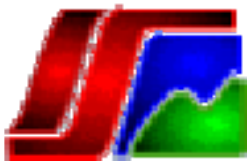


Instituto Superior Minero Metalúrgico
"Antonio Núñez Jiménez"
Facultad de Metalurgia-Electromecánica

SISTEMA AUTOMATIZADO PARA EL CONTROL DE LOS INDICADORES DE GESTIÓN DE UN CUADRO DE MANDO INTEGRAL

Trabajo para optar por el Título de Ingeniero Informático

Autor: Ariel Ricardo Cuenca Muguercia.

Tutor: Lic. Leonardo Torno Hidalgo.

Ing. Roy Rodriguez Noa.

Cotutor: MSc. Adis Nuvia Neyra Muguercia.

Consultante: Ing. Eugenio Reyes Chávez.

Moa, julio de 2010

Pensamiento

"… No se puede dirigir si no se sabe analizar y no se puede analizar si no hay datos verídicos, si no hay todo un sistema de recolección de datos confiables, si no hay toda una preparación de un sistema estadístico con hombres habituados a recoger el dato y transformar en números. De manera que es una tarea especial…"

Che.

Agradecimientos

A mis tutores

Por su incondicional apoyo en todo momento, y por su paciencia y comprensión.

A mis compañeros de aula

Por estar siempre a mi lado cuando los necesite.

A mis profesores

Por sus enseñanzas en todos estos años.

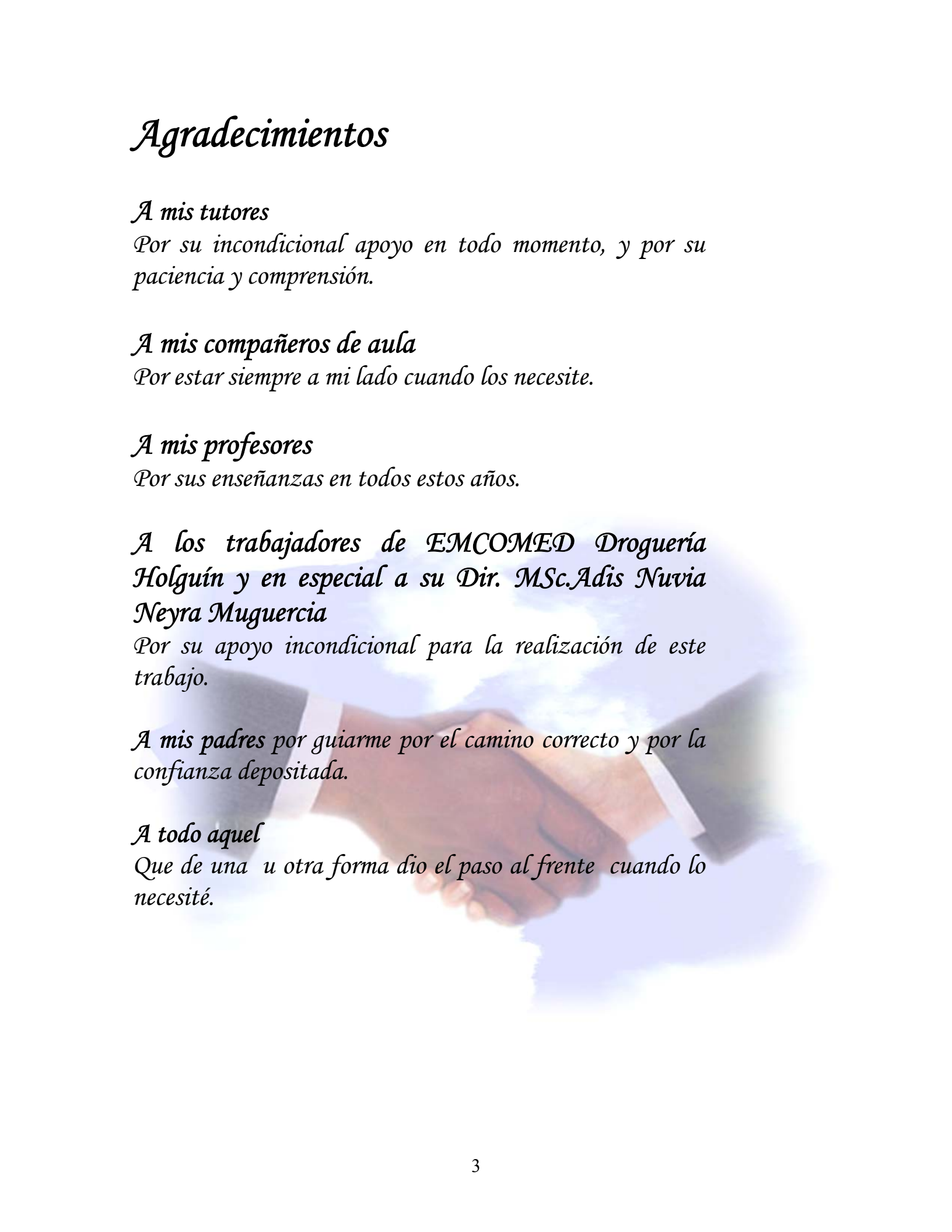
A los trabajadores de EMCOMED Droguería Holguín y en especial a su Dir. MSc.Adis Nuvia Neyra Muguercia

Por su apoyo incondicional para la realización de este trabajo.

A mis padres *por guiarme por el camino correcto y por la confianza depositada.*

A todo aquel

Que de una u otra forma dio el paso al frente cuando lo necesité.



Dedicatoria

A toda mi familia

Por haberme encaminado en la vida y mostrarme el camino correcto, por apoyarme en los momentos difíciles de mi vida como estudiante y por brindarme todo lo que he necesitado.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del trabajo de diploma titulado:
SISTEMA PARA EL CONTROL DE LOS INDICADORES DE GESTIÓN DE UN CUADRO DE MANDO INTEGRAL y que el mismo pertenece a la Empresa EMCOMED Droguería Holguín y a la Facultad de Metalurgia-Electromecánica del ISMM "Antonio Núñez Jiménez" de Holguín, para que se haga el uso que se estime pertinente con este trabajo.

Para que así conste firman la presente a los ___ días del mes de _____ del año _____.

_____

Autor: Ariel Ricardo Cuenca Muguercia.

_____

Tutor:

Ing. Roy Rodríguez Noa.

Prof. del Dpto. Informática.

_____

Tutor:

Lic. Leonardo Torno Hidalgo.

Esp. C. Ciencias Informática EMCOMED.

OPINIÓN DEL TUTOR

Título del trabajo de diploma: SISTEMA PARA EL CONTROL DE LOS INDICADORES DE GESTIÓN DE UN CUADRO DE MANDO INTEGRAL

Tutores del trabajo de diploma: Ing. Roy Rodríguez Noa y Lic. Leonardo Torno Hidalgo.

Consideramos que el estudiante se encuentra listo para ejercer como ingeniero informático, y se le propone la calificación de __5__ puntos.
Para que así conste firma la presente a los ___ días del mes de _____ del año _____.

_____

Tutor:
Ing. Roy Rodríguez Noa.
Prof. del Dpto. Informática.

_____

Tutor:
Lic. Leonardo Torno Hidalgo.
Esp. C. Ciencia Informática EMCOMED.

Resumen

Las Nuevas Tecnologías Informáticas posibilitan, de manera más efectiva, la atención a las diferentes ramas de los procesos empresariales pues para la aplicación de cualquier estrategia es necesario o aconsejable, en muchas ocasiones, debido a la necesidad de informatizar la información, consecuencia del desarrollo global de la automatización de las operaciones, apoyarse en algún software que ayude a recoger los datos históricos de los resultados obtenidos y permita optimizar la fluidez de estos.

En la Empresa EMCOMED Droguería Holguín anteriormente se realizaba el control de los indicadores de gestión por áreas, de forma individual por diversas vías (correo electrónico, archivos de textos, hojas de cálculo, etc.) y representados por distintos esquemas (gráficas, diagramas). Esto provoca demora en la información y que se cometan errores no intencionados que convierte en poco fiable la integridad de la información.

Por lo antes planteado se implementó una aplicación Web flexible, de administración sencilla e interacción agradable, que es capaz de ejecutar de forma eficiente el control de los indicadores de gestión del Cuadros de Mando Integral, contribuyendo a la mejora de la interpretación y evaluación de los mismos. Se utilizó la metodología **ICONIX** para el modelado del análisis y diseño del sistema, como gestor de base de datos **MySql** y lenguaje de programación **PHP**.

Este documento esta estructurado en cuatro capítulos. El capitulo1 contiene toda la Fundamentación teórica de los métodos de investigación, metodologías, tecnologías y herramientas utilizadas para el desarrollo del trabajo. El Capítulo 2, “Descripción de la Solución Propuesta”, contiene los aspectos de mayor peso del desarrollo del sitio Web propuesto, como son la correspondencia con la metodología *ICONIX*, el modelo de casos de uso y los mecanismos de persistencia que utiliza El Cuadro de Mando Integral. Además se hace un estudio de sostenibilidad según su impacto socio-humanista, administrativo, tecnológico y ambiental, así como su evolución. El Capítulo 3, “Elaboración y Evaluación de la Solución Propuesta”, se estructuran y refinan los requerimientos capturados a partir del flujo de trabajo de análisis y diseño, que da lugar a la implementación; incluye, además, el diagrama de despliegue, estándares de interfaz y de codificación, así como el consenso en cuanto a la factibilidad de la solución propuesta.

Abstract

The New information technologies make it possible, in a more effective attention to the different branches of business processes for the implementation of any strategy is necessary or advisable, in many cases, because of the need to computerize the information, following the global development automation of operations, supported by some software to help collect historical data results and to optimize the flow of these.

Drugstore EMCOMED Company previously conducted Holguín control of management indicators for areas individually by various means (email, text files, spreadsheets, etc...) And represented by different patterns (graphs, charts). This causes delay in information and unintentional mistakes that makes the integrity of unreliable information.

It raised before a Web application was implemented flexible, easy to administer and pleasant interaction, which is capable of running efficiently control management indicators Scorecard, helping to improve the interpretation and evaluation of themselves. ICONIX methodology was used for modelling analysis and system design, as manager of MySql database and PHP programming language.

This document is structured in four chapters. The Chapter 1 contains all the theoretical foundations of research methods, methodologies, technologies and tools used for development work. Chapter 2, "Description of the Settlement Proposal," contains the most important aspects of Web site development proposed, such as correspondence with the methodology ICONIX, the use case model and the persistence mechanisms used by the Box Scorecard. In addition, a study of sustainability according to their socio-humanistic, administrative, technological and environmental, as well as their evolution. Chapter 3, "Development and Evaluation of the proposed solution, structured and refined the requirements captured from workflow analysis and design, resulting in the implementation, also includes the deployment diagram, standards interface and coding, and consensus as to the feasibility of the proposed solution.

ÍNDICE

INTRODUCCIÓN

El logro de la competitividad de la organización debe estar referido a un plan, el cual fija la visión, misión, objetivos y estrategias corporativas con base en el adecuado diagnóstico situacional. Las áreas funcionales establecen, sobre la base del plan corporativo, unos objetivos, que garanticen el logro del éxito de la gestión de la organización; con base en esos objetivos y planes, cada área efectúa una asignación de requisitos para su ejecución. Tras la iniciación de la ejecución de los mencionados planes, surge una serie de inquietudes:

- ¿Tengo los recursos adecuados para ejecutar los planes?
- ¿Cómo vamos?
- ¿Dónde estamos frente al plan trazado?
- ¿Qué tan eficientemente estoy utilizando mis recursos?
- ¿Qué tan bien o qué tan mal voy?, ¿voy para donde es?
- ¿Cómo regreso al rumbo correcto?, ¿cómo lo mantengo?

El riesgo involucrado en administrar en el "limbo", o en medio de la desinformación precisa y objetiva es demasiado grande, y las condiciones actuales no permiten darse el gusto de seguir administrando con base en supuestos y vaguedades. De manera que una vez más surge como respuesta y garantía del éxito el hecho de contar y administrar correctamente un adecuado sistema de indicadores de gestión.

La Empresa EMCOMED Droguería Holguín, es la encargada de la comercialización y distribución de 5000 tipos de productos nacionales e importados; estos productos se clasifican en: medicamentos, reactivos químicos, material antiséptico y materias primas. La empresa tiene 21 proveedores y suministra productos a 463 clientes de los que el 93% pertenecen al Sistema Nacional de Salud. Además de asegurar el suministro de medicamentos; reactivos y otros productos a los Programas de Cooperación que se

desarrollan en el exterior, principalmente a la Republica Bolivariana de Venezuela, Bolivia, Ecuador y a otros países que intervienen en la Misión Milagro Nacional e Internacional; indicados por el Consejo de Estado. Por concepto de ventas de productos la empresa maneja entre 85-87 millones de CUP anuales.

Entre las actividades para lograr alcanzar los objetivos estratégicos recogidos en la visión: "Prestar servicios de calidad con la garantía de mantener la existencia de medicamentos y otros productos esenciales...mediante una operación eficiente, con las mejores condiciones de entrega, bajos costos…", la Empresa realiza un control y vigilancia estricta de los indicadores de gestión.

Como parte del control y vigilancia de los indicadores de gestión de la Empresa, frecuentemente se solicitan los resúmenes de los indicadores por áreas funcionales. Para la elaboración de estos resúmenes cada área funcional brinda la información correspondiente por distintas vías (correo electrónico, archivos de texto, hojas de calculo, etc.) los indicadores se recolectan y se representan en distintos tipos de esquemas (gráficas, diagramas). Sin duda esta es una tarea difícil si se piensa en la complejidad de una empresa con tantas áreas funcionales y que cada una brinda la información del estado de sus indicadores de gestión en un formato distinto; teniendo en cuenta también la celeridad en la entrega de información que demanda este tipo de organización.

La Empresa EMCOMED Droguería Holguín hace un uso intensivo de Las Tecnologías de la Información y las Comunicaciones (TICs) para la automatización de la gestión de la información de la organización. En la actualidad cuentan con un Sistema integrado de Gestión de la Calidad, un Sistema de Gestión Medioambiental, entre otros, integrados en una Intranet Empresarial. Son concientes de las ventajas que brindan las TICs para la construcción de herramienta que apoyen el proceso de toma de decisiones por los directivos. De ahí su interés en construir un sistema automatizado que le permita controlar los indicadores de gestión de la empresa.

Teniendo en cuenta la situación antes analizada se plantea el siguiente **Problema científico:** ¿Cómo automatizar el control de los indicadores de gestión de la Empresa EMCOMED Droguería Holguín que contribuya a mejorar la interpretación y evaluación de los mismos? El problema anterior está enmarcado en el ***objeto de estudio*:** el control de los

indicadores de gestión en la Empresa EMCOMED Droguería Holguín. Para resolver el problema planteado se propuso el siguiente ***objetivo***: construcción de una herramienta informática para la automatización del control de los indicadores de gestión de la Empresa EMCOMED Droguería Holguín que contribuya a mejorar la interpretación y evaluación de los mismos. El ***campo de acción es:*** Automatización del control de los indicadores de gestión de la Empresa EMCOMED Droguería Holguín.

Para guiar la investigación se plantea la siguiente ***hipótesis*:** Con la implementación de una aplicación Web para el control de los indicadores de gestión de la Empresa EMCOMED Droguería Holguín, se contribuirá a mejorar la interpretación, evaluación y representación histórica de estos indicadores.

Para darle respuesta a la hipótesis planteada y cumplir el objetivo trazado, se realizaron las siguientes ***tareas:***

1. Fundamentación teórica de las tecnologías Web y sistemas de gestión de bases de datos.
2. Búsqueda de posibles soluciones existentes con características similares a las del que se pretende elaborar.
3. Valoración de la sostenibilidad del sistema Web según su impacto socio-humanista, administrativo, tecnológico y ambiental, así como su evolución.
4. Elaboración del Manual de Usuario de la aplicación.
5. Elaboración de una aplicación Web para el control de los indicadores de gestión de la Empresa EMCOMED Droguería Holguín.
6. Búsqueda de consenso en cuanto a la validación de la solución propuesta mediante el criterio de expertos.

Para realizar las tareas antes mencionadas se emplearon métodos teóricos y empíricos de la investigación científica. Entre los **métodos teóricos** utilizados están:

Análisis y síntesis: Utilizamos este método para desglosar el problema en partes o subproblemas para de esta forma comprobar el correcto funcionamiento de las mismas, luego integrarlo todo para corroborar las relaciones entre estas y su integración como un todo, llegando así a una mejor solución, también para arribar a conclusiones parciales y generales de la investigación.

Modelación: Utilizamos este método para reproducir el fenómeno que se está estudiando y así lograr una mejor comprensión de la las situaciones a resolver.

Histórico-Lógico: para la comprensión de la evolución del empleo de herramientas para la gestión de la información que es generada por la interacción de dos o más personas.

Entre los **métodos empíricos** utilizados se encuentran:

Entrevista: la utilizamos con el objetivo de recopilar información, esta será la vía fundamental para la determinación de los requerimientos del sistema.

Encuestas: Fueron desarrolladas para elegir los expertos sobre los sitios Web y la Interfaz Gráfica de Usuario del sitio Web propuesto y obtener valoraciones conclusivas de éstos sobre el sistema. Para el tratamiento de las encuestas se hizo uso de métodos estadísticos, como el trabajo con por cientos y la frecuencia, y el método Delphi para buscar el consenso de los encuestados.

Revisión de documentos: lo utilizamos para conocer los detalles del funcionamiento la empresa EMCOMED es decir, reglas y particularidades de esta empresa además de permitirnos justificar nuestra solución al problema planteado.

El trabajo que se presenta consta de tres capítulos. **El *Capítulo 1*** se titula "Determinación del Estado del Arte" y contiene los fundamentos de los aspectos relacionados con el objeto de estudio y del uso de las TICs (Tecnologías de la Informática y las Comunicaciones). Además en este capítulo se hace una descripción de las principales tendencias y tecnologías para la construcción de la solución propuesta, la metodología de Ingeniería de Software empleada, el lenguaje de programación seleccionado y una descripción del objeto de estudio de la investigación.

El *Capítulo 2*, "Descripción de la Solución Propuesta", contiene los aspectos de mayor peso del desarrollo del sitio Web propuesto, como son la correspondencia con la metodología *ICONIX*, el modelo de casos de uso y los mecanismos de persistencia que utiliza El Cuadro de Mando Integral. Además se hace un estudio de sostenibilidad según su impacto socio-humanista, administrativo, tecnológico y ambiental, así como su evolución.

El *Capítulo 3*, "Elaboración y Evaluación de la Solución Propuesta", se estructuran y refinan los requerimientos capturados a partir del flujo de trabajo de análisis y diseño, que da lugar a la implementación; incluye, además, el diagrama de despliegue, estándares de

interfaz y de codificación, así como el consenso en cuanto a la factibilidad de la solución propuesta.

Para concluir se muestran las conclusiones a las que se arribaron, las recomendaciones que se proponen, la bibliografía utilizada y anexos con información necesaria sobre el trabajo.

CAPÍTULO 1

DETERMINACIÓN DEL ESTADO DEL ARTE

1.1 Introducción

Las Nuevas Tecnologías Informáticas posibilitan, de manera más efectiva, la atención a las diferentes ramas de los procesos empresariales pues para la aplicación de cualquier estrategia es necesario o aconsejable, en muchas ocasiones, debido a la necesidad de informatizar la información, consecuencia del desarrollo global de la automatización de las operaciones, apoyarse en algún software que ayude a recoger los datos históricos de los resultados obtenidos y permita optimizar la fluidez de estos. La efectividad del uso y explotación de una herramienta informática dentro de la gestión directiva solo puede ser evaluada y medida por el análisis exhaustivo de una amplia variedad de factores que incluyen desde la necesidad e importancia que produzca para la empresa la implantación de la misma, hasta la organización de los datos a evaluar **[27]**.

En este Capítulo se manejan los principales conceptos referidos a la investigación. Se fundamenta el uso de herramientas y metodologías así como también se determina el entorno de la aplicación a desarrollar durante el transcurso de las etapas que sostendrá el proyecto.

1.2 Estado del Arte

1.2.1 Conceptos Fundamentales

El Balanced Scorecard o Cuadro de Mando Integral

Es una metodología que pretende integrar los aspectos de la dirección estratégica y la evaluación del desempeño del negocio. Reconocidas corporaciones internacionales han obtenido excelentes resultados con esta metodología, y desde su divulgación en 1992 por

sus dos autores **Robert Kaplan** y **David Norton [2]**, su definición es la representación en una estructura coherente, de la estrategia del negocio a través de objetivos claramente encadenados entre sí, medidos con los indicadores de desempeño, sujetos al logro de unos compromisos (metas) determinados y respaldados por un conjunto de iniciativas o proyectos **[1]**, el cual ha sido incorporado a los procesos de gerencia estratégica de un 60% de las grandes corporaciones en los Estados Unidos, extendiéndose su uso a varias corporaciones europeas y asiáticas.

Indicador

Es un dato que pretende reflejar el estado de una situación, o de algún aspecto particular, en un momento y un espacio determinados. Habitualmente se trata de un dato estadístico (porcentajes, tasas, razones…) que pretende sintetizar la información que proporcionan los diversos parámetros o variables que afectan a la situación que se quiere analizar.

Un indicador se toma o mide dentro de un período de tiempo determinado, para poder comparar los distintos períodos. La comparación de mediciones permite ver la evolución en el tiempo y estudiar tendencias acerca de la situación que miden, adquiriendo así un gran valor como herramienta en los procesos de evaluación y de toma de decisiones **[36]**.

1.2.4 Sistemas automatizados existentes vinculados al campo de acción.

Los indicadores de gestión se trabajan muy bien dentro de cada una de las perspectivas del CMI; actualmente un gran número de organizaciones están adoptando el Cuadro de Mando Integral, según sus características y objetivos, como una de las principales herramientas para la Gestión Empresarial. En el mundo existen muy pocas aplicaciones automatizadas para el CMI, las que existen hay que pagarlas. La mayoría de las empresas que necesitan un CMI elaboran su propia aplicación o utilizan plantillas de Microsoft Excel.

Microsoft Excel es una herramienta informática capaz de realizar cálculos complejos, pero no lleva a cabo el almacenamiento de los datos obtenidos, ni valida la seguridad para los usuarios y no brinda la interacción cliente servidor.

En el mundo farmacéutico existen varias herramientas informáticas privatizadas, como la aplicación DELFOS , que sigue la metodología J2EE, con componentes Open Source, implementando JavaMail y JFreeChart para los gráficos estadísticos, además posee un

motor de bases de datos PostgreSQL **[26]**. Existen otras como el Software: QPR Scorecard 7.0 de la Empresa QPR Software Plc. de Noruega **[28]**, Delphos 350.10.0 de la Empresa: Deinsa de Costa Rica **[29],** SPImpact Balanced Scorecard NA Empresa SPImpact de los Estados Unidos **[30],** y Pilot BusinessMonitor NA Empresa Pilot Software de los Estados Unidos **[31]** entre otras aplicaciones de producción norteamericana.

En Cuba la automatización del mismo no está muy difundida, no obstante se conoce que en el 2003 la empresa de Inspección de Intermar posee un sistema llamado ODUN, diseñado para funcionar en un gestor de base de datos en MS ACCESS. En ese mismo año, la empresa GET Varadero implementó su propio sistema utilizando plantillas de Microsoft Excel. En el 2004 SEPSA de Cienfuegos diseñó y automatizó su propia aplicación; Web y en el Holguín en el 2007 se diseñó e implementó parcialmente un sistema en la Sucursal de Almacenes Universales S.A., con un gestor de de base de datos MY SQL.

La base de datos MS ACCESS tiene como inconveniente la restricción de tamaño de 2GB lo cual traería como consecuencia que si se planea desarrollar una aplicación Web con una base de datos grande y con gran volumen de información, podría tener problemas de almacenamiento en la base de datos .

El acceso multiusuario a una base de datos MS Access se vuelve lento si coinciden 5 usuarios simultáneos. Los sitios Web que tienen más de 20 usuarios simultáneos tendrán grandes dificultades con Access.

Se realizó un estudio profundo de las posibles aplicaciones, llegando a la conclusión que estas vías de solución para la Droguería Holguín no es factibles ya que tendrían que pagar una licencia por utilizar un gestor de base de datos en MS ACCESS, y esta aplicación no brinda la posibilidad de migrar a otro sistema operativo lo cual sería un obstáculo ya que EMCOMED Droguería Holguín emigró a la plataforma Linux. Además la Droguería Holguín es una empresa que posee una gran cantidad de trabajadores que tendrán acceso a la aplicación, por lo que una aplicación Web con un gestor de Base de Datos (BD) en Access disminuiría la velocidad de acceso a la aplicación.

Las restantes vías de solución se concentraron en realizar el cálculo del estado actual de los indicadores. Lo cuales no emplean las necesidades reales de la Organización en su automatización, y no dan muestra de un comportamiento histórico para que la Organización pueda tener una imagen de cómo proyectarse al futuro. Y las restantes soluciones por estar

privatizadas en compañías extranjeras, lo cual hay que pagar un costo muy elevado par su uso, en un momento crítico de nuestra economía nacional ante la crisis mundial.
Por lo antes planteado no se tuvo en cuenta ninguna vía de solución.

1.3 Tendencias y tecnología Actuales

1.3.1 Desarrollo Web

1.3.1.1 Arquitectura Cliente-Servidor en la Web

La arquitectura Cliente/Servidor es la plataforma abierta por excelencia, por la variedad de combinaciones de clientes y servidores que permite conectar en red. Sin embargo, elegir las plataformas, las herramientas, los proveedores y las bases de administración de la arquitectura Cliente/Servidor, además de la tecnología de creación, es una decisión difícil de tomar.
Elegir un servidor es una cuestión muy complicada; para aplicaciones pequeñas y medianas, todos los servidores han probado ser muy buenos, las diferencias se darán cuando se necesiten altísimos regímenes transaccionales, y dependerán de cómo cada uno vaya incorporando nuevas características como paralelismo, "read ahead". Cada nueva versión puede modificar las posiciones y los principales fabricantes están trabajando al ritmo de una gran versión nueva por año.
Hoy en día, el modelo Cliente/Servidor se considera clave para abordar las necesidades de las empresas. El proceso distribuido se reconoce actualmente como el nuevo paradigma de sistemas de información, en contraste con los sistemas independientes. Este cambio fundamental ha surgido como consecuencia de importantes factores (negocio, tecnología, proveedores), y se apoya en la existencia de una gran variedad de aplicaciones estándar y herramientas de desarrollo, fáciles de usar que soportan un entorno informático distribuido.

1.3.1.2 Características de la Arquitectura Cliente/Servidor

La arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

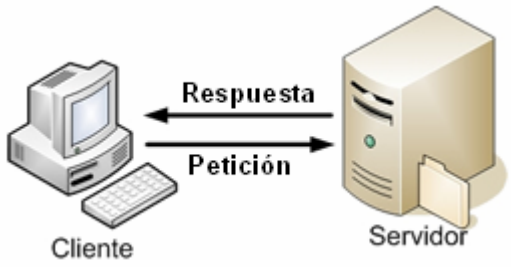


Figura 1-1. *Arquitectura Cliente-Servidor.*

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, inicia un proceso de diálogo: produce una demanda de información o solicita recursos. La computadora que responde a la demanda del cliente se conoce como servidor. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet. Cliente/Servidor es el modelo de interacción más común entre aplicaciones en una red. No forma parte de los conceptos de la Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo.

Se puede decir que la arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información, estableciendo así un enlace de comunicación transparente entre los elementos que conforman la estructura. No existe una definición específica adoptada universalmente de la Arquitectura Cliente/Servidor, las empresas de cómputo enfocan el concepto basándose en la funcionalidad que representa según los servicios que ellas mismas ofrecen **[7]**.

Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

- ✓ El servidor presenta a todos sus clientes una interfaz única y bien definida.
- ✓ El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.

- ✓ El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- ✓ Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Todos los sistemas desarrollados en arquitectura Cliente/Servidor poseen las siguientes características distintivas de otras formas de software distribuido **[7]**:

- ✓ Servicio: El servidor es un proveedor de servicios; el cliente es un consumidor de servicios.
- ✓ Recursos compartidos: Un servidor puede atender a muchos clientes al mismo tiempo y regular su acceso a recursos compartidos.
- ✓ Protocolos Asimétricos: La relación entre cliente y servidor es de muchos a uno; los clientes solicitan servicios, mientras los servidores esperan las solicitudes pasivamente**.**
- ✓ Transparencia de ubicación: El software Cliente/Servidor siempre oculta a los clientes la ubicación del servidor.
- ✓ Mezcla e igualdad: El software es independiente del hardware o de las plataformas de software del sistema operativo; se puede tener las mismas o diferentes plataformas de cliente y servidor.
- ✓ Intercambio basados en mensajes: Los sistemas interactúan a través de un mecanismo de transmisión de mensajes: la entrega de solicitudes y respuestas del servicio.
- ✓ Encapsulamiento de servicios: Los servidores pueden ser sustituidos sin afectar a los clientes, siempre y cuando la interfaz para recibir peticiones y ofrecer servicios no cambie.
- ✓ Facilidad de escalabilidad: Los sistemas Cliente/Servidor pueden escalarse horizontal o verticalmente. Es decir, se pueden adicionar o eliminar clientes (con apenas un ligero impacto en al desempeño del sistema); o bien, se puede cambiar a un servidor más grande o a servidores múltiples.
- ✓ Integridad: El código y los datos del servidor se conservan centralmente; esto implica menor costo de mantenimiento y protección de la integridad de los datos compartidos. Además, los clientes mantienen su individualidad e independencia.

La arquitectura Cliente/Servidor es una infraestructura versátil modular y basada en mensajes que pretende mejorar la portabilidad, la interoperabilidad y la escalabilidad del cómputo; además es una apertura del ramo que invita a participar a una variedad de plataformas, hardware y software del sistema **[7]**.

Cliente

El cliente es la entidad por medio de la cual un usuario solicita un servicio, realiza una petición o demanda el uso de recursos. Este elemento se encarga, básicamente, de la presentación de los datos y/o información al usuario en un ambiente gráfico.

Los clientes se suelen situar en PC's o en estaciones de trabajo se encargan de realizar el FRONT END, que es la parte de la aplicación que interactúa con el usuario, en ellos permanecen las aplicaciones particulares de cada usuario, y realizan funciones como:

- ✓ Manejo de la interfaz del usuario.
- ✓ Captura y validación de los datos de entrada.
- ✓ Generación de consultas e informes sobre las bases de datos.

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el cliente para hacer acceso a servidores remotos (por ejemplo, servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios.

Servidor

El servidor es la entidad física que provee un servicio y devuelve resultados; ejecuta el procesamiento de datos, aplicaciones y manejo de la información o recursos. En el servidor se realiza el BACK END que es la parte destinada a recibir las solicitudes del cliente y donde se ejecutan los procesos.

En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además, deben manejar los interbloqueos, la recuperación ante fallas, y otros aspectos afines. Por las razones anteriores, la plataforma computacional asociada con los servidores es más poderosa que la de los clientes. Por esta razón se utilizan PC’s poderosas, estaciones de trabajo, minicomputadores o sistemas grandes. Además deben manejar servicios como administración de la red, mensajes, control y administración de la entrada al sistema ("login"), auditoría y recuperación, y contabilidad.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- ✓ Gestión de periféricos compartidos.
- ✓ Control de accesos concurrentes a bases de datos compartidas.
- ✓ Enlaces de comunicaciones con otras redes de área local o extensa.
- ✓ Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste, le responde proporcionándolo.

Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en computadoras personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

En el servidor permanecen las aplicaciones que deben ser compartidas por varios usuarios. Normalmente, aunque con excepciones, el cliente y el servidor están ubicados en distintos procesadores; incluso, un servidor puede fungir como cliente de otros servidores.

Existen diversos servidores mismos que se clasifican basándose en su funcionalidad; estos son denominados servidores dedicados ya que administran el uso de algún recurso en particular, por ejemplo, Servidor de Archivos: El cliente envía solicitudes de registros de archivos al servidor, es un simple servicio de datos compartidos por medio de la red. Servidor de bases de datos: El cliente envía solicitudes de SQL en calidad de mensajes (un mensaje por instrucción); el servidor hace uso de su propia capacidad de procesamiento para encontrar los datos solicitados y devolverlos por medio de la red. Servidores Web: Se usan como una forma inteligente para comunicación entre empresas a través de Internet. Este servidor permite transacciones con el acondicionamiento de un browser específico. Este modelo está integrado por clientes compactos y portátiles en comunicación con servidores amplios. Tal comunicación se da mediante un protocolo denominado HTTP del inglés Hypertext Transfer Protocol.

1.3.1.3 Patrón de Diseño Modelo-Vista-Controlador (MVC) aplicado a la Web

Hoy en día se trabaja para mejorar la calidad del software (tanto de nuestros desarrollos como los de nuestros clientes). Para esto consideramos que es fundamental el uso de

patrones de diseño[1]. Uno de los patrones más conocidos en el desarrollo Web es el patrón MVC (Modelo Vista Controlador). Este patrón nos permite y obliga a separar la lógica de control (sabe qué cosas hay que hacer pero no cómo), la lógica de negocio (sabe cómo se hacen las cosas) y la lógica de presentación (sabe cómo interactuar con el usuario).

Para comprender cómo trabajan los frameworks Web existentes es imprescindible conocer el patrón MVC. El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados: el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio; el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información; el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.

1.4 Tendencias y Tecnologías Actuales a considerar

1.4.1 La Web 2.0

El término Web 2.0 fue aprobado por Dale Dougherty de O'Reilly Media Company **[8]** en el año 2003 para describir esta tendencia, término que se hizo muy importante, pero pocas personas saben qué representa realmente. En general, las compañías que incorporan la Web 2.0 usan esta tecnología como una plataforma para crear sitios colaborativos entre comunidades. La Web 2.0 ha sido popularizada por la compañía O'Reilly en la cumbre anual efectuada desde el año 2004, definiendo en uno de sus artículos que la Web 2.0 es un conjunto de patrones de diseño y modelos del negocio para las próximas generaciones de softwares.

El crecimiento de la Web 2.0 puede ser atribuido a algunos factores claves: Primero, el hardware es cada vez más barato y más rápido, con capacidades de memoria y velocidades que se incrementan paulatinamente; esto permite que el desarrollo de aplicaciones se realice con altas demandas.

[1] Patrón de diseño: Solución apropiada a un problema común en un contexto. A pesar de que este concepto incluye el término "diseño", no se suele considerar que se refiera a la fase de diseño de un programa, es una solución completa que incluye análisis, diseño e implementación.

Segundo, el uso del ancho de banda de Internet ha estallado: un estudio realizado en marzo de 2006 encontró que el 42% de adultos estadounidenses tenía alta velocidad de Internet en sus casas y un 35% de los usuarios de Internet tenían conexión directa. Sin esta velocidad del ancho de banda toda la documentación digital en línea no hubiera sido posible.

Tercero, la disponibilidad del software libre abundante ha resultado en opciones de software más barata (y a menudo libre). Esto hace más fácil empezar con la Web 2.0 en diferentes compañías y reducir el gasto del error enormemente.

¿Qué es la Web 2.0?

La Web 2.0 involucra al usuario; no sólo es el contenido creado por usuarios a menudo, si no que ayudan a organizarlo, compartirlo, criticarlo y actualizarlo; una manera de mirar la Web 1.0 es como una conferencia, una pequeña cantidad de catedráticos que informan a una audiencia grande de estudiantes. En comparación, la Web 2.0 es una conversación con todos teniendo la oportunidad de hablar y compartir diferentes puntos de vistas **[8]**.

La Web 2.0 acepta una *arquitectura de participación*, un diseño que apoya las interacciones del usuario y las contribuciones. Muchas compañías Web 2.0 son construidas casi completamente producto al contenido generado por el usuario y el control de la inteligencia colectiva. La trascendencia no sólo es tener contenido generado por usuario, sino cómo usarlo. La arquitectura de participación se ve en el desarrollo de software. El software de código abierto está disponible para cualquier persona pudiendo usarlo e incluso modificarlo con pocas o ningunas restricciones, esto ha tenido un papel muy importante en el desarrollo de la Web 2.0. Google, la compañía de mayor publicidad de búsquedas en Internet, envía a sus usuarios a sitios Web generados por usuario considerando que es lo que los usuarios colectivamente han valorado anteriormente. Para sitios Web como MySpace, Flickr, YouTube y Wikipedia, los usuarios crean el contenido, mientras que los sitios proveen las plataformas. Estas compañías confían en sus usuarios; sin tal confianza, los usuarios no pueden hacer contribuciones importantes a los sitios **[8]**.

Existen un conjunto de tecnologías que fueron usadas para crear aplicaciones que soportan la Web 2.0 y que ahora forman parte de la misma como:

RIAs (Rich Internet Applications): esta tecnología ofrece aplicaciones Web con características similares a aplicaciones Desktop; son un nuevo tipo de aplicaciones con más

ventajas que las tradicionales aplicaciones Web. Hay muchas herramientas para la creación de entornos RIA. Entre estas se puede mencionar las plataformas Adobe Flash y Adobe Flex de Adobe, OpenLaszlo, Silverlight de Microsoft, JavaFX Script de Sun Microsystems y Bindows de MB Technologies entre otras; Ajax (Asynchronous JavaScript and XML), XHTML (Extensible Hypertext Markup Language*)*, CSS (Cascading Style Sheets*),* DOM (The Document Object Model), XML (Extensible Markup Language): estas tecnologías incorporan JavaScript, como el lenguaje de programación que permite diseñar programas de computadora que aumentan la funcionalidad y la apariencia de páginas Web. JSF (Java Server Faces): Framework escrito en java para desarrollar aplicaciones Web basadas en el patrón MVC que proporciona una interfaz gráfica similar a aplicaciones desktop **[8]**.

Con el término Web 2.0, se subraya un cambio de paradigma sobre la concepción de Internet y sus funcionalidades, que ahora abandonan su marcada unidireccionalidad y se orientan más a facilitar la máxima interacción entre los usuarios y el desarrollo de redes sociales (tecnologías sociales) donde puedan expresarse y opinar, buscar y recibir información de interés, colaborar y crear conocimiento (conocimiento social), compartir contenidos. Podemos distinguir algunas como blog, wiki, podcast, YouTube, Flickr, SlideShare, Del.icio.us, RSS, XML, Atom, Bloglines, GoogleReader, buscadores especializados, BSCW, Ning, Second Life, Twitter entre otras.

La Web 2.0 pone en práctica una actitud y procedimientos que conducen a satisfacer la experiencia del usuario en Internet, a través de un mayor nivel de interacción con él. Partiendo del principio de funcionamiento de esta nueva filosofía Web, las compañías tienen acceso a un número más amplio de clientes, además de conseguir recopilar grandes BBDD; éstas últimas son una ventaja competitiva clave para posicionarse sobre el resto del mercado. Las compañías logran una evolución y mejora constante con el cliente, al integrar a los usuarios como probadores en tiempo real, y comprobar su comportamiento y las funcionalidades que utilizan.

La Web 2.0 simplifica y hace flexible el acceso del cliente a la compañía, posibilitando a su vez que éste aporte valor directa o indirectamente y enriquezca la información que ya se posee. En esta línea, facilita aplicar el principio de inteligencia colectiva y el modelo "menos es más": modelos de programación y negocio simples que permiten a las empresas alcanzar el éxito y la eficiencia. Antes el avance indiscutible de la automatización,

integración y reutilización de datos, pronto hablaremos de Web 3.0, o Web semántica, que desarrolla lenguajes capaces de expresar información procesable de forma automática.

1.4.2 Software Libre

En 1983 un grupo de hackers, liderados por Richard Stallman, consideran que el software no debe ser objeto comercial pues, para ellos, se trata de conocimiento científico y como tal, debe transmitirse libre y ser útil para el progreso de la humanidad. Con esta intención crean la Fundación para el Software Libre (FSF, Free Software Foundation) **[9]** y el proyecto GNU, con el objetivo de producir aplicaciones de libre distribución amparadas en una licencia que las proteja de las patentes comerciales; esta licencia se conoce con las siglas GPL (General Public License). En la actualidad de software tiene un papel muy destacado en la sociedad y es importante contar con métodos transparentes en sus diferentes fases de producción y explotación. El software libre, al dar acceso al código, es el único que puede garantizar esta transparencia **[4]**.

Actualmente, el software libre es una realidad, las soluciones diseñadas pueden ser utilizadas por cualquier persona, ya que ahora no hay que ser un experto en informática para poder trabajar con las aplicaciones creadas para el entorno de usuario final. Los consultores opinan que el software libre es una amenaza competitiva para los sistemas basados en software propietario. Una encuesta realizada por la consultora Forrester indica que más del 70% de las 50 grandes compañías con una facturación superior a 1.000 millones de dólares contemplan implantar sistemas de software libre en un futuro próximo, así como que más del 50% de las 2.500 mayores empresas utilizan algún producto informático basado en Open Source **[10]**.

Cuando los ingenieros informáticos se encuentran ante un programa del cual no pueden tener acceso al código fuente, es decir, que no es libre sino propietario, no pueden tener una comprensión total del mismo, y aun cuando sean capaces de detectar algún error y tener la solución en sus manos no podrán arreglarlo **[11]**.

Para que un software se considere libre debe basarse, según la Free Software Fundation, en las siguientes libertades **[4]**:

- ✓ Libertad para utilizar el programa para cualquier propósito.

- ✓ Libertad para poder estudiar cómo funciona el programa. Implica acceso al código fuente del mismo.
- ✓ Libertad para redistribuir el programa.
- ✓ Libertad para hacer modificaciones y distribuir las mejoras. Implica también acceso al código fuente del mismo.

El software libre se basa en la cooperación y la transparencia y garantiza una serie de libertades a los usuarios. Estos aspectos, junto al hecho de que su desarrollo ha sido paralelo al de Internet, han causado que sea abanderado para un gran número de usuarios que tienen una concepción libertaria del uso de las nuevas tecnologías **[4]**.

1.4.2.1 Open Source

Durante el año 1998, Eric S. Raymond, Bruce Perens y otros hackers involucrados en el desarrollo de software libre lanzaron la Open Software Initiative y propusieron el uso de término open source (código abierto) en contraposición al término free software (software libre) como término más atractivo al entorno empresarial. El término free software en el mundo anglófono creaba una situación incómoda debido a la doble acepción que en inglés tiene el término free (que puede significar gratuito o libre). La gran mayoría de empresas en Estados Unidos usan principalmente el término código abierto para evitar dar la percepción que el software libre es un recurso totalmente gratuito y para poner énfasis en valor diferencial que representa el hecho de que el código fuente está disponible **[4]**.

1.4.3 Lenguajes de desarrollo 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 alguno de los diferentes lenguajes de programación para Web que existen hoy en día. Dichos lenguajes se clasifican en dos partes fundamentales que reconocen la propia arquitectura Cliente/Servidor de esta plataforma de desarrollo.

Entre los lenguajes del lado del servidor se pueden encontrar entre los más sobresalientes al Practical Extracting and Reporting Language (PERL)**[12]**, Active Server Pages (ASP) **[13]**, Personal Home Page (PHP) **[14]**, Java Server Pages (JSP) **[15]**, Java Server Faces (JSF) **[16]**. 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, etc. Del lado del cliente se encuentran principalmente el JavaScript y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores.

PHP

PHP **[14]** (Personal Home Page) es el acrónimo de Hypertext Preprocessor. Es un lenguaje de programación gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML.

PHP, en el caso de estar montado sobre un servidor Linux o Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre 2 componentes COM[2] que se realizan entre todas las tecnologías implicadas en una página ASP.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones Web actuales PHP es la gran tendencia en el mundo de Internet. Últimamente se puede observar un ascenso imparable, ya que cada día son muchísimas más las páginas Web que lo utilizan para su funcionamiento, según las estadísticas, PHP se utiliza en más de 10 millones de páginas, y cada mes realiza un aumento del 15%.

Resumiendo, el PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 Bases de Datos y contiene unas 40 extensiones estables sin contar las que se están experimentando, además:

[2] Iniciales de Common Objet Model. Un Componente desarrollado por Microsoft para el trabajo con Aplicaciones Web.

- Es software libre, lo que implica menos costes y servidores más baratos que los disponibles para otras alternativas.
- Es muy rápido. Su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se esta familiarizado con esta sintaxis, le resultara muy fácil aprender PHP.
- Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP.
- PHP tiene una de las comunidades más grandes en Internet, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos.

¿Por qué PHP?

Hasta el momento se han analizado las características fundamentales de los lenguajes de programación candidatos para la implementación de la propuesta de este trabajo, para fundamentar nuestra elección haremos una comparación teniendo en cuenta algunas características que influyen directamente en el ambiente de trabajo donde se va a desarrollar la propuesta. En cuanto a:

- ✓ **Características de multiplataforma:** Menos el ASP, que es solamente soportado por 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 en los 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 estas comparaciones, para nosotros el PHP resulta mucho más favorecido, por tanto pensamos que es el adecuado para implementar la propuesta de sistema de este trabajo.

1.4.4 Desarrollo ágil de aplicaciones

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, librerías y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Un "Framework de Software" es un diseño reusable de un sistema (o subsistema), que está expresado por un conjunto de clases abstractas y el modo en que sus instancias colaboran para un tipo específico de software. Todos los frameworks de software son diseños Orientados a Objetos.

Si aplicamos la definición anterior al desarrollo Web, podemos llegar a la conclusión que un Framework Web es una estructura definida, reusable en el que sus componentes facilitan la creación de aplicaciones Web.

1.4.4.1 Comparación entre Codeigniter y otros Frameworks Web

Habiendo mencionado algunas características de los Frameworks Web, a continuación se ofrece una evaluación sobre las ventajas y desventajas observadas en cada uno de los frameworks más utilizados en la actualidad.

Struts

Ventajas:

- ✓ Más de 6 años demostrando que funciona. Gran cantidad de desarrollos de gran envergadura concretados exitosamente.
- ✓ Es el framework más popular de la comunidad Java por lo que existen infinidad de material disponible en la Web. Buenas prácticas conocidas.
- ✓ Documentación muy buena.
- ✓ Permite crear sitios internacionales de manera rápida y efectiva.
- ✓ Curva de aprendizaje mediana.
- ✓ Open Source (Licencia Apache).

Desventajas:

- ✓ No abstrae completamente al desarrollador del funcionamiento del protocolo HTTP.

- ✓ Aunque se adapta a las incorporaciones de diferentes bibliotecas de tags no está diseñado para facilitar la creación de componentes propios. No es natural el mapeo de los datos ingresados a los objetos del negocio.
- ✓ Las vistas quedan atadas al dispositivo en el cual se renderizan. No facilita el armado de vistas independientes del dispositivo.
- ✓ No es una especificación.

Tapestry

Ventajas:

- ✓ Open Source (Licencia Apache).
- ✓ Permite el desarrollo de componentes propios.
- ✓ Los diseñadores Web no necesitan aprender tags nuevos ni diferentes lenguajes ya que los templates se codifican en HTML estándar.
- ✓ La creación de componentes es relativamente sencilla.
- ✓ Separación completa entre la lógica y la presentación (HTML de Java).

Desventajas:

- ✓ Comunidad de desarrolladores pequeña-mediana.
- ✓ Escasa documentación. Pocos libros editados. Poca información en la Web.
- ✓ Se requiere configurar 3 archivos para cada página a crear.

ASP.NET

Ventajas:

- ✓ Curva de aprendizaje baja.
- ✓ Permite el desarrollo de controles propios y utilizar controles de terceros.
- ✓ Soporte oficial y amplia documentación.
- ✓ Permite binding directo entre los componentes y los orígenes de datos.
- ✓ Permite desarrollo con herramientas RAD.

Desventajas:

- ✓ Propietario de Microsoft. Sólo funciona con Information Server.
- ✓ Requiere un IDE como Visual Studio para un desarrollo productivo, lo que deviene en un costo por desarrollador por el licenciamiento del IDE.
- ✓ El control de navegación no está centralizado.

- ✓ Código cerrado. Ante la aparición de bugs dentro del framework se depende de Microsoft para solucionarlo.
- ✓ Varias de las funcionalidades importantes (maquetación, internacionalización) sólo están disponibles a partir de la versión 2.0.
- ✓ Requiere javascript y cookies para funcionar correctamente.
- ✓ El estado interno de la vista (viewstate) viaja codificado dentro de un campo hidden. Esto trae problemas de rendimiento y si se utiliza mal, problemas de seguridad.

Cocoon

Ventajas:

- ✓ Permite separar claramente el contenido de la presentación y de la lógica.
- ✓ Open Source (Licencia Apache).
- ✓ Permite modificar el comportamiento de la aplicación sin conocer el lenguaje en el que está implementado.

Desventajas:

- ✓ Requiere amplios conocimientos de hojas de estilo XSL por parte de los diseñadores gráficos.
- ✓ Comunidad relativamente pequeña.
- ✓ Curva de aprendizaje elevada.
- ✓ La transformación de XMLs requiere bastante capacidad de proceso.

Grails

Ventajas:

- ✓ Menor curva de aprendizaje, productividad asombrosa.
- ✓ El lenguaje Groovy es fácil de aprender para los desarrolladores Java.
- ✓ Usa Spring e Hibernate como base.

Desventajas:

- ✓ Rendimiento menor que los frameworks más "crudos".
- ✓ Puede ser difícil de "vender" a aquellos que les gusta Java.
- ✓ Virtualmente desconocido fuera de los blogs.

Ruby on Rails (ROR)

Ventajas:

- ✓ Alta productividad para desarrollar aplicaciones de tipo CRUD.
- ✓ Solución todo en 1. Desde la presentación hasta la persistencia.
- ✓ Es posible mantener ambientes separados de prueba y producción.
- ✓ No necesita configuración (al menos no mucha).
- ✓ Gran aceptación en la comunidad de desarrolladores.

Desventajas:

- ✓ Aún no existe constancia de aplicaciones de gran envergadura desarrolladas con este framework más allá de varias aplicaciones Web masivas.
- ✓ Utiliza lenguaje interpretado y débilmente tipado, difícil de depurar.

La evolución de los Frameworks Web marca una tendencia clara a la abstracción del protocolo en el que se sustenta (HTTP) para beneficiarse de los modelos basados en controles y componentes que tanto éxito tuvieron en el ámbito del escritorio. Las diferentes aproximaciones de los frameworks muestran que, en la mayoría de éstos, se atacan los mismos problemas: navegación, internacionalización, manejo de errores, validación de entradas, escalabilidad, etc. Los más antiguos como Struts permiten un manejo más directo de los datos que se procesan mientras que los más modernos como JSF, ASP.NET, Codeigniter o Tapestry buscan la abstracción casi total del protocolo pero a cambio de generar modelos de componentes fácilmente extensibles y Orientados a Eventos.

Cada Framework apunta a solucionar objetivos generales pero sale beneficiado cuando ataca problemas particulares. Por ejemplo, mientras que ROR permite crear aplicaciones de manera rapidísima pero a costa de sencillez extrema y falta de flexibilidad (si se empieza a configurar y redefinir, ya no es tan rápido el desarrollo), otros como Cocoon y JSF fueron diseñados para no depender del dispositivo de presentación, lo que otorga varios puntos a las aplicaciones que buscan verse bien en dispositivos móviles como celulares y PDAs. ASP.NET por su parte permite una productividad interesante a costa de limitar el trabajo a la herramienta que provee su fabricante y quedando atado a las futuras decisiones de la compañía.

Codeigniter es desarrollado por Ellis Lab **[31]** es un web framework para personas que construye sitios Web usando PHP. El objetivo es habilitar el desarrollo de proyectos mucho más rápido de lo que podría si se escribiese código desde cero, a través de un rico conjunto

de librerías para tareas comúnmente necesarias, tanto como una simple interface y estructura lógica para acceder a estas librerías. Permite concentrarse creativamente en su proyecto minimizando el volumen de código necesario para una tarea determinada.

Sus principales características:

Versatilidad: Quizás la característica principal de CodeIgniter, en comparación con otros frameworks PHP, CodeIgniter es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.

Compatibilidad: Es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos. Por supuesto, funciona correctamente también en PHP 5.

Facilidad de instalación: No es necesario más que una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.

Flexibilidad: Es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces nos las podemos saltar para trabajar como más a gusto encontremos. Algunos módulos como el uso de plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.

Ligereza: El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.

Documentación tutorializada: La documentación de CodeIgniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial. Esto no facilita mucho la referencia rápida, cuando ya sabemos acerca del framework y queremos consultar sobre una función o un método en concreto, pero para iniciarnos sin duda se agradece mucho.

¿Por qué Codeigniter?

- ✓ Es muy liviano. La ultima versión, la 1.7.2 apenas supera 1Mb.
- ✓ Ofrece un gran rendimiento.
- ✓ Ofrece compatibilidad con varias versiones de PHP. Concretamente desde la 4.3.2 a la 5.3.0
- ✓ Apenas requiere configuración.
- ✓ No requiere de línea de comandos para generar las aplicaciones.
- ✓ No sigue una línea de reglas estricta. Podemos adaptarlo a nuestras necesidades.
- ✓ No es una gran librería al estilo PEAR.
- ✓ No requiere aprender un lenguaje de platillas. Es opcional.
- ✓ Genera SEO urls para los buscadores.
- ✓ Tiene una documentación muy legible.

En la conferencia DrupalCom el mismísimo Rasmus Lerdorf **[33]**, creador del lenguaje PHP, tuvo unas interesantes opiniones acerca de la performance de los *frameworks* en general:

Para Lerdorf los frameworks actuales para PHP son muy pobres en cuanto a la performace. Y no sólo eso, su actitud de "hace lo todo" generalmente conduce a los desarrolladores por el camino equivocado porque no usan lo que es mejor para hacer el trabajo.

Como demostración, Lerdorf midió la cantidad de respuestas por segundo para imprimir simplemente en mensaje "Hola Mundo" que ofrecían algunos frameworks sin hacer ninguna llamada a la base de datos. La más rápida consiguió 120 por segundo, y la más lenta 8 por segundo; comparadas con las más de 600 que sirve Apache con un archivo HTML.

Rasmus sí puntualizó que de todos le gustaba CodeIgniter porque es rápido, pequeño y el que menos se parece a un framework.

1.4.5 Sistemas Gestores de Bases de Datos más utilizados

ORACLE

Oracle es básicamente una herramienta cliente/servidor para la gestión de base de datos, es el Sistema Manejador de Base de Datos Relacional (RDBMS) más usado y más vendido en el mundo, aunque la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general **[17]**. Es un RDBMS que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. Es el conjunto de datos que proporciona la capacidad de almacenar y acudir a estos de forma recurrente con un modelo definido como relacional. Además es una suite (conjunto de programas relacionados para un dominio específico) de productos que ofrece una gran variedad de herramientas.

Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitecturas de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. Esto es porque más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

PL/SQL: es un lenguaje que implementa Oracle, portable, procedural y de transacción muy potente y de fácil manejo.

A partir de la versión 7 de *Oracle* el usuario puede almacenar, en forma independiente, sus funciones y procedimientos sin tener que escribirlos repetidamente para cada forma, y pudiendo compilarlos independientemente de las formas que lo usen. Pero, además, las funciones y procedimientos se pueden agrupar en un paquete para compartir definiciones, variables globales, constantes, cursores y excepciones, así como garantizar y revocar los permisos a nivel de paquete. En el caso que sea necesario modificar el contenido del paquete, como el mismo se encuentra almacenado separadamente, no es necesario recompilar nada que use ese paquete, lo que facilita la gestión y mantenimiento de todos los procedimientos almacenados como una sola entidad para una determinada aplicación además existe un nuevo tipo de disparador llamado *de base de datos*, que es un procedimiento asociado a una tabla que se activa cuando se produce un suceso que afecta a

esa tabla. Su uso más común consiste en la definición de restricciones complejas de integridad.

SQLReport de Oracle realiza de forma flexible, sencilla y eficiente la creación de reportes, informes o listados permitiendo, entre otras facilidades, la visualización previa por pantalla con una gran variedad en estilos de presentación.

SQLSERVER

Microsoft SQL Server constituye un lanzamiento determinante para los productos de bases de datos de Microsoft, continuando con la base sólida establecida por SQL Server 6.5. Como la mejor base de datos para Windows NT, SQL Server es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software (ISVs) que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos **[18]**.

El SQL Server 7.0 es una versión de rastro de baja memoria con capacidades de replicación de multi-sitio. Las otras características tales como bloqueo a nivel de línea dinámica, el paralelismo intra-query, query distribuido, y mejoras para las bases de datos muy grandes (VLDB) hacen que el SQL Server 7.0 sea la elección ideal para sistemas OLTP de alta tecnología y Sistemas de Data Warehousing.

Las innovaciones del producto en SQL Server mejoran el proceso de Data Warehousing: Servicios de Transformación de Datos; manejo mejorado de las consultas complejas y bases de datos muy grandes; procesamiento analítico en línea e integrado; y el Microsoft Repository. Otro componente esencial es el soporte extenso para integración de terceros. Las áreas de liderazgo e innovación en el Microsoft SQL Server 7.0 incluyen:

- ✓ La primera BBDD en escalar desde la computadora portátil hasta la empresa utilizando la misma base de código y ofrecer el 100% de compatibilidad de código.
- ✓ La primera BBDD en soportar la auto-configuración y auto-sintonización.
- ✓ Primera BBDD con OLAP integrado.
- ✓ La primera BBDD con Servicios de Transformación de Datos integrado.
- ✓ El Data Warehousing Framework constituye el primer enfoque comprehensivo al problema de metadatos.

- ✓ La primera BBDD en proveer administración de multi-servidor para cientos de servidores.
- ✓ La más amplia gama de opciones de replicación de cualquier BBDD.
- ✓ La mejor integración con Windows NT Server.
- ✓ La mejor integración con Microsoft Transaction Server.
- ✓ Lanzamientos SQL Server Recientes.

HSQLDB (Hypersonic SQL Data Base) es un SGBD libre escrito en Java. Es un servidor de BBDD sencillo y a la vez poderoso. HSQLDB tiene varios modos de operación y características que lo hacen muy recomendado y le permiten ser usado en muchos escenarios distintos. Los niveles de uso de memoria, la velocidad y la accesibilidad por diferentes aplicaciones está influenciado por la manera en cómo HSQLDB ha sido desarrollado **[19]**.

Algunas características de HSQLDB son las siguientes:

- ✓ Completamente en Java.
- ✓ Es un SGBDR.
- ✓ Tiempo de arranque mínimo y gran velocidad en las operaciones de selección, inserción, borrado y actualización.
- ✓ Sintaxis SQL estándar.
- ✓ Integridad referencial (claves foráneas).
- ✓ Triggers.
- ✓ Tablas en disco de hasta 8 Gb.

Para decidir si usar HSQLDB como un servidor separado de la aplicación en la misma PC o en una diferente, o como una BBDD interna a la aplicación debe considerarse:

- ✓ Cuando HSQLDB se usa como un servidor separado de la aplicación en otra PC, debe ser aislado de las fallas de hardware y los colapsos de la PC que ejecuta la aplicación.
- ✓ Cuando HSQLDB se usa como servidor en la misma PC donde está la aplicación, debe aislarse de los fallos de la aplicación y escapes de memorias.
- ✓ Las conexiones al servidor son más lentas que las conexiones internas debido al alto en el flujo de datos para cada llamada JDBC.

En todos los modos de ejecución (servidor o in-process) HSQLDB soporta múltiples conexiones a la BBDD. En modo In-process (standalone) soporta conexiones para el cliente en la misma JVM, mientras que en modo servidor soporta conexiones sobre la red de muchos clientes diferentes.

Una aplicación que no sea multi-hilo y transaccional, tal como una aplicación para guardar acciones de login y logout, no necesita más que una conexión. La conexión puede permanecer abierta indefinidamente y reabierta sólo cuando se ha cerrado debido a problemas de red **[19]**.

Cuando se usa una BBDD en modo servidor (y en algunas extensiones en modo in-process), debe tenerse cuidado para evitar crear y cerrar conexiones JDBC muy frecuentemente. No hacer esto resultará en intentos de conexiones fallidos cuando la aplicación está bajo una sobrecarga.

Esta BBDD cumple algunas condiciones de las mencionadas, pero no todas las que necesitan. Es cierto que presenta muy buena velocidad sobre algunas de las operaciones fundamentales en BBDD, pero en ocasiones puede ser lento el acceso a datos con consultas complejas, siendo esta es una característica fundamental que hoy en día se considera importante.

POSTGRESQL

PostgreSQL [20] es un SGBDOR basado en POSTGRES Versión 4.2, desarrollado en la Universidad de California en el Departamento de Ciencias de la Computación de Berkeley. PostgreSQL es el descendiente *Open Source*[3] del original código de Berkeley, el mismo da soporte a gran parte del estándar SQL 2003 y ofrece muchas funcionalidades modernas, como son: consultas complejas, llaves extranjeras, disparadores, vistas, integridad transaccional, control concurrente multiversión. PostgreSQL puede ser extendido por el usuario de varias maneras, como puede ser definiendo nuevos: tipos de datos, funciones, operadores, funciones agregadas, métodos indexados, lenguajes procedurales. Además de esto, debido a la licencia bajo la cual es distribuido, PostgreSQL puede ser usado,

[3] De código abierto. Filosofía de desarrollo de software lidera por *Free Software Foundation* **[9]** que pone públicamente el código fuente la aplicación, el cual, al igual que la aplicación, puede ser usado, modificado, redistribuido, y en todo los casos libremente.

modificado y distribuido por cualquiera sin necesidad del pago de licencias independientemente del objetivo con el que sea usado, bien sea para uso privado, comercial o académico.

Postgres95 fue programado completamente en ANSI C y su tamaño se redujo en un 25% con respecto a su predecesor, además incluía muchos cambios internos que facilitaban su mantenimiento y aumentaban su rendimiento entre un 30% y 50% con respecto a POSTGRES. En 1996 el nombre de Postgre95 ya no era del agrado de los desarrolladores y se decidió cambiar el nombre a PostgreSQL, el cual representaba la relación entre la versión y el soporte de las nuevas versiones al lenguaje de consultas de bases de datos estándar **[20]**.

MYSQL

MySQL es un SGBDR, multi-hilo y multiusuario. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL (General Public Licence), pero empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado.

El servidor de BBDD MySQL es rápido, confiable y fácil de usar pues está basado en SQL. Fue originalmente desarrollado para manipular grandes BBDD de manera más rápida que las soluciones que existían por entonces y se ha utilizado con mucho éxito en muchos ambientes de producción por muchos años. Aunque está en constante desarrollo, MySQL ofrece una gran cantidad de funciones y uso. Su conectividad, velocidad y seguridad hacen de MySQL un servidor de BBDD muy recomendado.

MySQL funciona en aplicaciones tipo cliente/servidor, que consiste en un servidor SQL multi-hilo que soporta diferentes backends, varios programas clientes y librerías, herramientas administrativas y un amplio espectro de APIs. MySQL también se usa como

una librería embebida multi-hilo que se puede enlazar en una aplicación para obtener un producto más pequeño, rápido y fácil de utilizar.

Estas son las características más importantes de MySQL **[21]**:

- ✓ Trabaja en múltiples plataformas.
- ✓ Puede fácilmente utilizar múltiples CPUs si están disponibles.
- ✓ Provee motores de almacenamientos transaccional y no transaccional.
- ✓ Un sistema de localización de memoria muy rápido.
- ✓ Joins muy rápidos usando un one-sweep multi-join optimizado.
- ✓ Tablas hash In-memory, las cuales son usadas como tablas temporales.
- ✓ El servidor está disponible como un programa separado para usarse en aplicaciones cliente/servidor o como una librería para ser embebida en las aplicaciones.

MySQL es un SGBD, como ya se mencionó, rápida, segura y confiable, que reúne muchas de las características que lo harían muy recomendado para muchos tipos de aplicaciones, pero existe el conflicto de la licencia comercial. MySQL ofrece una licencia comercial que permite usar todo o parte del código fuente de MySQL en sus aplicaciones **[21]**.

¿Por que MYSQL?

MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos relativamente grandes como pequeños. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a bases de datos en Internet.

Razones

- ✓ Actualmente MySQL ha sido comprada por SUN lo que en un futuro lo convertirá en un SGBD más robusto debido a que esta prestigiosa empresa se encarga de darle soporte técnico, además de que sigue siendo de adquisición gratuita.

- ✓ El PHP maneja más fácil al MySQL que al SQL Server y al PostGreSQL debido a la gran cantidad de funciones que tiene explícitas.

- ✓ El MySQL además de ser multiplataforma, es muy veloz.
- ✓ La sintaxis del MySQL es legible.

Familiarización con el Gestor MySQL.

1.4.6 Servidor Web

Apache

Servidor Web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Apache es una muestra, al igual que el sistema operativo Linux (un Unix desarrollado inicialmente para PC), de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar.

La licencia Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia te permite hacer lo que quieras con el código fuente (incluso forks y productos propietarios) siempre que les reconozcas su trabajo.

¿Por qué Apache?

Luego de analizadas las características se decide usar el Apache como servidor Web, por las siguientes razones.

- ✓ Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- ✓ Es una tecnología gratuita de código abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto.
- ✓ Es un servidor altamente configurable de diseño modular.
- ✓ Trabaja con gran cantidad de lenguajes como Perl, PHP y otros lenguajes de script.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

- ✓ Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.

Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

1.4.7 Otras Herramientas Necesarias.

Como vamos a utilizar una aplicación Web para confeccionar la propuesta de este trabajo, se hace necesario tener en cuenta la utilización de un editor de páginas Web. Por la familiaridad, popularidad y calidad probada se seleccionó el Zend Studio. Con esa herramienta podremos desarrollar cualquier sitio Web personal con características de sitio profesional y utilizar casi todos los recursos de la Web, así como realizar aplicaciones que se ejecuten en servidor y vinculaciones dinámicas de datos, como es nuestro caso; además de contar con un soporte para aplicaciones PHP y utilización de bases MySQL

1.4.8 Metodología de Ingeniería de Software

¿Qué metodología utilizar para el desarrollo de un software? Es una pregunta que la mayoría de los desarrolladores de software se han hecho. Esta pregunta se torna muy importante si se desea tener un plano en el cual apoyarse durante el proceso de producción de un software. Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se cuenta con una metodología de por medio, lo que se obtiene es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Las metodologías de Ingeniería del Software han estado presentes durante mucho tiempo. No se han distinguido precisamente por ser muy exitosas. Aún menos por su popularidad. La crítica más frecuente a estas metodologías es que son burocráticas.

Hay tanto que hacer para seguir la metodología que el ritmo entero del desarrollo se retarda **[22]**.

Como una reacción a estas metodologías, un nuevo grupo de metodologías ha surgido en los últimos años. Durante algún tiempo se conocían como metodologías ligeras, pero el término aceptado ahora es metodologías ágiles. Para mucha gente el encanto de estas

metodologías ágiles es su reacción ante la burocracia de las metodologías monumentales. Estos nuevos métodos buscan un justo medio entre ningún proceso y demasiado proceso, proporcionando simplemente suficiente proceso para que el esfuerzo valga la pena.

El resultado de todo esto es que las metodologías ágiles cambian significativamente algunos de los énfasis de las metodologías tradicionales como Unified Process (UP) o Rational Unified Process (RUP). La diferencia inmediata es que son menos orientados al documento, exigiendo una cantidad más pequeña de documentación para una tarea dada. De muchas maneras son más bien orientados al código: siguiendo un camino que dice que la parte importante de la documentación es el código fuente.

De forma general las metodologías ágiles presentan las siguientes características **[22]**:

- ✓ Los métodos ágiles son adaptables en lugar de predictivos. Las metodologías tradicionales tienden a intentar planear una parte grande del proceso del software en gran detalle para un plazo largo de tiempo; esto funciona bien hasta que las cosas cambian. Así que su naturaleza es resistirse al cambio. Para los métodos ágiles, no obstante, el cambio es bienvenido. Intentan ser procesos que se adaptan y crecen en el cambio, incluso al punto de cambiarse ellos mismos.
- ✓ Los métodos ágiles son orientados a la gente y no orientados al proceso. La meta de las metodologías tradicionales es definir un proceso que funcionará bien con cualquiera que lo use. Las metodologías ágiles afirman que ningún proceso podrá nunca maquillar las habilidades del equipo de desarrollo, de modo que el papel del proceso es apoyar al equipo de desarrollo en su trabajo. Explícitamente puntualizan el trabajar a favor de la naturaleza humana en lugar de en su contra y enfatizan que el desarrollo de software debe ser una actividad agradable.

Varias metodologías encajan bajo el estandarte de ágil, pero la que ha recibido más atención es la metodología XP.

A continuación se describen las metodologías que fueron analizadas en este trabajo, siendo la última la seleccionada.

1.4.8.1 RUP

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

- ✓ Inicio, el objetivo en esta etapa es determinar la visión del proyecto.
- ✓ Elaboración, en esta etapa el objetivo es determinar la arquitectura óptima.
- ✓ Construcción, en esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- ✓ Transmisión, el objetivo es llegar a obtener el realce del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- ✓ Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- ✓ Requerimientos: Trasladando las necesidades del negocio a un sistema automatizado.
- ✓ Análisis y Diseño: Trasladando los requerimientos dentro de la arquitectura de software.
- ✓ Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- ✓ Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte

- ✓ Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- ✓ Administrando el proyecto: Administrando horarios y recursos.
- ✓ Ambiente: Administrando el ambiente de desarrollo.
- ✓ Distribución: Hacer todo lo necesario para la salida del proyecto.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un “que” entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada “que” entregable o en cada iteración.

Los elementos del RUP son:

- ✓ Actividades, Son los procesos que se llegan a determinar en cada iteración.
- ✓ Trabajadores, Vienen hacer las personas o involucrados en cada proceso.
- ✓ Artefactos, Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software **[23]**.

1.4.8.2 XP

Extreme Programming (XP) es una de las metodologías de desarrollo de software más exitosa en la actualidad utilizada para proyectos de corto plazo y corto equipo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Características de XP:

- ✓ Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que se pueden hacer pruebas que puedan prever las fallas que pudieran ocurrir en el futuro.
- ✓ Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- ✓ Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué propone XP?

- ✓ Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- ✓ El manejo del cambio se convierte en parte sustantiva del proceso.
- ✓ El costo del cambio no depende de la fase o etapa.
- ✓ No introduce funcionalidades antes que sean necesarias.
- ✓ El cliente o el usuario se convierte en miembro del equipo.

Derechos del Cliente:

- ✓ Decidir qué se implementa.
- ✓ Saber el estado real y el progreso del proyecto.
- ✓ Añadir, cambiar o quitar requerimientos en cualquier momento.
- ✓ Obtener lo máximo de cada semana de trabajo.
- ✓ Obtener un sistema funcionando cada 3 ó 4 meses.

Derechos del Desarrollador

- ✓ Decidir cómo se implementan los procesos.
- ✓ Crear el sistema con la mejor calidad posible.
- ✓ Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- ✓ Estimar el esfuerzo para implementar el sistema.
- ✓ Cambiar los requerimientos en sobre la base de los nuevos descubrimientos.

Lo fundamental en este tipo de metodología es:

- ✓ La comunicación, entre los usuarios y los desarrolladores.
- ✓ La simplicidad, al desarrollar y codificar los módulos del sistema.
- ✓ La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

1.4.8.3 ICONIX

ICONIX se ubica entre el gran tamaño y complejidad de RUP (Rational Unified Process – Proceso Unificado de Rational) y el pequeño y compacto de XP (eXtreme Programming) **[24]**. Al igual que RUP, ICONIX es una metodología conducida por casos de uso, pero no incorpora tantos artefactos UML. Es una metodología relativamente pequeña al igual que XP, pero no descarta las etapas de análisis y diseño. Utiliza UML en sus etapas, de modo que se pueden seguir los requerimientos y adaptarse a nuevos cambios.

El objetivo de esta metodología es buscar un subconjunto mínimo, pero suficiente de artefactos UML para hacer un buen trabajo de Ingeniería del Software en poco tiempo.

Esquema General

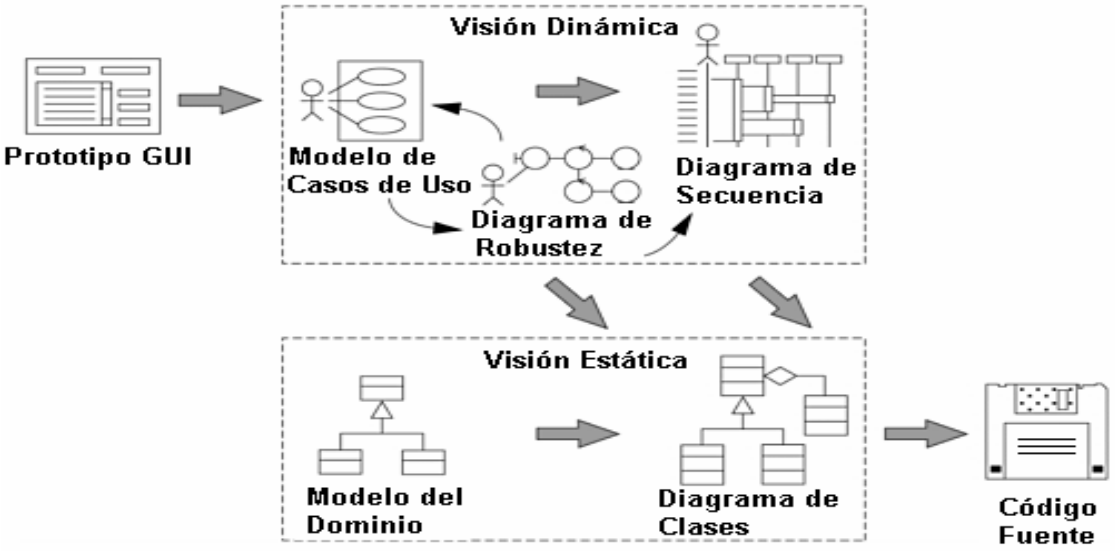


Figura 1.2 *Esquema General de ICONIX.*

1.4.6.5 Metodología seleccionada

Actualmente existen metodologías que permiten desarrollar software de superior calidad, debido a la facilidad de control que éstas proporcionan, aparejado al aumento de la productividad de los ingenieros y la posibilidad de concebir inicialmente las bases para el desarrollo del software y su éxito en el tiempo y costo fijados.

Cada metodología, a pesar de suministrar los aspectos antes mencionados, posee características peculiares que facilitan la selección de una de éstas a partir de las necesidades de la organización y las características específicas del proyecto a desarrollar.

Teniendo en cuenta lo anteriormente dicho, se determinó escoger para el desarrollo del proyecto la metodología **ICONIX**.

De la misma manera que RUP, el proceso de *ICONIX* maneja casos de uso pero más ligero que RUP. También es relativamente pequeño y firme, como XP, pero no desecha el análisis y diseño que hace dicha metodología.

Este proceso también hace uso aerodinámico del UML mientras guarda un enfoque afilado en el seguimiento de requisitos, por lo que el proceso se queda igual a la visión original de

Jacobson del" manejo de casos de uso; esto produce un resultado concreto, específico y casos de uso fácilmente entendibles, que un equipo de un proyecto puede usar para conducir el esfuerzo hacia un desarrollo real.

ICONIX propone un enfoque aerodinámico al desarrollo del software, que incluye un juego mínimo de diagramas de UML y algunas valiosas técnicas que se toman de los casos del uso para codificar rápida y eficazmente. El enfoque es flexible y abierto; siempre se puede seleccionar de los otros aspectos del UML para complementar los materiales básicos.

La principal diferencia de *ICONIX* con respecto a otras metodologías es su uso de análisis de robustez, un método para llenar la brecha entre el análisis y el diseño. El Análisis de robustez reduce la ambigüedad en las descripciones de caso de uso, asegurando que sean escritos en el contexto de un modelo de dominio acompañando. Este proceso hace los casos de uso mucho más fácil de diseñar, evaluar y calcular.

Existen tres rasgos significantes de este enfoque:

Primero, es reiterativo e incremental. Las iteraciones múltiples ocurren entre el desarrollo del modelo del dominio e identificar y analizar los casos de uso. Otras iteraciones existen también, como los procesos del equipo a través del ciclo de vida. El modelo estático se refina incrementalmente durante las iteraciones sucesivas a través del modelo dinámico (compuesto de casos de uso, análisis de robustez y el diagrama de secuencia).Es de señalar, que el acercamiento no requiere hitos formales y la teneduría de muchos libros; más bien, los esfuerzos de refinamiento producen los hitos naturales como el equipo del proyecto que gana conocimiento y experiencia.

Segundo, el enfoque ofrece un alto grado de seguimiento. Por el camino, a cada paso usted consultará de alguna manera los requisitos anteriores. Nunca hay un punto en que el proceso le permita desviarse lejos de las necesidades del usuario. Seguimiento se refiere también al hecho que usted puede seguir los objetos paso a paso como el análisis dentro del diseño.

Tercero, el enfoque ofrece uso aerodinámico del UML. Los pasos que se describirán en los siguientes temas representan un mínimo del acercamiento, ellos comprenden el juego mínimo de pasos que se han encontrado para ser necesarios y suficiente en el desarrollo de

un proyecto Orientado a Objetos exitoso. Enfocando en un subconjunto del grande y pesado UML, un equipo del proyecto también puede dirigirse fuera de "la parálisis del análisis".

La solución de *ICONIX* incluye un ancho rango de ofrecimientos de servicios de negocios. Las soluciones de negocios de extremo a extremo se concentran en los servicios en tres áreas primarias, con la estrategia y planeación recubriendo cada área. La especialización equilibrada en las tres áreas (la experiencia del usuario, funcionalidad comercial, e infraestructura) contribuye al éxito de las soluciones que se entrega a los clientes

Las características antes mencionadas garantizan que la metodología *ICONIX* sea la apropiada para llevar a cabo el proyecto que se emprende.

Conclusiones del Capítulo

En este capítulo se abordaron elementos necesarios para la compresión y fundamentación de la solución propuesta. Las tendencias y tecnologías actuales relacionadas con el tema. Se hizo una valoración del lenguaje de programación, el sistema gestor de bases de datos, y la metodología de desarrollo. Una vez conocidas las herramientas y conceptos a usar se puede proseguir con el diseño y la construcción de la solución propuesta.

CAPITULO 2

DESCRIPCIÓN Y ELABORACIÓN DE LA SOLUCIÓN PROPUESTA.

2.1 Introducción

El objetivo de este capítulo es contribuir a una correcta comprensión de los conceptos que se encuentran relacionados en el dominio de los Sistemas Web, por ser éste el dominio donde se enmarca el sitio Web propuesto: Sistema de Control de los Indicadores de Gestión del Cuadro de Mando Integral. Además en este capítulo figuran otros aspectos como el modelo de casos de uso y de análisis y el estudio de sostenibilidad según el impacto socio-humanista, administrativo, tecnológico y ambiental.

El proceso de *ICONIX* propone para estos casos realizar un modelo del dominio que no es más que una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés, por lo que permite mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema. Tal modelo no incluye las responsabilidades que llevan a cabo las personas, sólo describe el contenido de información de la organización.

A continuación se muestra el marco conceptual sobre el que se construye el modelo del dominio de la solución propuesta.

2.2 Análisis del Sistema

2.2.1 Modelo del Dominio

En la Tabla 1.1 se muestran los conceptos o clases que se consideran más importantes para el desarrollo de un Sistema de Control de los Indicadores de Gestión de los Cuadro de Mando Integral.

Concepto	Definición
Indicador	Es un medio para medir lo que realmente sucede en comparación con lo que se ha planificado en términos de calidad, cantidad y puntualidad.
Áreas	Está integrada por 2 grupos, las Áreas de Resultados Claves (ARC) y las Otras Áreas. Su principal diferencia radica en el grado de incidencia en la obtención de los objetivos.
Especialistas	Personas que utilizan o trabajan con algún objeto, son los encargados de interactuar con el sistema.
Proyección Histórica	Es la presentación gráfica del comportamiento a largo plazo del indicador.

Tabla 2-1 *Definición de conceptos del marco de trabajo del Sistema de Control de los Indicadores de Gestión de los Cuadro de Mando Integral.*

Un **indicador** pretende sintetizar la información que proporcionan los diversos parámetros o variables que afectan a la situación que se quiere analizar para poder comparar los distintos períodos. La comparación de mediciones permite ver la evolución en el tiempo y estudiar tendencias acerca de la situación que miden.

En las **Áreas** están los aspectos decisivos para alcanzar los factores claves de éxito en función de la satisfacción de las necesidades de los clientes y el cumplimiento del objeto social de la empresa. Establecen los lugares donde se van a situar los recursos y esfuerzos individuales y colectivos, estos aspectos principalmente pertenecen a las Áreas de Resultados Claves (ARC). Los aspectos que inciden de manera indirecta, producen actividades que generan resultados en un segundo plano. Contribuyendo al buen desarrollo de los procesos de realización aportándoles los recursos necesarios Aunque no crean valor directamente perceptible por el cliente, son necesarios para el funcionamiento permanente del organismo y a su perennidad (Otras Áreas).

Los **Especialistas** no son más que los actores del sistema, o sea cualquiera que requiera los servicios del sistema. Es la persona a la que va destinada dicho producto una vez que ha

superado las fases de desarrollo correspondientes. El software se desarrolla pensando en la comodidad de los mismos, y por esto se presta especial interés y esfuerzo en conseguir una interfaz de usuario lo más clara y sencilla posible.

La **Proyección Histórica** es la representación gráfica del comportamiento a largo plazo del indicador en comparación con margen de error.

2.2.2 Diagrama de Clases del Modelo del Dominio

En la Figura 2-1 se muestra mediante un diagrama de clases las relaciones entre los conceptos definidos en el epígrafe 2.2.1.

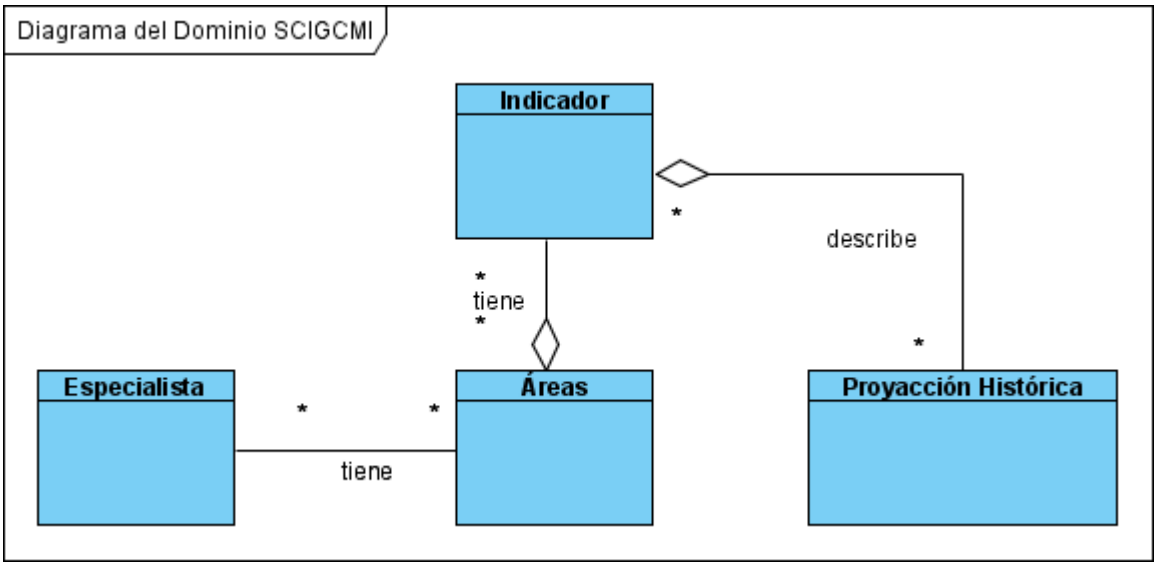


Figura 2-1. *Modelo del Dominio del Sistema de Control de los Indicadores de Gestión de los Cuadro de Mando Integral(SCIGCMI).*

2.2.3 Requerimientos del Sistema

La intención principal en la fase de requisitos es desarrollar el modelo del sistema a construir. Una forma adecuada de realizarlo es la creación de casos de uso del sistema a partir de requerimientos funcionales identificados, lo que proporciona un entendimiento común entre los actores y los analistas del sistema. Por otra parte, los no funcionales, a pesar de no asociarse a ningún caso de uso en concreto, son comunes e impactan en los casos de uso.

2.2.3.1 Requerimientos Funcionales

Los requerimientos funcionales determinados definen las funciones que el sistema será capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Describen además el comportamiento de las entradas para producir salidas y surgen de la razón fundamental de la existencia del producto. Estos requisitos se separan por subsistemas con el propósito de proporcionar a partir de esta fase en adelante una forma organizada para la comprensión de los mismos.

R1 Insertar Usuario.

R2 Modificar Usuario.

R3 Eliminar Usuario.

R4 Validar Usuario.

R5 Iniciar Sección

R6 Cerrar Sección

R7 Insertar Área.

R8 Eliminar Área.

R9 Actualizar Área

R10 Insertar datos del Indicador.

R11 Modificar datos Indicador.

R12 Visualizar datos Indicador.

R13 Mostrar Historial de Indicador

R14 Mostrar Historial Indicador Principal.

R15 Realizar reporte de incidencias.

2.2.3.2 Requerimientos no Funcionales

Los requerimientos no funcionales definidos limitan al sistema y se caracterizan por hacerlo más atractivo, usable, rápido, seguro y confiable. Se precisan con la intención de obtener el éxito, reflejada en la aceptación de los usuarios finales, así como el buen funcionamiento, la flexibilidad y escalabilidad que proporciona el mismo. Al mismo tiempo, se encuentran vinculados con los requerimientos funcionales a pesar de no alterar la funcionalidad del sistema.

Requerimientos de apariencia o interfaz externa

La interfaz debe ser agradable para conseguir la confianza de los usuarios en la utilización del sistema, utilizando recursos que atraigan la atención del usuario, para lograr una mejor concentración sin desviar demasiado su atención.

Portabilidad: Debe ser posible interactuar con él independientemente del sistema operativo con que se cuente porque se piensa que en un futuro en esta empresa se utilice el Sistema Operativo Linux ya que brinda las posibilidades de Software Libre.

Extensibilidad: La arquitectura del sistema debe ser tal que la incorporación de nuevas funcionalidades no implique cambios a las existentes.

Confidencialidad: Puesto que EMCOMED es una empresa que pertenece al MIMBAS y maneja la disponibilidad y reserva de medicamentos del país, y este software incluye información confidencial de la empresa y el sistema será accesible por todos, es preciso garantizar que el acceso sea controlado, cumpliendo las reglas de confidencialidad establecidas.

Diseño: Los colores que se pueden usar son el blanco, el azul y el rojo, no solo porque son los colores de la bandera sino porque también son los colores distintivos de la empresa.

Documentación: Debe suministrarse como complemento al sistema un manual de usuario.

Requerimientos mínimos de software y hardware del componente

Requerimientos de los servidores

Hardware:

- Procesador Pentium 3 (Genuino o Celerón).
- RAM, 256 MB mínimo, 512 MB recomendado.
- Tarjeta de red a 10 Mb
- 15 GB disponibles en servidor Web.

Software:

- Sistema Operativo: Windows 2003 Server, Linux Debian Sarge.
- Servidor Web. Tiene que soportar integración con PHP, se recomienda Apache.
- Servidor de Bases de Datos. MySQL 5.0.*.

Requerimientos de las estaciones clientes

Hardware: Se recomiendan como mínimo una Pentium 2

Software: Navegador Web. Se recomiendan Mozilla FireFox, Intenet Explorer 5 ó Superior, Opera

Sistema Operativo: Se recomiendan Linux y Windows XP.

Requerimientos de la red: Conectividad entre todas las estaciones clientes y el servidor Web, así como entre el servidor Web y el servidor de datos. Velocidad de 10 Mb.

2.2.4 Modelo de Casos de Uso del Sistema

En esta etapa se precisan cuáles son las funcionalidades del Sistema Control de los Indicadores de Gestión del Cuadro de Mando Integral. Para ello se utiliza el artefacto UML modelo de casos de uso.

Los casos de uso del sistema son una técnica para especificar el comportamiento del software y se parte de la identificación de los requerimientos del sistema. El modelo de casos de uso describe lo que hace el sistema para el usuario; la forma en que éste usa el sistema se representa con casos de uso, derivados de los requisitos funcionales que los relacionan. La representación de cada caso de uso facilita especificar la secuencia de acciones que el sistema puede llevar a cabo interactuando con el o los actores, incluyendo alternativas dentro de la secuencia.

Con el fin de representar lo antes dicho y ajustar mejor los casos de uso a los actores correspondientes, se realizaron por subsistemas los diagramas de casos de uso del sistema.

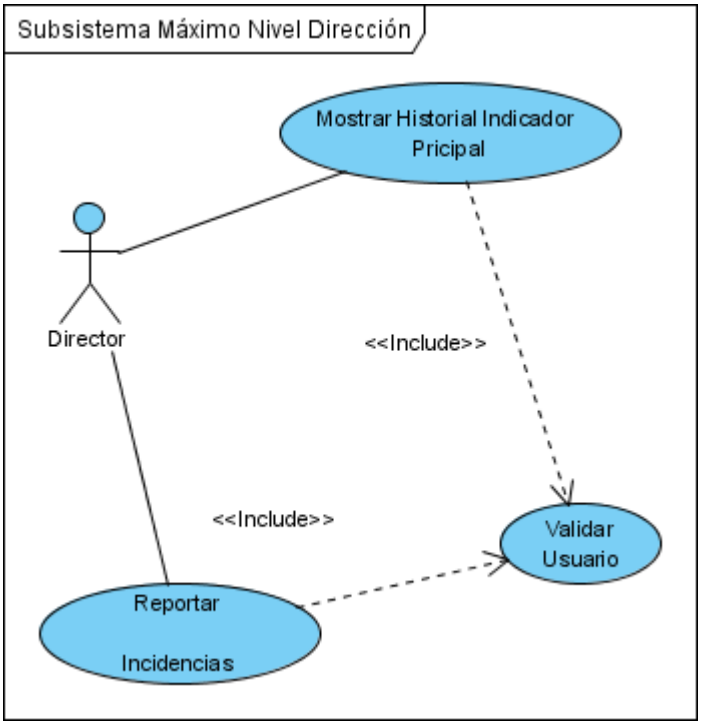


Figura 2-2. *Diagrama de Casos de Uso del subsistema Máximo Nivel de Dirección.*

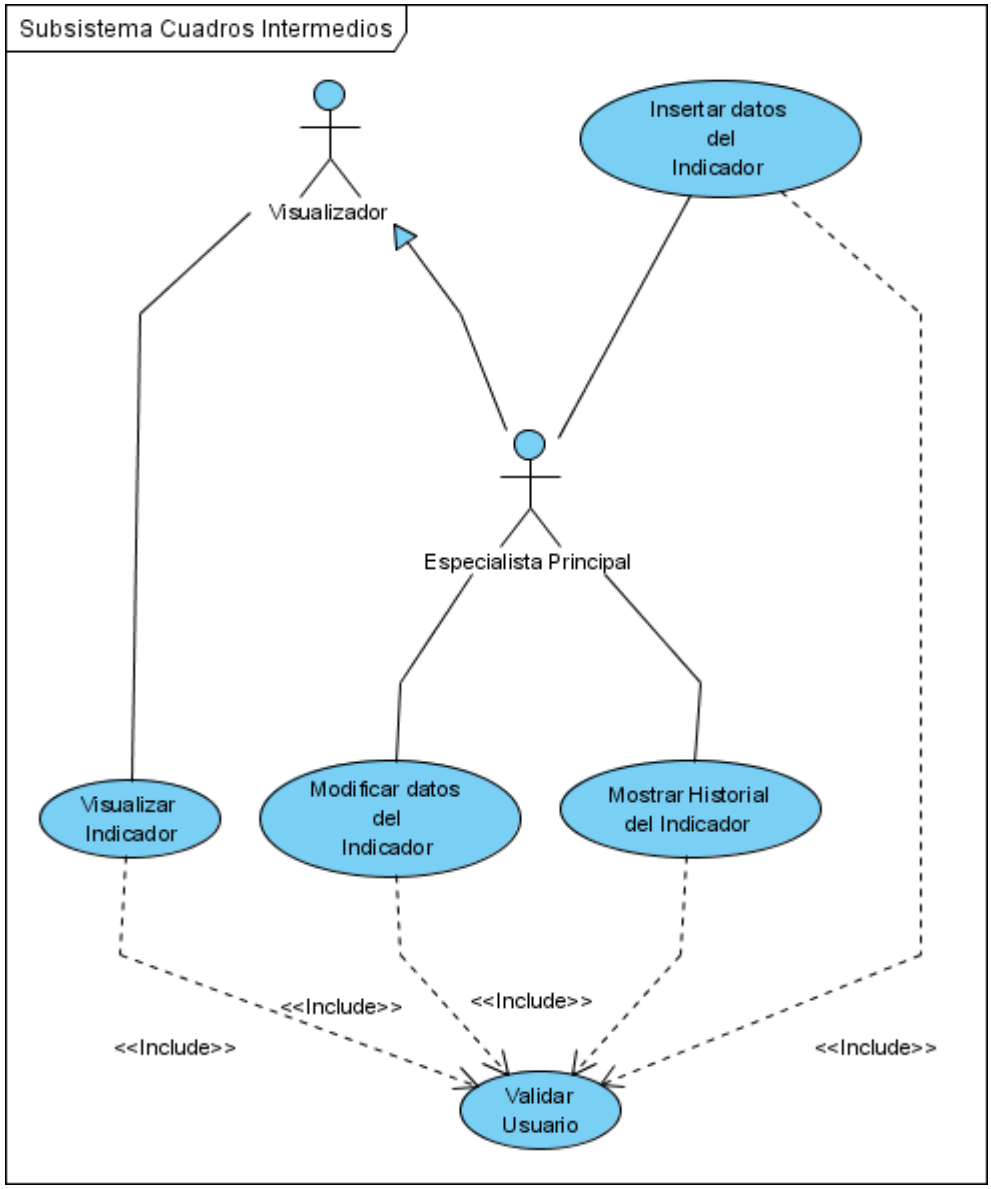


Figura 2-3. *Diagrama de Casos de Uso del subsistema cuadros intermedios.*

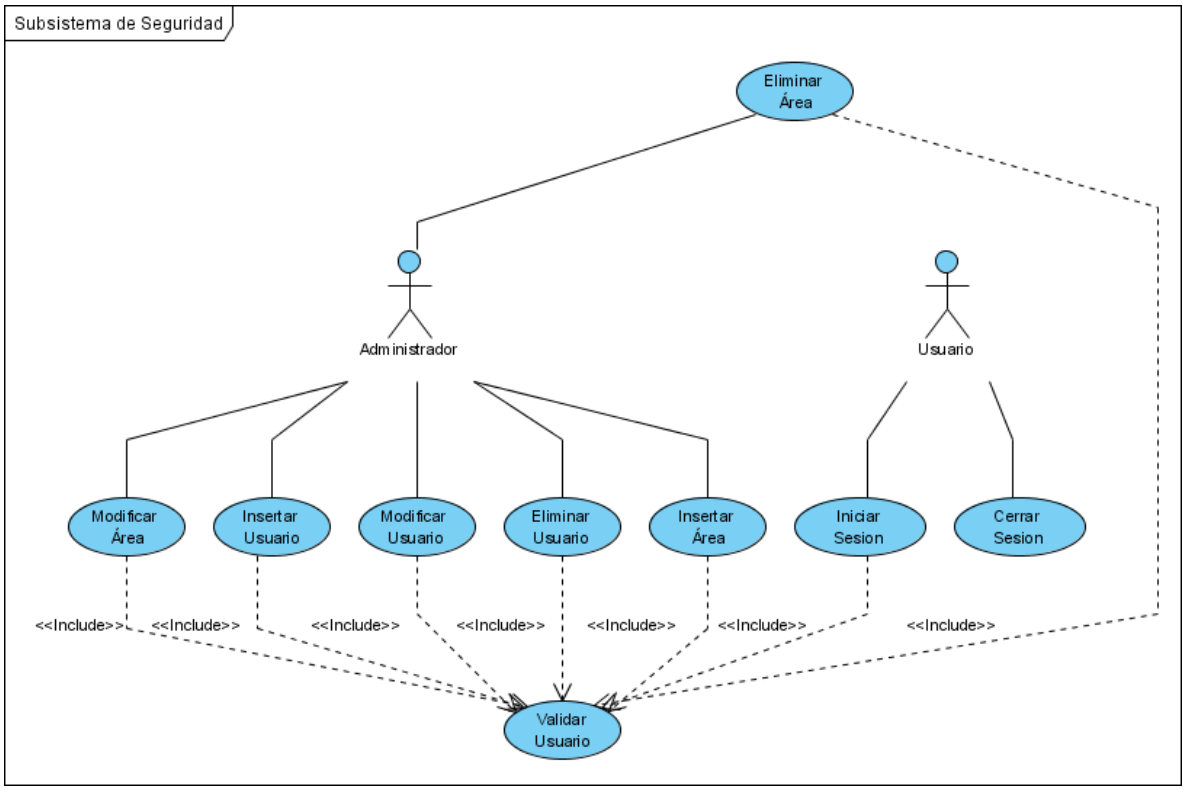


Figura 2-4. *Diagrama de Casos de Uso del subsistema de seguridad*

Con el objetivo de detallar las relaciones entre los casos de usos y los actores, y cómo juntos constituyen el modelo de casos de uso, se desarrollan las descripciones textuales de los principales casos de uso en el siguiente tópico para comprender lo que el sistema necesita hacer cuando interactúa con sus actores.

2.2.4.1 Descripción de los Principales Casos de Uso

Descripción del Actor

Actores	**Explicación**
Visualizador	Este actor puede ser cualquier trabajador de la empresa que desee conocer el estado del funcionamiento de la misma. A este actor se le mostrará solo para lectura el reporte del cuadro de mando integral.
Directora EMCOMED	Es el que va a realizar el reporte del CMI para conocer el estado de la empresa y tomar decisiones según los resultados del mismo.

Especialista Principal	Es el responsable de cada área, será el encargado de actualizar e insertar los indicadores.
Administrador	Es el responsable de insertar los usuarios y darle el nivel de acceso correspondiente y determinar los departamentos correspondientes a la entidad.
Usuario	Es una generalización de todos los otros actores, los únicos permisos que va a tener es iniciar y cerrar sesión.

Tabla 2.2 *Descripción de los actores del sistema.*

Descripción de los Casos de Uso.

Nombre del CU :	Realizar Reporte de Incidencias
Actores:	Directora (Inicia)
Propósito:	Realizar en un reporte de las Incidencias de los indicadores principales en el cumplimiento de la misión de la Organización
Resumen: El caso de uso inicia cuando la Directora decide realizar el reporte de las incidencias de los indicadores principales, operación que puede ejecutar es realizar reporte. El caso de Uso termina cuando la Directora decide salir del Sistema o hacer otras opciones.	
Requerimientos:	R13, validar usuario es un caso de uso incluido del caso de uso realizar reporte de incidencias
Precondiciones	La directora debe haberse autenticado como Directora satisfactoriamente.
Post Condiciones	Se realizo el reporte.

Tabla 2.3 *Descripción del CUS Realizar Reporte de Incidencias.*

Nombre del CU :	Mostrar Hist. Indicador
Actores:	Especialista Principal (Inicia)
Propósito:	Mostrar Historial Indicador
Resumen: Este caso de uso inicia cuando el Especialista Principal decide introducir nuevos datos del indicador y se actualice mostrará el comportamiento histórico. Las operaciones que puede realizar son Mostrar Hist. Indicador, pero para esto se hace una búsqueda por ID y después se mostrará el comportamiento de ese indicador. Este caso de uso termina cuando el Especialista decide escoger otra opción o salir del sistema.	
Requerimientos:	R7, validar usuario es un caso de uso incluido del caso de uso Mostrar Hist. Indicador.
Precondiciones	El Especialista Principal debe haberse autenticado como Especialista Principal satisfactoriamente.
Post Condiciones	Se Muestra el Hist. Del Indicador.

Tabla 2.4 *Descripción del C.U.S Mostrar Hist. Indicador*

Nombre del CU :	Insertar datos del Indicador
Actores:	Especialista Principal (Inicia)
Propósito:	Insertar Indicador
Resumen: Este caso de uso inicia cuando el Especialista Principal decide Insertar un indicador. Este caso de uso termina cuando el Especialista Principal decide escoger otra opción o salir del sistema.	
Requerimientos:	R7, validar usuario es un caso de uso incluido del caso de uso Insertar indicador.
Precondiciones	El Especialista Principal debe haberse autenticado como Especialista Principal satisfactoriamente.
Post Condiciones	Se insertó el indicador

Tabla 2.5 *Descripción del C.U.S Insertar Indicador*

Nombre del CU :	Modificar datos del Indicador
Actores:	Especialista Principal (Inicia)
Propósito:	Modificar Indicador
Resumen: Este caso de uso inicia cuando el Especialista Principal decide Modificar un indicador La operaciones que puede realizar son Modificar un Indicador,.Este caso de uso termina cuando el Especialista Principal decide escoger otra opción o salir del sistema.	
Requerimientos:	R7, validar usuario es un caso de uso incluido del caso de uso Modificar indicador.
Precondiciones	El Especialista Principal debe haberse autenticado como Especialista Principal satisfactoriamente.
Post Condiciones	Se Modificó el Indicador.

Tabla 2.6 *Descripción del C.U.S Modificar Indicador*

Nombre del CU :	Mostrar Hist. Indicador Principal
Actores:	Directora (Inicia)
Propósito:	Mostrar Historial Indicadores Principales
Resumen: Este caso de uso inicia cuando la Directora decide ver el comportamiento histórico de los indicadores principales que inciden en el cumplimiento de los Objetivos Estratégicos. Este caso de uso termina cuando el Especialista decide escoger otra opción o salir del sistema.	
Requerimientos:	R7, validar usuario es un caso de uso incluido del caso de uso Mostrar Hist. Indicadores principales.
Precondiciones	La Directora debe haberse autenticado satisfactoriamente.
Post Condiciones	Se Muestra el Hist. Del Indicadores principales.

Tabla 2.7 *Descripción del C.U.S Mostrar Hist. Indicadores Principales*

2.2.5.1 Descripción del Sistema Propuesto

Partiendo del modelo del dominio y los requerimientos funcionales capturados se desarrolla el artefacto modelo de casos de uso que incluye la identificación de los actores y casos de uso del sistema; los actores deben corresponderse con los usuarios finales y los casos de uso con las actividades a desarrollar por estos usuarios a través del sistema. El modelo de casos de uso sirve como acuerdo de requisitos entre los usuarios finales y desarrolladores, así como también proporciona la entrada al análisis, el diseño y las pruebas.

Cuando un usuario del sistema propuesto inicia sesión satisfactoriamente, se le muestra la página de inicio con la proyección histórica del indicador principal. Después puede acceder a las páginas del área que pertenece y a la de documentación donde podrá obtener información.

2.2.5.2 Entradas de Datos

El usuario puede acceder a las páginas de configuración de áreas y de usuarios facilitan una mejor interacción al administrador. Además el usuario especialista principal puede acceder a la página indicadores pudiendo actualizar o insertar los datos de los indicadores.

2.2.5.3 Reportes

A través de la página reportes se puede tener acceso a un reporte del indicador principal de la situación actual del Cuadro Mando Integral.

De forma general se puede apreciar que el sistema implementado cumple con los requerimientos enunciados en la metodología descrita en el capítulo anterior. Presenta, además, homogeneidad en los colores y líneas de diseño, ya que fue concebido siguiendo los patrones de diseño existentes. Los datos que entran los usuarios al sistema son validados antes de realizar el procesamiento de los mismos, indicándosele en el mensaje de advertencia correspondiente, que le facilitará la localización de la falta cometida.

En el sistema el empleo del mouse y el teclado es posible en la aplicación, y esto se le deja a gusto del usuario que lo va a utilizar; de hecho cada interfaz está concebida para que la accesibilidad a sus distintos componentes se realice utilizando cualquiera de estos dos recursos.

2.2.5.4 Diagramas de Paquetes

Partiendo de las flexibilidades brindadas por UML y la previa detección del tamaño del modelo de casos de uso del sistema, se consideró útil dividir en paquetes o subsistemas los requerimientos anteriormente definidos para tratar el tamaño del modelo, que de igual forma se manejará esta organización en las fases venideras**.**

(Ver **figura 2.5**).

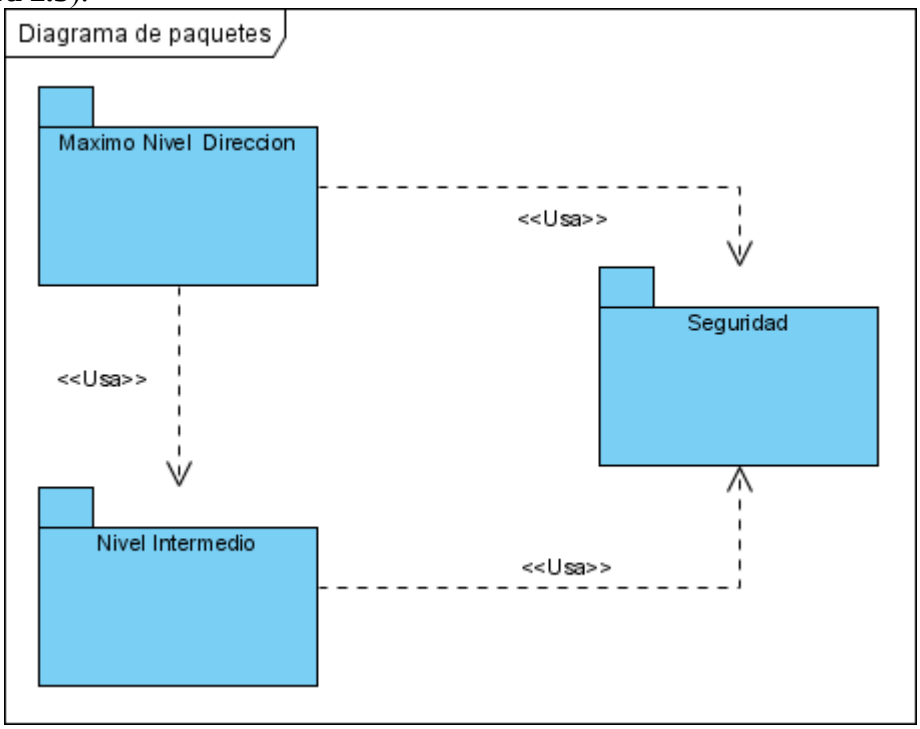


Figura 2.5 *Diagrama de Paquetes del Análisis.*

2.3 Valoración de Sostenibilidad según su impacto Social, Económico, Tecnológico y Ambiental.

2.3.1 Planificación

2.3.1.1 Características del proyecto

A diferencia de la fase del requisito de los sistemas de software, el análisis del dominio cubre una clase entera de problemas. El análisis del dominio intenta caracterizar el tamaño y la complejidad de un dominio elegido. Si el dominio es demasiado grande, es necesario dedicar gran cantidad de tiempo para recolectar y evaluar la información. Esto implica un largo proceso de desarrollo de un sitio Web y por tanto, el costo del mismo sería alto.

Por otra parte, si se elige un dominio demasiado estrecho, se reduce la aplicabilidad del sitio Web. Es importante tener en cuenta qué funcionalidades son necesarias, cuáles son las que realmente se necesitan en los requerimientos y las que son innecesarias o superfluas.

Con el tiempo el sitio Web llega a ser más maduro, cambia y se desarrolla por consiguiente. Este proceso puede representar la alteración de su arquitectura de configuración, pues aparecen nuevos requerimientos que no tenían soporte, y muchas otras nuevas causales.

2.3.2 Valoración de Sostenibilidad

Cuando se desarrolla un software, es de vital importancia tener en cuenta desde las primeras fases el impacto social, económico, tecnológico y ambiental que este tendrá. Se deben incluir en el análisis desde los procesos a informatizar, el personal vinculado en la actividad informática, el tiempo de uso de los recursos, hasta las características económicas, físicas, mentales y clínicas de las personas que van a interactuar con el producto informático.

2.3.2.1 Dimensión Administrativa

En la dimensión administrativa se valora si la solución planteada ahorra recursos, se tienen presente los gastos implicados para desarrollarla e implantarla, la calidad de la producción y los servicios, así como otros aspectos que garanticen la sostenibilidad administrativa de la solución.

Para el desarrollo del sistema como solución propuesta, se recurrió al Modelo Constructivo de Costos (COCOMO II, por sus siglas en inglés) para realizar el análisis de factibilidad.

COCOMO II (*CONSTRUCTIVE COST MODEL*, Modelo constructivo de costes) es una herramienta utilizada para la estimación de algunos parámetros (costes en personas, tiempo,...) en el diseño y construcción de programas y de la documentación asociada requerida para desarrollarlos, operarlos y mantenerlos, es decir, en la aplicación práctica de la Ingeniería del Software[4].

Entre algunas de sus ventajas se encuentran los ajustes a la medida dependiendo del software a desarrollar, involucrando en la estimación del coste a los puntos objeto, puntos función y líneas de código fuente; utilizando modelaciones no lineales para atender a la reingeniería y reusabilidad del software, permitiendo hacer estudios prediseño. La utilización del COCOMO dio el siguiente resultado.

[4] Concepción García, Rita. Procedimiento para la valoración de la sostenibilidad de un PI.

Los elementos que se tienen en cuenta para calcular el total de puntos de función desajustados se encuentra en la Tabla 2-8.

Elementos	Bajo		Medio		Alto		Sub-total
	No	x Peso	No	x Peso	No	x Peso	
Entrada Externa (EI)	8	*3=24	2	*4=8	0	*6=0	**32**
Salida Externa (EO)	1	*4=4	0	*5=0	1	*7=7	**11**
Peticiones(P)	0	*3=0	0	*4=0	1	*6=6	**6**
Archivo Lógico Interno (ILF)	29	*7=259	0	*10=0	0	*15=0	**203**
Archivo Interfaz Externa (ELF)	74	*5=5	0	*7=0	0	*10=0	**370**
Total de puntos de función desajustados (UFP)							**622**

Tabla 2.8 *Puntos de función desajustados.*

La estimación de las instrucciones fuentes del proyecto (Ver Tabla 2-9) se basa en la cantidad de instrucciones fuentes por punto de función del lenguaje a usar en la implementación del sistema. Donde UFP es el total de puntos de función desajustados, y ratio es una constante para las SLOC de cada lenguaje de programación en este caso tiene un valor para PHP de 90.

Cálculo de la cantidad de instrucciones fuentes.

Para el cálculo de las instrucciones fuentes (SLOC) se utilizó la fórmula siguiente:

SLOC = UFP * ratio

SLOC = 622 * 90

SLOC = 55980

Lenguajes	UFP del lenguaje	Ratio	UFP(leng)*Ratio
PHP	**622**	90	*55980*

Tabla 2.9 *Instrucciones fuentes por lenguaje utilizado*

KSLOC = *55,980* (Miles de líneas de código)

Cálculo del esfuerzo y del tiempo.

Luego de calculada la cantidad de instrucciones fuentes, se utilizó este valor en el cálculo del esfuerzo dado por la fórmula de Bohem:

$$PM_{NS} = A \times Size^{E} \times \prod_{i=1}^{n} EM_{1}$$

donde:

$$E = B + 0.01 \times \sum_{j=1}^{s} SF_j$$

Se tiene además los valores de A y B como valores constantes de 2.94 y 0.91 respectivamente.

Para el cálculo del tiempo se empleó la formula:

$$TDEV_{NS} = C \times \left(PM_{NS}\right)^{F} \quad \text{dc}$$

donde:

$$F = D + 0.2 \times 0.01 \times \sum_{j=1}^{s} SF_j$$

ó

$$F = D + 0.2 \times (E - B)$$

Para obtener los resultados de las fórmulas anteriormente expuestas, se calcularon los valores de cada factor de escala (SF_j) y de cada multiplicador de esfuerzo (EM_i).

Factor de Escala	**Valor**	**Justificación**
PREC	1.24	Resulta algo familiar para los desarrolladores el tipo de aplicación.
FLEX	1.01	Hubo cierto acuerdo de forma general en cuanto a las interfaces de diseño y los requisitos del software.
RESL	2.83	Se tomó ciertas estrategias para tener el mínimo de riesgos en el entorno de la aplicación.
TEAM	3.29	Bastas experiencias en el trabajo en equipo. Buen acoplamiento de forma general a la hora de trabajo.
PMAT	7.80	Existe gran madurez en cuanto a la complejidad del software.

Tabla 2.10 *Factores de Escala*

Multiplicador	Valor	Justificación
PERS	0.83	Los desarrolladores tienen en general alto conocimiento en la programación de sistemas, se considera alta las capacidades de los analistas y de los programadores. No se esperan cambios significativos en el personal del equipo de desarrollo.
RCPX	1.00	El producto tiene una moderada complejidad, existe una alta confiabilidad de la documentación. La base de datos que se utiliza tiene un volumen mediano de información por lo que se considera de tamaño moderado.
RUSE	1.07	En la implementación del sistema existe una alta reusabilidad de códigos, con vistas a la construcción de componentes a través del proyecto.
PREX	1.00	Basta experiencia en cuanto al lenguaje, se conoce el tipo de software y herramientas para el desarrollo de aplicaciones de este tipo. Por tanto se valora como nominal.
SCED	1.00	Es nominal la expansión y dilatación del tiempo para desarrollar el sistema.
FCIL	0.73	Se utilizan herramientas modernas de programación como Visual Studio.NET, lenguaje C#, SQL y HTML. Así como para la documentación se utilizó la notación UML y para su modelado visual se empleó la herramienta Visual Paradigm for UML 5.3 Enterprise Edition.

Tabla 2.11 *Multiplicadores de Escala*

Características	Valor
Puntos de función desajustados	55,980
Lenguaje	PHP
Instrucciones fuentes por puntos de función	90
Instrucciones fuentes	55980

Tabla 2.12 *Características*

De los resultados anteriores se obtiene:

Sumatoria de los factores de escala:

ΣSF = PREC + FLEX + RESL + TEAM + PMAT ΣSF = 1.24 + 1.01 + 2.83 + 3.29 + 7.80 ΣSF=16.17

ΠEM = RCPX * RUSE * PERS * PREX * FCIL * SCED ΠEM= 1.00 * 1.07 * 0.83 * 1.00 * 0.73 * 1.00 ΠEM= 0,648313

Cálculo de esfuerzo

$E= B + 0.01 * \Sigma SFi$ $E= 0.91 + 0.01 * 16.17 = 1.0717$

$PM = A * (Size)^{E} * \prod EMi$

PM= 2.94 * $55{,}980^{1.0717}$* 0.648313=141.9057 hombres-mes.

Se necesitan aproximadamente 142 personas en un mes para realizar el software.

Calculamos el tiempo de Desarrollo

Al saberse el valor del esfuerzo se puede calcular el tiempo de desarrollo (TDEV) estimado del software, es decir, cantidad de meses necesarios para desarrollar el software.

Siglas	Indicador	Valor o fórmula
TDES	Tiempo de desarrollo	$C * (PM)^{F}$
C	Constante	3.67
PM	Esfuerzo	141.9057 hombres-mes
F	Exponente de escala	D + 0.2 * (E - B)
D	Exponente base para la ecuación del cronograma (constante)	0.28
E	Agregado de 5 factores de escala	B + 0.01 * Σ SFi
B	Exponente de base escalado para la ecuación de esfuerzo que puede ser calibrado (constante)	0.91
ΣSF	Factores de escala	16.17

Tabla 2.13 *Datos para calcular Tiempo de Desarrollo.*

F= D + 0.2 * (E – B)

F = 0.28 + 0.2 (1.0717 - 0.91) = 0.31234

$TDES= C * (PM)^{F}$

$TDES = 3.67 * (141.9057)^{0.31234} = 17.25$

EL tiempo necesario para desarrollar el proyecto es de17 meses aproximadamente.

Determinar la cantidad de hombres

Teniendo en cuenta el tiempo de desarrollo se calcula la cantidad de personas (CH) necesarios para desarrollar el software.

Siglas	Indicador	Valor o fórmula
CH	Cantidad de hombres por mes	PM/TDES
PM	Esfuerzo	141.9057 hombres-mes
TDES	Tiempo de desarrollo	17.25 meses

Tabla 2.14 *Constantes y fórmulas para el cálculo de tiempo de desarrollo*

CH = 141.9057 / 17.25 = 8.23 personas

Se necesitan 8 persona para realizar el software en 17.25 meses:

CH^*= 1 personas

TEDV = PM/ CH^* = 141.9057 / 1 = 141.9057 meses

Son necesarios 142 meses para que 1 persona desarrolle el software.

Determinar el costo del software

El costo del software depende del salario promedio de las personas que lo desarrollan y del esfuerzo que ellas realizan para la ejecución del mismo.

Siglas	**Indicador**	**Valor o fórmula**
C	Costo del proyecto	CHM *PM
CHM	Costo de hombres por mes	CH^* * SP
SP	Salario básico de un Ingeniero	$ 225.00
PM	Esfuerzo	189,9753 hombre-mes

Tabla 2.15 *Datos para determinar Costo del Software.*

El salario medio es de $ 225.00

C = 1 * 225 * 141.9057 = $ 31928.7825

El software cuesta $ 31928.7825

Resultado de la estimación de esfuerzo, tiempo de desarrollo, cantidad de hombre y costo del proyecto.

Cálculo de:	Valor	Justificación
Esfuerzo	141.9057 hombres-mes	Cantidad de tiempo que una persona invierte trabajando en el desarrollo de un proyecto
Tiempo de desarrollo	141.9057 meses	Cantidad de meses para terminar el proyecto.
Cantidad de personas	1	Cantidad de personas necesarias para terminar el proyecto en 189,9753 meses.
Costo	$ 31928.7825	Cantidad de dinero que cuesta el proyecto después de terminado.
Salario medio	$ 225.00	Salario básico de un ingeniero

Tabla 2.16 *Valores Finales*

Beneficios Tangibles e Intangibles esperados

EMCOMED no cuenta con ningún software de apoyo a la toma de decisiones por lo que el mismo será de gran ayuda ya que utilizando este sistema se tendrá un mayor conocimiento de como esta funcionando la misma, se tendrá un mayor argumento a la hora de tomar decisiones rápidas y certeras.

EL software influirá en la calidad de los servicios prestados ya que se tendrá un conocimiento exacto de cómo esta funcionando las metas propuestas y las posibles causas que podrían estar afectando el cumplimiento de las mismas

Las herramientas utilizadas para la elaboración de esta aplicación Web son libre lo cual no generara gastos de pago de licencia, además el software es multiplataforma lo cual será de gran beneficio si la empresa decide emigrar a software libre.

Análisis de costos y beneficios

El desarrollo de todo producto tiene siempre un costo que puede ser justificado o no. En el caso de los productos informáticos esto depende en gran medida de los beneficios tanto tangibles como intangibles que produce el mismo.

Este sistema surge por la necesidad que tiene la empresa EMCOMED Droguería Holguín de tener un sistema que realice el Control de los Indicadores de Gestión del CMI, pero la estructura que propone sirve para cualquier sistema con propósitos similares. También este sistema podrá comercializarse ya que las herramientas utilizadas para su desarrollo es libre lo cual podrá ser una fuente de ingreso para la Organización.

Por estas razones se concluye que el software representará beneficios y que debe llevarse a cabo su implementación.

No solo debemos llevar a cabo un proyecto cuando los cálculos indiquen que es beneficioso, sino que debemos tener en cuenta también si es sostenible o no, y en caso de serlo, ver las implicaciones que tendría en distintos aspectos tales como el administrativo, socio-humanista, tecnológico y ambiental.

2.3.2.2 Socio Humanista

Con la implantación de este software se contribuirá a maximizar la rentabilidad de la empresa ya que ofrecerá una imagen gráfica y clara de las operaciones del negocio. Permitirá medir el grado de contribución personal con los resultados de la empresa mediante los indicadores. Permitirá mostrar la información necesaria para tomar decisiones oportunas.

La implantación de este software no generara desempleo ya que este es un software de apoyo a la toma de decisiones y siempre se necesitaran a los especialistas y a los trabajadores para ordenar y ejecutar las decisiones tomadas

Para mitigar el rechazo al cambio de los usuarios se utilizaron interfaces parecidas a las de Windows y se utilizaron gráficas para mostrar proyección histórica de los indicadores.

EL software fue desarrollado con herramientas Web libre por tanto no existirá problemas con un registro de la propiedad intelectual como derecho de autor, patentes, marcas cuando se decida comercializar.

Por lo antes expuesto se llega a la conclusión que el sistema no tendrá gran repercusión en lo Socio humanista por lo cual debe llevarse a cabo su implementación.

2.3.2.3 Tecnológico

El impacto tecnológico que provoque el producto informático será mínimo debido a que no es necesaria la capacitación de los trabajadores para el uso computadoras, pues las habilidades que han adquirido en su trabajo y los cursos de computación que se han impartido en la entidad son suficientes para utilizarlo, se utilizaron para el diseño del software metáforas visuales conocidas por los usuarios que les proporcionara información necesaria para identificar algunas funciones del sistema. Se proveerá de un manual de usuario para una mejor comprensión de las funcionalidades del software

El software va a ser de gran ayuda en la empresa EMCOMED ya que el Cuadro de Mando Integral dará un alto conocimiento de cómo va funcionando la empresa en las cuatro perspectiva, financiera, cliente, procesos y aprendizaje y crecimiento, lo cual permitirá tomar decisiones para llevar la empresa adelante y hacerla mas competitiva.

En Cuba muchas empresas quieren aplicar un Cuadro de Mando Integral por las ventajas que este brinda, este software podrá comercializarse sin problemas ya que no se necesitara pagar una licencia por ser un software libre y multiplataforma.

Por lo antes expuesto se llega a la conclusión que el sistema no tendrá gran repercusión en el impacto Tecnológico por lo cual se considera que debe llevarse a cabo su implementación.

2.3.2.4 Ambiental

Se puede definir impacto ambiental como cualquier alteración que se produzca en el medio ambiente al realizarse un proyecto o cualquier actividad humana.

Este software no producirá gastos en cuanto a máquinas ya que con las que se dispone serán suficientes para el empleo del mismo. El gasto de papel no será considerable ya que solo se imprimirá un reporte, al cual tendrá acceso un único usuario.

Para evitar que los usuarios tengan problemas visuales se utilizaron colores claros y refrescantes como el azul y el blanco, se utilizaron botones e hipervínculos de tamaño adecuado para evitar que el usuario tenga que fijar mucho la vista, el sistema no provocara ruidos.

Este software ayudara a disminuir el estrés en los trabajadores de la entidad ya que contaran con una herramienta que les proporcionará la información necesaria para la toma de decisiones y les mostrará gráficamente el estado de los indicadores.
Como el software y sus componentes son libres podrán ser reutilizados para crear nuevos sistemas informáticos.

2.3.3 Evolución

La evolución es un problema crucial en el desarrollo de software. Cada vez es mayor el número de autores que consideran la evolución como una propiedad intrínseca del proceso de desarrollo, pues el software sufre continuos cambios durante su diseño y construcción, así como durante su uso. La Ingeniería del Software es cada día más consciente de la necesidad de sentar las bases metodológicas y las herramientas que ayuden a crear software con capacidades evolutivas **[35**].

Las necesidades de creación de software con capacidades evolutivas son:

- ✓ Posibilitar la interacción con un entorno cambiante permitiendo la adaptación a este entorno.
- ✓ Permitir la adaptación a diferentes usuarios con diferentes necesidades.
- ✓ Dejar realizar cambios en la especificación de requisitos durante el desarrollo y uso del sistema, y que la incorporación de estos cambios sea lo más sencilla posible para el equipo de desarrollo.
- ✓ Permitir que el software sea independiente de los lenguajes de programación, sistemas de bases de datos, sistemas operativos, etc., que lo soportan.
- ✓ Disponer de diversas versiones del software.

Las dos tendencias fundamentales al abordar la evolución de software en la actualidad son **[34]**:
Aquellas que se ocupan de cómo se realiza el proceso evolutivo, incluyendo los mecanismos y las herramientas que permitan un control sistemático del proceso. Esta tendencia se enfoca en los mecanismos y herramientas, los métodos y medios con los cuales el cambio progresivo y el crecimiento pueden ser logrados de una manera

sistemática y controlada. Hay que cuidar en esta tendencia de no confundir evolución con cambio en el software, que es más específico.

Las que consideran el qué y el por qué de la evolución, buscando encontrar la naturaleza de la evolución y su impacto. Una mayor penetración y una mejor comprensión de estos aspectos tienen que guiar a métodos mejorados para planificar, administrar e implementar el software.

Las tres vistas de la evolución, el cómo, el qué y el por qué, son complementarias. Ambas son necesarias. Juntas y en asociación con la teoría concebida incrementan el potencial de una mejora bien fundada y para la validación del valor práctico que puede entregar **[35]**.

2.4 Conclusiones

La parte económica es otro de los problemas en la elaboración de los sitios Web, pues hay que estimar el costo de construir el software y por consiguiente, las utilidades que generaría esa inversión.

Sobre la base del análisis anterior se puede inferir que el sitio Web es sostenible en los aspectos económico, social, tecnológico y ambiental, en el contexto en el que se empleará y que evolucionará.

CAPÍTULO 3

ELABORACIÓN Y EVALUACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

En este capítulo se describen las etapas de diseño e implementación de la solución propuesta a través de la metodología ICONIX, así como el diagrama de despliegue, estándares de interfaz y de codificación.

Se hará una exposición de los resultados que se alcanzaron con la implementación del sistema, a través de encuestas y entrevistas que midieron la satisfacción de los usuarios del sistema con respecto al mismo.

3.2 Diseño e Implementación del sistema

3.2.1 Diagrama de Clases de Diseño Web del Sistema

El diagrama de clases de diseño Web del sistema (diagrama de clases para diseño orientado a objetos) se obtuvo como resultado del refinamiento del modelo conceptual y se basó fundamentalmente en los diagramas de interacción que no son más que una abstracción a la implementación del sistema. A continuación se mostrarán los diagramas de interacción o secuencia principales del sistema.

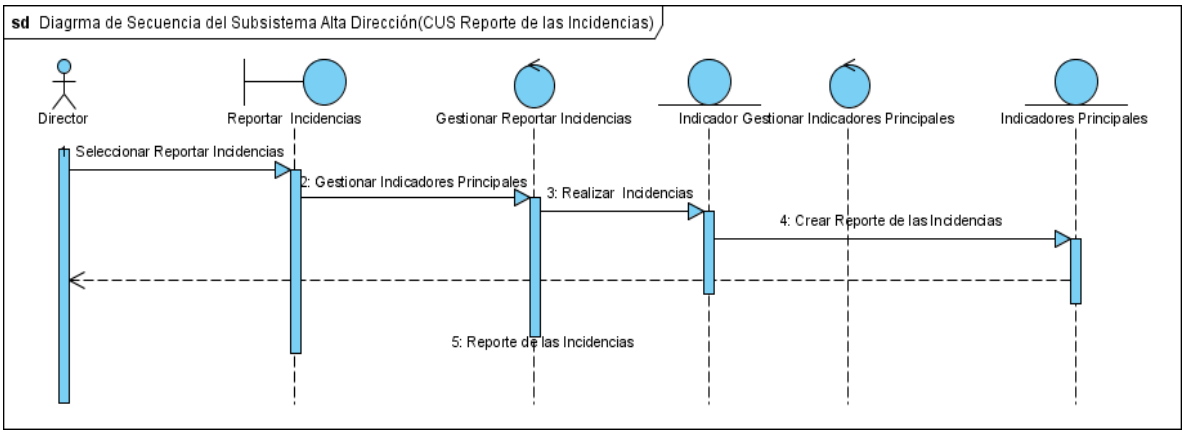


Figura 3.1 *Diagrama de Secuencia del paquete Nivel Alta Dirección (CUS Reporte de las Incidencias).*

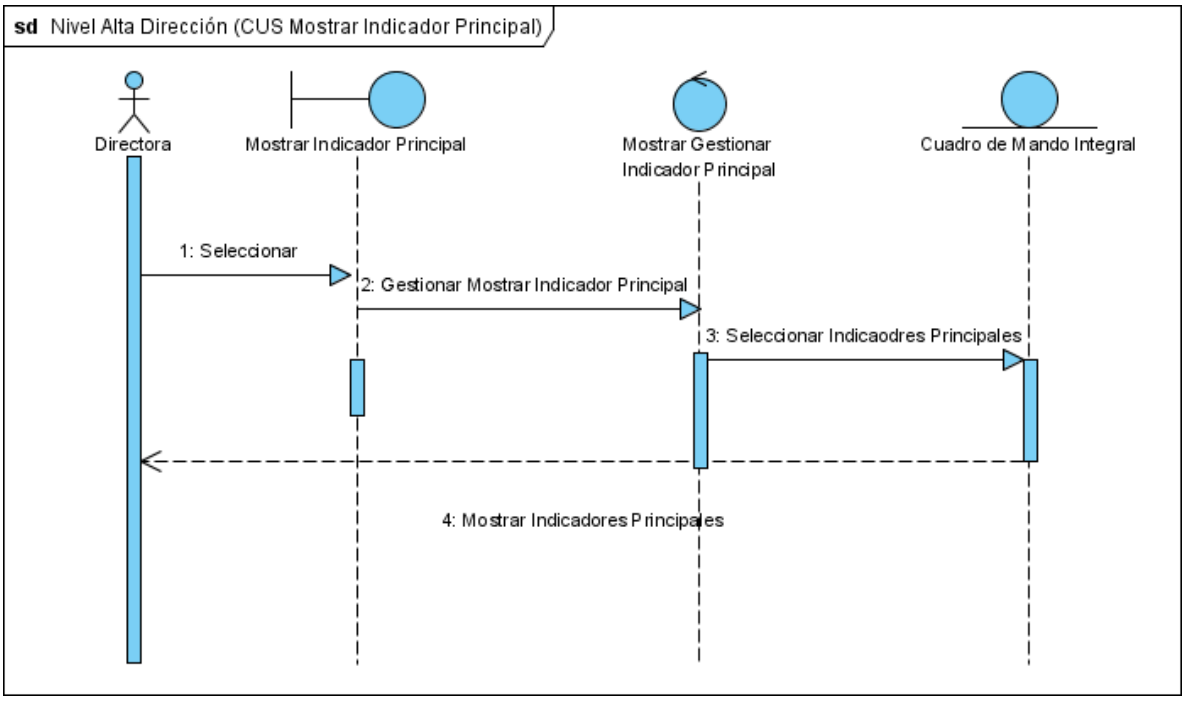


Figura 3.2 *Diagrama de Secuencia del paquete Nivel Alta Dirección (CUS Mostrar Indicadores Principales).*

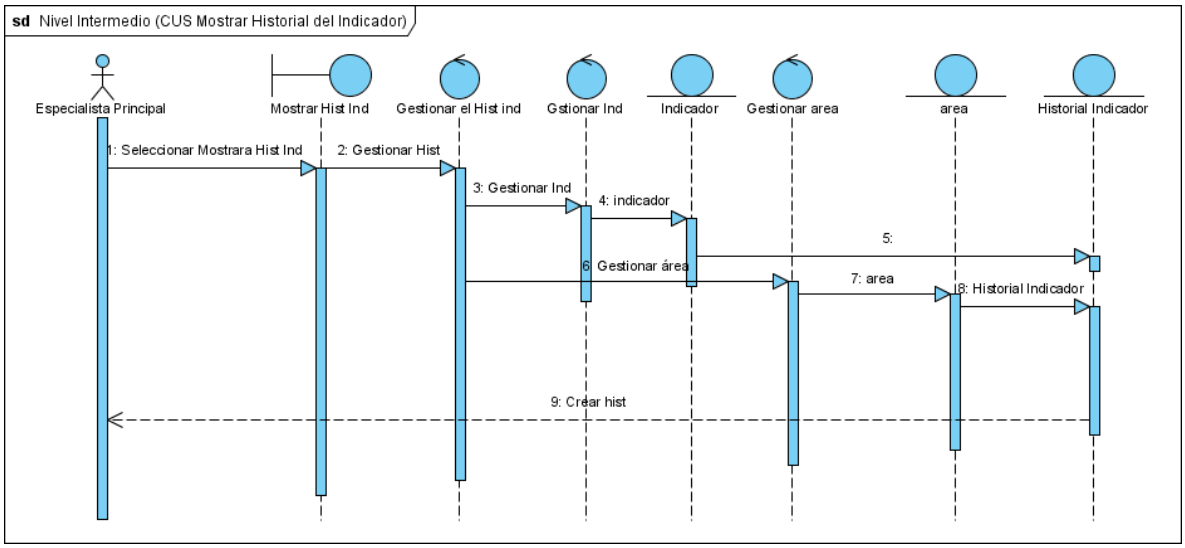


Figura 3.2 *Diagrama de Secuencia del paquete Nivel Intermedio (CUS Mostrar Historial del Indicador).*

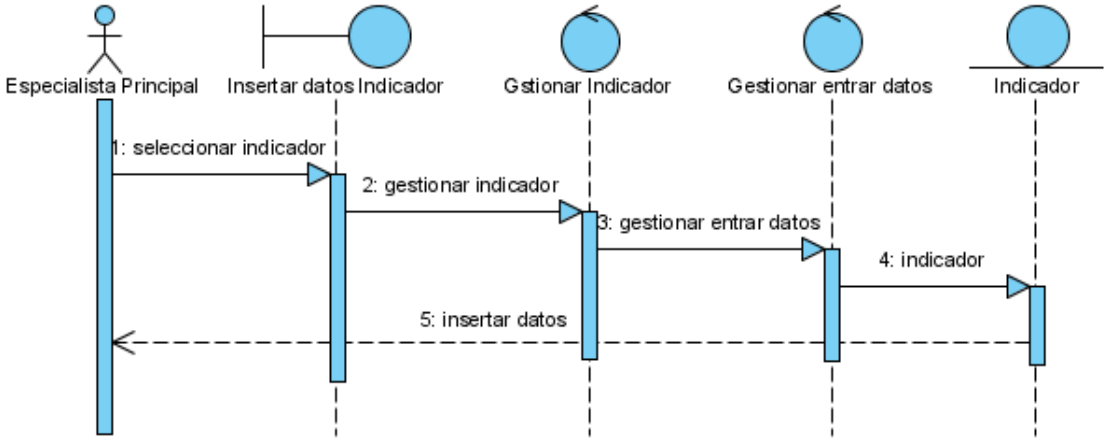


Figura 3.3 *Diagrama de Secuencia Nivel de Seguridad (CUS Insertar datos al Indicador).*

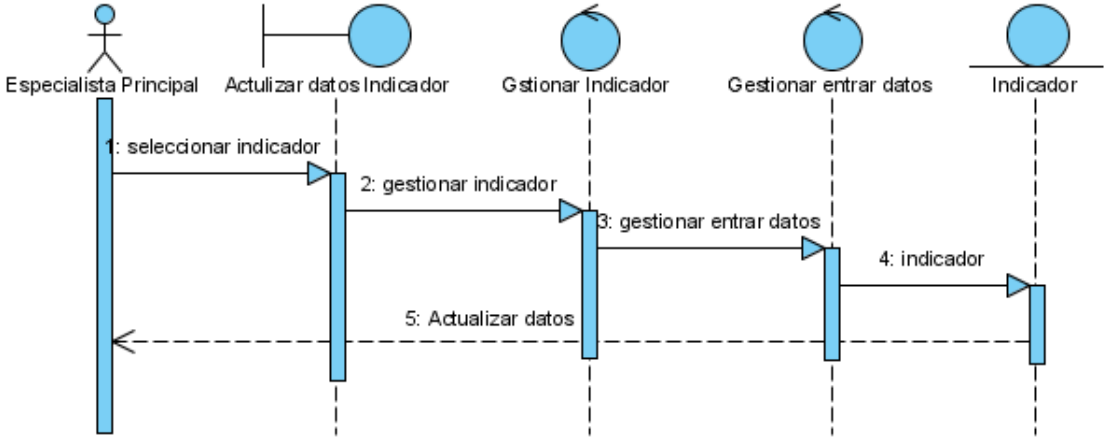


Figura 3.4 *Diagrama de Secuencia del paquete del Nivel de Seguridad (CUS Actualizar datos al Indicador).*

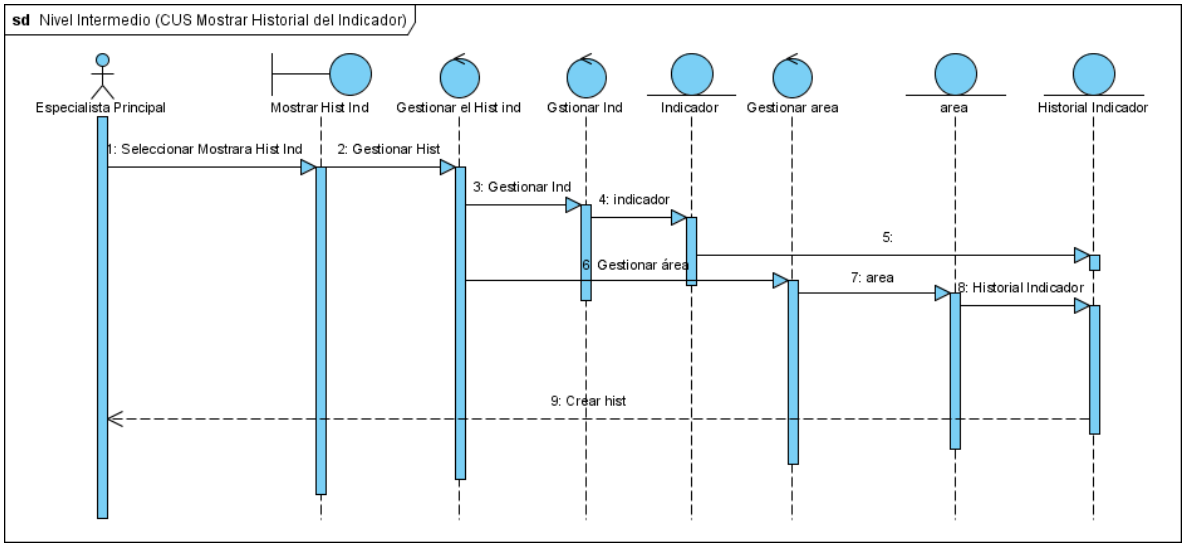


Figura 3.5 *Diagrama de Secuencia del paquete del Nivel Intermedio (CUS Mostrar Historial del Indicador)*

Con el objetivo de alcanzar una comprensión más precisa de los requisitos que se describieron en la captura de requerimientos, refinarlos y estructurar el sistema entero, se desarrolló el modelo de análisis o robustez, el cual permitió además razonar los aspectos internos del sistema así como la flexibilidad ante los cambios y la reutilización.

El modelo de análisis o robustez fue la primera aproximación al modelo de diseño, en el cual se pudo moldear el sistema y encontrar la forma que diera vida a los requisitos incorporados en el sistema, que crea un punto de partida apropiado para las actividades de implementación subsiguientes, ya que permitió descomponer los trabajos de implementación en partes más manejables.

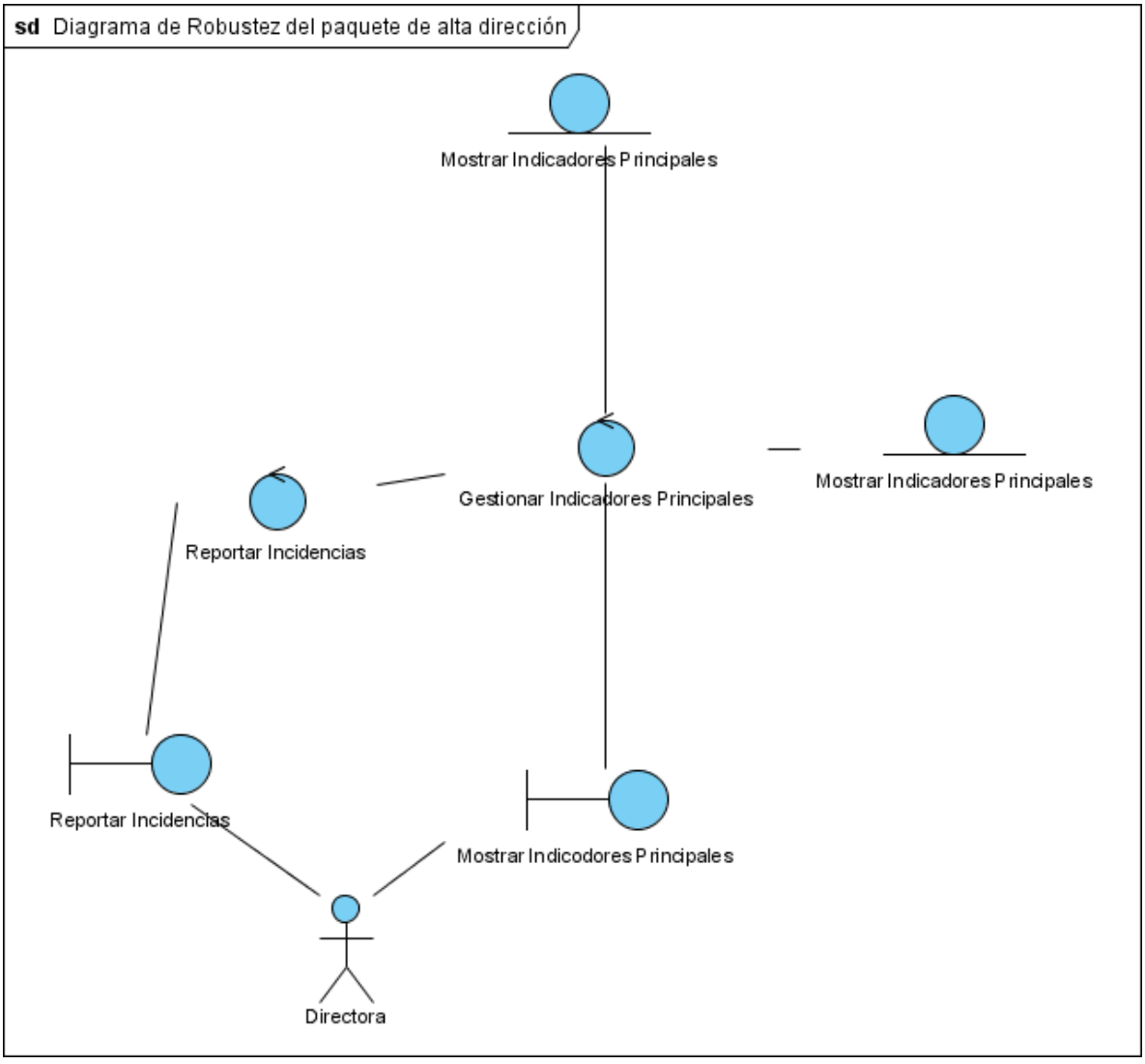


Figura 3.6 *Diagrama de Robustez del paquete Nivel de Alta Dirección.*

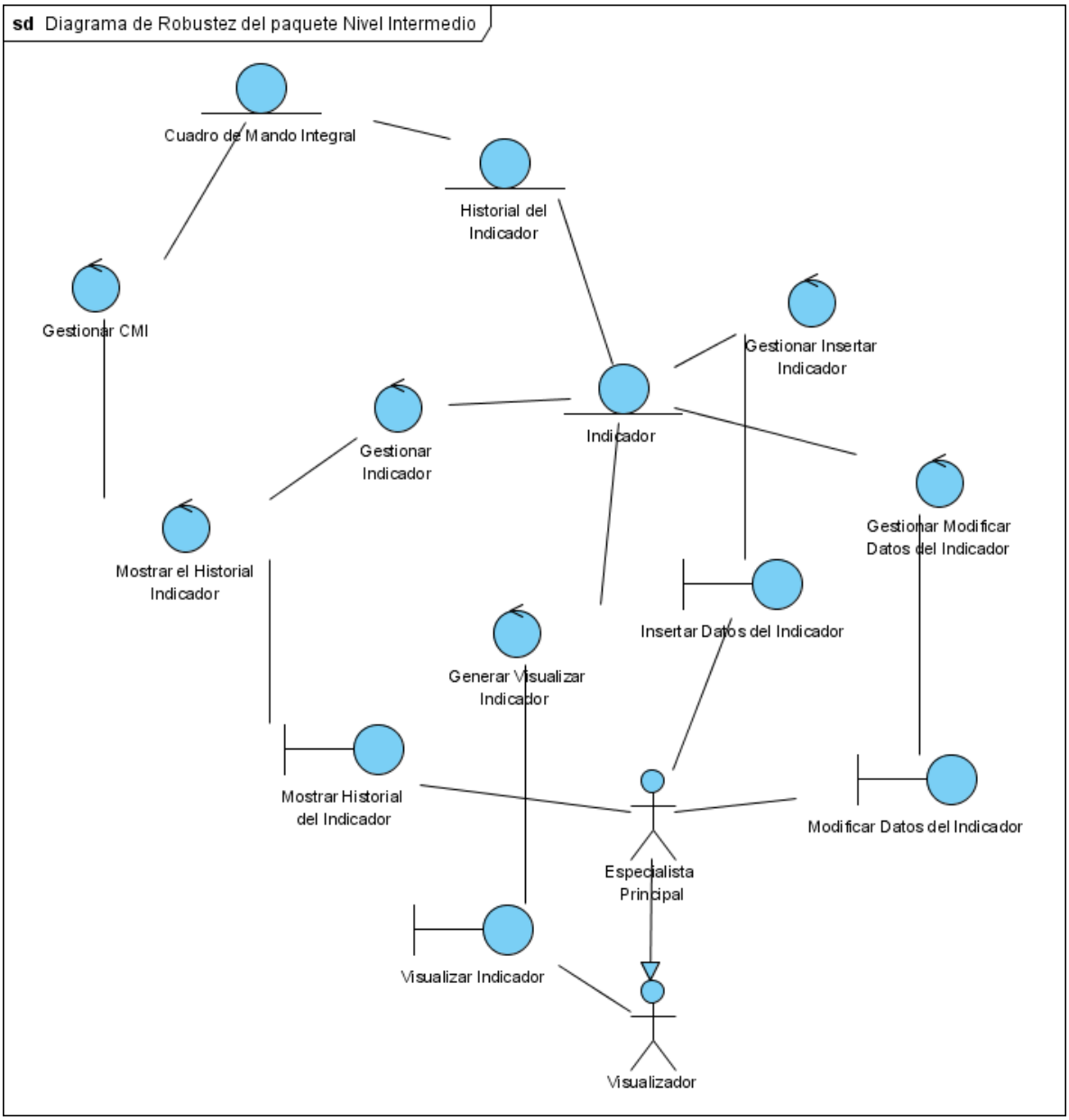


Figura 3.7 *Diagrama de Robustez del paquete Nivel Intermedio*

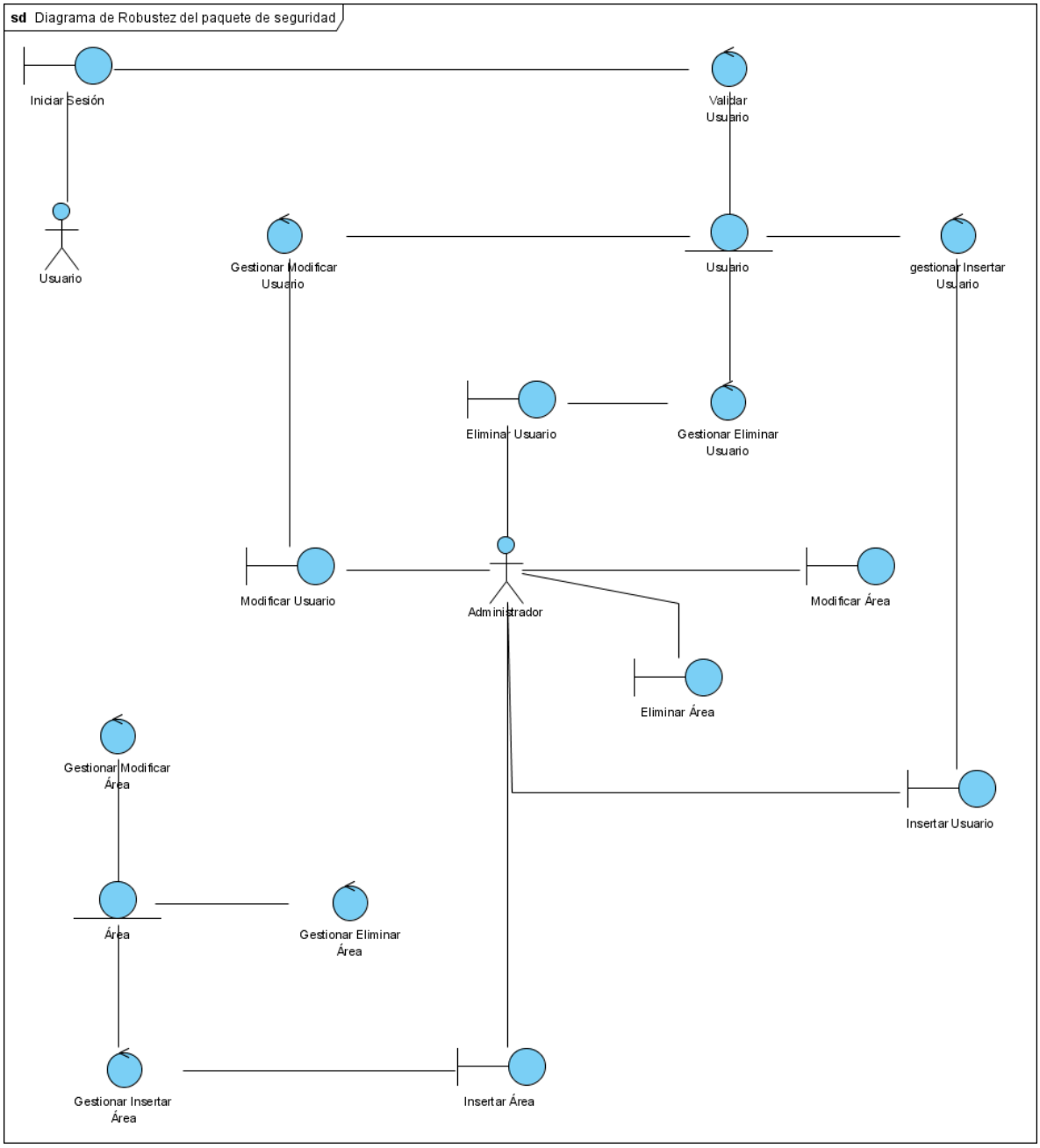


Figura 3.8 *Diagrama de Robustez del paquete de Nivel de Seguridad*

El sistema propuesto esta compuesto por páginas web. Para representar estas páginas en la etapa de diseño se tienen las clases del diseño que se separan en tres grupos: clases clientes, clases servidoras y clases formularios, los cuales simbolizan los estereotipos *Client Page*, *Server Page* y *HTML Form* respectivamente. Las clases que almacenan la información persistente de la aplicación, se nombran clases entidades que representa el estereotipos *Entity*. A continuacion se mostrara las principales clases de diseño.

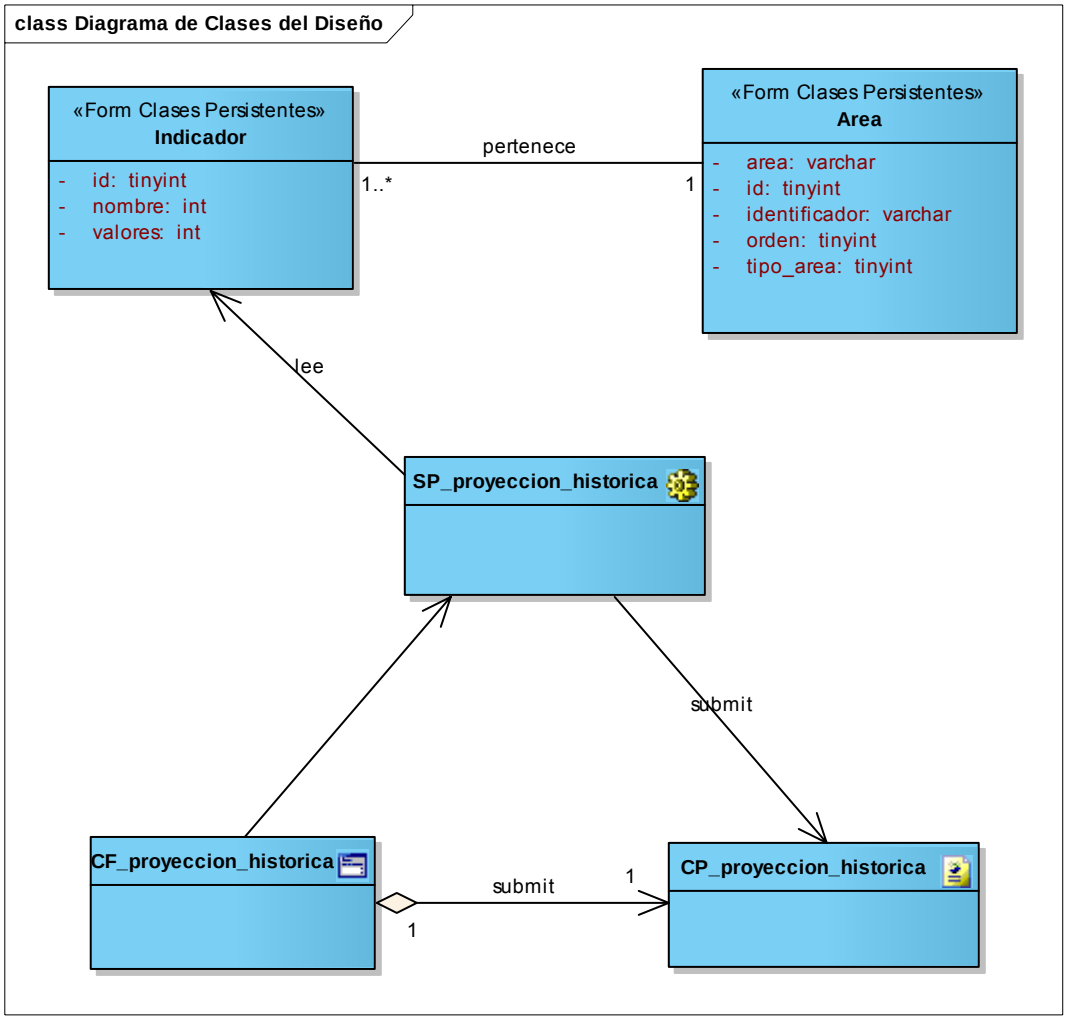


Figura 3.9 *Diagrama de Clases del Diseño*

3.2.2 Principios de Diseño

Con el fin de apoyar la calidad y aceptación del sistema por parte de los usuarios finales se tuvieron en cuenta aspectos de diseño, tales como: la interfaz de usuario, formato de salida, tratamiento de errores, etc.

3.2.2.1 Estándares en la Interfaz del Sistema

Partiendo de la variedad de usuarios que pueden interactuar con el sistema propuesto y la importancia de facilitar el trabajo y satisfacer al usuario a través de una comunicación sencilla, amigable e intuitiva, se tuvo en cuenta el diseño visual del sistema. Para ello, se definieron estándares en la interfaz, acorde con las características tanto de los usuarios como del lugar donde se implantará el sistema.

Se emplearon combinaciones agradables a la vista de colores sobre la gama del azul, negro y blanco por ser los colores corporativos de la Organización. El negro y el blanco se trataron para las letras y el azul en el fondo de las páginas, lo que proporciona un contraste atractivo y favorable a la lectura.

Para el logro de un buen diseño y evitar el rechazo al cambio, hizo falta discernir los requisitos no funcionales de apariencia, así como las interfaces precisadas por los usuarios, para facilitar al actor del sistema la ejecución de los casos de uso de manera eficiente. Para ello, se contó con el criterio y apoyo de los diseñadores y algunos trabajadores de la empresa.

3.2.2.2 Tratamiento de Excepciones

Con el fin de tratar situaciones inusuales que pueden ocurrir durante la ejecución del sistema y evitar así la interrupción del flujo normal de las sentencias, en los puntos propensos a irrumpir con la integridad se chequea la ocurrencia de excepciones para ser capturadas y tratadas. El flujo de datos tanto de entrada como de salida hacia el servidor, la entrada de datos por teclado en la interfaz del sistema, así como la lectura de textos y tablas son los principales puntos críticos identificados en el sistema. Ante los errores que pueden ocurrir, se emplea la filosofía de mantener informado al usuario de manera explícita y constante, con el objetivo de guiarlo e incrementar las opciones de solución. El tratamiento de excepciones recobra importancia no sólo en las alternativas antes mencionadas para el manejo y recuperación de errores, sino que proporciona además un diseño mucho más estructurado, legible, robusto y fácil de mantener.

3.2.3 Estándares de Codificación

Para el desarrollo del sistema, se establecieron estándares de codificación para asegurar la legibilidad del código, tanto para los programadores como los responsables de mantenimiento/actualización, facilitando además la portabilidad entre plataformas y aplicaciones.

Variables locales:

La declaración de variables locales se realizó en líneas separadas y los nombres se especificaron en minúscula, de forma abreviada e intuitiva.

En las asignaciones, ya sean de valores fijos o resultados de una función, se dejaron espacios a ambos lados del signo igual.

Ejemplo:

```
switch ( $this->session->userdata ( 'error_login' ) ) {
    case 0:
      $data_content ['error_log'] = '';
      break;
    case 1:
      $data_content ['error_log'] = '&iexcl;Campos vacios!';
      break;
    case 2:
      $data_content ['error_log'] = '&iexcl;El usuario no existe!';
      break;
    case 3:
      $data_content ['error_log'] = '&iexcl;Contrase&ntilde;a incorrecta!';
      break;
    default:
      $data_content ['error_log'] = '';
}
```

Figura: 3.10 *Fragmento de código de la función index.*

Variables globales (constantes):

Los nombres de las variables globales son escritos en mayúsculas, y las palabras se separan con guiones bajos (_).

Ejemplo:

```
$area = $_POST['area'];
$identificador = $_POST['identificador'];
$orden = $_POST['orden'];
$tipo_area = $_POST['tipo_area'];
```

Figura: 3.11 *Fragmento de código de la función insert.*

Estructuras de control:

En las estructuras de control se dejaron espacios entre la palabra clave y el paréntesis de apretura, para diferenciarlos de las llamadas a funciones.

Ejemplo:

```
while ( $info = mysql_fetch_object ( $valores ) ) {
  $hcpu [] = $info->h_cpu_rotos;
  $mcpu [] = $info->m_cpu_rotos;
  $dias [] = sacarmesdia ( $info->fecha );
  $hsum = $hsum + $info->h_cpu_rotos;
  $msum = $msum + $info->m_cpu_rotos;
  $cd++;
}
$hmed = $hsum / $cd;
$mmed = $msum / $cd;

for ($i = 0; $i <= $cd - 1; $i++)
  $hdl = ( $hmed - $hups [ $i ] ) * ( $hmed - $hups [ $i ] );
$hdesv = sqrt ( $hdl / ( $cd - 1 ) );
for ($i = 0; $i <= $cd - 1; $i++)
  $mdl = ( $mmed - $mups [ $i ] ) * ( $mmed - $mups [ $i ] );
$mdesv = sqrt ( $mdl / ( $cd - 1 ) );
```

Figura: 3.12 *Fragmento de código de la función (Gráficas de equipos CPU). g_equip_cpu.*

Definición de clases y funciones:

En el nombre de las clases no se hace uso de abreviaturas y en las funciones, se fu lo más descriptivo posible.

Ejemplo de clase:

```
class Login extends Controller {

    function Login()
    {
        parent::Controller();
    }
```

Figura: 3.13 *Fragmento de código de la clase Login.*

Ejemplo de función:

```
function sacarmesdia ( $fech ) {
  $spf = explode ( '-', $fech );
  return $spf [2] . "/" . $spf [1];
}
```

Figura: 3.14 *Fragmento de código de la clase jpgraph.*

Comentarios:

Se hizo uso de comentarios para facilitar la comprensión del código, se abundó más en los procedimientos complejos. Se escribieron arriba del método correspondiente.

Ejemplo:

```
/*Consulta mostrar las áreas de resultados claves */
$query = $this->db->query ( 'SELECT * FROM cmi_areas where tipo_area=0 order by orden' );
/* Consulta mostrar las otras áreas */
$consulta = $this->db->query ( 'SELECT * FROM cmi_areas where tipo_area=1 order by orden' );
/* Consulta mostrar todas las áreas */
$consulta_todas = $this->db->query ( 'SELECT * FROM  cmi_areas order by orden' );
```

Figura: 3.15 *Fragmento de código de la clase Configuracion_areas.*

3.2.4 Modelo de Despliegue

En el modelo de despliegue (Ver figura 3.16) se describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre nodos de cómputo, es decir, representa la correspondencia entre la arquitectura software y la arquitectura hardware. Este modelo se utilizó como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

El segundo procesador simboliza los usuarios, solo hace falta contener un navegador Web como elemento de software. La comunicación entre los usuarios, el Servidor Web y de Bases de Datos se establece utilizando el protocolo de comunicación TCP/IP.

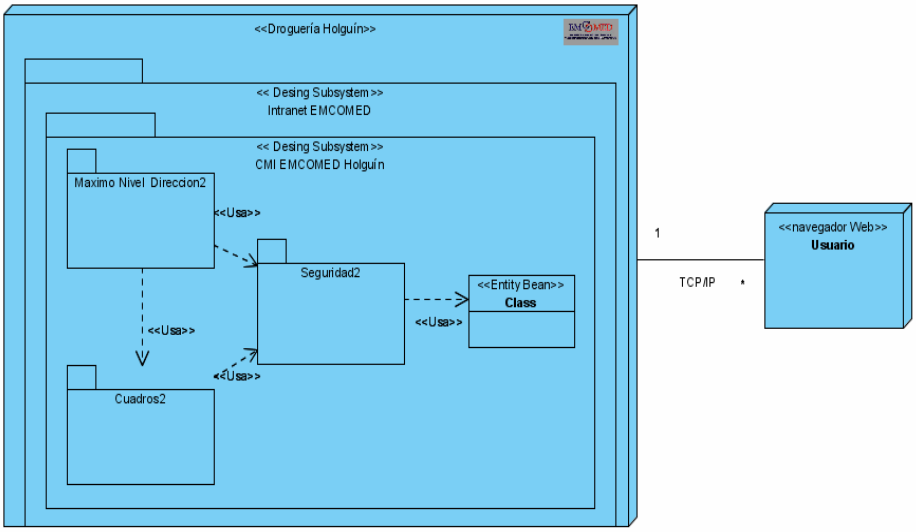


Figura: 3.16 *Diagrama de Despliegue del sistema propuesto*

3.2.5 Modelo de Implementación

En el modelo de implementación se describieron los elementos resultantes del diseño en términos de componentes. Este modelo predomina en la fase de construcción y tiene como

propósito principal desarrollar la arquitectura y el sistema como un todo, a pesar de haber capturado la mayor parte de ésta durante el diseño. Permite además definir la organización del código, planificar las integraciones necesarias al sistema en cada iteración e implementar las clases y subsistemas identificados durante el diseño.

3.2.5.1 Diagrama de Componentes

Partiendo de la agrupación de los elementos del diseño a implementar por componente, se desarrollaron los diagramas de componentes, se puede apreciar en la (figura 3.17), con el propósito de modelar el apartado de la vista estática del sistema para mostrar la organización y las dependencias entre los componentes.

A partir del diagrama de componentes se pudo cristalizar lo definido en los flujos anteriores, así como implementar el sistema propuesto durante la etapa de construcción.

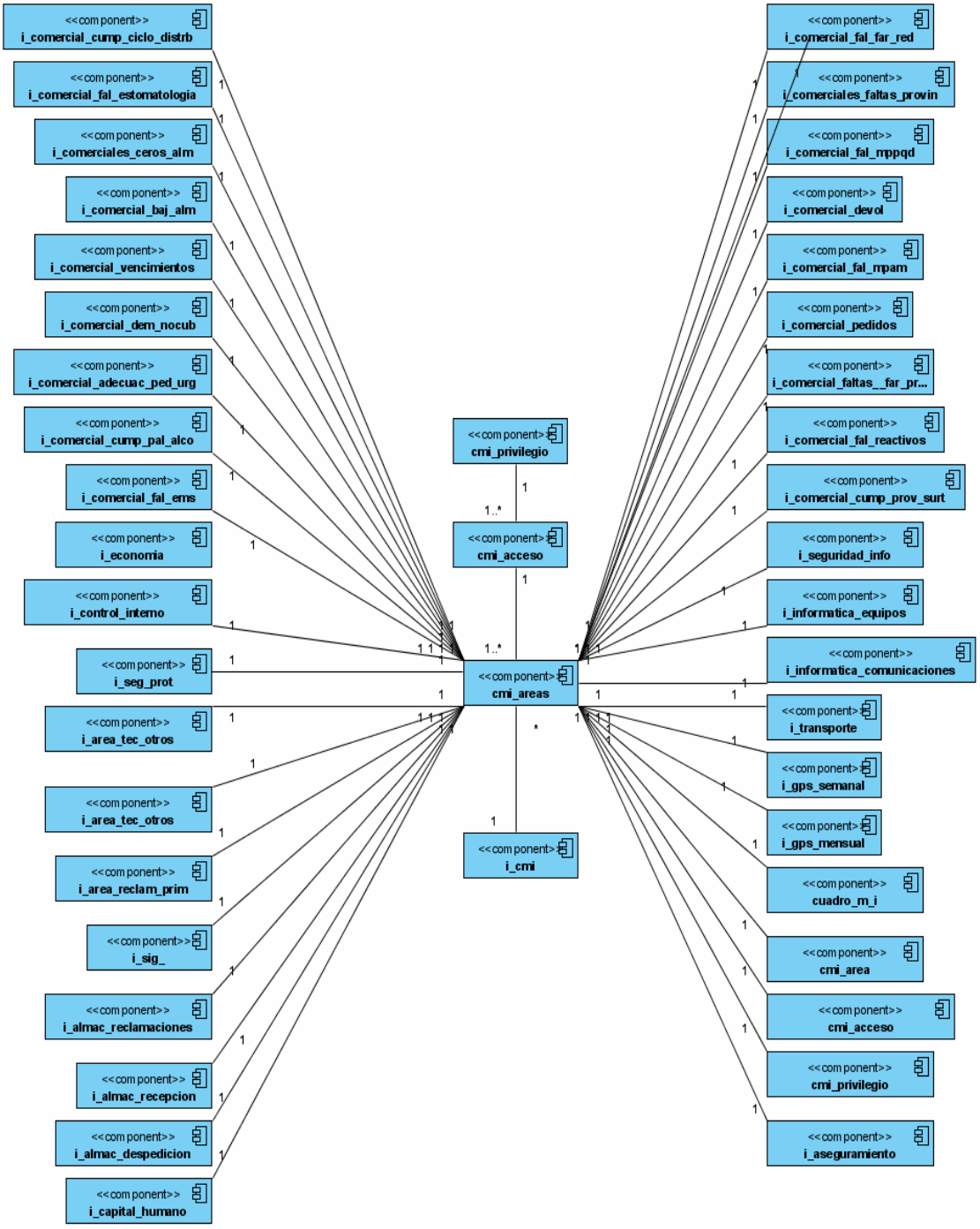


Figura 3.17 *Diagrama de Componentes*

3.2.6 Modelo de Prueba. Validación de los resultados obtenidos

Dado que el modelo de prueba es la garantía de la calidad del software, se desarrollaron comprobaciones de manera sistemática a los componentes ejecutables en el modelo de

implementación con pruebas de integración y de sistema. Los defectos detectados como resultado de las pruebas realizadas se tomaron en cuenta y a la vez fueron corregidos en iteraciones posteriores.

Por otra parte se realizaron pruebas con el propósito de obtener valoraciones conclusivas del sistema, a partir del procesamiento de la encuesta de opinión de los expertos mediante el método Delphi. Para la selección de éstos, se consultaron profesionales con experiencia, creatividad, capacidad de análisis y de pensamiento en el tema tratado. Se tuvo en cuenta, además, la disposición de éstos para responder la encuesta elaborada, para obtener a partir del procesamiento de la misma el coeficiente de competencia. Determinando este coeficiente, a partir de la opinión del experto sobre su nivel de conocimiento, acerca del problema que se está resolviendo y con las fuentes que le permiten argumentar sus criterios. Se escogieron sólo los de alta (5) y mediana (4) competencia para un total de 9 expertos seleccionados de los 15 encuestados, los cuales poseen preparación y conocimiento referente a los sistemas de bases de datos e interfaz gráfica de Usuario, con una experiencia mínima de 5 años.

Con el objetivo de evaluar el criterio de expertos, se realizaron encuestas para obtener el coeficiente de competencia en la interfaz visual (anexo1) y en el sistema de gestión de bases de datos (anexo2) del sitio Web Cuadro de Mando Integral. Una vez seleccionados los expertos se realizó una encuesta en el aspecto interfaz visual (anexo 3) y en sistemas de gestión de bases de datos del sistema (anexo5).

3.2.6.1 Encuestas

La encuesta aplicada (anexo3) consta de seis preguntas, procesadas por el método Delphi (anexo4), con el objetivo de encontrar el consenso de los encuestados en los aspectos:

1. Organización de componentes y ventanas de forma consistente.
2. Encontrar, buscar y manipular distintos objetos en la aplicación.
3. Interacción persona ordenador.
4. Proporcionar máxima funcionalidad (Eficiencia).
5. Usando asistentes realizar tareas de gran complejidad.
6. Permitir al usuario de forma intuitiva interactuar con la aplicación.

Después de realizar la encuesta para la GUI se llegó a la conclusión de que el sitio Web Sistema para el Control de los Indicadores de Gestión del Cuadro de Mando Integral (SCIGCMI) cumple con los siguientes parámetros del diseño de interfaz:

Bastante Relevante

Organización de componentes y ventanas de forma consistente.

Encontrar, buscar y manipular distintos objetos en la aplicación.

Muy Relevante

Interacción persona ordenador.

Proporcionar máxima funcionalidad (Eficiencia).

Usando asistentes realizar tareas de gran complejidad.

Permitir al usuario de forma intuitiva interactuar con la aplicación.

La otra encuesta aplicada (anexo 4) consta también de seis preguntas, procesadas por el método Delphi (anexo 5), con el objetivo de encontrar el consenso de los encuestados en los aspectos:

1. Proporciona una vista abstracta de los datos.
2. Almacenar y obtener los datos eficientemente.
3. Proporcionar integridad de los datos.
4. Definir control de acceso sobre diferentes clases de usuarios.
5. Permitir la protección de los usuarios de los efectos de las fallas del sistema.
6. Realizar consultas flexibles.

Después de realizar la encuesta para el DBMS se llegó a la conclusión de que el sitio Web Cuadro de Mando Integral cumple con los siguientes parámetros de la gestión de datos.

Muy Relevante

Proporciona una vista abstracta de los datos.

Almacenar y obtener los datos eficientemente.

Proporcionar integridad de los datos.

Permitir la protección de los usuarios de los efectos de las fallas del sistema.

Realizar consultas flexibles.

Bastante Relevante

Definir control de acceso sobre diferentes clases de usuarios.

3.3 Conclusiones

Al finalizar este capítulo se puede concluir que el sistema una vez implementado puede integrarse sin problemas a la intranet de la UEB Droguería Holguín.

Se recogen las etapas de la metodología que permitieron refinar los requisitos capturados, modelar y estructurar el sistema, así como razonar sus aspectos internos, lo que facilitó representar la correspondencia entre la arquitectura software y la arquitectura hardware, además de darle vida al sistema, a través de las actividades de implementación y el diseño definido para almacenar la información a persistir en el tiempo.

El diseño de interfaces y el manejo de errores estuvieron en concordancia con los propuestos en la fase de diseño y construcción del sistema. La satisfacción de los expertos fue percibida a través de encuestas que arrojaron que el sistema cumple con las expectativas de los mismos y resuelve el problema definido en esta investigación.

CONCLUSIONES

- ✓ Se logró un levantamiento amplio de los fundamentos teóricos que rigen las disciplinas relacionadas con los Indicadores de Gestión del Cuadros de Mando Integral.
- ✓ La metodología utilizada para el análisis, diseño y desarrollo de la aplicación resultó eficiente y queda disponible para su utilización en sistemas similares.
- ✓ Teniendo presente que el sitio Web Sistema para el Control de los Indicadores de Gestión de un Cuadro de Mando Integral es sostenible, se puede afirmar que el mismo evolucionará dando lugar a nuevas versiones y adaptándose a cambios necesarios e inevitables.
- ✓ Al finalizar la fase de elaboración del *Sistema para el Control de los Indicadores de Gestión de un Cuadro de Mando Integral* se concluye que se cumplió con el objetivo planteado al inicio de esta investigación y se demuestra la hipótesis supuesta.

RECOMENDACIONES

Sobre la base de la investigación realizada y la experiencia acumulada durante la realización de este trabajo, se presentan a continuación una serie de recomendaciones para una posterior ampliación, modificación, mejora y construcción de nuevas versiones des Cuadro de Mando Integral:

- ✓ Mantener actualizado el Sistema para el Control de los Indicadores de Gestión de un Cuadro de Mando Integral con los nuevos aportes y la asimilación de los cambios de la disciplina.
- ✓ Profundizar en el estudio de metodologías para la Ingeniería de Software para los sitios Web.
- ✓ Permitir al usuario la internacionalización del sistema.

GLOSARIO DE TÉRMINOS

Aplicación (Sistema): Sistema que ofrece a un usuario final un conjunto coherente de casos de uso.

Aplicación Web (Web-Application): Es un software o sistema que está basado en las especificaciones de la World Wide Web Consortium (W3C) y provee diferentes recursos como contenidos y servicios, los cuales se usan a través de una interfaz de usuario conocida como navegador Web.

Arquitectura: Conjunto de decisiones significativas acerca de la organización de un sistema software. La arquitectura del software se interesa no solo por la estructura y el comportamiento, flexibilidad al cambio, reutilización, comprensión, economía y tecnología, así como por aspectos estéticos.

Base de Datos: Es un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una base de datos puede considerarse una colección de datos variables en el tiempo.

Cliente Web (Web-Client): Software o aplicación que envía solicitudes a través del protocolo HTTP a un servidor Web.

CSS: Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML.

Despliegue: Ocurre cuando varios trabajos mas o menos independientes (lujos de control, procesos) se distribuyen entre diferentes dispositivos de hardware (procesadores).

Diagrama: Figura gráfica que representa las relaciones entre las diferentes partes de un conjunto o sistema.

Dirigido por los casos de uso: En el contexto del ciclo de vida del software, indica que los casos de uso se utilizan como artefacto principal para definir el comportamiento deseado para el sistema, y para comunicar este comportamiento entre las personas involucradas en el sistema.

Flujo de trabajo: Realización de un caso de uso o parte de el.

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 interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Hardware: Conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

HTML: Es el Lenguaje de Marcado de Hipertexto (HyperText Markup Language), que se utiliza para crear los documentos a los que se accede a través de navegadores World Wide Web.

HTTP: Es el Protocolo de Transferencia de Hipertexto (HyperText Transfer Protocol), un protocolo Web que controla las peticiones y servicios de documentos HTML.

ICONIX: Es una metodología ágil conducida por casos de uso para el desarrollo de software utilizada en proyectos de corto plazo y corto equipo.

IDE: Integrated Development Environment (Entorno de Desarrollo Integrado). Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación.

Implementación (Flujo de trabajo): Flujo fundamental que ofrece una vista de la arquitectura de un sistema. Abarcando los componentes usados para el ensamblado y lanzamiento del sistema físico.

Ingeniería: La ingeniería se define como la profesión en la cual los conocimientos de las matemáticas y las ciencias naturales obtenidos a través del estudio, la experiencia y la práctica, son aplicados con criterio y con conciencia al desarrollo de medios para utilizar económicamente con responsabilidad social y basados en una ética profesional, los materiales y las fuerzas de la naturaleza para beneficio de la humanidad. Las personas que se dedican a ella reciben el nombre de ingenieros.

Interfaz Gráfica de Usuario (GUI): Interfaz a través de la cual un usuario interactúa con un sistema.

Interfaz: Parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.

Java Beans: Son un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java.

JAVA: Es un lenguaje de programación diseñado para ser utilizado en la red, no está instalado localmente y se utiliza para dar a las páginas Web características extras no disponibles en HTML.

JSF: JavaServer Faces. Es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL.

JSP: JavaServer Pages. Es una tecnología Java que permite a los programadores generar dinámicamente HTML o algún otro tipo de página Web y permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático.

Middleware: Es una interfaz lógica estándar de los servicios de red.

Motor de Base de Datos: Es el proceso que ocurre cuando el software (SGBD) accede a la base de datos física, en fin el servidor de base de datos es la computadora en la cual el motor de base de datos está corriendo.

Multiplataforma: Indica la capacidad o característica de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas. Por ejemplo la posibilidad de utilizar un programa o software determinado en sistemas Windows y Linux.

MVC: Modelo Vista Controlador. Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.

Navegador Web (Web-Browser): Software o aplicación (Cliente) con la capacidad de mostrar documentos HTML y otros contenidos Web.

Página Principal (Homepage): Página principal de varias páginas que se interrelacionan y comparten contenido.

Página Web (Web-Page): Es una página sencilla en HTML que puede incluir diferentes tipos de documentos (Videos, sonido).

Patrón de diseño: Solución apropiada a un problema común en un contexto, es una solución completa que incluye análisis, diseño e implementación.
Patrón: Solución común a un problema común de un determinado contexto.

Plataforma: Basamento, ya sea de hardware o software, sobre el cual un programa puede ejecutarse. Ejemplos típicos incluyen: arquitectura de hardware, sistema operativo, lenguajes de programación y sus librerías de tiempo de ejecución.

Portabilidad: Grado en el que un sistema en funcionamiento en un determinado entorno de ejecución, puede ser convertido fácilmente en un sistema funcionando en otro entorno de ejecución.

Programación: Se llama programación al acto de crear un programa de computadora, un conjunto concreto de instrucciones que una computadora puede ejecutar. El programa se escribe en un lenguaje de programación, aunque también se pueda escribir directamente en lenguaje de máquina, con cierta dificultad. Un programa se puede dividir en diversas partes, que pueden estar escritas en lenguajes distintos.

Programador: Un programador es una persona que se dedica a la programación o que escribe programas. También puede decirse que es la persona que traduce algoritmos a un lenguaje que pueda entender una computadora.

Protocolo: Es una serie de reglas que utilizan dos computadoras para comunicarse entre sí. Cualquier producto que utilice un protocolo dado puede funcionar con otros productos que utilicen el mismo protocolo.

Pruebas (Flujo de Trabajo): Flujo de trabajo fundamental cuyo propósito esencial es comprobar el resultado de la implementación mediante las pruebas de cada construcción.

Requerimiento: Variable de un proyecto que pone en peligro i impide su éxito.

Requerimientos (Flujo de trabajo): Flujo de trabajo fundamental cuyo propósito esencial es orientar el desarrollo hacia el sistema correcto.

Reusabilidad: Capacidad de ser modificable por diferentes herramientas.

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

Servidor Web (Web-Server): Software o aplicación, que espera solicitudes de los clientes y envía las respuestas a tavés del protocolo HTTP.

Servidor Web: Es un programa que implementa el protocolo HTTP (Hypertext Transfer Protocol), diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (Hypertext Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Servlets: Son objetos que corren dentro del contexto de un Servidor Web y extienden su funcionalidad. Su uso más común es generar páginas Web de forma dinámica a partir de los parámetros de la petición que envíe el navegador Web.

Sistema de Base de Datos: Conjunto de programas que permiten la implantación, acceso y mantenimiento de la base de datos así como la Base de Datos física y los usuarios que interactúan con esta.

Sistema: Conjunto de componentes interrelacionados entre si que actúan bajo determinadas leyes.

Sitio Web (Web-Site): Conjunto de múltiple páginas que normalmente pertenecen a una compañía de negocios o institución.

Software: Conjunto de programas que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Es el conjunto de instrucciones que permite la utilización del equipo.

SQL: Lenguaje de Consultas Estructurado (Structured Query Language). Lenguaje de consultas y programación de BD que desarrolló originalmente IBM para grandes sistemas. Se utiliza ampliamente para tener acceso a datos, consultar, actualizar y administrar sistemas de bases de datos relacionales. En la actualidad hay una definición de SQL de estándar ANSI para todos los sistemas.

TCP/IP: Protocolo de Control de Transmisión / Protocolo de Internet (Transmission Control Protocol/Internet Protocol): es el conjunto de protocolos que rigen todas las comunicaciones entre todas las computadoras en Internet.

Tomcat: Funciona como un contenedor de Servlets desarrollados. Implementa las especificaciones de los Servlets y de JavaServer Pages (JSP) de Sun Microsystems.

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.

Usuario Web (Web-User): Persona que utiliza los servicios que ofrece la Web.

Usuario: Humano que interactúa con un sistema.

Versión: Conjunto de artefactos relativamente completo y consistente entregado a un usuario interno o externo.

XP: Extreme Programing, metodología ágil para el desarrollo de software utilizada en proyectos de corto plazo y corto equipo.

BIBLIOGRAFÍA Y REFERENCIAS BIBLIOGRÁFICAS

[1] César Manuel Álvarez Borbolla y Yuriel Noa Góngora [2005]. Diseño de un Cuadro de Mando Integral para las Empresas turísticas su desarrollo en el Hotel Pernil de L Cadena de Turismo ISLAZUL, división. Tesis presentada para opción al titulo de Ingeniero Industrial. TUTOR. Dr. Guillermo Armando Ronda Pupo. / CEGEM. Uho, 95 pp.

[2] Robert S. Kaplan y David P. Norton Creando la organización focalizada en la estrategias con el balanced scorecard" (The Balanced Scorecard Collaborative (www.bscol.com).

[3] Apuntes sobre la Dirección Estratégica. ¿Cómo Intregar los niveles Estratégicos, Táctico y Operativo? Autor: Guillermo Armando Ronda Pupo y José Miguel Marcané Lacera [2003].

[4] Jordi Masi Hernández. Software libre: técnicamente viable, económicamente sostenible y socialmente justo primera Edición 2005.

[5] Framework [documento en línea]. http://es.wikipedia.org/wiki/Framework. (Consulta: 10/02/2010).

[6] Framework [documento en línea]. http://es.wikipedia.org/wiki/Framework. (Consulta: 10/02/2010).Buscar el sitio de Code Igniter.

[7] Renaud Paul E. (2nd ed). (1996) Introduction to client/server systems. A practical guide for systems professionals. John Wiley &Sons New York, USA.

[8] Paul J. Deitel, Harvey M. Deitel;(2008) AJAX, Rich Internet Applications, and Web Development for Programmers: DEITEL® DEVELOPER SERIES. Published by Prentice-Hall. ISBN 0-13-158738-2.

[9] Free Software Foundation Web Site. http://www.fsf.org 23/octubre/2009.

[10] Andrade Carmen y García Ludi. Ventajas del Software Libre. http://www.lambdaux.com.12/nov/2009.

[11] Rodríguez C.J., (2006) Propuesta de un Framework para el desarrollo de Sistemas Colaborativos Distribuidos. Trabajo para optar por el Título de Ingeniero Informático Universidad de Holguín.

[12] Álvarez, Miguel. ¿Qué es Perl?, http://www.desarrolloweb.com, 3/dic/2009.

[13] Fahnle, Pablo. ¿Qué es ASP? http://www.programacion.com, 3/diciembre 2009.

[14] Lenguajes de Programación (PHP): Programación Web. http://lenguajesdeprogramacion.com/programacion-web.shtml .3/dic/2009.

[15] Chopra, V; Sing, L; Rupert, J; Jon, E; Bell, J (2005) Beginning JavaServer Pages. Published by Wiley Publishing, Inc. 10475 Crosspoint Boulevard Indianapolis,Indiana IN 46256. ISBN: 0-7645-7485-X.

[16] Mann, K(2005) JavaServer Faces in Action. Published by Manning Publications Co. ISBN 1-932394-11-7.

[17] Oracle Web Site. http://oracle.org/. 23/sept/2009.

[18] SQL Server Web Site. http://hsqldb.org/ 23/sept/2009.

[19] HSQLDB Web Site. http://hsqldb.org/. 23/sept/2009.

[20] PostgreSQL Web Site. http://www.pgsql.com/ .23/sept/2009.

[21] MySQL Web Site. http://www.mysql.com/ .23/sept/2009.

[22] Fowler, M. The New Methodology. ThoughWorks, 2003.

[23] Mendoza, M. Metodologías de desarrollote software. http://www.informatizate.net, Elaborado 2004.3/ene/2008.

[24] Escofet, E: Diseño de una Aplicación Groupware de Consultas, Tesis DEA. Universidad de Granada, España, 2004.

[25] Rosenberg, Doug and Scott, K. Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example. Addison Wesley, 2001.

[26] Desarrollo e implementación de un Sistema de Control Gerencial basado en el BSC y conducción del proceso de planificación. Boris Albarracín Espinoza, Rubén Rivera Gómez, Jaime Lozada.

[27] http://www.monografias.com/trabajos69/sistema-control-gestion-enfoque-proceso/sistema-control-gestion-enfoque-proceso2.shtml . Alicia Rodriguez Ocampo.

[28] http://www.qpr.com.

[29] http://www.deinsa.com.

[30] http://www.spimpact.com.

[31] http://www.pilotsofware.com.

[32] http://www.desarrolloweb.com/articulos/codeigniter.html.

[33] http://www.vivaphp.com.ar/eventos/rasmus-lerdorf-sobre-los-frameworks.

[34] Rodríguez Fórtiz, M. J. y Parets Llorca, J. Evolución de Sistemas Software. Actas del taller de Evolución del Software. Ciudad Real. Noviembre 2001.

[35] Rodríguez Fórtiz, María José. Evolución del Software: Una Formalización Basada en Lógica Temporal de Predicados y Redes de Petri Coloreadas. Tesis Doctoral. Universidad de Granada, 2000.

[36]

http://www.oect.es/portal/site/Observatorio/menuitem.1a9b11e0bf717527e0f945100bd061ca/?vgnextoid=b80b5052be683110VgnVCM100000dc0ca8c0RCRD&vgnextchannel=e68f6a5f01d63110VgnVCM100000dc0ca8c0RCRD.

ANEXOS

Anexos:

ENCUESTA PARA DETERMINAR EL COEFICIENTE DE COMPETENCIA DEL EXPERTO.

Nombre y apellidos: _____

Usted ha sido seleccionado para ser consultado respecto al grado de relevancia de un modelo de competencia para valorar el sitio Web Sistema de Control para los Indicadores de Gestión del Cuadro Mando Integral que es una herramienta de trabajo que permite evaluar el comportamiento de los indicadores de la empresa EMCOMED.

Necesitamos antes de realizarle la consulta correspondiente como parte del método empírico de investigación "consulta a expertos", determinar su coeficiente de competencia en este tema, a los efectos de reforzar la validez del resultado de la consulta que realizaremos. Por esta razón le rogamos que responda las siguientes preguntas de la forma más objetiva que le sea posible.

1.- Marque con una cruz (X), en la tabla siguiente, el valor que se corresponde con el grado de conocimientos que usted posee sobre el tema **"Diseño de Interfaz Gráfica de Usuario"**. Considere que la escala que le presentamos es ascendente, es decir, el conocimiento sobre el tema referido va creciendo desde 0 hasta 10.

0	1	2	3	4	5	6	7	8	9	10

2.- Realice una autovaloración del grado de influencia que cada una de las fuentes que le presentamos a continuación, ha tenido en su conocimiento y criterio sobre el **"diseño de interfaz de usuario"**. Para ello marque con una cruz (X), según corresponda, en **A** (alto), **M** (medio) o **B** (bajo).

Fuentes de argumentación.	Grado de influencia de cada una de las fuentes.		
	A (alto)	**M**(medio)	**B** (bajo)
Análisis teórico realizado por usted.			
Su experiencia obtenida.			
Trabajo de autores nacionales.			
Trabajo de autores extranjeros.			
Su propio conocimiento del estado del problema en			

el extranjero.			
Su intuición.			

Anexo 1. *Encuesta para determinar el coeficiente de competencia en GUI.*

ENCUESTA PARA DETERMINAR EL COEFICIENTE DE COMPETENCIA DEL EXPERTO.

Nombre y apellidos: _____

Usted ha sido seleccionado para ser consultado respecto al grado de relevancia de un modelo de competencia para valorar el sitio Web Cuadro Mando Integral que es una herramienta de trabajo que permite evaluar el comportamiento de los indicadores de la empresa EMCOMED.

Necesitamos antes de realizarle la consulta correspondiente como parte del método empírico de investigación "consulta a expertos", determinar su coeficiente de competencia en este tema, a los efectos de reforzar la validez del resultado de la consulta que realizaremos. Por esta razón le rogamos que responda las siguientes preguntas de la forma más objetiva que le sea posible.

1.- Marque con una cruz (X), en la tabla siguiente, el valor que se corresponde con el grado de conocimientos que usted posee sobre el tema **"Sistemas de Gestión de Bases de Datos"**. Considere que la escala que le presentamos es ascendente, es decir, el conocimiento sobre el tema referido va creciendo desde 0 hasta 10.

0	1	2	3	4	5	6	7	8	9	10

2.- Realice una autovaloración del grado de influencia que cada una de las fuentes que le presentamos a continuación, ha tenido en su conocimiento y criterio sobre el **"Sistemas de Gestión de Bases de Datos"**. Para ello marque con una cruz (X), según corresponda, en **A** (alto), **M** (medio) o **B** (bajo).

Fuentes de argumentación.	Grado de influencia de cada una de las fuentes.		
	A (alto)	**M**(medio)	**B** (bajo)
Análisis teórico realizado por usted.			
Su experiencia obtenida.			
Trabajo de autores nacionales.			
Trabajo de autores extranjeros.			

Su propio conocimiento del estado del problema en el extranjero.			
Su intuición.			

Anexo 2. *Encuesta para determinar el coeficiente de competencia.*

ENCUESTA A EXPERTOS EN (GUI).

Nombre y apellidos: _____

Institución a la que pertenece: _____

Cargo actual: _____

Calificación profesional, grado científico o académico:

Profesor: _____.

Licenciado: _____.

Especialista: _____.

Master: _____.

Doctor: _____.

Años de experiencia en el cargo: _____.

Años de experiencia docente y/o en la investigación: _____.

A continuación le presentamos una tabla que contiene los aspectos fundamentales que debe cumplir el diseño de interfaz gráfica del sitio Web Cuadro de Mando Integral con el objetivo de valorar sobre estos principios la aplicación

A la derecha aparece la escala:

MR: Muy relevante. **BR:** Bastante relevante. **R: Relevante**.

PR: Poco relevante **NR: No** relevante.

- Marque con una cruz (X) en la celda que se corresponda con el grado de relevancia que usted otorga a cada aspecto que se refleja en la interfaz gráfica de usuario del sitio Web Cuadro de Mando Integral. Le agradecemos anticipadamente el esfuerzo que sabemos hará para responder, con la mayor fidelidad posible a su manera de pensar la presente encuesta.

Valoración de la interfaz gráfica del Cuadro de Mando Integral					
	MR	BR	R	PR	NR
Permite la organización de componentes y ventanas de forma consistente. (organización lógica)					

Permite interacción persona ordenador (grado de reacción)					
Permite proporcionar máxima funcionalidad (Eficiencia)					
Permite usando asistentes realizar tareas de gran complejidad					
Permite al usuario de forma intuitiva interactuar con la aplicación (Predecibilidad)					
Permite encontrar , buscar y manipular distintos objetos en la aplicación (Escalabilidad)					

- Escriba a continuación que características deben ser incluidos o eliminados en está propuesta:

Características que se proponen ser incluidos	Características se proponen ser eliminados

- Señale a continuación, si considera que alguna de los aspectos a tener en cuenta , debe ser cambiada:

El aspecto aparece como	El aspecto debe ser cambiado por

- Otra sugerencia que usted desee hacer sobre la interfaz gráfica de usuario del sitio Web Cuadro Mando Integral que estamos sometiendo a su consideración.

Anexo 3. *Encuesta a Expertos en GUI (Graphic User Interface).*

PROCESAMIENTO DE LA ENCUESTA

TABLA DE FRECUENCIA ABSOLUTA						
	MR	BR	R	PR	NR	TOTAL
1	5	2	2	0	0	9
2	5	2	2	0	0	9
3	4	5	0	0	0	9
4	4	4	1	0	0	9
5	6	2	1	0	0	9
6	6	1	2	0	0	9

TABLA DE FRECUENCIA ABSOLUTA ACUMULADA					
	MR	BR	R	PR	NR
1	5	7	9	9	9
2	5	7	9	9	9
3	4	9	9	9	9
4	4	8	9	9	9
5	6	8	9	9	9
6	6	7	9	9	9

TABLA DEL INVERSO DE LA FRECUENCIA ABSOLUTA ACUMULADA				
	MR	BR	R	PR
1	0,5556	0,7778	1	1
2	0,5556	0,7778	1	1
3	0,4444	1	1	1
4	0,4444	0,8889	1	1
5	0,6667	0,8889	1	1
6	0,6667	0,7778	1	1

TABLA DE DETERMINACION DE LOS PUNTOS DE CORTE

	MR	BR	R	PR	Suma	Promedio	N - Prom.
1	0,14	0,76	3,49	3,49	7,88	1,97	0,15
2	0,14	0,76	3,49	3,49	7,88	1,97	0,15
3	-0,14	3,49	3,49	3,49	10,33	2,58	-0,46
4	-0,14	1,22	3,49	3,49	8,06	2,02	0,1
5	0,43	1,22	3,49	3,49	8,63	2,16	-0,04
6	0,43	0,76	3,49	3,49	8,17	2,04	0,08

Suma	0,86	8,21	20,94	20,94	50,95		
Punto de corte	0,14	1,37	3,49	3,49	8,49	2,12	= N (Prom. Gen.)
		0,14	1,37	3,49	3,49		

CONCLUSIONES					
	MR	BR	R	PR	NR
1	-	SI	-	-	-
2	-	SI	-	-	-
3	SI	-	-	-	-
4	SI	-	-	-	-
5	SI	-	-	-	-
6	SI	-	-	-	-

Anexo 4. *Procesamiento de la encuesta de opinión de expertos en GUI aplicando el método Delphi.*

ENCUESTA A EXPERTOS EN (SGBD).

Nombre y apellidos: _____

Institución a la que pertenece: _____

Cargo actual: _____

Calificación profesional, grado científico o académico:

Profesor: _____.

Licenciado: _____.

Especialista: _____.

Master: _____.

Doctor: _____.

Años de experiencia en el cargo: _____.

Años de experiencia docente y/o en la investigación: _____.

A continuación le presentamos una tabla que contiene los aspectos fundamentales que debe cumplir todo SGBD (Database Management System) con el objetivo de valorar sobre estos principios el software derbyStor.

A la derecha aparece la escala:

MR: Muy relevante. BR: Bastante relevante. R: Relevante.

PR: Poco relevante **NR: No** relevante.

- Marque con una cruz (X) en la celda que se corresponda con el grado de relevancia que usted otorga a cada aspecto que se refleja en el cliente Web derbyStor .Le agradecemos anticipadamente el esfuerzo que sabemos hará para responder, con la mayor fidelidad posible a su manera de pensar la presente encuesta.

Valoración del Cliente Web derbyStor					
	MR	BR	R	PR	NR
Proporciona una vista abstracta de los datos (Independencia de los datos).					
Permite almacenar y obtener los datos eficientemente					
Proporciona integridad de los datos					
Permite definir control de acceso sobre diferentes clases de usuarios					
Permite la protección de los usuarios de los efectos de las fallas del sistema					
Permite realizar salvas de BBDD flexibles					

- Escriba a continuación que características deben ser incluidos o eliminados en está propuesta:

Características que se proponen ser incluidos	Características se proponen ser eliminados

- Señale a continuación, si considera que alguna de los aspectos a tener en cuenta , debe ser cambiada:

El aspecto aparece como	El aspecto debe ser cambiado por

- Otra sugerencia que usted desee hacer sobre Sistema de Gestión del Cuadro de mando Integral que estamos sometiendo a su consideración.

Anexo 5. *Encuesta a Expertos en SGBD (Data Base Management System).*

PROCESAMIENTO DE LA ENCUESTA

TABLA DE FRECUENCIA ABSOLUTA						
	MR	BR	R	PR	NR	TOTAL

1	5	2	2	0	0	9
2	5	4	0	0	0	9
3	5	3	1	0	0	9
4	6	1	1	1	0	9
5	5	2	2	0	0	9
6	5	4	0	0	0	9

TABLA DE FRECUENCIA ABSOLUTA ACUMULADA					
	MR	BR	R	PR	NR
1	5	7	9	9	9
2	5	9	9	9	9
3	5	8	9	9	9
4	6	7	8	9	9
5	5	7	9	9	9
6	5	9	9	9	9

TABLA DEL INVERSO DE LA FRECUENCIA ABSOLUTA ACUMULADA				
	MR	BR	R	PR
1	0,5556	0,7778	1	1
2	0,5556	1	1	1
3	0,5556	0,8889	1	1
4	0,6667	0,7778	0,8889	1
5	0,5556	0,7778	1	1
6	0,5556	1	1	1

TABLA DE DETERMINACIONDE LOS PUNTOS DE CORTE							
	MR	BR	R	PR	Suma	Promedio	N - Prom.
1	0,14	0,76	3,49	3,49	7,88	1,97	0,16
2	0,14	3,49	3,49	3,49	10,61	2,65	-0,52
3	0,14	1,22	3,49	3,49	8,34	2,09	0,04
4	0,43	0,76	1,22	3,49	5,9	1,48	0,65

5	0,14	0,76	3,49	3,49	7,88	1,97	0,16
6	0,14	3,49	3,49	3,49	10,61	2,65	-0,52

Suma	1,13	10,48	18,67	20,94	51,22		
Punto de corte	0,19	1,75	3,11	3,49	8,54	2,13	= N (Prom. Gen.)
		0,19	1,75	3,11	3,49		

CONCLUSIONES					
	MR	BR	R	PR	NR
1	SI	-	-	-	-
2	SI	-	-	-	-
3	SI	-	-	-	-
4	-	SI	-	-	-
5	SI	-	-	-	-
6	SI	-	-	-	-

Anexo 6. *Procesamiento de la encuesta de opinión de expertos en DBMS aplicando el método Delphi.*