

Universidad de las Ciencias Informáticas

Facultad 2



**Sistema de gestión de indicadores de ciencia y técnica y sus
proyecciones en CESIM**

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

Autores: Lisandra Hernández Sánchez.

Damián Abel Frías Céspedes.

Tutores: Ing. Leodan Vega Izaguirre.

Msc. Darling Darías Pérez.

La Habana, junio de 2014

“Año 56 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los <x> días del mes de junio del año 2014.

Autores:

Lisandra Hernández Sánchez

Autor 1

Damian Abel Frías Céspedes

Autor 2

Tutores:

Ing. Leodan Vega Izaguirre

Tutor principal

Msc. Darling Darías Pérez

Co tutor

DATOS DE CONTACTO

Tutores:

Ing. Leodan Vega Izaguirre. Profesor Asistente. Graduado de Ingeniero en Ciencias Informáticas, egresado de la UCI en el año 2008. Ha impartido las asignaturas de Matemática 1, Álgebra Lineal, Práctica Profesional, Gestión de Software, Sistemas de Base de Datos, Historia de la Informática y Metodología de la Investigación Científica. Se desempeñó como profesor en la antigua Facultad 7 y se desempeña actualmente como Subdirector del Centro de Informática Médica. Miembro de la Sociedad Cubana de Informática Médica.

Ing. Darling Darías Pérez. Profesora Instructora. Graduada de Ingeniero en Ciencias Informáticas, egresado de la UCI en el año 2008. Ha impartido las asignaturas de Gestión de Software, Sistemas de Base de Datos e Ingeniería de Software. Es profesora de la Facultad 2 y se desempeñó como Vicedecana de Investigación y Postgrado de la antigua Facultad 7. Graduada de Máster en Informática en Salud en el año 2014. Miembro de la Sociedad Cubana de Informática Médica.

DEDICATORIA

Lisandra:

Dedico este trabajo a mis padres, abuelos y hermanos pero especialmente a la persona que más quiero y admiro en el mundo por ser padre y madre “mi mamá”, Tamara Sánchez Reyes, a la cual le debo lo que soy y lo que seré en el futuro. Primeramente por haberme dado el privilegio de la vida, en la que me ha entregado, su amor, dedicación y esmero, por haber influido directamente en mi educación y formación, con sacrificio y apoyo incondicional, por haberme dado la oportunidad de contar con ella en todas las etapas de mi vida. Por esto y tantas razones le doy gracias, a mi Padre porque con sus virtudes y defectos lo idolatro con la vida y de una forma que no sabría expresar en palabras, siendo mi mayor anhelo que tanto el cómo mi madre y mi familia en general puedan estar siempre orgullosos de mí.

Damián:

Este trabajo va dedicado a mi familia por ser mi razón de ser, principalmente a las cuatro joyas más valiosas de mi vida: Mi mamá lo más importante de mi vida, mi abuela que sin ella no hubiese logro alguno, mi tía la guía más incondicional que tengo, y a mi novia por ser todas ellas en una sola.

AGRADECIMIENTOS

Lisandra:

A mis padres por su gran amor, apoyo incondicional y confianza plena.

A mis abuelos, gracias por haber fomentado en mí el deseo de superación y el anhelo de triunfo en la vida mil palabras no bastarían para agradecerles su apoyo, su comprensión y sus consejos en los momentos difíciles.

A mis hermanos, por su amor amistad y dedicación.

A mis tías Cari, Babi, Damaris, Mirtha, Leticia, Yunia por su entrega y dedicación, gracias las quiero mucho

A mi tía Marla que aunque no esté físicamente acompañadme en este momento sé que estaría muy feliz por mí, gracias te quiero.

A Ramón por ser un ejemplo a seguir y acogerme como su familia, gracias

A mi tutor, por guiándome y aconsejándome de manera especial para poder vencer las metas trazadas

A mi amigo más que amigo hermano Arnel por su amistad incondicional y ayudarme en cada momento que lo necesitaba.

A mi novio Yasel, gracias por estar a mi lado en las buenas y en las malas, por soportar mis malacrianzas y levantarme los ánimos cuanto más los necesito, no sabes lo mucho que significas para mí, por ser mi amigo y compañero, te amo mucho. Mi cuñadito Yasmany, te quiero mucho mi vida.

A mi familia habanera en especial a mis suegros Yoel y Rosy que es como mi otra mamá, gracias por apoyarme siempre, y acogerme como una más de la familia, los quiero.

A Yusvel por guiarme y enseñarme la mejor manera de ver las cosas para que no me volviera loquita cuando tenía cualquier problema con la tesis, gracias por todo el apoyo y las fuerzas.

A las personitas que compartieron la mayoría de los momentos que viví en la UCI, Yohana, Yasel, Isa, Yasniel, Maty, Yani, Carlos y familia, mi diablito Jorge, Daysi, Mailet, Yania y a todos los que de una forma u otra me brindaron su apoyo en un momento determinado cuando más lo necesitaba, gracias.

A todos mis amigos de la UCI, tanto profesores como estudiantes, por ofrecerme su amistad desinteresada y compartir tantos momentos lindos, que durarán en mi memoria para siempre.

Damián:

A toda mi familia por darme apoyo en todo momento

A mi tía por ser más que una madre para mí

A mi novia que mejor que ella imposible

A todos mis amigos, en especial a Maikel, Dunia, Yordanis, Leosvanis quienes mantuvieron su apoyo en todos los momentos.

A mis compañeros de la escuela.

Así como a todos los que de una forma u otra aportaron un grano de arena para mi preparación y para la realización de este trabajo.

RESUMEN

Las universidades del país tienen entre sus principales objetivos aumentar o propiciar la obtención de resultados de ciencia y técnica en cada una de las tareas planteadas en la práctica investigativa. La actualización y control de los indicadores se realizan de forma manual, provocando un alto nivel de desactualización de la información y haciendo engorroso el proceso. Atentando esto contra la integridad de la información que se gestiona.

El presente trabajo tiene como objetivo desarrollar un sistema que permite de forma homogénea, la gestión a todos los niveles de la información referente al desarrollo y la superación profesional, científica y técnica de los especialistas, adiestrados y trabajadores del CESIM.

Para la implementación se utilizó como IDE Visual Studio 2013 con tecnología ASP.NET MVC 4 y C-Sharp como lenguaje de programación. Para el modelado se empleó la herramienta *CASE Enterprise Architect 7.5*, BPMN 1.1 como notación para el modelado de los procesos de negocio y el Lenguaje Unificado de Modelado para la creación de los diagramas de ingeniería. Finalmente el proceso de desarrollo de software estuvo guiado por la metodología RUP.

El desarrollo de este sistema tiene como objetivo proteger la integridad de la información restringiendo los privilegios de inserción, eliminación y actualización en dependencia del rol que desempeñe cada usuario que hace uso del sistema.

Se logró un sistema que cumple con los requerimientos establecidos por el cliente, haciendo eficiente la gestión de los indicadores que se publican en el sistema a través de una interfaz de usuario flexible y productivo.

Palabras claves: Indicadores de ciencia y técnica, planificación de los indicadores, sistema de gestión de indicadores.

TABLA DE CONTENIDOS

INTRODUCCIÓN	10
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	2
1.1. CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.	2
1.2. METODOLOGÍA, LENGUAJES Y HERRAMIENTAS DE MODELADO A UTILIZAR PARA EL DESARROLLO.	6
1.3. ENTORNO DE DESARROLLO INTEGRADO.	9
1.4. TECNOLOGÍAS Y LENGUAJES DE PROGRAMACIÓN A UTILIZAR.....	10
1.5. SISTEMA DE GESTOR DE BASES DE DATOS.....	11
CONCLUSIONES PARCIALES.....	12
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.....	13
2.1. DESCRIPCIÓN DEL NEGOCIO ACTUAL.	13
2.2. TRABAJADORES DEL NEGOCIO.....	14
2.3. PROCESO DE NEGOCIO.....	15
2.4. SOLUCIÓN PROPUESTA.....	16
2.5. ESPECIFICACIÓN DE LOS REQUISITOS DE <i>SOFTWARE</i>	17
2.6. DEFINICIÓN DE LOS ACTORES DEL SISTEMA.	25
2.7. DIAGRAMA DE CASOS DE USO DEL SISTEMA.	26
2.8. DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.	26
2.9. CONCLUSIONES DEL CAPÍTULO.	29
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.....	30
3.1. MODELO ARQUITECTÓNICO.	30
3.2. DISEÑO.....	33
3.3. DESCRIPCIÓN DE LAS CLASES.	36
3.4. CONCLUSIONES DEL CAPÍTULO.	36
CAPÍTULO 4. IMPLEMENTACIÓN.....	37
4.1. ELEMENTOS DE IMPLEMENTACIÓN: MVC, NHIBERNATE Y JQUERY.....	37
4.2. DIAGRAMA DE COMPONENTES.....	40
4.3. DIAGRAMA DE DESPLIEGUE.....	42

4.4. ESTÁNDARES DE CODIFICACIÓN.	44
4.5. TRATAMIENTO DE ERRORES.	47
4.6. CONSTATACIÓN DE HIPÓTESIS.	47
4.7. CONCLUSIONES DEL CAPÍTULO.	47
CONCLUSIONES	49
RECOMENDACIONES	50
REFERENCIAS BIBLIOGRÁFICAS	51
BIBLIOGRAFÍA	54
ANEXOS	58
<i>ANEXO 1. DESCRIPCIÓN DEL PROCESO BALANCE DE CIENCIA Y TÉCNICA.</i>	58
<i>ANEXO 2. PREFIJOS DE LOS REQUERIMIENTOS NO FUNCIONALES POR CATEGORÍA.</i>	59
<i>ANEXO 3. DESCRIPCIÓN AMPLIADA DE LOS CASOS DE USO SIGNIFICATIVOS.</i>	59
<i>ANEXO 4. DIAGRAMAS DE CLASES DEL DISEÑO DE LOS CASOS DE USO ARQUITECTÓNICAMENTE SIGNIFICATIVOS.</i>	68
<i>ANEXO 5. DIAGRAMAS DE SECUENCIA DE LOS CASOS DE USO ARQUITECTÓNICAMENTE SIGNIFICATIVOS.</i>	70
<i>ANEXO 6. DESCRIPCIONES DE CLASES.</i>	74
<i>ANEXO 7. ENTREVISTAS A VICEDECANOS DE INVESTIGACIÓN Y POSTGRADO.</i>	84
GLOSARIO DE TÉRMINOS	85

INTRODUCCIÓN

La mayoritaria y amplia participación de los profesores y estudiantes universitarios en actividades de investigación científica, es un rasgo distintivo de la educación superior cubana. (1) La práctica investigativa en las universidades ha sido y es un factor de contribución a la elevación de la calidad del proceso de formación de profesionales, a la sólida sustentación del desarrollo de los grados científicos y maestrías como parte del proceso de formación académica, a la creación de nuevos conocimientos científicos sobre la naturaleza y la sociedad y en general, al desarrollo cultural y político de la nación.

La ciencia y la técnica que se desarrolla en las universidades con un alto nivel de jerarquización, tiene como objetivo contribuir de forma trascendente con los requerimientos del desarrollo económico y social que demanda el país, en correspondencia con el importante potencial científico técnico creado en esas instituciones. (2) Es razón por la cual, como parte de la misión del Ministerio de Educación Superior (MES) cubano, se particulariza, extender como parte de la actividad científica de las universidades el perfeccionamiento de las condiciones organizativas, jurídicas e institucionales para establecer tipos de organización económica que garanticen la combinación de investigación científica e innovación tecnológica, desarrollo rápido y eficaz de nuevos productos y servicios, su producción eficiente con estándares de calidad apropiados y la gestión comercializadora interna y exportadora, convirtiendo el vínculo científico y tecnológico con la universidad en un estímulo y aporte a la sociedad. (3)

En las universidades cubanas, con vista de lograr un mayor desempeño sobre la base de las tareas planteadas en las prácticas investigativas realizadas, se elabora un sistema de indicadores que posibilita a cada centro determinar la cantidad de publicaciones, participación en eventos, premios, capacitación, registros, maestrías y doctorados así como superación y categoría docente que deben cumplir las diferentes áreas, así como medir el cumplimiento de los objetivos principales del desarrollo científico y tecnológico, convirtiéndolo en promotor de dichas actividades mediante su utilización como instrumento de gestión.

En la actualidad la Universidad de las Ciencias Informáticas (UCI), cuenta con el Sistema de Indicadores de Ciencia, Investigación y Tecnología encargado de evaluar el desarrollo científico de dicha entidad. El Sistema de Indicadores de Ciencia, Investigación y Tecnología posibilita calcular los indicadores necesarios para dar cumplimiento a las actividades científicas de la universidad, pero no permite hacer un seguimiento disgregado por las diferentes áreas de la Universidad, se reduce al control de las evidencias y cantidades,

teniendo como protagonista de la gestión de la información al Vicedecano de Investigación y Postgrado de cada facultad.

El CESIM, una vez que se recibe la planificación de los indicadores para el año siguiente, se elabora el plan para cada uno de los Departamentos Docentes y Productivos. Esta planificación se realiza entre los meses de diciembre y enero del año siguiente. La misma se realiza bajo la premisa de garantizar que sea posible cumplir los compromisos de la misma y a su vez de potenciar los resultados científicos y de investigación de forma confiable. Cada departamento una vez que tiene su plan, distribuye y organiza por cada uno de sus profesores, adiestrados y especialistas los compromisos de publicaciones, eventos en los que deberá participar, premios a obtener y aquellos que deben cambiar de categoría docente y científica. Cada una de estas planificaciones debe quedar reflejada en el plan de resultados anual del profesor, en el plan de adiestramiento o en el plan de resultados del especialista.

Actualmente la gestión de la planificación de los indicadores, su actualización y el control presenta deficiencias tales como, la realización de forma manual de los compromisos de cada uno de los profesores o investigadores, así como su relación con el plan de resultado y los compromisos con el departamento. Esto provoca que en ocasiones no se tenga un control exacto en los resultados del profesor o del investigador con relación a un evento próximo en cuanto al estado de sus publicaciones o al estado de su maestría, generando un alto nivel de desactualización de la información ya que para poder resumir los resultados de un departamento es necesario solicitar la información a todos sus profesores y si es a nivel de facultad hay que exigir a cada uno de los departamentos de la facultad la actualización. Luego de haber obtenido esta información el Jefe de Departamento procede a actualizar.

Como buena práctica en cada departamento se ha ido creando y organizando un activista de ciencia y técnica, sin embargo esta persona utiliza un documento Excel a través de cual registra todos los compromisos, pero cada uno de los investigadores está obligado a reportarle a esa persona la evidencia cada vez que obtiene una publicación. En muchas ocasiones el proceso de entrega de evidencia se demora y es lo que realmente respalda que el resultado se obtuvo, por tanto se mantiene también un alto nivel de desactualización de la información, además como se almacena en documentos digitales o impresos no exige una forma viable de hacérselo llegar al Vicedecanato de Investigación y Postgrado de manera constante.

En ocasiones cuando solicitan una actualización de los Indicadores de Ciencia y Técnica, el activista no tiene claridad de cuales evidencias ya entregó, viéndose en la obligación de llevar todas las del año a consecuencia de que los profesores no actualizan automáticamente. Agregado a esto los profesores no tienen una información actualizada sobre convocatorias a eventos, revistas especializadas, premios, cursos de postgrados, convocatorias a maestrías, convocatorias a doctorados, que le permitan tener una relación constante de los temas de ciencia y técnica.

Por otro lado en ocasiones se lanzan convocatorias de maestrías y doctorados que se pueden identificar posibles investigadores a partir de ese proceso pero no se tienen currículum vitae de esa persona. Entonces hay que pedirle a esa persona que actualice el currículum vitae y en muchas ocasiones este currículum no es posible verificarlo del todo porque no se tienen todas las evidencias del profesional involucrado.

Numerosos programas de maestría y doctorado obligan a certificar este currículum vitae o sea hacer una carta de anuencia en el que se diga que lo que plantea ese investigador es cierto pero como no se tienen todas las evidencias de esa persona entonces no se puede afirmar con seguridad que ese currículum vitae es del todo cierto. También existe desconocimiento acerca de la experticia de las personas en algunos temas pues en ocasiones se lanzan convocatorias para profesores para dar un postgrado especializado y entonces como no se tiene un resumen de cuáles son las personas que tienen cierta experticia porque trabajan en esos temas o han impartido postgrados similares o tienen temas de investigación similar no se puede gestionar de manera rápida quién es el que puede optar por ese programa.

Todas las problemáticas expuestas anteriormente afectan la integridad de los indicadores de ciencia y técnica que se manejan en el centro, definiendo la integridad como uno de los principios básicos dentro del proceso de aseguramiento de la información y se centra en mantener los datos tal cual fueron generados, sin ser manipulados o alterados por personas o procesos no autorizados.

Se define por tanto, como **Problema a resolver**: Durante la planificación y gestión de los resultados de ciencia y técnica en CESIM se manifiestan deficiencias, centradas en la integridad de la información.

El **Objeto de Estudio** de este trabajo está enmarcado en el proceso de la gestión de indicadores de ciencia y técnica, tomando como **Campo de acción**: La integridad de la información de la planificación y el control de resultados de los indicadores de ciencia y técnica en CESIM.

Para la solución del problema planteado se identifica como **Objetivo general**: Desarrollar un sistema informático para la gestión de los principales indicadores de ciencia y técnica y sus proyecciones en CESIM que permita elevar los niveles de integridad de la información que se procesa.

La **hipótesis** de trabajo se puede expresar entonces como: Con la utilización del sistema informático para la gestión de los indicadores de ciencia y técnica y sus proyecciones en el CESIM, se elevarán los niveles de la integridad de la información que se procesa.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **Tareas de la Investigación**:

- Describir las dificultades actuales para la gestión de los indicadores de ciencia y técnica y sus proyecciones en CESIM.
- Realizar un análisis de los sistemas informáticos de la gestión de Indicadores de Ciencia y Técnica a nivel nacional e internacional, estableciendo similitudes con la investigación en curso.
- Asimilar las tendencias, técnicas, tecnologías, plataformas, metodologías y herramientas que se usarán para el desarrollo de la aplicación.
- Analizar los procesos de negocio asociados a la gestión de la información relacionada con el proceso de planificación de los indicadores de ciencia y técnica, logrando un modelo único como guía para la implementación del sistema.
- Generar los artefactos correspondientes a las fases de desarrollo: “Modelamiento del Negocio”, “Requisitos”, “Análisis y Diseño” e “Implementación”, sirviendo de base a los desarrolladores.
- Implementar el sistema de planificación de los indicadores de ciencia y técnica siguiendo la arquitectura y las pautas de diseño definidas por el proyecto, así como lo establecido en el documento de Especificación de Requisitos de Software.

Para realizar la presente investigación se tuvieron en cuenta los siguientes métodos científicos:

Métodos teóricos:

- **Analítico - sintético**: A través del mismo se realiza una descripción de las principales metodologías, tecnologías y herramientas posibles a utilizar en el desarrollo de la investigación y se seleccionan las más convenientes, en dependencia de sus características.

- **Histórico - Lógico:** Permitió, a través del análisis de la evolución de los sistemas de gestión de la información, determinar las principales características que debe poseer el sistema que se describe en la presente investigación.
- **Modelación:** Permitió reflejar la estructura, relaciones internas y características de la solución a través de diagramas mediante el lenguaje de modelado *Unified Modeling Language (UML)*.

Métodos empíricos:

- **Entrevistas:** Se aplicó a los trabajadores de los vicedecanatos de Investigación y Postgrado de las diferentes facultades de la universidad para constatar las condiciones con que cuentan para la gestión de la información generada a partir de la actividad científica del centro. Lo que permitió realizar el levantamiento de requisitos, así como un mejor entendimiento de acerca del proceso de gestión de indicadores de ciencia y tecnologías y sus proyecciones en CESIM.
- **Observación:** Con este método fue posible recopilar una serie de datos confiables que ayudan a la comprensión del fenómeno y a la definición de lo primordial de la problemática. Al mismo tiempo esto posibilitó el planteamiento del problema permitiendo enmarcar el objeto de estudio y el campo de acción, lo cual propicia enfocar la investigación hacia lo que se necesita alcanzar y cómo alcanzarlo.

El presente documento se estructura en varias secciones que incluyen cuatro capítulos. Para lograr una mejor comprensión de estas secciones, a continuación se brinda la descripción capitular:

Capítulo 1: Se describen los conceptos fundamentales asociados al dominio del problema. Se expone el estado del arte del tema tratado, tanto a nivel nacional como internacional se hace un análisis entre las soluciones ya existentes. Además, se explican y justifican las tendencias, tecnologías y herramientas en las que se apoya la solución del problema.

Capítulo 2: Se describe el flujo actual de la gestión de la información llevada a cabo en el Vicedecanato de Investigación y Postgrado de la antigua Facultad 7. Se especifican las características que tendrá el sistema propuesto, lo que posibilitará un mayor entendimiento del mismo. Se detallan todas las funcionalidades a implementar. Además, se muestran otros artefactos propios de la metodología de desarrollo, que se generan en estas etapas.

Capítulo 3: Contiene lo referente al análisis y diseño del sistema. Como parte de la solución se modelan los diagramas de clases del análisis, los diagramas de clases del diseño, así como los diagramas

de interacción correspondientes. Se exponen los patrones de diseño empleados en la solución y se muestra la estructura del sistema a través de la arquitectura del mismo.

Capítulo 4: Aborda todo lo relacionado con el flujo de trabajo de implementación. En el mismo se especifican los componentes ejecutables del sistema, así como su interacción. Se exponen además los diagramas de implementación y despliegue; artefactos generados en esta fase de construcción del sistema.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Introducción

La complejidad de los sistemas de software en la actualidad así como los exigentes tiempos de desarrollo, crean día a día la necesidad de eficacia y aprovechamiento del tiempo en cada una de las tareas que se realizan durante el ciclo de vida del software. Arrojando como resultado el surgimiento de asistentes que ayudan a la optimización de estas tareas en el ámbito computacional.

Este capítulo ofrece una panorámica del resultado de las investigaciones realizadas, tomando como base el objeto de estudio y los sistemas informáticos existentes en el ámbito internacional, nacional y en la UCI, aplicados al área de la investigación. Detalla la metodología, lenguaje y herramientas utilizadas para el desarrollo de la solución.

1.1. Conceptos asociados al dominio del problema.

Para una mayor comprensión de los elementos que componen la propuesta del diseño de un sistema para la gestión de indicadores sobre el desarrollo científico y tecnológico de los profesores del CESIM, es preciso partir de un conjunto de definiciones que implicarán el sustento teórico de la presente investigación.

1.1.1. Indicadores.

Los indicadores son herramientas para clarificar y definir, de forma más precisa, objetivos e impactos. Son medidas verificables de cambios o resultados diseñadas para contar con un estándar contra el cual evaluar, estimar o demostrar el progreso con respecto a metas establecidas. (4) Según la definición de indicadores que dieran Mario Albornoz y Eduardo Martínez:

Los indicadores representan una medición agregada y compleja que permite describir o evaluar un fenómeno, su naturaleza, estado y evolución; articula o correlaciona variables y su unidad de medida es compuesta o relativa. Los indicadores suelen presentar las características siguientes: generalidad, correlación entre variables distintas o de distintos contextos, cuantificabilidad, temporalidad, y posibilidad de constituirse en componentes básicos de desarrollos teóricos. (5)

1.1.2. Indicadores de Ciencia, Tecnología e Investigación.

Mario Albornoz y Hernán Jaramillo expresan: “Es algo tan simple como un valor numérico que expresa un rasgo, el nivel de desarrollo de una dimensión del sistema de ciencia y tecnología de un país”.

La definición de indicadores de ciencia y tecnología realizada por Rémi Barré, en otra publicación: “Son los conocimientos cuantitativos sobre las actividades científicas, tecnológicas y de innovación útiles para establecer, ejecutar y seguir las políticas de investigación”. (6)

1.1.3. Sistemas de gestión de indicadores para medir la Ciencia, Tecnología e Investigación.

En el mundo existen disímiles sistemas de gestión de indicadores que permiten medir los resultados científicos e investigativos alcanzados por las diferentes instituciones, estos objetivos se logran a través de un conjunto de indicadores, que pueden variar en dependencia de las necesidades específicas que presente la empresa o el ámbito donde se aplique el mismo, para llevar un mayor control de las evaluaciones. A continuación se describen algunos de estos sistemas análogos afines, que son de gran ayuda para las diferentes instituciones.

1.1.4. Sistemas similares.

Sistema Integrado de Información sobre Investigación Científica y Tecnológica de México.

El Sistema Integrado de Información sobre Investigación Científica y Tecnológica de México (SIICYT), es un instrumento que refuerza la integración y solidez del Sistema Nacional de Ciencia y Tecnología. El SIICYT integra los esfuerzos de diferentes instituciones educativas, centros de investigación, organismos públicos, empresas y personas físicas y morales del sector público y privado. Este sistema es un instrumento para articular la información sobre las políticas, programas, áreas estratégicas, proyectos y participantes en el sistema nacional de ciencia-tecnología-empresa. Las empresas y dependencias comparten los resultados de apoyo a proyectos de investigación y desarrollo tecnológico sobre su estrategia de negocios, competitividad y crecimiento. (7)

Sistema Nacional de Ciencia, Tecnología e Innovación de Venezuela.

Es un sistema abierto del cual forman parte las políticas, estrategias, programas, metodologías y mecanismos para la gestión, promoción, financiación, protección y divulgación de la investigación científica y la innovación tecnológica, así como las organizaciones públicas, privadas o mixtas que realicen o promuevan el desarrollo de actividades científicas, tecnológicas y de innovación. (8)

Una revisión de los principales aspectos que conforman el SNCTI venezolano debe incluir condiciones macroeconómicas, marco legal y de incentivos, organismos públicos que administran y coordinan a las instituciones de ciencia y tecnología, capacidades en comunicaciones y tecnologías de información y en investigación y desarrollo, instituciones de formación de personal, principalmente universidades y postgrados, vinculaciones entre las instituciones académicas y científicas con el sector empresarial, servicios tecnológicos (metrología, normalización, información y asistencia técnica), régimen de protección a la propiedad industrial, sistemas de financiamiento y caracterización del sector empresarial. (9)

Departamento Administrativo de Ciencia, Tecnología e Innovación en Colombia.

Es el Departamento Administrativo de Ciencia, Tecnología e Innovación en Colombia (COLCIENCIA) y crea el Sistema Nacional de Ciencia, Tecnología e Innovación. COLCIENCIAS es el rector que formula, orienta, dirige, coordina, ejecuta e implementa la política del Estado en el sector de ciencia, tecnología e innovación. Busca llevar al país a niveles más elevados de competitividad mediante el desarrollo tecnológico-innovador y la integración de actividades de los diversos sectores para impulsar áreas del conocimiento estratégicas de clase mundial. Es una organización líder en generación de políticas y capacidades que permiten incorporar la Ciencia la Tecnología y la Innovación a la cultura del país y convierten el conocimiento en motor de desarrollo local, regional y nacional. (10)

Sistema de Indicadores de Ciencia, Tecnología e Innovación.

El Sistema de Indicadores de Ciencia, Tecnología e Innovación (CYTCES), se encuentra actualmente vigente en las instituciones y universidades del MES y el Ministerio de Ciencias, Tecnología y Medio Ambiente (CITMA) de Cuba. El mismo consta de una serie de indicadores ponderados. (11)

Relevancia: Son aquellos que miden premios y reconocimientos obtenidos como resultado de la investigación científica, otorgados por instituciones de prestigio nacional o internacional.

Visibilidad: Indicadores bibliométricos relacionados con publicaciones científicas. Incluye participación en congresos y conferencias científicas nacionales e internacionales.

Tecnología: Incluye patentes y registros como resultado del desarrollo tecnológico.

Pertinencia: Indicadores evaluativos del monto de recursos financieros ingresados por la universidad como resultado de la comercialización de productos de diferentes clases.

Impacto: Incluye la evaluación de los aportes económicos de los productos universitarios en la economía y en la sociedad cubana.

En Cuba, particularmente en la Universidad de las Ciencias Informáticas se creó desde sus inicios la Dirección de Investigaciones que trabaja tenazmente por la organización y el buen desempeño de la actividad científica de la institución. Sus principales esfuerzos están dirigidos a organizar las investigaciones en la Universidad estimulando la participación de profesores y estudiantes, a establecer alianza estratégica con el MES, el CITMA y el Polo Científico, e incorporar el centro al Sistema Científico Nacional.

La Dirección de Investigaciones que gestiona toda la información referente a las investigaciones, innovaciones y publicaciones que se llevan a cabo en la Universidad. El Sistema de Indicadores de Ciencia, Investigación y Tecnología (SIndiCIT) existente en la UCI, brinda la posibilidad de llevar un control de los indicadores relacionados con la ciencia, tecnología e innovación que se especifican hoy en día en cada una de las facultades que componen el centro, aunque también puede ser utilizado con el mismo propósito en otros centros. Es un sistema de indicadores que permite evaluar la producción científica de los profesores, investigadores y estudiantes de la Universidad, que potencia los resultados científicos y de innovación, que premia el trabajo en equipo y se adapta a las características de la universidad. Este sistema contribuye a ponderar de manera diferenciada aquellas investigaciones científicas de ciclo completo (Investigación + Desarrollo + Producción + Comercialización) con relación a aquellas investigaciones puramente académicas.

Es significativo señalar que el mismo fue creado para ser usado por personas con determinados niveles de acceso, principalmente por el personal de la dirección de investigaciones de la UCI. Si se desea registrar los datos, se hace necesario estar registrado en la aplicación y tener los permisos sobre la facultad en la que se desea trabajar. Siendo el administrador del sistema el encargado de conceder los permisos, constituyendo esto una política de seguridad beneficiosa para el sistema.

1.1.5. Valoración del estudio realizado.

Con el estudio realizado sobre los sistemas de ciencia, tecnología e investigación existentes a nivel internacional se llegó a la conclusión de que no existe ninguno que cumpla con las características que requiere CESIM. El sistema debe estar básicamente guiado por un conjunto de indicadores generales, que mediante los criterios de calificación permitan medir la actividad científica en dicho centro así como las

diferentes actividades en las que participan los profesores. A continuación se muestran los indicadores generales de medición de la actividad de ciencia y técnica, necesarios para la implementación del sistema.

- Premios.
- Publicaciones.
- Trabajos en Eventos.
- Maestrías/Doctorados.
- Categoría Docente.

En cuanto a los sistemas de ciencia, tecnología e investigación existentes a nivel nacional se puede decir que están creados para llevar el control de las diferentes actividades que realiza cada profesor o estudiante en el centro donde se encuentre y los indicadores generales que se evalúan en el SIndiCIT son similares a los que presenta el sistema para medir el desarrollo científico de cada profesional. Este sistema está centrado en la gestión del resultado de indicador, no así en el del investigador. El mismo está diseñado para introducir los datos luego de haberse obtenido los resultados, además de realizar una evaluación de forma general, y no específica por cada área del CESIM la participación de cada profesional. El sistema no permite que el investigador interactúe con el mismo dado que el Vicedecano de investigación y postgrado es el principal protagonista. Por tales características no cumple con las expectativas del Centro Informática Médica.

1.2. Metodología, lenguajes y herramientas de modelado a utilizar para el desarrollo.

1.2.1. Metodologías RUP.

El proceso unificado de desarrollo *Rational Unified Process* (RUP por sus siglas en inglés) es una metodología para la ingeniería de *software* que va más allá del mero análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de *software*.

El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. Incluye artefactos y roles. (12)

RUP brinda importantes beneficios que fomentan su usabilidad y su potencial de desarrollo:

- Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del *software* mediante una gestión sistemática de los riesgos. (12)
- Enriquece la productividad en equipo y proporciona prácticas óptimas de *software* a todos sus miembros. (13)
- Proporciona guías explícitas para áreas tales como modelado de negocios, arquitectura Web, pruebas y calidad.
- Unifica todo el equipo de desarrollo de *software* y mejora la comunicación, al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado y un punto de vista de cómo desarrollar un *software*.
- No solo garantiza que los proyectos abordados serán ejecutados íntegramente sino que además evita desviaciones importantes respecto a los plazos. (14)

1.2.2. Lenguaje Unificado de Modelado UML.

El lenguaje de modelado utilizado en el desarrollo de la aplicación fue UML 2.0 (*Unified Modeling Language*) que permite especificar, visualizar y documentar los artefactos que son generados a través de todo el ciclo de desarrollo del *software*, fundamentalmente en las fases de análisis, diseño e implementación.

UML está construido sobre los principales conceptos de la orientación a objetos (OO) y es por naturaleza el más utilizado para lenguajes orientados a objetos como C++, Java y C#. Es independiente de la metodología que se desee utilizar, lo que ha provocado sin dudas el amplio uso que tiene en la industria de desarrollo de *software*. (15)

1.2.3. Notación para el Modelado de Procesos de Negocio: BPMN.

BPMN (*Business Process Modeling Notation*) es la notación para el modelado de procesos de negocio por excelencia. Provee una notación fácilmente entendible por todos los usuarios de negocio. Es independiente de cualquier metodología de modelado de procesos. Está diseñado para cubrir varios tipos de modelado y permite la creación tanto de segmentos de proceso como de procesos de negocio de comienzo a fin, y todo ello en diferentes niveles de representatividad. Brinda una técnica de modelamiento de flujos natural y consistente con la manera de pensar y actuar de los analistas de negocios. (16) Es mantenida por *Object Management Group (OMG)*, la misma compañía encargada de UML.

1.2.4. Modelo Integral de Madurez y Capacidad.

Modelo Integral de Madurez y Capacidad (CMMI por sus siglas en inglés) es un marco de trabajo de los elementos principales del proceso para el desarrollo de sistemas. Brinda una vista integradora de mejora de procesos a través de múltiples disciplinas, establece objetivos y prioridades al proceso de mejora, provee orientación para la calidad de procesos y brinda una norma para evaluar las prácticas actuales.

CMMI posee dos enfoques, siendo estos la representación continua y la escalonada. La representación continua brinda un método flexible para la mejora de procesos. Con este, una empresa puede elegir mejorar el rendimiento de un problema único de un proceso, trabajar en varias áreas que están fuertemente alineadas con los objetivos del negocio de la organización y mejorar diferentes procesos hasta varios niveles.

La representación escalonada brinda un método sistemático y estructurado para el proceso de mejoras. La obtención de cada nivel asegura que una mejora adecuada ha establecido una base para el próximo, estableciéndose de esta manera una mejora incremental. (17)

1.2.5. Ciclo de vida del proyecto.

La representación escalonada del CMMI está caracterizada por cinco niveles, dentro de esto la UCI trabaja específicamente para certificar el nivel 2, el cual comprende la administración básica del proyecto.

Dentro del nivel 2 existen siete áreas de procesos, las cuales son: Aseguramiento de la Calidad del Producto y el Proceso (PPQA), Planificación de Proyectos (PP), Seguimiento y Control del Proyecto (PMC), Gestión de Acuerdos con Proveedores (SAM), Gestión de la Configuración (CM), Medición y Análisis (MA) y Administración de Requisitos (REQM). Este último tiene como objetivo administrar los requerimientos de los productos del proyecto y los componentes del producto e identificar inconsistencias entre estos requerimientos y los planes y productos de trabajo del proyecto.

En la universidad ha quedado definido el Libro de Proceso para la Administración de Requisitos: IPP-3510_2009, donde se realiza una descripción del ciclo de vida de desarrollo del proyecto, el cual está dividido en nueve fases. Las actividades, los roles y los artefactos para cumplir con las prácticas y sub-prácticas de REQM son también descritas en este libro, siguiendo como guía RUP. De esta manera queda especificado UML como lenguaje de modelado para las fases de desarrollo del proyecto, sin incluir el modelado de procesos en la que se emplea BPMN.

1.2.6. Herramienta CASE para el modelado.

Enterprise Architect 7.5

Para realizar el modelado se utilizará *Enterprise Architect* (EA, por sus siglas en inglés) en su versión 7.5. EA es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de *software* desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. Además, es multi-usuario, basada en *Windows*, diseñada para ayudar a construir *software* robusto y fácil de mantener.

Esta herramienta ayuda a manejar grandes modelos y complejas informaciones, permite compartir modelos o aspectos específicos de estos, proporciona una poderosa generación y reporte de documentos con editor de plantillas WYSIWYG para Formato de Texto Enriquecido (RTF por sus siglas en inglés) o HTML soporta la ingeniería inversa de código fuente para lenguajes como C++, C#, Java, Delphi, VB.Net, Visual Basic, ActionScript, Python and PHP; así como para sistemas DBMS como Oracle 9i, 10g o 11g; SQL Server; MySQL; Access 2007 y PostgreSQL. (18)

1.3. Entorno de desarrollo integrado.

Visual Studio 2013

Es un conjunto de herramientas de desarrollo basadas en componentes y otras tecnologías para la construcción de potentes aplicaciones de alto rendimiento. Visual Studio está optimizado para el diseño basado en el equipo, el desarrollo y despliegue de soluciones empresariales. Soporta varios lenguajes de programación tales como: *Visual C++*, *Visual C#*, *Visual J#* y *Visual Basic.NET* y permite crear aplicaciones, servicios web, sitios y aplicaciones web en cualquier entorno que posea la plataforma .NET a partir de la versión del 2002. (19)

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles, además soporta tecnologías aprovechando las ventajas de los nuevos sistemas operativos, entre ellas se encuentran: *Windows Communication Foundation (WCF)* y *Windows Presentation Foundation (WPF)*. También puede crear aplicaciones web con interfaces de usuario de próxima generación y con componentes de cliente reutilizables que utilicen las nuevas características de Visual Studio. (20)

1.4. Tecnologías y lenguajes de programación a utilizar.

Asp.NET MVC 4

Es un lenguaje totalmente orientado a objeto que permite a los programadores crear páginas web dinámicas, aplicaciones web y servicios web XML. Siendo una tecnología gratuita. Este *Framework* está construido sobre el *Common Language Runtime*, lo que brinda la posibilidad de escribir código ASP.NET usando cualquier lenguaje admitido por el *.NET Framework*.

ASP.NET ofrece diversas mejoras como (21):

- Servicios web: brinda herramientas para compartir datos en información entre diferentes sitios.
- Seguridad: cuenta con diferentes herramientas que garantizan la seguridad de nuestras aplicaciones.
- Rendimiento: permite el almacenamiento del cache en el servidor. Además la aplicación se compila una sola vez al lenguaje nativo, y luego en cada petición tiene una compilación *Just In Time*, es decir, se compila desde el código nativo, lo que ha llevado a un mejor rendimiento.
- Rapidez en la programación: se puede mostrar toda una base de datos y hacer rutinas complejas en cuestión de pocos minutos y escasas líneas a través de diversos controles.

C# 4.0

Es un lenguaje de programación multi-paradigma, inflexible, imperativo, declarativo, funcional, genérico, orientado a objetos y a componentes ubicados en las disciplinas de programación. Fue desarrollado por *Microsoft* dentro de su plataforma *.NET* y posteriormente aprobado como un estándar por el ECMA (ECMA-334) y la ISO (ISO / IEC 23270:2006). Está diseñado para la infraestructura del lenguaje común.

C # tiene la intención de ser simple, moderno y de propósito general, por lo que se propone como lenguaje de programación a utilizar. (22)

NHibernate 3.1

NHibernate, es la conversión de Hibernate del lenguaje Java a C# para su integración en la plataforma *.NET*. Al igual que muchas otras herramientas de esta plataforma, es compatible con el *Framework .NET 4.5*.

Como todas las herramientas de su tipo, busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información le permite a la aplicación manipular los datos almacenados en la base de datos operando sobre objetos, con todas las características de la Programación Orientada a Objetos (OOP).

Este *framework* soporta los más habituales motores de base de datos existentes en el mercado: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otros. Una ventaja es la total independencia con el motor de base de datos como se tiene totalmente independiente la capa de datos con la capa de lógica de negocio. Por tanto, bastará con cambiar una línea de un fichero de conexión y se tendrá la aplicación rodando bajo otro motor de base de datos de los que soporta. Esta herramienta brinda facilidad de programación, ya que la orientación a objetos facilita pasar de un diagrama a código fuente.

Utiliza el lenguaje de consulta Hibernate (HQL, por sus siglas en inglés) como lenguaje de consultas orientado a objetos. No produce generación de código extra en el procedimiento de construcción. Se ocupa de los dos lados del problema; no solo cómo llegan los objetos a la base de datos, sino cómo extraerlos nuevamente. Permite especificar exactamente el SQL que NHibernate usará para persistir los objetos. (23)

1.5. Sistema de Gestor de Bases de Datos.

PostgreSQL 9.1.1

Sistema de gestión de base de datos relacional orientado a objetos y libre, desarrollado en la Universidad de California en *Berkeley* Departamento de Ciencias de la Computación y con un código fuente disponible libremente. Es un potente motor de Base de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de base de datos comerciales.

Posees características tales como:

- Incluye herencia entre tablas.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes.
- Permite la declaración de funciones propias, así como la definición de disparadores.

- Soporta el uso de índices, reglas y vistas.
- Incorpora una estructura de datos *array*.

Debido a la licencia liberal, PostgreSQL puede ser utilizado, modificado y distribuido por cualquier persona libre de forma gratuita para cualquier propósito, ya sea privado, comercial o académico.

Conclusiones parciales

En el capítulo se realizó el análisis del estado del arte y de las tendencias actuales necesarias para facilitar la comprensión de la propuesta de sistemas para la gestión de indicadores, se profundizó en las características de las aplicaciones a nivel nacional e internacional. De lo que se obtuvo que ninguna cumple con las características de una aplicación centrada en la planificación de los resultados que debe obtener el investigador. Se evidencia la inconformidad del Vicedecanato de investigación y postgrado de la anterior Facultad 7, en cuanto al control de manera centralizada del proceso de gestión de indicadores. Se seleccionó dentro de un grupo de tecnologías, herramientas y metodologías las adecuadas para realizar el *software* que se requiere: ASP.NET como plataforma de desarrollo, *Enterprise Architect* como herramienta CASE y RUP como metodología de desarrollo las cuales cumplen con las herramientas a utilizar por el departamento.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

En este capítulo se realiza la descripción del negocio para la gestión de los indicadores sobre el desarrollo científico y tecnológico que tiene lugar en la antigua Facultad 7, consiguiendo una mayor comprensión del mismo y sus características. Se modelara el negocio identificando las actividades correspondientes a cada proceso identificando los requisitos funcionales y no funcionales con los que cuenta el sistema así como los trabajadores del negocio. Se presentan algunas planillas necesarias para poder desarrollar con la metodología seleccionada.

2.1. Descripción del negocio actual.

La Dirección de Investigaciones mide los resultados científicos de la universidad, mediante un conjunto de indicadores. Para conocer el comportamiento de estos en cada facultad, los Vicedecanos elaboran un Excel con dicha información y lo envían a la Dirección de Investigaciones unido a los documentos que avalan dichos resultados. Los indicadores de medición de la actividad de Ciencia y Técnica para los centros de Educación Superior (adaptados a la UCI) son:

- Premios obtenidos.
- Publicaciones científicas.
- Patentes y registros.
- Participación en proyectos.
- Resultados introducidos.
- Trabajos presentados en eventos.

En la antigua Facultad 7 este proceso se realizaba de la siguiente forma:

Después de ser enviado a la antigua Facultad 7 el plan de actividades que deben realizar los profesionales anualmente, el vicedecano de investigación y postgrado se encarga de repartir las actividades de forma equilibrada para cada uno de los jefes de las diferentes áreas ya sea un jefe de departamento docente o un jefe de centro productivo. Los departamentos docentes son: Ciencias Básicas, Ciencias Sociales, Técnicas de programación, Ingeniería de *software* y Gestión tecnológica y en los centros de producción: Desarrollo de componentes, Desarrollo de aplicaciones y Práctica Profesional.

En cada uno de los departamentos se encuentra un activista de investigación y postgrado el cual es el encargado de entregar a cada jefe de área la cantidad de actividades que debe realizar el profesor en el año. En el caso de los profesores docentes es igual, se reparten las actividades a los jefes de áreas y ellos mandan a los profesores las diferentes tareas que deben cumplir. Todos los profesionales no pueden participar en las mismas actividades, ya que esto está en dependencia de la categoría docente que presente el profesor:

- Profesor auxiliar.
- Profesor titular.
- Asistente.
- Instructor.

Después que el profesor participa en las tareas correspondientes deben entregar al jefe de área o a la activista de investigación y postgrado un certificado como constancia de participación. En el caso de los profesores docentes la entrega del certificado es al jefe de área y éste se lo envía al vicedecano de investigación y postgrado, que es el encargado de hacer el balance de ciencia y técnica de la facultad. Este balance es el que se entrega a la Dirección de Investigaciones de la Universidad, que es la encargada de medir los resultados científicos de la misma mediante un conjunto de indicadores y realizar el balance a nivel de universidad.

2.1.1. Objetos a informatizar.

Para llevar a cabo la gestión de indicadores de ciencia y técnica de CESIM existen varios procesos que deben ser informatizados, pues el manejo de estos datos de forma manual o vía correo resulta engorroso y muchas veces complicado. Esto trae consigo una serie de problemas que atentan contra la organización de los mismos. Serán objetos de informatización los procesos vinculados con la publicación de artículos de interés para los profesores, la gestión de documentos útiles como trabajos realizados y de consultas, la gestión de eventos, así como la actualización y la generación de reportes estadísticos sobre la labor investigativa, siendo esta última de vital importancia en la confección del balance general.

2.2. Trabajadores del negocio.

Los trabajadores del negocio son directivos de las diferentes facultades de la universidad como el Vicedecano de Investigación y Postgrado (VDIP). Encargado de solicitar información sobre el desarrollo profesional de los profesores y el progreso del Balance de Ciencia-Técnica (BCT). Así como analizar cada uno de los procesos referentes a las actividades y superación de los profesionales, siendo estos los máximos categóricos de cada una de las solicitudes efectuadas por los profesores.

Jefe de Departamento: es el encargado de gestionar toda la información investigativa de cada uno de los profesores pertenecientes al departamento.

Investigador: es el encargado de gestionar su plan de resultados y verificar la participación en los diferentes indicadores.

2.3. Proceso de Negocio.

El modelo del negocio describe el funcionamiento del negocio de una organización. Un proceso de negocio es una colección de actividades estructurales que en conjunto cumplen un objetivo determinado del negocio. Son la base para comprender mejor la forma en que opera un negocio en sus diferentes áreas y son una herramienta fundamental para acceder a modelos de calidad y eficiencia. (24)

A partir de la investigación realizada se identificó el proceso de Balance de Ciencia y técnica referente a las actividades realizada por los investigadores en el CESIM.

2.3.1. Realizar Balance de Ciencia y Técnica.

Antes de realizar el Balance de Ciencia y Técnica del CESIM, el VDIP se reúne con sus jefes de departamentos donde analiza el cumplimiento de la planificación de las actividades para cada uno de las diferentes áreas.

En el [Anexo 1](#) se encuentra la descripción del proceso y la Figura 1 muestra el flujo de actividades del mismo.

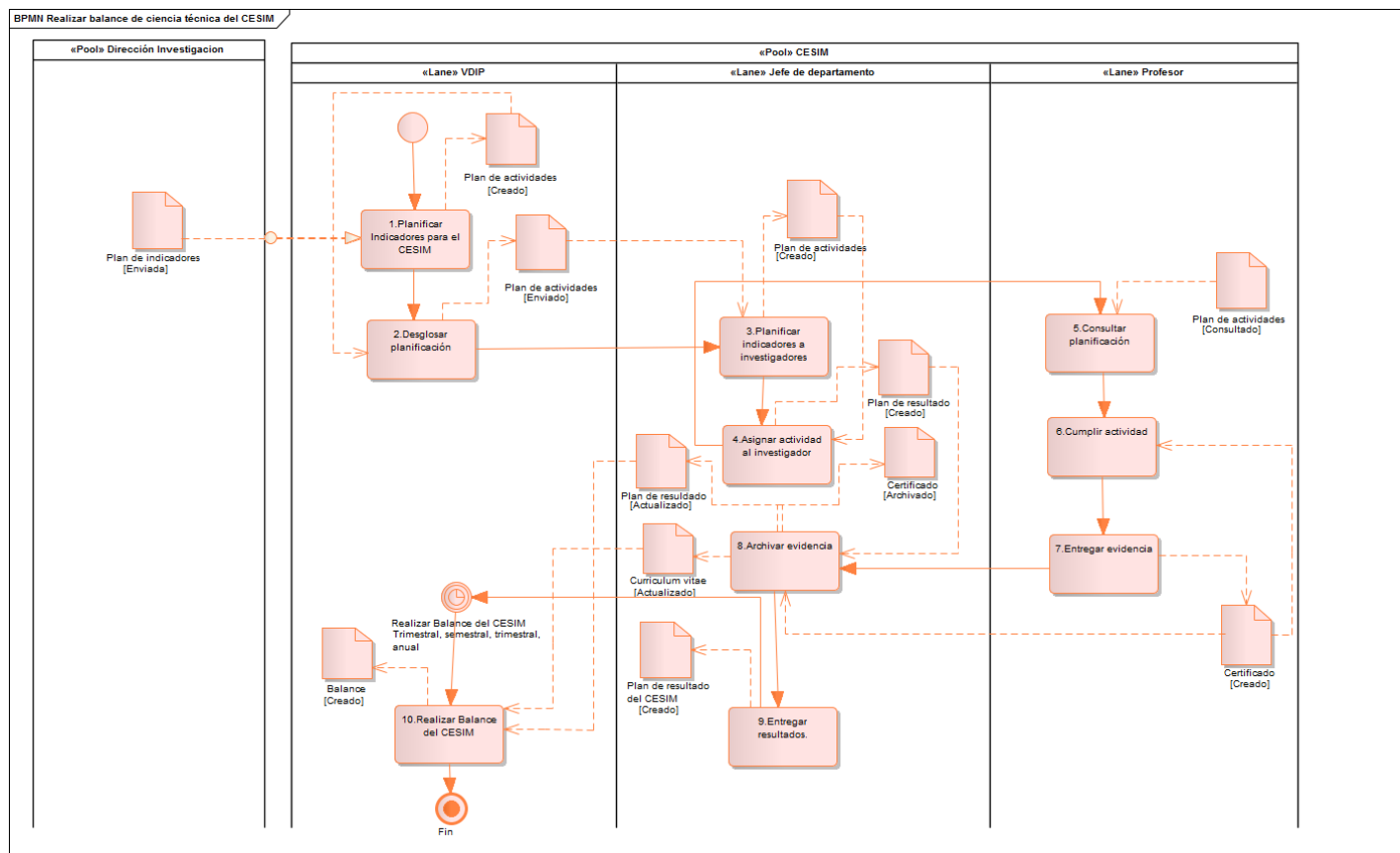


Figura 1 Realizar balance de ciencia y técnica.

2.4. Solución propuesta.

Para darle solución a la situación problemática, se propone desarrollar un sistema que logre la centralización de la información que se maneja y de esta forma mejorar la calidad con que se mide el desarrollo científico en el CESIM y la realización del balance anual. Permitirá llevar de forma constante un control sobre todos los datos competentes a la superación profesional de todo el personal, permitiéndole a todo jefe de departamento y vicedecano actualizar la información de un determinado profesor, ya sea eliminarlos, modificarlos como adicionarlos al sistema y a su vez generar un ranking por área según las políticas de superación del CESIM. Se dispondrá de publicaciones de artículos interesantes para la comunidad de profesores así como las convocatorias a eventos. El sistema permitirá además la gestión de documentos útiles, la actualización sobre las actividades de ciencia y técnicas desarrolladas por cada uno de los profesores, la revisión de trabajos realizados, así como la generación de reportes sobre la labor investigativa

por cada profesor, por cada departamento y del CESIM en general. Se habilitarán tres tipos de usuarios con roles diferentes los cuales van a ser los encargados de formalizar toda la información referente al sistema, limitar el acceso, dar soporte, mantenimiento e interactuar con el sistema.

2.5. Especificación de los requisitos de software.

2.5.1. Requerimientos funcionales.

Los requerimientos funcionales son servicios que el sistema debe proporcionar. Determinan cómo el sistema debería reaccionar a una determinada entrada y cómo debe comportarse en situaciones particulares. (25)

A partir del análisis de los procesos del negocio se definen las actividades que se deben automatizar, las cuales constituyen la base para identificar los requisitos funcionales del sistema.

La tabla 1 muestra los requisitos funcionales para el sistema gestión de indicadores de ciencia y técnica.

Requerimiento	Descripción
RF1.1_Añadir investigador	El sistema permite añadir un investigador al sistema desde ldap.
RF1.2_Modificar investigador	Permite modificar los privilegios referentes a un investigador en el sistema.
RF1.3_Eliminar investigador	Permite borrar los datos referentes a los investigadores.
RF1.4_Ver datos de los investigadores	Permite listar los datos de los trabajadores.
RF2.1_Autenticar investigador	El investigador puede autenticarse para tener acceso a la información en la aplicación.
RF3.1_Añadir premios	Permite registrar un premio por el cual el investigador o equipo puede optar.
RF3.2_Asignar premios	El sistema brinda la posibilidad de asignar premios por la participación en un determinado evento.

RF3.3_Modificar premios	Permite que los datos referentes a un premio puedan ser modificados en cuanto al evento, categoría y el año para cada una de las diferentes áreas.
RF3.4_Eliminar premios	Permite borrar toda la información referente a un premio.
RF3.5_Ver Premios	Permite listar los premios obtenido por cada una de las diferentes áreas.
RF4.1_Adicionar publicación	Permite adicionar una nueva publicación al sistema.
RF4.2_Asignar publicación	Permite asignar una publicación a cada uno de los investigadores.
RF4.3_Modificar publicación	Permite modificar los datos referentes a una publicación como el estado en que se encuentre (planificada, publicada, aceptada rechazada o en revisión).
RF4.4_Eliminar publicación	Permite borrar todos los datos referentes a una publicación.
RF4.5_Ver publicación	Permite listar todas las publicaciones realizadas por los diferentes investigadores en cada una de las diferentes áreas.
RF5.1_Adicionar evento	Permite registrar un evento en el cual el investigador o equipo deberá participar.
RF5.2_Asignar evento	Permite asignar eventos a cada uno de los investigadores de las diferentes áreas.
RF5.3_Modificar evento	Permite cambiar la información de un determinado evento
RF5.4_Eliminar evento	Permite Borrar toda la información de un evento.
RF5.5_Ver evento	Permite listar la cantidad de eventos.
RF6.1_Adicionar revista	Permite adicionar los datos de una revista en la cual el investigador ha realizado una publicación.
RF6.2_Modificar datos de la revista	Permite modificar los datos que fueron insertados con

	anterioridad
RF6.3_Eliminar revista	Permite eliminar la revista si no está usándose en una publicación.
RF6.4_Ver revistas	Permite mostrar la revista en la que un investigador ha realizado publicaciones.
RF7.1_Adicionar maestría	Permite adicionar una maestría por el cual el investigador podrá optar.
RF7.2_Asignar maestría	Permite asignar una determinada maestría al investigador que cumpla los requisitos para la realización de la misma.
RF7.3_Modificar maestría	Permite cambiar la información de una determinada maestría.
RF7.4_Eliminar maestría	Permite borrar los datos referentes a una maestría.
RF7.5_Ver maestría	Permite listar las maestrías por las que podrán optar los investigadores.
RF8.1_Adicionar doctorado	Permite adicionar un doctorado por el cual el investigador podrá optar
RF8.2_Asignar doctorado	Permite asignar un determinado doctorado al investigador que cumpla los requisitos para la realización del mismo.
RF8.3_Modificar doctorado	Permite modificar cualquiera de los datos que referentes a un doctorado.
RF8.4_Ver estado del doctorado	Permite ver al investigador si la solicitud del doctorado fue aceptada o rechazada.
RF9.1_Adicionar área	Permite adicionar una nueva área al sistema.
RF9.2_Modificar área	Brinda la posibilidad de poder modificar cualquier dato referente a un área determinada.
RF9.3_Eliminar área	Permite borrar toda la información almacenada en el sistema de

	una determinada área
RF9.4_Ver área	Permite borrar toda la información almacenada en el sistema de una determinada área.
RF10.1_Añadir plan anual por área	Permite añadir el plan de actividades que debe cumplir cada una de las diferentes áreas del CESIM. El plan de actividades no es más que los eventos, publicaciones, premios obtenidos así como la categoría docente de cada investigador.
RF10.2_Modificar plan anual por área	Permite hacer cambios en el plan de actividades asignado a cada una de las diferentes áreas.
RF10.3_Ver plan anual por área	Permite mostrar el plan anual de cada una de las diferentes áreas planificado con el fin de lograr los compromisos de CESIM.
RF10.4_Eliminar plan anual por área	Permite eliminar el plan anual del año anterior si así lo desea el investigador.
RF11.1_Incluir plan anual del CESIM	Permite añadir el plan anual para el CESIM que no es más que la cantidad de publicaciones, eventos, y demás indicadores de Ciencia, Tecnología e Investigación con que debe cumplir cada facultad. También entran aquí la cantidad de años, o tiempo al que debe hacer el cambio de categoría el profesor.
RF11.2_Modificar plan anual del CESIM	Permite realizarle cambios a al plan del CESIM para realizar la planificación de las actividades a cada uno de sus investigadores.
RF11.3_Eliminar plan anual del CESIM	Permite eliminar un plan que ya este obsoletos.
RF11.4_Ver plan anual del CESIM	Permite a los investigadores del sistema ver cuál es el plan del CESIM en el año.

RF12.1_Ver plan de resultados del profesor	Permite mostrar el plan de resultados de un determinado profesor.
RF13.1_Ver currículum Vitae	Permite ver el currículum Vitae de un investigador.
RF14.1_Adicionar categoría docente	Permite adicionar una categoría la cual será asignada a un investigador determinado del sistema.
RF14.2_Modificar categoría docente	Permite modificar la categoría docente, dado que el investigador puede cambiar de categoría.
RF14.3_Ver categoría docente	Permite ver la categoría docente que tiene cada investigador en el sistema.
RF15.1_Adicionar Registro	Permite adicionar los datos referentes a un registro.
RF15.2_Modificar Registro	Permite modificar en dependencia de si es un producto o un servicio los datos del registro.
RF15.3_Ver registro	Permite listar los registros que se encuentran en el sistema.
RF16.1_Asignar Departamento a un Centro.	Permite vincular un departamento a un determinado centro.
RF17.1_Realizar balance por área	Permite generar el balance general del desarrollo científico y tecnológico por cada una de las diferentes áreas del CESIM.
RF18.1_Realizar balance del CESIM.	Permite generar el balance general del desarrollo científico y tecnológico del CESIM.

Tabla 1 Requisitos Funcionales.

La Figura 2 muestra los requisitos funcionales para el sistema de gestión de indicadores de ciencia y técnica.

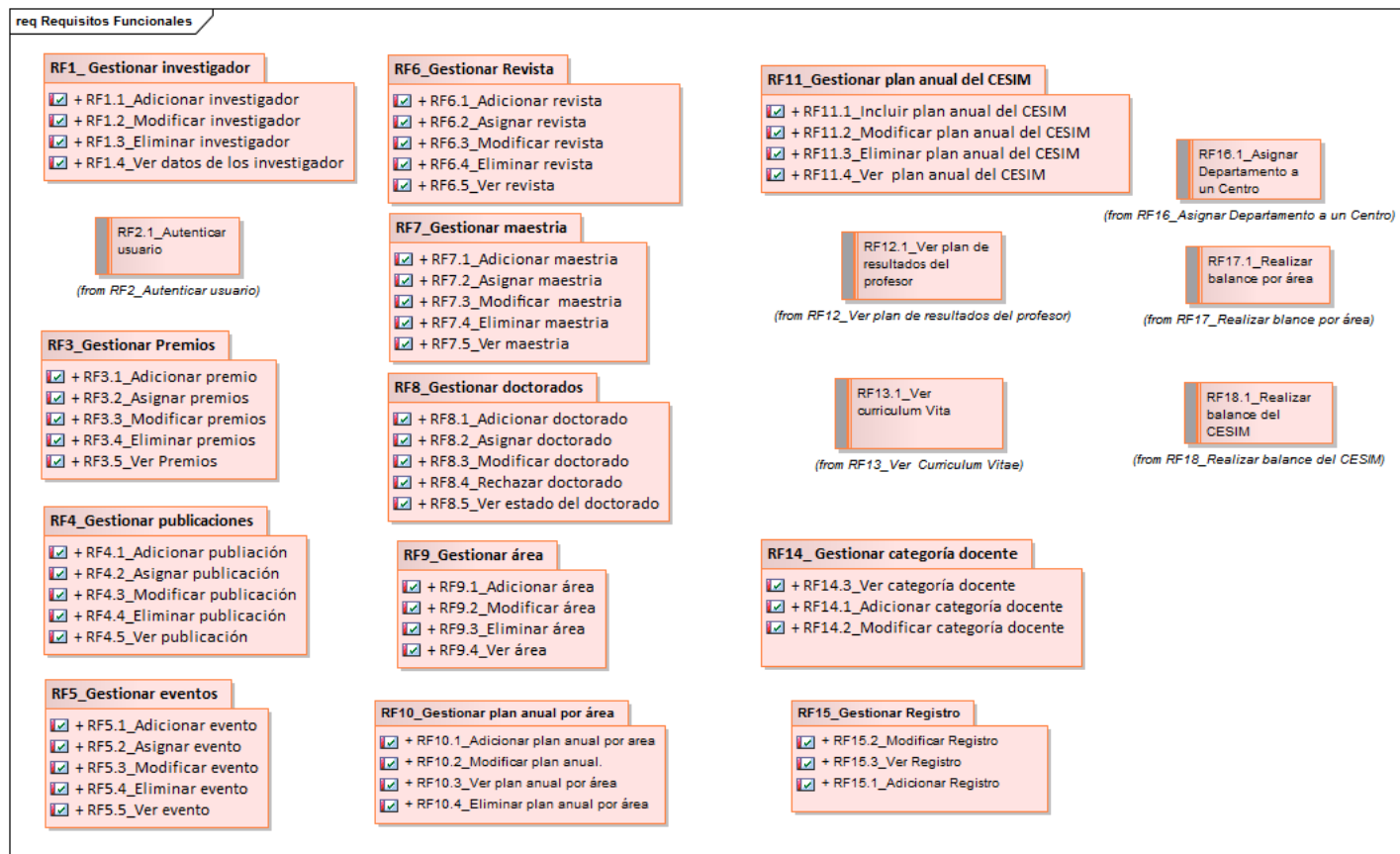


Figura 2 Diagrama de requisitos funcionales.

2.5.2. Requerimientos no funcionales.

Los requisitos no funcionales definen propiedades y restricciones del sistema. Estos requisitos pueden ser más críticos que los requisitos funcionales, ya que son normalmente a los que debe apuntar la arquitectura y si estos no son cumplidos, el *software* puede no funcionar o el cliente simplemente no acepta el producto. (25)

Los requisitos no funcionales fueron nombrados asignándoles un prefijo, de acuerdo a las Pautas para la Fase de Requerimientos del Departamento SWMI. En el [Anexo 3](#) se muestra la tabla de los prefijos según la categoría del requisito.

La Tabla 2 muestra los requisitos funcionales del sistema gestión de indicadores de ciencia y técnica.

Requerimiento	Descripción
RNU 1. Facilidad de empleo para investigador sin experiencia.	El sistema debe tener una interfaz de fácil aprendizaje para que investigadores inexpertos puedan familiarizarse con él.
RNU 2. Los investigadores deberán autenticarse para realizar cualquier operación en el sistema.	Deben de autenticarse los investigadores para que cada rol se comporte y trabaje solo con los permisos que se le asignan. Esta autenticación se realizará a través del dominio uci.
RNU 3. El sistema debe ser confiable y robusto dando muestra de seguridad en la interacción con los investigadores	Brinda la posibilidad que la información que se maneja sea confiable.
RNU 4 Los datos almacenados serán guardados de la manera correcta sin correr riesgos a que los mismos puedan desaparecer	En caso de que falle el sistema, no se pierden los datos almacenados ya que no le ocasionará problemas al Gestor de Base de Datos, además se realizarán salvas periódicas en la BD por si hay alguna falla en esta.
RNU 5 El sistema debe responder las peticiones del investigador en un tiempo razonable, a pesar de que va a interactuar con la base de datos.	El sistema no presentara demoras para responder a las peticiones de los investigadores.
RNDI 1. Uso de <i>Framework.Net 4.5</i> .	Se especifica el uso de <i>Microsoft Framework.Net 4.5</i> que ofrece mejoras en cuanto a administración y rendimiento. El lenguaje de programación C# depende de este <i>Framework</i> .
RNDI 2. C# como lenguaje de programación.	Se deberá utilizar C# como lenguaje de programación pues está diseñado y optimizado para la plataforma .NET.

RNDI 3. Entorno Integrado de Desarrollo <i>Microsoft Visual Studio 2013</i> .	Se utilizará <i>Microsoft Visual Studio 2013</i> como entorno integrado de desarrollo. Este IDE utiliza como marco de trabajo el <i>Framework.Net 4.5</i> y soporta C# como lenguaje de programación.
RNDI 5 Librerías a utilizar.	Se utiliza la librería <i>iTextSharp</i> para generar los reportes en formato DOC.
RNDI 6. Estándar de codificación.	Se utilizará el estándar con notación <i>CamelCase</i> (atributos), <i>PascalCase</i> (métodos) y Notación Húngara (interfaces).
RNDI 7. Uso de ASP.NET4.5 para el diseño de la interfaz de investigador.	La interfaz de investigador será diseñada utilizando la tecnología ASP.NET4.5.
RNDI 8 Uso de <i>Enterprise Architect</i> como herramienta CASE.	Se utilizará como herramienta CASE <i>Enterprise Architect</i> en su versión 7.5.
RNDI 9 Uso de UML como lenguaje de modelado.	Se utilizará como lenguaje de modelado UML en su versión 2.1.
RNDI 10 Patrones a utilizar para el diseño del sistema.	Los patrones a utilizar para el diseño son repositorio, y patrones <i>GRASP</i> tales como: Experto, Creador, Controlador, Unidirección, <i>Singleton</i> .
RNF 1. Disponibilidad del sistema siempre.	El sistema debe estar disponible siempre.
RNF 2. Exactitud en las salidas del sistema.	El sistema debe brindar salidas precisas.
RNE 1. Tiempo de respuesta al generar un reporte será el menor	Ante la solicitud de generar un reporte de complejidad media o simple, el tiempo de respuesta podrá ser de tres

posible en dependencia de las características del servidor.	segundo en dependencia de las características del servidor. En caso de que sea un reporte masivo, que requiera una gran cantidad de información, el tiempo de respuesta será mayor.
RNFO 1. Memoria RAM de 512 MB o superior.	Para el correcto funcionamiento del sistema, Este debe ser instalado sobre un ordenador con un CPU <i>Dual Pentium IV</i> 3.0GHz y 512 Mb de memoria RAM o superior.
RNFO 2. CPU <i>Dual Pentium IV</i> 3.0GHz.	
RNFO 3. <i>Gigabit Ethernet NIC</i> .	
RNFO 4. Sistema operativo <i>Windows XP Service Pack 3</i> o superior.	El <i>software</i> debe instalarse sobre un servidor <i>con Windows XP Service Pack 3</i> o superior como sistema operativo.
RNFO 5 Capacidad de disco duro 20 GB como mínimo.	La capacidad del disco duro debe ser como mínimo de 20 GB.
RNIU 1 Deben prevalecer colores verdes, para su identificación con la línea de desarrollo.	Deben prevalecer colores verdes, para la identificación del sistema con la línea de desarrollo.

Tabla 2. Requisitos no funcionales.

2.6. Definición de los actores del sistema.

Luego de describir los procesos del negocio y los requisitos funcionales y no funcionales de la gestión de indicadores para el balance de ciencia y técnica del CESIM, se define los actores que interactúan con los casos de uso del sistema, el mismo se muestra en la siguiente tabla.

Actor	Justificación
Planificador (VDIP)	Realiza la gestión de contenido del sistema y permisos de usuarios. Cuenta con posibilidades ilimitadas para ejecutar

	todas las funciones administrativas del sistema.
Planificador (Jefe de área)	Es el encargado de generar los reportes de su departamento. Navega como un usuario autenticado.
investigador	Accede a toda la información básica del sistema, puede gestionar sus evidencias.

Tabla 3. Actores del sistema.

2.7. Diagrama de casos de uso del sistema.

La Figura 3 se muestra el diagrama de casos de uso para la gestión de Indicadores de ciencia y técnica.

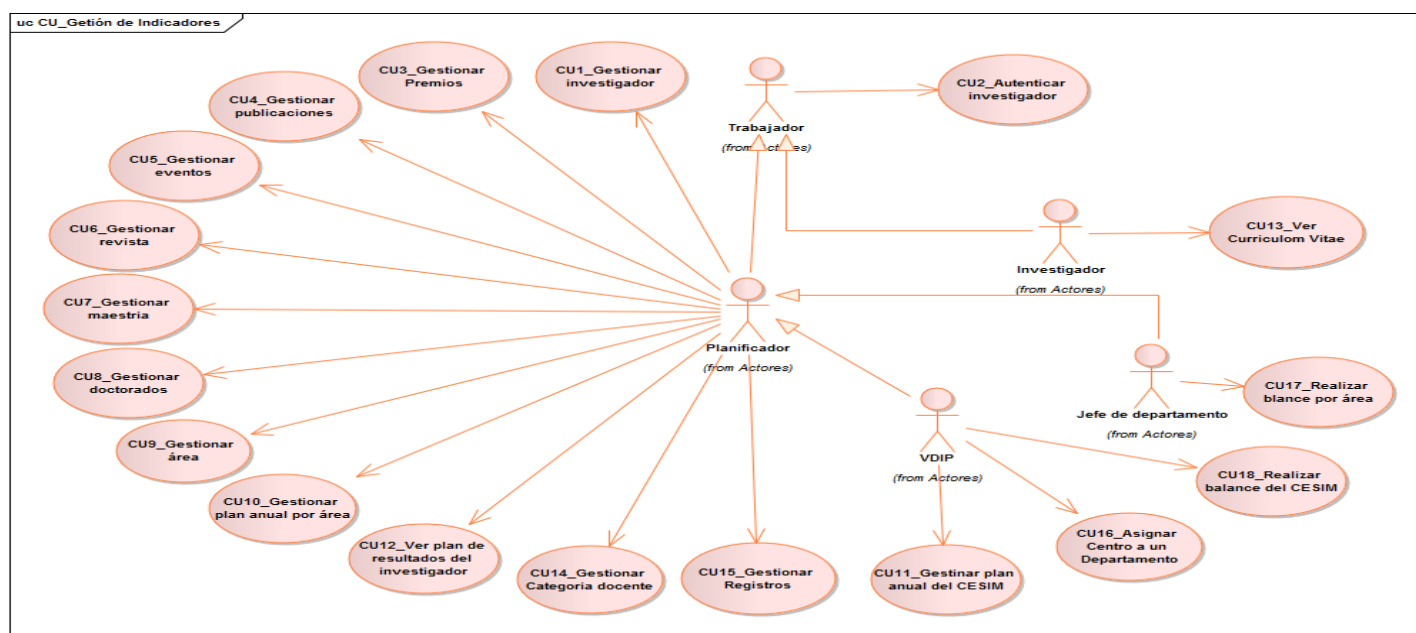


Figura 3. Diagrama de casos de uso gestión de Indicadores de ciencia y técnica.

2.8. Descripción de los casos de uso del sistema.

Las tablas que se muestran a continuación, contienen un resumen de los casos de uso del sistema arquitectónicamente significativos. En el [Anexo 4](#) se muestran las descripciones textuales de los mismos.

- Caso de uso - Gestionar plan anual de CESIM.

Objetivo	El objetivo de este CU es adicionar, asignar, modificar o eliminar el plan anual del centro.
Actores	VDIP (Inicia).
Resumen	El VDIP selecciona la opción “Gestionar Plan anual del CESIM” para consultar algunas de las funcionalidades que brinda. Luego de selecciona la funcionalidad deseada y el sistema muestra la interfaz correspondiente a la misma. El VDIP ejecuta los pasos necesarios para adicionar, modificar, eliminar o listar el plan anual del CESIM. El sistema da respuesta a la opción seleccionada. Realizando la planificación de las tareas para cada una de las diferentes áreas.
Complejidad	Media.
Prioridad	Alta.
Referencias	RF 11, RF 11.1, RF 11.2, RF 11.3, RF 11.4

Tabla 4.

- Caso de uso - Gestionar plan por área.

Objetivo	El objetivo de este CU es adicionar, asignar, modificar, eliminar o listar el plan anual de un área determinada.
Actores	Planificador (Inicia).
Resumen	El Planificador selecciona la opción “Gestionar Plan anual por área” para consultar algunas de las funcionalidades que brinda. Luego de selecciona la funcionalidad deseada y el sistema muestra la interfaz correspondiente a la misma. Planificador ejecuta los pasos

	necesarios para adicionar, modificar, eliminar o listar el plan anual por área. El sistema da respuesta a la opción seleccionada. Realizando la planificación de las diferentes actividades que deberá cumplir cada uno de los investigadores pertenecientes a la misma. Con el fin de lograr cumplir con los compromisos asignados en el año.
Complejidad	Media.
Prioridad	Alta.
Referencias	RF 10, RF 10.1, RF 10.2, RF 10.3, RF 10.4

Tabla 5.

- Caso de uso – Realizar balance de CESIM.

Objetivo	El objetivo que persigue el actor al realizar este CU es obtener el balance general del centro CESIM.
Actores	VDIP (Inicia).
Resumen	El VDIP selecciona a cuál de las diferentes áreas desea realizarle el Balance de ciencia y técnica. Luego de selecciona la funcionalidad deseada, el sistema muestra la interfaz correspondiente a la misma. El VDIP ejecuta los pasos necesarios para realizar el balance general. El sistema da respuesta a la opción seleccionada. Mostrando los resultados obtenidos por cada uno de las diferentes áreas, lo que le permitirá ver si se cumplió con la planificación que le fue asignada al CESIM.
Complejidad	Media.
Prioridad	Alta.

Referencias	RF18.1.
--------------------	---------

Tabla 6.

- Caso de uso - Realizar balance por área.

Objetivo	El objetivo que persigue el actor al realizar este CU es obtener el balance general por cada una de las diferentes áreas.
Actores	Jefe de departamento (Inicia).
Resumen	El Jefe de departamento selecciona a cuál de las diferentes áreas desea realizarle el Balance de ciencia y técnica. Luego de selecciona la funcionalidad deseada, el sistema muestra la interfaz correspondiente a la misma. El Jefe de departamento ejecuta los pasos necesarios para realizar el balance por área. El sistema da respuesta a la opción seleccionada. Mostrando los resultados obtenidos por cada uno de los investigadores de las diferentes áreas, lo que le permitirá ver si se cumplió con la planificación que le fue asignada a un área en específico.
Complejidad	Media.
Prioridad	Alta.
Referencias	RF17.1

Tabla 7.

2.9. Conclusiones del capítulo.

En el capítulo se establecieron y especificaron los procesos del negocio, y los requisitos funcionales y no funcionales. Se identificaron y describieron los casos de uso del sistema.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se presenta la arquitectura del sistema así como los diagramas de clases de diseño y de secuencia que participan en la realización de los casos de uso. Además se realiza una breve descripción de las clases involucradas en la realización de los casos de uso arquitectónicamente significativos.

3.1. Modelo arquitectónico.

La arquitectura es un nivel de diseño que consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del *software*, cumpliendo los objetivos del sistema. Tiene como misión principal la identificación de requerimientos críticos en la estructura del sistema y la reducción de riesgos asociados a la construcción del mismo. Debe orientarse a las necesidades del cliente y ser flexible en cuanto a futuras modificaciones tanto de *software* como de *hardware* evitando que efectos inesperados puedan tener repercusión en el sistema. (26)

La arquitectura debe ser resistente a cambios de *hardware*, *software* y en lo posible de tecnologías de desarrollo, siempre priorizando que sea resistente a los cambios que pueden experimentar los requisitos de *software*.

El Sistema de Gestión de Indicadores es un *software* con una arquitectura híbrida al usar las arquitecturas cliente servidor y otros estilos como arquitectura N capas, en un nivel inferior el patrón arquitectónico Modelo Vista Controlador como especificación de la arquitectura N capas, en un nivel más inferior se tienen como patrones de diseño Repositorio y los patrones *GRASP* de asignación de responsabilidades.

Dentro de los patrones *GRASP* usamos: Experto (en la clase Área donde se realiza el balance de ciencia y técnica de cada área, puesto que esta clase contiene la información de las áreas necesaria para realizar balance), Creador (cada clase consulta se encarga de persistir la clase asociada a ella en la base de datos cuando sea necesario) Controlador (cada vista del sistema tiene asociado una clase controladora que se encarga de manejar las peticiones, Unidirección (se evidencia en las clases consulta que enlaza la clase controladora con las clases entidades y *Singleton* (para crear una instancia de la base de datos para su uso en el momento que se requiera. Los elementos que lo componen responden a decisiones arquitectónicas para hacer cumplir requerimientos funcionales y no funcionales. Como es el caso de la base de datos la cual almacena los principales datos de los indicadores para permitir realizar búsquedas más rápidas de ellas. Los clientes están separados del servidor, como lo define el estilo arquitectónico predominante.

Otro de los patrones utilizados es el de Bajo acoplamiento y Alta cohesión. Su aplicación se pone de manifiesto donde la asignación de responsabilidades se realiza permitiendo la colaboración entre las clases, sin afectar el proceso de reutilización. También lo constituyen ejemplos de su uso la creación de clases controladoras, que permitió realizar las operaciones del sistema las cuales reflejan los procesos del negocio, factibles de manejar en otras capas, como la de interfaz o de presentación. (27)

Arquitectura cliente-servidor

Por ser el sistema una aplicación web debido a la necesidad de tener acceso a la información y muchos usuarios haciendo uso de la aplicación en diferentes niveles de permiso se utiliza el principio de la presente arquitectura.

Esta arquitectura es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta.

Ventajas:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red.
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas y diseñadas para el paradigma Cliente - Servidor que aseguran la seguridad en las transacciones y la facilidad de empleo.

Desventajas:

- La saturación de la red. A pesar de esta ser una desventaja de peso la aplicación no presenta grandes volúmenes de datos, ni de mucho nivel de procesamiento por lo que no se debe colapsar la red. (28)

Arquitectura por capas

El Patrón de arquitectura por capas es una de las técnicas más comunes que los arquitectos de *software* utilizan para dividir sistemas de *software* complicados. Al pensar en un sistema en términos de capas, se imaginan los principales subsistemas de *software* ubicados de la misma forma que las capas de un pastel, donde cada capa descansa sobre la inferior. En este esquema la capa más alta utiliza varios servicios definidos por la inferior, pero la última es inconsciente de la superior. Además, normalmente cada capa oculta las capas inferiores de las siguientes superiores a esta. (29)

Patrón arquitectónico Modelo-Vista-Controlador (MVC)

Patrón MVC (Modelo-Vista-Controlador) es una especificación del patrón arquitectónico por capas, el mismo se encarga de separar interfaces, fue creado con *Smalltalk-807*. El modelo es la capa del dominio, la vista es la capa de presentación y en el controlador están los objetos de flujo de trabajo.

Es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU8), se percibe frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Descripción del patrón:

El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos. Por otra parte la vista es la encargada de transformar la información obtenida por el modelo en las páginas web a las que acceden los usuarios y el controlador es quien coordina todos los demás elementos y transforma las peticiones del usuario en operaciones sobre el modelo y la vista. (30)

Patrón de diseño Repositorio

Según Martin Fowler el diseño del repositorio se define como "el mediador entre el dominio y las capas de asignación de datos utilizando una interfaz de adquisición, para el acceso a los objetos de dominio". Esta

definición tradicional describe el patrón de diseño de repositorio como una clase que se asigna a una tabla de base de datos, que contiene insertar, modificar, seleccionar, eliminar, y funciones de recaudación por persistir un objeto de clase especial en la base de datos. En esta aplicación web se ha creado una implementación específica del modelo de repositorio. Mientras que la actual versión no puede contener todo el conjunto de funciones *CRUD*, aun así la aplicación todavía se beneficia de las características del modelo de repositorio de código desacoplado y la separación lógica de capas, ayudando a gestionar las consultas de forma transparente, sin necesidad de acabar escribiendo código SQL y manteniendo desacoplado el código de negocio, de la forma de almacenar la información.

Patrones *GRASP* para la asignación de responsabilidades

Experto: Asignar una responsabilidad al experto en información. La clase que cuenta con la información necesaria para cumplir la responsabilidad.

Creador: Asignarle a la clase B la responsabilidad de crear una instancia de clase A en uno de los siguientes casos:

B agrega los objetos A.

B contiene los objetos A.

B registra las instancias de los objetos A o

B utiliza especialmente los objetos A.

B tiene los datos de inicialización que serán transmitidos hacia A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A). B es un creador de los objetos A. Si existe más de una opción, prefiera la clase B que agregue o contenga la clase A.

Controlador: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones: el sistema global (controlador de fachada). La empresa u organización global (controlador de fachada), algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas). (31)

3.2. Diseño

El diseño es un elemento importante a tener en cuenta cuando va a construir un sistema. En toda aplicación se debe representar y documentar el mismo, para esto se realiza una actividad fundamental en este flujo de

trabajo que es el Modelo de diseño. Su principal objetivo es producir una representación técnica del *software* que se va a desarrollar. El mismo representa la entrada al flujo de trabajo de la implementación, por tanto es necesario guardar y mantener el modelo de diseño a través de todo el ciclo de vida del *software*. Por lo que se puede decir que en la etapa de diseño se transforman los requerimientos en un modelo de diseño del sistema a implementar, se define la arquitectura robusta para el sistema y se adapta el mismo a un entorno de implementación, diseñado pensando en la eficiencia. Lo que permite que al sistema cumplir con los parámetros como extensibilidad, reusabilidad, compatibilidad, portabilidad y robustez, logrando que todo el *software* alcance un alto nivel de calidad. (32)

La línea base de la arquitectura sostendrá posteriormente los requerimientos del sistema. En esta etapa los requerimientos son transformados en un conjunto de clases que, relacionadas entre sí, logran llevarlos a cabo. Además se generan artefactos que expresan gráficamente cómo se realiza ese proceso.

Los patrones de diseño son soluciones simples y refinadas a problemas específicos y comunes del diseño orientado a objetos. Estos expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de *software*. Al entender el funcionamiento de estos patrones los diseños serán mucho más flexibles, modulares y reutilizables. (33)

Para que el sistema pueda ser implementado sin imprecisiones es necesario que el diseño esté bien delineado, y su principal objetivo es la elaboración de los diagramas de clases de diseño, que muestra las clases participantes en la realización de un caso de uso con todos sus atributos. Los diagramas de las clases del diseño correspondientes a los casos de uso arquitectónicamente significativos se muestran en el [Anexo 4](#).

Los diagramas de secuencia ilustran gráficamente las interacciones del actor y de las operaciones a que dan origen. Con el objetivo de alcanzar un mejor entendimiento de las actividades que se llevan a cabo en los casos de uso arquitectónicamente significativos, en el [Anexo 5](#) se manifiestan los diagramas de secuencia de dichos casos de uso.

3.2.1. Modelo de Datos.

Los modelos de datos aportan la base conceptual para crear aplicaciones que hacen un uso intensivo de datos. De igual modo proponen la base formal para las herramientas y técnicas utilizadas en el desarrollo y

3.3. Descripción de las clases.

En el paradigma de la programación orientada a objetos la clase es el componente más importante. Esta se presenta a sí misma, como una descripción en materia de acciones y propiedades pertenecientes a un objeto de la vida real. El paradigma de programación mencionado se basa en la identificación de esas clases que se evidencian en el dominio en que se enmarca el sistema, mediante sus operaciones y relaciones, logrando llevar a cabo la funcionalidad esperada del mismo.

La descripción de las clases permite obtener detalles de las mismas, principalmente las que están involucradas en la realización de los casos de uso arquitectónicamente significativos, estas descripciones pueden ser consultadas en el [Anexo 6](#).

3.4. Conclusiones del capítulo.

En este capítulo se presentaron los principales aspectos para el comienzo de la implementación del sistema, entre ellos los diagramas de clase del diseño, los diagramas de secuencia de los casos de uso arquitectónicamente significativos. Además, se dan a conocer los detalles de la arquitectura empleada y así como los diferentes patrones.

CAPÍTULO 4. IMPLEMENTACIÓN

El flujo de trabajo de implementación se inicia con el resultado del diseño y se implementa el sistema en términos de componentes. En este capítulo se presentan el modelo de componentes y el modelo de despliegue del componente desarrollado dando una visión de cómo quedará desarrollada y distribuida la aplicación.

El propósito principal de la fase de implementación es desarrollar la arquitectura y el sistema como un todo es decir, la identificación de componentes arquitectónicamente significativos, tales como componentes ejecutables y la asignación de componentes a los nodos en las configuraciones de redes. (34)

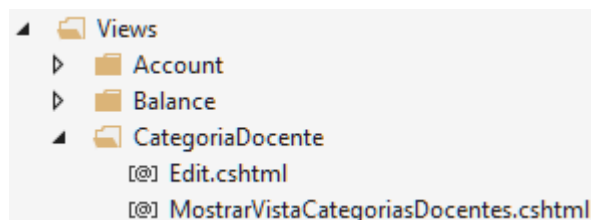
4.1. Elementos de implementación: MVC, NHibernate y jQuery.

Para la fase de implementación del sistema se usaron diversas librerías que permitieron la simplificación y reutilización del código, así como una mejor organización del mismo. El patrón arquitectónico MVC permitió separar el sistema en tres partes fundamentales: las clases modelos, las controladoras y las vistas.

A continuación se presenta una parte de la clase controladora “CategoriaDocenteController”, la vista correspondiente notifica a la controladora de los eventos de los usuarios, la controladora actualiza y obtiene los datos de la clase modelo “CategoriaDocente” y los muestra en la vista.

```
namespace Tesis.Controllers
{
    public class CategoriaDocenteController : Controller
    {
        public ActionResult MostrarVistaCategoriasDocentes()
        {
            return View();
        }
    }
}
```

Cada método obtenido de la clase controladora representa una vista en la página web.



Para el mapeo objeto-relacional que ayuda a identificar en el código los atributos de una clase modelo con las columnas de las tablas de la base de datos se utilizó NHibernate, haciendo posible manejar la base de datos a través de los objetos de .NET, con lo que se consigue una mayor abstracción y posibilidad de mantener el código.

Con el objetivo de establecer la comunicación con la base de datos los atributos de la clases modelos de cada entidad son mapeados utilizando la librería “FluentNHibernate”, para que cada atributo de la entidad se una con su elemento correspondiente en la base de datos.

NHibernate trae definido varios métodos que son las encargadas de conectarse con la base de datos y obtener, crear, modificar y eliminar los elementos necesarios en la misma, en dependencia de la petición enviada. A continuación se muestra un ejemplo de la clase “CategoriaDocenteQueries” la cual contiene todas las consultas que se pueden realizar a la entidad “CategoriaDocente” en la base de datos.

```
using ConfigurationModule.DB.Entities;
using NHibernate.Linq;

namespace ConfigurationModule.DB.Queries
{
    public class ConsultaCategoriaDocente
    {
        public static int AdicionarCategoriaDocente(CategoriaDocente
CategoriaDocente)
        {
            using (var session = DBFactory.GetSession())
            {
                using (var transaction = session.BeginTransaction())
                {
                    try
                    {
                        var NuevaCategoriaDocente = (int)session.Save(CategoriaDocente);
                        transaction.Commit();
                        return NuevaCategoriaDocente;
                    }
                    catch (Exception)
                    {
                        transaction.Rollback();
                        return -1;
                    }
                }
            }
        }
    }
}
```

Este estilo de programación, permite abstraer al programador de la manipulación de cada una de las características o atributos que conforman el objeto de la clase “CategoriaDocente” para hacerlos coincidir con cada uno de los campos de la tabla en la que finalmente será insertada la información.

Cuando se realiza una consulta a la base de datos con NHibernate, es necesario controlar la sesión. A través de la clase “DBFactory” utilizando métodos como: *SessionFactory*, *Provider*, *InitilizeFactory*, *GetSession* e *InitializeDBManager* se establece la conexión con la base de datos.

```
using NHibernate;
using NHibernate.Linq;
using NHibernate.Tool.hbm2ddl;

namespace ConfigurationModule.DB
{
    public class DBFactory
    {
        private static ISessionFactory _sessionFactory;

        private static ISessionFactory SessionFactory
        {
            get
            {
                try
                {
                    if (_sessionFactory == null)
                        return _sessionFactory;
                }
                catch (Exception ex)
                {
                    throw new TypeInitializationException("_sessionFactory", ex);
                }
                return _sessionFactory;
            }
        }
    }
}
```

Un ejemplo donde se trabaja con jQuery es en la página Realizar Planificación Anual, donde se realizan los compromisos para todo el CESIM, en la vista aparece un fieldset donde se recogen los compromisos generales del centro que se oculta con una función implementada en *JavaScript* para ampliar el área de trabajo al interactuar con los compromisos de los departamentos.

```
<script language="javascript" type="text/javascript">

var ver = true
function OcultarCompFac()
{

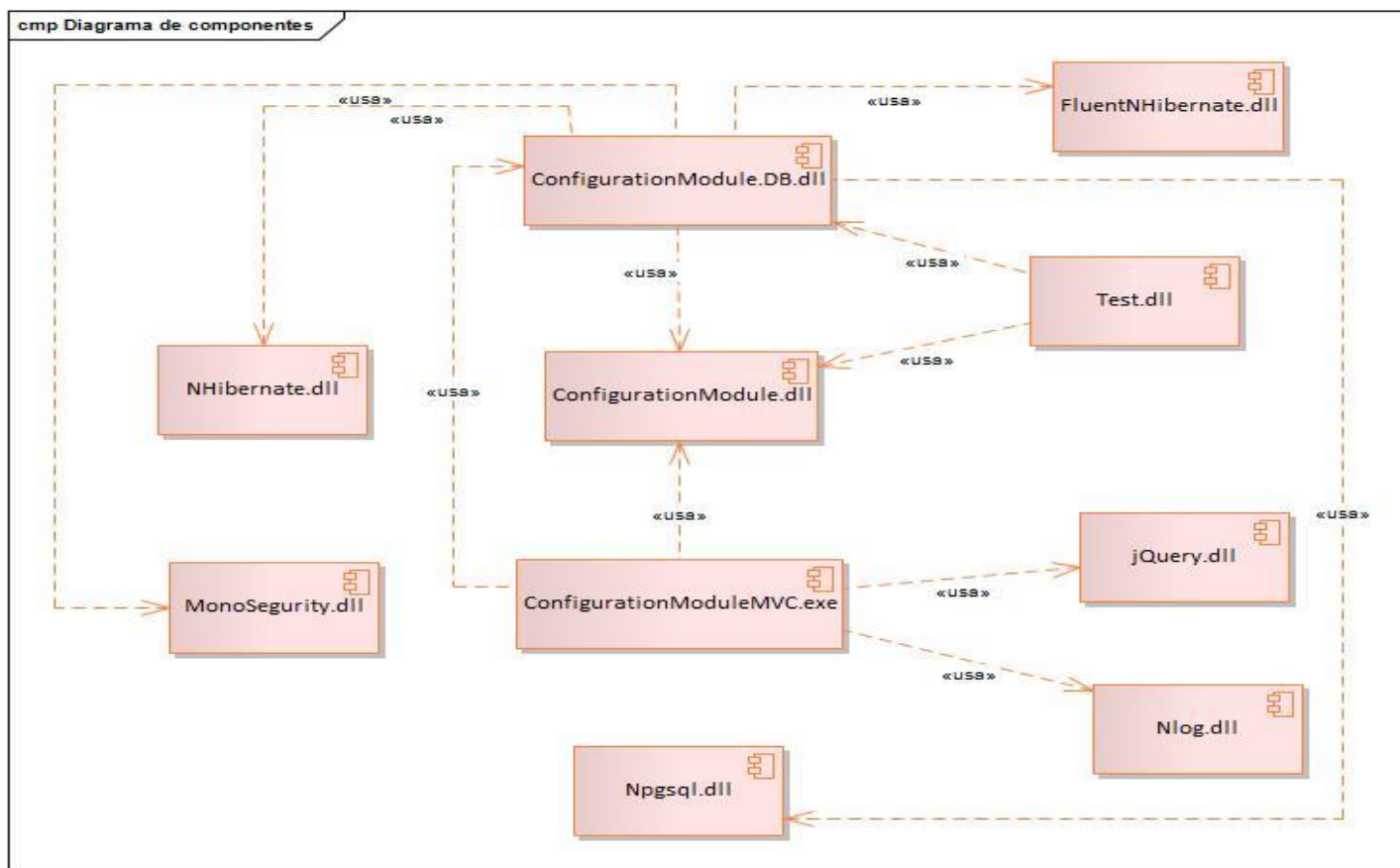
    if(ver)
    {
        document.getElementById('CompFac').style.display = 'block';
        document.getElementById('tablafac').style.display = 'block';
        ver = false
    }
    else
    {
        document.getElementById('CompFac').style.display = 'none';
        document.getElementById('tablafac').style.display = 'none';
        ver = true;
    }
}
}
```

4.2. Diagrama de componentes.

El diagrama de componentes describe la estructura de los componentes del sistema agrupados por paquetes lógicos. Se utiliza para estructurar el modelo de implementación en términos de subsistemas y modelar la vista estática de un sistema, describiendo sus elementos físicos y las relaciones entre estos. Representa cómo un sistema de *software* es dividido en componentes y muestra la organización y las dependencias lógicas entre estos.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente se refiere a los servicios ofrecidos por otro componente. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del *software*, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. (35)

En la figura 4 se muestra el diagrama de componentes asociado al componente desarrollado para la Gestión de Indicadores de Ciencia y Técnicas y sus proyecciones en CESIM.



4.2.1. Descripción de los componentes.

Componente	Descripción
ConfigurationModuleMVC.dll	Componente donde se encuentran todas las vistas y clases controladoras de la aplicación.
ConfigurationModule.dll	Librería que permite la gestión de la configuración del servidor de base de datos.
ConfigurationModule.DB.dll	Librería que permite la conexión con la base de datos.
Test.dll	Librería que permite verificar la comunicación con la base de

	datos.
NHibernate.dll	Librería que permite el mapeo y comunicación con la base de datos.
MonoSecurity.dll	Librería que le permite a la Npgsql.dll la comunicación con PostgreSQL a través de protocolos seguros.
Npgsql.dll	Librería que permite la comunicación del framework .NET con el PostgreSQL.
Log4Net.dll	Librería que permite la gestión de los sucesos ocurridos en el componente.
jQuery.js	Es una amplia librería de clases de JavaScript que proporciona funciones para el manejo y la navegación DOM.

Tabla 8 Descripción del diagrama de componentes.

4.3. Diagrama de despliegue.

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de *hardware* (nodos) y muestra cómo los elementos y artefactos del *software* se trazan en esos nodos. (36)

La figura 5 muestra el modelo de despliegue del componente de *software* para la gestión de Indicadores de Ciencia y Técnica y sus proyecciones en CESIM, el cual responde al despliegue de la solución en su conjunto.

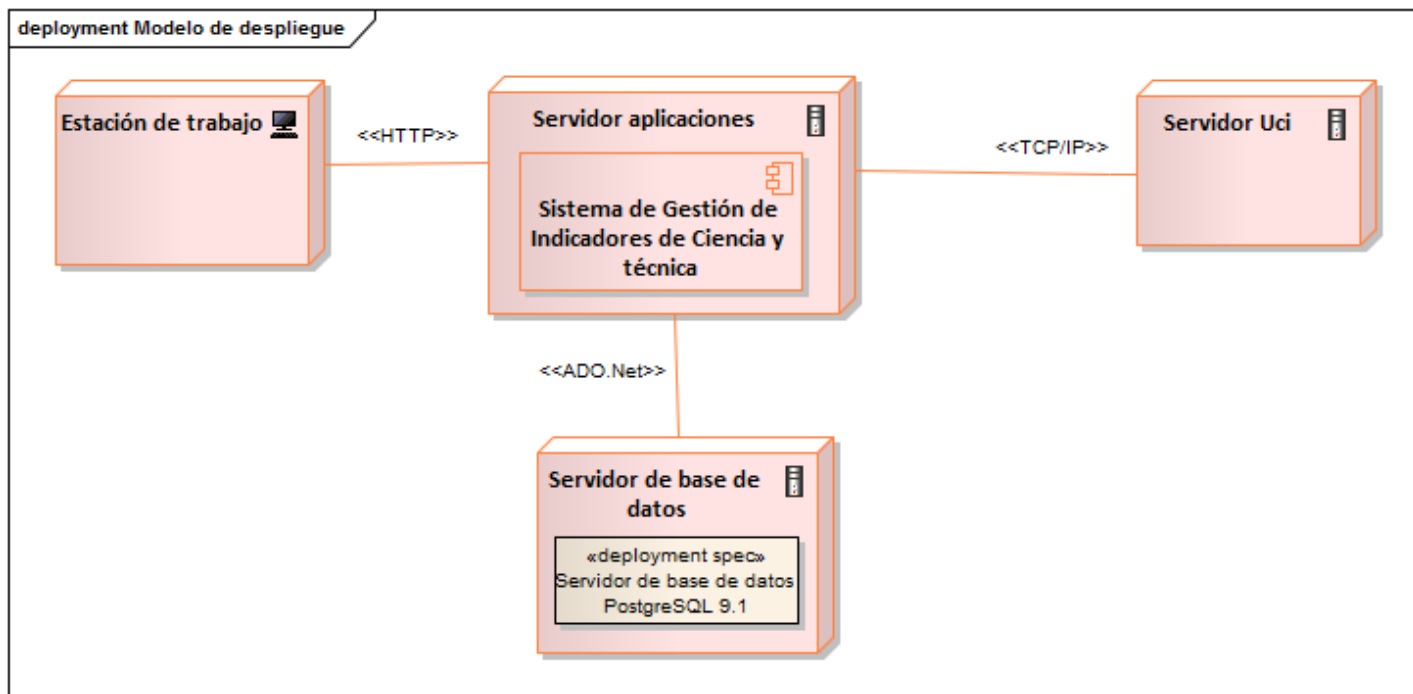


Figura 5 Modelo de despliegue.

4.3.1. Descripción de los nodos.

Nodo	Descripción
Estación trabajo	<p>Son las que consumen los servicios brindados por el servidor.</p> <p>Requerimientos de <i>hardware</i> y <i>software</i>: sistema operativo <i>Microsoft Windows XP</i> o Superior; requiere de la instalación de <i>Microsoft .NET Framework 4.5</i>. HDD: 80 GB, TR: 100 Mbps, RAM: 2 GB, CPU: Dual Core 2.5 Ghz, TV: 512 MB.</p>
Servidor de aplicaciones	<p>Proporciona servicios que soportan la ejecución y disponibilidad de las aplicaciones desplegadas.</p> <p>Requerimientos de <i>hardware</i> y <i>software</i>: Requiere</p>

	de la instalación de <i>Microsoft .NET Framework 4.5</i> e <i>InternetInformationServices 6.0</i> o superior. TR: 100 Mbps, RAM: 4 GB, CPU: <i>Dual Core 2.5 Ghz</i> .
Servidor de base de datos	Es donde se almacenan los datos del sistema y del servicio web propuesto. TR: 100 Mbps, RAM: 2 GB, CPU: <i>Dual Core 2.5 Ghz</i> .
Servidor de adquisición y ruteo de los datos	Es el servidor que permite el ruteo de los diferentes que se manejan en la universidad los equipos de adquisición de datos y son enviados a las estaciones clientes. TR: 100 Mbps, RAM: 2 GB, CPU: <i>Dual Core 2.5 Ghz</i> .

Tabla 9

4.4. Estándares de codificación.

Los estándares de codificación son modelos de programación que describen los aspectos de la generación de código que no están enfocados a la lógica de la aplicación, sino que para facilitar la lectura, comprensión y mantenimiento del código, están centrados en la estructura y apariencia física del mismo. Para ello los desarrolladores definen que aspectos deben estandarizar y que estilo se debe utilizar para cada aspecto.

Los aspectos para los que generalmente se establecen estándares son los siguientes:

- Identificadores.
- Indentación.
- Líneas y espacios en blanco.
- Comentarios.
- Declaraciones.
- Algoritmización.

4.4.1. Identificadores.

En la implementación del sistema se utilizó *UperCamelCase* como estilo codificación para las clases y las funciones, el cual define que cada palabra interna en los identificadores compuestos comiencen con mayúscula incluyendo la primera palabra del identificador. Los nombres de las variables se representaron con la primera palabra en minúscula, para el caso en el que el nombre sea compuesto se usó la notación *lowerCamelCase* en la cual la primera palabra comienza con minúscula y las demás con mayúsculas. Para los identificadores de las tablas los nombres se escribieron en minúscula antecedido de `tb_`, en caso de que el nombre sea compuesto se recurre a la notación *lowerCamelCase*.

En la clase controladora “EventosController” en el método “AsignarEventoTrabajador” se definió la siguiente variable donde se evidencia la utilización del estilo aplicado:

```
var iterador = ConsultaUsuarioEventos.ObtenerTodosUsuariosEventos().Count();
```

Otro ejemplo lo constituye en la clase “CategoriaDocenteMap” donde se definió el nombre de los métodos como se muestra a continuación utilizando el estilo apropiado:

```
public class CategoriaDocenteMap : ClassMap<CategoriaDocente>  
{  
    public const string TableName = "tb_categoria_docente";
```

4.4.2. Llaves.

En la implementación del sistema se definió que las llaves de apertura se colocarán solitarias en la línea siguiente e indentadas al nivel de la línea cabecera del bloque. Las llaves de cierre se colocarán solitarias en la línea que sigue a la última línea dentro del bloque. En el caso de cuerpos de bloque con una sola sentencia se podrá o no usar las llaves a gusto del programador. Este estilo agrega más líneas de código al programa al ubicar las llaves solitarias en una línea, pero a su vez se gana en legibilidad del código. A continuación se muestra un ejemplo de la utilización de este estilo en la implementación del sistema:

```
public static void AdicionarEventos(Eventos Evento)
{
    using (var session = DBFactory.GetSession())
    {
        using (var transaction = session.BeginTransaction())
        {
            try
            {
                session.Save(Evento);
                transaction.Commit();
            }
            catch (Exception)
            {
                transaction.Rollback();
            }
        }
    }
}
```

4.4.3. Líneas y espacios en blanco.

Con el objetivo de obtener una mejor legibilidad en el código se utilizan líneas en blanco para separar segmentos de código que pueden corresponder a clases, funciones, declaraciones, implementaciones, comentarios y bloques. En ocasiones cada operador es separado por espacios de su respectivo operando, paréntesis, identificadores, símbolos y algunos lenguajes exigen que se separen las palabras propias del vocabulario de las adyacentes para ser comprendidas por los compiladores.

Durante la implementación del sistema fueron colocados espacios en blanco después de cada punto y coma, al igual que después de las comas en las funciones y declaraciones. Y líneas en blanco entre las funciones y en las declaraciones en las vistas.

4.4.4. Comentarios.

El lenguaje C# utiliza la sintaxis “//” para definir comentarios lineales o de una sola línea. Estos comentarios fueron utilizados para describir las funciones y clases de las clases modelos. Cada comentario fue colocado encima de la línea a la que se le quiso aplicar el comentario o encima de la línea cabecera del bloque al que se le desea aplicar. A continuación se representa un bloque de comentarios correspondiente a la clase “CategoriaDocenteMap”:

```
/// <summary>
/// Esta clase representa el mapeo de la entidad Categoría Docente en la
/// base de datos de la solución
///
/// </summary>
```

4.5. Tratamiento de errores.

Durante la ejecución de un sistema pueden surgir numerosas situaciones de error producto de variables que no pueden ser controladas por el sistema. Por lo que es imprescindible tomar precauciones frente a situaciones como estas a fin de evitar cualquier tipo de error emergente que pueda surgir.

Para el manejo de este tipo de situaciones C# ofrece un sistema denominado manejo de excepciones. Una excepción es una situación anormal que se da a lo largo de la ejecución del programa. El manejador de la excepción, es una estructura de control que permite ejecutar un código de acuerdo a esa situación anormal de forma controlada. (37)

En el desarrollo del sistema se utilizó el manejo de excepciones en los métodos de las clases controladoras a través de bloques de estructura try/catch, los bloques try se utilizan para separar el código al que puede afectar una excepción y los bloques catch para controlar las excepciones resultantes. Esto hace posible capturar las excepciones que puedan arrojar los diversos métodos que se invoquen en el programa.

4.6. Constatación de hipótesis.

Una vez terminada la implementación del sistema se realizó una validación informal de la hipótesis. Para la que se tuvo en cuenta aspectos como: completitud, corrección y precisión de la información que se procesa.

Completitud: El sistema está diseñado de tal manera que recoge toda la información referente a los indicadores de ciencia y técnica para realizar el balance general.

Corrección: El sistema permite modificar la información de acuerdo a los permisos, en dependencia al rol que desempeñe cada usuario serán los privilegios que podrán tener en el mismo.

Precisión: Con la gestión de la evidencia, se podrá verificar que la información registrada en el sistema está acorde con los datos de los resultados planificados.

Estas dimensiones demuestran en su conjunto, sin tomar en cuenta la variable responsabilidad, que se elevaron los niveles de la integridad de la información.

4.7. Conclusiones del capítulo.

En este capítulo se presentaron elementos relacionados con la implementación del sistema, tales como el diagrama de componentes y la disposición física de los nodos que componen el sistema a través del

diagrama de despliegue. Así como los elementos generales relacionados con la implementación de los casos de usos arquitectónicamente significativos. Se definieron las políticas de implementación como por ejemplo el tratamiento de excepciones para identificar los posibles errores de a solución.

CONCLUSIONES

A partir del estudio de los sistemas de gestión de indicadores a nivel mundial se evidenció que todos los sistemas que controlan indicadores de ciencia y técnica se desarrollan de forma centralizada proponiendo una arquitectura cliente servidor.

El análisis de los procesos de negocio que se llevan a cabo durante el despliegue de la solución permitió identificar y priorizar las funcionalidades del *software* desarrollado.

Se definió una arquitectura híbrida cliente servidor que utiliza como patrón arquitectónico principal MVC y patrones de diseño Repositorio y *GRASP*, que garantizan características deseables como robustez, adaptabilidad, estabilidad, facilidad de mantenimiento, flexibilidad y extensibilidad.

Al finalizar la presente investigación se desarrolló el sistema de gestión de indicadores de ciencia y técnica, el cual permitirá medir el desarrollo científico y tecnológico de cada uno de los investigadores pertenecientes al CESIM. A través de esta solución se logra proteger y elevar la integridad de la información restringiendo los privilegios de inserción, eliminación y actualización en dependencia del rol que desempeñe cada usuario.

RECOMENDACIONES

Para perfeccionar el sistema desarrollado los autores recomiendan:

- Formular otros indicadores de ciencia técnica que no fueron priorizados para el desarrollo de la presente investigación.
- Aplicar técnicas de inteligencia artificial en el sistema para la realización de una planificación automática de los resultados de ciencia y técnica.

REFERENCIAS BIBLIOGRÁFICAS

1. González Rodríguez, Walfredo. *La utilización de un sistema de indicadores de ciencia y tecnología para la gestión de la actividad de investigación en las universidades cubanas*. 2008.
2. Benítez Cárdenas, Francisco y García Cueva, José Luís. *La utilización de un sistema de indicadores de ciencia y tecnología para la gestión de la actividad de investigación en las universidades cubanas*. 2010.
3. *Lineamiento de la política económica y social del partido y la revolución*. 2011. pág. 22.
4. Mondragón Pérez, Angélica Rocío. *Definición de indicadores*. s.l. : Asesora de la Oficina de la Presidencia del Instituto Nacional de Estadística, Geografía e Informática., 2002.
5. Albornoz, Mario y Jaramillo, Hernán. *El universo de la medición: La perspectiva de la ciencia y la tecnología*. 2007. pág. 13.
6. Albornoz, Mario y Martínez, Eduardo. *Indicadores de ciencia y tecnología: balance y perspectivas*. págs. 11-12.
7. México.gov. Sistema Integrado de Información sobre Investigación Científica y Tecnológica de México. [En línea] 2013. <http://www.siicyt.gob.mx/siicyt/>.
8. Arias, Francisco Javier. Edutecno. [En línea] 2009. http://edutecno.org/2009/02/ley_1286de2009.
9. Marianela Lafuente, Genatios Carlos. [En línea] 2004. <http://www.voltairenet.org/article120763.html>.
10. Dadak. buenastareas. [En línea] 2010. <http://www.buenastareas.com/ensayos/Colciencias/142424.html>.
11. Pérez Nobrega, Orlando. *Integración del Sistema de Indicadores Cientíomicos con la plataforma de Ciencia, Tecnología e Innovación de la UCI*. La Habana : s.n., 2010.
12. Delgado, Erly. Metodologías de desarrollo de software. [En línea] 2008. [Citado el: 18 de Febrero de 2014.] <http://www.redalyc.org/pdf/1939/193915935003.pdf>.
13. Rational Unified Process. *iteraproces.com*. [En línea] 2009. <http://www.iteraproces.com/beneficios-de-rational-unified-process-rup.html>.

14. Chacón, julio Cesar. Aplicación de la Metodología RUP para el Desarrollo Rápido de Aplicaciones Basado en el Estándar J2EE. [En línea] Marzo de 2006. http://biblioteca.usac.edu.gt/tesis/08/08_0308_CS.pdf.
15. Anacleto, Lic. Valerio Adrián. *Base de conocimiento de Epidata*. [En línea] 2008. http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15#Conceptos_b_sicos_sobre_UML.
16. Rodríguez Zurita, Elvia del Pilar. *Biblioteca UTPL*. [En línea] <http://dspace.utpl.edu.ec/bitstream/123456789/5467/1/TESIS%20GESTION%20DE%20PROCESOS%20DE%20NEGOCIOS.pdf>.
17. CMMI, Product Team. [En línea] Agosto de 2002. <http://www.fastwrite.com/dharma/02tr028.pdf>.
18. Sparx Systems. *Enterprise Architect User Guide*. 2009.
19. Visual Studio 2012. [En línea] 2010. <http://social.msdn.microsoft.com/Search/en-US/vstudio?query=visual%20studio%202012&Refinement=195&ac=3>.
20. ITTalent LEARNING SOLUTIONS. [En línea] <http://www.ittalent.com.co:26619/Certificaciones/VISUALST/Paginas/default.aspx>.
21. Fahnle, Pablo. Programación en Catellano. [En línea] 2006. http://www.programacion.com/articulo/que_es_asp_net_227.
22. Sharp, John. *Microsoft Press Microsoft Visual C Sharp 2010. Step by Step*. Washington : Microsoft Press, 2010.
23. Patricio, Anthony. JBoss Comunity. [En línea] 2010. <https://community.jboss.org/wiki/NHibernateForNET>.
24. Modelado de procesos de negocio. *TeraLoc*. [En línea] TeraLOC, 2009. [Citado el: 1 de Diciembre de 2011.] http://www.teraloc.com/portal/index.php?option=com_content&view=article&id=51&Itemid=92.
25. Sommerville, Ian. *Software engineering*. s.l. : Addison-Wesley, 2007.

26. Shaw, Mary y Garlan, David. *An introduction to software architecture*. s.l. : School of Computer Science, Carnegie Mellon University, 1994.
27. Crespo Pérez, David y Nelson, Francisco Fernández. *Desarrollo del Módulo Configuración del Sistema de Información Hospitalaria alas HIS*. Gestión Hospitalaria, Universidad de las Ciencias Informáticas. 2010. Trabajo de diploma.
28. Darke, Jose M. *Introducción a los sistemas distribuido: Paradigma cliente/servidor*. 2008.
29. Arevalo Lizardo, Maria Eugenia. Introducción al Patrón de Arquitectura por Capas. [En línea] 12 de Febrero de 2010. <http://arevalomaria.wordpress.com/2010/12/02/introduccion-al-patron-de-arquitectura-por-capas/>.
30. Márquez Gómez, José Jorge. José Jorge Márquez Gómez. [En línea] 2011. <http://jorge.queideas.com/wp-content/uploads/2011/11/Arquitectura-MVC.pdf>.
31. Visconti, Marcello y Astudillo, Hernan. Fundamentos de Ingeniería de software. <http://www.inf.utfsm.cl>. [En línea] 2002. <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.
32. Brito, Aristides Lescay y Jiménez Estrada, Pedro Manuel. *Componentes web Psicología y Psiquiatría del Sistema Integral para la Atención Primaria de Salud*. 2012.
33. Gracia, Joaquin. Patrones de diseño. Diseño de Software Orientado a Objetos. *Ingeniero Software*. [En línea] 27 de Mayo de 2005. [Citado el: 27 de Mayo de 2014.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
34. Poquioma, Jeysson. Ing. de Gestión de Software. Implementacion. *Scribd*. [En línea] 2010. <http://es.scribd.com/doc/27701099/Explicacion-de-la-fase-de-IMPLEMENTACION-de-la-metogologia-RUP>.
35. Teoría 1: El arte de modelar. [En línea] 2010. <http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf>.
36. Cartelle, Maria de Jesús. *Sistema para la planificación quirúrgica ortopédica*. 2011.
37. Roca, Pablo D. *Tratamiento de errores y control de recursos*. Argentina : s.n., 2011.

BIBLIOGRAFÍA

Albornoz, Mario y Jaramillo, Hernán. 2007. *El universo de la medición: La perspectiva de la ciencia y la tecnología.* 2007. pág. 13.

Albornoz, Mario y Martínez, Eduardo. 2007. *Indicadores de ciencia y tecnología: balance y perspectivas.* págs. 11-12.

Anacleto, Lic. Valerio Adrián. 2008. *Base de conocimiento de Epidata.* [En línea] 2008. http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15#Conceptos_b_sicos_sobre_UML.

Arevalo Lizardo, Maria Eugenia. 2010. Introducción al Patrón de Arquitectura por Capas. [En línea] 12 de Febrero de 2010. <http://arevalomaria.wordpress.com/2010/12/02/introduccion-al-patron-de-arquitectura-por-capas/>.

Arias, Francisco Javier. 2009. Edutecno. [En línea] 2009. http://edutecno.org/2009/02/ley_1286de2009.

Benítez Cárdenas, Francisco y García Cueva, José Luís. 2010. *La utilización de un sistema de indicadores de ciencia y tecnología para la gestión de la actividad de investigación en las universidades cubanas.* 2010.

Brito, Aristides Lescay y Jiménez Estrada, Pedro Manuel. 2012. *Componentes web Psicología y Psiquiatría del Sistema Integral para la Atención Primaria de Salud.* 2012.

Cartelle, Maria de Jesús. 2011. *Sistema para la planificación quirúrgica ortopédica.* 2011.

Chacón, julio Cesar. 2006. Aplicación de la Metodología RUP para el Desarrollo Rápido de Aplicaciones Basado en el Estándar J2EE. [En línea] Marzo de 2006. http://biblioteca.usac.edu.gt/tesis/08/08_0308_CS.pdf.

CMMI, Product Team. 2002. [En línea] Agosto de 2002. <http://www.fastwrite.com/dharma/02tr028.pdf>.

Crespo Pérez, David y Nelson, Francisco Fernández. 2010. *Desarrollo del Módulo Configuración del Sistema de Información Hospitalaria alas HIS.* Gestión Hospitalaria, Universidad de las Ciencias Informáticas. 2010. Trabajo de diploma.

- Dadak.** 2010. buenastareas. [En línea] 2010.
<http://www.buenastareas.com/ensayos/Colciencias/142424.html>.
- Darke, Jose M. 2008.** *Introducción a los sistemas distribuido: Paradigma cliente/servidor.* 2008.
- Delgado, Eryly. 2008.** Metodologías de desarrollo de software. [En línea] 2008. [Citado el: 18 de Febrero de 2014.] <http://www.redalyc.org/pdf/1939/193915935003.pdf>.
- Fahle, Pablo. 2006.** Programación en Catellano. [En línea] 2006.
http://www.programacion.com/articulo/que_es_asp_net_227.
- González Rodríguez, Walfredo. 2008.** *La utilización de un sistema de indicadores de ciencia y tecnología para la gestión de la actividad de investigación en las universidades cubanas.* 2008.
- Gracia, Joaquin. 2005.** Patrones de diseño. Diseño de Software Orientado a Objetos. *Ingeniero Software.* [En línea] 27 de Mayo de 2005. [Citado el: 27 de Mayo de 2014.]
<http://www.ingenierosoftware.com/analysisydiseno/patrones-diseno.php>.
- 2009.** Ingeniería de Software. *Arquitectura J2EE - Patron MVC.* [En línea] Marzo de 2009. [Citado el: 26 de Febrero de 2014.] <http://caraballomaestre.blogspot.com/2009/02/arquitectura-j2ee-patron-mvc.html>.
- ITTalent LEARNING SOLUTIONS. [En línea]
<http://www.ittalent.com.co:26619/Certificaciones/VISUALST/Paginas/default.aspx>.
- 2011.** *Lineamiento de la política económica y social del partido y la revolución.* 2011. pág. 22.
- Marianela Lafuente, Genatios Carlos. 2004.** [En línea] 2004. <http://www.voltairenet.org/article120763.html>.
- Márquez Gómez, José Jorge. 2011.** José Jorge Márquez Gómez. [En línea] 2011.
<http://jorge.queideas.com/wp-content/uploads/2011/11/Arquitectura-MVC.pdf>.
- México.gov. 2013.** Sistema Integrado de Información sobre Investigación Científica y Tecnológica de México. [En línea] 2013. <http://www.siicyt.gob.mx/siicyt/>.
- 2009.** Modelado de procesos de negocio. *TeraLoc.* [En línea] TeraLOC, 2009. [Citado el: 1 de Diciembre de 2011.] http://www.teraloc.com/portal/index.php?option=com_content&view=article&id=51&Itemid=92.
- Mondragón Pérez, Angélica Rocío. 2002.** *Definición de indicadores:* s.l. : Asesora de la Oficina de la Presidencia del Instituto Nacional de Estadística, Geografía e Informática., 2002.

- Mora, David. 2010.** Patrón Modelo-Vista-Modelo de Vista (MVVM). *Scribd*. [En línea] 2010. <http://maromasdigitales.net/2010/05/patron-mvvm-explicado/>.
- Patricio, Anthony. 2010.** JBoss Community. [En línea] 2010. <https://community.jboss.org/wiki/NHibernateForNET>.
- Pérez Nobrega, Orlando. 2010.** *Integración del Sistema de Indicadores Cientométricos con la plataforma de Ciencia, Tecnología e Innovación de la UCI*. La Habana : s.n., 2010.
- Poquioma, Jeysson. 2010.** Ing. de Gestión de Software. Implementacion. *Scribd*. [En línea] 2010. <http://es.scribd.com/doc/27701099/Explicacion-de-la-fase-de-IMPLEMENTACION-de-la-metodologia-RUP>.
- 2009.** Rational Unified Process. *iteraproces.com*. [En línea] 2009. <http://www.iteraproces.com/beneficios-de-rational-unified-process-rup.html>.
- Roca, Pablo D. 2011.** *Tratamiento de errores y control de recursos*. Argentina : s.n., 2011.
- Rodríguez Zurita, Elvia del Pilar. Biblioteca UTPL. [En línea]** <http://dspace.utpl.edu.ec/bitstream/123456789/5467/1/TESIS%20GESTION%20DE%20PROCESOS%20DE%20NEGOCIOS.pdf>.
- Sánchez, Jorge. 2004.** *MySQL*. 2004.
- Sharp, John. 2010.** *Microsoft Press Microsoft Visual C Sharp 2010. Step by Step*. Washington : Microsoft Press, 2010.
- Shaw, Mary y Garlan, David. 1994.** *An introduction to software architecture*. s.l. : School of Computer Science, Carnegie Mellon University, 1994.
- Sommerville, Ian. 2005.** *Ingeniería del Software, 7ma Edición*. Madrid : Pearson Educación, SA, 2005.
- . **2007.** *Software engineering*. s.l. : Addison-Wesley, 2007.
- Sparx Systems. 2009.** *Enterprise Architect User Guide*. 2009.
- 2010.** Teoría 1: El arte de modelar. [En línea] 2010. <http://fineans.usac.edu.gt:8001/rid=1HV0BP15X-15DBYBZ-FH/UML-diagramaComponentes.pdf>.
- Visconti, Marcello y Astudillo, Hernan. 2002.** Fundamentos de Ingeniería de software. <http://www.inf.utfsm.cl>. [En línea] 2002. <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.

2010. Visual Studio 2012. [En línea] 2010. <http://social.msdn.microsoft.com/Search/en-US/vstudio?query=visual%20studio%202012&Refinement=195&ac=3>.

ANEXOS

Anexo 1. Descripción del proceso Balance de ciencia y técnica.

Nombre:	Balance de Ciencia y técnica
Objetivos:	Planificar y gestionar los indicadores de ciencia y técnica del CESIM.
Evento(s) que lo generan:	Entrega de la planificación de indicadores al CESIM.
Precondiciones:	Debe haberse planificado un plan de actividades.
Postcondiciones:	Se archivó el Certificado y se actualizó el Curriculum Vitae. Se actualizó Plan de resultado y se creó el balance del CESIM.
Reglas de Negocio:	Ver documento 010110_Reglas_de_negocio del expediente de proyecto.
Responsables:	VDIP
Cientes internos:	VDIP, Activista de ciencia y técnica. Investigadores.
Cientes externos:	No procede.
Entradas:	Plan de indicadores (DOC).
Salidas:	Plan de Resultado (DOC). Curriculum Vitae (DOC). Certificado (JPG). Balance del CESIM (DOC).
Actividades:	<ol style="list-style-type: none"> 1. Planificar indicadores del CESIM. 2. Distribuir plan de indicadores. 3. Planificar indicadores a investigadores. 4. Asignar actividades al investigador. 5. Consultar Plan de actividades. 6. Cumplir con la planificación. 7. Entregar evidencia. 8. Archivar evidencia. 9. Entregar los resultados del departamento.

10. Realizar balance del CESIM.

Anexo 2. Prefijos de los requerimientos no funcionales por categoría.

Categoría	Prefijo	Ejemplo
Usabilidad	RNU	RNU 1. Requisito de usabilidad A. RNU 2. Requisito de usabilidad B.
Diseño e Implementación	RNDI	RNDI 1. Requisito de diseño e implementación.
Legal	RNL	RNL 1. Requisito legal.
Fiabilidad	RNF	RNF 1. Requisito de fiabilidad.
Seguridad	RNS	RNS 1. Requisito de seguridad.
Eficiencia	RNE	RNE 1. Requisito de eficiencia.
Funcionamiento	RNFO	RNFO 1. Requisito de funcionamiento.
Soporte	RNSO	RNSO 1. Requisito de soporte.
Interfaz de usuario	RNIU	RNIU 1. Requisito de interfaz de usuario.
Interconexión	RNI	RNI 1. Requisito de interconexión.
Componentes Comprados	RNCC	RNCC 1. Requisito de componentes comprados.
Licencia	RNLI	RNLI 1. Requisito de licencia.

Anexo 3. Descripción ampliada de los casos de uso significativos.

- Caso de uso - Gestionar Plan anual del CESIM.

Objetivo	El objetivo de este CU es adicionar, eliminar, modificar y listar los datos del plan anual del CESIM.
Actores	VDIP (Inicia).
Resumen	El VDIP selecciona la opción "Gestionar Plan anual del CESIM" para consultar algunas de las funcionalidades que brinda. Luego de selecciona la funcionalidad deseada y el sistema muestra la interfaz correspondiente a la misma. El VDIP ejecuta los pasos necesarios para adicionar, modificar, eliminar o listar el plan anual del CESIM. El sistema da respuesta a la

	opción seleccionada. Realizando la planificación de las tareas para cada una de las diferentes áreas.
Complejidad	Media.
Prioridad	Crítico.
Referencias	RF 11, RF 11.1, RF 11.2, RF 11.3, RF 11.4
Precondiciones	El VDIP se ha autenticado en el sistema. Se han gestionado los diferentes eventos, publicaciones, premios, doctorados maestría, cambio de categoría a asignar por cada una de las diferentes áreas.
Postcondiciones	Los datos generales del plan anual por área fueron correctamente adicionados / modificados / eliminados / listados de la configuración del sistema.
Flujo de eventos	
Flujo básico “Gestionar Plan anual del CESIM”	
<ol style="list-style-type: none"> 1. El VDIP selecciona la opción “Plan anual del CESIM” de la página principal. 2. El Sistema muestra la página solicitada con las opciones correspondientes. 3. El VDIP selecciona la opción que desee realizar. <ul style="list-style-type: none"> – Si selecciona la opción “Asignar el plan anual del CESIM”, ir a la Sección 1 “Asignar plan anual de CESIM”. – Si selecciona el ícono “Modificar” para modificar el plan anual del CESIM, ir a la Sección 2 “Modificar datos del plan anual del CESIM”. – Si selecciona el ícono “Eliminar” para eliminar plan anual del CESIM, ir a la Sección 3 “Eliminar plan anual del CESIM”. – Si selecciona la opción “Listar plan anual del CESIM”, ir a la Sección 4 “Listar plan anual del CESIM”. 4. Termina el caso de uso. 	
Sección 1 “Asignar un Plan anual del CESIM”	
<ol style="list-style-type: none"> 1. El Sistema muestra en la vista “Asignar un Plan anual del CESIM” los datos necesarios para realizar la asignación de las diferentes actividades del plan anual por cada una de las diferentes áreas. 2. El VDIP introduce los datos del nuevo plan anual de la facultad (publicaciones, eventos, maestría doctorados, y categoría científica). 	

<ol style="list-style-type: none"> 3. Presiona la opción "Aceptar". 4. El Sistema valida los datos introducidos por el VDIP. 5. El Sistema registra los datos introducidos. 6. Termina el caso de uso.
Flujos alternos
3a. Existen campos vacíos.
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje de error indicando que "Existen campos vacíos".
3b. Entrada de datos no válidos.
<ol style="list-style-type: none"> 1. El sistema muestra el mensaje "Caracteres incorrectos". 2. Regresa al paso 2 de flujo básico.
2a. El VDIP selecciona la opción "Cancelar".
<ol style="list-style-type: none"> 1. El sistema carga la página de los "Plan anual del CESIM". 2. Termina el caso de uso.
Sección 2 "Modificar los datos del Plan anual del CESIM"
<ol style="list-style-type: none"> 1. El VDIP selecciona de la página "Plan anual del CESIM" en la tabla donde se listan los datos del plan anual del CESIM el ícono Modificar para modificar el plan anual del CESIM. 2. El Sistema muestra los datos actuales del plan anual del CESIM en la vista "Modificar plan anual del CESIM". 3. El VDIP modifica los datos de configuración que necesite. 4. Selecciona la opción "Aceptar" para actualizar los datos. 5. El Sistema valida los datos introducidos por el VDIP. 6. El Sistema actualiza los datos modificados. 7. Termina el caso de uso.
Flujos alternos
4a. Existen campos vacíos.
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje de error indicando que "Existen campos vacíos".
4b. Entrada de datos no válidos.
<ol style="list-style-type: none"> 1. El sistema muestra el mensaje "Caracteres incorrectos". 2. Regresa al paso 2 de flujo básico.

3a. El VDIP selecciona la opción “Cancelar”.		
<ol style="list-style-type: none"> 1. El sistema carga la página de los “Plan anual del CESIM”. 2. Termina el caso de uso. 		
Sección 3 “Eliminar el Plan anual del CESIM”		
<ol style="list-style-type: none"> 1. El VDIP selecciona de la página “Plan anual del CESIM” en la tabla donde se listan los datos referentes al plan anual del CESIM el ícono Eliminar para eliminar un plan anual del CESIM. 2. El Sistema muestra un mensaje de confirmación “Se eliminará el plan anual seleccionado. Al seleccionar Aceptar se perderán todos los datos. ¿Desea continuar? ”. 3. El VDIP selecciona la opción “Sí”. 4. El Sistema elimina el plan anual del CESIM. 5. Termina el caso de uso. 		
Flujos alternos		
3c. El VDIP selecciona la opción “No”.		
<ol style="list-style-type: none"> 1. El sistema carga la página de los “Plan anual del CESIM”. 2. Termina el caso de uso. 		
Sección 4 “Listar plan anual del CESIM”		
<ol style="list-style-type: none"> 1. El Sistema muestra en la página “Plan anual del CESIM” un listado del plan anual del CESIM que está registrado en el sistema. 		
Flujo básico		
No procede.		
Relaciones	CU Incluidos	No aplicable.
	CU Extendidos	No aplicable.
Requisitos funcionales	no	No aplicable.
Asuntos pendientes		No aplicable.

- Caso de uso - Gestionar Plan anual por área.

Objetivo	El objetivo de este CU es adicionar, eliminar, modificar y listar los datos de un plan anual por área.
-----------------	--

Actores	Planificador (Inicia).
Resumen	El Planificador selecciona la opción “Gestionar Plan anual por área” para consultar algunas de las funcionalidades que brinda. Luego de selecciona la funcionalidad deseada y el sistema muestra la interfaz correspondiente a la misma. Planificador ejecuta los pasos necesarios para adicionar, modificar, eliminar o listar el plan anual por área. El sistema da respuesta a la opción seleccionada. Realizando la planificación de las diferentes actividades que deberá cumplir cada uno de los investigadores pertenecientes a la misma. Con el fin de lograr cumplir con los compromisos asignados en el año.
Complejidad	Media.
Prioridad	Crítico.
Referencias	RF 10, RF 10.1, RF 10.2, RF 10.3, RF 10.4
Precondiciones	El Planificador se ha autenticado en el sistema. Se han gestionado los diferentes eventos, publicaciones, premios, doctorados maestría, cambio de categoría a asignar por cada una de las diferentes áreas.
Postcondiciones	Los datos generales del plan anual por área fueron correctamente adicionados / modificados / eliminados / listados de la configuración del sistema.
Flujo de eventos	
Flujo básico “Gestionar Plan anual por área”	
<ol style="list-style-type: none"> 1. El planificador selecciona la opción “Plan anual por área” de la página principal. 2. El Sistema muestra la página solicitada con las opciones correspondientes. 3. El planificador selecciona la opción que desee realizar. <ul style="list-style-type: none"> – Si selecciona la opción “Adicionar el plan anual por área”, ir a la Sección 1 “Adicionar el plan anual por área”. – Si selecciona el ícono “Modificar” para modificar el plan anual por área, ir a la Sección 2 “Modificar datos del plan anual por área”. – Si selecciona el ícono “Eliminar” para eliminar plan anual por área, ir a la Sección 3 “Eliminar plan anual por área”. 	

<ul style="list-style-type: none"> – Si selecciona la opción “Listar plan anual por área”, ir a la Sección 4 “Listar plan anual por área”. <p>4. Termina el caso de uso.</p>
Sección 1 “Adicionar un Plan anual por área”
<ol style="list-style-type: none"> 1. El Sistema muestra en la vista “Agregar un plan anual por área” los datos necesarios para realizar la asignación de las diferentes actividades del plan anual por cada una de las diferentes áreas. 2. El planificador introduce los datos del nuevo plan anual por área (publicaciones, eventos, maestría doctorados, y categoría científica). 3. Presiona la opción “Aceptar”. 4. El Sistema valida los datos introducidos por el planificador. 5. El Sistema registra los datos introducidos. 6. Termina el caso de uso
Flujos alternos
3a. Existen campos vacíos.
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje de error indicando que “Existen campos vacíos”.
3b. Entrada de datos no válidos.
<ol style="list-style-type: none"> 1. El sistema muestra el mensaje “Caracteres incorrectos”. 2. Regresa al paso 2 de flujo básico.
2a. El planificador selecciona la opción “Cancelar”.
<ol style="list-style-type: none"> 1. El sistema carga la página de los “Plan anual por área”. 2. Termina el caso de uso.
Sección 2 “Modificar los datos del Plan anual por área”
<ol style="list-style-type: none"> 1. El planificador selecciona de la página “Plan anual por área” en la tabla donde se listan los datos del plan anual por área el ícono Modificar para modificar el plan anual por área. 2. El Sistema muestra los datos actuales del plan anual por área en la vista “Modificar plan anual por área”. 3. El planificador modifica los datos de configuración que necesite. 4. Selecciona la opción “Aceptar” para actualizar los datos. 5. El Sistema valida los datos introducidos por el planificador. 6. El Sistema actualiza los datos modificados. 7. Termina el caso de uso.
Flujos alternos

4a. Existen campos vacíos.		
1. El sistema muestra un mensaje de error indicando que “Existen campos vacíos”.		
4b. Entrada de datos no válidos.		
1. El sistema muestra el mensaje “Caracteres incorrectos”.		
2. Regresa al paso 3 de flujo básico.		
3a. El Jefe de departamento selecciona la opción “Cancelar”.		
1. El sistema carga la página de los “Plan anual por área”.		
2. Termina el caso de uso.		
Sección 3 “Eliminar el Plan anual por área”		
1. El planificador selecciona de la página “Plan anual por área” en la tabla donde se listan los datos referentes al plan anual por área el ícono Eliminar para eliminar un plan anual por área.		
2. El Sistema muestra un mensaje de confirmación “Se eliminará el plan anual por área seleccionado. Al seleccionar Aceptar se perderán todos los datos. ¿Desea continuar? ”.		
3. El planificador selecciona la opción “Sí”.		
4. El Sistema elimina el plan anual por área.		
5. Termina el caso de uso.		
Flujos alternos		
3a. El planificador selecciona la opción “No”.		
1. El sistema carga la página de los “Plan anual por área”.		
2. Termina el caso de uso.		
Sección 4 “Listar plan anual por área”		
1. El Sistema muestra en la página “Plan anual por área” un listado el plan anual por cada una de las diferentes áreas que están registrados en el sistema.		
2. Termina el caso de uso.		
Flujo básico		
No procede.		
Relaciones	CU Incluidos	No aplicable.
	CU Extendidos	No aplicable.

Requisitos funcionales	no	No aplicable.
Asuntos pendientes		No aplicable.

- Caso de uso - Realizar balance por área.

Objetivo	El objetivo de este CU es Realizar balance de ciencia y técnica por cada una de las áreas del CESIM.
Actores	Jefe de departamento (Inicia).
Resumen	El Jefe de departamento selecciona a cuál de las diferentes áreas desea realizarle el Balance de ciencia y técnica. Luego de selecciona la funcionalidad deseada, el sistema muestra la interfaz correspondiente a la misma. El Jefe de departamento ejecuta los pasos necesarios para realizar el balance por área. El sistema da respuesta a la opción seleccionada. Mostrando los resultados obtenidos por cada uno de los investigadores de las diferentes áreas, lo que le permitirá ver si se cumplió con la planificación que le fue asignada a un área en específico.
Complejidad	Media.
Prioridad	Alta.
Referencias	RF17, RF 17.1.
Precondiciones	El VDIP se ha autenticado.
Postcondiciones	El balance se realizó de manera correcta y eficiente.
Flujo de eventos	
Flujo básico “Realizar balance por área.”	
<ol style="list-style-type: none"> 1. El VDIP selecciona la opción “Realizar balance por área.”. 2. El sistema muestra una nueva ventana. 3. El VDIP selecciona el área a la que desea realizarle el balance en una lista despegable. 4. El VDIP da clic en el botón “Aceptar”. 5. El sistema muestra el balance del área seleccionada. 	

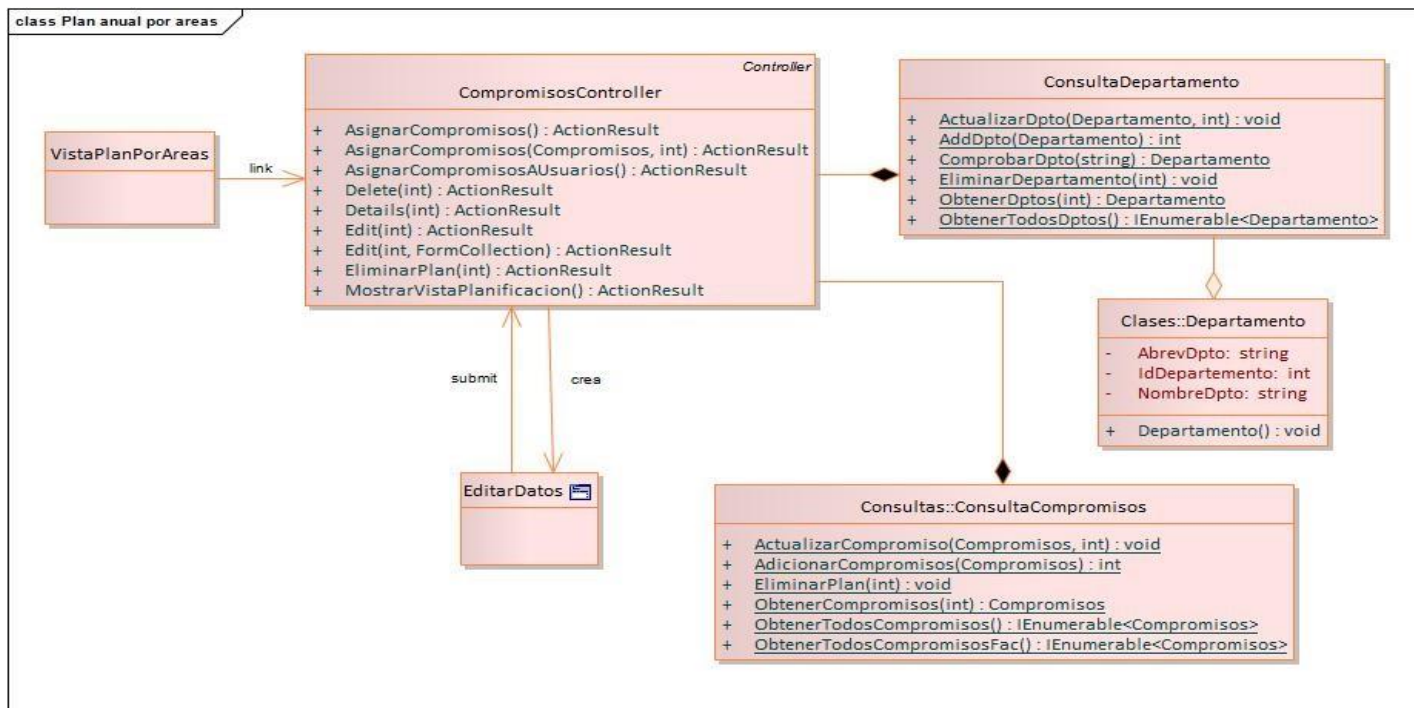
6. Termina el caso de uso.		
4a. Debe seleccionar el área.		
1. El sistema muestra un mensaje de error indicando que "Debe seleccionar el área".		
4b. Entrada de datos no válidos.		
1. El sistema muestra el mensaje "Selección incorrecta".		
2. Regresa al paso 2 de flujo básico.		
3a El VDIP selecciona la opción "Cancelar".		
1. El sistema carga la página "Realizar balance".		
2. Termina el caso de uso.		
Relaciones	CU Incluidos	No aplicable.
	CU Extendidos	No aplicable.
Requisitos funcionales	no	No aplicable.
Asuntos pendientes		No aplicable.

- Caso de uso - Realizar balance del CESIM.

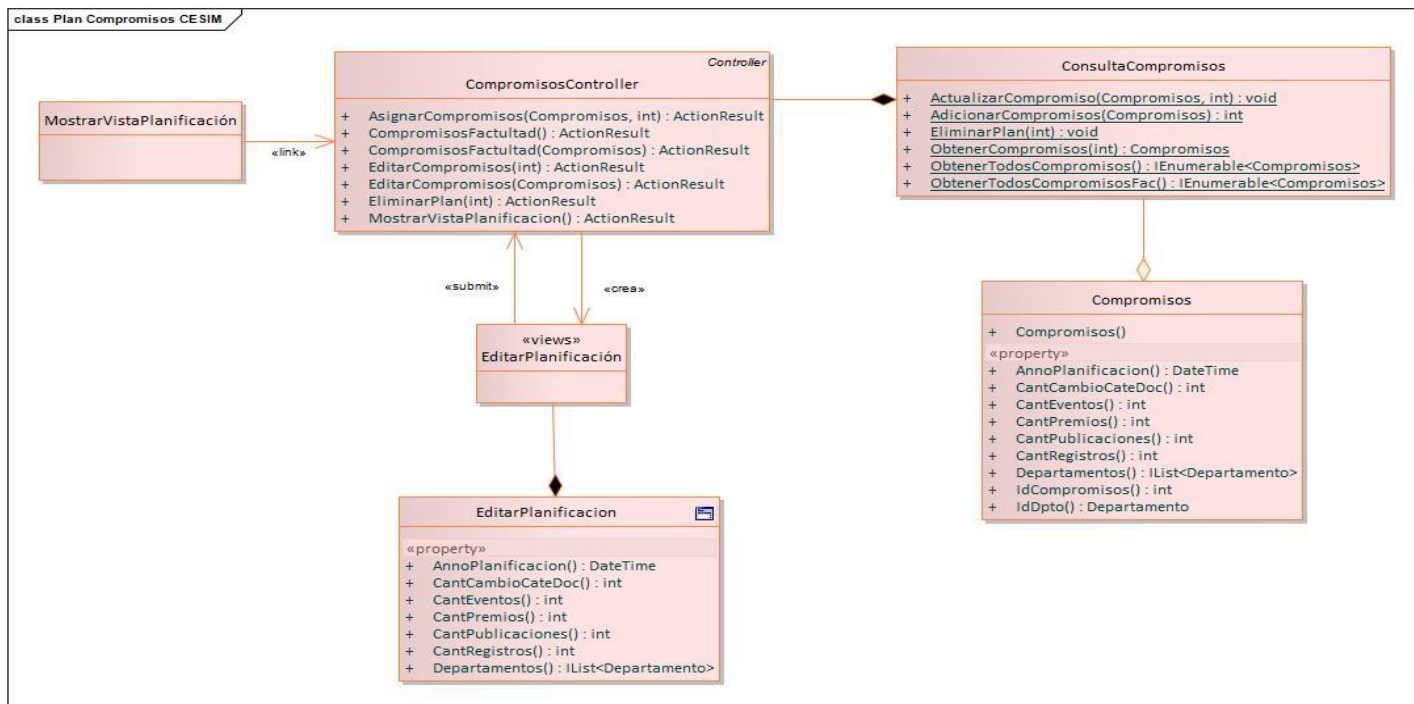
Objetivo	El objetivo de este CU es Realizar balance de ciencia y técnica del CESIM
Actores	VDIP (Inicia).
Resumen	El VDIP selecciona a cuál de las diferentes áreas desea realizarle el Balance de ciencia y técnica. Luego de selecciona la funcionalidad deseada, el sistema muestra la interfaz correspondiente a la misma. El VDIP ejecuta los pasos necesarios para realizar el balance general. El sistema da respuesta a la opción seleccionada. Mostrando los resultados obtenidos por cada uno de las diferentes áreas, lo que le permitirá ver si se cumplió con la planificación que le fue asignada al CESIM.
Complejidad	Media.
Prioridad	Alta.

Referencias	RF 18.1	
Precondiciones	El VDIP se ha autenticado.	
Postcondiciones		
Flujo de eventos		
Flujo básico “Realizar balance del CESIM.”		
<ol style="list-style-type: none"> 1. El VDIP selecciona la opción “Realizar balance del CESIM.”. 2. El sistema muestra la interfaz correspondiente. 3. El VDIP da clic en el botón “Aceptar”. 4. El sistema muestra el balance del CESIM. 5. Termina el caso de uso. 		
Flujos alternos		
3a. Selección incorrecta.		
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje de error indicando que ha realizado “Selección incorrecta”. 2. Regresa al paso 2 de flujo básico. 		
2a. El planificador selecciona la opción “Cancelar”.		
<ol style="list-style-type: none"> 1. El sistema carga la página “Realizar Balance.”. 2. Termina el caso de uso. 		
Relaciones	CU Incluidos	No aplicable.
	CU Extendidos	No aplicable.
Requisitos funcionales	no	No aplicable.
Asuntos pendientes	No aplicable.	

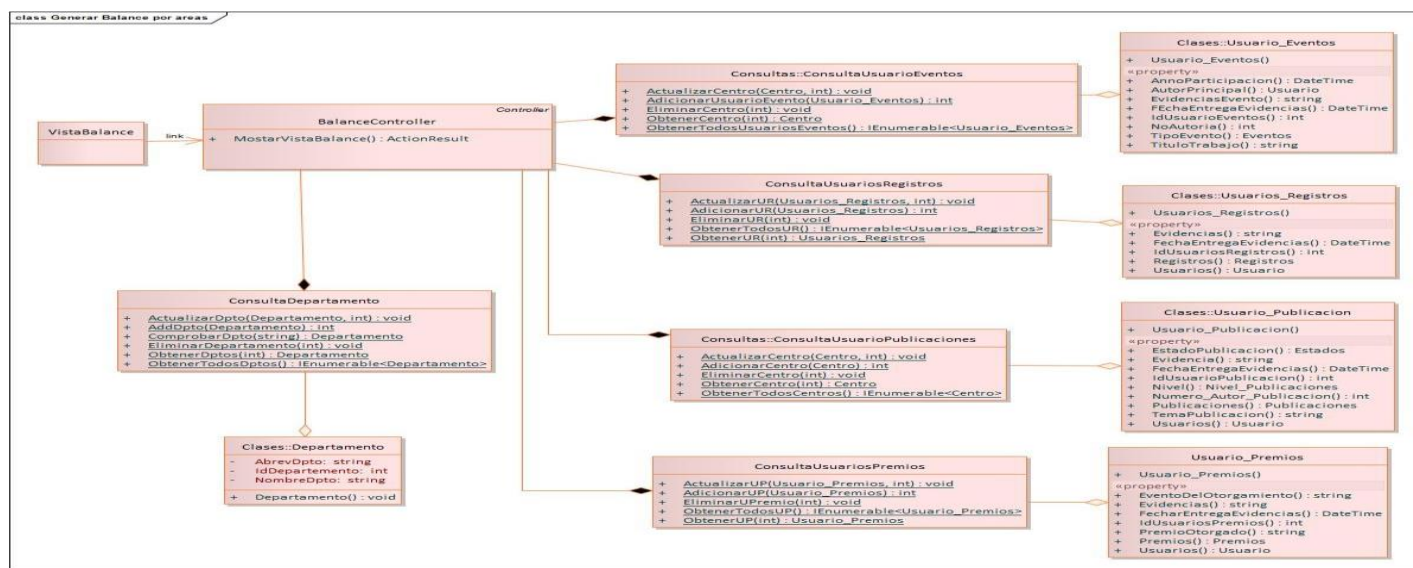
Anexo 4. Diagramas de clases del diseño de los casos de uso arquitectónicamente significativos.



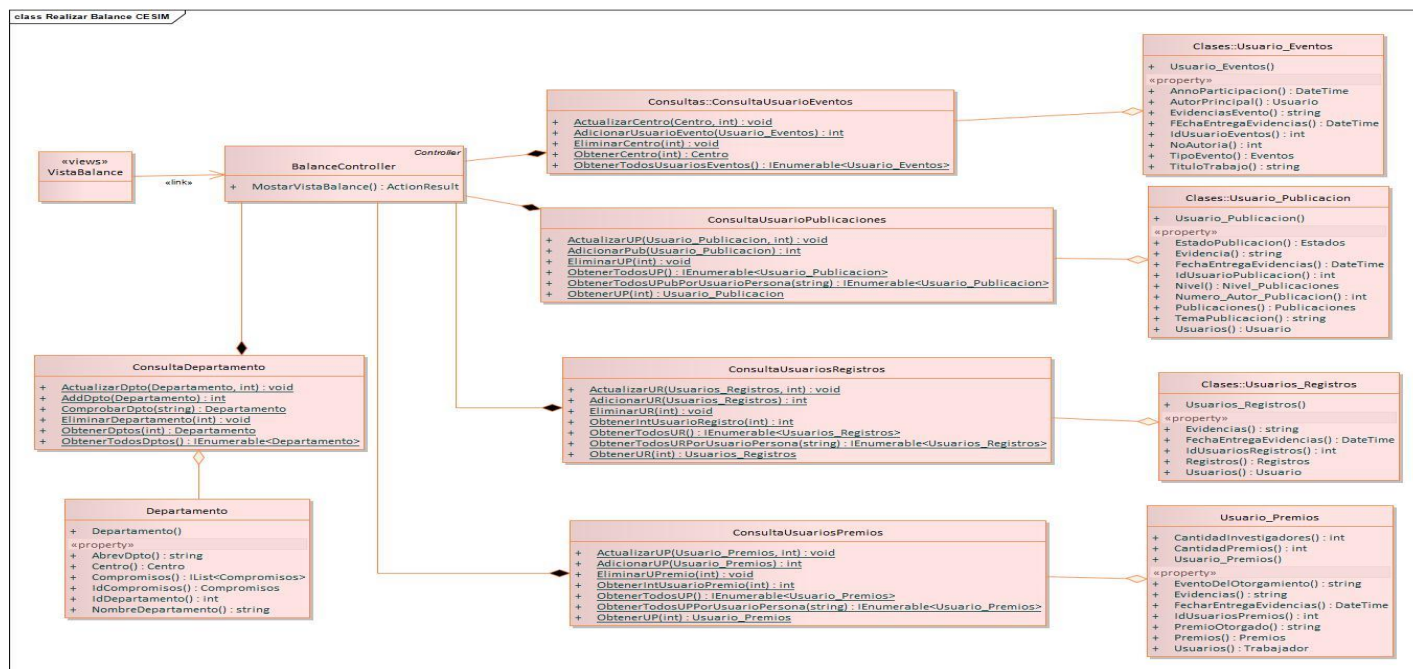
DC_Gestionar plan por área.



DC_Gestionar plan del CESIM

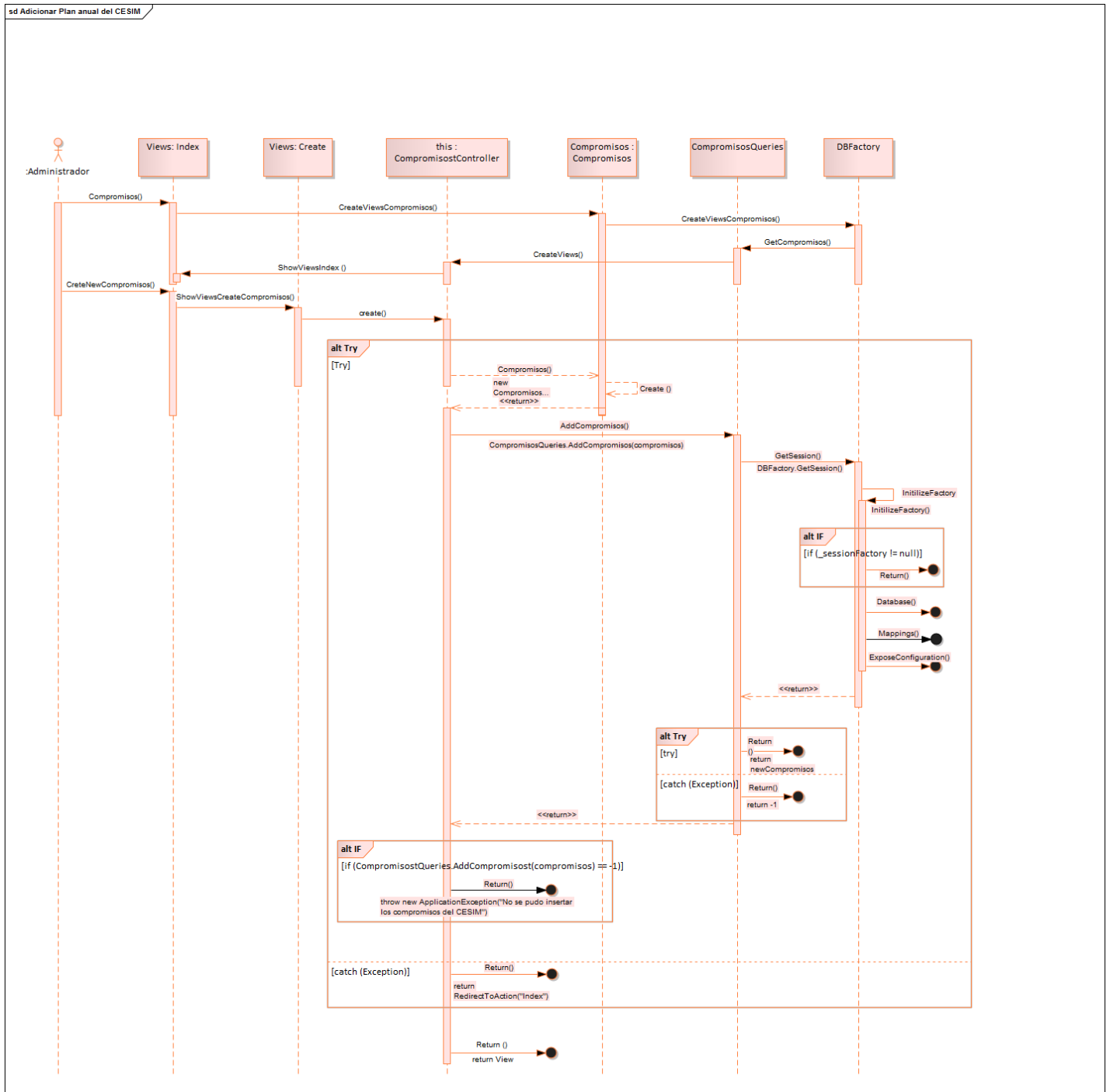


DC_Realizar balance por área.

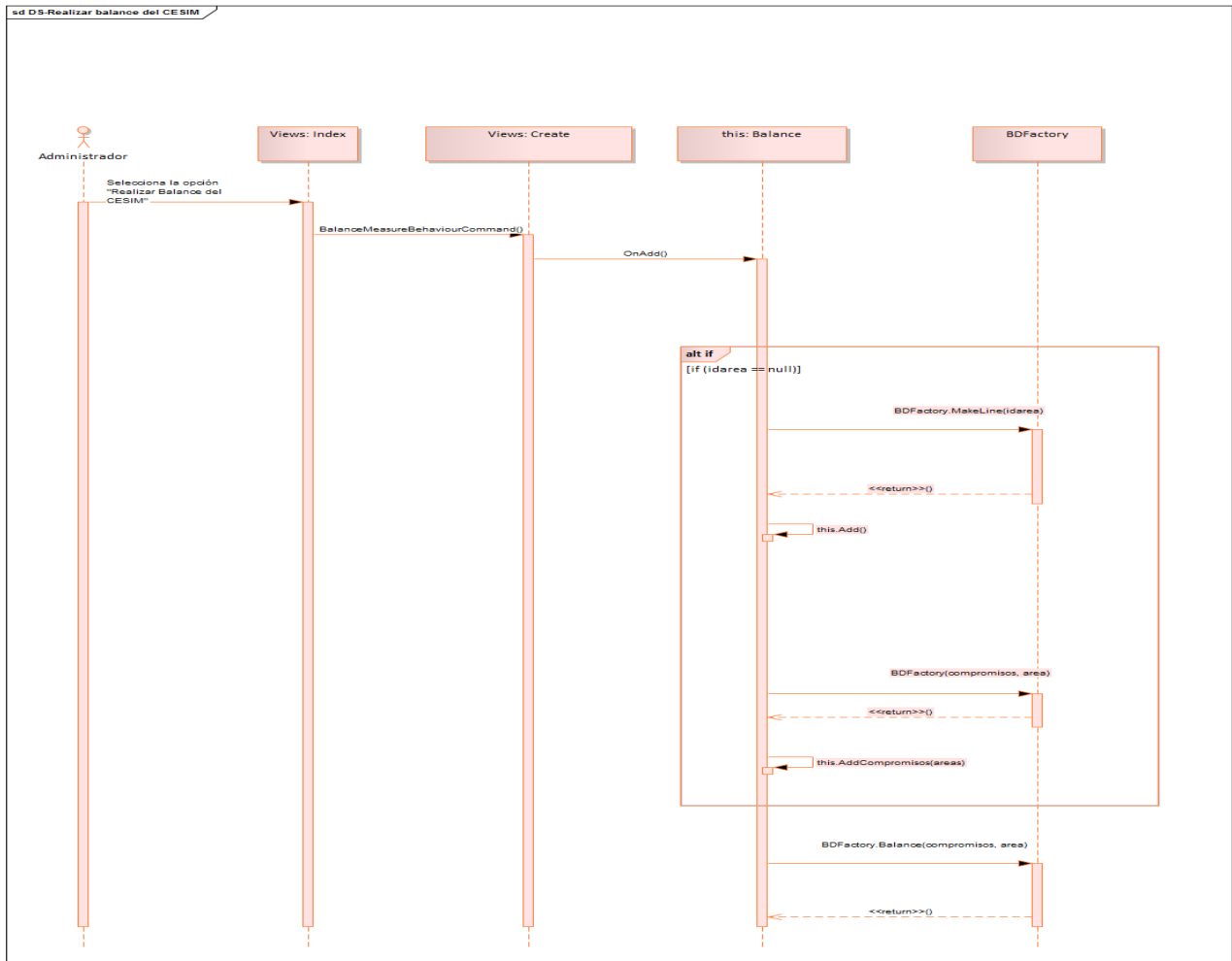


DC_Realizar Balance de CESIM.

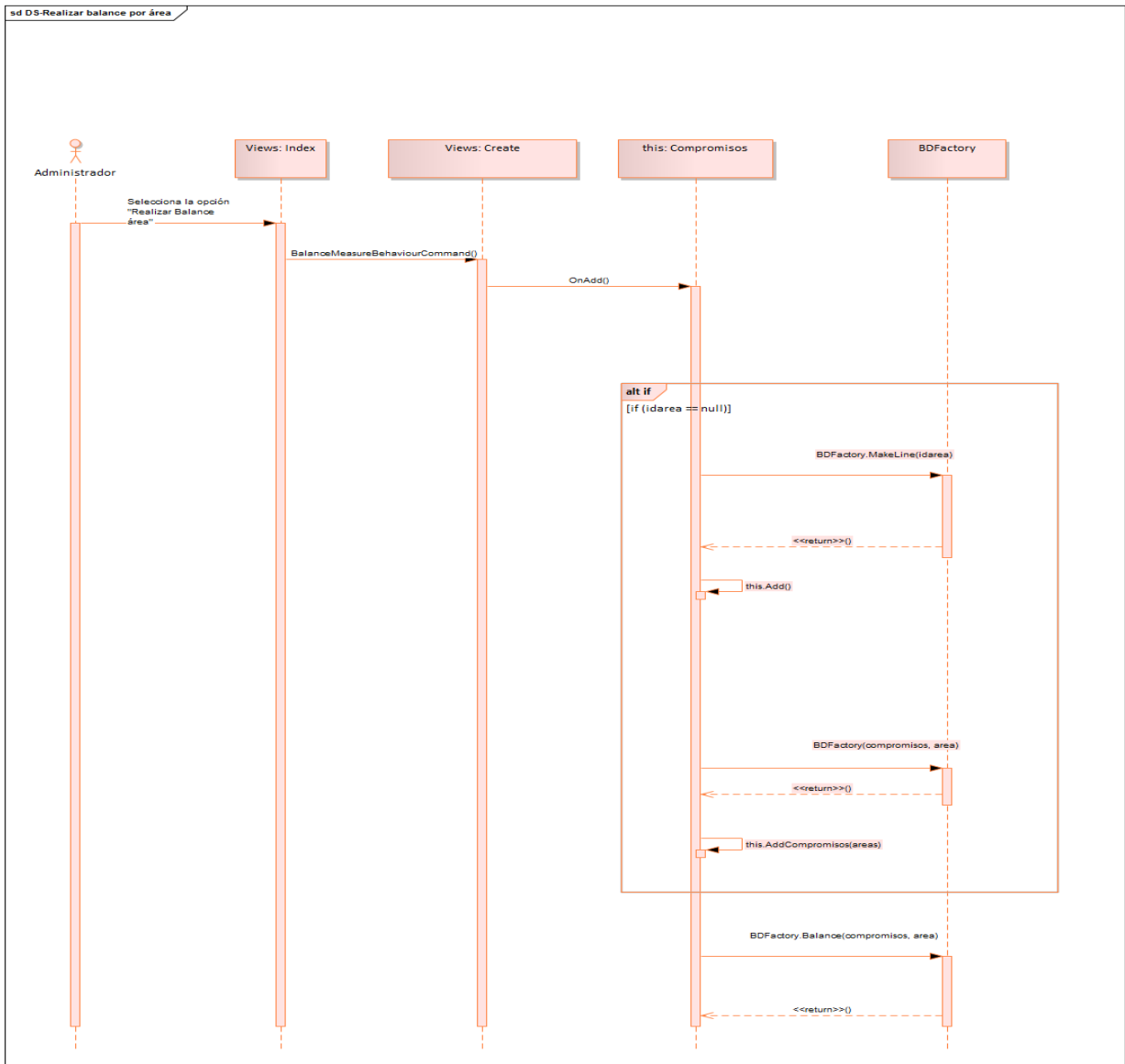
Anexo 5. Diagramas de secuencia de los casos de uso arquitectónicamente significativos.



DS_Adicionar Plan anual del CESIM.



DS-Realizar balance del CESIM.



DS-Realizar balance por área.

Anexo 6. Descripciones de clases.

- Clase ComprimisosControler

Propósito: Manejar los compromisos que realiza el usuario en la vista.

Descripción

Nombre: CompromisosController.	
Tipo de clase: controladora	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad:	
Nombre:	MostrarVistaPlanificacion()
Descripción:	Crea la Vista Planificacion.
Nombre:	CompromisosFactultad(Compromisos Compromisos)
Descripción:	Crea la vista parcial para crear un compromiso.
Nombre:	AsignarCompromisos(Compromisos comp, int id_dpto)
Descripción:	Le asigna un compromisos al departamento seleccionado
Nombre:	EditarCompromisos(int idCompromisos)
Descripción:	Crea la vista EditarCompromisos
Nombre:	EditarCompromisos(Compromisos Compromiso)
Descripción:	Edita los atributos de la clase compromisos
Nombre:	EliminarPlan(int IdCompromisos)
Descripción:	Elimina el plan de compromisos

- Clase Departamento

Propósito: Contiene toda la información actual de los departamentos en el sistema.

Descripción

Nombre: Departamento	
Tipo de clase: Entidad.	
Atributo	Tipo
IdDepartamento	Int
NombreDepartamento	String

string AbrevDpto	String
Centro	Centro
Para cada responsabilidad:	
Nombre:	No procede
Descripción:	No procede

- Clase ConsultaDepartamento.

Propósito: Permite persistir en la base de datos la clase departamento.

Descripción

Nombre: ConsultaDepartamento.	
Tipo de clase: Entidad.	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad:	
Nombre:	AdicionarDepartamento(Departamento dpto)
Descripción:	Adiciona en base de datos la clase departamento
Nombre:	ObtenerDepartamento(int p)
Descripción:	Obtiene un departamento pasado por parámetros
Nombre:	ComprobarDepartamento(string nombre)
Descripción:	Comprueba que el nombre del departamento no exista en base de datos
Nombre:	ObtenerTodosDepartamentos()
Descripción:	Devuelve una lista con todos los departamentos existentes
Nombre:	ActualizarDepartamento(Departamento dpto, int a)
Descripción:	Actualizar un departamento especificado con nuevos datos.
Nombre:	EliminarDepartamento(int h)

Descripción:	Elimina el departamento
--------------	-------------------------

- Clase ConsultaCompromisos.

Propósito: Permite persistir en la base de datos la clase compromisos.

Descripción

Nombre: ConsultaCompromisos	
Tipo de clase: Entidad.	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad:	
Nombre:	AdicionarCompromisos(Compromisos comp)
Descripción:	Adiciona un compromiso en base de datos
Nombre:	ObtenerCompromisos(int p)
Descripción:	Obtiene un compromisos pasado pro parámetro
Nombre:	ObtenerTodosCompromisos()
Descripción:	Devuelve un lista con todo los compromisos
Nombre:	ActualizarCompromiso(Compromisos nuevosdatos, int a)
Descripción:	Actualiza un compromiso
Nombre:	EliminarCompromiso(int h)
Descripción:	Elimina un compromiso

- Clase BalanceController.

Propósito: Manejar las acciones del usuario en la vista balance.

Descripción

Nombre: BalanceController

Tipo de clase: controladora	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad:	
Nombre:	MostarVistaBalance()
Descripción:	Crea la vista Balance

- Clase ConsultaInvestigadorEvento.

Propósito: Permite persistir en la base de datos la asignación de evento de un investigador.

Descripción

Nombre: ConsultaInvestigadorEvento	
Tipo de clase: Entidad.	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad:	
Nombre:	AdicionarInvestigadorEvento(InvestigadorEventos UsuarioEvento)
Descripción:	Adiciona una asignación de evento al investigador
Nombre:	ObtenerTodosEventosPorUsuarioPersona(string usuario)
Descripción:	Obtiene todos los eventos para un investigador por el usuario
Nombre:	ObtenerInvestigadorEvento(int p)
Descripción:	Obtiene el eventos del investigador que se pasa por parámetro
Nombre:	ObtenerTodosInvestigadorEventos()
Descripción:	Devuelve todo los eventos que tienen los investigadores
Nombre:	ActualizarInvestigadorEvento(InvestigadorEventos nuevonombre, int a)
Descripción:	Actualiza la clase con los nuevos datos

Nombre:	EliminarInvestigadorEvento(int h)
Descripción:	Elimina la asignación de evento del investigador

- Clase ConsultaInvestigadorregistro.

Propósito: Permite persistir en la base de datos la asignación de registro de un investigador.

Descripción

Nombre: ConsultaInvestigadorEvento	
Tipo de clase: Entidad.	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad:	
Nombre:	AdicionarInvestigadorRegistro(InvestigadorRegistros InvestigadorRegistro)
Descripción:	Adiciona una asignación de registro a un investigador
Nombre:	ObtenerInvestigadoresRegistro(int p)
Descripción:	Obtiene una asignación de registro de un investigador
Nombre:	ObtenerTodosInvestigadoresRegistro()
Descripción:	Devuelve una lista de asignaciones de registro de los investigadores
Nombre:	ActualizarInvestigadorRegistros(InvestigadorRegistros nuevonombre, int a)
Descripción:	Actualiza los nuevos datos de la asignación
Nombre:	EliminarInvestigadorRegistros(int h)
Descripción:	Elimina la asignación de registro del investigador

- Clase ConsultaInvestigadorPublicaciones.

Propósito: Permite persistir en la base de datos la asignación de publicaciones de un investigador.

Descripción

Nombre: ConsultaInvestigadorPublicaciones	
Tipo de clase: Entidad.	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad:	
Nombre:	AdicionarPub(InvestigadorPublicacion up)
Descripción:	Adiciona una asignación publicación a un investigador
Nombre:	ObtenerInvestigadorPublicacion(int p)
Descripción:	Obtiene una asignación de publicación a un investigador
Nombre:	ObtenerTodosInvestigadorPublicacion()
Descripción:	Devuelve una lista con todas las asignaciones de publicaciones de los investigadores
Nombre:	ActualizarInvestigadorPublicacion(InvestigadorPublicacion nuevonombre, int a)
Descripción:	Actualiza la asignación con nuevos datos
Nombre:	EliminarInvestigadorPublicacion(int h)
Descripción:	Elimina la asignación

- Clase ConsultaInvestigadorPremios.

Propósito: Permite persistir en la base de datos la asignación de premios de un investigador.

Descripción

Nombre: ConsultaInvestigadorPremios	
Tipo de clase: Entidad.	
Atributo	Tipo
No procede	No procede
Para cada responsabilidad:	

Nombre:	AdicionarInvestigadorPremios(InvestigadorPremios up)
Descripción:	Adiciona una asignación premios a un investigador
Nombre:	ObtenerInvestigadorPremios(int p)
Descripción:	Obtiene una asignación de premios a un investigador
Nombre:	ObtenerTodosInvestigadorPremios()
Descripción:	Devuelve una lista con todas las asignaciones de premios de los investigadores
Nombre:	ActualizarInvestigadorPremios(InvestigadorPremios nuevonombre, int a)
Descripción:	Actualiza la asignación con nuevos datos
Nombre:	EliminarInvestigadorPremios(int h)
Descripción:	Elimina la asignación

- Clase InvestigadorEvento.

Propósito: Permite almacenar los atributos de la asignación de evento del investigador.

Descripción

Nombre: InvestigadorEvento	
Tipo de clase: Entidad.	
Atributo	Tipo
IdUsuarioEventos	int
TituloTrabajo	string
NoAutoria	int
AnnoParticipacion	DateTime
EvidenciasEvento	string
FechaEntregaEvidencias	DateTime
Eventos	Eventos
AutorPrincipalTrabajo	Investigador

Para cada responsabilidad:	
Nombre:	No procede
Descripción:	No procede

- Clase InvestigadorPublicaciones.

Propósito: Permite almacenar los atributos de la asignación de publicaciones del investigador.

Descripción

Nombre: InvestigadorPublicaciones	
Tipo de clase: Entidad.	
Atributo	Tipo
IdUsuarioPublicacion	int
Numero_Autor_Publicacion	int
TemaPublicacion	string
EstadoPublicacion	Estados
Nivel	string
Evidencia	string
FechaEntregaEvidencias	DateTime
Publicaciones	Publicaciones
Investigadores	Investigador
Para cada responsabilidad:	
Nombre:	No procede
Descripción:	No procede

- Clase InvestigadorRegistro.

Propósito: Permite almacenar los atributos de la asignación de registro del investigador.

Descripción

Nombre: InvestigadorRegistro	
Tipo de clase: Entidad.	
Atributo	Tipo
IdUsuariosRegistros	int
ProductoServicioARegistrar	string
Version	string
Evidencias	string
EstadoRegistro	Estados
FechaEntregaEvidencias	DateTime
Registros	Registros
Investigadores	Investigador
Para cada responsabilidad:	
Nombre:	No procede
Descripción:	No procede

- Clase InvestigadorPremios.

Propósito: Permite almacenar los atributos de la asignación de premios del investigador.

Descripción

Nombre: InvestigadorPremios	
Tipo de clase: Entidad.	
Atributo	Tipo
IdUsuariosPremios	int
EventoDelOtorgamiento	string
PremioOtorgado	string

Evidencias	string
FechaEntregaEvidencias	DateTime
Premios	Premios
Investigadores	Investigador
Para cada responsabilidad:	
Nombre:	No procede
Descripción:	No procede

Anexo 7. Entrevistas a Vicedecanos de investigación y postgrado.

Facultad:

Nombres y apellidos:

Cargo que desempeña:

Relación de preguntas

1. ¿Quiénes intervienen en la planificación y control de los indicadores de ciencia y técnica? ¿Cómo se realiza este proceso?
2. ¿Cuáles son los indicadores de ciencia y técnica que se planifican y a los que se les lleva el control?

¿Se realiza un balance con el resultado registrado? ¿A qué niveles se desarrolla y cada qué tiempo?

GLOSARIO DE TÉRMINOS

Diagrama: representación gráfica de un conjunto de elementos. Visualiza un sistema desde diferentes perspectivas.

Metodología: proceso de software detallado que define con precisión los artefactos, roles y actividades involucradas.

Framework (Marco de trabajo): estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.