alert_foreing	Integer	Llave foránea.
Id_formula	Integer	Llave foránea.

Tabla 3.16: Descripción de la tabla de la base de datos “T_indicator”.

Nombre: T_period		
Descripción: Almacena los datos referentes a periodos de indicadores.		
Atributo	Tipo	Descripción
id_period	Integer	Llave primaria de la tabla.
period	Char(60)	Período del indicador.

Tabla 3.17: Descripción de la tabla de la base de datos “T_period”.

Nombre: T_perspective		
Descripción: Almacena los datos referentes a las perspectivas del sistema.		
Atributo	Tipo	Descripción
id_perspective	Integer	Llave primaria de la tabla.
perspective	Char(60)	Nombre de la perspectiva.

Tabla 3.18: Descripción de la tabla de la base de datos “T_perspective”.

Nombre: T_system_alert		
Descripción: Almacena los sistemas de alerta que tiene cada indicador.		
Atributo	Tipo	Descripción
id_system_alert	Integer	Llave primaria de la tabla.
system	Char(60)	Nombre del sistema de alerta.

Tabla 3.19: Descripción de la tabla de la base de datos "T_system_alert".

Nombre: T_formula		
Descripción: Almacena los datos referentes a periodos de indicadores.		
Atributo	Tipo	Descripción
id_formula	Integer	Llave primaria de la tabla.
name	Char(60)	Nombre de la fórmula.
formula	Char(60)	Fórmula del indicador.

Tabla 3.20: Descripción de la tabla de la base de datos "T_formula".

Nombre: T_perspective_indicador		
Descripción: Almacena la lista de indicadores que tiene cada perspectiva.		
Atributo	Tipo	Descripción
id_perspective	Integer	Llave primaria de la tabla T_perspective.
id_indicador	Integer	Llave primaria de la tabla T_indicador.
id_formula	Integer	Llave foránea de la tabla T_indicador.

Tabla 3.21: Descripción de la tabla de la base de datos "T_perspective_indicador".

Nombre: T_result		
Descripción: Almacena los resultados de cada indicador.		
Atributo	Tipo	Descripción
id_result	Integer	Llave primaria de la tabla.
result	Float	Resultado del cálculo de un indicador.
id_color_result	Integer	Llave foránea.

Tabla 3.22: Descripción de la tabla de la base de datos "T_result".

Nombre: T_indicador_result		
Descripción: Almacena la lista de indicadores con resultados.		
Atributo	Tipo	Descripción
id_result	Integer	Llave primaria de la tabla T_result.
id_indicador	Integer	Llave primaria de la tabla T_indicador.
id_formula	Integer	Llave foránea de la tabla T_indicador.

Tabla 3.23: Descripción de la tabla de la base de datos "T_indicador_result".

Nombre: T_unidad_medida		
Descripción: Almacena la unidad de medida de cada indicador.		
Atributo	Tipo	Descripción
id_um	Integer	Llave primaria de la tabla.
name_um	Char	Nombre de la unidad de medida
simbol	Char	Símbolo de la unidad de medida.

Tabla 3.24: Descripción de la tabla de la base de datos "T_unidad_medida".

Nombre:T_indicador_um		
Descripción: Almacena la lista de indicadores con unidades de medidas.		
Atributo	Tipo	Descripción
id_indicador	Integer	Llave primaria de la tabla T_indicador.
id_um	Integer	Llave primaria de la tabla T_unidad_medida.
id_formula	Integer	Llave foránea de la tabla T_indicador.

Tabla 3.25: Descripción de la tabla de la base de datos "T_indicador_um".

Nombre:T_variable		
Descripción: Almacena las variables		
Atributo	Tipo	Descripción
Id var	Integer	Llave primaria de la tabla.
name_var	char (60)	Nombre de la variable.

Tabla 3.26: Descripción de la tabla de la base de datos "T_variable".

Nombre:T_formula_variable		
Descripción: Almacena las variables de cada fórmula.		
Atributo	Tipo	Descripción
Id var	Integer	Llave primaria de la tabla T_variable.
id_formula	integer	Llave primaria de la tabla T_formula.

Tabla 3.27: Descripción de la tabla de la base de datos "T_formula_variable".

Nombre:T_system_alert_indicador		
Descripción: Almacena el sistema de alerta de cada indicador.		
Atributo	Tipo	Descripción
id_system_alert	Integer	Llave primaria de la tabla T_system_alert.
id_indicador	Integer	Llave primaria de la tabla T_indicador.
id_formula	Integer	Llave foránea de la tabla T_indicador.

Tabla 3.28: Descripción de la tabla de la base de datos "T_system_alert_indicador".

Nombre:T_period_indicator		
Descripción: Almacena el periodo de cada indicador.		
Atributo	Tipo	Descripción
id_period	Integer	Llave primaria de la tabla T_period.
id_indicator	Integer	Llave primaria de la tabla T_indicator.
id_formula	Integer	Llave foránea de la tabla T_indicator.

Tabla 3.29: Descripción de la tabla de la base de datos "T_period_indicator".

3.8 Definiciones de diseño

3.8.1 Patrones

La utilización de los patrones ha tomado un gran auge a partir del desarrollo del modelo orientado a objetos. La frase de Grady Booch que a continuación se presenta demuestra lo anterior.

“Una arquitectura orientada a objetos bien estructurada está llena de patrones. La calidad de un sistema orientado a objetos 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”.

3.8.1.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). El nombre se eligió para indicar la importancia de captar (grasping) estos principios, si se quiere diseñar eficazmente el software orientado a objetos.

- *Experto.* Asigna una responsabilidad al experto la clase que cuenta con la información necesaria para cumplir la responsabilidad. Un ejemplo en el sistema que se propone son las clases Interfaz e Interfazhelp, quienes tienen delegadas las responsabilidades de generar el código HTML de formularios.
- *Creador.* Asigna a la clase **B** la responsabilidad de crear una instancia de la clase **A**. La clase Manager y AuxManager son las que se encargan de crear los objetos fabricados por la clase Modeltree, en el caso del sistema descrito.
- *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. En el sistema propuesto existen varias clases controladoras. Ellas son las clases Manager y AuxManager.

3.8.1.2 Patrones de diseño Web

- *Alternating Row Colors.* Este patrón es usado para mostrar datos en una tabla con muchos campos y fila; permite además darle más legibilidad al lector gracias al uso de colores alternados.
- *Action Button.* Es usado para diferenciarle al usuario los vínculos de las páginas a los eventos más importantes, como estrategias e indicadores.
- *Footer Bar.* Se utiliza para dar a conocer las condiciones bajo las cuales se puede usar la aplicación. Este patrón está expresado en la parte inferior de la aplicación, donde se explica los derechos de copyright que se atribuye a la universidad sobre este software.
- *Tabs.* El uso de los tabs es para llevar una navegación horizontal, agrupar la información por categoría y mostrar parcialmente el contenido por secciones. El uso de los tabs en la aplicación es muy extendida, por lo que este patrón ha facilitado mucho la realización de la misma.
- *Image Menú.* Muy usado para ahorrar espacio en el menú y puede dar la posibilidad de enfatizar el icono seleccionado. Este tipo de diseño permitió crear una barra de herramienta para ubicar las funcionalidades de la aplicación.
- *Topic Pages.* Este patrón describe el uso de encabezados de documentos para ayudar a comprender al usuario el tema expuesto. En el caso de la aplicación se usan los encabezados en la cabecera de las ventanas para explicar a que se dedica la ventana levantada.
- *Retractable Menú.* Este tipo de menú es muy útil en aquellos que ocupan mucho espacio, por ende la posibilidad de encoger el menú es una opción muy buena a la hora de optimizar el espacio dedicado a la navegación o al tema principal de la aplicación. En la aplicación, se encuentra una navegación hecha por un árbol que pudiera robar mucho espacio, por tanto el uso de este patrón ha dado solución al problema planteado.

- *Split Navigation*. Este patrón describe que la navegación puede ser a través de una estructura jerárquica o por niveles. Permite ir más directo a la información que se busca, sin tener que pasar por otras páginas primero. El uso de la navegación en la aplicación a través de un árbol permite las facilidades de la estructura jerárquica enunciadas por el split navigation.
- *Customizable Windows*. Se refiere a la posibilidad de dar al usuario la opción de personalizar las ventanas desplegadas por el software. Este patrón se manifiesta en todas las ventanas del Cuadro de Mando Integral al poder, el usuario, redimensionar las ventanas al tamaño que estime.

3.9 Tratamiento de errores

Los males están provocados por defectos y errores del sistema que se minimizan a través de una serie de estrategias implementadas y otras que ya brinda el Framework de PHP Code Igniter. Una vez terminada la programación en cada caso de uso, se somete a determinados escenarios de entrada de datos para evitar la presencia de errores y se evalúan las posibles respuestas del sistema.

3.10 Seguridad

La seguridad fue tomada en cuenta a la hora de la entrada de los datos y en la selección del framework, ya que, este aporta una estructura e implementa políticas de seguridad para el desarrollo de la aplicación. Antes que los datos sean guardados en la base de datos es pasado por un filtrado evitando posibles errores o ataques intencionales del usuario. Son validadas las entradas vacías y se filtra las posibles inyecciones SQL y XSS, este último filtrado lo proporciona el framework de una manera automática.

3.11 Interfaz de usuario

La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

Para el desarrollo de la interfaz se tuvo en cuenta los siguientes aspectos:

- Sencillez.
- Atractiva.
- Flexibilidad.

- Apariencia entorno de escritorio para dar más característica de una aplicación y semejanza con otros sistemas que también implementan el Cuadro de Mando Integral y sirva para trasladar la experiencia a este sistema.
- Colores serios.
- Mensajes de errores certeros.

Se tuvieron además las *siguientes características*:

- Utilizar una misma tipografía, forma y estilo en todas las páginas.
- La facilidad del usuario de poder navegar desde cualquier punto a otro dentro de la aplicación.
- Se tuvo presente siempre el ancho de banda y por ello se utilizaron formato de imágenes de compresión favorables.
- La simplicidad y consistencia, favoreciendo la usabilidad de la aplicación.
- Navegación simple en todas las páginas de la aplicación, de forma tal que siempre sea accesible por el usuario.
- Estabilidad y uniformidad del diseño, para así poder ubicar al usuario dentro del mismo y hacerlo sentir parte de él.

3.12 Concepción de la ayuda

La ayuda se encuentra accesible, pues radica en la barra de herramienta brindada por el sistema. El usuario puede encontrar aspectos teóricos que le ayuden a comprender el uso del sistema y la importancia de la gestión del mismo. La ayuda está caracterizada para que los usuarios no necesiten ningún tipo de preparación o instrucción previa para utilizar el sistema.

3.13 Conclusiones

Se arribó al final de este de capítulo y se obtuvo como resultado del mismo los diagramas de clases de la aplicación y el diseño de su base de datos. Además se describió los principios de diseño seguidos, específicamente la concepción del tratamiento de errores y sistema de ayuda.

Capítulo 4: IMPLEMENTACIÓN

4.1 Introducción

Este capítulo se refiere al flujo de trabajo de implementación, el cual describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

Los diagramas de despliegue y componentes conforman lo que se conoce como un modelo de implementación al desplegar los componentes a construir, su organización y dependencia, entre nodos físicos en los que funcionará la aplicación.

4.2 Diagrama de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo; donde cada nodo representa un procesador o un dispositivo de hardware similar.

El diagrama de despliegue muestra la topología del hardware sobre el que se ejecuta el sistema. (Ver figura 4.1).

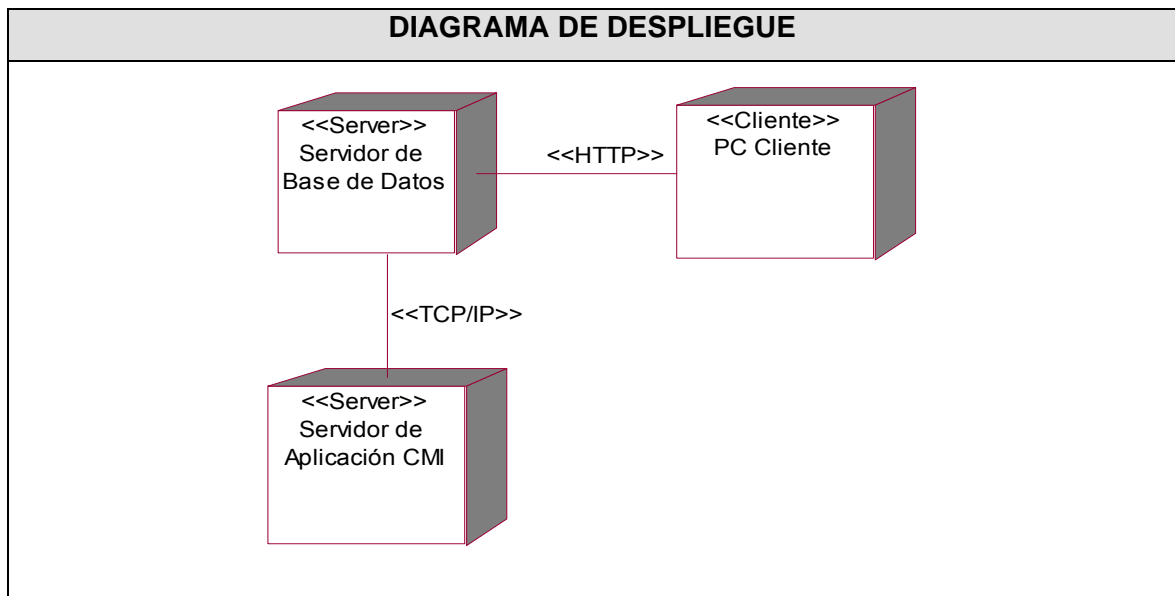


Figura 4.1: Diagrama de despliegue.

4.3 Diagrama de componentes

Es un diagrama que muestra un conjunto de elementos del modelo: componentes (sean estos de código fuente, librerías, ejecutables, tablas, archivo), subsistemas de implementación y sus relaciones, donde estas pueden ser <<link>>, <<compile>> u otro. (Ver figura 4.2- 4.3).

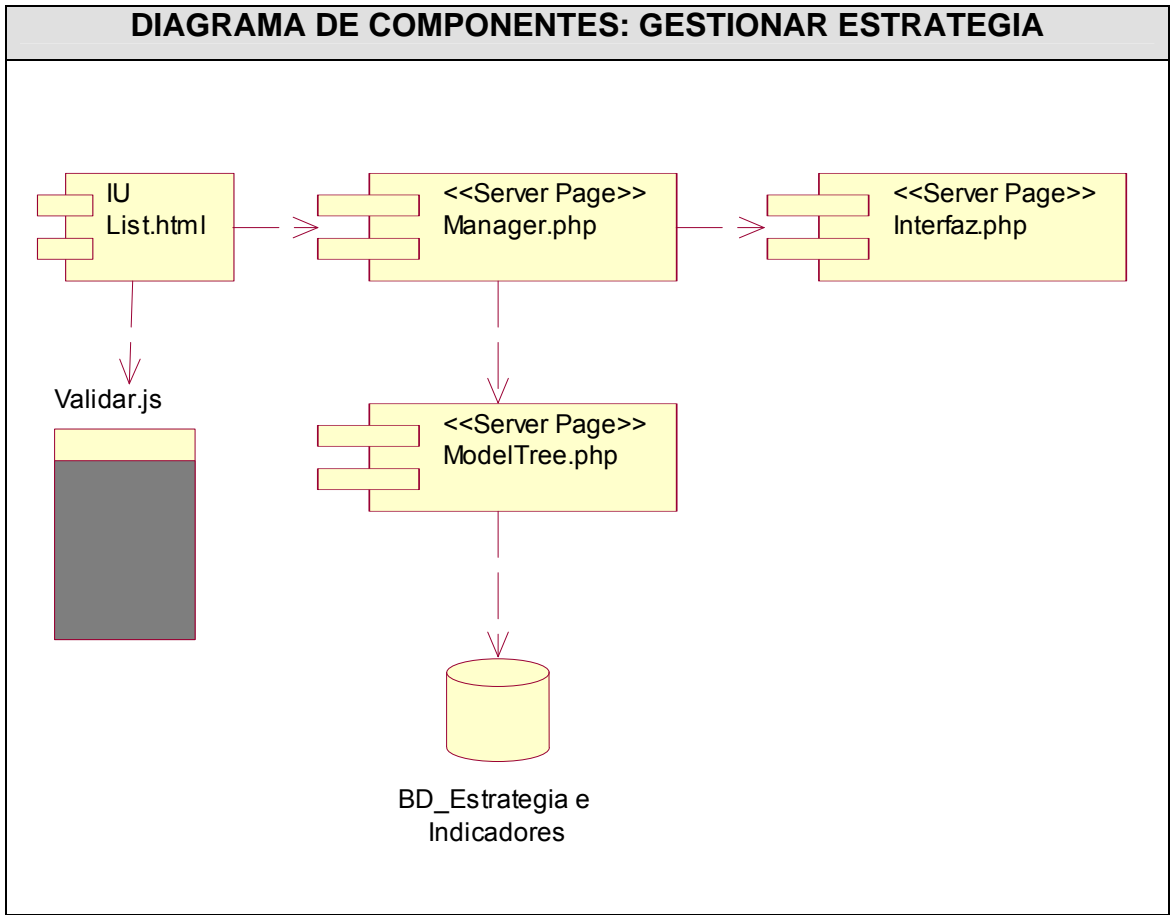


Figura 4.2: Diagrama de componente "Gestionar Estrategia".

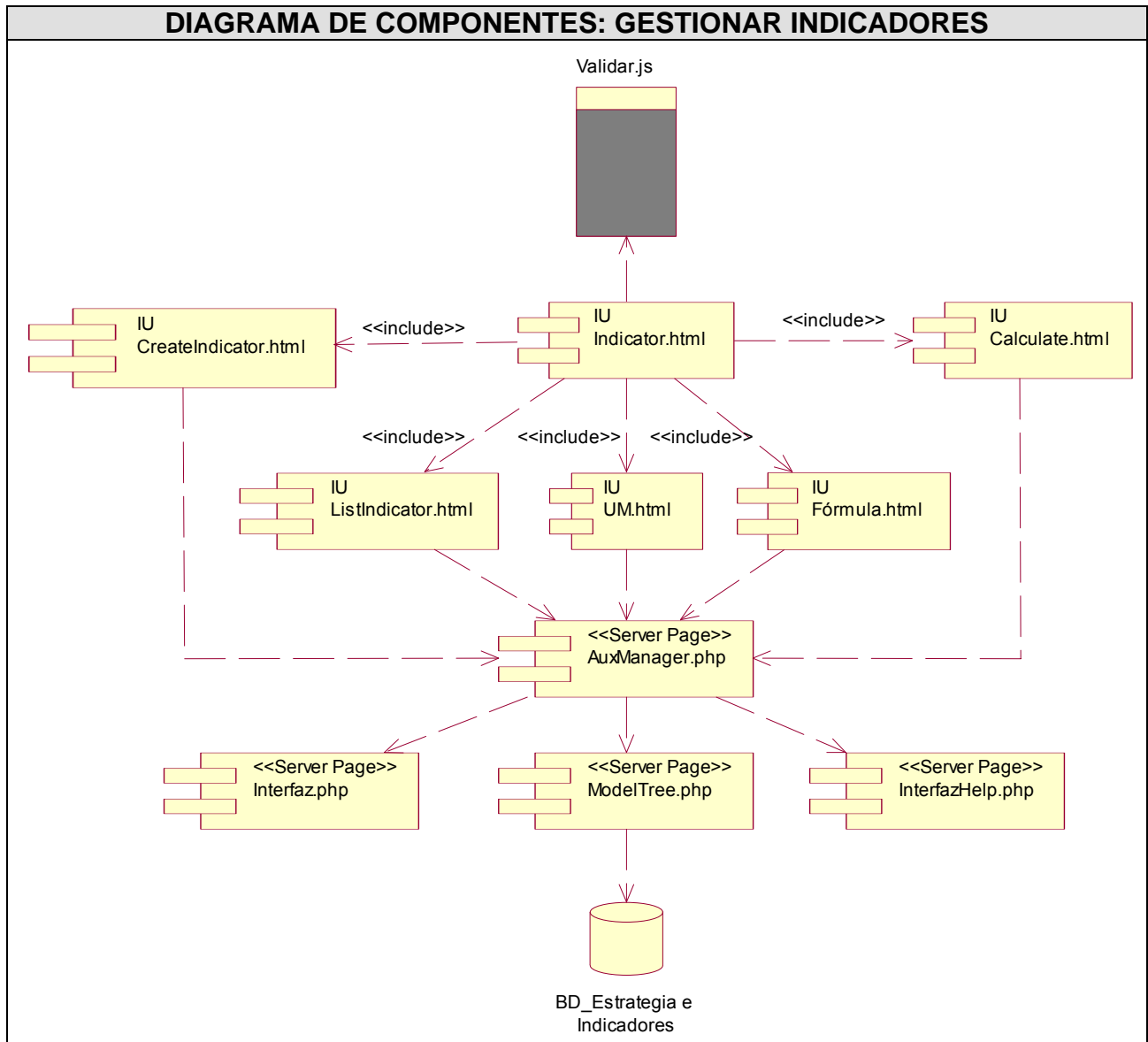


Figura 4.3: Diagrama de componente” Gestionar Indicadores”.

4.4 Conclusiones

En este capítulo que se concluye, se analizaron y realizaron los distintos diagramas de implementación; componente y despliegue que se corresponden con el diseño e implementación del software desarrollado, obteniéndose de esta manera, la topología del hardware sobre el que se ejecuta el sistema y la estructura del mismo.

Capítulo 5: ESTUDIO DE FACTIBILIDAD

5.1 Introducción

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, siendo así, en este capítulo se efectúa la estimación del mismo a través de la planificación basada en casos de uso.

5.2 Planificación basada en casos de uso.

5.2.1 Identificar los puntos de casos de usos desajustados

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de casos de uso sin ajustar

UAV: Factor de peso de los actores sin ajustar

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

5.2.2 Cálculo de UAV

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
Total			3

5.2.3 Cálculo de UUCW

Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5	0*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10	3*10
Complejo	El Caso de Uso contiene más de 8 transacciones	15	2*15
Total			60

UUCP = UAV + UUCW

UUCP = 3 + 60

UUCP = 63

5.3 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

EF: Factor de ambiente

5.3.1 Cálculo de TCF

TCF = 0.6 + 0.01 * Σ (Peso_i * Valor_i)

Donde:

0: No presente o sin influencia

1: Influencia incidental o presencia incidental

2: Influencia moderada o presencia moderada

3: Influencia media o presencia media

4: Influencia significativa o presencia significativa

5: Fuerte influencia o fuerte presencia

$$TCF = 0.6 + 0.01 * \sum (\text{Peso}_i * \text{Valor}_i)$$

$$TCF = 0.6 + 0.01 * 41$$

$$TCF = 1.01$$

5.3.2 Cálculo de EF

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor}_i)$$

Factor	Descripción	Peso	Valor	Comentario	$\sum (\text{Peso}_i * \text{Valor}_i)$
T1	Sistema distribuído.	2	0	El sistema es centralizado.	0
T2	Objetivos de performance o tiempo de respuesta.	1	4	El tiempo de respuesta es bastante rápido.	4
T3	Eficiencia del usuario final.	1	4	Es necesario que el usuario final conozca el sistema.	4
T4	Procesamiento interno complejo.	1	4	Presenta cálculos complejos.	4
T5	El código debe ser reutilizable.	1	5	Se requiere que el código sea reutilizable.	5
T6	Facilidad de instalación.	0.5	3	Posee algunos requerimientos de instalación.	1.5
T7	Facilidad de uso.	0.5	5	Fácil de usar.	2.5
T8	Portabilidad.	2	5		10
T9	Facilidad de cambio.	1	4		4
T10	Concurrencia.	1	0	No hay concurrencia.	3
T11	Incluye objetivos especiales de seguridad.	1	1		1
T12	Provee acceso directo a terceras partes.	1	0	Los usuarios web no tienen acceso.	0

T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	2	Pocos usuarios internos, sistema fácil de usar.	2
Total					41

Factor	Descripción	Peso	Valor	Comentario	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	3	El grupo está un poco familiarizado con el modelo.	4.5
E2	Experiencia en la aplicación.	0.5	1	La mayoría del grupo no tiene mucha experiencia en la aplicación.	0.5
E3	Experiencia en orientación a objetos.	1	4	Grupo programa en objetos.	4
E4	Capacidad del analista líder.	0.5	0	No se contrató a un especialista.	0
E5	Motivación.	1	4	El grupo está motivado.	4
E6	Estabilidad de los requerimientos.	2	2	Se esperan cambios.	4
E7	Personal part-time.	-1	4	El grupo todo el tiempo no es full-time.	-4
E8	Dificultad del lenguaje de programación.	-1	2	Se usará lenguaje PHP.	-2
Total					11

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor}_i)$$

$$EF = 1.4 - 0.03 * 11$$

$$EF = 1.07$$

Luego

$$UCP = UUCP * TCF * EF$$

$$UCP = 63 * 1.01 * 1.07$$

$$UCP = 68.08$$

5.4 Calcular esfuerzo FT implementación

$$E = UCP * CF$$

Donde:

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

5.4.1 Cálculo de CF

CF = 20 horas-hombre (si Total_{EF} ≤ 2)

CF = 28 horas-hombre (si Total_{EF} = 3 ó Total_{EF} = 4)

CF = abandonar o cambiar proyecto (si Total_{EF} ≥ 5)

$$\text{Total}_{EF} = \text{Cant EF} < 3 \text{ (entre E1 -E6)} + \text{Cant EF} > 3 \text{ (entre E7, E8)}$$

$$\text{Total}_{EF} = 2 + 0$$

$$\text{Total}_{EF} = 2$$

CF = 20 horas-hombre (porque Total_{EF} ≤ 2)

Luego

$$E = UCP * CF$$

$$E = 68.08 * 20 \text{ horas hombre}$$

$$E = 1361.6 \text{ horas hombre}$$

5.5 Esfuerzo de todo el proyecto

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	341 horas-hombre
Diseño	20%	681 horas-hombre
Implementación	40%	1362 horas-hombre
Prueba	15%	511 horas-hombre
Sobrecarga	15%	511 horas-hombre
Total	100%	3405 horas-hombre

Suponiendo que la jornada laboral es de 8 horas, y un mes tiene como promedio 30 días; la cantidad de horas que puede trabajar una persona en 1 mes es 240 horas.

Si $E_T = 3405$ horas-hombre y por cada 240 horas se tiene 1 mes, eso daría un $E_T = 14.19$ mes-hombre.

Una persona puede terminar el proyecto en aproximadamente 14 meses y medio, **al equipo constar de 2 personas, el proyecto puede terminarse en aproximadamente 7 meses.**

5.6 Beneficios tangibles e intangibles

Con la implementación y puesta en marcha del sistema automatizado para la gestión de estrategias e indicadores de un Cuadro de Mando Integral se evidencian beneficios tanto tangibles como intangibles, que se manifiestan en el corto y largo plazo.

La ejecución de este trabajo tiene como objetivo controlar las actividades y beneficios de carácter intangibles que se realizan en las empresas cubanas. Ejemplo:

- Realizar una aplicación flexible, de interfaz agradable.
- Proliferar el uso del software para lograr mayor eficiencia y rapidez en el flujo de trabajo.
- Gestionar las estrategias y los indicadores de la empresa de forma eficiente.
- Beneficiar la seguridad e integridad de los datos que se obtengan.

En cuanto a los beneficios tangibles, a pesar de que se verán a largo plazo podemos destacar que se mejorará el rendimiento y la rentabilidad de las empresas cubanas, pues con la automatización de la gestión de estrategias e indicadores se ganará en rapidez de gestión y además se logrará identificar con mayor precisión los aspectos en los que las empresas deben sobresalir.

5.7 Análisis del costo

Todo proyecto informático lleva implícito costo monetario, recursos y esfuerzo, por eso, es necesario justificar la realización de cualquier producto de este tipo, partiendo de los beneficios económicos y sociales que se obtendrán a partir de su implantación y utilización. Con la puesta en uso de este sistema en las empresas cubanas aumenta la eficiencia de las actividades que se ejecuten en las mismas, ya que, la toma de decisiones se efectúa de una manera más rápida y certera.

Para la realización de este sistema se utilizaron tecnologías libres, por lo que no fue necesario incurrir en gastos por licencias de uso.

Costo del proyecto:

Se asume como salario promedio mensual \$125.00

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH = 2 hombres

CHM = CH * Salario Promedio

CHM = 2 * 125

CHM = 250.00 \$/mes

Costo = CHM * ET

Costo = 250 * 14.19

Costo = \$ 3547.50

Tiempo = ET / CH

Tiempo = 14.19 / 2

Tiempo = 7 meses

De los resultados obtenidos se interpreta que con 2 hombres trabajando en el proyecto el mismo se desarrolla en aproximadamente 7 meses y su costo total se estima que sea \$ 3547.50.

5.8 Conclusiones

En este capítulo se describió el estudio de factibilidad realizado al sistema propuesto para controlar el uso automatizado de la herramienta de Cuadro de Mando integral en las empresas cubanas, teniéndose en cuenta factores importantes como la duración del proyecto con respecto a la cantidad de personas que

trabajan en el mismo. Para ello se efectuó un análisis de los costos del proyecto, arribándose a la conclusión de que su precio es factible si se tiene en cuenta los beneficios tangibles e intangibles que este reporta.

CONCLUSIONES

Este trabajo de diploma facilitó la estandarización del Cuadro de Mando Integral y contribuyó a elevar la efectividad de la gestión empresarial cubana dándole cumplimiento al objetivo trazado: *automatizar el módulo gestión de estrategias e indicadores de las perspectivas del Cuadro de Mando Integral (CMI) para apoyar el servicio empresarial cubano*, ya que, los casos de uso satisficieron las necesidades funcionales del mismo, reafirmando así, la utilidad y validez de emplear las tecnologías informáticas para apoyar las labores que se desarrollan en cualquier tipo de esfera.

Se realizó una ardua labor con el tratamiento de errores para evitar la entrada de datos incoherentes, asimismo, con la seguridad, ayuda e interfaz del sistema implementado.

Se solucionó el problema de no poder generar informes rápidos, realizar análisis de la información de manera sistemática y tener en cada momento detalles sobre el control de las empresas logrando responder a instancias superiores con brevedad y veracidad; y contar con información requerida para la toma de decisiones a través de los indicadores y estrategias en el instante preciso.

RECOMENDACIONES

Se recomienda:

- Continuar el estudio del Cuadro de Mando Integral con el objetivo de añadir nuevas funcionalidades al sistema, entre estas, la implementación del cálculo de los indicadores por el método de BD.
- Implementar un módulo referido a la administración del sistema, así como realizar el mapa estratégico que debe llevar un CMI.
- Proponer, tras aprobar un desempeño exitoso, la utilización y generalización de este sistema en las diferentes empresas cubanas que lleva a cabo el CMI.

REFERENCIAS BIBLIOGRÁFICAS

1. Rivadeneira, R.M., *Trabajo en el VII Congreso Nacional de Control Interno en las entidades del estado*, 2001, p. 12.
2. *Resolución Económica del V Congreso del Partido Comunista de Cuba*. 1998.: La Habana, Cuba. p. 25.
3. http://en.wikipedia.org/wiki/Oracle_Corporation
4. http://www.netpecos.org/docs/mysql_postgres/x57.html#AEN71.
5. <http://www.linalco.com/apache.html>.
6. http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.asp.
7. Schuller, J., *Aprendiendo UML en 24 horas*. Prentice Hall ed. 2000.
8. <http://www.xtec.cat/~cdorado/cdora1/esp/disseny.htm>.
9. <http://es.wikipedia.org/wiki/Framework>.
10. <http://php.cujae.edu.cu/modules.php?op=modload&name=Reviews&file=index&req=showcontent&id=9>.
11. <http://nideaderedes.urlansoft.com/2007/02/27/code-igniter-framework-php/>
12. Microsoft, *Software License Agreement (BSD License)*.

BIBLIOGRAFÍA

Aplicaciones Web a la medida. Disponible en:

http://www.intellia.com.mx/esp/servicios/aplicaciones_web_a_la_medida.php > [Fecha de Consulta 2 - 2 - 2007]

Aplicación Web. Disponible en: http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web > [Fecha de Consulta 2 - 2 - 2007]

Características del Software Strategic. Disponible en:

http://www.estrategiaempresarial.com/sw_caracteristica.asp [Fecha de Consulta 8 - 1 - 2007]

Decide Soft. Disponible en: http://www.decidesoft.net/esp/info_c.htm [Fecha de Consulta 14 - 1 - 2007]

Delphos. Disponible en: <http://www.businesssoft.com.ni/delphos.htm> [Fecha de Consulta 9 - 1 - 2007]

Desarrollo Orientado a Objetos con UML. Disponible en:

<http://www.clikear.com/manuales/uml/introduccion.asp> > [Fecha de Consulta 4 - 2 - 2007]

Dialog Strategy: Características. Disponible en:

<http://www.cacitgroup.com/productos/faqs%20nuevo%20dialog.htm> [Fecha de Consulta 16 - 1 - 2007]

Dispelling the Myths. Disponible en: <http://dev.mysql.com/tech-resources/articles/dispelling-the-myths.html> > [Fecha de Consulta 15 - 1 - 2007]

Dreamweaver. Disponible en: <http://www.desarrolloweb.com/articulos/332.php> > [Fecha de Consulta 3 - 2 - 2007]

El CMI de en la gerencia SEPSA de Cienfuegos. Disponible en:

<http://www.fordes.co.cu/xtras/eventos/SemanasTec/memorias/Primera/lsemana.htm> > [Fecha de consulta 7 - 1 - 2007]

El Cuadro de Mando Integral como herramienta del control de Gestión. Disponible en:

http://www.wikilearning.com/el_cuadro_de_mando_integral_como_herramienta_del_control_de_gestion_estrategica-wkccp-12627-4.htm > [Fecha de consulta 18 - 2 - 2007]

El Cuadro de Mando Integral. Disponible en: <http://www.gestiopolis.com/canales3/ger/cmi.htm> > [Fecha de consulta 18 - 12 - 2006]

El Cuadro de Mando Integral en Cuba. Disponible en:

<http://www.monografias.com/trabajos31/cuadro-mando-integral/cuadro-mando-integral.shtml#cuadro> > [Fecha de consulta 19 - 12 - 2006]

El servidor Web Apache, Instalación. Disponible en:

<<http://es.tldp.org/Tutoriales/doc-servir-web-escuela/doc-servir-web-escuela-html/apache.html#AEN384>>

[Fecha de Consulta 29 – 1 - 2007]

EMS SQL Manager for MySQL. Disponible en:

<http://sqlmanager.net/en/products/mysql/manager/features> [Fecha de Consulta 4 – 2 - 2007]

Estilos y Patrones en la estrategia de Arquitectura de Microsoft. Disponible en:

<http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/style.asp> [Fecha de Consulta 13 – 1 - 2007]

Evaluando Zend Studio, 03.11.2003. Disponible en:

<<http://www.maestrosdelweb.com/editorial/zendstudio/>> [Fecha de Consulta 26 – 1 - 2007]

Gestión de Información. Disponible en:

<http://www.innovapyme.com/innovacion/links_recursos.asp> [Fecha de Consulta 7 – 1 - 2007]

JavaScript. Disponible en: <[http://64.233.167.104/search?q=cache:X2PM-](http://64.233.167.104/search?q=cache:X2PM-rl6bm0J:es.wikipedia.org/wiki/JavaScript+Java+Script&hl=es&ct=clnk&cd=1&gl=cu&lr=lang_es)

[rl6bm0J:es.wikipedia.org/wiki/JavaScript+Java+Script&hl=es&ct=clnk&cd=1&gl=cu&lr=lang_es](http://64.233.167.104/search?q=cache:X2PM-rl6bm0J:es.wikipedia.org/wiki/JavaScript+Java+Script&hl=es&ct=clnk&cd=1&gl=cu&lr=lang_es)> [Fecha de Consulta 6 – 2 - 2007]

Kaplan, Robert S. y Norton, David P., «Using the Balanced Score card as a Strategic Management System», *Harvard Business Review*, Boston, enero-febrero de 1996.

Las novedades de Dreamweaver 8, 08.08.2005. Disponible en:

<<http://www.maestrosdelweb.com/actualidad/2334/>> [Fecha de Consulta 17 - 1 - 2007]

Larman, Craig. *UML y Patrones*. La habana: Editorial Felix Varela, 2004

Manual de PHP, 26 de febrero del 2006. Disponible en:

<<http://www.php.net/manual/es/>> [Fecha de Consulta 4 – 2 - 2007]

Modelo Vista Controlador. Disponible en: <http://es.wikipedia.org/wiki/Modelo_Vista_Controlador> [Fecha de Consulta 13 – 1 - 2007]

MySQL. Disponible en:

<<http://es.wikipedia.org/wiki/MySQL>> [Fecha de Consulta 20 – 1 - 2007]

Perspectivas de Estrategia empresarial. Disponible en:

<<http://www.ee-iese.com>> [Fecha de Consulta 19 - 12 - 2006]

PostGreSQL vs. MySQL. Disponible en

http://www.netpecos.org/docs/mysql_postgres/index.html> [Fecha de Consulta 27 – 2 - 2007]

Proceso Unificado de Racional. Disponible en:

http://es.wikipedia.org/wiki/Rational_Unified_Process [Fecha de Consulta 9 - 1 - 2007]

Servidor Web Apache. Disponible en: <http://www.linalco.com/apache.html> > [Fecha de Consulta 1- 2 - 2007]

Sistema de gestión de Base de Datos. Disponible en:

http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos [Fecha de Consulta 7 – 2 - 2007]

Ventajas e inconvenientes de las aplicaciones Web. Disponible en:

<http://www.avidos.net/blogold/aplicaciones-web/> > [Fecha de Consulta 2 - 2 - 2007]

YUI FAQ. Disponible en: <http://developer.yahoo.com/yui/articles/faq/>> [Fecha de Consulta 3 – 2 - 2007]

ANEXOS

Anexo 1. Descripción textual de los casos de uso del negocio.

Nombre del Caso de Uso	Solicitar estrategia
Actores	Directivo (inicia)
Propósito	Permitir solicitar la estrategia según las necesidades del directivo.
Resumen	El caso de uso se inicia cuando el directivo solicita la estrategia de la organización, luego se llevan a cabo un conjunto de actividades y se toman decisiones hasta dar como respuesta una estrategia, y de esta forma finaliza el caso de uso.
Curso Normal de los eventos	
Acciones del Actor	Respuestas del proceso del negocio
1- El directivo solicita una estrategia.	
	2- El jefe de área recibe la solicitud de la estrategia.
	3- El jefe de área define las estrategias de la organización.
	4- Si se encuentra la estrategia, el jefe de área la entrega.
5- El directivo recibe la estrategia.	
Curso Alternativo de los eventos	
Acción 4.	Si no se encuentra la estrategia, el jefe de área la crea y se ejecuta la acción 4.
Prioridad	Crítico

Mejoras	
Otros	

Tabla 1: Caso de uso "Solicitar estrategia".

Nombre del Caso de Uso	Solicitar cálculo del indicador	
Actores	Directivo (inicia)	
Propósito	Permitir solicitar cálculo del indicador según las necesidades del directivo.	
Resumen	El caso de uso se inicia cuando el directivo solicita el valor de un indicador, luego se llevan a cabo un conjunto de actividades y se toman decisiones hasta dar como resultado dicho valor, y de esta forma finaliza el caso de uso.	
Curso Normal de los eventos		
Acciones del Actor	Respuestas del proceso del negocio	
1- El directivo solicita calcular un indicador.		
	2- El jefe de área recibe la solicitud del indicador.	
	3- El jefe de área identifica el indicador.	
	4- si se encuentra el indicador, el jefe de área lo calcula.	
	5- El jefe de área entrega el indicador calculado.	
6- El directivo recibe el resultado del indicador.		
Curso Alternativo de los eventos		
Acción 4.	Si no se encuentra el indicador, el jefe de área lo crea y ejecuta la acción 4.	
Prioridad	Crítico	
Mejoras		
Otros		

Tabla 2: Caso de uso "Solicitar cálculo del indicador".

Anexo 2. Diagrama de actividades.

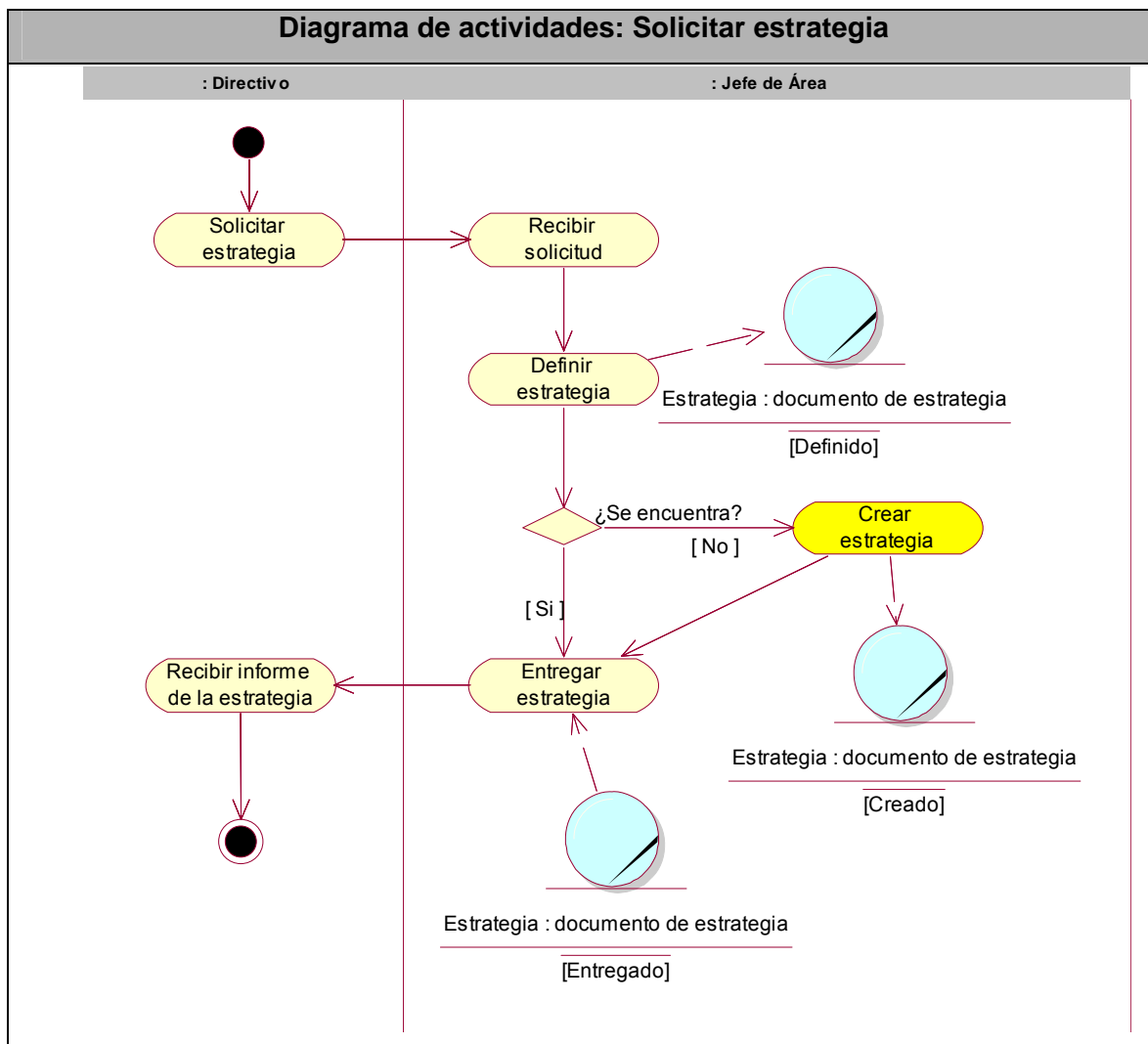


Figura 1: Diagrama de actividades “Solicitar estrategia”

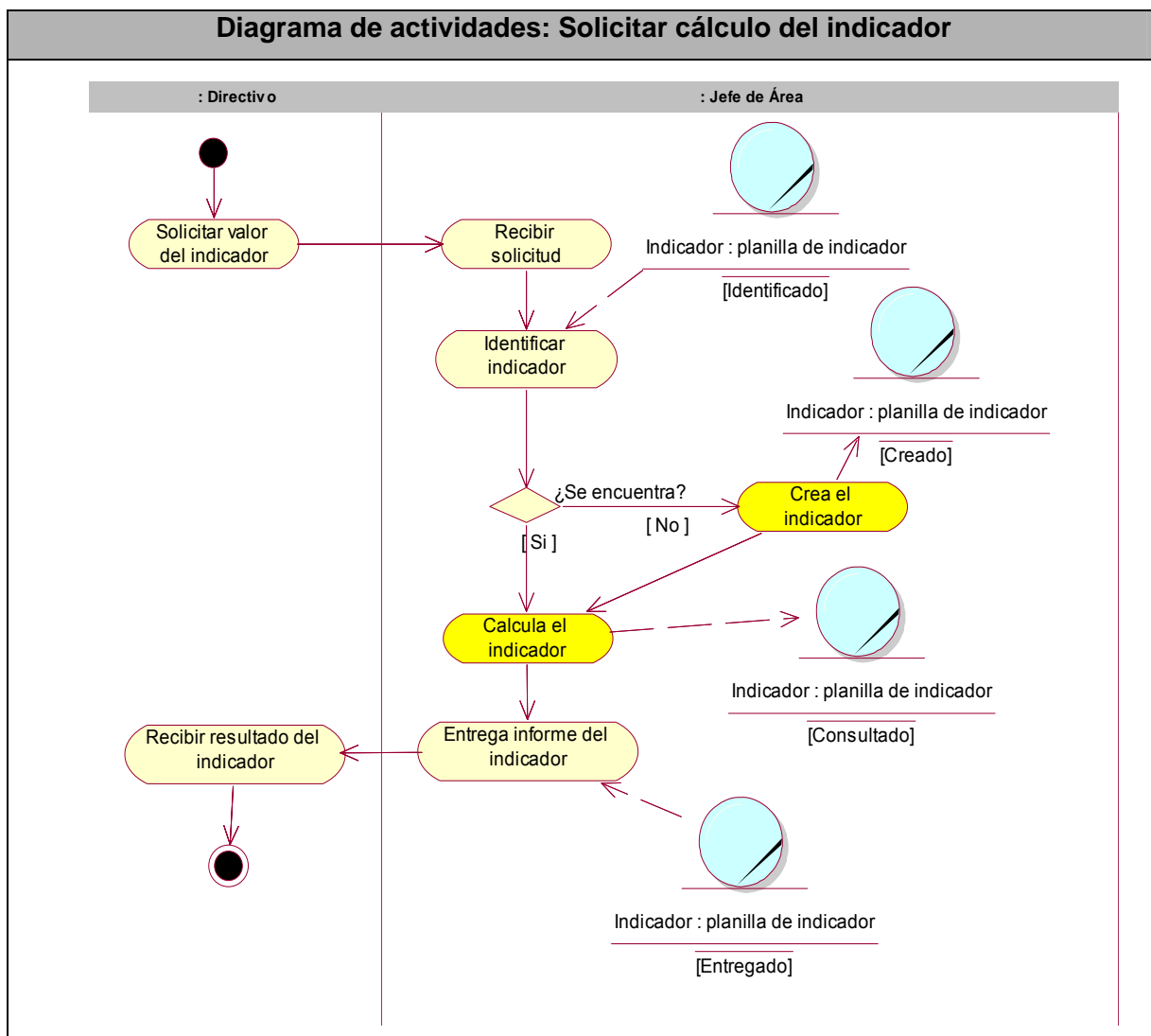


Figura 2: Diagrama de actividades "Solicitar cálculo del indicador"

Anexo 3. Diagrama de clases del Modelo de Objetos.

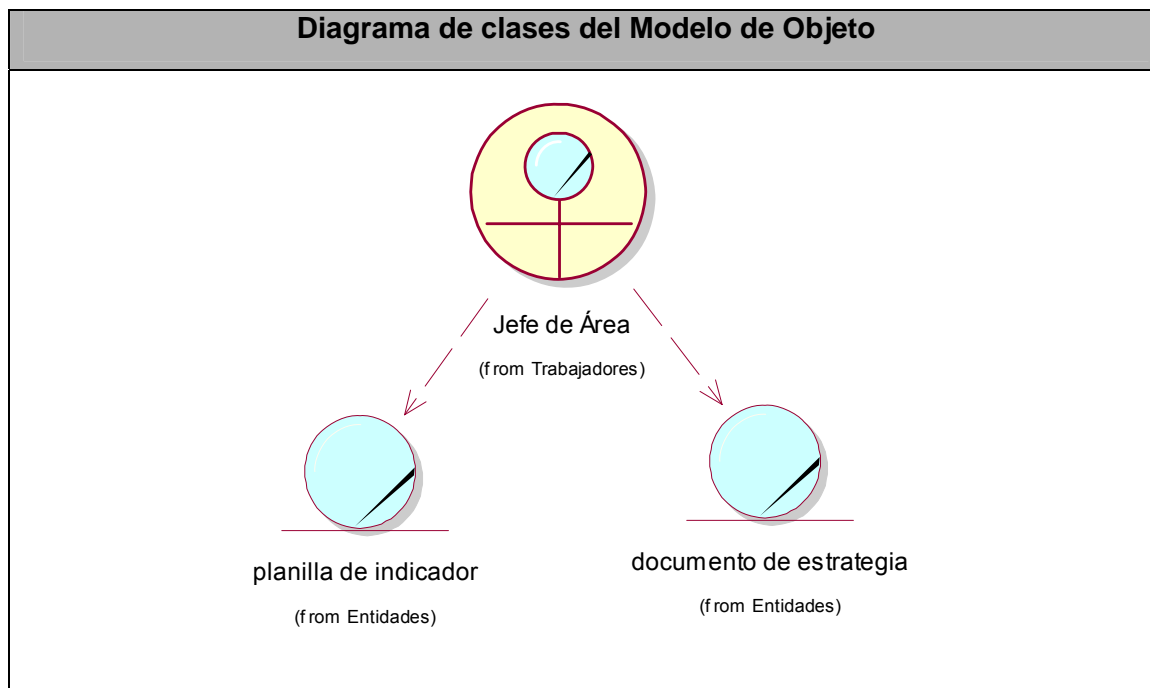


Figura 3: Diagrama de clases del modelo de objeto

Anexo 4. Descripción textual de los casos de uso del sistema.

2.5.3.1 Caso de uso “Gestionar Estrategia”

Caso de Uso	Gestionar Estrategia
Actores	Directivo (inicia)
Propósito	Permitir al directivo gestionar (crear, editar o eliminar) las estrategias.
Resumen	El caso de uso se inicia cuando el directivo desea gestionar una estrategia, el sistema muestra las opciones que puede realizar sobre la misma, el directivo selecciona la opción que desea: insertar, editar o eliminar, luego el sistema lleva a cabo la acción seleccionada y termina el caso de uso del sistema.
Referencias	R.1.1, R.1.2, R.1.3.
Precondiciones	Para las siguientes acciones: Editar: Debe existir alguna estrategia registrada en el sistema. Eliminar: Debe existir alguna estrategia registrada en el sistema.
Poscondiciones	Para las siguientes acciones: Crear: Se crea una nueva estrategia para que luego sea desarrollada. Editar: Se edita una estrategia para que luego sea desarrollada. Eliminar: Se elimina totalmente una estrategia.
Prototipo	



Curso Normal de los eventos

Acciones del Actor	Respuestas del sistema
1. El directivo selecciona la opción gestionar estrategia.	1.1 El sistema muestra un listado de estrategias con las opciones de crear, editar y eliminar una estrategia.
Sección 1: Crear estrategia	
1. El directivo desea crear una estrategia.	1.1 El sistema muestra una interfaz para entrar los datos referentes a la estrategia.
2. El directivo teclea los datos de la estrategia que desea crear y activa esta opción.	2.1 El sistema valida los datos y crea la estrategia. El Caso de Uso comienza nuevamente.
Sección 2: Editar estrategia	
1. El directivo desea editar una	1.1 El sistema muestra el listado de estrategias que

estrategia.	existen.
2. El directivo selecciona la estrategia que desea editar y se activa esta opción.	2.1 El sistema le muestra una interfaz con los datos de la estrategia para editar.
3. El directivo realiza los cambios necesarios a la estrategia seleccionada.	3.1 El sistema valida los datos y edita la estrategia. El Caso de Uso comienza nuevamente.
Curso alternativo de los eventos	
Acción 1.1	Si no existe ninguna estrategia en la base de dato, no se podrá realizar la acción 2 en adelante y el caso de uso concluye automáticamente.
Sección 3: Eliminar estrategia	
1. El directivo desea eliminar una estrategia.	1.1 El sistema muestra el listado de estrategias que existen.
2 El directivo selecciona la estrategia que desea eliminar y activa esta opción.	2.1 El sistema muestra un mensaje de confirmación previendo su total eliminación.
	2.2 El sistema elimina la estrategia seleccionada. El Caso de Uso comienza nuevamente.
Curso alternativo de los eventos	
Acción 1.1	Si no existe ninguna estrategia en la base de dato, no se podrá realizar la acción 2 en adelante y el caso de uso concluye automáticamente.
Prioridad	Crítico

Tabla 3: Caso de uso "Gestionar Estrategia".

2.5.3.2 Caso de uso “Gestionar Indicadores”

Caso de Uso	Gestionar Indicadores
Actores	Directivo (inicia)
Propósito	Permitir al directivo gestionar (crear, editar y eliminar) los indicadores.
Resumen	El caso de uso se inicia cuando el directivo desea gestionar los indicadores, el sistema muestra las opciones que puede realizar sobre la misma, el directivo selecciona la opción que desea: crear, editar o eliminar, luego el sistema lleva a cabo la acción seleccionada y termina el caso de uso del sistema.
Referencias	R.2.1, R.2.2, R.2.3.
Precondiciones	Para las siguientes acciones: Editar: Debe existir algún indicador registrado en el sistema. Eliminar: Debe existir algún indicador registrado en el sistema.
Poscondiciones	Para las siguientes acciones: Crear: Se crea un indicador para medir luego los objetivos. Editar: Se edita un indicador para medir luego los objetivos. Eliminar: Se elimina totalmente un indicador.
Prototipo	

The screenshot shows a web application window titled 'Indicadores'. It has a navigation bar with buttons for 'lista', 'Indicador', 'Unid. Medida', 'Fórmula', and 'Calcular'. The main form contains the following fields:

- Nombre:
- Unidad de Medida:
- Perspectiva:
- Sistema de Alerta:
- Período:
- Descripción:
- Fórmula:

A 'Crear' button is located at the bottom right of the form.

Curso Normal de los eventos

Acciones del Actor	Respuestas del sistema
1. El directivo selecciona la opción gestionar indicadores.	1.1 El sistema muestra un listado de indicadores con las opciones de crear, editar y eliminar un indicador.
Sección 1: Crear indicadores	
1. El directivo desea crear un indicador.	1.1 El sistema muestra una interfaz para entrar los datos referentes al indicador.
2. El directivo teclea los datos del indicador que desea crear, incluyendo su unidad de medida, fórmula y método de cálculo de ese indicador. Activa	2.1 El sistema valida los datos y crea el indicador. El Caso de Uso comienza nuevamente.

esta opción.	
Sección 2: Editar indicadores	
. El directivo desea editar un indicador.	1.1 El sistema muestra el listado de indicadores que existen.
2. El directivo selecciona el indicador que desea editar y se activa esta opción.	2.1 El sistema le muestra una interfaz con los datos del indicador para editar.
3. El directivo realiza los cambios necesarios al indicador.	3.1 El sistema valida los datos y edita el indicador. El Caso de Uso comienza nuevamente.
Curso alternativo de los eventos	
Acción 1.1	Si no existe ningún indicador en la base de dato, no se podrá realizar la acción 2 en adelante y el caso de uso concluye automáticamente.
Sección 3: Eliminar indicadores	
1. El directivo desea eliminar un indicador.	1.1 El sistema muestra el listado de indicadores que existen.
2 El directivo selecciona el indicador que desea eliminar y activa esta opción.	2.1 El sistema muestra un mensaje de confirmación previendo su total eliminación.
	2.2 El sistema elimina el indicador seleccionado. El Caso de Uso comienza nuevamente.
Curso alternativo de los eventos	
Acción 1.1	Si no existe ningún indicador en la base de dato, no se podrá realizar la acción 2 en adelante y el caso de uso concluye automáticamente.
Prioridad	Crítico

Tabla 4: Caso de uso "Gestionar Indicadores".

2.5.3.3 Caso de uso "Calcular indicador"

Caso de Uso	Calcular Indicador
Actores	Directivo (inicia)
Propósito	Permitir al directivo calcular el indicador deseado.
Resumen	El caso de uso se inicia cuando el directivo desea calcular un indicador, el sistema muestra el listado de los indicadores y este selecciona la opción de calcular, luego el sistema lleva a cabo la acción seleccionada y termina el caso de uso del sistema.
Referencias	R.5.1, R.5.2.
Precondiciones	Para la siguiente acción: Calcular: Debe existir alguna fórmula registrada en el sistema.
Poscondiciones	Para la siguiente acción: Calcular: Se obtiene un resultado del indicador.
Prototipo	

Indicadores

lista | **Indicador** | Unid. Medida | Fórmula | **Calcular**

Variables:
 AC: PC:
 Calcular

Resultado
 AC:

Indicadores	Acciones
Indicador general	
Indicador economico	
Ind	
I	
Yu	

Curso Normal de los eventos

Acciones del Actor	Respuestas del sistema
1. El directivo selecciona la opción Calcular.	1.1 El sistema muestra un listado de indicadores con la opción de calcular un indicador.
Sección 1: Calcular indicadores por BD	
1. El directivo desea calcular un indicador.	1.1 El sistema muestra el listado de indicadores que existen.
2. El directivo selecciona el indicador que desea calcular y activa esta opción.	2.1 El sistema muestra una interfaz con la opción de conectarse a la BD.
3. El directivo selecciona la BD por la que desea conectarse.	3.1 El sistema muestra los campos de la BD.

4. El directivo selecciona los campos con el que trabajará el indicador.	
5. El directivo configura la planificación del indicador y activa la opción de calcular.	5.1 El sistema valida los datos y calcula el indicador. El Caso de Uso comienza nuevamente.
Curso alternativo de los eventos	
Acción 3	Si el directivo desea cambiar de BD, mediante la opción Editar, puede realizarlo.
Acción 4	Si el directivo desea modificar los campos con el que trabajará el indicador, puede realizarlo.
Acción 5	Si el directivo desea reconfigurar la planificación del indicador, puede realizarlo.
Sección 2: Calcular indicadores manual	
1. El directivo desea calcular un indicador.	1.1 El sistema muestra el listado de indicadores que existen.
2. El directivo selecciona el indicador que desea calcular y activa esta opción.	2.1 El sistema valida los datos y calcula el indicador.
Prioridad	Crítico

Tabla 5: Caso de uso "Calcular indicador".

2.5.3.3 Caso de uso "Gestionar Fórmula"

Caso de Uso	Gestionar Fórmula
Actores	Directivo (inicia)
Propósito	Permitir al directivo gestionar (insertar, editar y eliminar) las fórmulas para calcular los indicadores.
Resumen	El caso de uso se inicia cuando el directivo desea gestionar una fórmula, el sistema muestra las opciones

	que puede realizar sobre la misma, el directivo selecciona la opción que desea: insertar, editar y eliminar, luego el sistema lleva a cabo la acción seleccionada y termina el caso de uso del sistema.
Referencias	R.3.1, R.3.2, R.3.3.
Precondiciones	Para las siguientes acciones: Editar: Debe existir alguna fórmula registrada en el sistema. Eliminar: Debe existir alguna fórmula registrada en el sistema.
Poscondiciones	Para las siguientes acciones: Insertar: Se crea una fórmula. Editar: Se edita una fórmula. Eliminar: Se elimina totalmente una fórmula.
Prototipo	

Indicadores

lista Indicador Unid. Medida **Fórmula** Calcular

Nombre:

Fórmula

Nombre de las variables:

AE:

AA:

IB:

Curso Normal de los eventos

Acciones del Actor	Respuestas del sistema
1. El directivo selecciona la opción gestionar fórmula.	1.1 El sistema muestra un listado de fórmulas con las opciones de insertar, editar y eliminar una fórmula.
Sección 1: Insertar fórmula	
1. El directivo desea insertar una fórmula.	1.1 El sistema muestra una interfaz para entrar los datos referentes a la fórmula.
2. El directivo teclea los datos de la fórmula que desea insertar. Activa esta opción.	2.1 El sistema valida los datos e inserta la fórmula. El Caso de Uso comienza nuevamente.
Curso alternativo de los eventos	

Sección 2: Editar fórmula	
1. El directivo desea editar una fórmula.	1.1 El sistema muestra el listado de las fórmulas que existen.
2. El directivo selecciona la fórmula que desea editar y se activa esta opción.	2.1 El sistema le muestra una interfaz con los datos de la fórmula para editar.
3. El directivo realiza los cambios necesarios a la fórmula	3.1 El sistema valida los datos y edita la fórmula. El Caso de Uso comienza nuevamente.
Curso alternativo de los eventos	
Acción 1.1	Si no existe ninguna fórmula en la base de dato, no se podrá realizar la acción 2 en adelante y el caso de uso concluye automáticamente.
Sección 3: Eliminar fórmula	
1. El directivo desea eliminar una fórmula.	1.1 El sistema muestra el listado de las fórmulas que existen.
2 El directivo selecciona la fórmula que desea eliminar y activa esta opción.	2.1 El sistema elimina la fórmula seleccionada. El Caso de Uso comienza nuevamente.
Curso alternativo de los eventos	
Acción 1.1	Si no existe ninguna fórmula en la base de dato, no se podrá realizar la acción 2 en adelante y el caso de uso concluye automáticamente.
Prioridad	Crítico

Tabla 6: Caso de uso "Gestionar Fórmula".

2.5.3.4 Caso de uso “Gestionar Unidad de Medida”

Caso de Uso	Gestionar Unidad de Medida
Actores	Directivo (inicia)
Propósito	Permitir al directivo gestionar (insertar, editar o eliminar) las unidades de medidas
Resumen	El caso de uso se inicia cuando el directivo desea gestionar una unidad de medida, el sistema muestra las opciones que puede realizar sobre la misma, el directivo selecciona la opción que desea: insertar, editar o eliminar, luego el sistema lleva a cabo la acción seleccionada y termina el caso de uso del sistema.
Referencias	R.4.1, R.4.2, R.4.3.
Precondiciones	Para las siguientes acciones: Editar: Debe existir alguna unidad de medida registrada en el sistema. Eliminar: Debe existir alguna unidad de medida registrada en el sistema.
Poscondiciones	Para las siguientes acciones: Insertar: Se crea una nueva unidad de medida. Editar: Se edita una unidad de medida. Eliminar: Se elimina totalmente una unidad de medida.
Prototipo	

Indicadores

lista Indicador **Unid. Medida** Fórmula

Nombre:

Símbolo:

Unidades de Medida	Símbolos	Acciones
Dollar	US \$	
Horas	Hs	
S	S	

Curso Normal de los eventos

Acciones del Actor	Respuestas del sistema
1. El directivo selecciona la opción gestionar unidad de medida.	1.1 El sistema muestra un listado de unidades de medidas con las opciones de crear, editar y eliminar una unidad de medida.
Sección 1: Insertar unidad de medida	
1. El directivo desea insertar una unidad de medida.	1.1 El sistema muestra una interfaz para entrar los datos referentes a la unidad de medida.
2. El directivo teclea los datos de la unidad de medida que desea crear y activa esta opción.	2.1 El sistema valida los datos e inserta la unidad de medida. El Caso de Uso comienza nuevamente.
Sección 2: Editar unidad de medida	

1. El directivo desea editar una unidad de medida.	1.1 El sistema muestra el listado de unidades de medida que existen.
2. El directivo selecciona la unidad de medida que desea editar y se activa esta opción.	2.1 El sistema le muestra una interfaz con los datos de la unidad de medida para editar.
3. El directivo realiza los cambios necesarios a la unidad de medida seleccionada.	3.1 El sistema valida los datos y edita la unidad de medida. El Caso de Uso comienza nuevamente.
Curso alternativo de los eventos	
Acción 1.1	Si no existe ninguna unidad de medida en la base de dato, no se podrá realizar la acción 2 en adelante y el caso de uso concluye automáticamente.
Sección 3: Eliminar unidad de medida	
1. El directivo desea eliminar una unidad de medida.	1.1 El sistema muestra el listado de unidades de medida que existen.
2 El directivo selecciona la unidad de medida que desea eliminar y activa esta opción.	2.1 El sistema muestra un mensaje de confirmación previendo su total eliminación.
	2.2 El sistema elimina la unidad de medida seleccionada. El Caso de Uso comienza nuevamente.
Curso alternativo de los eventos	
Acción 1.1	Si no existe ninguna unidad de medida en la base de dato, no se podrá realizar la acción 2 en adelante y el caso de uso concluye automáticamente.
Prioridad	Crítico

Tabla 7: Caso de uso "Gestionar Unidad de Medida".

Anexo 5. Diagrama de clases del Modelo de Objetos.

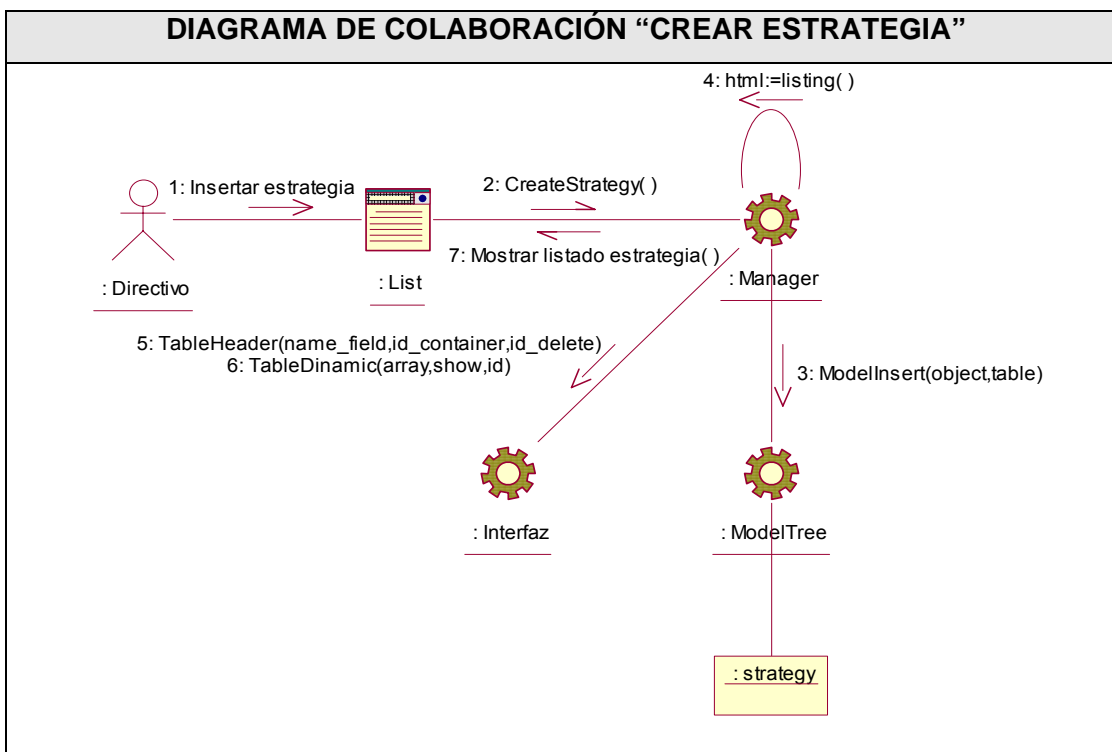


Figura 4: Diagrama de colaboración "Crear Estrategia"

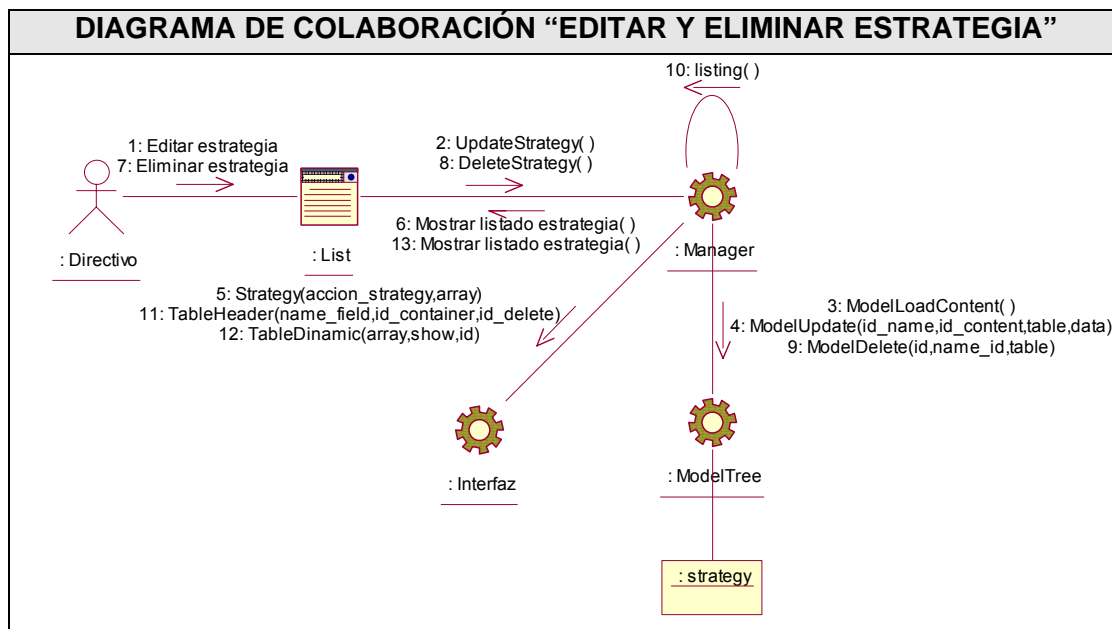


Figura 5: Diagrama de colaboración "Editar y Eliminar Estrategia"

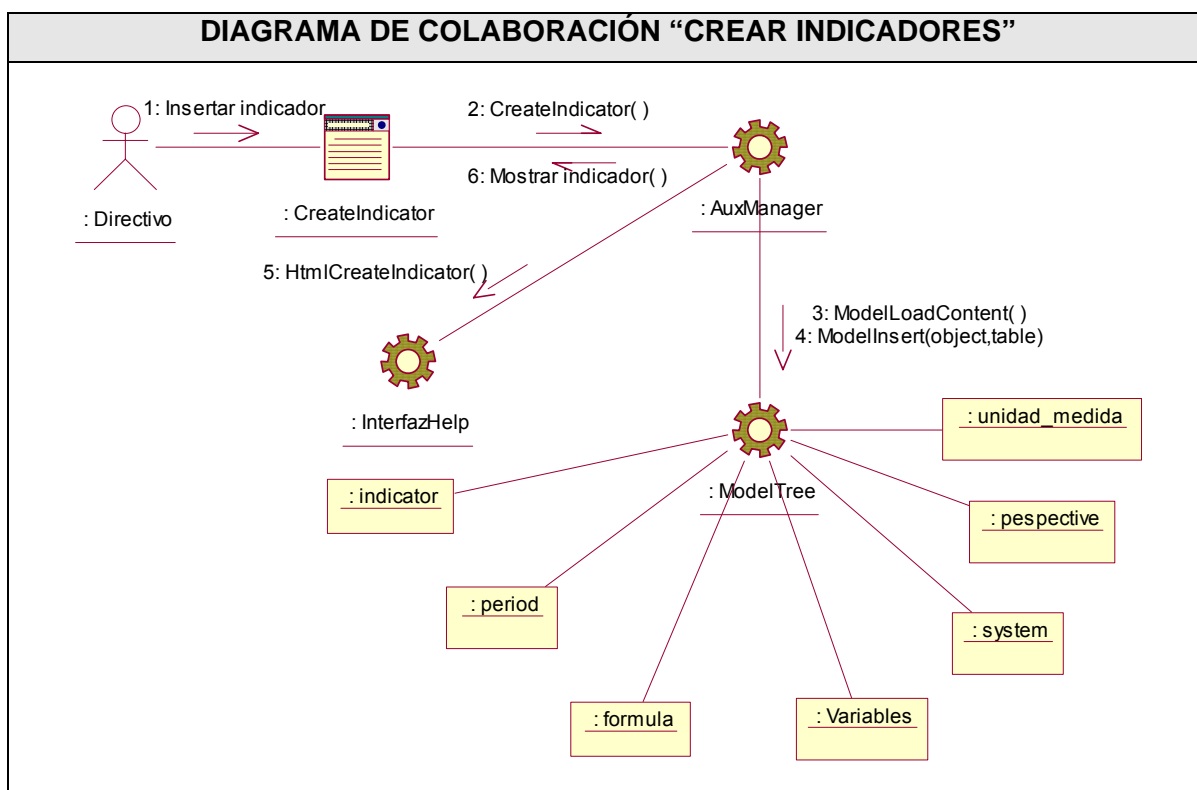


Figura 6: Diagrama de colaboración “Crear Indicadores”

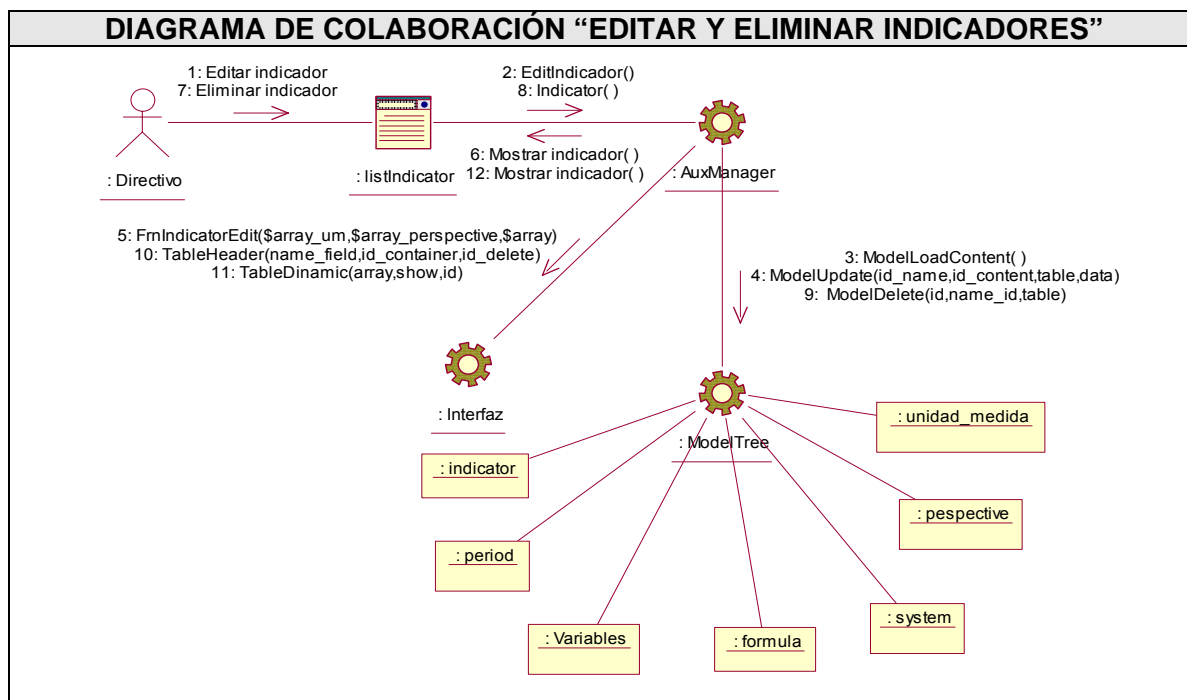


Figura 7: Diagrama de colaboración "Editar y Eliminar Indicadores"

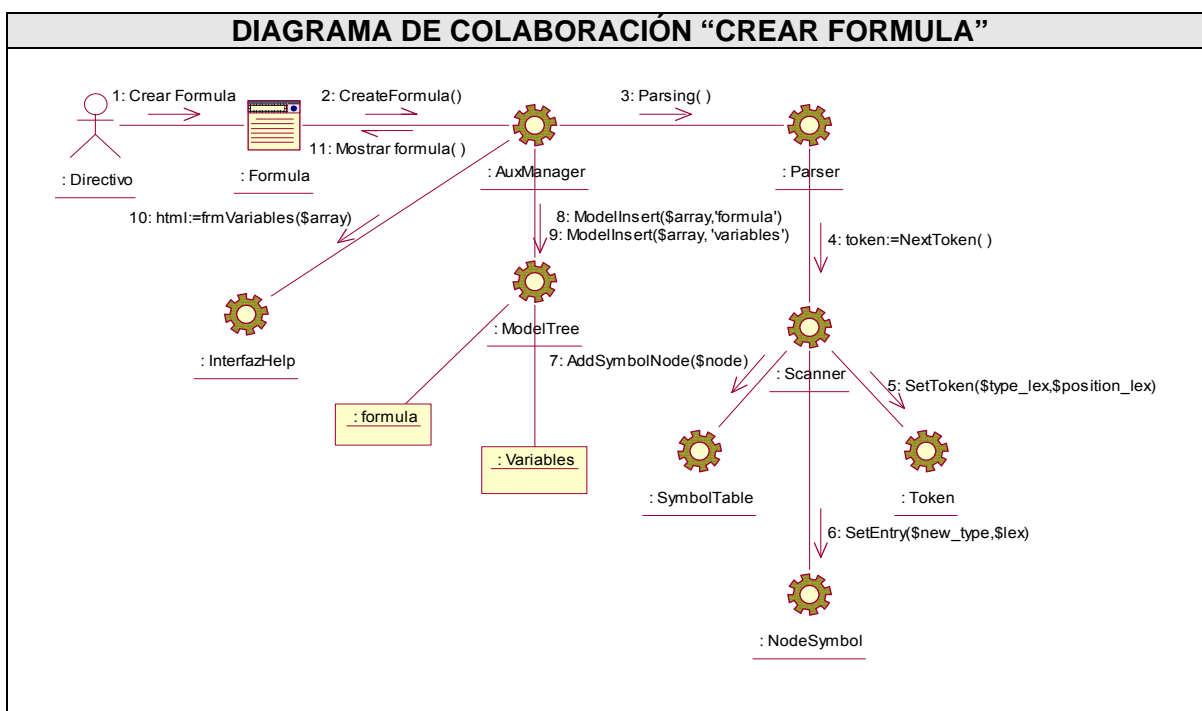


Figura 8: Diagrama de colaboración "Crear Fórmula"

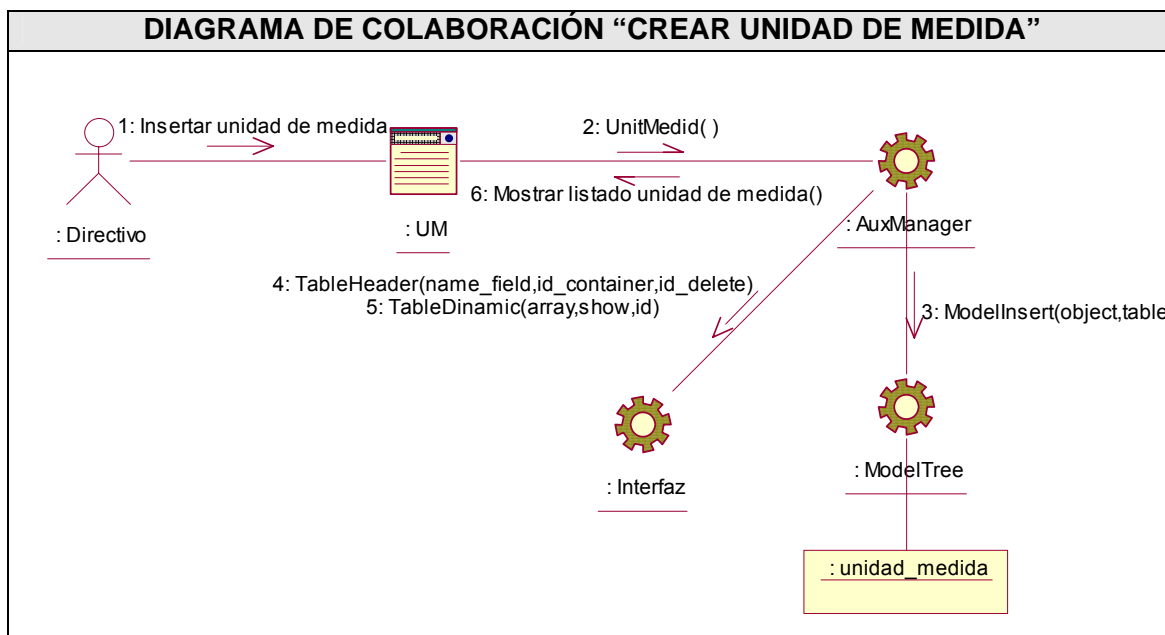


Figura 9: Diagrama de colaboración “Crear Unidad de Medida”

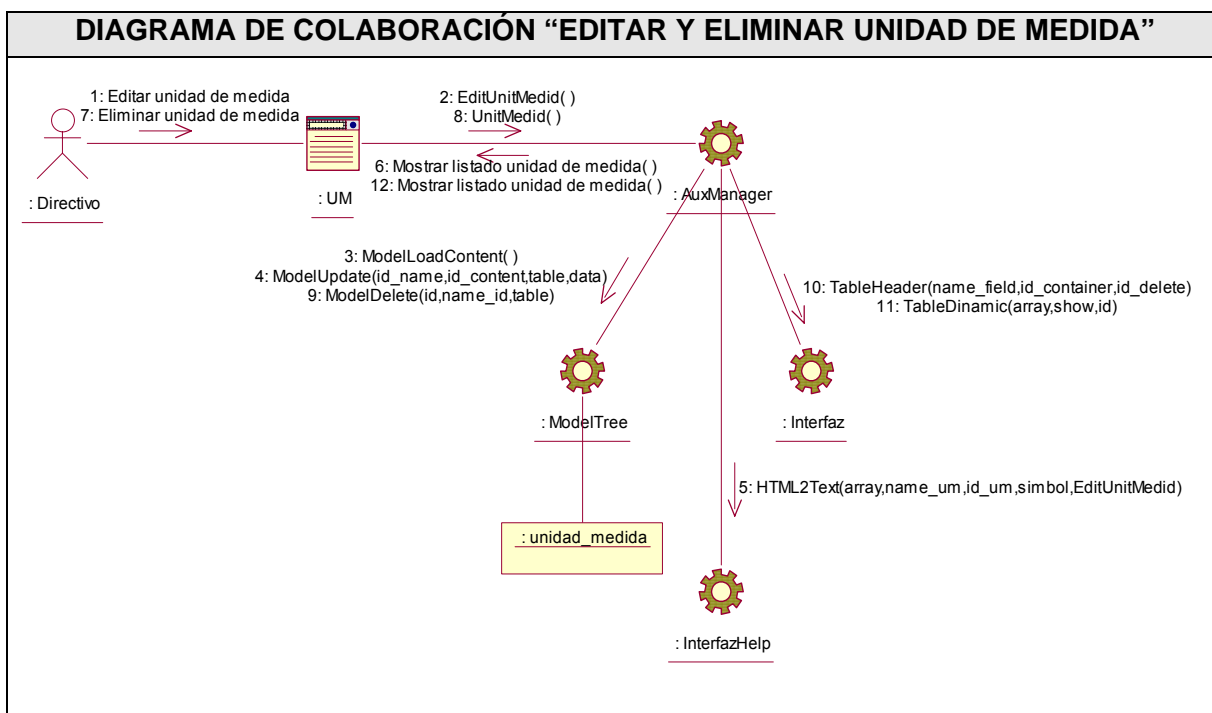


Figura 10: Diagrama de colaboración “Editar y Eliminar Unidad de Medida”

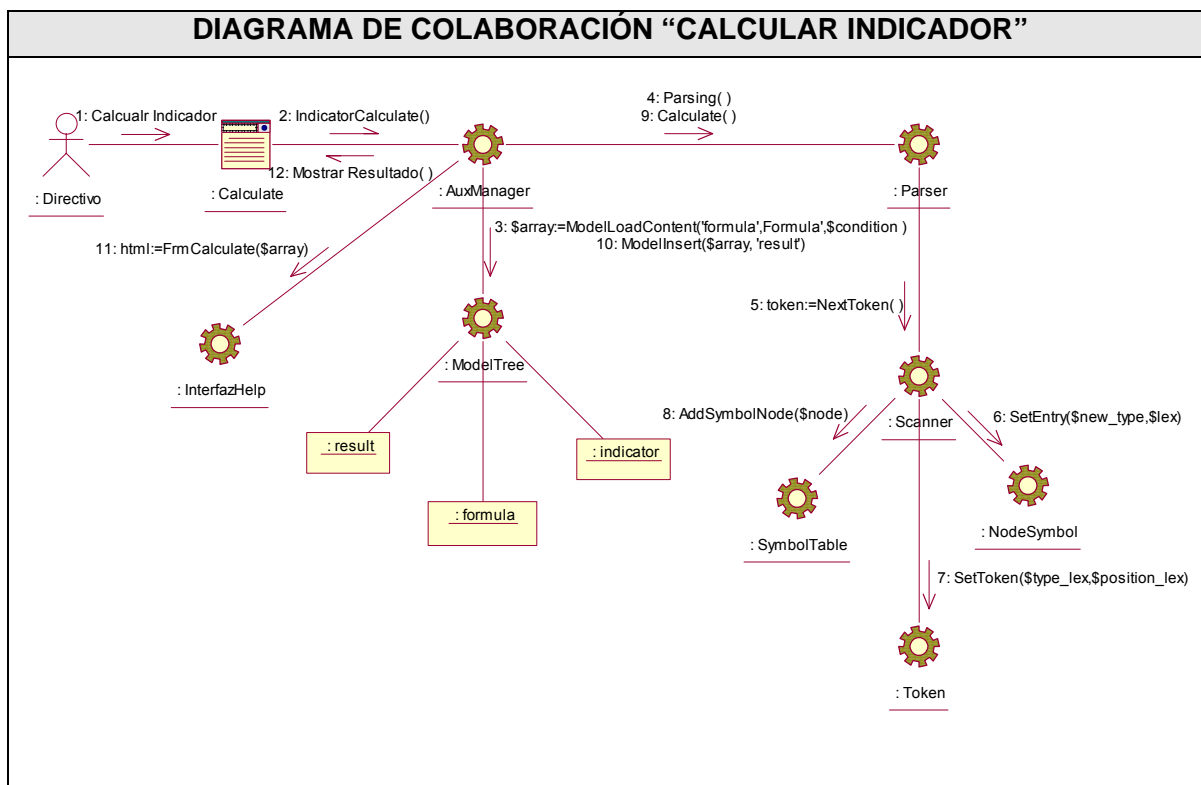


Figura 11: Diagrama de colaboración “Calcular Indicador”

GLOSARIO DE TÉRMINOS

AJAX: Asynchronous JavaScript And XML.

APACHE: Servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1.

BD: Base de Datos.

CMI: Cuadro de Mando Integral.

COCOMO: Modelo para la estimación de costos de productos informáticos.

CUN: Caso de Uso del Negocio.

CUS: Caso de Uso del Sistema.

Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.

HTML: HyperText Markup Language. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

HTTP: HyperText Transfer Protocol. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.

MVC: Modelo Vista Controlador.

PHP: HyperText Transfer Protocol. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.

PostgreSQL: Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.

RUP: Rational Unified Process (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.

SGBD: Sistema de Gestión de Bases de Datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

UML: Unified Modeling Language. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.