

Universidad de las Ciencias Informáticas

Facultad 3



“Gestor de reglas y criterios de medidas para evaluar el cumplimiento de objetivos estratégicos en el Sistema de Planificación de Actividades SIPAC”.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Autor(es): Lisandra Santamaria Pérez
Alejandro José Rodríguez Cruz

Tutor(es): Ing. Mairelys Fernández González
Ing. Ariadna Rendón Artola

La Habana, Mayo de 2014

“Año 56 de la Revolución”

DECLARACIÓN DE AUTORÍA

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

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

Lisandra Santamaria Pérez

Autor

Alejandro José Rodríguez Cruz

Autor

Mairelys Fernández González

Tutor

Ariadna Rendón Artola

Tutor

AGRADECIMIENTOS

Agradezco a mi familia por todo el amor y comprensión que me brindaron, en especial a mi mamá por ser el ejemplo que siempre guiará todos mis pasos, a mi hermana Sandra Elena por su apoyo y comprensión, a Nana por ser mi segunda madre y brindarme su cariño, a Tatatico, mi hermano de corazón, a mi abuelita Pipa que nunca olvidaré.

A los amigos que hicieron inolvidable mi estancia en esta Universidad, en especial a Diana y Greter por aguantarme y quererme como soy. A mis compañeros de aula y profesores por convertirse en mi familia.

A mis tutoras Mairelys y Ariadna por la exigencia y el apoyo brindado. Y a Alejandro por haber sido el mejor compañero de tesis que he tenido.

...Lisandra Santamaria Pérez...

Quiero agradecer a todas aquellas personas que formaron parte de mi vida a lo largo de la carrera, a mis profesores, a mis tutores Mairelys, Ariadna. A mis compañeros de aula del 15112, del 3209, del 3505. a los profesionales del Sistema de planificación de actividades SIPAC. A mis amigos Carlos, Yandry, Dian, Rafa, Rubén, Lili, Rinat. A mi familia a mi hermanos May, Ely y Huguito, a mi padre Hugo. A mi abuela Elena, a Damaris, a mis tíos Kano, Omar, Eddy, Marlenis, Marisa y Maira. A mi compañera de tesis Lisandra.

....Alejandro José Rodríguez Cruz....

DEDICATORIA

El presente trabajo de diploma va dedicado a mi madre, por ser la persona que me ama por encima de su propia vida, a pesar de mis defectos. Por enseñarme que nada es imposible cuando de hacer un sueño realidad se trata. Por ser padre, amiga, hermana. Por demostrarme lo que es el amor incondicional y por llenar mi vida de tantas alegrías y vivencias que siempre guardaré en lo más profundo de mi corazón.

....Lisandra Santamaria Pérez....

Este trabajo se lo dedico a Magdeline Cruz Ricardo, me siento muy orgulloso de ser tu hijo, gracias por todo el amor, paciencia, apoyo y ternura que me has dedicado. Has sido para mí una amiga, una hermana, un padre, una madre que siempre ha estado a mi lado guiándome y cuidando de mí, muchas gracias de todo corazón.

....Alejandro José Rodríguez Cruz....

RESUMEN

El Sistema de Planificación de Actividades (SIPAC) constituye una herramienta para la gestión de las actividades a todos los niveles organizacionales, basado en la Instrucción no.1 del Presidente de los Consejos de Estado y de Ministros para la planificación en Cuba. Permite interrelacionar objetivos de trabajo y actividades en tiempo real; garantizando el seguimiento del desarrollo y cumplimiento de los objetivos y tareas principales en las entidades como parte de la Planeación Estratégica y Operativa. Independientemente de las funcionalidades que brinda el sistema hoy, no posibilita evaluar el cumplimiento de los objetivos estratégicos a partir de indicadores y criterios de medidas previamente definidos, lo que dificulta el seguimiento y control de la planificación tanto para la entidad como a nivel de gobierno. El presente trabajo comprende el desarrollo del Gestor de reglas y criterios de medidas para evaluar el cumplimiento de objetivos, solución informática en la que se definen indicadores que cualifican el cumplimiento de los objetivos trazados en el plan de la organización, estableciendo un balance del cumplimiento de los indicadores y criterios de medidas, así como la estimación de los posibles riesgos en los procesos de Ejecución y Control de la planificación.

Palabras claves: criterios de medida, evaluación, indicadores, objetivos, reglas.

ÍNDICE

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Introducción	6
1.2 Marco Conceptual	6
1.2.1 Planificación	6
1.2.2 Objetivo	6
1.2.3 Indicador	7
1.2.4 Criterio de medida	7
1.3 Estudio del Estado del Arte	8
1.3.1 Metodología para la evaluación de criterios de medida definidos para alcanzar los objetivos estratégicos del Ministerio de Salud Pública	8
1.3.2 Elementos del Sistema de Planificación establecidos en el Reglamento para la Implantación y Consolidación del Sistema de Dirección y Gestión Empresarial Estatal	8
1.3.3 Indicadores para el seguimiento de los objetivos de desarrollo del milenio	9
1.3.4 Evaluación del cumplimiento de objetivos en los sistemas de incentivación del desempeño	10
1.3.5 Modelo de evaluación de la aplicación de la Instrucción No.1 del Presidente de los Consejos de Estado y de Ministros para la planificación en Cuba	10
1.3.6 Soluciones informáticas para evaluar el cumplimiento de objetivos	11
1.3.7 Resumen de soluciones informáticas para evaluar el cumplimiento de objetivos	12
1.4 Valoración del Estado del Arte	13
1.5 Lenguajes, tecnologías y herramientas propuestos para el desarrollo de la solución ..	14
1.5.1 Lenguajes de modelado	14
1.5.2 Lenguajes de Desarrollo del lado del cliente	15
1.5.3 Lenguajes de Desarrollo del lado del servidor	15
1.5.4 Tecnologías y herramientas de desarrollo	16
1.5.5 Modelo de desarrollo	19
1.6 Conclusiones del capítulo	20
CAPÍTULO 2: ANÁLISIS Y DISEÑO	22

ÍNDICE

2.1	Introducción	22
2.2	Propuesta de solución.....	22
2.3	Modelado del negocio	23
2.3.1	Diagrama de procesos de negocio	23
2.3.2	Especificación del proceso de negocio: Evaluar el cumplimiento de los objetivos de un pla.....	24
2.3.3	Validación del proceso de negocio	26
2.3.4	Modelo conceptual	26
2.4	Requisitos	27
2.4.1	Técnicas de captura de requisitos	28
2.4.2	Listado de requisitos funcionales.....	28
2.4.3	Especificación de requisitos funcionales.....	29
2.4.4	Validación de los requisitos funcionales	31
2.4.5	Requisitos no funcionales.....	31
2.5	Modelado de la solución.....	32
2.5.1	Diagrama de clases del diseño web	32
2.5.2	Diagrama de secuencia.....	35
2.5.3	Modelo de datos.....	36
2.5.4	Diagrama de componentes.....	37
2.6	Patrones del diseño.....	39
2.6.1	Patrones GRASP	40
2.6.2	Patrones GoF (Gang Of Four)	41
2.7	Conclusiones del capítulo	41
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN		43
3.1	Introducción	43
3.2	Implementación de la solución propuesta.....	43
3.2.1	Evaluación de un objetivo en la solución implementada	43
3.2.2	Estándares de codificación.....	48
3.2.3	Diagrama de despliegue.....	49
3.3	Validación de la solución propuesta	50

ÍNDICE

3.3.1 Validación del diseño propuesto	50
3.3.2 Pruebas de software	55
3.3.3 Resultados de las pruebas aplicadas	61
3.4 Conclusiones del capítulo	62
CONCLUSIONES GENERALES	64
RECOMENDACIONES	65
BIBLIOGRAFÍA REFERENCIADA	66
BIBLIOGRAFÍA CONSULTADA	69
GLOSARIO DE TÉRMINOS	70
ANEXOS	72

ÍNDICE DE TABLAS

Tabla 1. Sistema de calificación del Plan de Desarrollo Estratégico del MINSAP	8
Tabla 2. Resumen de soluciones informáticas para evaluar el cumplimiento de objetivos.....	12
Tabla 3. Especificación del proceso de negocio: Evaluar el cumplimiento de los objetivos de un plan.	24
Tabla 4. Especificación del requisito: Adicionar regla	29
Tabla 5. Descripción de las principales clases del diagrama de clases del diseño	34
Tabla 6 . Caso de prueba para el camino básico # 1	59
Tabla 10. Caso de prueba de caja negra para validar el requisito funcional Adicionar Regla	60
Tabla 8. Material de medición de la métrica Tamaño operacional de la clase (TOC).....	76
Tabla 9. Material de medición de la métrica Relaciones entre clases (RC).....	77
Tabla 10. Caso de prueba para el camino básico # 2.....	77
Tabla 11. Caso de prueba para el camino básico # 3.....	78
Tabla 12. Caso de prueba para el camino básico # 4.....	79
Tabla 13. Juego de datos a probar para el caso de prueba del requisito funcional: Adicionar reglas.	79

ÍNDICE DE FIGURAS

Figura 1. Propuesta de solución para evaluar el cumplimiento de los objetivos en el SIPAC	23
Figura 2. Proceso de negocio: Evaluar el cumplimiento de los objetivos de un plan.....	24
Figura 3. Modelo conceptual del Gestor de reglas y criterios de medida para la evaluación del cumplimiento de objetivos.	27
Figura 5. Modelo de clases del diseño del gestor de reglas y criterios de medida para evaluar el cumplimiento de objetivos.	34
Figura 6. Diagrama de secuencia Adicionar Indicador.....	35
Figura 7. Modelo de datos actualizado para el Gestor de reglas y criterios de medida para evaluar el cumplimiento de objetivos estratégicos en el SIPAC.....	37
Figura 8. Representación del subsistema Planificación con sus componentes.....	38
Figura 9. Diagrama del componente Planeación en el sistema SIPAC.....	39
Figura 10. Configurar parámetros de evaluación de un objetivo.	44
Figura 11. Adicionar criterio de medida	45
Figura 12. Adicionar regla	46
Figura 13. Adicionar indicador para evaluación cuantitativa	46
Figura 14. Registro de objetivos	47
Figura 15. Diagrama de despliegue de escenario para PC cliente con disco.....	49
Figura 16. Diagrama de despliegue de escenario para PC cliente sin disco.....	50
Figura 17. Representación de la cantidad de clases y el número de procedimientos que contienen.....	51
Figura 18. Representación de la cantidad de clases y el número de procedimientos que contienen.....	51
Figura 19. Representación del valor en % del atributo Responsabilidad.	52
Figura 20. Representación del valor en % del atributo Complejidad de implementación.	52
Figura 21. Representación del valor en % del atributo Reutilización.....	53
Figura 22. Representación de las asociaciones de uso por cantidad de clase.	53
Figura 23. Representación de las asociaciones de uso por cantidad de clase.	54
Figura 24. Representación del valor en % del atributo Acoplamiento.	54
Figura 25. Representación del valor en % del atributo Complejidad de Mantenimiento.	54
Figura 26. Representación del valor en % del atributo Cantidad de pruebas.....	55
Figura 27. Representación del valor en % del atributo Reutilización.....	55

Figura 28. Grafo de flujo asociado al procedimiento <i>modificarindicadorAction</i>	58
Figura 29. No conformidades detectadas durante las iteraciones de pruebas realizadas.	62
Figura 29. <i>Vista</i> del patrón arquitectónico Modelo-Vista-Controlador del diseño de clases.	72
Figura 30. <i>Controlador</i> del patrón arquitectónico Modelo-Vista-Controlador del diseño de clases.	72
Figura 31. <i>Modelo</i> del patrón arquitectónico Modelo-Vista-Controlador del diseño de clases....	73
Figura 32. Adicionar elemento a regla	74
Figura 33. Asociar criterio de medida a regla	74
Figura 34. Adicionar indicador para evaluación cualitativa	75
Figura 35. Configurar rangos para la evaluación del objetivo	75
Figura 36. Fragmento de código con el algoritmo <i>modificarindicadorAction</i>	76

INTRODUCCIÓN

En la sociedad cubana actual, la planificación, más allá de constituir un proceso organizativo que utilizan las empresas para cumplir sus objetivos a corto, mediano y largo plazo, es un proceso continuo e ininterrumpido que expresa la voluntad e iniciativas de las empresas por obtener un desarrollo económico y social que dé cumplimiento a sus objetivos generales con la mayor eficiencia posible.

La actualización del modelo económico cubano y con ello la puesta en marcha de la implementación de los Lineamientos de la Política Económica y Social del Partido y la Revolución, hace necesario mantener un seguimiento y control eficaz sobre el cumplimiento de los objetivos trazados en los diferentes organismos e instituciones estatales. Para ello, el Centro de Informatización de Entidades (CEIGE) de la Universidad de Ciencias Informáticas lleva a cabo actualmente el desarrollo del Sistema de Planificación de Actividades (SIPAC 2.0).

SIPAC constituye una herramienta para la gestión de las actividades a todos los niveles organizacionales, basado en la Instrucción No.1 del Presidente de los Consejos de Estado y de Ministros para la planificación en Cuba¹. Permite interrelacionar objetivos de trabajo y actividades en tiempo real; garantizando el seguimiento del desarrollo y cumplimiento de los objetivos y tareas principales en las entidades como parte de la Planeación Estratégica y Operativa. Cuenta con cuatro módulos (Configuración, Planeación, Notificaciones y Recuperaciones) encargados de generar las configuraciones necesarias para el seguimiento de las tareas principales de cada entidad, gestión de los posibles involucrados, nomencladores y niveles de subordinación basados en reglas de la compartimentación de la información.(1)

Independientemente de las funcionalidades que brinda el sistema, no posibilita evaluar el cumplimiento de los objetivos estratégicos previamente definidos, lo que dificulta el seguimiento y control de la Planeación Estratégica y Operativa tanto para la entidad como a nivel de gobierno teniendo en cuenta que no es posible:

¹ Instrucción No. 1 del Presidente de los Consejos de Estado y de Ministros para la planificación de los objetivos y actividades en los órganos, Organismos de la Administración Central del Estado (OACE), entidades nacionales y las administraciones locales del Poder Popular. Tiene como objetivo establecer el procedimiento para llevar a cabo el proceso de planificación del Gobierno, que permita dar cumplimiento a los acuerdos y resoluciones aprobadas en el VI Congreso del Partido Comunista de Cuba, las decisiones de la Asamblea Nacional del Poder Popular, el Consejo de Ministros y la actualización de los planes de la economía.

- Contar con indicadores que cualifiquen el cumplimiento de los objetivos trazados en el plan de la organización.
- Establecer un balance del cumplimiento de los indicadores y criterios de medidas.
- Identificar elementos claves para el cumplimiento de un objetivo.
- Evaluar el incumplimiento a partir de qué lo determinó, quiénes fueron los principales involucrados, la medida e impacto del incumplimiento.
- Estimar los posibles riesgos en la ejecución del proceso de Planeación Estratégica y Operativa en la organización.

Dada la problemática expuesta se identifica el siguiente **problema a resolver**:

¿Cómo evaluar el cumplimiento de los objetivos estratégicos en el Sistema de Planificación de Actividades, de manera que facilite el seguimiento y control de la Planeación Estratégica y Operativa?

Se define como **objeto de estudio**: Evaluación de objetivos en la Planeación Estratégica y Operativa.

De esta forma la investigación se enmarca en el **campo de acción**: Evaluación de objetivos en la Planeación Estratégica y Operativa en el Sistema para la Planificación de Actividades SIPAC.

Para darle solución al problema planteado se traza el siguiente **objetivo general**:

Desarrollar el paquete de funcionalidades: Gestor de reglas y criterios de medidas para la evaluación del cumplimiento de objetivos estratégicos en el Sistema de Planificación de Actividades SIPAC, de forma tal que facilite el seguimiento y control de la Planeación Estratégica y Operativa.

Dicho objetivo general es desglosado en **objetivos específicos** de la manera siguiente:

1. Estudiar el estado del arte referente a soluciones informáticas relacionadas con el campo de acción, así como analizar la Arquitectura del SIPAC para conocer las características fundamentales, marcos de trabajo, herramientas y tecnologías definidas para el desarrollo.
2. Modelar el negocio asociado al proceso de evaluación del cumplimiento de los objetivos, para comprender los elementos significativos del mismo.
3. Diseñar la solución que permita realizar la evaluación del cumplimiento de los objetivos estratégicos, en el sistema SIPAC 2.0

4. Implementar la solución que permita evaluar el cumplimiento de los objetivos estratégicos, en el sistema SIPAC 2.0
5. Validar la solución desarrollada mediante los métodos de pruebas: caja negra y caja blanca.

Tareas de la investigación:

1. Caracterización de los procesos relacionados con la evaluación del cumplimiento de los objetivos en la Planeación Estratégica y Operativa.
2. Análisis de la arquitectura del sistema SIPAC 2.0 para conocer las características fundamentales, marcos de trabajo, herramientas y tecnologías definidas para el desarrollo, así como la estrategia de integración entre los diferentes componentes.
3. Estudio del Modelo de Desarrollo del CEIGE.
4. Estudio de los patrones de diseño.
5. Elaboración del Modelo conceptual y el Glosario de términos asociados a la solución.
6. Identificación, especificación y validación de los requisitos funcionales del software.
7. Diseño de los prototipos de interfaz de usuario.
8. Estudio y actualización del Modelo de datos del SIPAC.
9. Elaboración del diseño de las clases asociado a la solución.
10. Validación del diseño aplicando las métricas adecuadas para esta etapa.
11. Implementación de los requisitos identificados.
12. Validación mediante la aplicación de pruebas de caja blanca y pruebas de caja negra a la solución desarrollada

Una vez expuestos los elementos anteriores se define como **idea a defender**: Si se desarrolla una solución para evaluar el cumplimiento de los objetivos estratégicos, se facilitará el seguimiento y control de la Planeación Estratégica y Operativa.

Métodos investigativos utilizados:

Para llevar a cabo la investigación se utilizaron los siguientes métodos teóricos:

Analítico-sintético: Permite descomponer el problema en sus partes y cualidades. Y a través de la síntesis hacer una unión entre esas partes, previamente analizadas descubriendo posibles relaciones y características generales entre los elementos de la realidad que lo rodea. (2) De

esta manera se analizaron independientemente elementos como: objetivos, indicadores y criterios de medida, identificando las características esenciales de los mismos. Mediante la síntesis se pudo identificar las relaciones existentes entre estos elementos y la dependencia que poseen entre sí.

Histórico-lógico: Permitió estudiar el comportamiento y los acontecimientos relacionados con el problema en el decursar de su historia.(2) A partir de este método fue posible hacer un estudio investigativo sobre la forma de evaluar el cumplimiento de los objetivos de las empresas estatales y organismos del país, fue posible analizar así, como ha ido evolucionado dicho proceso en las mismas.

Modelación: Posibilitó hacer una reproducción simplificada de la realidad lo que permitió descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio.(2) A través de este método se crearon modelos con vistas a investigar la realidad, representando en dichos modelos los principales conceptos y elementos a tener en cuenta para resolver el problema. Así como la modelación de esquemas y diagramas que permiten obtener una mejor representación del objeto de estudio, estos modelos deben responder a la norma de modelado UML.

También se hizo uso del método empírico:

Entrevista: Permitió obtener información necesaria y oportuna de personas con conocimientos suficientes sobre la planificación por objetivos que se lleva a cabo en los organismos y entidades presupuestadas del país.

Estructura del documento:

El presente documento se ha estructurado en 3 capítulos de la siguiente manera:

Capítulo 1: Fundamentación teórica: En este capítulo se aborda el estado del arte referente a la evaluación del cumplimiento de los objetivos estratégicos a nivel de país. Se describen algunas metodologías y sistemas que evalúan el cumplimiento de objetivos, además se identifican y caracterizan las herramientas y tecnologías utilizadas para el diseño e implementación de la solución.

Capítulo 2: Análisis y diseño de la solución: En este capítulo se describen las características esenciales que debe tener la solución propuesta, dando cumplimiento a las fases de

modelación y descripción de requisitos definidas en el modelo de desarrollo aplicado. Se obtiene el diseño del paquete de funcionalidades y la arquitectura que soportará el mismo.

Capítulo 3: Implementación y prueba: Este capítulo comprende todos los aspectos referentes a la implementación del Gestor de reglas y criterios de medida para evaluar el cumplimiento de objetivos estratégicos. Además se aplican una serie de pruebas a las funcionalidades implementadas con el fin de garantizar la calidad y validación de la solución.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se aborda el estudio realizado sobre los conceptos y procesos asociados al dominio del problema. Se estudian metodologías y sistemas que realizan la evaluación del cumplimiento de objetivos, y a partir de este estudio se identifican aspectos que pueden representar alternativas a utilizarse en el desarrollo del paquete de funcionalidades que da solución al problema planteado. Se realizará además un análisis de las herramientas, técnicas y tecnologías que serán usadas, así como una breve descripción del modelo de desarrollo a utilizar.

1.2 Marco Conceptual

1.2.1 Planificación

Según Jiménez: “La planificación es un proceso de toma de decisiones para alcanzar un futuro deseado, teniendo en cuenta la situación actual y los factores internos y externos que pueden influir en el logro de los objetivos”. (3)

Por otra parte los autores Rafael Valdés Toyos y Ana Cecilia Reyes definen la planificación como el proceso de establecer objetivos con el fin de alcanzar determinados resultados y elegir un futuro curso de acción para lograrlos. (4)

Ajustando estas concepciones a la realidad cubana, se entiende la planificación como el instrumento de dirección básico, que contempla los aspectos técnicos, productivos, económicos, sociales y financieros de forma integral, con vistas a potenciar la iniciativa y los esfuerzos de la empresa en el cumplimiento de sus funciones y objetivos generales, con el máximo de eficiencia y racionalidad, en la utilización de los recursos materiales y financieros, y en la activa participación de los trabajadores en los resultados.

1.2.2 Objetivo

Algunas de las definiciones asociadas al término objetivo expresan que constituye un elemento programático que identifica la finalidad hacia la cual deben dirigirse los recursos y esfuerzos para dar cumplimiento a la misión, tratándose de una organización, o a los propósitos institucionales, si se trata de las categorías programáticas. Expresión cualitativa de un propósito en un periodo determinado; el objetivo debe responder a la pregunta "qué" y "para qué". (5)

Los objetivos estratégicos son las metas y estrategias planteadas por una organización para reforzar, a largo plazo, la posición de la organización en un mercado específico, es decir, son los resultados que la empresa espera alcanzar en un tiempo mayor a un año, realizando acciones que le permitan cumplir con su misión, inspirados en la visión.(6)

Según, Alberto Reynaldo Ayuso, es necesario establecer indicadores que midan el cumplimiento de los objetivos de las organizaciones para verificar su efectividad y realizar las acciones correctivas en tiempo. Los objetivos deben ser medibles en el tiempo para poder determinar con precisión su cumplimiento. Es una de las condiciones básicas que deben cumplir para ser realmente efectivos. Siempre que se cuente con información oportuna y confiable, se pueden determinar indicadores que, apoyados en esa información, midan su grado de cumplimiento a través del tiempo de manera objetiva. (7)

1.2.3 Indicador

Un indicador siempre debe estar unido a la definición de objetivos a alcanzar. El indicador es una medida cuantitativa del desempeño, que sólo cobrará significado si se pone en consonancia con el objetivo que previamente se haya trazado. Es su comparación con este objetivo lo que permitirá saber si se está actuando de manera adecuada, si los procesos son efectivos y eficientes. (8)

Otras fuentes lo definen como una herramienta para clarificar y definir, de forma más precisa, objetivos e impacto, son medidas verificables de cambio y resultado, diseñadas para contar con un estándar contra el cual evaluar, estimar o demostrar el progreso con respecto a metas establecidas, facilitan el reparto de insumos, produciendo productos y alcanzando objetivos.(9)

1.2.4 Criterio de medida

Un criterio de medida establece la medida o unidad de medición que se utiliza para evaluar el éxito de un objetivo trazado y así medir su eficacia e impacto en la sociedad. Generalmente permiten medir la amplitud, profundidad y alcance de dicho impacto.

Se puede entender además como la asignación de un número a una propiedad física de algún objeto o conjunto de objetos con propósitos de comparación, y la medida da el número de unidades de la propiedad dada.

1.3 Estudio del Estado del Arte

1.3.1 Metodología para la evaluación de criterios de medida definidos para alcanzar los objetivos estratégicos del Ministerio de Salud Pública

Es una metodología de alcance nacional, para la evaluación de los objetivos estratégicos del Ministerio de Salud Pública (MINSAP). Esta metodología consiste en realizar el Plan de Desarrollo Estratégico del MINSAP siguiendo el esquema en el que se definen los objetivos estratégicos y se les asocia criterios de medida para evaluar su cumplimiento, estos deben ser definidos en el tiempo y ser cuantificables. (10)

Esta metodología propone realizar un análisis valorativo del comportamiento de los criterios de medida para el alcance del objetivo, el cual será evaluado en una escala de 1-5 que permita apreciar su cumplimiento para la etapa que se evalúa. Esta evaluación se ofrece basada en el siguiente sistema de calificación:

Tabla 1. Sistema de calificación del Plan de Desarrollo Estratégico del MINSAP

	5	4	3	2 y 1
Calificación	E	B	R	M

Aspectos Relevantes: Se definen objetivos estratégicos asociados a criterios de medida, estos últimos deben ser cuantificables y medibles en el tiempo.

1.3.2 Elementos del Sistema de Planificación establecidos en el Reglamento para la Implantación y Consolidación del Sistema de Dirección y Gestión Empresarial Estatal

El reglamento para la implantación y consolidación del sistema de dirección y gestión empresarial estatal constituye el instrumento de dirección para que las empresas estatales y organizaciones superiores de dirección, puedan, de forma ordenada, realizar las transformaciones necesarias con el objetivo de lograr la máxima eficacia y eficiencia en su gestión integral.(11)

Entre los elementos que lo componen se establece un sistema de indicadores con los que deben trabajar las empresas, los cuales se dividen en 3 grupos:

Indicadores directivos: Son los fundamentales y caracterizan el grado de cumplimiento de la tarea estatal y social asignada a la empresa, así como el nivel de eficiencia en la utilización de

los recursos y la fuerza de trabajo. Por lo tanto, el incumplimiento de al menos uno de ellos, implica que la empresa ha incumplido el plan anual y conlleva a la pérdida del reconocimiento social. Los indicadores directivos que deben emplear las empresas que aplican el sistema en cuestión son:

- Ventas de producciones y servicios seleccionados.
- Ventas netas.
- Utilidad del período.
- Aporte en divisas a la cuenta única.
- Relación gastos totales por peso de ingresos totales.

Indicadores límites: Estos representan determinados límites máximos en la utilización de recursos, para el cumplimiento del plan anual de la empresa. Son indicadores límites el consumo material, portadores energéticos, el presupuesto de las inversiones, asignaciones de productos que se balancean centralmente, la inversión de equipos automotores, reforzamiento alimentario, productos de aseo, ropa y calzado, gastos en divisas para la estimulación a los trabajadores, gastos de alimentos, entre otros.

Otros indicadores: Son indicadores físicos, económicos y financieros empleados por las empresas en la gestión diaria, tienen carácter disímil en dependencia de las diferentes actividades que conforman la empresa. Su empleo es determinado por la empresa, las organizaciones superiores de dirección y los organismos en dependencia de las necesidades informativas y de control.

Aspectos Relevantes: Se definen diferentes tipos de indicadores que, en consecuencia del área que abarcan, permiten medir el grado de eficiencia en el cumplimiento de los objetivos trazados por parte de las empresas cubanas.

1.3.3 Indicadores para el seguimiento de los objetivos de desarrollo del milenio

Se trata de un material donde quedan definidos un conjunto de indicadores para llevar a cabo el seguimiento de los objetivos de desarrollo del milenio. Estos indicadores son una medida cuantitativa que permite medir los progresos hacia los ocho objetivos y 18 metas resultantes de la Declaración del Milenio de las Naciones Unidas en el año 2000 donde participaron 189 países.

La lista de indicadores fue elaborada utilizando diversos criterios, no es rígida; trata más bien de tener en cuenta el contexto nacional y las opiniones de las diversas partes interesadas al preparar los informes de los países. En este documento quedan plasmados un conjunto de indicadores los cuales responden a un objetivo y meta determinada, además de comprender la justificación de su elección y el método de cálculo.

Aspectos Relevantes: A través de este documento se obtiene una representación de los posibles indicadores que necesitan cumplir las entidades y sectores de la sociedad, dígase: salud, educación, deporte, economía, cultura, todos ellos destinados a elevar el nivel de desarrollo de los países del mundo en cada una de estas áreas. (12)

1.3.4 Evaluación del cumplimiento de objetivos en los sistemas de incentivación del desempeño

Consiste en un sistema desarrollado por la Agencia Valenciana de Salud para la evaluación del cumplimiento de los objetivos de los Acuerdos de Gestión² y se basa en la valoración combinada de tres componentes: grado de aproximación a la meta, evolución respecto al año anterior y posición relativa respecto al resto.

Tiene en cuenta tres componentes fundamentales: la selección apropiada de los indicadores, la fijación equitativa de metas y la evaluación justa de los resultados. (13)

Aspectos Relevantes: Su principal limitación es que requiere un número mínimo de casos y no es aplicable a colectivos con pocos sujetos. En cambio para la mayoría de casos en que el mismo indicador es utilizado para grupos de 15 a 50 o más centros, servicios o profesionales, este enfoque tiene una gran riqueza y una demostrada equidad para resolver las situaciones más problemáticas.

1.3.5 Modelo de evaluación de la aplicación de la Instrucción No.1 del Presidente de los Consejos de Estado y de Ministros para la planificación en Cuba

El modelo de evaluación de la aplicación de la Instrucción No.1 del Presidente de los Consejos de Estado y de Ministros para la planificación en Cuba es la herramienta mediante la cual se

² Acuerdos de Gestión: constituyen una herramienta básica de la Agencia Valenciana de Salud para evaluar el resultado del desempeño de sus diferentes departamentos, de sus unidades funcionales e inclusive de los profesionales a título individual, esto se hace a través de la evaluación de un conjunto de indicadores definidos anualmente.

realiza la evaluación del cumplimiento de los objetivos trazados por el Grupo de Planificación de la Secretaría del Comité Ejecutivo del Consejo de Ministros.

Este modelo permite establecer indicadores y criterios de medida por cada objetivo identificado. Cada uno de estos elementos representa un porcentaje determinado en el cumplimiento de un objetivo, cuya suma total debe ser igual a 100. De esta forma se logra identificar los principales elementos que influyen en el incumplimiento del plan anual y el grado de relevancia que cada indicador o criterio de medida tiene en su definición junto a un objetivo.

El porcentaje de cumplimiento de los objetivos se pone en consonancia con una evaluación cualitativa del desempeño del objetivo. Esto permite conocer el desempeño que tiene el objetivo en el transcurso de la planificación estratégica, arrojando criterios que muestren el avance (avanza), mantenimiento (se mantiene) o retroceso (retrocede) del cumplimiento del objetivo evaluado, los cuales determinan las evaluaciones de Bien, Regula o Mal, por cada criterio respectivamente.

Aspectos relevantes: Se establecen indicadores y criterios de medida por cada objetivo trazado. Cada indicador o criterio de medida representa un porcentaje del cumplimiento del objetivo, cuyo comportamiento proporciona una evaluación cualitativa de su desempeño.

1.3.6 Soluciones informáticas para evaluar el cumplimiento de objetivos

SAGI

SAGI es un sistema implementado dentro de la Escuela Superior Politécnica del Litoral (ESPOL), ubicada en Guayaquil - Ecuador y utilizado por la Facultad de Ingeniería en Electricidad y Computación (FIEC) para seguir el cumplimiento de objetivos estratégicos por medio del cálculo automatizado de indicadores y reportes gráficos a través de una aplicación web de interfaz amigable para el usuario. Una de las funcionalidades que posee permite la evaluación de indicadores por objetivos, esta se realiza por medio de una fórmula matemática ingresada por el usuario, la cual es procesada por el sistema. Una vez evaluados estos indicadores, son generados reportes gráficos. (14)

Observaciones: SAGI permite que una vez identificados los indicadores sea posible seguir el cumplimiento de los objetivos estratégicos por medio de cálculos automatizados y reportes gráficos. El sistema no cuenta con la infraestructura que le permita ampliar el número de usuarios que usan la aplicación.

SASCO

El Sistema Automatizado para el Seguimiento y Control de los Objetivos en las Instituciones de Educación Superior (IES) del MES es uno de los soportes que apoyan el Control de Gestión (CG) de dichas instituciones. Tiene como objetivo esencial brindar información útil para la toma de decisiones relevantes en torno a la capacidad que presentan las IES en cuanto al cumplimiento de sus resultados planificados. Se apoya en el principio de que la CG no debe ser reducido a la función de control (entendido como evaluación y corrección del desempeño solamente) sino que comprende también la fase de planificación (porque durante el proceso de definir objetivos se determinan las formas de medirlos, y su cuantificación en el indicador).

Como parte del proceso de Evaluación del cumplimiento de los objetivos, el SASCO apoya el proceso del chequeo Parcial y Final de los resultados, para cada una de las Estrategias Maestras y Áreas de Resultados Claves, ya que posibilita evaluarlas. En el caso de las Áreas de Resultados Claves, los niveles de cumplimiento de los indicadores se apoyan en un sistema de alerta basado en una gama de colores y en el cálculo del porcentaje asociado a los mismos, lo cual representa una gran ayuda para los directivos en el análisis de los resultados y la toma de decisiones relacionadas con acciones de ajuste y de planificación para el próximo periodo. (15)

Observaciones: Es un sistema multi-entidades, lo que significa que en una única instalación del sistema se pueden explotar múltiples áreas o lo que es igual, procesar la información de diferentes instituciones, facultades o departamentos. Cada entidad es completamente independiente del resto y su información no se mezcla, pues cada una se crea en una carpeta propia donde estará ubicada su información.

1.3.7 Resumen de soluciones informáticas para evaluar el cumplimiento de objetivos

Tabla 2. Resumen de soluciones informáticas para evaluar el cumplimiento de objetivos

Sistemas	Elementos utilizados para la evaluación de objetivos	Interpretación de los niveles de cumplimiento de los objetivos	Adaptabilidad al proceso de Planeación Estratégica y Operativa
SAGI	Se establecen indicadores los cuales se evalúan por una	A partir de las evaluaciones de indicadores se generan reportes gráficos que, por	Carencia de funcionalidades que faciliten la realización de

	fórmula matemática la cual tiene una frecuencia de medición, así como su responsable, meta y valor.	medio de colores, ilustran el estado actual de los indicadores y cumplimiento de los objetivos.	las tareas y la ampliación del número de usuarios que usan la herramienta.
SASCO	Se chequea el cumplimiento parcial y final de los resultados, para cada una de las Estrategias Maestras y Áreas de Resultados Claves, las cuales tiene asociados indicadores.	Los niveles de cumplimiento de los indicadores se apoyan en un sistema de alerta basado en una gama de colores y en el cálculo del porcentaje asociado a los mismos.	A pesar de que es un sistema multientidades no permite consolidar la información de cada área según niveles generales de planificación Se basa fundamentalmente en los enfoques de la Dirección y Planificación Estratégica, así como la Dirección por Objetivos.

1.4 Valoración del Estado del Arte

Una vez realizado el estudio del estado del arte referente a metodologías para la evaluación del cumplimiento de objetivos se arriba a la conclusión de que, al trazarse objetivos y metas, las empresas y organismos establecen criterios de medida e indicadores que según su comportamiento, proporcionan el saber en qué medida se ha cumplido con dichos objetivos. Además se tiende a establecer diferentes criterios para valorar dicho cumplimiento como lo son: Si fue cumplido o no, en qué medida fue cumplido o evaluar cualitativamente dicho cumplimiento (generalmente se usa: E, B, R, M).

El análisis de sistemas foráneos que informatizan algunas actividades asociadas a la planificación por objetivos reafirmó la necesidad de contar con un sistema que íntegramente permita tanto la gestión como la evaluación de los objetivos precisados durante la planificación

estratégica y operativa. SIPAC debe permitir hacer esta evaluación a todos los niveles organizacionales, logrando la flexibilidad y efectividad que se requiere en este proceso clave para la toma de decisiones.

1.5 Lenguajes, tecnologías y herramientas propuestos para el desarrollo de la solución

En todo desarrollo de software, uno de los factores de mayor importancia son los lenguajes de modelado y de desarrollo que se utilizan para dar solución a un problema determinado. Dichos lenguajes permiten especificar la estructura o comportamiento que tendrá el sistema, además de propiciar una guía para la construcción del mismo. Los lenguajes, tecnologías y herramientas que a continuación se presentan han sido seleccionados y aprobados por el centro CEIGE para el desarrollo de la solución.

1.5.1 Lenguajes de modelado

Los lenguajes de modelados son aquellos que a través de símbolos estandarizados permiten diseñar organizadamente el desarrollo de un software. Como lo planteara José Enrique González Cornejo, Gerente General de DocIRS en su ensayo *¿Qué es UML?* (Enero 2008): “El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño”. El lenguaje de modelado que se propone para la solución es el siguiente(16):

UML v2.1

El Lenguaje de Modelado Unificado o UML (Unified Modeling Language) fue concebido para modelar los elementos de un sistema de software de una manera estandarizada que incluya conceptos del proceso de negocio y funciones del sistema. Este modelado será de fácil comprensión para cualquier desarrollador con conocimientos sobre UML, y podrá ser utilizado en cualquier tipo de desarrollo.(17)

BPMN

BPMN (Business Process Modeling Notation) es empleado en el desarrollo de la solución que se propone para presentar gráficamente las diferentes etapas del proceso de Evaluar el cumplimiento de los objetivo de un plan. Este estándar de modelado de procesos de negocio ha sido diseñado específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.

1.5.2 Lenguajes de Desarrollo del lado del cliente

HTML

El Lenguaje de Marcas de Hipertexto o HTML (Hyper Text Markup Language) es un lenguaje sencillo escrito mediante etiquetas (Tags) que permiten definir y ubicar los distintos elementos que componen una página web. Se integra con lenguajes como JavaScript y PHP y no todos los navegadores interpretan este lenguaje de la misma forma, lo cual puede ser visto como una de sus desventajas.(18)

CSS

Las Hojas de Estilo en Cascada o CSS (Cascading Style Sheets) constituye un estándar que permite establecer el estilo (dígase tamaños, iconos, imágenes, tipografías, colores, espacios y bordes) de documentos estructurados, dígase la capa de presentación de una página web. Este lenguaje permite definir el aspecto visual del documento, mientras que separa la parte semántica (HTML) de la presentacional (style sheets). (19)

JavaScript

JavaScript es un lenguaje que no requiere compilación ya que es interpretado por todos los navegadores y es utilizado para la creación de páginas web dinámicas. Dicho lenguaje permite crear diferentes efectos e interactuar con los usuarios. Además interactúa perfectamente con códigos basados en HTML, lo cual constituye una de sus principales ventajas.(20)

1.5.3 Lenguajes de Desarrollo del lado del servidor

PHP v5.3

PHP 5.3 es el lenguaje que se empleará para programar del lado del servidor. Es interpretado y completamente orientado al desarrollo de aplicaciones web dinámicas. Es gratuito, fácil de usar y aprender, portable, de código abierto y multiplataforma. Presenta interfaces para una gran cantidad de sistemas de base de datos diferentes, así como bibliotecas incorporadas para muchas tareas web habituales. Sin embargo, este lenguaje debido a su flexibilidad, puede convertir al sitio en un punto de fácil acceso a piratas informáticos.(21)

1.5.4 Tecnologías y herramientas de desarrollo

Marcos de Trabajo

Un marco de trabajo o framework es una estructura de soportes de programas, librerías y lenguajes de scripting. Es considerado una arquitectura de software que modela las relaciones generales de los componentes del proyecto que lo implementa; provee una estructura y manera de trabajo la cual utilizan las aplicaciones del proyecto. La finalidad de los frameworks es facilitar el desarrollo de software, permitiéndoles a diseñadores y programadores concentrarse en los requerimientos del proyecto, reduciendo los posibles problemas con las tecnologías utilizadas, así como facilitando ciertas funcionalidades básicas y comunes.(22)

Sauxe v1.0

Es un marco de trabajo, fusionado bajo tecnologías totalmente libres (entre ellas PHP, Postgresql, Apache) que posee el desarrollo de tecnologías propias basadas en otros marcos de trabajo como ZendFramework para el manejo de la lógica de negocio, Doctrine para el acceso a datos y ExtJS para la capa de presentación. Cuenta con una arquitectura en capas que a su vez presenta en su capa superior un MVC (patrón arquitectónico Modelo Vista Controlador). Contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo.

ZendFramework v1.5

ZendFramework2 o ZF2 es un marco de trabajo para el manejo de la lógica de negocio. Es una implementación que uso código 100% orientado a objetos. Es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. Brinda facilidades de uso y poderosas funcionalidades, posee buenas capacidades de ampliación y proporciona un sistema de caché de forma que se puedan almacenar diferentes datos, así como los componentes que forman la infraestructura del patrón Modelo-Vista-Controlador. Consta de mecanismos de filtrado y validación de entradas de datos. Permite convertir estructuras de datos PHP a JSON y viceversa, para su utilización en aplicaciones AJAX y provee capacidades de búsqueda sobre documentos y contenidos. (23)

Doctrine v1.0

Doctrine es empleado para la capa de acceso a datos. Es un sistema ORM (en inglés Object Relational Mapper) para PHP 5.2 o superior que incorpora una DBL (capa de abstracción a base de datos). Uno de sus rasgos importantes es la habilidad de escribir opcionalmente las preguntas de la base de datos orientada a objeto. Esto les proporciona una alternativa poderosa a diseñadores de SQL, manteniendo un máximo de flexibilidad sin requerir la duplicación del código innecesario. Además, exporta una base de datos existente a sus clases correspondientes y convierte clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.(24)

Ext v2.2

ExtJs es el marco de trabajo empleado para el desarrollo de la capa de presentación. Está basado completamente en la programación orientada a objeto. Cada objeto contiene lo típico: propiedades, métodos y eventos. Basa toda su funcionalidad en JavaScript a través de librerías. Así, en tiempo de ejecución carga y crea todos los objetos HTML a través del uso intenso de Document Object Model (DOM). Los datos son obtenidos con AJAX. Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos. Además permite que exista un balance entre el Cliente – Servidor, posibilitando que la carga de procesamiento se distribuya, permitiendo que el servidor al tener menor carga, pueda manejar más clientes al mismo tiempo.(25)

1.4.4.2 Técnicas de desarrollo

AJAX

Ajax por sus siglas en inglés Asynchronous JavaScript And XML (JavaScript asíncrono y XML) es la técnica de desarrollo web que se usa para poder hacer consultas asíncronas al servidor sin necesidad de recargar la página. Esta surge de la combinación de tres tecnologías ya existentes: HTML (o XHTML y Hojas de Estilo en Cascada (CSS) para presentar la información, DOM y JavaScript, para interactuar dinámicamente con los datos, además de XML y XSLT, para intercambiar y manipular datos de manera de sincronizada con un servidor web.(26)

CASE: Visual Paradigm v8.0

Se emplea Visual Paradigm for UML v8.0 como herramienta CASE. Utiliza UML v2.1 como lenguaje de modelado, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Ayudando a construir aplicaciones de calidad más rápido, mejor y a bajo costo.(27)

Sistema de control de versiones: Subversión

Subversión (SVN) 1.6 es la herramienta de entorno colaborativo que se utiliza para el control de versiones. Se encuentra preparado para funcionar en red y se distribuye bajo licencia libre. Mantiene versiones no sólo de archivos, sino también de directorios y versiones de los metadatos asociados a esos directorios. Además de los cambios en el contenido de los documentos, se mantiene la historia de todas las operaciones de cada elemento, incluyendo la copia, cambio de directorio o de nombre. Brinda facilidades de soporte tanto de ficheros de texto como binarios. (28)

Entorno integrado de desarrollo (IDE): NetBeans

La presente solución se desarrolla sobre el IDE de programación multiplataforma NetBeans 7.1 El cual tiene soporte para la versión 5.2 de PHP, JavaScript, el diseño de Hojas de Estilo (CSS) y HTML. Se integra con varias herramientas como el Subversión y servidores web. Es un producto de código abierto, con todos los beneficios del programa disponible en forma gratuita. Hace uso de plugins para ampliar sus funcionalidades, lo que le da una gran facilidad de uso.(29)

Servidor web: Apache

Se utiliza como servidor web Apache 2.0 pues es una tecnología gratuita de código abierto compatible con muchos Sistemas Operativos. Tiene todo el soporte que se necesita para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Tiene una alta configuración en la creación y gestión de registros de

actividad. Apache permite la creación de ficheros de registro a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor. (30)

Sistema gestor de bases de datos: PostgreSQL

Como Sistema Gestor de Base de Datos (SGBD) se emplea la versión 9.1 de PostgreSQL. Constituye un sistema de gestión de bases de datos relacional. Es una herramienta de código abierto, de bajo coste y multiplataforma. Se destaca en ejecutar consultas complejas, subconsultas y uniones de gran tamaño. Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Soporta transacciones, claves ajenas con comprobaciones de integridad referencial y almacenamiento de objetos de gran tamaño. Cuenta con varias herramientas gráficas de diseño y administración de bases de datos como el pgAdmin. (31)

1.5.5 Modelo de desarrollo

Se emplea el Modelo de desarrollo definido por el centro CEIGE el cual permite identificar las actividades de cada una de las fases por las que se debe transitar el ciclo de vida del proyecto y el conjunto de artefactos a generar en cada una de ellas. Este modelo define varias disciplinas por cada una de las fases por las que transita.

Dichas fases se describen a continuación:

1. Inicio o Estudio preliminar

En esta fase se realiza un estudio de factibilidad del proyecto que permite determinar si este tiene alcance, además de hacer un análisis para planificar y asignar los recursos necesarios para realizar el proyecto.

2. Desarrollo: Esta fase es la encargada de ejecutar las actividades requeridas para desarrollar el software, con el objetivo de obtener un sistema que satisfaga las necesidades por las que surgió el proyecto.

Disciplinas que componen los proyectos de desarrollo de software:

En la fase de desarrollo se definen los requisitos, se lleva a cabo el análisis y el diseño, se implementa y se libera la solución informática. Estos aspectos constituyen las disciplinas que se deben ejecutar en los proyectos de desarrollo de software. (32) A continuación se identifican las disciplinas que serán utilizadas para obtención de la solución esperada:

1. Modelado del Negocio: En esta fase se lleva a cabo un estudio que permite identificar y analizar los procesos que tendrán que ser llevados a cabo para la realización de la solución que se desea implementar.
2. Descripción de Requisitos: Esta disciplina permite obtener un modelo del software que se quiere construir. Genera un grupo de artefactos que permite determinar, administrar y desarrollar los requisitos de la solución.
3. Análisis y diseño: Es necesario contar con esta fase, pues en la misma se llevará a cabo el modelado del sistema, convirtiéndolo en un plano para la siguiente fase.
4. Implementación: Esta disciplina constituye la parte resultante del análisis y el diseño pues en ella se llevará a cabo la implementación del sistema a partir de la generación de componentes que se unirán para lograr la solución deseada.
5. Pruebas Internas: Se hace necesario realizarle pruebas al sistema resultante de la implementación para corregir posibles errores cometidos tanto en la documentación como en el software, por ello la importancia de esta fase.
6. Pruebas de Liberación: Para obtener la aceptación final del cliente se deben realizar pruebas al entregable del proyecto por una entidad externa capacitada para realizar dicha labor.

La realización de este trabajo se enfoca en darle cumplimiento a la fase de *Desarrollo* del modelo presentado, comprendiendo las disciplinas de Modelado, Descripción de requisitos, Análisis y diseño, Implementación y Pruebas internas.

1.6 Conclusiones del capítulo

Con el desarrollo del presente capítulo se pudo arribar a las siguientes conclusiones:

1. Con la conceptualización de elementos como: planificación, objetivos, indicadores y criterios de medida se pudo obtener un mayor conocimiento sobre la forma de analizar cuantitativa y cualitativamente el cumplimiento de los objetivos en la Planeación Estratégica y Operativa de las empresas cubanas.
2. El análisis de diferentes metodologías existentes para evaluar el cumplimiento de objetivos en diferentes organismos y entidades permitió la apropiación de los aspectos

de mayor significación para solucionar el problema que da pie a la investigación. De la misma manera, el estudio de diferentes sistemas informáticos que miden el cumplimiento de objetivos, permitió identificar elementos afines con la solución que se desea desarrollar.

3. La identificación de los lenguajes, tecnologías y herramientas, así como el estudio realizado sobre el modelo de desarrollo a utilizar en el desarrollo de la investigación, aportó un mayor conocimiento sobre los mismos, lo que facilitará su posterior utilización.

CAPÍTULO 2: ANÁLISIS Y DISEÑO

2.1 Introducción

En el presente capítulo se describen las características de la solución propuesta. De esta forma se detalla una propuesta de solución para la evaluación del cumplimiento de los objetivos, se definen los principales conceptos asociados al negocio, se describen y especifican los requisitos funcionales, así como los no funcionales. Se realiza una descripción de la solución en términos de componentes y se definen los patrones de diseño utilizados.

2.2 Propuesta de solución

Una vez analizadas y valoradas diferentes soluciones que llevan a cabo la evaluación del cumplimiento de objetivos se propone una alternativa que permite realizar la evaluación del cumplimiento de los objetivos trazados por las entidades y organizaciones cubanas teniendo en cuenta indicadores, criterios de medida y reglas que permitan evaluar dicho cumplimiento. Por consiguiente, a los objetivos, se podrán asociar indicadores y/o criterios de medidas, según el aporte de estos al cumplimiento de dicho objetivos.

Si se desea evaluar el objetivo a través de indicadores, a estos se le podrán asociar actividades, en este caso se obtiene una evaluación cualitativa del objetivo según el cumplimiento o no de la actividad. En caso de evaluar el indicador a través de reglas se obtendrá una evaluación cuantitativa del cumplimiento del indicador. Las reglas permitirán establecer combinaciones de criterios de medida a cumplirse para que se cumpla un objetivo. Una vez configurados los parámetros para evaluar los objetivos anteriormente mencionados se puede obtener la evaluación del cumplimiento de un objetivo. Para ello es necesario configurar un rango de evaluación que determine, según el comportamiento de los parámetros a evaluar, si el objetivo está *No cumplido*, *Parcialmente cumplido* o *Cumplido*. Para mayor entendimiento se muestra la siguiente ilustración:

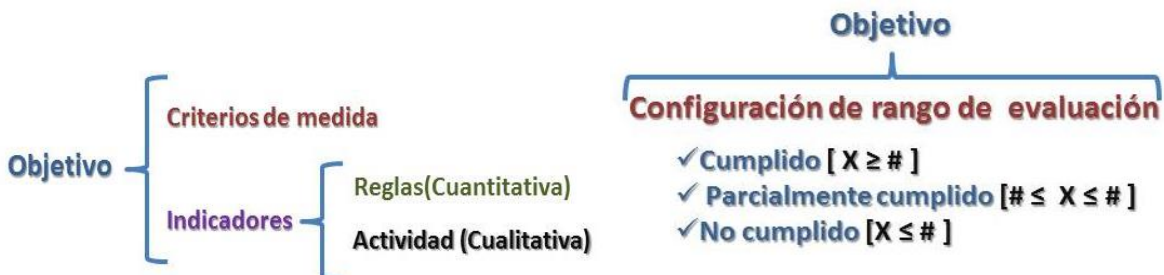


Figura 1. Propuesta de solución para evaluar el cumplimiento de los objetivos en el SIPAC

Todos estos elementos podrán ser introducidos en el sistema en dependencia del nivel de subordinación de los usuarios, permitiendo que los directivos puedan evaluar el desempeño de su empresa o entidad lo que facilitará el seguimiento y control de la Planeación Estratégica y Operativa que se lleva a cabo actualmente en el país.

2.3 Modelado del negocio

Los artefactos correspondientes a esta disciplina se encuentran en el expediente del proyecto SIPAC en la dirección SIPAC 2.0\Expediente 3.4\Ingeniería\Modelo de Negocio.

2.3.1 Diagrama de procesos de negocio

En el presente acápite se modela el proceso de negocio del paquete de funcionalidades Gestor de reglas y criterios de medida para evaluar el cumplimiento de objetivos estratégicos. Para la representación del mismo se hizo uso de la notación BPMN, la cual permitió representar las etapas del proceso de la solución propuesta. Se relacionan así roles, eventos y artefactos que intervienen en su desarrollo.

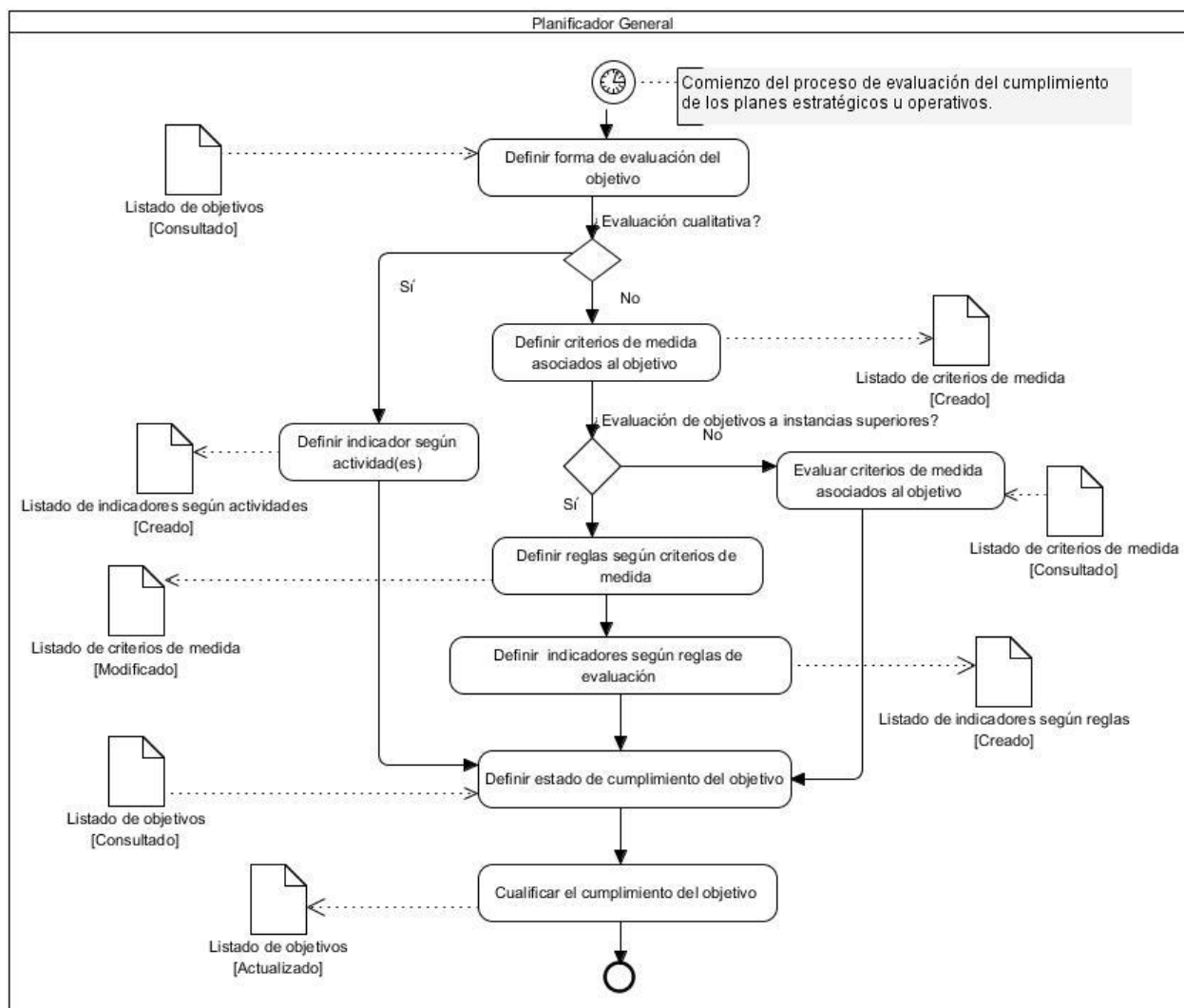


Figura 2. Proceso de negocio: Evaluar el cumplimiento de los objetivos de un plan.

2.3.2 Especificación del proceso de negocio: Evaluar el cumplimiento de los objetivos de un plan

Tabla 3. Especificación del proceso de negocio: Evaluar el cumplimiento de los objetivos de un plan.

Objetivo	Evaluar el cumplimiento de los objetivos de un plan.
Evento(s) que lo genera(n)	Comienza el proceso de evaluación del cumplimiento de los planes estratégicos u operativos.
Precondiciones	Existe un listado de objetivos aprobado.

Marco legal	N/A
Clientes internos	Evaluación de un plan Cumplimiento de los objetivos de un plan
Clientes externos	N/A
Entradas	Listado de objetivos (Documento Word)
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. Se define la forma en que desea evaluar el cumplimiento de los objetivos, ya que puede ser de forma cuantitativa si se quiere obtener un porcentaje del cumplimiento de los objetivos, o cualitativa si se quiere obtener una cualificación (No cumplido, Cumplido Parcialmente o Cumplido) del cumplimiento de los objetivo. 2. Si se desea obtener una evaluación cuantitativa, se establecen y asocian los criterios de medida que tributan al cumplimiento del objetivo. 2.a Se desea obtener una evaluación cualitativa 3. Si se va a evaluar un objetivo de una instancia superior, se definen reglas basadas en distintas combinaciones de los criterios de medida definidos. Si se cumple alguna de las reglas se genera una evaluación. 3.a Se evalúa el cumplimiento de un objetivo de una instancia inferior. 4. Se definen indicadores según las reglas de evaluación establecidas y se asocian al objetivo. 5. Se define el estado de cumplimiento del objetivo según el cumplimiento de los indicadores definidos. 6. Se cualifica el cumplimiento del objetivo. 	
Pos-condiciones	N/A
Salidas	
1.	Listado de objetivos actualizado (Documento Word)
Flujos paralelos	
1. N/A	
Pos-condiciones	N/A
Salidas	
1. N/A	
Flujos alternos	
Flujo alternativo 2.a Se desea obtener una evaluación cualitativa.	
<ol style="list-style-type: none"> 1. Se definen indicadores según actividad(es), las cuales deben cumplirse para que se cumpla el indicador. 2. Continúa con el paso 5 del flujo básico. 	
Pos-condiciones	N/A

Flujo alternativo 3.a Se evalúa el cumplimiento de un objetivo de una instancia inferior.

1. Se evalúan los criterios de medida asociados al objetivo que se quiere evaluar
2. Continúa con el paso 5 del flujo básico.

Pos-condiciones N/A

Asuntos pendientes N/A

2.3.3 Validación del proceso de negocio

El proceso de negocio Evaluar el cumplimiento de los objetivos de un plan en el SIPAC se encuentra validado a través de los criterios definidos en el documento CIG-SPA-N-Validación de procesos de negocio-SPA del expediente de proyecto de SIPAC 2.0 del centro CEIGE. Estos criterios son:

- El proceso debe tener el proveedor incluido en el listado de proveedores válidos.
- El proceso debe tener un identificador único.
- El proceso está completo si, y solo si, se cumplen los siguientes elementos: si se han descrito todos los flujos básicos, alternativos y paralelos del proceso; si se han completado todas las secciones de la planificación; si todas las figuras, tablas y diagramas están etiquetados; si todos los acrónimos, abreviaturas, siglas, términos y unidades de medidas se han definido.
- El proceso debe ser consistente (seguro) si: se ha identificado su proceso padre, si procede; si se han identificado los subprocesos, si procede; si se han representado todas su relaciones con otros procesos.
- El resultado del proceso debe ser evaluado de positivo.

2.3.4 Modelo conceptual

El modelo conceptual es una representación global de los principales elementos lógicos de la realidad que se analiza. Comprende y describe las clases más importantes dentro del contexto del sistema.

Elemento de la planificación: determina cuáles serán los componentes que se deberán tomar en consideración dentro de la Planificación estratégica y operativa. Se refiere a elementos que son considerados claves en la planificación dígame planes, objetivos y actividades.

Objetivos: son las metas y estrategias trazadas por una organización para ser alcanzadas a corto o largo plazo.

Indicador: es la forma de medir los objetivos para evaluar en cualquier momento el nivel de cumplimiento de los mismos.

Criterio de Medida: establece una medida que permite cuantificar el cumplimiento de los objetivos trazados por la entidad.

Regla: Constituye la herramienta para evaluar el cumplimiento de los objetivos según la combinación de criterios de medidas e indicadores.

Actividad: constituye una acción acometida por una entidad cuyo cumplimiento tributa a la evaluación del cumplimiento de los objetivos definidos en su plan estratégico u operativo.

La siguiente **figura** representa el modelo conceptual asociado al proceso de evaluación del cumplimiento de los objetivos estratégicos:

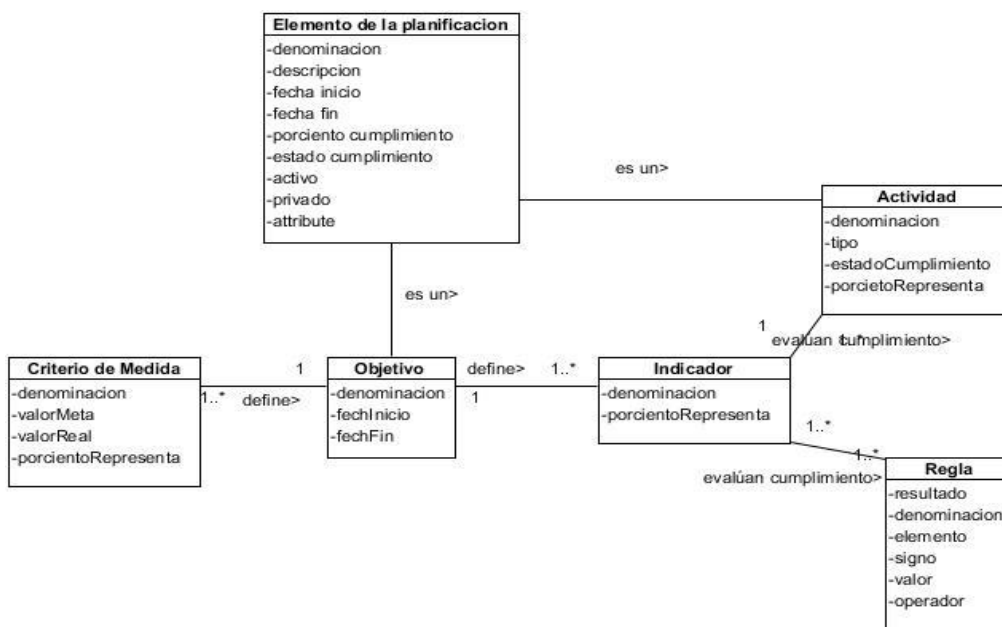


Figura 3. Modelo conceptual del Gestor de reglas y criterios de medida para la evaluación del cumplimiento de objetivos estratégicos.

2.4 Requisitos

Un requisito de software es, según la IEEE una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Se puede ver también como una condición o

capacidad que tiene que ser alcanzada o poseída por un sistema para satisfacer formalmente un contrato definido entre las partes interesadas. (33)

En este acápite se abordarán los elementos referentes a la descripción de los requerimientos que debe cumplir la solución implementada para lograr la calidad de esta y la satisfacción del cliente.

2.4.1 Técnicas de captura de requisitos

La calidad con que se realice la captura de los requisitos va a influenciar en todo el proceso de desarrollo del software repercutiendo en el resto de las fases de desarrollo del mismo. Las técnicas usadas en la recopilación de los requisitos que debe cumplir la solución propuesta fueron:

- **Entrevistas:** Permitted entender mejor el problema y comprender los objetivos de la solución buscada, para ello fueron entrevistados especialistas de los Organismos de la Administración Central del Estado (OACE) en términos de Planificación Estratégica y Operativa, además de profesionales del proyecto SIPAC, con cuantiosos conocimientos sobre el objeto de estudio.
- **Talleres:** Se pudo intercambiar opiniones e ideas entre los miembros del equipo de desarrollo del SIPAC, logrando abarcar todas las posibles necesidades del cliente, identificando así los requerimientos necesarios para obtener una solución que cumpla con las expectativas del mismo.

2.4.2 Listado de requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares.(34)

De esta forma quedan plasmados a continuación los requisitos funcionales identificados para el desarrollo del Gestor de reglas y criterios de medida para la evaluación del cumplimiento de objetivos estratégicos en el SIPAC:

RF1 Gestionar Indicador:

1.1 Insertar indicador.

RF3 Gestionar regla:

3.1 Adicionar regla.

1.2 Modificar indicador.

1.3 Eliminar indicador.

1.4 Listar indicadores.

RF2 Gestionar criterio de medida:

2.1 Adicionar criterio de medida.

2.2 Modificar criterio de medida.

2.3 Eliminar criterio de medida.

2.4 Listar criterios de medida.

RF4 Buscar según criterios de búsqueda.

3.2 Modificar regla.

3.3 Eliminar regla.

3.4 Listar regla.

RF 5 Establecer asociación de indicadores de cumplimiento a los objetivos definidos.

RF6 Establecer asociación de las actividades que inciden en el cumplimiento a los objetivos definidos.

RF7 Obtener evaluación de un objetivo.

2.4.3 Especificación de requisitos funcionales

Mediante la especificación de requisitos se obtiene una descripción completa del comportamiento de las funcionalidades que se van a desarrollar. Para su redacción se utiliza un lenguaje sencillo, de forma que sea fácilmente comprensible para todas las partes involucradas en el desarrollo.

A continuación se muestra la especificación del requisito *Adicionar regla*, la especificación de los restantes requisitos funcionales se encuentran descritos en el expediente de proyecto SIPAC 2.0 en el artefacto CIG-SPA-N-i2605.

Tabla 4. Especificación del requisito: Adicionar regla

Precondiciones	Debe haber creado en el sistema al menos un objetivo. Deben existir indicadores y/o criterios de medida insertados en el sistema.
Flujo de eventos	
Flujo básico Adicionar Regla	
1.	El usuario accede a la pantalla principal de Evaluar Objetivos y selecciona el objetivo al que desea adicionar la regla.
2.	El sistema muestra una interfaz que permite al usuario introducir los siguientes datos de la regla que desea asociar al objetivo: <ul style="list-style-type: none"> • Denominación • Resultado

	<ul style="list-style-type: none"> • Elemento • Signo • Valor • Operador
3.	El usuario introduce los datos de la regla y hace clic en el botón Aceptar.
4.	El sistema valida que los datos introducidos estén correctos y los registra.
5.	Concluye el requisito.
Pos-condiciones	
1.	Se registra en el sistema una nueva regla.
Flujos alternativos	
Flujo alternativo 3a. El usuario introduce valores incorrectos	
1.	El sistema no permite insertar datos erróneos y muestra un mensaje al usuario señalando los datos válidos para ese campo.
2.	El usuario introduce los datos correctamente y presiona la opción Aceptar.
3.	Volver al paso 4 del flujo básico.
Pos-condiciones	
1.	Se registra en el sistema una nueva regla.
Flujo alternativo 3b. El usuario introduce información incompleta	
	El sistema señala los datos vacíos y permite corregirlos.
	El usuario corrige los datos y presiona la opción Aceptar.
	Volver al paso 4 del flujo básico.
Pos-condiciones	
1.	Se registra en el sistema una nueva regla.
Flujo alternativo 3c. El usuario cancela la acción.	
1.	Concluye el requisito.
Pos-condiciones	
1	No se registran los datos
Validaciones	
1.	Se validan los datos según lo establecido en el Modelo conceptual CIG-SPA-N-i1302
Conceptos	N/A
Requisitos	N/A

especiales	
Asuntos pendientes	N/A

2.4.4 Validación de los requisitos funcionales

La validación de requisitos permite ratificar los requerimientos, es decir, verificar que todos los requerimientos que aparecen en el documento especificado son los que realmente satisface la petición del cliente. De esta forma puede asegurarse que los requisitos validados representan una descripción, por lo menos, aceptable, del sistema que se debe implementar. Esto implica verificar que los requerimientos sean consistentes y que estén completos. (35)

Para la validación de los requisitos funcionales identificados se utilizó la técnica: **Construcción de prototipos**. Se aplicaron además: Criterios para validar los requisitos del cliente y se realizó el Acta de aceptación de los requisitos, según lo establece el modelo de desarrollo de CEIGE.

2.4.5 Requisitos no funcionales

Los requisitos no funcionales son los requisitos que imponen restricciones al diseño o funcionamiento del sistema. Generalmente son aplicados al sistema en su totalidad, es por ello que los requisitos no funcionales concernientes a la solución se encuentran descritos en el expediente del proyecto SIPAC 2.0 en el artefacto CIG-SPA-N-i3507-RNF.

Los requisitos no funcionales que impactan directamente la solución: Gestor de reglas y criterios de medidas para la evaluación del cumplimiento de objetivos estratégicos en el SIPAC, son:

Funcionalidad:

- ✓ Precisión

Breve descripción: Capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.

1. Los datos identificativos del lugar, operación o dato con error, mostrados en los mensajes de error del sistema, deberán ser 100% correctos.
2. Los datos asociados a la gestión de tiempo en el sistema deben tener como mínimo una precisión de dd/mm/aaaa para las fechas y h: m para los horarios.
3. Los datos asociados al porcentaje de cumplimiento de los objetivos deberán redondearse por defecto, admitiendo solo dos cifras significativas después de la coma.

4. Las notificaciones generadas por el sistema al usuario activo deberán especificar siempre la fecha de emisión de la misma así como al elemento de planificación al que se encuentra relacionada y el autor de la operación que generó dicha notificación.

Eficiencia

- ✓ Rendimiento

Breve descripción: Capacidad del producto de software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados, al realizar su función bajo condiciones establecidas.

1. El sistema no excede los 5 s de respuesta al efectuar acciones de cargar un registro con gran cantidad de datos dispersos y los 2 s para 20 registros con datos poco dispersos (esta cifra no incluye los retardos por concepto de tráfico de red).
2. El sistema no excede los 1.5 s para efectuar acciones de salvar información (esta cifra no incluye los retardos por concepto de tráfico de red).

Usabilidad

- ✓ Operabilidad

Breve descripción: Capacidad del producto de software para permitirle al usuario operarlo y controlarlo.

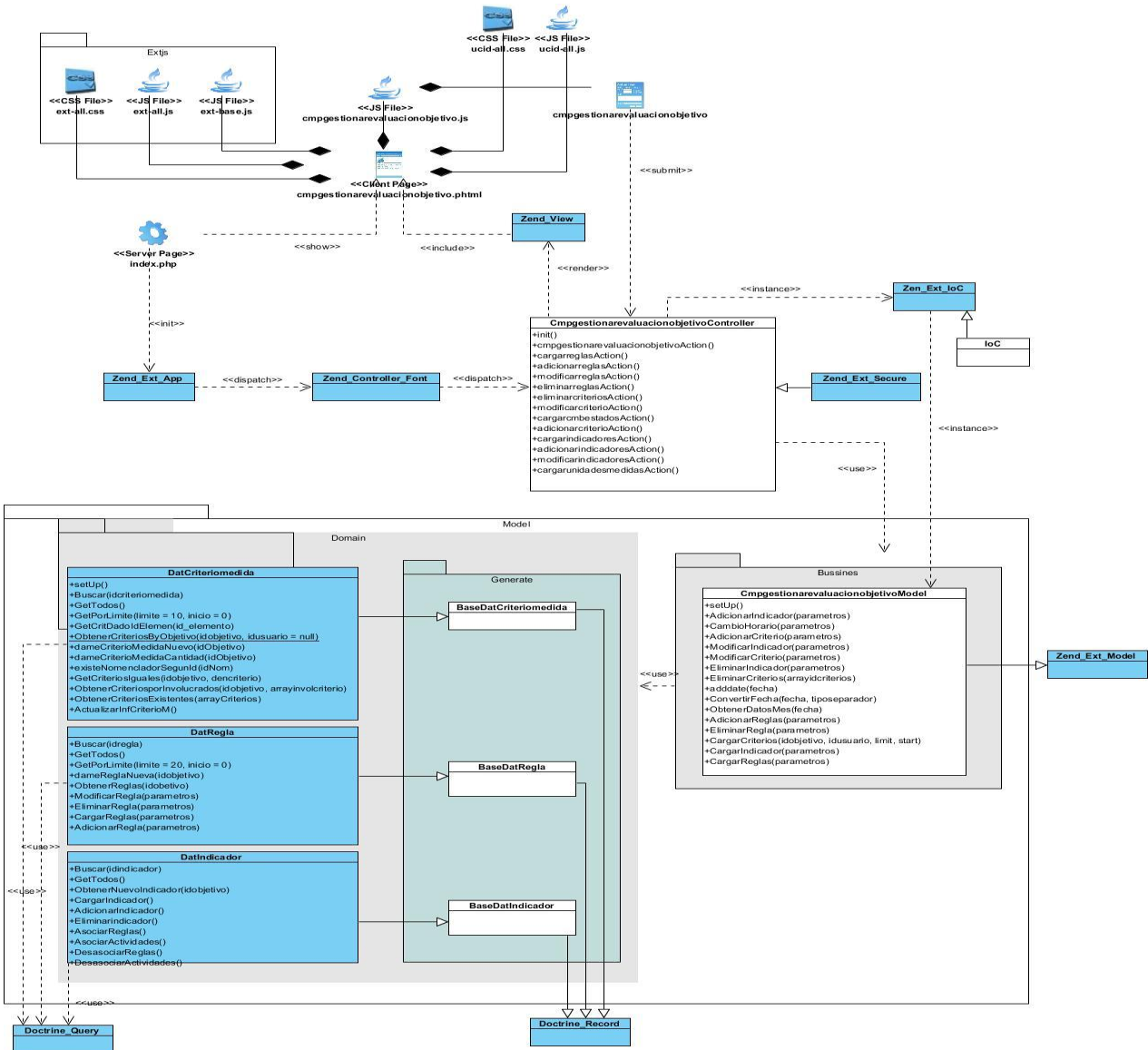
1. El sistema contará con un menú que permitirá acceder a todas las funcionalidades para introducir los datos y procesar la información según flujo de operaciones del negocio.
2. El sistema expondrá el menú general en todo momento para que pueda ser utilizado por el usuario en cualquier momento.
3. El sistema permitirá en cada escenario en el que se liste una determinada información realizar búsquedas sobre criterios específicos.

2.5 Modelado de la solución

2.5.1 Diagrama de clases del diseño web

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. En la figura que se muestra a continuación se presenta el diagrama de clases del diseño para la solución propuesta, el cual está basado en

estereotipos web y en el patrón arquitectónico MVC³. Para obtener una imagen ampliada de las diferentes partes del patrón MVC que se utiliza en el diagrama de clases del diseño web, ver Anexos 1, 2 y 3.



³ (MVC). Modelo Vista Controlador: es un estilo de arquitectura que separa los datos de una aplicación, la interfaz de usuario y la lógica de control, en tres componentes distintos.

Figura 4. Modelo de clases del diseño del gestor de reglas y criterios de medida para evaluar el cumplimiento de objetivos estratégicos.

Tabla 5. Descripción de las principales clases del diagrama de clases del diseño

Clase	Descripción
cmpgestionarevaluacionobjetivo.phtml	Página encargada de visualizar, a través de los js que debe incluir, la información necesaria para generar la evaluación de los objetivos.
cmpgestionarevaluacionobjetivo.js	Encargado de manejar los datos persistentes dentro del componente. Contiene el Bussines y el Domain.
cmpgestionarevaluacionobjetivoController.php	Clase encargada de controlar la comunicación entre la Vista y el Modelo. Es la responsable de realizar las diferentes funcionalidades sobre los objetivos, según las peticiones del usuario.
ZendExt_Controller_Secure	Encargada de gestionar acciones personalizadas y está integrada a la seguridad.
ZendExt_Model	Modelo gestor de negocio que permite entre otras funcionalidades iniciar la conexión a la base de datos.
Extjs	Contiene los componentes generados a través de la librería JavaScript Extjs.
Paquete Model	Encargado de manejar los datos persistentes dentro del componente. Contiene el Bussines y el Domain.

2.5.2 Diagrama de secuencia

Los diagramas de secuencia muestran la interacción de un conjunto de objetos en una aplicación a través del tiempo. Se modelan para cada requisito funcional y contienen detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el mismo, además de los mensajes intercambiados entre los objetos. A continuación se presenta el diagrama de secuencia correspondiente al requisito funcional Adicionar Indicador, como parámetro de evaluación de un objetivo:

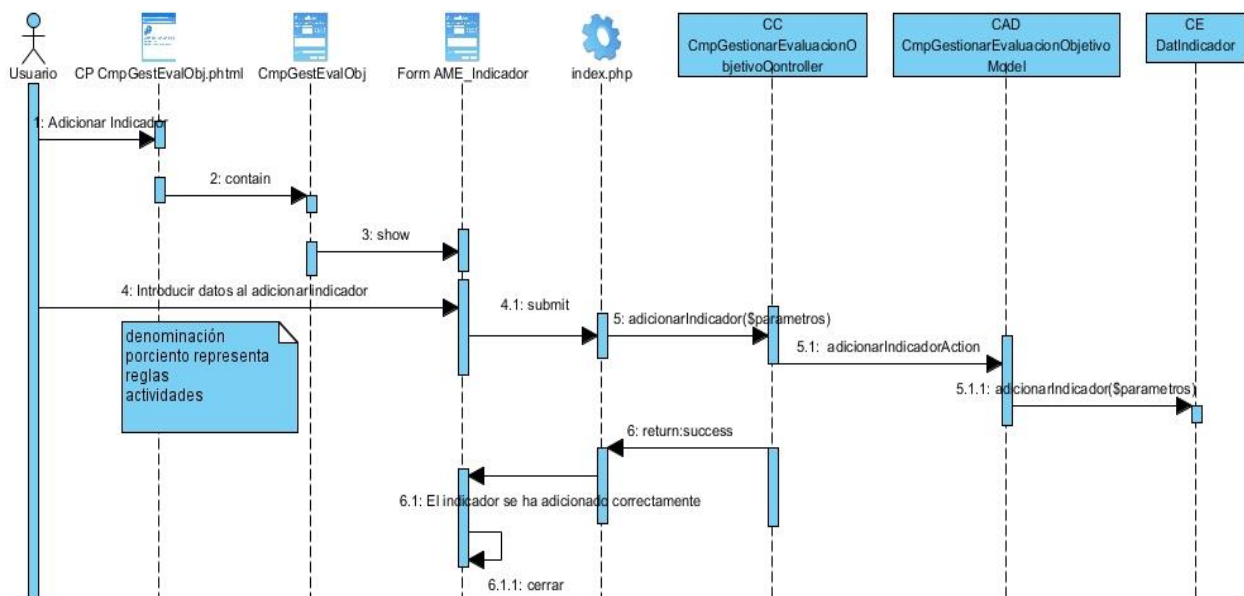


Figura 5. Diagrama de secuencia Adicionar Indicador

Inicialmente el usuario accede a la interfaz principal donde se muestran indicadores, reglas y criterios de medida que se pueden asociar a los objetivos para configurar la evaluación de los mismos. A continuación accede a la pestaña que contiene la configuración de los indicadores y selecciona la opción Adicionar indicador. Paso seguido se muestra un formulario donde el usuario debe introducir los datos del nuevo indicador que desea insertar y presiona el botón Aceptar para enviar los datos. La clase controladora *CmpgestionarevaluacionobjetivoController* envía la petición a la clase modelo *Cmpgestionarevaluacionobjetivo* la cual realiza la consulta a la clase de acceso a datos *DatIndicador*. Esta a su vez actualiza la tabla de Indicadores donde

se insertaran los datos del indicador adicionado. Finalmente se muestra un mensaje mostrando que el indicador fue adicionado correctamente.

2.5.3 Modelo de datos

Un modelo de datos es una colección de conceptos que se emplean para describir la estructura de una base de datos. Está compuesto por entidades, atributos y relaciones. La mayoría de los modelos de datos poseen un conjunto de operaciones básicas para especificar consultas y actualizaciones de la base de datos.(36)

El estudio del modelo de datos de SIPAC, evidencia la necesidad de crear nuevas tablas para almacenar los datos referentes a los indicadores, reglas y criterios de medida, así como a las evaluaciones que debe generar el Gestor de reglas y criterios de medidas para evaluar el cumplimiento de los objetivos estratégicos. Para ello se incorporan cuatro nuevas entidades las cuales se distinguen del resto por el color amarillo: `dat_criteriomedida`, `dat_reglas`, `dat_indicadores` y `dat_configuracion_evaluacion`. Para el funcionamiento del nuevo paquete de funcionalidades se utilizan dos entidades existentes en el modelo de datos del sistema SIPAC: `dat_actividades`, y `dat_objetivo`, para el manejo de los datos referentes a las actividades y los objetivos como elementos de la planificación.

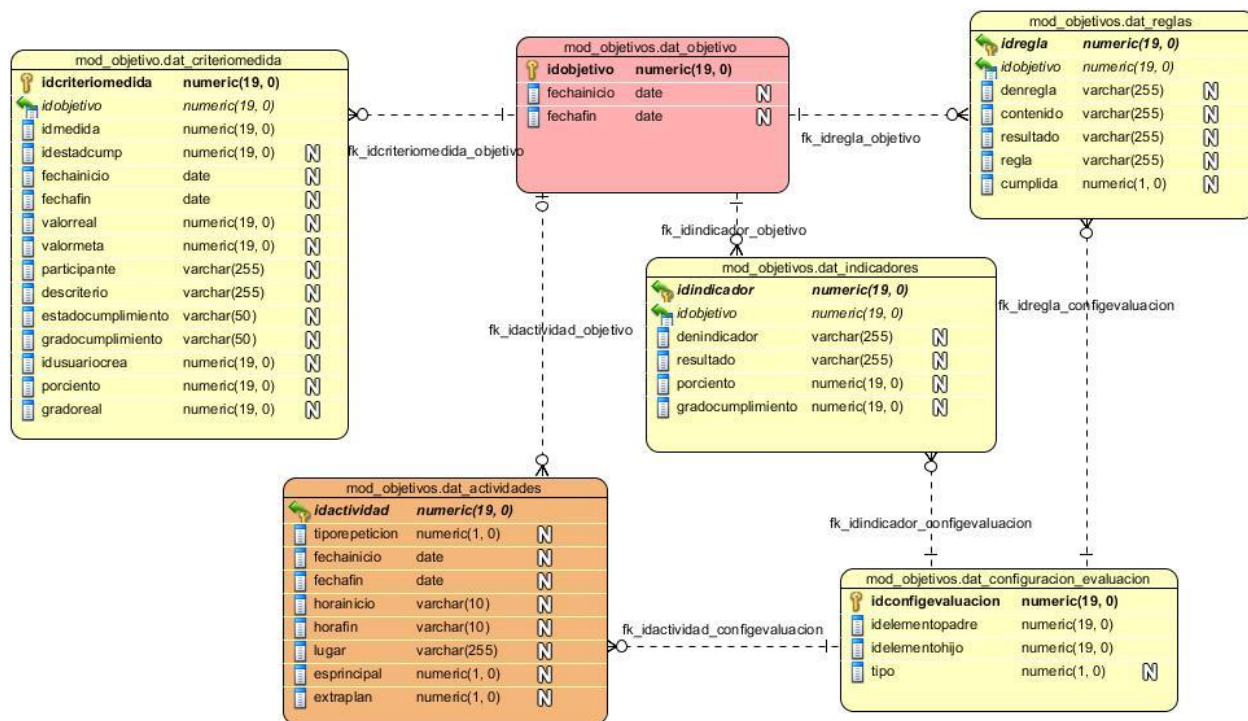


Figura 6. Modelo de datos actualizado para el Gestor de reglas y criterios de medida para evaluar el cumplimiento de objetivos estratégicos en el SIPAC.

2.5.4 Diagrama de componentes

El sistema SIPAC se encarga de interrelacionar objetivos de trabajo y actividades en tiempo real de tal forma que se garantice el seguimiento del desarrollo y control del cumplimiento de los objetivos y tareas principales en las entidades. La arquitectura base de SIPAC presenta una estructura que responde a diferentes niveles de empaquetamiento: subsistema, componente, funcionalidad y requisito.(37)

De esta forma el subsistema Planificación está constituido por los siguientes componentes:

Configuración: Permite configurar y gestionar los usuarios y grupos de usuarios, permisos y nomencladores del sistema.

Planeación: Permite gestionar los planes, áreas de resultados clave, objetivos y criterios de medida, así como actividades y factores que influyen en la planificación.

Notificaciones: Permite gestionar las acciones que generarán una notificación así como notificar al usuario activo sobre la asignación o alteración de algún elemento de planificación.

En la siguiente figura muestra las relaciones que se establecen entre dichos componentes:

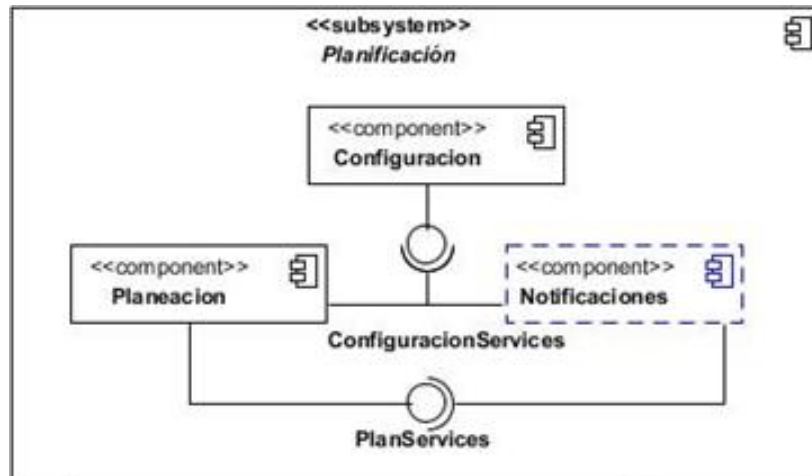


Figura 7. Representación del subsistema Planificación con sus componentes.

En el componente Planeación, la funcionalidad Gestor de reglas y criterios de medida para la evaluación del cumplimiento de objetivos estratégicos se encuentra relacionada con las funcionalidades Gestionar objetivos y Gestionar actividades. A dichas funcionalidades se le realizaron modificaciones asociadas a la solución.

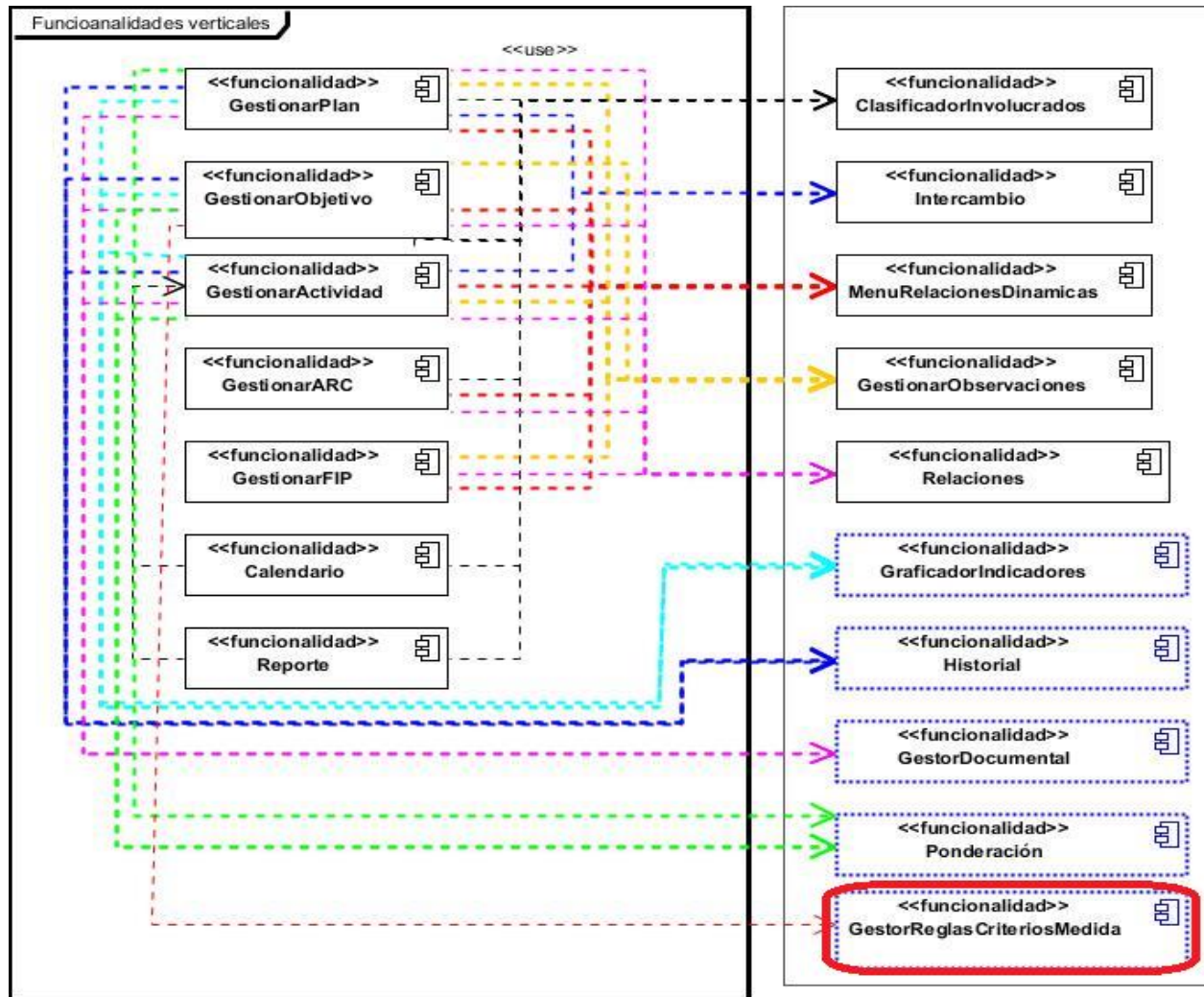


Figura 8. Diagrama del componente Planeación en el sistema SIPAC.

2.6 Patrones del diseño

Los patrones de diseño son soluciones bien pensadas a problemas conocidos de programación lo cuales proporcionan catálogos de elementos reusables en el diseño de sistemas de software, evitando así la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.

2.6.1 Patrones GRASP

Los patrones GRASP (General Responsibility Assignment Software Patterns) son patrones generales de software para asignación de responsabilidades.(38)

Con el objetivo de obtener un sistema flexible y reusable, se pusieron en práctica los siguientes patrones GRASP en el diseño de la solución:

- **Experto:** Determina la clase que posee mayor jerarquía para asignarle una responsabilidad según la información que posee, garantizando el encapsulamiento de la información y facilitando el bajo acoplamiento en las aplicaciones. El uso de este patrón se evidencia en las clases *DatIndicador*, *DatCriterioMedida* y *DatRegla* las cuales contienen la información necesaria para manejar la evaluación de los objetivos según indicadores, criterios de medida y reglas respectivamente.
- **Creador:** Se utiliza cuando se quiere a partir de una clase con alta jerarquía obtener clases descendientes o instancias a partir de las clases obtenidas. Se evidencia su uso en la clase *CmpgestionarevaluacionobjetivosController* la cual se encarga de crear las instancias de la clase *CmpgestionarevaluacionobjetivoModel*, para así usar las funcionalidades de esta
- **Bajo Acoplamiento:** Se encarga de mantener las clases lo menos ligadas entre sí posible. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.
- **Alta Cohesión:** Se basa en que la información que almacena una clase sea coherente y esté en la medida de lo posible, relacionada con la clase. Con el uso de este patrón se logra un incremento de la claridad y comprensión, lo que simplifica el mantenimiento. Este patrón fue utilizado en el diseño del paquete de funcionalidad de manera general.
- **Controlador:** Define quién debe responder a determinados eventos. Una clase controladora debe ser la encargada de manejar los eventos dentro de la funcionalidad, así la aplicación del patrón conlleva a separar la lógica de negocios de la capa de presentación, al aplicar estos principios, el controlador no realiza las actividades mencionadas sino que las delega en otras clases. El patrón Controlador se ve evidenciado en la solución en la clase *CmpgestionarevaluacionobjetivosController*.

2.6.2 Patrones GoF (Gang Of Four)

Los patrones GoF describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos y otros ámbitos referentes al diseño de interacción o interfaces.(38) Estos se clasifican según su propósito en 3 grupos: Creacional, Estructural, Comportamiento.

En la solución propuesta fueron utilizados los siguientes patrones GoF:

- **Fachada (Estructural):** Este patrón proporciona una interfaz unificada de alto nivel para un subsistema, que oculta las interfaces de bajo nivel de las clases que lo implementan simplificando así la interacción con el subsistema. Se utiliza para proporcionar un fácil acceso a subsistemas complejos. En el diseño de la solución que se propone, la clase que se utiliza como fachada es *ZendExt_IoC*, para acceder a los servicios de otros componentes.
- **Cadena de Responsabilidad (Comportamiento):** Este patrón permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada. Se evidencia al distribuir las responsabilidades, ya que ante la ocurrencia de un error al realizarse una determinada consulta a la base de datos el mismo es manejado por el Modelo, creando una nueva excepción de tipo *ZendExt_Exception*. Dicha excepción debe ser propagada al Controlador, el cual será el encargado de capturarla y enviarla a la Vista ya traducida, esta última por su parte mostrará un mensaje al usuario en un lenguaje entendible notificando el error.

2.7 Conclusiones del capítulo

El desarrollo del capítulo actual permitió arribar a las siguientes conclusiones:

4. Con la modelación de negocio se definieron los principales conceptos y procesos asociados a la implementación del Gestor de reglas y criterios de medida para evaluar el cumplimiento de objetivos estratégicos, lo que permitió tomar mayor entendimiento del negocio.
5. Llevar a cabo la captura, descripción y validación de requisitos funcionales y no funcionales posibilitó tomar mayor conocimiento sobre los principales requerimientos que debe cumplir la solución propuesta.

6. Se obtuvo como resultado el diseño de la solución en términos componentes posibilitando establecer las bases para el desarrollo de la solución.
7. El diseño de las clases principales del negocio permitió una mejor comprensión de cómo están estructuradas las clases que conforman la solución y la relación entre ellas. Esto permitió diseñar el modelo de datos correspondiente a la solución propuesta.
8. La definición de los patrones de diseño necesarios para el desarrollo posibilitó una entrada apropiada como punto de partida a las actividades de implementación, con la máxima de lograr una mayor calidad del producto y la satisfacción del cliente. Por consiguiente, una vez precisadas las funcionalidades y la estructura de la solución es posible el comienzo a la implementación de la solución.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN

3.1 Introducción

En el presente capítulo se detalla la arquitectura de la solución, los estándares de codificación por los que se registrará el código fuente, la estrategia de integración a seguir y demás elementos referentes a la implementación de la solución: “Gestor de reglas y criterios de medida para evaluar el cumplimiento de los objetivos estratégicos en el SIPAC”. Además se valida la solución a través de un conjunto de pruebas para garantizar que cumpla con las métricas de calidad establecidas y corregir posibles errores existentes.

3.2 Implementación de la solución propuesta.

3.2.1 Evaluación de un objetivo en la solución implementada

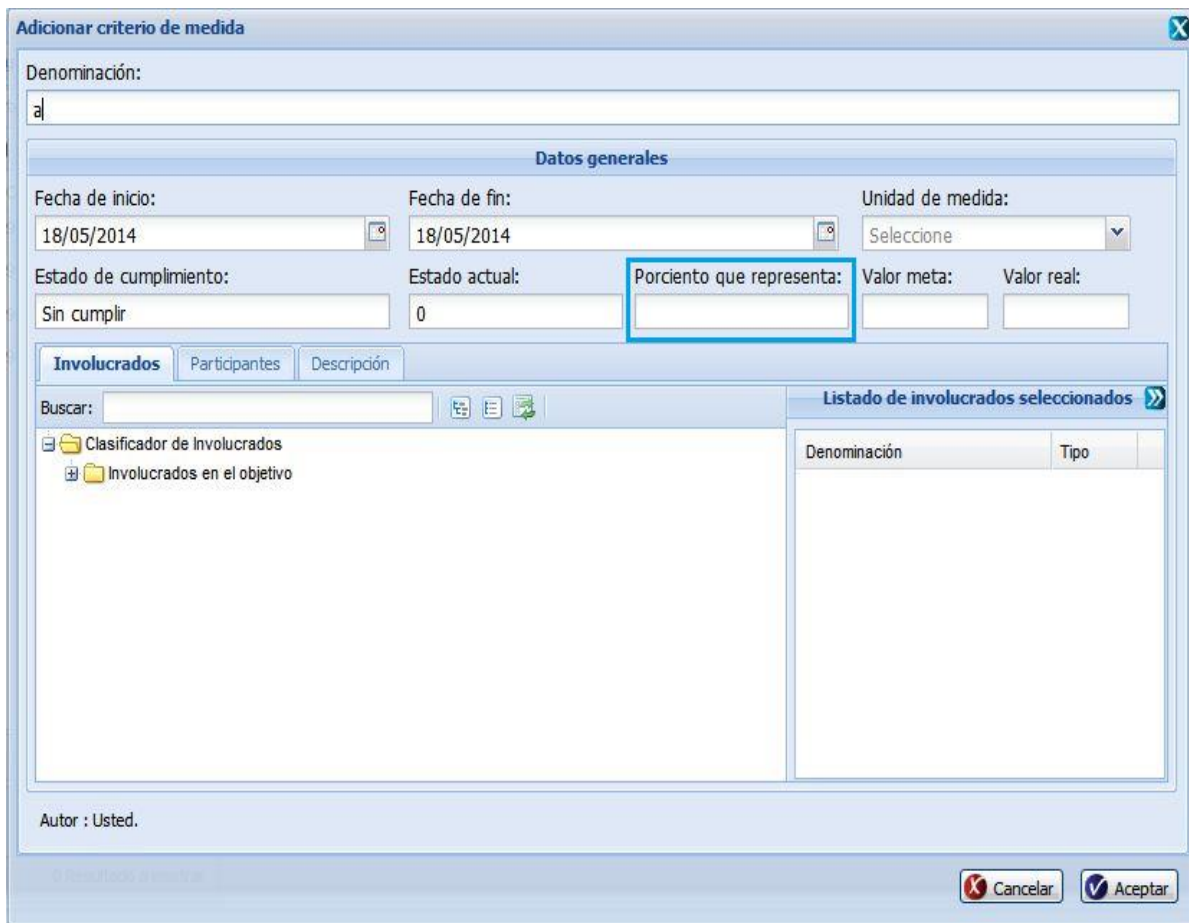
El paquete funcionalidades implementado para gestionar reglas y criterios de medida para evaluar el cumplimiento de los objetivos estratégicos en el sistema SIPAC provee al usuario de una interfaz amigable que le permite evaluar un objetivo. A continuación se detallan los pasos a seguir para la ejecución de dicho proceso:

1. Se selecciona el objetivo que se desea evaluar del registro de objetivos, el sistema muestra la interfaz Configurar parámetros de evaluación de objetivos, ver Figura 10. Dicha interfaz está compuesta por tres pestañas mediante las cuales se configura la evaluación de un objetivo a través de la gestión de criterios de medida, reglas e indicadores.



Figura 9. Configurar parámetros de evaluación de un objetivo.

2. En correspondencia con el tipo de planeación se configura la evaluación del objetivo:
 - 2.1 Si se desea configurar la evaluación de un objetivo trazado en la Planeación Operativa; la cual se encarga de establecer metas a cumplirse a corto plazo, el sistema permite hacerlo mediante criterios de medida. Al insertar los criterios de medida se les debe especificar como uno de sus parámetros, el porcentaje que representan en el cumplimiento del objetivo en cuestión. Ver Figura 11.



Adicionar criterio de medida

Denominación:
a

Datos generales

Fecha de inicio: 18/05/2014 Fecha de fin: 18/05/2014 Unidad de medida: Seleccione

Estado de cumplimiento: Sin cumplir Estado actual: 0 Porcentaje que representa: Valor meta: Valor real:

Involucrados Participantes Descripción

Buscar:

- Clasificador de involucrados
 - Involucrados en el objetivo

Listado de involucrados seleccionados

Denominación	Tipo
--------------	------

Autor : Usted.

Cancelar Aceptar

Figura 10. Adicionar criterio de medida

Es necesario que la suma de los porcentos que representa cada uno de los criterios de medida asociados al objetivo sea igual a 100, lo cual expresa el cumplimiento total del objetivo, de lo contrario el sistema no permite obtener la evaluación deseada.

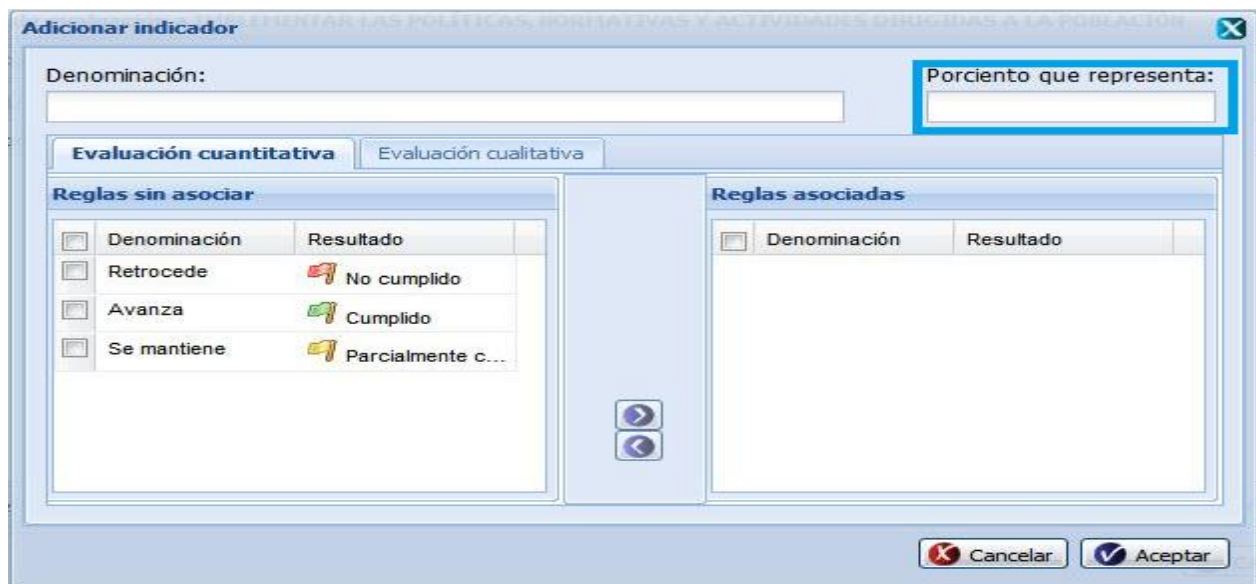
2.2 Si se pretende configurar la evaluación de un objetivo estratégico (Planeación Estratégica), el Gestor de reglas y criterios de medida del sistema SIPAC permite ingresar reglas que expresan el nivel de cumplimiento de un indicador. Estas reglas permiten establecer condiciones que se deben cumplir para considerar el indicador: Cumplido, Parcialmente cumplido o No cumplido. Ver Figura 12.



Figura 11. Adicionar regla

Para la definición de una regla se pueden insertar fórmulas matemáticas haciendo uso de los operadores aritméticos: suma, resta, multiplicación y división. Ver Anexo 4. Se puede, además, asociar criterios de medida como variables en la confección de la fórmula. Ver Anexo 5.

Una vez definidas las reglas se definen indicadores, los cuales también tienen asociado el parámetro: por ciento que representa, y de igual manera que para los criterios de medida, su suma debe ser igual a 100. Ver Figura 13.



Reglas sin asociar	Denominación	Resultado
<input type="checkbox"/>	Retrocede	No cumplido
<input type="checkbox"/>	Avanza	Cumplido
<input type="checkbox"/>	Se mantiene	Parcialmente c...

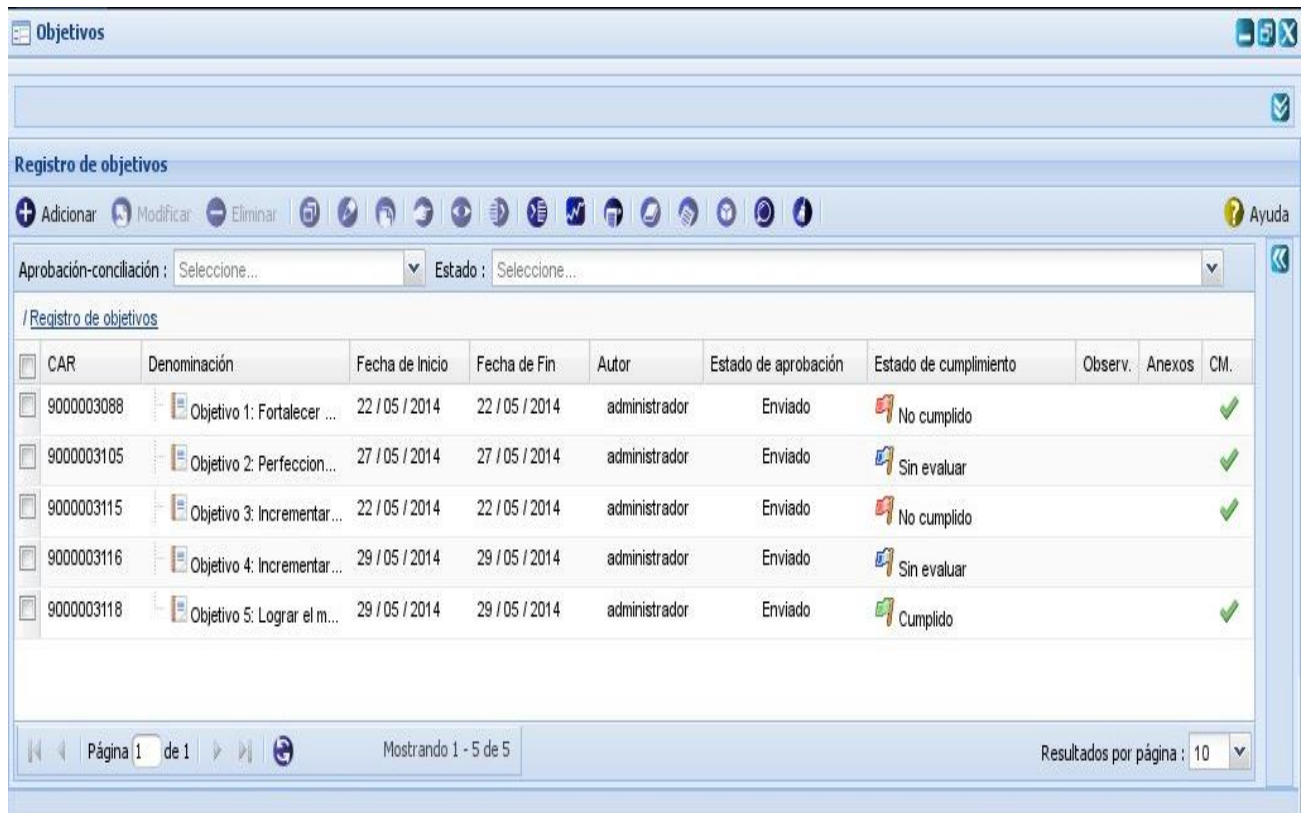
Reglas asociadas	Denominación	Resultado
<input type="checkbox"/>		

Figura 12. Adicionar indicador para evaluación cuantitativa

Las reglas son asociadas al indicador para proporcionar una evaluación cuantitativa del mismo. Un indicador a su vez, puede ser evaluado cualitativamente mediante la asociación de actividades que contribuyan al cumplimiento del indicador. Ver Anexo 6.

3. Una vez configurados los parámetros anteriormente presentados se puede evaluar el objetivo deseado, para ello es necesario especificar el rango de evaluación que cada planificador, acorde a las necesidades, desee establecer. Esto permitirá, según el porcentaje de cumplimiento que presente el objetivo, obtener una calificación que exprese si el objetivo fue cumplido, parcialmente cumplido o no cumplido. Ver Anexo 7.

Se muestra el resultado de la evaluación del cumplimiento de los objetivos en el campo Estado de cumplimiento del Registro de objetivos. Ver Figura 14.



CAR	Denominación	Fecha de Inicio	Fecha de Fin	Autor	Estado de aprobación	Estado de cumplimiento	Observ.	Anexos	CM.
9000003088	Objetivo 1: Fortalecer ...	22 / 05 / 2014	22 / 05 / 2014	administrador	Enviado	No cumplido			✓
9000003105	Objetivo 2: Perfeccion...	27 / 05 / 2014	27 / 05 / 2014	administrador	Enviado	Sin evaluar			✓
9000003115	Objetivo 3: Incrementar...	22 / 05 / 2014	22 / 05 / 2014	administrador	Enviado	No cumplido			✓
9000003116	Objetivo 4: Incrementar...	29 / 05 / 2014	29 / 05 / 2014	administrador	Enviado	Sin evaluar			✓
9000003118	Objetivo 5: Lograr el m...	29 / 05 / 2014	29 / 05 / 2014	administrador	Enviado	Cumplido			✓

Figura 13. Registro de objetivos

3.2.2 Estándares de codificación

Los estándares de codificación permiten establecer reglas sobre la escritura del código fuente, permitiendo seguir un estilo de programación homogéneo de forma tal que todos los participantes de un proyecto puedan entender el mismo y el código, en consecuencia, sea mantenible.(39) A continuación se definen las tres partes fundamentales dentro de un estándar de programación:

- Convención de nomenclatura: Define cómo nombrar variables, funciones y clases.
- Convenciones de legibilidad de código: Es la forma de organizar el código y lograr que independientemente de quien desarrolle se entienda como un todo.
- Convenciones de documentación: Define cómo establecer comentarios, archivos de ayuda, entre otros.

Nomenclatura de las clases

Los nombres de las clases deben comenzar con mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing, la cual define que los identificadores y nombres de variables, métodos y clases que están compuestos por múltiples palabras juntas, inicia cada palabra con letra mayúscula y sin usar ningún artículo posibilitando que con solo leer el nombre de la clase ya se reconozca la función de la misma.

Nomenclatura según el tipo de clases

Clases controladoras: Las clases controladoras después del nombre llevan la palabra: “Controller”. Ejemplo: CmpGestionarEvaluacionObjetivosController.

Clases de los modelos:

- ❖ Business (Negocio): Las clases que se encuentran dentro de Business después del nombre llevan la palabra: “Model”. Ejemplo: CmpGestionarEvaluacionObjetivosModel.
- ❖ Domain (Dominio): Las clases que se encuentran dentro de Domain el nombre que reciben es el de la tabla en la base de datos. Ejemplo: DatRegla.
- ❖ Generated (Dominio base): Las clases que se encuentran dentro de Generated el nombre comienza con la palabra: “Base” y seguido el nombre de la tabla en la base de datos. Ejemplo: BaseDatRegla.

Nomenclatura de las funcionalidades

El nombre a emplear para las funciones se escribe con la inicial del identificador en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing que es similar a la PascalCasing con la excepción de la primera letra.

Nomenclatura según la clase donde se encuentren las funciones:

- En la clase controladora: Las principales funcionalidades de las clases controladoras se les pone el nombre y seguida la palabra: “Action”. Ejemplo: eliminarReglaAction().
- En las clases de los modelos: Las funcionalidades se nombran de manera que al leerlo se identifique su propósito. Ejemplo de función: eliminarRegla ().

Nomenclatura de los comentarios

Los comentarios deben ser lo bastante claros y precisos de forma tal que se entienda el propósito de lo que se está desarrollando. En caso de ser una función complicada se debe comentar para lograr una mejor comprensión del código.

3.2.3 Diagrama de despliegue

A continuación se describe como queda distribuido físicamente el sistema entre los diferentes nodos de cómputo. Existen dos posibles escenarios para el despliegue de la solución: El escenario para PC cliente con disco y el escenario para PC cliente sin disco, también conocido como cliente ligero.



Figura 14. Diagrama de despliegue de escenario para PC cliente con disco.

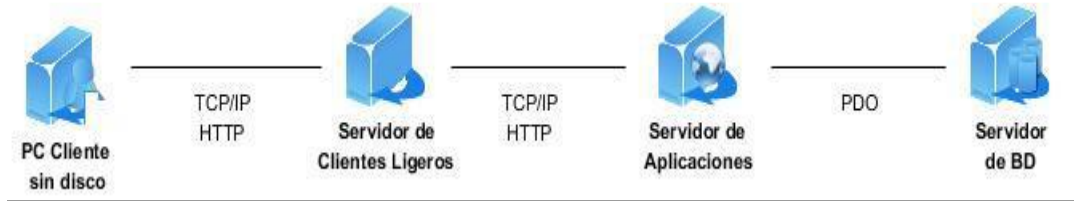


Figura 15. Diagrama de despliegue de escenario para PC cliente sin disco.

3.3 Validación de la solución propuesta

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es medible y varía de un sistema a otro, y se ha convertido en un elemento de gran importancia en la actualidad ya que influye directamente en la competitividad de las empresas.(40) Para detectar y corregir los posibles errores existentes en la solución implementada se detalla a continuación la validación de la solución propuesta así como las métricas para la validación del diseño propuesto y se describen las pruebas de aplicación realizadas.

3.3.1 Validación del diseño propuesto

Para validar las clases del diseño se utilizaron las métricas: Tamaño operacional de clases (TOC) y Relación entre clases (RC). Estas métricas fueron diseñadas para validar los siguientes atributos de calidad:

- Responsabilidad: consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto de la problemática propuesta.
- Complejidad de implementación: grado de dificultad que tiene implementar un diseño de clases determinado.
- Reutilización: grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- Acoplamiento: grado de dependencia o interconexión de una clase o estructura de clase con otras, está muy ligada a la característica de Reutilización.
- Complejidad del mantenimiento: grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software.
- Cantidad de pruebas: numero o grado de esfuerzo para realizar las pruebas de calidad (unidad) del producto (componente, modulo, clase, conjunto de clases, etc.) diseñado.

Métrica TOC: Tamaño operacional de clase

Está dado por el número de métodos asignados a una clase. En la aplicación de esta métrica se tuvieron en cuenta las clases del Modelo y del Domain, así como la clase Controladora de acuerdo al diseño de la solución desarrollada. Se tuvo en cuenta además, la cantidad de procedimientos que contienen cada una de las clases, obteniendo los siguientes resultados:

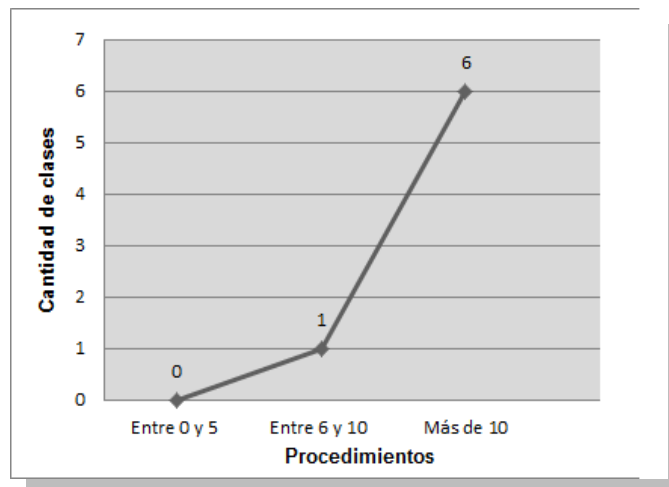


Figura 16. Representación de la cantidad de clases y el número de procedimientos que contienen.

Para ver el modelo de medición utilizado en la métrica TOC ir al Anexo 9.

Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos:



Figura 17. Representación de la cantidad de clases y el número de procedimientos que contienen.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Responsabilidad**:



Figura 18. Representación del valor en % del atributo Responsabilidad.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Complejidad de implementación**:

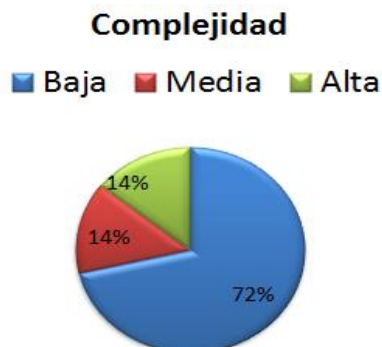


Figura 19. Representación del valor en % del atributo Complejidad de implementación.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Reutilización**:



Figura 20. Representación del valor en % del atributo Reutilización.

Análisis de los resultados obtenidos en la evaluación de la métrica TOC:

Teniendo en cuenta que el 72% de las clases contienen un número de funcionalidades inferior a la media de procedimientos por clases (cuyo valor asciende a 9) se puede afirmar que el diseño de clases elaborado se encuentra en un nivel satisfactorio dentro de los límites de calidad. Por consiguiente, el 72% de las clases que intervienen en el diseño tienen un nivel alto de Reutilización.

Métrica RC: Relación entre clases

Está dado por el número de relaciones de uso de una clase con otra. Dicha métrica está determinada por los atributos Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas, existiendo una relación directa con los tres primeros e inversa con el último antes mencionado. Al aplicarse la métrica Relación entre clases se obtuvo el siguiente resultado:

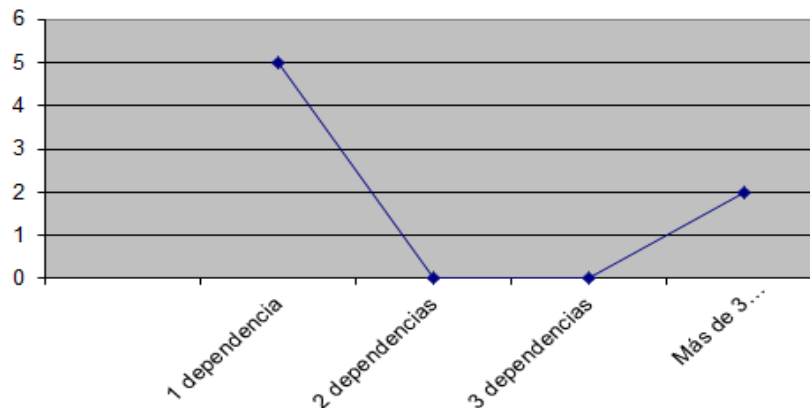


Figura 21. Representación de las asociaciones de uso por cantidad de clase.

Para ver el modelo de medición utilizado en la métrica RC ir al Anexo 10.

Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos:

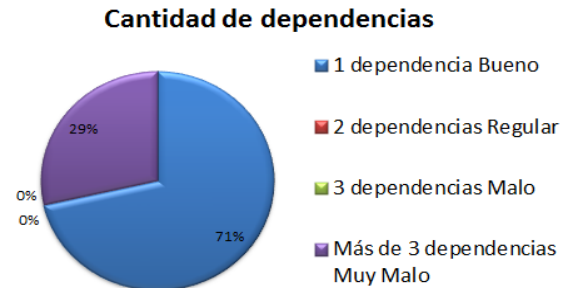


Figura 22. Representación de las asociaciones de uso por cantidad de clase.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Acoplamiento**:



Figura 23. Representación del valor en % del atributo Acoplamiento.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Complejidad de Mantenimiento**:

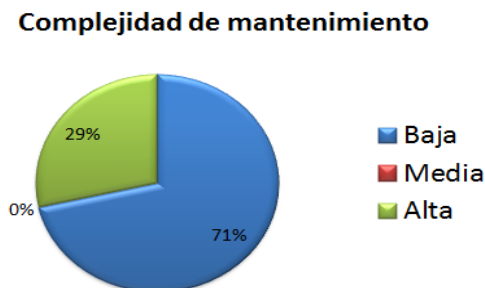


Figura 24. Representación del valor en % del atributo Complejidad de Mantenimiento.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Cantidad de pruebas**:



Figura 25. Representación del valor en % del atributo Cantidad de pruebas.

Representación en % de la incidencia de los resultados obtenidos en el atributo **Reutilización:**



Figura 26. Representación del valor en % del atributo Reutilización.

Análisis de los resultados obtenidos en la evaluación de la métrica RC:

Después de aplicar la métrica RC se demostró que el 71% de las clases poseen menos de 3 dependencias respecto a otras. Los atributos de calidad se encuentran en un nivel satisfactorio; ya que los atributos Complejidad de Mantenimiento, la Cantidad de Pruebas y la Reutilización se comportan favorablemente para un 71% de las clases.

3.3.2 Pruebas de software

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto software antes de su puesta en marcha. Básicamente, es una fase en el desarrollo de software que consiste en probar las aplicaciones construidas bajo condiciones especificadas, los resultados son observados o registrados, y una evaluación es realizada de un aspecto del sistema o componente.(41) Existen varios tipos de pruebas de software las cuales se describen a continuación:

- **Pruebas unitarias:** Se aplican a un componente o función del software. Su objetivo es aislar cada parte del programa y mostrar que las partes individuales son correctas. Tienen como propósito encontrar defectos en un módulo o función.
- **Pruebas de integración:** Consiste en construir el sistema a partir de los distintos componentes y probarlo con todos integrados. Su propósito es asegurar que no existan errores de interfaces y encontrar defectos en el sistema.
- **Pruebas de aceptación:** Son las únicas pruebas que son realizadas por los usuarios expertos, todas las anteriores las lleva a cabo el equipo de desarrollo. Consiste en comprobar si el producto está listo para ser implantado para el uso operativo en el entorno del usuario.
- **Pruebas funcionales:** Este tipo de prueba se realiza sobre el sistema funcionando, comprobando que cumpla con la especificación (normalmente a través de los casos de uso). Para estas pruebas, se utilizan las especificaciones de casos de prueba.

Métodos de prueba

Los métodos de prueba del software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. Ayudan a definir conjuntos de casos de prueba aplicando ciertos criterios. Los métodos de prueba que se especifican a continuación permiten probar cada una de las condiciones existentes en el programa, identificar claramente las entradas, salidas y estudiar las relaciones que existen entre ellas, permitiendo así maximizar la calidad de las pruebas y de este modo obtener un sistema más estable y confiable.

➤ Pruebas de caja blanca

También suelen ser llamadas estructurales o de cobertura lógica. Con ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba que ejecutan cada posible ruta del programa o módulo, considerándose una ruta como una combinación específica de condiciones manejadas por un programa.(41) Este tipo de prueba le permite al ingeniero de software obtener casos de pruebas que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.

- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Aplicación de las pruebas de caja blanca

Existen diferentes técnicas de pruebas de caja blanca:

Prueba del Camino Básico: Permite obtener una medida de la complejidad lógica de un diseño y usar la misma como guía para la definición de un conjunto de caminos básicos. Los casos de prueba obtenidos garantizan que durante la prueba se ejecute al menos una vez cada sentencia del programa.

Prueba de Condición: Ejercita las condiciones lógicas contenidas en el módulo de un programa. Garantiza la ejecución por lo menos una vez de todos los caminos independientes de cada módulo, programa o método.

Prueba de Flujo de Datos: Se seleccionan caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa. Garantiza que se ejerciten las estructuras internas de datos para asegurar su validez.

Prueba de Bucles: Método de prueba que se centra exclusivamente en la validez de las construcciones de bucles. Garantiza la ejecución todos los bucles en sus límites operacionales.

Con el objetivo de valorar la calidad con la que se llevó a cabo la implementación de la solución Gestor de reglas y criterios de medida para evaluar el cumplimiento de objetivos estratégicos, fue necesario aplicar una de las técnicas descritas anteriormente: **la prueba del camino básico**.

Para ello fue necesario seguir los siguientes pasos básicos:

1. A partir del diseño o del código fuente, dibujar el grafo de flujo asociado.
2. Calcular la complejidad ciclomática del grafo.
3. Determinar un conjunto básico de caminos independientes.
4. Preparar los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Dando cumplimiento a los anteriores pasos básicos se enumeran cada una de las sentencias de código de uno de los procedimientos de la clase *CmpgestionarevaluacionobjetivoController*, específicamente la funcionalidad *modificarindicadorAction*. Ver Anexo 8.

Luego de realizado el procedimiento del primer paso, se hace necesario representar el grafo de flujo asociado al código presentado anteriormente. Para ello se utilizan aristas, nodos y regiones.

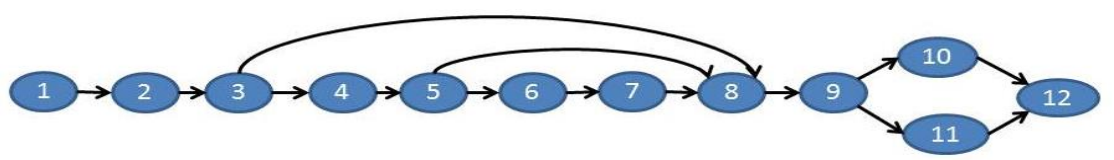


Figura 27. Grafo de flujo asociado al procedimiento *modificarindicadorAction*.

Una vez representado el gráfico de flujo asociado al procedimiento *modificarindicadorAction* se prosigue a calcular la complejidad ciclomática de este, la cual garantiza la menor cantidad de casos de pruebas a realizar, para que se ejecute al menos una vez cada sentencia de código. El cálculo de la complejidad ciclomática es necesario efectuarlo mediante tres vías o fórmulas, utilizando siempre el mismo grafo en cada caso

✓ **Fórmula 1:** $V(G) = (A - N) + 2$

A: cantidad total de aristas del grafo.

N: cantidad total de nodos del grafo.

Resultado: $V(G) = (14 - 12) + 2 = 4$

✓ **Fórmula 2:** $V(G) = P + 1$

P: cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

Resultado: $V(G) = 3 + 1 = 4$

✓ **Fórmula 3:** $V(G) = R$

R: cantidad total de regiones existentes en el grafo, se incluye el área exterior del grafo, como una región más.

Resultado: $V(G) = 4$

Luego de calculada la complejidad ciclomática a través de la fórmulas, arrojando un resultado de 4, se concluye que existen 4 caminos básicos, lo cual representa el número mínimo de casos de prueba que se deben aplicar para valorar la calidad con que se implementó la solución propuesta. A continuación se representan los 4 caminos básicos por donde puede circular el flujo asociado al procedimiento en cuestión:

Camino básico # 1: 1-2-3-8-9-10-12

Camino básico # 2: 1-2-3-4-5-8-9-10-12

Camino básico # 3: 1-2-3-4-5-6-7-8-9-10-12

Camino básico # 4: 1-2-3-4-5-6-7-8-9-11-12

A continuación se describen los casos de prueba para cada uno de los caminos básicos determinados en el grafo de flujo:

Tabla 6 . Caso de prueba para el camino básico # 1

Camino básico 1: 1-2-3-8-9-10-12	
Descripción	Se modifica el indicador seleccionado.
Condición de ejecución	El indicador que se desea modificar tiene reglas asociadas, las cuales son modificadas.
Entrada	\$id=9000003467 \$denindi= Cumplimiento de la zafra. \$porcentaje= 45% \$idobjetivo= 9000003345 \$idactividad= null \$idreglas= [9000005432, 9000006531]
Resultado esperado	Se muestra un mensaje al usuario diciendo que el indicador se ha modificado correctamente.
Resultado	Satisfactorio

Ir a Anexos 11, 12 y 13 para ver los restantes casos de prueba para los caminos básicos 2, 3, y 4 respectivamente.

Luego de aplicar los casos de prueba correspondientes a cada camino básico identificado en el grafo de flujo asociado, se pudo comprobar el correcto funcionamiento del método *modificarindicadorAction*.

➤ **Pruebas de caja negra**

También suelen ser llamadas funcionales y basadas en especificaciones. Con ellas se pretende examinar el programa para verificar que cuente con las funcionalidades que debe tener y la forma de llevar a cabo las mismas, analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos. Las pruebas no se hacen en base al código, sino a la interfaz. (41) Este tipo de prueba intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Aplicación de las pruebas de caja negra

Existen también diferentes técnica de caja negra:

- **Partición equivalente:** Se divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Se basa en una evaluación de las clases de equivalencia para una condición de entrada.
- **Análisis de valores límites (AVL):** Permite la elección de casos de prueba que ejerciten los valores límites. Es una técnica de diseño de casos de prueba que complementa la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de casos de prueba en los extremos de la clase, en lugar de centrarse en las condiciones de entrada, el AVL también obtiene casos de prueba para el campo de salida.

Se aplica la prueba de partición de equivalencia como parte de las pruebas de caja negra realizadas sobre el requisito funcional *Adicionar Regla*. Para ello se definieron variables de equivalencia que representan un conjunto de estados válidos (representan entradas válidas al programa) y no válidos (representan valores de entrada erróneos) para las condiciones de entrada del sistema.

Tabla 7. Caso de prueba de caja negra para validar el requisito funcional *Adicionar Regla*

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Adicionar regla.	1. El sistema debe permitir que el usuario al seleccionar la opción Adicionar Regla, adicione una nueva regla.	EP 1.1: Flujo Básico de eventos	<ol style="list-style-type: none"> 1. Presionar clic en la ruta Inicio – Planificación – SIPAC – Planeación – Objetivos – Configurar parámetros de evaluación del objetivo- Reglas. 2. Hacer clic en la opción Adicionar, llenar los datos y dar clic en la opción Aceptar. 3. Se adiciona la regla.
		EP 1.2: Se introducen valores incorrectos.	<ol style="list-style-type: none"> 1. Presionar clic en la ruta Inicio – Planificación – SIPAC – Planeación – Objetivos – Configurar parámetros de evaluación del objetivo - Reglas. 2. Hacer clic en la opción Adicionar e insertar caracteres diferentes de letras en la denominación y dar clic en la opción Aceptar. 3. El sistema no permite al usuario introducir caracteres diferentes de letras.
		EP 1.3: Se introduce información incompleta.	<ol style="list-style-type: none"> 1. Presionar clic en la ruta Inicio – Planificación – SIPAC – Planeación – Objetivos – Configurar parámetros de evaluación del objetivo - Reglas. 2. Hacer clic en la opción Adicionar, dejar campos en blanco y dar clic en la opción Aceptar. 3. El sistema muestra un mensaje indicándole al usuario que tiene campos en blanco y da la posibilidad de corregirlos.

Para ver los juegos de datos a probar, ir al Anexo 14.

3.3.3 Resultados de las pruebas aplicadas

Luego de aplicados los métodos de prueba a las funcionalidades implementadas se pudo concluir que los resultados obtenidos hasta el momento han sido satisfactorios desde el punto de vista funcional. Las no conformidades detectadas fueron debidamente atendidas en aras de lograr el correcto comportamiento de la solución desarrollada ante diferentes situaciones. A

continuación se muestra la cantidad de no conformidades detectadas en cada iteración de pruebas realizada:

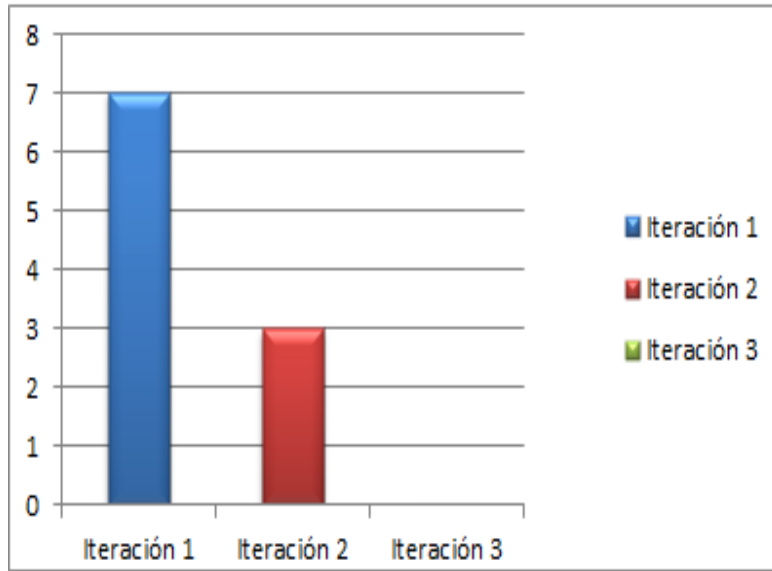


Figura 28. No conformidades detectadas durante las iteraciones de pruebas realizadas.

3.4 Conclusiones del capítulo

El desarrollo del capítulo actual permitió arribar a las siguientes conclusiones parciales:

9. La definición de los principales aspectos que influyeron en la implementación de la solución propuesta fue crucial para el desarrollo de una solución que evalúe el cumplimiento de objetivos estratégicos haciendo uso de indicadores, reglas y criterios de medida.
10. Se definió la estructura física de la solución haciendo también referencia a los estándares de codificación utilizados, los prototipos de las interfaces con las que podrá interactuar el usuario, así como los elementos necesarios para su despliegue representados en un diagrama por cada escenario.
11. Se validó además el diseño propuesto a partir de la aplicación de métricas (Tamaño operacional de clase y Relaciones entre clases) que permitieron evaluar aspectos como la complejidad de la implementación, la responsabilidad y reutilización de las clases.

12. La aplicación de pruebas de caja blanca (Técnica del camino básico) y pruebas de caja negra (Técnica partición de equivalencia) posibilitó verificar el correcto funcionamiento del Gestor de reglas y criterios de medida para evaluar el cumplimiento de objetivos estratégicos.

CONCLUSIONES GENERALES

La realización del presente trabajo y los resultados obtenidos con el mismo permitieron llegar a las siguientes conclusiones generales:

- El estudio de metodologías y sistemas informáticos que realizan la evaluación de objetivos, evidencia la necesidad de un sistema que permita evaluar el cumplimiento de los objetivos estratégicos durante la Planeación Estratégica y Operativa a todos los niveles organizacionales.
- La solución desarrollada permite un mayor seguimiento y control en la Planeación Estratégica y Operativa, mediante indicadores que cualifican el cumplimiento de los objetivos trazados en el plan de la organización, así como establecer un balance del cumplimiento de los indicadores y criterios de medidas.
- La evaluación de los elementos que influyen en el cumplimiento de un objetivo en el sistema SIPAC permite estimar los posibles riesgos en la ejecución del proceso de Planeación Estratégica y Operativa en las organizaciones.
- La obtención de una solución funcional que evalúa el cumplimiento de los objetivos de forma cuantitativa y cualitativa permite precisar los factores determinantes en el incumplimiento de un objetivo, los posibles involucrados, así como su impacto dentro de la organización.

RECOMENDACIONES

Al concluir el presente trabajo de diploma, considerando cumplidos los objetivos trazados en el mismo, se recomienda:

- Incorporar una funcionalidad que permita generar reportes que reflejen la información asociada a la evaluación del cumplimiento de los objetivos.
- Optimizar el algoritmo de evaluación del cumplimiento de los objetivos utilizando criterios de ponderación.
- Comprobar la factibilidad de la solución desarrollada tomando en cuenta diferentes entornos organizacionales.

BIBLIOGRAFÍA REFERENCIADA

1. CEIGE. *Manual de Usuario para el Sistema de Planificación de Actividades* [online]. 2012. Available from: SIPAC 2.0/Exp 3.4/1. Ingeniería/1.3 Implementación y prueba/manuales/Manual de Usuario 1.2
2. MARTINTO, Msc. Pedro Carlos Pérez. El diseño metodológico de la investigación científica. Teoría de Muestreo: población y muestra. Diseño experimental y métodos. In : .
3. Definición Planificación. [online]. [Accessed 22 May 2014]. Available from: <http://www.apuntesgestion.com/definicion-planificacion/>
4. Planeación Estratégica aplicada. [online]. [Accessed 22 May 2014]. Available from: <http://planeacionestrategicadbdb.blogspot.com/>
5. MACHADO SCULL, SANDY and BERNAL VIDAL, NÉSTOR and GALÁN RAMÍREZ, YULIET. *INFORMATIZACIÓN DE LA PLANIFICACIÓN POR OBJETIVOS PARA EL CONTEXTO NACIONAL ACTUAL*.
6. Objetivos Estratégicos de Hernán Araya Lallana en Prezi. [online]. [Accessed 22 May 2014]. Available from: <http://prezi.com/xazozgyxkrox/objetivos-estrategicos/>
7. Medición de Objetivos Cualificables. [online]. [Accessed 22 May 2014]. Available from: <http://winred.com/management/medicion-de-objetivos-cualificables/gmx-niv116-con3069.htm>
8. UNIVERSIDAD DE GRANADA. Guía para la definición e implantación de un sistema de indicadores. [online]. 15 November 2007. [Accessed 22 May 2014]. Available from: <http://www.ugr.es/~rhumasitioarchivos/noticias/Indicadores.pdf>
9. MONDRAGÓN PÉREZ, ANGÉLICA ROCÍO. ¿Qué son lo indicadores? [online]. [Accessed 2 April 2014]. Available from: <http://www.inegi.gob.mx/inegi/contenidos/espanol/prensa/contenidos/articulos/economicas/indicadores.pdf>
10. DR CASTELL- FLORIT SERRATE, C. PASTOR. “*METODOLOGIA PARA LA EVALUACIÓN DE CRITERIOS DE MEDIDA DEFINIDOS PARA ALCANZAR LOS OBJETIVOS ESTRATÉGICOS.*”
11. MSC. SOSA PORTEIRO, MARISEL. *Elementos del Sistema de Planificación establecidos en el Reglamento para la Implantación y Consolidación del Sistema de Dirección y Gestión Empresarial Estatal.*

12. Objetivos de Desarrollo del Milenio de la ONU. [online]. [Accessed 22 May 2014]. Available from: <http://www.un.org/es/millenniumgoals/stats.shtml>
13. GÓMEZ, GREGORIO. *EVALUACIÓN DEL CUMPLIMIENTO DE OBJETIVOS EN LOS SISTEMAS DE INCENTIVACIÓN DEL DESEMPEÑO*.
14. MONTOYA FREIRE, MARIA LORENA. Análisis, diseño e implementación de un sistema para seguir el cumplimiento de objetivos estratégicos por medio del cálculo automatizado de indicadores y reportes gráficos. .
15. DRA. GALARZA LÓPEZ, C. JUDITH. *UNA HERRAMIENTA INFORMÁTICA PARA APOYAR EL SEGUIMIENTO Y CONTROL DE LOS OBJETIVOS EN LAS INSTITUCIONES DE EDUCACIÓN SUPERIOR ADSCRITAS AL MINISTERIO DE EDUCACIÓN SUPERIOR (SASCO)*.
16. GONZÁLEZ CORNEJO ,JOSÉ ENRIQUE. El Lenguaje de Modelado Unificado (UML). [online]. Enero 2008. [Accessed 22 May 2014]. Available from: <http://www.docirs.cl/uml.htm>
17. UML: Un Lenguaje Modelo. *Cientec* [online]. [Accessed 22 May 2014]. Available from: <http://www.cientec.com/analisis/ana-uml.html>
18. INTRODUCCION A HTML (páginas web). [online]. [Accessed 22 May 2014]. Available from: <http://eusalud.uninet.edu/Cursos/doc99/INTROHTML.html>
19. Manual de CSS, hojas de estilo. [online]. [Accessed 22 May 2014]. Available from: <http://www.desarrolloweb.com/manuales/manual-css-hojas-de-estilo.html>
20. Definicion de JavaScript. [online]. [Accessed 22 May 2014]. Available from: <http://www.pergaminovirtual.com.ar/definicion/JavaScript.html>
21. PHP: Hypertext Preprocessor. [online]. [Accessed 22 May 2014]. Available from: <http://www.php.net/>
22. RECAMAN CHAUX, HERNANDO. Marco de trabajo para aplicaciones web de código abierto en instituciones universitarias. 29 February 2012.
23. ROB, ALLEN. Introducción a Zend Framework. *www.akrobat.com* [online]. 2010. [Accessed 22 May 2014]. Available from: <http://alemohamad.com/tutorial-zend-framework/>
24. Doctrine 1.2 ORM Manual. [online]. [Accessed 22 May 2014]. Available from: <http://docs.doctrine-project.org/projects/doctrine1/en/latest/en/manual/introduction.html#about-this-version>

25. Curso extjs. *Buenas Tareas* [online]. November 2011. [Accessed 22 May 2014]. Available from: <http://www.buenastareas.com/ensayos/Curso-Extjs/3156528.html>
26. Introducción a AJAX. *Libros Web* [online]. [Accessed 22 May 2014]. Available from: http://librosweb.es/ajax/capitulo_1.html
27. What's new in Visual Paradigm 11.1? [online]. [Accessed 22 May 2014]. Available from: <http://www.visual-paradigm.com/whats-new/>
28. Apache Subversion. [online]. [Accessed 22 May 2014]. Available from: <http://subversion.apache.org/>
29. NetBeans IDE -The Smarter and Faster Way to Code. [online]. [Accessed 22 May 2014]. Available from: <https://netbeans.org/features/index.html>
30. Apache - Servidor HTTP Apache 2.0. [online]. [Accessed 22 May 2014]. Available from: <http://httpd.apache.org/docs/2.0/es/invoking.html>
31. RENDÓN ARTOLA, ARIADNA. *CIG-SPA-N-i3504-DT (Vista Entorno de Desarrollo Tecnológico)*.
32. CEIGE, UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS. *Modelo de Desarrollo*.
33. *IEEE: Standard Glossary of Software Engineering Terminology*. [no date].
34. IAN SOMMERVILLE. *Ingeniería de Software*. 7ma. [no date].
35. ARIAS CHAVES, MICHAEL. 790-1199-1-PB (La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software.).
36. LÓPEZ JIMÉNEZ, SUSANA ALEJANDRA. *fbd-tutorial-i2(modelo de datos)*.
37. RENDÓN ARTOLA, ARIADNA. *CIG-SPA-N-i3502-AB (Arquitectura de Software-Guía Base-SIPAC 2.0)*.
38. Patrones de Diseño.pdf. [online]. [Accessed 29 March 2014]. Available from: [https://eseida.wikispaces.com/file/view/Tema 6/ Patrones de Diseño.pdf](https://eseida.wikispaces.com/file/view/Tema+6/+Patrones+de+Diseño.pdf)
39. CENTRO DE INFORMATIZACIÓN DE ENTIDADES. *Estándares de codificación Proyectos con el marco de trabajo Sauxe del CEIGE*.
40. ACIMED - Un enfoque actual sobre la calidad del software. [online]. [Accessed 22 May 2014]. Available from: http://scielo.sld.cu/scielo.php?pid=S1024-94351995000300005&script=sci_arttext
41. SERGIO OCHOA D. CC51A – Ingeniería de Software-Pruebas de Software.

BIBLIOGRAFÍA CONSULTADA

1. Castro, Raúl. Instrucción No. 1 Del Presidente de Los Consejos de Estado y de Ministros para la Planificación de los Objetivos y Actividades en los Órganos, Organismos de La Administración Central del Estado, Entidades Nacionales y las Administraciones Locales del Poder Popular. Habana, 2012.
2. Bernal Vidal, Néstor y Galán Ramírez, Yuliet. Proceso de Planificación por Objetivos en las entidades de las FAR. UCI. Habana, 2008.
3. Rendón Artola, Ariadna y Castellano Pérez, Manuel A. Modelación y construcción de los componentes Gestor de actividades y Calendario para el Subsistema Planificación por Objetivos del Sistema Integral de Gestión de Entidades Cedrux. UCI. Habana, 2011.
4. Fernández González, Mairelys y Zorrilla Rivera, Osley. Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Sistema Integral de Gestión de Entidades CEDRUX. UCI. Habana, 2010.
5. Pressman, Roger. Ingeniería de Software. Un enfoque práctico, 1998.
6. Ramírez, Jaime. Unidad de Programación. Métodos de Prueba del Software.
8. González Doria, Heidi. Métricas en el desarrollo del Software. Capítulo 4. Universidad de las Américas Puebla, 2010.
9. Salazar, Ing. Tte. Leonel. Curso de Capacitación por Roles: Generalidades de los Patrones de Diseño. s.n. Habana, 2008.
10. Saavedra Jorge. El mundo informático, 2013. [<http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/>].
11. Prieto, Félix. 2009. Patrones de diseño.
12. Díaz, Antonio. 2001. IRqA y el desarrollo de proyectos: Experiencias Prácticas. I Jornadas e Ingeniería de Requisitos JIRA 2001. s.n. Sevilla, España, 2001.
13. Sommerville, Ian. 2005. Ingeniería de Software. Séptima edición. Madrid: Pearson Educación S.A., 2005. 84-7829-074-5.

TÉRMINOS

GLOSARIO DE TÉRMINOS

CG (Control de gestión): proceso mediante el cual los directivos aseguran la obtención de recursos y su utilización eficaz y eficiente en el cumplimiento de los objetivos de la organización.

Estrategias Maestras: Misiones, propósitos, objetivos y políticas básicas

Áreas de Resultados Claves: son aquellas áreas o líneas de trabajo dentro de la organización de cuyo desempeño depende el cumplimiento de la misión.

DocIRS (*Document Information Retrieval Systems*, 'Sistemas de Recuperación de Información de Documentos'): es una empresa experta en desarrollar soluciones en el ámbito de los sistemas de información, con el objetivo de contribuir como equipo al negocio del cliente.

DOM (*Document Object Model*, 'Modelo de Objetos del Documento'): es una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

XML (*eXtensible Markup Language*, 'Lenguaje de marcas extensible'): es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

Herramientas CASE (*Computer Aided Software Engineering*, 'Ingeniería de Software Asistida por Computadora'): son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software.

Plugins (Complemento): es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

IEEE (*Institute of Electrical and Electronics Engineers*, 'Instituto de Ingeniería Eléctrica y Electrónica'): es una asociación mundial de técnicos e ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas.

SQLite: es una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración.

ANEXOS

Anexo 1. Figura ampliada de la *vista* del patrón arquitectónico Modelo-Vista-Controlador del diagrama de clases del diseño.

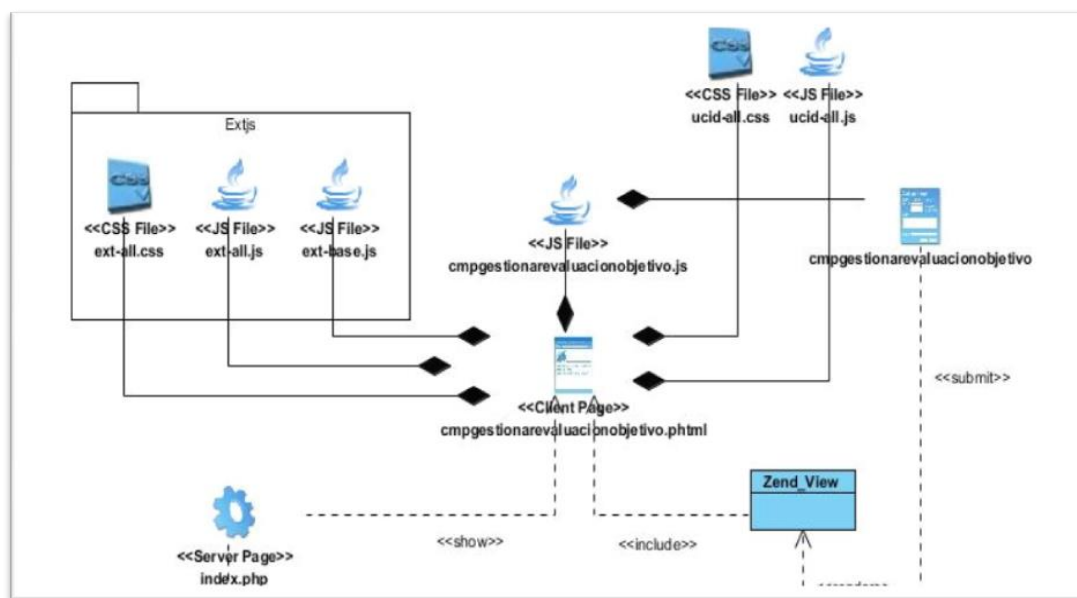


Figura 29. Vista del patrón arquitectónico Modelo-Vista-Controlador del diseño de clases.

Anexo 2. Figura ampliada del *Controlador* del patrón arquitectónico Modelo-Vista-Controlador del diagrama de clases del diseño.

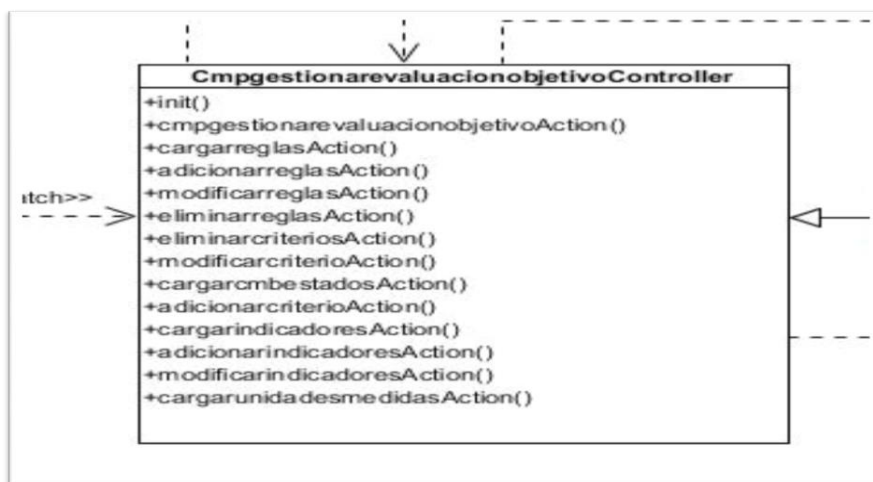


Figura 30. Controlador del patrón arquitectónico Modelo-Vista-Controlador del diseño de clases.

Anexo 3. Figura ampliada del *modelo* del patrón arquitectónico Modelo-Vista-Controlador del diagrama de clases del diseño.

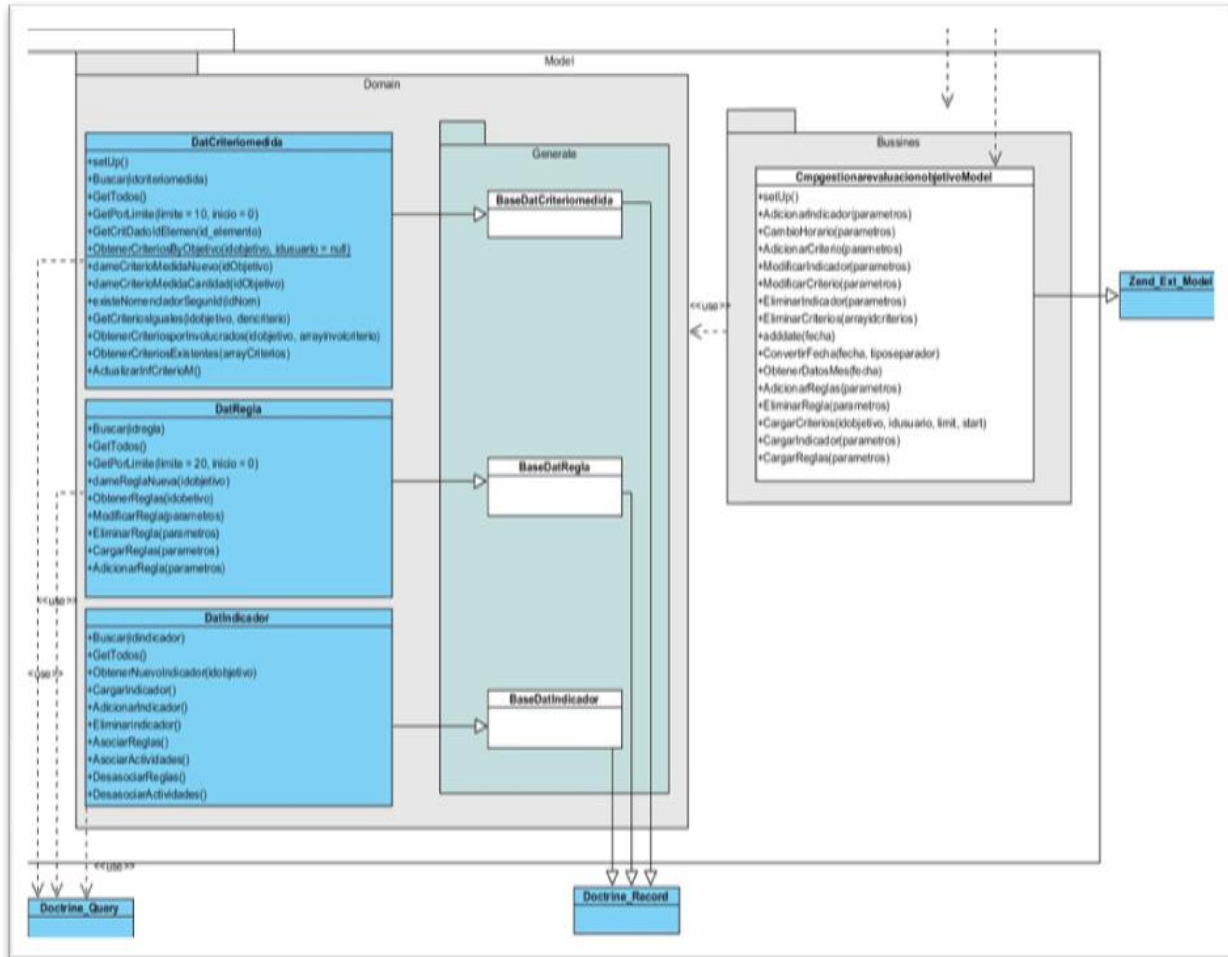


Figura 31. Modelo del patrón arquitectónico Modelo-Vista-Controlador del diseño de clases.

Anexo 4. Adicionar elemento a regla.



Figura 32. Adicionar elemento a regla

Anexo 5. Asociar criterio de medida a regla

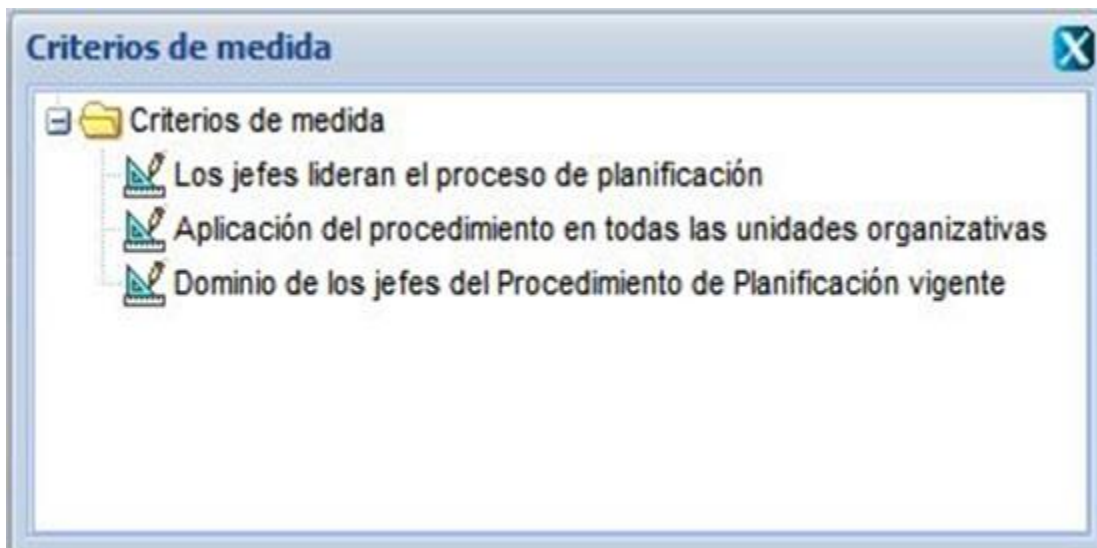


Figura 33. Asociar criterio de medida a regla

Anexo 6. Adicionar indicador para evaluación cualitativa

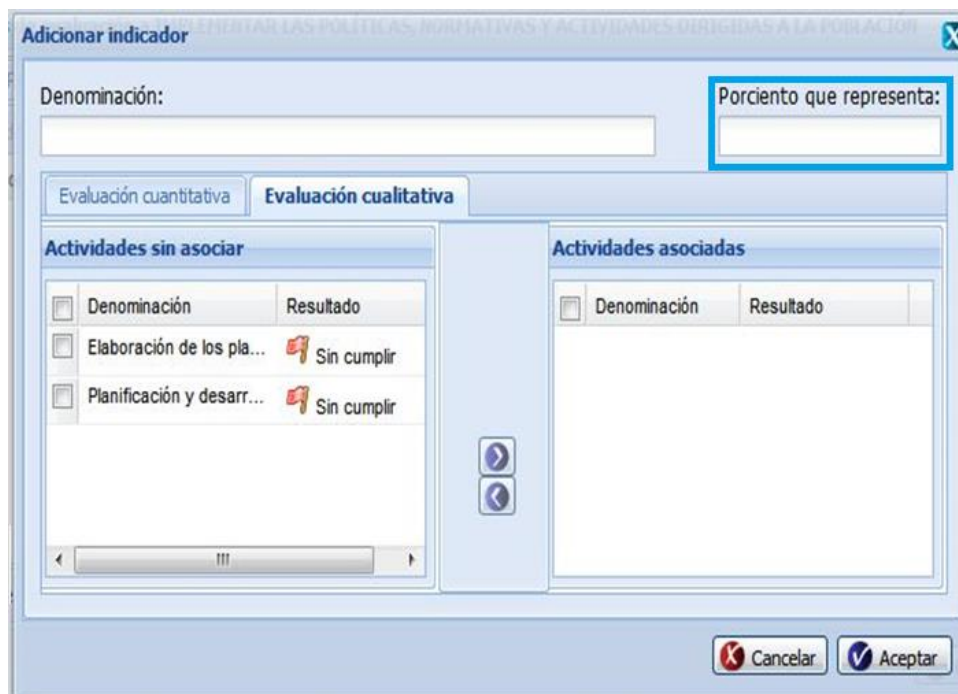


Figura 34. Adicionar indicador para evaluación cualitativa

Anexo 7. Configurar rangos para la evaluación del objetivo

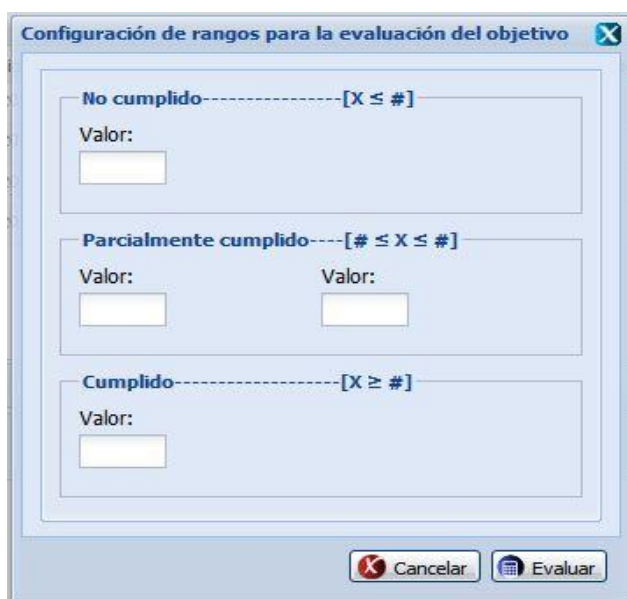


Figura 35. Configurar rangos para la evaluación del objetivo

Anexo 8. Fragmento de código con el algoritmo *modificarindicadorAction*

```

//funcion para modificar indicadores
function modificarindicadorAction() { //1
    $id = $this->_request->getPost('id');//1
    $denindi = $this->_request->getPost('denom');//1
    $porcentaje = $this->_request->getPost('porcentaje');//1
    $idobjetivo = $this->_request->getPost('idobjetivo');//1
    $idactividad = json_decode(stripslashes($this->_request->getPost('idactividad')));//1
    $idreglas = json_decode(stripslashes($this->_request->getPost('idregla')));//1

    $datindicador = new DatIndicadores();//1
    $existe = $datindicador->ExisteIndicadorAsd($denindi,$idobjetivo);//1

    if($idreglas != null){ //2
        $modif = new CmpgestionarevaluacionobjetivoModel();//3
        $modificar=$modif->modificarIndicador($id,$denindi,$porcentaje);//3
        $regla= $modif-> EvaluarContenido($denindi);//3
        $send= $modif-> ModificarResultado($denindi,$regla);//3
    }

    if($idactividad != null){ //4
        $modificar=$datindicador->modifIndicador_Actividades($idactividad,$id,$denindi,$porcentaje); //5
    }

    if($idactividad == null &&$idactividad == null ){ //6
        $modificar=$modif->modificarIndicador($id,$denindi,$porcentaje); //7
    }

    if($existe==false){ //8

        if($modificar){ //9
            echo("{'codMsg':1,'mensaje': 'Se ha modificado el indicador correctamente.'}");//10
        }

    } else{ //11
        echo ("{'codMsg':3 , 'mensaje' : ' La denominación de el indicador ya existe por favor verifique ' }");//11
    }

} //12

```

Figura 36. Fragmento de código con el algoritmo *modificarindicadorAction*.

Anexo 9. Material de medición de la métrica Tamaño operacional de la clase (TOC)

Tabla 8. Material de medición de la métrica Tamaño operacional de la clase (TOC)

Características	Categoría	Criterio
Responsabilidad	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Complejidad implementación	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.

	Media	Entre Prom. y 2* Pom.
	Alta	<= Prom.

Anexo 10. Material de medición de la métrica Relaciones entre clases (RC)

Tabla 9. Material de medición de la métrica Relaciones entre clases (RC)

Características	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	> 2
Complejidad mantenimiento	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	<= Prom.
Cantidad de pruebas	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.

Anexo 11. Caso de prueba para el camino básico # 2

Tabla 10. Caso de prueba para el camino básico # 2

Camino básico 2: 1-2-3-4-5-8-9-10-12	
Descripción	Se modifica el indicador seleccionado.
Condición de ejecución	El indicador que se desea modificar tiene actividades asociadas, las cuales son modificas.

Entrada	\$id=9000002547 \$denindi= Funcionamiento del CB/UJC. \$porciento= 25% \$idobjetivo= 9000002275 \$idactividad= [9000002653, 9000002014] \$idreglas= null
Resultado esperado	Se muestra un mensaje al usuario diciendo que el indicador se ha modificado correctamente.
Resultado	Satisfactorio

Anexo 12. Caso de prueba para el camino básico # 3

Tabla 11. Caso de prueba para el camino básico # 3

Camino básico 3: 1-2-3-4-5-6-7-8-9-10-12	
Descripción	Se modifica el indicador seleccionado.
Condición de ejecución	Al indicador que se desea modificar no se le modifica las reglas ni actividades asociadas.
Entrada	\$id=9000003452 \$denindi= Producción de láminas de aluminio. \$porciento= 10% \$idobjetivo= 9000002159 \$idactividad= null \$idreglas= null
Resultado esperado	Se muestra un mensaje al usuario diciendo que el indicador se ha modificado correctamente.
Resultado	Satisfactorio

Anexo 13. Caso de prueba para el camino básico # 4

Tabla 12. Caso de prueba para el camino básico # 4

Camino básico 4: 1-2-3-4-5-6-7-8-9-11-12	
Descripción	No se modifica el indicador correctamente.
Condición de ejecución	Existe en el sistema un indicador con igual denominación a la que el usuario ha insertado al modificar el indicador seleccionado.
Entrada	\$id=9000002547 \$denindi= Funcionamiento del CB/UJC. \$porciento= 45% \$idobjetivo= 9000003851 \$idactividad= null \$idreglas= [9000004792, 9000002013]
Resultado esperado	Se muestra un mensaje al usuario diciendo que ya existe un indicador con esa denominación.
Resultado	Satisfactorio

Anexo 14. Juego de datos a probar para el caso de prueba del requisito funcional: Adicionar reglas.

Tabla 13. Juego de datos a probar para el caso de prueba del requisito funcional: Adicionar reglas.

Id del escenario	Escenario	Denominación	Resultado	Elemento	Signo	Valor	Operador	Respuesta del sistema
o								

EP 1.1:	Adicionar una nueva regla introduciendo datos válidos.	V(Cantidad de graduados) o)	V(Cumplido)	4	V(=)	4	V(Y_)	El sistema muestra una ventana informando que los datos han sido guardados satisfactoriamente.
		V(Disminuir los costos de producción)	V(Parcialm ente cumplido)	V(20)	V(>)	V(15)	V(O_)	
EP 1.2:	Adicionar una nueva regla dejando campos requeridos en blanco.	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío) o)	I(Vacío) o)	I(Vacío)	Se muestra el campo requerido, sin llenar, en rojo que indica que no se pueden dejar campos requeridos en blanco y se muestra una ventana informando que hay datos incorrectos.
EP 1.3:	Adicionar una nueva regla introduciendo datos inválidos.	I(%&/\$.*14)	V(No cumplido)	V(5)	V(<)	V(10)	I(***)	El sistema no permite que se introduzcan caracteres extraños ni números en campos de texto que solo admiten ni escribir en el campo de selección Operador. El sistema muestra el mensaje: "Por favor verifique

que hay campo(s) con valores incorrecto(s).”

V(Disminuir los costos de producción)	I(Bien)	V(5)	V(=)	V(45)	V(Y_)	El sistema no admite escribir en el campo de selección Resultado, se muestra una ventana
V(Disminuir los costos de producción)	V(Cumplido)	I(Todos)	V(=)	I(Todos)	V(O_)	El sistema no permite escribir letras en los campos de selección Elemento y Valor, se muestra una ventana
V(Disminuir los costos de producción)	V(Cumplido)	V(30)	I(Todos)	V(52)	V(O_)	El sistema no permite escribir en el campo Signo.

EP 1.4	Cancelar	NA	NA	NA	NA	NA	NA	Se cancela la operación y se cierra la ventana.
--------	----------	----	----	----	----	----	----	---