

Universidad de las Ciencias Informáticas

Facultad 1



Solución para la gestión de la estructura investigativa de la Universidad de las Ciencias Informáticas

*Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas*

Autores: Yudaimy Estevez Muñoz
Ariel Gómez Fal

Tutores: MSc. Osiris Perez Moya
Ing. Sergio René Vazquez Rodriguez
Ing. Mara Morales Alfonso

Ciudad de La Habana, junio de 2013

Declaración de autoría

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

Para que así conste firmamos la presente a los ____ días del mes de Junio del año 2013

Yudaimy Estevez Muñoz

Firma del Autor

Ariel Gómez Fal

Firma del Autor

MSc. Osiris Perez Moya

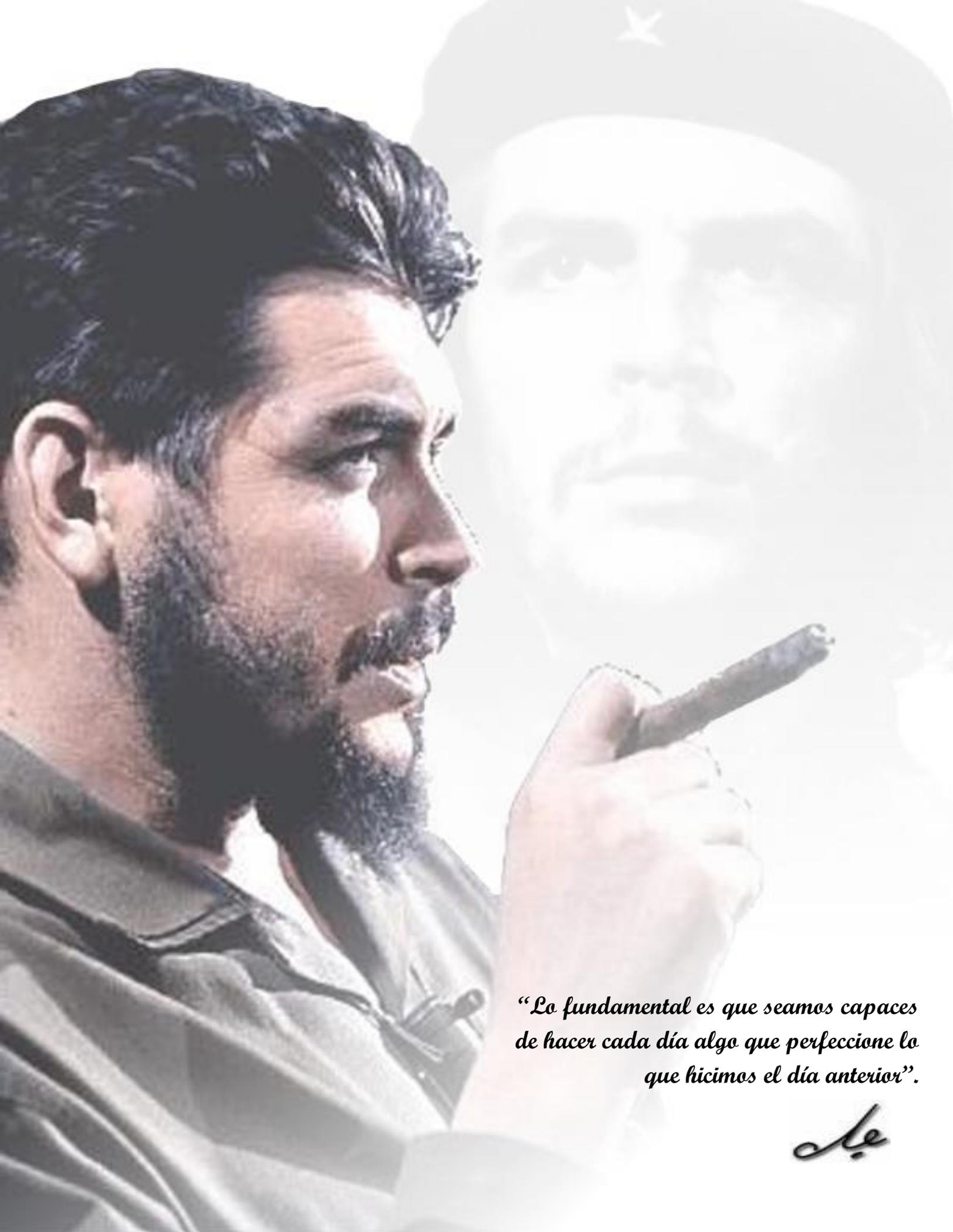
Firma del Tutor

**Ing. Sergio René Vazquez
Rodriguez**

Firma del Tutor

Ing. Mara Morales Alfonso

Firma del Tutor



“Lo fundamental es que seamos capaces de hacer cada día algo que perfeccione lo que hicimos el día anterior”.

Le

Agradecimientos generales

Queremos agradecer:

Primeramente, a nuestros tutores el MSc. Osiris Perez Moya, el Ing. Sergio René Vazquez Rodriguez y la Ing. Mara Morales Alfonso, por toda la ayuda brindada para la realización de este trabajo de diploma y por tener tanta paciencia y dedicación.

A todos los amigos y profes que construyeron nuestro día a día en la universidad y contribuyeron a nuestra formación como persona y como profesional.

A todos los que de una forma u otra contribuyeron a la realización de este trabajo.

A Fidel y a la Revolución Cubana por darnos la oportunidad de convertirnos en profesionales.

A todos muchas gracias

Agradecimientos

De Yudaimy

Mi más sincero agradecimiento:

A mis padres, porque nunca será suficiente lo que haga por ellos. Porque son mi guía, mi apoyo, por siempre confiar en mí, porque todo lo que soy se lo debo a ellos y porque gracias a ellos estoy hoy aquí.

A Luis Daniel, mi Niño, por estar aquí a mi lado durante en estos últimos 3 años en la Universidad, por soportarme como soy, por brindarme su apoyo en todo momento, en los buenos y en los malos momentos y por enseñarme que solo porque alguien no te ama de la forma que quieras, no significa que no te ame con todo lo que puede, porque hay personas que nos aman, pero no saben demostrarlo.

A mis abuelos, por todo su amor, comprensión, por darme las fuerzas de seguir adelante y por estar siempre cuando los necesito.

A mi hermano que quiero mucho.

A toda mi familia por educarme y apoyarme siempre.

A mi otra familia en la universidad, mi papá Jose, mi mamá Yare, mis hermanas Diana, Leydi, y sobrina Mayi, gracias a ustedes pasé momentos que siempre recordaré.

A todos esos amigos que he conocido durante estos 5 años de la carrera, llevo lo mejor de cada uno de ustedes, gracias por su ayuda.

A mis compañeros de grupo en estos años porque juntos compartimos momentos inolvidables y en especial a Ripoll, por escucharme cuando lo he necesitado.

A mi compañero de tesis, por compartir la realización de este trabajo. Por confiar en mí y tenerme presente. Gracias por todo.

A los que han formado y forman parte de mi vida y siempre se han preocupado por mí.

Muchas gracias a todos, los quiero mucho.

Agradecimientos

De Ariel

A mi mamá, a mi hermano y a mi papá un “superagradecimiento”, por ser las personas más importantes de mi vida, por todo su apoyo incondicional y por creer siempre en mí.

A Malena por haber cargado conmigo en estos 5 años, estando ahí para lo que necesitaba, y estar siempre a mi lado.

A todas las personas que hicieron esto posible, a mis amigos de la Universidad que han estado desde primer año conmigo y los que se fueron incorporando, en general quiero agradecerle a todo el grupo 7 desde el 1107 hasta el 1507, en especial a Israel, Pedro, Sergio, Yander, Guillermo, Addiel, Rosalina, Alisnay, Saeret.

A mis amigos del barrio, a mi amigo Erick que siempre estuvo ahí para escucharme y darme los mejores consejos, a mis amigos Arturo, Muñí, Magdiel y Piqui, gracias por hacer de los fines de semana un momento de relajación y que se me olvidaran todos los problemas de la escuela.

A mi compañera de tesis, ya que sin ella esto no hubiese sido posible, gracias por tener paciencia y confiar en mí.

En general gracias a todos los que intervinieron de una forma u otra en este trabajo, gracias por todo.

Dedicatoria

De Yudaimy

Difícil es fijar en el poco espacio que brinda un papel lo mucho que varias personas ocupan en nuestras vidas. Este trabajo va dedicado:

A mi mamá Haydeé que es la razón por la que vivo, por apoyarme en cada momento de mi vida. A mi papá Anel por amarme tanto, por darme las fuerzas de seguir adelante. Gracias por todo, gracias por no exigir, por saber guiarme y sobre todo por confiar en mí, ustedes son lo que más quiero en el mundo.

A mi hermano y a mis abuelos por apoyarme, gracias por estar siempre preocupados por mí.

Los quiero mucho.

De Ariel

Le quiero dedicar este trabajo a mi familia por siempre estar presentes en los momentos malos y buenos, principalmente a mi mamá, papá y hermano que siempre fueron una fuente de inspiración para mí, ya que siempre quise seguir sus pasos y convertirme en profesional como ustedes.

Resumen

Dentro del Sistema de Gestión de Investigaciones de la Universidad de las Ciencias Informáticas se encuentra el módulo Estructura Investigativa, con el mismo se pretende facilitar la gestión de la estructura investigativa que tiene la Universidad, la cual se encarga de gestionar las líneas, grupos y proyectos de investigación. Actualmente el proceso de gestión de la estructura investigativa se realiza de forma manual utilizando herramientas ofimáticas, provocando un engorroso procesamiento de datos así como complejidad en la búsqueda de información por la diversidad de fuentes y archivos a consultar. La información es enviada a niveles superiores a través de correo electrónico, por lo que en todo momento la información no se encuentra actualizada. Para dar solución a la problemática planteada se desarrolló un módulo, el cual se llevó a cabo utilizando las herramientas y tecnologías establecidas por el Centro de Informatización Universitaria (CENIA). Se utilizó como marco de trabajo GUUD, guiado por el proceso de desarrollo con enfoque ágil basado en el segundo nivel de CMMI. Con el desarrollo del módulo se mejoró el proceso referente a la gestión de la estructura investigativa que se realiza en la Universidad. Una vez implementado el módulo se realizaron pruebas funcionales y se corrigieron los errores detectados, garantizando de esta forma que todas las funcionalidades se comporten de la manera correcta.

Palabras clave: estructura investigativa, grupo, línea, proyecto, sistema de gestión.

Índice

Resumen	V
Introducción	1
Capítulo 1: Fundamentación teórica de la investigación	6
1.1 Introducción	6
1.2 Marco conceptual de la investigación	6
1.3 Análisis de soluciones existentes.....	8
1.3.1 Sistemas en el ámbito internacional.....	8
1.3.2 Sistemas en el ámbito nacional	10
1.4 Proceso de desarrollo de software.....	11
1.5 Entorno de desarrollo.....	13
1.5.1 Modelado de procesos.....	14
1.5.2 Lenguajes de desarrollo.....	14
1.5.3 Herramientas de desarrollo.....	15
1.6 Conclusiones parciales	20
Capítulo 2: Análisis y diseño del sistema	21
2.1 Introducción	21
2.2 Descripción del proceso de negocio	21
2.2.1 Descripción del proceso Estructura investigativa	22
2.2.2 Descripción del sub-proceso Crear grupo de investigación.....	23
2.2.3 Descripción del sub-proceso Crear proyecto de investigación	26
2.2.4 Descripción del sub-proceso Grupo de investigación priorizado	26
2.2.5 Descripción del sub-proceso Grupo de investigación no priorizado	27
2.2.6 Descripción del sub-proceso Asignar personas a un proyecto de investigación.....	27
2.3 Propuesta de solución	27
2.4 Técnicas de obtención de requisitos	29
2.5 Definición de los requisitos funcionales	30

2.5.1	Reglas del negocio	32
2.6	Especificación de requisitos funcionales	33
2.7	Definición de los requisitos no funcionales.....	35
2.8	Arquitectura del sistema	36
2.8.1	Estilo arquitectónico.....	36
2.8.2	Patrón arquitectónico.....	36
2.9	Patrones de diseño.....	38
2.9.1	Patrones de diseño GoF	38
2.9.2	Patrón de diseño GRASP	39
2.10	Modelo de despliegue.....	40
2.11	Modelo de datos	41
2.12	Conclusiones parciales	44
Capítulo 3:	Implementación y pruebas.....	45
3.1	Introducción.....	45
3.2	Técnicas de programación.....	45
3.2.1	Programación Orientada a Objetos.....	45
3.3	Estándar de codificación.....	45
3.4	Tratamiento de errores	47
3.5	Validación de los requisitos funcionales.....	47
3.5.1	Técnicas de validación.....	47
3.6	Seguridad	48
3.7	Proceso de pruebas.....	48
3.7.1	Pruebas de integración	53
3.7.2	Resultados de las pruebas	54
3.8	Conclusiones parciales	55
Conclusiones	56
Recomendaciones	57
Bibliografía referenciada	58

Bibliografía consultada.....	61
Glosario de términos.....	65
Anexo 1: Sub-procesos del modelo de proceso de negocio.....	67
Anexo 2: Descripción de los requisitos funcionales.....	73
Anexo 3: Guía de preguntas	81
Anexo 4: Modelo físico y lógico de datos	82
Anexo 5: Descripción de las tablas de la base de datos.....	84

Índice de figuras

Figura 1: Proceso Estructura investigativa.	24
Figura 2: Proceso Estructura investigativa (continuación).	25
Figura 3: Integración del Sistema de Gestión de Investigaciones.	29
Figura 4: Modelo Cliente-Servidor.....	36
Figura 5: Patrón Modelo-Vista-Controlador en G.U.U.D.....	37
Figura 6: Modelo de despliegue.	41
Figura 7: Modelo físico.....	42
Figura 8: Modelo físico (continuación).....	43
Figura 9: Resultado de las pruebas.....	55
Figura 10: Proceso Estructura investigativa (completo).....	67
Figura 11: Sub-proceso Crear grupo de investigación.	68
Figura 12: Sub-proceso Crear proyecto de investigación.	69
Figura 13: Sub-proceso Grupo de investigación priorizado.	70
Figura 14: Sub-proceso Grupo de investigación no priorizado.	71
Figura 15: Sub-proceso Asignar personas a un proyecto de investigación.....	72
Figura 16: Modelo físico de datos.	82
Figura 17: Modelo lógico de datos.	83

Índice de tablas

Tabla 1: Requisitos funcionales del sistema.....	30
Tabla 2: Especificación de requisitos funcionales.	33
Tabla 3: Requisitos no funcionales del sistema.....	35
Tabla 4: Caso de prueba RF5_Crear clasificación de línea.....	50
Tabla 5: Prueba de integración módulo Personal.....	53
Tabla 6: Prueba de integración módulo Estructura y composición.	53
Tabla 7: Prueba de integración módulo Traza.....	53
Tabla 8: Prueba de integración módulo Seguridad.....	54
Tabla 9: Nomenclatura de las estructuras.	84
Tabla 10: Tablas de datos.....	84
Tabla 11: Tablas de nomencladores.	85
Tabla 12: Tablas de relación.....	85
Tabla 13: Tablas del núcleo.	86



Introducción

En Cuba, el sistema educacional constituye un elemento de gran importancia, distinguiéndose por el continuo perfeccionamiento para garantizar los más altos niveles de formación de los profesionales graduados en las universidades. La educación superior tiene un rol significativo en el sistema educacional ya que, provee los graduados universitarios que el país necesita y aporta la mayor parte de la educación continua y de postgrado.

Las universidades en el país se caracterizan por presentar una fuerte relación con respecto a los procesos de docencia-extensión-investigación, empleando del modo más eficiente el potencial científico de los estudiantes y profesores. Esta relación se realiza con el objetivo de elevar el nivel y la calidad de la docencia universitaria, la formación de los profesionales a graduar, y además, promover la cultura general integral como expresión de crecimiento personal.

La investigación científica es una actividad fundamental en el modelo de educación superior cubano, como está definido en la política educacional aprobada en 1976 con la creación del Ministerio de Educación Superior (MES). Por otro lado, su inserción en el sistema empresarial, con el objetivo de lograr productos competitivos que satisfagan las necesidades de la economía y la sociedad cubana, aparece como uno de los principales lineamientos al VI Congreso del Partido Comunista de Cuba (1).

La investigación representa una parte importante de la misión de la universidad, y se encarga de ir transformando, en forma paulatina, la universidad en un centro de investigación científica, donde sus tareas se desarrollan cotidianamente vinculando la masa estudiantil y profesional.

La Universidad de las Ciencias Informáticas (UCI) es una institución innovadora que cuenta con una infraestructura productiva centrada en la producción de software, atendiendo las demandas de diversos sectores, por ejemplo, salud, educación, entre otros; tanto nacionales como para las exportaciones, y además, atender las demandas propias de la Universidad.

En la UCI se realizan investigaciones científicas, que representan un área fundamental para el desarrollo de la misma, y a su vez, juegan un papel decisivo en la creación, difusión y aplicación del conocimiento. El modelo educacional en la Universidad enlaza la producción a los procesos de docencia-extensión-investigación, donde los estudiantes vinculados a los proyectos productivos, trabajan en la producción de software y realizan investigaciones científicas en la misma medida en que van desarrollando sus estudios. Las actividades de investigación, en gran medida se subordinan al esfuerzo productivo y son parte de la formación de los profesionales a graduarse en la universidad.



La investigación científica en la institución contribuye significativamente a la superación profesional y formación académica de su claustro, así como mejorar su competencia como profesores y favorecer la calidad del proceso de aprendizaje de los estudiantes, mediante su incorporación al trabajo científico vinculado al proceso de desarrollo e innovación (2).

La gestión de la investigación en la Universidad se rige por las políticas que define el Ministerio de Ciencia Tecnología y Medio Ambiente (CITMA), organismo encargado de dirigir, ejecutar y controlar la política del Estado y del Gobierno en la actividad científica y tecnológica, la política ambiental y de uso pacífico de la energía nuclear, propiciando su integración coherente para contribuir al desarrollo sostenible del país (3).

La UCI cuenta con una Dirección de Investigaciones (DI) que desde sus inicios trabaja incansablemente por la organización y el buen desempeño de la actividad científica del centro. En la actualidad, 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 (4).

La DI se encarga de la organización de los trabajos científicos que se realizan en el centro, así como de llevar la relación de los premios otorgados en los distintos eventos en que participan estudiantes y profesores de la Universidad. También se ocupa de la gestión de la estructura investigativa con la que cuenta la institución, es decir, las líneas, grupos y proyectos de investigación donde se desarrollan las temáticas en las que se basan las investigaciones. La DI ha intentado informatizar el proceso de gestión de la estructura investigativa; sin embargo no se ha logrado porque:

- ✓ Se realizó un análisis parcial del problema existente con relación a la estructura investigativa, ya que para realizar la gestión de la misma, es necesario que se encuentre vinculado con algunos procesos de la misma área y estos procesos se encuentran de forma aislados.
- ✓ Actualmente, los datos de las líneas, grupos y proyectos de investigación son gestionados de forma manual o mediante el uso de herramientas ofimáticas, de esto resulta un gasto innecesario de materiales de oficina. Además, se pierde tiempo en la búsqueda de los datos.
- ✓ No se valida la ejecución completa del proceso de revisión, aceptación y registro de la estructura investigativa, el cual es muy complejo.
- ✓ La documentación gestionada se almacena en ordenadores personales. Esto trae como consecuencias que se pueda perder total o parcialmente la información en caso de ocurrir cualquier falla técnica en el ordenador, debido a que no se realizan salvadas de seguridad.
- ✓ La gestión de reportes y búsqueda de datos se hace también compleja por la diversidad de fuentes y archivos a consultar.
- ✓ El proceso tarda más debido a que la comunicación de los líderes científicos con la DI sobre el estado de la propuesta se realiza mediante correo electrónico o llamadas telefónicas.



- ✓ El control de las actividades de los grupos y proyectos de investigación se realiza con herramientas ofimáticas y utilizando el correo electrónico. Esto trae como consecuencia un difícil y engorroso procesamiento de datos y que se cometan errores; por lo que en determinados momentos se acude a informaciones desactualizadas para generar nuevos informes.

Teniendo en cuenta lo anteriormente planteado se define como **problema a resolver**: ¿cómo facilitar la gestión de la estructura investigativa de la Universidad de las Ciencias Informáticas?

El **objeto de estudio** de la presente investigación es la gestión de estructuras organizativas. De ahí que el **campo de acción** quede enmarcado en la gestión de la estructura investigativa en la Universidad de las Ciencias Informáticas.

Se define como **objetivo general**: desarrollar una solución informática con tecnologías libres que facilite la gestión de la estructura investigativa mediante su inserción en el Sistema de Gestión de Investigaciones de la Universidad de las Ciencias Informáticas.

Durante la investigación se defiende la idea de que con el desarrollo de una solución informática con tecnologías libres, se facilitará la organización del proceso de gestión de la estructura investigativa en la Universidad de las Ciencias Informáticas.

Para el cumplimiento del objetivo general se trazaron las siguientes **tareas de investigación**:

1. Desarrollo del basamento teórico sobre la gestión de estructuras investigativas.
2. Análisis de sistemas que incluyan funcionalidades relacionadas con la gestión de estructuras investigativas.
3. Análisis del proceso de desarrollo de software para la guía del desarrollo.
4. Análisis de las características de diseño del Sistema de Gestión de Investigaciones (SGI) y el Sistema de Gestión Universitaria (SGU).
5. Análisis de las tecnologías, herramientas y lenguajes de programación para el desarrollo de las funcionalidades.
6. Aplicación de técnicas de recopilación de información a los proveedores de requisitos.
7. Especificación de los requerimientos funcionales y no funcionales que el sistema debe cumplir.
8. Creación de los prototipos para las funcionalidades de la herramienta.
9. Creación de casos de prueba funcionales del sistema.
10. Creación de la base de datos.
11. Implementación de los requisitos descritos.
12. Integración de las funcionalidades implementadas en el SGI.
13. Ejecución de pruebas funcionales del sistema.
14. Ejecución de pruebas de integración con el SGU.

Con el cumplimiento de estas tareas se espera obtener los siguientes resultados:



- ✓ Especificación de los procesos y requisitos necesarios para la gestión que realiza la DI de la información relacionada con la estructura investigativa.
- ✓ Facilidad en el control de grupos y proyectos de investigación.
- ✓ Facilidad en la obtención de datos para la creación de estructuras investigativas, pues algunos se tomarán directamente del SGU.
- ✓ Posibilidad de consulta rápida sobre el estado en la gestión de líneas, grupos y proyectos de investigación.
- ✓ Completitud de los datos, pues se gestionarán directamente desde la fuente.
- ✓ Almacenamiento histórico de datos relacionados con la estructura investigativa.
- ✓ Proporcionar los datos necesarios para la generación de reportes.
- ✓ Proveer información necesaria para la gestión de indicadores cuantitativos de la Universidad.

En el desarrollo de la investigación se emplearán los siguientes métodos científicos:

Métodos teóricos:

Histórico Lógico: permitió realizar un estudio profundo y un mejor análisis de la evolución de diferentes teorías y herramientas empleadas para la construcción de sistemas, mediante la búsqueda de diferentes bibliografías.

Análítico Sintético: permitió realizar un análisis de documentos, libros, artículos sobre técnicas, métodos y herramientas existentes para el proceso de gestión de estructura investigativa.

Métodos empíricos:

Observación: permitió hacer un análisis del comportamiento de algunas herramientas que apoyan el proceso de gestión de estructura investigativa.

El desarrollo de la presente investigación está compuesto por una introducción, un resumen, un índice general, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía, glosario de términos y anexos. A continuación se resume el contenido de los tres capítulos:

Capítulo 1: Fundamentación teórica de la investigación, se presentan los elementos teóricos que sirven de base a la investigación. Se detallan las tecnologías, metodologías, herramientas y lenguajes a utilizar. Además, se realiza un estudio del estado del arte de los sistemas homólogos.

Capítulo 2: Análisis y diseño del sistema, se realiza un estudio de los procesos de gestión de la estructura investigativa en la DI y se modelan dichos procesos que serán automatizados. Se identifican los requisitos funcionales y no funcionales del sistema y se hace una descripción de los mismos. Se describe la propuesta de solución y se explica la arquitectura del sistema utilizado. Se muestra el modelo de datos y se hace referencia a los patrones de diseño empleado.



Capítulo 3: Implementación y pruebas, se presenta todo lo relacionado con la implementación del sistema, los estándares utilizados en la implementación y los resultados de las pruebas realizadas al sistema.



Capítulo 1: Fundamentación teórica de la investigación

1.1 Introducción

En este capítulo se abordan los principales conceptos referentes a la presente investigación. Se realiza un estudio del estado del arte relacionado con soluciones informáticas existentes para la gestión de estructuras investigativas tanto a nivel nacional como internacional. Además, se estudian y describen las tecnologías, metodologías, herramientas y lenguajes que se emplean para el desarrollo de la solución informática.

1.2 Marco conceptual de la investigación

¿Qué es una estructura?

A través del término estructura se pueden referir diversas cuestiones. Por un lado, por estructura se entiende, la distribución y orden de las partes más importantes que componen un todo. Pero por otro lado, se puede hablar de un sistema de conceptos coherentes que se encuentran enlazados entre sí con el objetivo de precisar la esencia del objeto de estudio, es decir, tanto la realidad como el lenguaje que se habla, tendrán su propia estructura.

En muchísimos ámbitos también usan el término estructura para referir su funcionamiento, disposición o materia de estudio como en las ciencias sociales, las ciencias aplicadas, la astronomía, la ingeniería, las matemáticas, entre otras (5).

Por tanto, se define que una estructura es el marco en torno al cual un grupo se organiza.

¿Qué es una estructura organizativa?

Toda empresa cuenta, en forma explícita o implícita, con una cierta jerarquía y atribuciones asignadas a los miembros que componen la misma. En consecuencia, se puede definir que la estructura organizativa de una empresa es una estructura que representa el esquema de jerarquización y división de las funciones que realizan los miembros componentes de la misma. Es decir, es la dirección de la organización, la cual les informa a los participantes cómo está formada la misma y cómo funciona (6).

Estructura investigativa

Una estructura investigativa es una estructura dirigida fundamentalmente a la organización de la investigación dentro de una empresa, o sea, como se organizan las actividades científicas dentro de la institución. El trabajo científico en la Universidad está organizada fundamentalmente por una estructura



investigativa, la cual está compuesta por las *líneas, grupos y proyectos de investigación* con los que cuenta la institución.

¿Qué es una línea de investigación?

Una línea de investigación es un eje temático mono o interdisciplinario en el que confluyen actividades de investigación realizadas por uno o más grupos de investigación que tengan resultados visibles en su producción académica y en la formación de recursos humanos mediante el desarrollo de trabajos finales o tesis (7).

Las líneas de investigación son el núcleo de investigación que desarrolla proyectos en una misma dirección y deben propiciar el fortalecimiento de los programas de formación académica de la universidad en sus diferentes niveles, especialmente la maestría y el doctorado.

Una línea de investigación surge de una problemática válida, aún no resuelta, o resuelta parcialmente en el interior de un área de estudio específico en cualquier disciplina. Generalmente debe ser de carácter interdisciplinaria.

¿Qué es un grupo de investigación?

El Departamento Administrativo de Ciencia, Tecnología e Innovación (COLCIENCIAS) define al grupo de investigación científica o tecnológica como el conjunto de personas que se reúnen para realizar investigaciones en una temática dada, formulan uno o varios problemas de su interés, trazan un plan estratégico de largo o mediano plazo para trabajar en él y producir resultados de conocimiento sobre el tema en cuestión. Un grupo existe siempre y cuando demuestre producción de resultados tangibles y verificables, fruto de proyectos y otras actividades de investigación convenientemente expresadas en un plan de acción debidamente formalizado (8).

Por tanto se define que un grupo de investigación es la unidad básica para la realización de la investigación formada por un conjunto de profesores e investigadores de la universidad y de becarios de doctorado o postgrado de la institución, así como estudiantes vinculados a la producción que tiene temáticas de investigación definidas, con continuidad en el tiempo y con el objetivo de alcanzar resultados, los cuales pueden ser publicaciones, patentes, programas informáticos, desarrollo de proyectos, consultorías, entre otros (9). En los grupos de investigación se hace uso del trabajo en equipo y el deseo de estudiar problemas de manera sistémica, donde la labor es organizada en torno a las líneas comunes de actividad y coordinadas al menos por un investigador responsable.

¿Qué es un proyecto de investigación?



Según lo establece el Departamento Administrativo de Ciencia, Tecnología e Innovación, un proyecto de investigación se refiere a un conjunto articulado y coherente de actividades orientadas a alcanzar uno o varios objetivos relacionados con la generación, adaptación o aplicación creativa de conocimiento. Para ello se sigue una metodología definida que prevé el logro de determinados resultados bajo condiciones limitadas de recursos y tiempo, especificados en un presupuesto y en un cronograma (8).

Por otro lado se define que un proyecto de investigación es un conjunto de acciones no repetitivas, únicas, orientadas a alcanzar uno o varios objetivos siguiendo una metodología, de duración determinada, formalmente organizadas, que utilizan personas, empresas, máquinas, herramientas, materiales o dinero necesario para llevar a cabo las actividades que mediante la aplicación del método científico, procura obtener información relevante, para entender, verificar, corregir y aplicar el conocimiento (9).

Se asume en la presente investigación que un proyecto de investigación es el desarrollo de un conjunto de actividades científicas y tecnológicas organizadas de manera sistemática mediante un plan de trabajo, el cual orienta las acciones que debe realizar el grupo de investigación por el cual está compuesto dicho proyecto para dar cumplimiento a los objetivos definidos.

1.3 Análisis de soluciones existentes

Para el desarrollo de un nuevo sistema es necesario estudiar los antecedentes del mismo desarrollados tanto a nivel nacional como internacional. Es por ello que se realizó un estudio sobre algunos Sistemas de Gestión de Información fundamentalmente en los que gestionan estructuras organizativas, viendo en particular si alguno gestionan estructuras investigativas para así determinar algunas características que cubran las necesidades descritas y ver si pueden ser adaptadas en la UCI. En caso contrario, este estudio permitirá analizar todos los aspectos que puedan ser de utilidad para el desarrollo de una propuesta de solución, ajustada a las necesidades de la institución.

1.3.1 Sistemas en el ámbito internacional

NetOffice es una aplicación web basada en herramientas de gestión de proyectos que posee utilidades que se necesitan para administrar proyectos de negocios. Esta aplicación permite gestionar y compartir todo tipo de información sobre tareas, equipos, proyecto. Está desarrollado en el lenguaje PHP y utiliza como base de datos MySQL para almacenar la información. Diseñada especialmente para el trabajo en grupo, es intuitivo y de fácil uso. Algunas características que presenta:

- ✓ Colaboración de los clientes.
- ✓ Administración de usuarios.
- ✓ Múltiples niveles de acceso.
- ✓ Permite publicar tareas y proyectos.



- ✓ Seguimiento en tiempo real.
- ✓ Permite publicar los sitios de los proyectos de los clientes (10).

NetOffice permite el manejo de las diferentes informaciones relacionadas con la publicación de tareas y proyectos. Su mayor inconveniente es que no presenta funcionalidades relacionadas con el proceso de revisión, aceptación y registro de las estructuras.

eGroupWare es un producto que incorpora un conjunto de herramientas para facilitar el trabajo en grupo. Incluye la gestión de calendario, contactos y tareas pendientes. Incluye además, funciones de gestor de proyectos sencillo, gestor de ficheros y base de datos de conocimientos. Presenta una interfaz web agradable y fácil de usar, es una solución para gestionar calendarios individuales y compartidos de los grupos de cualquier organización. Algunas funciones que incluye eGroupWare son las siguientes:

- ✓ Calendario (programación de horarios de grupos, de recursos y de contactos).
- ✓ Gestor de contactos con base de datos en SQL o LDAP.
- ✓ Cliente de correo integrado del tipo *webmail* usando el protocolo IMAP.
- ✓ Infolog, una aplicación para tareas y notas.
- ✓ Gestor de proyectos integrado con las otras aplicaciones.
- ✓ Gestor de recursos (inventario).
- ✓ Gestor de archivos.
- ✓ Sistema de autor de web con listas de control de acceso.
- ✓ Seguimiento de proyecto integrado con el gestor de proyectos.
- ✓ Seguimiento de errores (11).

Este producto no presenta funcionalidades relacionadas con la gestión de las líneas de investigación.

Cynertia es un sistema español encargado en desplegar los proyectos de forma exitosa, basándose en metodologías ágiles como SCRUM y generales como PMBoX, ayudando a los clientes a gestionar sus proyectos en los siguientes aspectos:

- ✓ Planificación y seguimiento del equipo de proyecto.
- ✓ Control de la producción: recursos, equipos y clientes.

Cuenta con los servicios de CMMI para la gestión de proyectos. Las herramientas que utilizan para la gestión de los mismos son seleccionadas por los clientes, ejemplo: *Redmine, Atlassian Jira, Microsoft Project Server 2010*.

Con estas herramientas se garantiza la planificación ágil de los proyectos, la gestión de cambios y peticiones, la integración con terceras herramientas y también se realizan adaptaciones a las necesidades del cliente (12).



El sistema cuenta con diferentes características que se asemejan a las de la investigación presente, como son la gestión, planificación y seguimiento del equipo de proyecto, sin embargo no cuenta con la gestión de líneas y grupos de investigación que son funcionalidades necesarias para la solución informática que se desea desarrollar, solo realiza la gestión de los proyectos y las tareas que se le asignan a cada uno.

Secretaría de Investigación y Vinculación (SIV), es un sistema argentino creado para garantizar el impulso y el aseguramiento de la máxima calidad de la investigación. Tiene como principales funciones:

- ✓ Ejecutar la política de investigación de la universidad, promoviendo su fortalecimiento.
- ✓ Gestionar los medios necesarios para la ejecución de las políticas.
- ✓ Coordinar y facilitar las actividades que realicen las distintas unidades académicas.
- ✓ Promover el establecimiento de relaciones académicas y científicas con centros de investigaciones de universidades (13).

El sistema cuenta con las funciones de presentar un proyecto de investigación, evaluación de proyectos y las convocatorias presentes.

Este sistema no realiza la gestión de los grupos de investigación ni de las líneas, sólo realizan algunas funciones relacionadas con los proyectos.

1.3.2 Sistemas en el ámbito nacional

GESPRO es una herramienta utilizada y desarrollada en la UCI para la gestión de proyectos. Se encarga de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, tiempo y coste definidos. Esta herramienta es un paquete para la gestión de proyectos, comercializable desde las empresas comercializadoras asociadas a la Universidad (14).

El sistema cuenta con funcionalidades que pueden ser utilizadas en la presente investigación, tales como:

- ✓ Gestión de proyectos en la organización.
- ✓ Gestión documental en proyectos informáticos.
- ✓ Seminario de proyecto.
- ✓ Gestión de recursos humanos.
- ✓ Gestión de costos.

El estudio de esta herramienta presenta grandes características, a pesar de que el sistema no se puede implantar en el SGU, con esta herramienta no se pueden gestionar las líneas y grupos de investigación.

Dirección de Investigaciones es un portal web en la que se apoya la DI en la UCI, en sentido general facilita la divulgación de información. Entre sus principales funcionalidades se destacan:



- ✓ Brindar toda la información referente a la política científica de la universidad, líneas de investigación, doctorados y procedimientos.
- ✓ Publicar documentación útil para realizar una publicación o documentación de proyectos y temas afines, normas de publicación para la serie interna de la universidad y notas informativas.
- ✓ Publicar todos los premios obtenidos en el ámbito universitario o eventos nacionales e internacionales.
- ✓ Publicar informes de balance de ciencia y técnica y las convocatorias de los diversos eventos que se realizan en la universidad, así como sus resultados (4).

Mediante este portal no se gestiona el proceso de revisión, aceptación y registro de la estructura investigativa.

Conclusión sobre el análisis de los sistemas homólogos

Después de haber realizado este estudio, cabe destacar que no se encontró ningún sistema que realice la gestión de las estructuras investigativas, sólo se gestionan algunos de sus elementos como los grupos y proyectos; y de las líneas, sólo se publican informaciones relacionadas con las mismas pero no cómo son gestionadas, punto fundamental para el desarrollo de la solución informática. Esta una de las causas principales por lo que ninguno de estos sistema se adopta como solución porque no resuelven los problemas que existen actualmente en la DI para la gestión de las estructura investigativas.

Su utilización en la Universidad no es posible, porque estos sistemas no cumplen con todas los elementos propuestos para la gestión de estructuras investigativas ya que no validan la ejecución completa del proceso de revisión, aceptación y registro de la estructura investigativa. Además, la gestión de líneas y grupos de investigación son indispensables para la gestión de proyectos y en estos sistemas no se realizan, sólo se gestionan los proyectos sin vínculo alguno con los demás elementos de la estructura.

Otra de las causas por lo que no se adoptan como solución es que estos sistemas no pueden ser integrados al SGU ya que las tecnologías utilizadas en los mismos no coinciden con las establecidas por el centro para los sistemas de gestión de información basados en tecnologías web.

Estos sistemas no cumplen con determinadas funcionalidades que son esenciales para el módulo a desarrollar como el proceso de revisión, aceptación y registro de los elementos de la estructura investigativa, sin embargo, sí se tienen en cuenta la gestión de proyectos y la gestión de tareas.

1.4 Proceso de desarrollo de software

Un **proceso de desarrollo de software** tiene como propósito la producción de un software que reúna los requisitos del cliente. Es un conjunto de actividades y resultados que elaboran un software, los cuales son



llevados a cabo por los ingenieros. No existe un proceso de software universal que sea efectivo para todos los contextos de proyectos de desarrollo.

Existen cuatro actividades fundamentales que son comunes para todos los procesos de desarrollo de software, estas actividades son:

1. **Especificación del software:** los clientes e ingenieros definen las funcionalidades y restricciones operacionales que debe cumplir el software.
2. **Diseño e implementación del software:** se diseña y construye el software de acuerdo a la especificación.
3. **Validación:** se debe validar el software, para asegurar que cumpla con lo que el cliente quiere.
4. **Evolución:** el software debe evolucionar, para adaptarlo a las necesidades del cliente y el mercado (15).

Un proceso de desarrollo es ágil cuando el desarrollo de software es incremental ya que se realizan entregas pequeñas de software, con ciclos rápidos; cooperativos porque el cliente y los desarrolladores trabajan juntos constantemente con una cercana comunicación; sencillo dado que el método en sí mismo es fácil de aprender y modificar, bien documentado; y adaptable porque se permite realizar cambios de último momento.

CMMI (*Capability Maturity Model Integration*, en español Modelo de Capacidad y Madurez Integrado), fue desarrollado por el Instituto de Software e Ingeniería, que es un centro de investigación y desarrollo, gestionado por la universidad de Carnegie-Mellon.

CMMI es un modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de administración como de ingeniería de sistemas y software. Está compuesto por 22 áreas de proceso (AP), las que se distribuyen en 5 niveles según las representaciones del modelo (16).

El modelo CMMI define dos tipos de representaciones, una continua y otra por etapas; ambas tienen el mismo contenido pero diferente estructura.

La representación escalonada o por etapas que es la utilizada en la Universidad, se centra en un conjunto de áreas de proceso claves, que son identificadas dentro de ciertos niveles de madurez. Según este modelo la organización no puede alcanzar el siguiente nivel de madurez hasta que no haya alcanzado el nivel previo.

Esta representación hace especial énfasis en el grado de madurez de los procesos, de forma que cada área se asocia a uno de los cinco, que sirven como punto de referencia para conocer el grado de madurez total que posee una organización. Los cinco niveles son: inicial, gestionado, definido, gestionado cuantitativamente y optimizando. Estos cinco niveles reflejan el hecho de que CMMI es un modelo para la



mejora de la capacidad de las organizaciones de software. Las prioridades en el modelo no están dirigidas hacia proyectos individuales sino a procesos que aporten valor a la organización en su conjunto.

Proceso de desarrollo con enfoque ágil y nivel 2 de CMMI

La UCI, no ajena a los problemas presentes en la industria de software y en aras de lograr mayores beneficios, se encuentra inmersa en un proceso de mejora de las tareas que se desarrollan como parte de la construcción de sistemas informáticos. Para este proceso de mejora de software se utiliza el modelo CMMI, buscando con ello aumentar la madurez y capacidad de la organización en la producción de software.

El proceso de mejora que se desarrolla actualmente en la universidad está encaminado a que se alcance una certificación internacional del nivel dos del modelo CMMI y tiene como objetivo la implementación de los procesos necesarios para cubrir las siete áreas de procesos de este nivel. Estas áreas son: Planeación del Proyecto (PP), Monitoreo y Control de Proyecto (PMC), Administración de Acuerdos con proveedores (SAM), Medición y Análisis (MA), Aseguramiento de la calidad para procesos y productos (PPQA), Administración de la Configuración (CM) y Administración de Requisito (REQM). Garantizando con ello la reducción de varios de los problemas que afectan la producción de software en la universidad (17).

Con el cumplimiento de este objetivo, la Universidad se convertiría en la primera institución en el país en alcanzar el nivel dos del modelo CMMI. Esencialmente lo que se pretende con este nivel es conseguir que en los proyectos haya una gestión de los requisitos y que los procesos estén planeados, ejecutados, medidos y controlados.

El Centro de Informatización Universitaria (CENIA) lleva a cabo la producción de sistemas informáticos. Para ello se hace necesario utilizar un proceso de desarrollo ágil con el objetivo de facilitar la construcción rápida de las aplicaciones. Ello se debe a las demandas de nuevas aplicaciones y a la necesidad, por parte de los clientes, de ser usadas con inmediatez (17).

1.5 Entorno de desarrollo

Para darle solución al problema planteado en la investigación es necesario estudiar los elementos que van a conformar el entorno de trabajo de la misma. A la hora de realizar este estudio se tuvo en cuenta que la solución informática que se pretende desarrollar va integrada al SGU, por lo que se realizó un estudio del entorno de desarrollo que este sistema utiliza. Además, se tienen en cuenta que las herramientas estudiadas y analizadas a continuación, para el desarrollo de la solución, son las definidas por el CENIA.



1.5.1 Modelado de procesos

El modelado de procesos es utilizado para capturar, documentar o rediseñar procesos de negocio. Cuando un proceso es modelado, pueden apreciarse con facilidad las interrelaciones existentes entre distintas actividades, analizarse cada actividad, así como identificar los subprocessos comprendidos. De igual forma, los problemas pueden ponerse de manifiesto claramente dando la oportunidad de acciones de mejora.

En la actualidad existen un gran número de lenguajes, notaciones y herramientas de software que han sido diseñadas para la modelación de los procesos (18). Para la modelación de procesos de negocio se utilizará la notación BPMN.

Notación BPMN

La notación BPMN (*Business Process Modeling Notation*, en español Notación de Modelado de Procesos de Negocio) es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. Esta notación define un tipo de diagrama de procesos de negocio que se utiliza para modelar procesos de negocio, así como los sub-procesos y tareas que lo componen (18).

Ventajas

- ✓ Permite modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento a todas las personas de una organización.
- ✓ Está diseñada para modelar procesos manuales, automáticos, físicos o virtuales.
- ✓ Es capaz de representar procesos complejos.
- ✓ Es independiente de cualquier metodología de modelado de procesos.

1.5.2 Lenguajes de desarrollo

Son un conjunto de signos y reglas que permiten la comunicación con un ordenador. Con la evolución de la informática han surgido los lenguajes de modelado y de programación que facilitan el trabajo a la hora de desarrollar un producto.

1.5.2.1 Lenguaje de modelado

UML (versión 2.0)

UML (*Unified Modeling Language*, en español Lenguaje Unificado de Modelado) es un lenguaje de modelado de sistemas de software orientado a objetos. Se usa para especificar, visualizar, construir y documentar los artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre tales sistemas (19).



1.5.2.2 Lenguajes de programación

Un **lenguaje de programación** es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un programa informático. Un lenguaje de programación permite a un programador especificar de manera precisa: sobre qué datos una computadora debe operar, cómo deben ser almacenados y transmitidos estos datos y, qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural (20).

PHP (versión 5.3)

PHP (*Hypertext Pre-processor*, en español Pre-procesador de Hipertextos), es un lenguaje de programación de código abierto e interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de diferentes tipos de programas incluyendo aplicaciones con interfaz gráfica. Como característica muy importante, se puede decir que puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Además, permite la conexión a diferentes tipos de servidores de bases de datos tales como: MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server y SQLite. Este lenguaje también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (Linux o Mac OS X) y Microsoft Windows (21).

JavaScript (versión 1.3.2)

JavaScript es un lenguaje interpretado orientado a las páginas web, con una sintaxis semejante a la del lenguaje Java. Es fácil de aprender y usar. Además, es un lenguaje de programación bastante sencillo del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Está pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con solo un poco de práctica (22).

1.5.3 Herramientas de desarrollo

Las herramientas de desarrollo son aquellos programas o aplicaciones que tengan cierta importancia en el desarrollo de un programa.



1.5.3.1 Entorno integrado de desarrollo «IDE»

IDE (*Integrated Development Environment*, en español Entorno Integrado de Desarrollo), es un conjunto de herramientas que ha sido creado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Un IDE provee un marco de trabajo amigable para la mayoría de los lenguajes de programación y en algunos casos puede funcionar como un sistema en tiempo de ejecución en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto (23).

NetBeans (versión 7.0.1)

Es una aplicación de código abierto diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas. Hace uso de la tecnología Java y dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones y creación de aplicaciones compatibles con teléfonos móviles. Soporta algunos lenguajes de programación como son C/C++, Ruby y PHP, posibilitando el desarrollo de aplicaciones web con el marco de trabajo CodeIgniter. La plataforma Netbeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos (24).

1.5.3.2 Administrador de base de datos

PgAdmin III (versión 1.14.1)

Es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia de código abierto. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis (25).

1.5.3.3 Herramientas de modelado y diseño

Las herramientas de modelado y diseño se usan para capturar, guardar, rechazar e integrar automáticamente información y diseño de documentación, labores difíciles de lograr con un simple procesador de texto. Existen muchas herramientas que permiten modelar un proceso empleando la notación BPMN, entre ellas se encuentra la herramienta BizAgi.

BizAgi Process Modeler (versión 2.3.05)

BizAgi es una herramienta gratuita, muy poderosa utilizada para diagramar y documentar procesos de manera gráfica, usando la notación estándar BPMN. Los procesos modelados y la documentación correspondiente a este modelo pueden exportarse fácilmente en un archivo de imagen PNG, JPG y BMP,



además, se exportan en un fichero de Word o en un documento de extensión PDF. BizAgi es una solución que permite diseñar, modelar, integrar, automatizar y monitorear los procesos de negocio por medio de un ambiente gráfico sin necesidad de programar. Este software ofrece a las organizaciones una completa plataforma de automatización de procesos diseñada para apoyar la transformación empresarial. Reduce el tiempo para mercado de nuevas ideas y estrategias de negocio, y facilita el mejoramiento continuo de sus procesos (26).

Es capaz de generar un documento Word con el diagrama general, los sub-diagramas, las figuras, un listado ordenado de las actividades y flujos con las propiedades, lo que permite obtener el documento de procesos automáticamente ahorrando tiempo.

ForeUI (versión 1.4)

ForeUI es una poderosa herramienta para crear prototipos estáticos o interactivos para el software o sitio web que se desea construir, con la función de arrastrar y soltar. Muchas organizaciones e individuos la están usando como herramienta principal de diseño de interacción. Además, con esta herramienta se puede definir el comportamiento de cada elemento en el prototipo y ejecutar la simulación en el navegador web. Incluso se pueden crear elementos propios para acelerar los prototipos de desarrollo futuros.

Esta herramienta admite varios temas de la interfaz de usuario para la creación de prototipos. Además, el prototipo se puede exportar como imágenes, PDF y archivos HTML5, o ejecutar la simulación interactiva en su navegador web (27).

Visual Paradigm (versión 8.0)

Es una herramienta CASE para apoyar al proceso de desarrollo de software. Soporta estándares de modelado tales como: Lenguaje Unificado de Modelado (UML), Diagramas de Flujo de Datos (DFD). Esta herramienta soporta y facilita el trabajo de los equipos de desarrollo durante la captura de requisitos, planeación del software, ingeniería de código, modelamiento de clases, modelamiento de datos (28). Además, tiene la capacidad de integrarse a entornos de desarrollo como el NetBeans y es una herramienta multiplataforma.

1.5.3.4 Marco de trabajo

Un marco de trabajo, se puede considerar como una aplicación genérica incompleta y configurable, a la se le pueden añadir las últimas piezas para construir una aplicación concreta. Su genialidad consiste en que simplifica y acelera considerablemente el proceso de desarrollo de una aplicación, ya que automatiza algunos de los patrones utilizados para resolver las tareas más comunes, mediante el encapsulamiento de operaciones complejas en instrucciones sencillas. El marco de trabajo a utilizar en el desarrollo de la



aplicación es el Sistema Integral de Gestión Universitaria, Universidad Digital y Gestión Documental (GUUD) en su versión 1.0.

GUUD es un marco de trabajo para desarrollar aplicaciones web haciendo uso de PHP propuesto por el CENIA. El mismo está conformado por los marcos de trabajo *CodeIgniter* en su versión 1.7.3 y *jQuery* 1.3.2 en una sola infraestructura, razón por la cual posee las mismas características que estos marcos de trabajo, con el objetivo de permitir a los desarrolladores crear proyectos con mayor facilidad. Se incluyen además, en esta integración, un conjunto de novedades o mejoras y algunas modificaciones hechas específicamente al *CodeIgniter*. También la creación de *plugins* y componentes de interfaz de usuario en *jQuery*. Por otro lado, presenta adaptaciones que facilitan el trabajo con módulos en las aplicaciones desarrolladas con este marco de trabajo. A continuación se muestran algunas novedades que incorpora este marco de trabajo GUUD.

Del lado del cliente: se modificaron una serie de *widgets* presentes en la base del *jquery-ui* para ser utilizados de interfaz de usuarios como por ejemplo: *date*, *tab* (ambos son interfaces de los *widgets* de mismo nombre de *jquery-ui*) y *popup* (interfaz de diálogo de *jquery-ui*).

Además, se implementaron nuevos entre los que se encuentran: *attach*, *menu*, *message*, *tooltip*, *form* el cual es construido de la unión del *form* de jQuery el cual se utiliza para el envío de formularios AJAX y el *validate* utilizado para validar formularios. También se implementó el *grid* que utiliza como *plugin* el *jqgrid*; el *multiselect* para hacer selecciones múltiples, el *navbar* para el uso de barras de navegación, el *tree* para la creación de árboles jerárquicos, y el *graph* que utiliza la librería *highchart*. Se implementó un *plugin* a jQuery para el manejo de espacios de nombre e internacionalización y también se implementaron funciones comunes para todo el sistema, contenidas en los archivos *core.js* y *common.js*, entre las que se destacan: *loadIn*, *getDataJson*, *createSelect*, *isArray*, *isFunction*, *site_url*.

Del lado del servidor: se le incorporó la programación orientada a aspectos. Se le agregó manejo de excepciones y mensajes, también se le implementó el IoC (*inversion of control*) para la interacción entre los módulos del sistema. Se le añadió la característica de la modularidad, o sea, que una aplicación pueda dividirse en módulos. Además se añadieron los *plugins export_pi* que permite exportar a los formatos: pdf, csv y xls) e *import_pi* que permite importar desde archivos en formatos csv o xls. Se añadieron, modificaron y extendieron los *helpers* o asistentes entre los que se encuentran:

- ✓ **Añadidos:** son plantillas o *template* que brinda la posibilidad de ser usadas, característica que no posee *CodeIgniter*. Para esto se añaden los componentes *grid*, *json*, la librería *template* y los *assets*. Estos dos últimos son utilizados para la integración en las vistas de *JavaScript*, *css*, imágenes y el *template*.
- ✓ **Modificados:** *form*, *array*, *security* (25).



jQuery

Es un nuevo tipo de biblioteca o marco de trabajo de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX al sistema. Al igual que otras librerías, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta librería se logran grandes resultados en menos tiempo y espacio. Tiene como ventaja que permite cambiar el contenido de una página web sin necesidad de recargarla, utilizando DOM y AJAX de manera extremadamente sencilla gracias a su sintaxis (25).

CodeIgniter

Es un marco de trabajo para el desarrollo de aplicaciones web usando PHP. Tiene como objetivo permitir el desarrollo de proyectos más rápido que si se escribiera código desde cero. Provee una rica colección de librerías para las tareas necesarias más comunes, así como una interfaz sencilla y una estructura lógica para acceder a esas librerías. Permite concentrarse en el desarrollo del proyecto en cuestión, minimizando la cantidad de código necesario para realizar las tareas. Usa el patrón de diseño arquitectónico Modelo-Vista-Controlador como paradigma de arquitectura de desarrollo, el cual es separada en tres capas distintas: la representación de datos, el interfaz de usuario y el controlador de eventos respectivamente (25).

1.5.3.5 Sistema gestor de base de datos

Un sistema gestor de bases de datos (SGBD) es una colección de programas con el objetivo de servir de interfaz entre la base de datos, el usuario y las aplicaciones. Permite la definición, construcción y manipulación de bases de datos para diversas aplicaciones (29).

En la actualidad existe gran cantidad de gestores, cada uno con características distintas, así como ventajas y desventajas a la hora de ser usado. El centro define por política la utilización de PostgreSQL.

PostgreSQL (versión 8.4.1)

Es un SGBD relacional orientado a objetos y libre, publicado bajo licencia de código abierto. Soporta distintos tipos de datos; además del soporte para los tipos base, también para datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes. También permite la creación de tipos propios.

Incorpora funciones de diversa índole: manejo de fechas y orientación a operaciones con redes. Soporta el uso de índices, reglas y vistas. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. Permite la declaración de funciones propias, así como la definición de disparadores. Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este



gestor de bases de datos se le incluye entre los gestores objeto-relacionales. Sirve como herramienta gráfica de diseño y administración de base de datos. Cuenta con herramientas visuales para la elaboración de consultas y para la administración efectiva de la seguridad.

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo (25).

1.5.3.6 Servidor web

Un servidor web puede ser tanto un ordenador de grandes dimensiones y capacidad como un programa informático que utiliza el protocolo de comunicaciones HTTP para recibir las peticiones de información de un programa cliente (navegador) en el ordenador del usuario. Una vez interpretada la solicitud que el servidor cliente le hace, el servidor web recupera la información solicitada de sus carpetas y la envía al programa cliente para que esta información sea visualizada (30).

Apache (versión 2.2)

Apache es un servidor web de código abierto, uno de los más usados por la amplia gama de opciones que brinda. Al ser una tecnología de código abierto permite realizar modificaciones en el código fuente, corre además en una multitud de sistemas operativos que lo hace prácticamente universal. Es configurable en la creación y gestión de registro de actividades de un sistema y permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor (31).

1.6 Conclusiones parciales

Después de haber analizado los conceptos relacionados con el campo de acción, se pudo definir el marco conceptual asociado a la investigación. El estudio y análisis de los sistemas homólogos demostró que tanto en el ámbito internacional como nacional, no existe un sistema para la gestión de las líneas, grupos y proyectos de investigación que erradique los problemas planteados anteriormente. Se estudiaron las políticas de desarrollo de software definidas en el CENIA y que serán utilizadas en la elaboración de la solución informática, abordando de las mismas sus principales características.



Capítulo 2: Análisis y diseño del sistema

2.1 Introducción

En el presente capítulo se hace una propuesta general del sistema a desarrollar, después de haber realizado un análisis del proceso de negocio que tiene lugar en la DI de la Universidad. Se obtienen como artefactos el modelo de proceso, los requerimientos funcionales y no funcionales correspondientes con el proceso de desarrollo ágil. Igualmente, se incluirán la arquitectura y los patrones de diseño utilizados, así como la descripción del modelo de datos del sistema. Se identificarán las reglas que deben cumplir el negocio y las técnicas utilizadas para el levantamiento de requisitos.

2.2 Descripción del proceso de negocio

La DI es la estructura encargada de promover las políticas científicas de la Universidad. Además, entre otras funciones, tiene la misión de llevar el control de la vinculación de los trabajadores y estudiantes con las líneas de investigación, los grupos de investigación y a los diferentes proyectos productivos existentes en la Universidad.

Cada línea de investigación tendrá un líder científico que será el que responda por todas las tareas de investigaciones y la estrategia a seguir para dar cumplimiento a las mismas. Además, la DI es la encargada de controlar los grupos y proyectos de investigación.

Los grupos de investigación podrán estar compuestos por personal docente, especialistas e investigadores, así como estudiantes vinculados a la producción. Los miembros de un grupo de investigación podrán pertenecer a diferentes facultades, departamentos, centros, u otras entidades de la Universidad o externas a la misma.

Roles que intervienen en el proceso

Investigador: solicita la creación de una nueva línea de investigación. Puede ser cualquier persona de la Universidad, un estudiante, un profesor, un decano y otras personas.

Líder científico: encargado de presentar, revisar y corregir las propuestas de líneas, grupos y proyectos de investigación.

Responsable de investigación del área: es el encargado de revisar y decidir si son aceptadas o no las propuestas de líneas de investigación presentadas por los líderes científicos en las áreas. Puede realizarse por el vicedecano de investigación de la facultad o el responsable de investigación del área.



Comisión científica del área: es una comisión conformada por los principales dirigentes del área para revisar y decidir el estado de las propuestas de líneas, grupos y proyectos de investigación presentadas por los responsables de investigaciones del área. Puede ser el decano de la facultad, vicedecano de investigación y demás directivos del área y trabajadores con alto prestigio en las actividades investigativas.

Dirección de investigaciones: es la encargada de gestionar y controlar todo lo referente a la estructura investigativa de las áreas de la Universidad.

Consejo científico UCI: principal órgano asesor de la universidad en la rama científica. Se responsabiliza de trazar las principales estrategias del Sistema de Ciencia, Tecnología e Innovación, además de emitir consideraciones con respecto a la evaluación de líneas, grupos y proyectos de investigaciones, y otros temas de interés para la Universidad. Está constituido por el rector y demás miembros de la dirección de la Universidad.

La gestión de la estructura investigativa cuenta con un proceso general Estructura investigativa (ver figura 1 y figura 2) y cinco sub-procesos fundamentales (ver anexo 1), los cuales son descritos a continuación.

La descripción de las actividades presente en el proceso de negocio se realiza en el artefacto CENIA_INV_MEI_0115_MPN_BPM.doc, el cual se encuentra en el expediente de proyecto del SGI.

2.2.1 Descripción del proceso Estructura investigativa

El proceso inicia cuando un investigador propone una nueva línea de investigación al líder científico del área, quien debe completar los datos necesarios para entonces presentarla al responsable de investigación para que revise la propuesta. Si la revisión determina que no está elaborada correctamente se deben corregir los errores señalados y presentarla nuevamente; en caso que sea rechazada se notifica al líder científico, y si se acepta se presenta a la comisión científica del área para que valore la propuesta. Si es aceptada la propuesta, se debe analizar si es una línea de investigación priorizada, en caso de no serlo se crea y se registra en el área; y en caso de ser priorizada se presenta la propuesta a la DI para que determine el estado de la misma. En el caso de que sea aceptada, se presenta la propuesta al consejo científico de la UCI y si es aceptada, la DI la registra y se establece como una línea priorizada de la Universidad.

Una vez registrada la línea de investigación, tanto en el área como en la DI, se debe asignar las personas que conformarán el grupo de trabajo de esta línea y el líder científico crea el plan de trabajo. Luego se debe verificar si se desea asignar un grupo de investigación a la línea, en el caso negativo finaliza el proceso; y en el caso positivo, se verifica si se crea un nuevo grupo de investigación o si se asigna a alguno que ya esté creado, luego se asocian al grupo de investigación las temáticas específicas.



Después de asociar el grupo de investigación a una temática se verifica si se desea asociar a un proyecto de investigación. En el caso que no se decide asociar, se procede a controlar el grupo en base a su plan de trabajo, verificando el cumplimiento del mismo. Luego, se pregunta si se desea archivar el grupo de investigación, si se decide que sí, se archiva el grupo de investigación y entonces se termina el proceso. En el caso que no se decida archivar el grupo y no desea asociarlo a un proyecto, entonces se vuelve a controlar el grupo.

En el caso que se desee asociar a un proyecto, se define si se asocia a un proyecto existente o si se crea un nuevo proyecto de investigación, para luego asociar el grupo a dicho proyecto. Luego, se realizan controles a este proyecto de investigación verificando el cumplimiento de su cronograma. Después se archiva el proyecto y finaliza el proceso.

Una vez creado el plan de trabajo del grupo de investigación, cada tres meses se realiza un balance de ejecución del plan de trabajo. Si a los tres meses después de ejecutado el balance, el plan de trabajo sigue en ejecución se vuelve a realizar el balance, de lo contrario, finaliza el proceso.

Los líderes o coordinadores de línea, junto con el responsable administrativo deben realizar una presentación del estado de cumplimiento de la línea en una sesión conjunta del Consejo Científico y Productivo, cada seis meses.

2.2.2 Descripción del sub-proceso Crear grupo de investigación

El sub-proceso inicia cuando el líder científico elabora una propuesta de grupo de investigación para que sea revisada por la comisión científica del área.

Si la comisión científica del área detecta algunos errores en la propuesta, la envía al líder científico para que sean corregidos y presentadas nuevamente. Por otro lado, si determina que la propuesta no es aceptada, se le notifica el rechazo al líder científico que la presentó y finaliza el proceso. En caso que sea aceptada la propuesta, se analizan los recursos y líneas de investigación del área para determinar si es un grupo de investigación priorizado o no priorizado. Luego de crearse el grupo, ya sea priorizado o no, la DI registra el grupo de investigación, finalizando entonces el proceso.

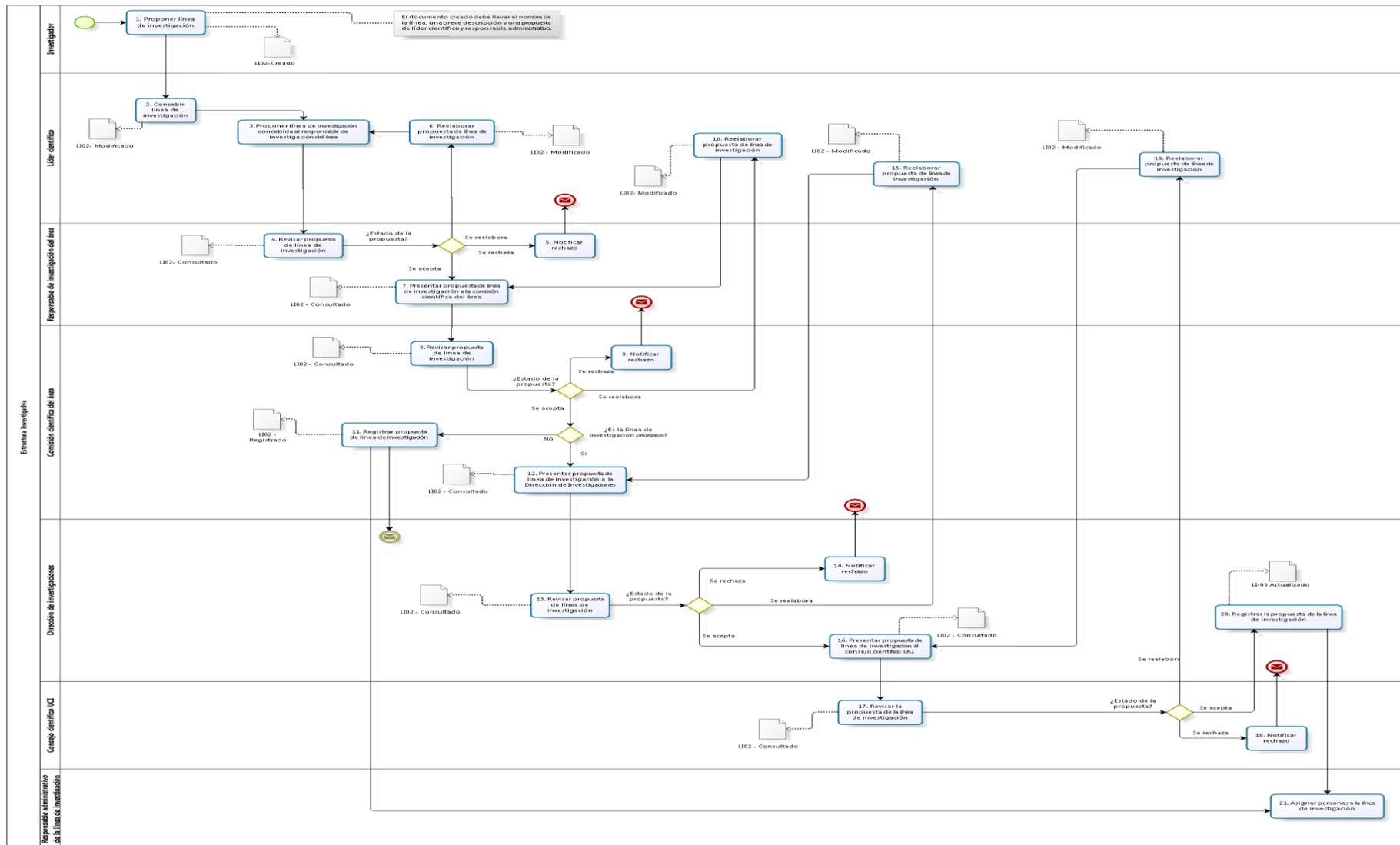


Figura 1: Proceso Estructura investigativa.

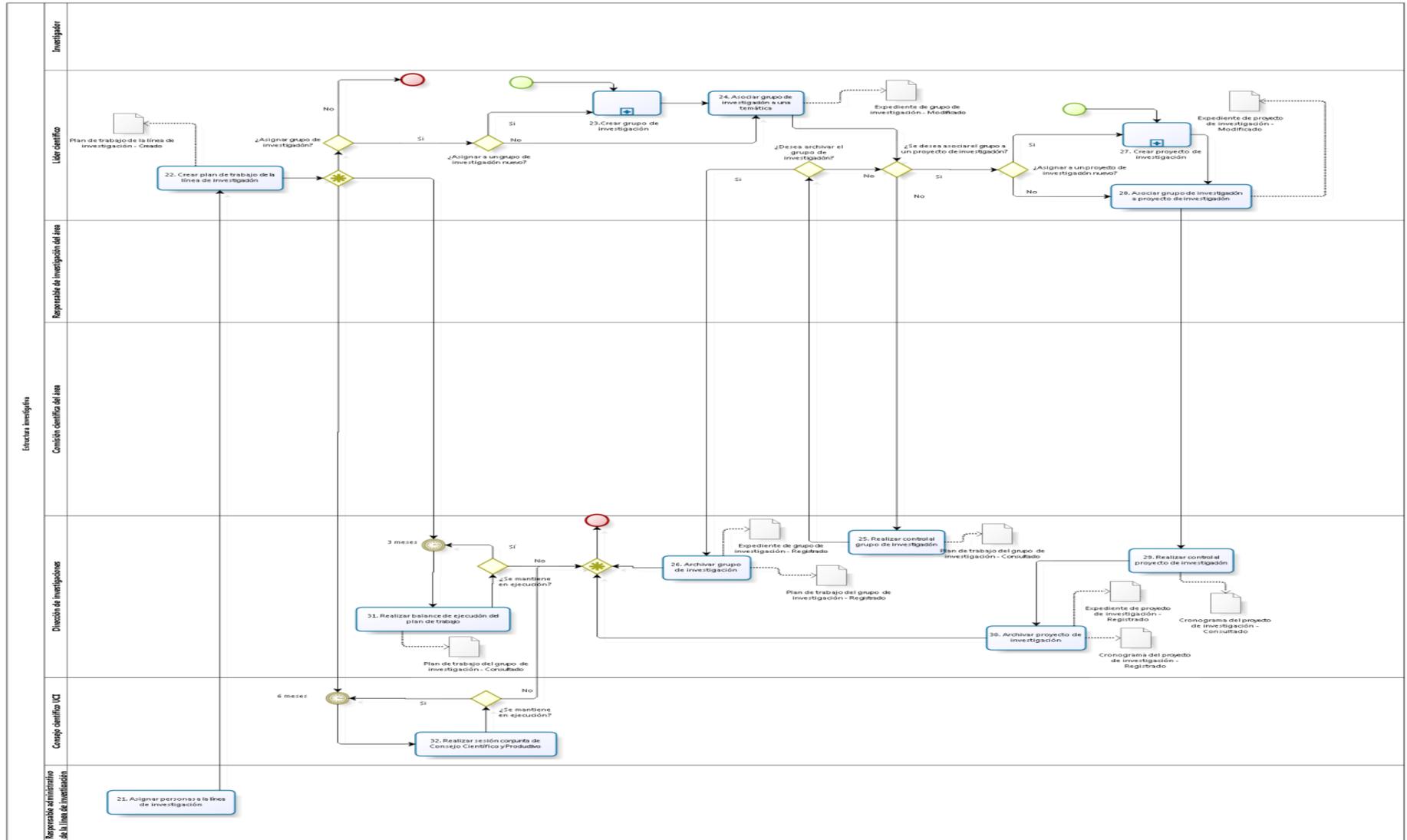


Figura 2: Proceso Estructura investigativa (continuación).



2.2.3 Descripción del sub-proceso Crear proyecto de investigación

El sub-proceso inicia cuando el líder científico elabora una propuesta de proyecto de investigación la cual incluye el nombre del proyecto, una breve descripción del mismo y una propuesta de jefe de proyecto para presentarla a la comisión científica del área.

Si la comisión científica del área rechaza la propuesta se notifica al líder y finaliza el proceso. En caso que se señalen algunos errores, entonces el líder científico debe corregirlos y presentar nuevamente la propuesta. Por otra parte, si es aceptada, se analizan los recursos y líneas de investigación del área para determinar si es un proyecto de investigación priorizado o no priorizado.

Si el proyecto no es priorizado, o sea, que será un proyecto de la misma área donde se desea crear, se registra en el área y se notifica a la DI que se ha creado un nuevo proyecto. Por otra parte, si el proyecto es priorizado, o sea, que necesita recursos de otras áreas de la Universidad, se envía la propuesta al jefe de la comisión evaluadora que revisará la propuesta. Si se determina que la propuesta no está correctamente elaborada, se le notifican los errores al líder científico para que sean corregidos. Si de 0 a 10 días hábiles antes del próximo consejo científico no se ha reelaborado, se rechaza la propuesta y se archiva, finalizando entonces el proceso. En caso que se haya reelaborado en el tiempo establecido vuelve a ser revisada hasta que quede correctamente elaborada, para que luego pase a ser revisada por los evaluadores anónimos asignados, quienes emiten un dictamen general.

Luego la DI envía la propuesta al consejo científico de la UCI donde se determine si es aceptada o rechazada. En caso de ser aceptada, la DI registra el proyecto y lo notifica a la comisión científica del área, de lo contrario, se notifica su rechazo finalizando el proceso.

Luego de registrarse el proyecto de investigación, ya sea en el área donde se realizó la propuesta o en la DI en caso de ser un grupo priorizado, la comisión científica del área puede asignar personas a dicho proyecto y luego estos proyectos serán controlados por la DI quien verificará el cumplimiento del cronograma de trabajo, finalizando entonces el proceso.

2.2.4 Descripción del sub-proceso Grupo de investigación priorizado

El sub-proceso inicia cuando la comisión científica del área presenta la propuesta de un grupo de investigación priorizado para ser revisada por el consejo científico de la UCI. Si la revisión determina que no está elaborada correctamente se deben corregir los errores señalados en la propuesta y presentarla nuevamente; en caso de que sea rechazada se notifica a la comisión científica del área y si se acepta se informa a la DI y se le proponen las personas que formaran parte del grupo de investigación, revisando las propuestas y notificando a la comisión científica del área los recursos humanos que la integrarán.



2.2.5 Descripción del sub-proceso Grupo de investigación no priorizado

El sub-proceso inicia cuando el coordinador del grupo de investigación presenta la propuesta de grupo de investigación no priorizado para ser revisada por la comisión científica del área. Si la comisión revisora determina que no cumple con las pautas definidas, se deben corregir los errores señalados en la propuesta y presentarla nuevamente; en caso de que sea rechazada se notifica al coordinador del grupo de investigación y si se acepta se registra el grupo de investigación, se informa a la DI y finaliza el proceso.

2.2.6 Descripción del sub-proceso Asignar personas a un proyecto de investigación

El jefe del proyecto de investigación presenta una propuesta de investigadores a pertenecer al proyecto de investigación.

En el caso que sea un proyecto priorizado se presenta la propuesta a la DI para que sea valorada. Si en la revisión se determina que es rechazada se notifica al jefe del proyecto; en el caso que se determine que no está elaborada correctamente se deben corregir los errores señalados y presentarla nuevamente; y si es aceptada se registran las personas asociadas al proyecto de investigación y se notifica al jefe del proyecto quien elabora el cronograma de trabajo.

En el caso que sea un proyecto no priorizado se presenta la propuesta a la comisión científica del área para que sea valorada. Si se rechaza la propuesta se notifica al jefe del proyecto; en caso que se señalen algunos errores en la propuesta el jefe del proyecto debe corregirlos y presentarla nuevamente y si es aceptada se registra la propuesta y se notifica a la DI que se han asignado personas al proyecto de investigación y el jefe del proyecto elabora el cronograma de trabajo.

2.3 Propuesta de solución

El estudio realizado permitió visualizar las dificultades existentes en la DI y en sus niveles correspondientes con el manejo y la administración de toda la información que se necesita gestionar, controlar y obtener. Teniendo en cuenta de que no existe un sistema que cumpla con las necesidades actuales de la DI, se propone desarrollar el módulo Estructura Investigativa, el cual aporte soluciones satisfactorias a los problemas existentes sobre la gestión de las estructuras investigativa en la DI.

La solución informática a desarrollar podrá gestionar procesos fundamentales como: gestión de líneas, grupos y proyectos de investigación, a los cuales se les asocian sub-líneas, investigadores, entidades y tareas. A los grupos se le asigna una línea principal y a los proyectos un grupo principal.



Actualmente en la Universidad existe un portal que permite mostrar las investigaciones existentes, pero no existe un sistema que logre la gestión de las mismas, por lo que el sistema a desarrollar podrá ser utilizado por todas las facultades de la Universidad. Una vez desarrollado el sistema, será utilizado para la gestión y el control de la estructura investigativa en los centros de producción. Además, se le integrarán las tareas de investigación, los responsables de cada estructura investigativa y las instituciones asociadas a las mismas.

El módulo será capaz de archivar toda la información referente a la estructura investigativa. Los usuarios del sistema solo tendrán acceso a las interfaces que están autorizados a acceder.

Integración del módulo

El SGI está compuesto por dos módulos: Eventos Externos y Estructura Investigativa, los cuales se relacionan entre sí mediante la base de datos (ver figura 3). A su vez, se encuentra integrado al núcleo del SGU, el cual cuenta con componentes que son horizontales para algunos sistemas, entre los que se encuentran: Personal, Trazas, Seguridad y Estructura y composición. A continuación se detallan estos componentes relacionados al núcleo del SGU.

Personal: se obtienen las personas del núcleo para luego ser asociadas a la estructura investigativa.

Trazas: gestiona todo lo referente a las incidencias de un usuario sobre el sistema, registrando el usuario, la acción realizada y el momento en que se ejecutó. Las acciones de los usuarios sobre el módulo que se propone serán almacenadas por este módulo.

Seguridad: permite la autenticación de los usuarios y gestiona los permisos de estos sobre las funcionalidades del sistema.

Estructura y composición: gestiona la información referente a toda la estructura administrativa y la jerarquía de la institución, así como la asignación de responsabilidades a las estructuras, establecidas por el centro donde se decida implantar el SGU.

Esta integración permite uniformidad en la arquitectura, la centralización de archivos e información y la eliminación de redundancias en el código, puesto que con el uso del IoC cualquier módulo en el SGU puede utilizar las funcionalidades y componentes ya implementados por los sistemas que lo componen.



Figura 3: Integración del Sistema de Gestión de Investigaciones.

2.4 Técnicas de obtención de requisitos

Los requisitos se clasifican en dos tipos: requisitos funcionales y requisitos no funcionales. Los requisitos funcionales describen lo que el sistema debe hacer. Mientras que los requisitos no funcionales son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como: la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento (15).

La obtención de requisitos es el proceso mediante el cual los interesados en un sistema de software descubren, revelan, articulan y entienden sus requisitos. En muchos casos, se requiere tiempo para llegar a especificar claramente lo que el cliente espera de la aplicación informática, por lo que se hace necesario por parte de los analistas el empleo de técnicas que permitan establecer una buena comunicación con los interesados del producto y así lograr la satisfacción del cliente.

Las técnicas aplicadas en esta investigación para la obtención de requisitos son:

- ✓ **Entrevistas:** se llevaron a cabo conversaciones dirigidas por los analistas a clientes y usuarios, con el objetivo de entender el dominio del problema y sus necesidades. Se realizaron preguntas buscando obtener las opiniones de los clientes o usuarios entrevistados. En general estas entrevistas se realizaron de manera informal, tanto personal como en grupo. En el anexo 3 se muestra la guía de preguntas realizadas en la entrevista.



- ✓ **Prototipos:** la realización de prototipos es una técnica moderna para obtener los requerimientos de un sistema informático. Con esta técnica se logró obtener más información acerca del problema y sus posibles soluciones. Además, se obtuvieron los requerimientos preliminares que fueron utilizados para construir una versión inicial del sistema, es decir, un prototipo. Estos prototipos fueron mostrados al cliente, quienes aportaron nuevos requerimientos adicionales. Luego se realizaron modificaciones al diseño del prototipo, según los nuevos requerimientos detectados y se mostraron nuevamente al cliente. Este proceso repetitivo, continúa durante un determinado número de iteraciones o hasta que el producto cumple con las necesidades del negocio.
- ✓ **Sistemas existentes:** se analizaron distintos sistemas ya desarrollados que estén relacionados con el objeto de estudio. Se analizaron las interfaces de usuario, observando el tipo de información que se maneja y cómo es manejada. Algunas funcionalidades e interfaces sirvieron de base para la propuesta de solución.

2.5 Definición de los requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, es decir, define qué es lo que el sistema debe hacer, así como las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas, para producir salidas (32). A continuación se muestra la tabla de los requisitos funcionales obtenidos.

Tabla 1: Requisitos funcionales del sistema.

Requisito funcional – iteración 1: Gestión de nomencladores	Complejidad
Crear clasificación de línea de investigación	Media
Modificar clasificación de línea de investigación	Media
Ver detalles clasificación de línea de investigación	Baja
Listar las clasificaciones de línea de investigación	Media
Crear sub-línea de la línea de investigación	Media
Modificar sub-línea de la línea de investigación	Media
Ver detalles sub-líneas de la línea de investigación	Baja
Listar las sub-línea de la línea de investigación	Media
Crear recurso material	Media
Modificar recurso material	Media
Ver detalles de recurso material	Baja
Listar los recursos materiales	Media
Crear clasificación de proyecto de investigación.	Media
Modificar clasificación de proyecto de investigación	Media
Ver detalles de clasificación de proyecto de investigación	Baja
Listar las clasificaciones de proyecto de investigación	Media
Crear problema	Media



Modificar problema	Media
Ver detalles de problema	Baja
Listar problemas	Media
Crear clasificación de resultado	Media
Modificar clasificación de resultado	Media
Ver detalles de clasificación de resultado	Baja
Listar clasificación de resultados	Media
Requisito funcional – iteración 2: Gestión de línea de investigación	Complejidad
Crear línea de investigación	Alta
Insertar investigadores a la línea de investigación	Media
Listar los recursos humanos de la línea de investigación	Media
Eliminar investigadores de la línea de investigación	Media
Modificar línea de investigación	Media
Archivar línea de investigación	Media
Desarchivar línea de investigación	Media
Listar líneas de investigación	Media
Ver detalles generales de la línea de investigación	Media
Valorar propuesta de líneas de investigación	Alta
Revisar propuesta de línea de investigación	Alta
Reelaborar propuesta de línea de investigación	Alta
Registrar línea de investigación	Media
Listar los grupos de investigación asociados a la línea de investigación	Media
Requisito funcional – iteración 3: Gestión de grupo de investigación	Complejidad
Crear grupo de investigación	Media
Insertar investigadores al grupo de investigación	Media
Eliminar investigadores del grupo de investigación	Media
Listar los recursos humanos del grupo de investigación	Media
Modificar grupo de investigación	Media
Archivar grupo de investigación	Media
Desarchivar grupo de investigación	Media
Listar grupos de investigación	Media
Ver detalles generales del grupo de investigación	Media
Valorar propuesta de grupo de investigación	Alta
Revisar propuesta de grupo de investigación	Alta
Reelaborar propuesta de grupo de investigación	Alta
Registrar grupo de investigación	Media
Listar los proyectos de investigación asociados al grupo de investigación	Media



Requisito funcional – iteración 4: Gestión de proyecto de investigación	Complejidad
Crear proyecto de investigación	Media
Insertar investigadores al proyecto de investigación	Media
Eliminar investigadores del proyecto de investigación	Media
Listar los recursos humanos del proyecto de investigación	Media
Modificar proyecto de investigación	Media
Archivar proyecto de investigación	Media
Desarchivar proyecto de investigación	Media
Listar proyecto de investigación	Media
Ver detalles generales del proyecto de investigación	Media
Revisar propuesta de proyecto de investigación	Alta
Valorar propuesta de proyecto de investigación	Alta
Reelaborar propuesta de proyecto de investigación	Alta
Registrar proyecto de investigación	Media
Requisito funcional – iteración 5: Gestión de tarea	Complejidad
Insertar tarea	Media
Cambiar estado de una tarea	Alta
Modificar tarea	Media
Archivar tarea	Baja
Desarchivar tarea	Baja
Listar tarea	Media
Ver detalles de tarea	Media
Gestionar macro-proyecto de la línea de investigación	Media
Gestionar plan de trabajo del grupo de investigación	Media
Gestionar plan de trabajo del proyecto de investigación	Media

Durante el análisis fueron identificados un total de 75 requisitos funcionales, de ellos 11 de complejidad alta, 56 de complejidad media y ocho de complejidad baja, los cuales quedaron distribuidos en un plan de cinco iteraciones. El listado de los requisitos funcionales y sus descripciones se encuentran en el anexo 2.

2.5.1 Reglas del negocio

Las reglas del negocio describen operaciones, normas y políticas que deben ser cumplidas. Definen además, restricciones que las organizaciones tienen definidas y son vitales para lograr sus objetivos. Algunas de las reglas que debe cumplir el negocio son:

- ✓ Los miembros de la DI están autorizados a realizar cualquier modificación que estimen conveniente en cualquier elemento de la estructura investigativa.



- ✓ Los elementos de la estructura investigativa podrán ser archivados y desarchivados por la DI en cualquier momento.
- ✓ Una línea de investigación deberá tener un líder científico y un responsable administrativo.
- ✓ Inicialmente la línea de investigación pertenece al área a la que pertenezca su líder científico, es decir, pertenece al área que la haya gestionado.
- ✓ Un grupo de investigación deberá tener un coordinador de grupo de investigación.
- ✓ No podrán existir varias estructuras de investigación con el mismo nombre.
- ✓ El grupo de investigación deberá contar con un mínimo de seis profesores o especialistas de la UCI.
- ✓ Los grupos de investigación, podrán estar compuestos por personal docente, especialistas e investigadores; y asociados, estudiantes de pregrado.
- ✓ Los grupos de investigación deberán estar asociados a líneas de investigación.
- ✓ Los proyectos de investigación deberán estar asociados a grupos de investigación.

2.6 Especificación de requisitos funcionales

Para la especificación de los requisitos se hace uso de la plantilla Especificación de requisitos del software, la cual cuenta con secciones que permiten describir cada requisito funcional para su mejor entendimiento. En la tabla 2 se muestra un ejemplo de la especificación del requisito 29 Listar líneas de investigación.

Podrán encontrarse las demás especificaciones de requisitos en el artefacto Especificación de Requisitos de Software EI.doc en el expediente de proyecto del SGI.

Tabla 2: Especificación de requisitos funcionales.

Nº	Nombre	Pre-condiciones	Complejidad	Prioridad para cliente
RF29_EI	Listar líneas de investigación	El usuario se autentica en el SGU. El usuario selecciona el sistema Investigaciones. El usuario selecciona el módulo "Estructura Investigativa". El usuario debe seleccionar en la barra de menú lateral derecha la opción "Líneas de Investigación".	Media	Alta
Descripción				
<ol style="list-style-type: none"> 1. El sistema muestra un listado de todas las líneas de investigación registradas hasta la fecha y en el área de iconos flotantes las opciones "crear" y "refrescar". 2. Se muestra un campo de búsqueda para buscar por el nombre de la línea de investigación. 3. De cada línea se muestra Nombre, Estado y Líder. 4. El usuario puede filtrar la búsqueda por los campos: Estado, Clasificación. 5. El sistema muestra en el área de iconos internos: <ol style="list-style-type: none"> a. Para el estado <i>Registrada</i>, las opciones "Archivar X", "Modificar X" y "Detalles generales de X" (donde X es el nombre de la línea). 				



- b. Para el estado *Elemento propuesto* la opción “*Completar*”.
 - c. Para los estados *Completada*, *Aceptada por Responsable*, *Aceptada por Comisión área*, *Aceptada por Dirección de investigaciones* y *Aceptada por consejo UCI* la opción “*Revisar línea X*” (donde X es el nombre de la línea).
 - d. Para el estado *Reelaborar por A* (donde A es el rol que revisó la propuesta), la opción “*Reelaborar línea X*” (donde X es el nombre de la línea).
 - e. Para el estado *Rechazada*, la opción “*Detalles generales X*” (donde X es el nombre de la línea).
6. Si el usuario selecciona la opción:
- a. “**Crear**”, ir al RF1_EI.
 - b. “**Completar**”, ir al RF1_EI, escenario 2.
 - c. “**Modificar**”, ir al RF24_EI.
 - d. “**Archivar**”, ir al RF25_EI.
 - e. “**Desarchivar**”, ir al RF26_EI.
 - f. “**Detalles generales**”, ir al RF28_EI.
 - g. “**Revisar línea**”, ir al RF30_EI.
 - h. “**Reelaborar línea**”, ir al RF31_EI.

Prototipo

Líneas de investigación



Filtrar búsqueda:

Cantidad por página

Nombre
Estado
Líder

Bioinformática	Registrada	Rafael Rodríguez Puente	  
Formación del Ingeniero Informático	Elemento propuesto	Yordani Cruz Segura	
Redes de telecomunicaciones	Aceptada por Responsable	Adrian Puentes Quintero	
Seguridad informática y criptografía	Completada	Alejandro Arias Benítez	
Software libre y sistemas de código abierto	Reelaborar por Responsable	Luis Felipe Diaz Barrios	

 
 Página de 2  

Resultados encontrados: 8

Campos	Tipos de Datos	Reglas o Restricciones
Buscar	Text Box	Solo admite valores alfanuméricos de 2 a 50 caracteres. Valor inicial: <i>Nombre línea de investigación</i> .
Filtrar búsqueda	Combo Box Simple	Muestra los criterios para realizar los filtros de búsqueda de las líneas existentes en el sistema. Valor inicial: “ <i>Seleccione</i> ”.
Cantidad por página	Combo Box Simple	Muestra la cantidad de elementos que se pueden mostrar (valores: “5”, “10”, “15”, “20”).



			Valor inicial: "5".
	Paginado	Text Box	Muestra al usuario información adicional de la página en la que se encuentra de un total. Valor inicial: "1".
	Observaciones		

2.7 Definición de los requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, debe pensarse en propiedades que hacen al producto atractivo, usable, rápido, o confiable (15). Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste, como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. En la tabla 3 se describen los requisitos no funcionales que debe cumplir el sistema.

Tabla 3: Requisitos no funcionales del sistema.

Usabilidad	
RNF 1	El módulo debe presentar una interfaz intuitiva que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.
RNF 2	El módulo debe adaptarse a los lenguajes y términos utilizados por los usuarios con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.
RNF 3	El sistema debe presentar un menú lateral mostrando las agrupaciones del módulo, un área de íconos flotantes que permitan el acceso rápido a la información por parte de los usuarios y también deberá presentar íconos internos.
Seguridad	
RNF 4	La seguridad de la base de datos está a nivel de roles, con el fin de mantener la integridad de los datos en función del acceso de cada uno de ellos, trayendo consigo además la protección de la información.
RNF 5	El módulo de Seguridad debe garantizar las políticas de accesibilidad a las diferentes funcionalidades del módulo Estructura Investigativa en dependencia del nivel de autorización que presente un usuario determinado.
Restricciones del diseño	
RNF 6	Servidor de base de datos con PostgreSQL 8.4 o superior bajo el sistema operativo GNU/Linux para servidores.
RNF 7	Servidor de aplicaciones web: Apache 2.2 o superior
RNF 8	Navegador web: Mozilla Firefox 2.3 o superior
Requisitos para la documentación de usuarios en línea	
RNF 9	La documentación asociada al modelo de proceso de negocio, especificación de requisitos, diccionario de datos, casos de pruebas del módulo Estructura Investigativa esté actualizada.
Interfaz	
RNF 10	La interfaz debe ser sencilla, con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo.
Interfaz de hardware	
RNF 11	La comunicación entre el servidor de aplicaciones y la base de datos se lleva a través del protocolo TCP/IP.



RNF 12	La comunicación entre el cliente y el servidor de aplicaciones deberá ser a través del protocolo HTTP/HTTPS
-----------	---

2.8 Arquitectura del sistema

La arquitectura de software es la estructura de un sistema. Contiene conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software. Una correcta definición de la arquitectura permite facilitar el diseño y construcción de los componentes y sus relaciones. La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación y la comunicación entre ellos.

2.8.1 Estilo arquitectónico

Se utiliza como estilo arquitectónico Llamada y Retorno, empleando específicamente la arquitectura Cliente-Servidor (ver figura 4).

Cliente-Servidor es un modelo basado en la idea del servicio, en el que el cliente es un proceso consumidor de servicios y el servidor es un proceso proveedor de servicios. Además esta relación está establecida en función del intercambio de mensajes. Esta es la arquitectura más extendida en la realización de Sistemas distribuidos.

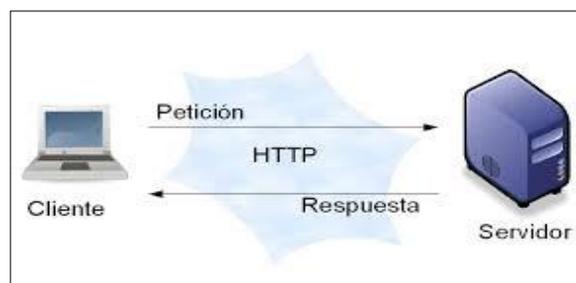


Figura 4: Modelo Cliente-Servidor

Cliente: es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. El cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario, además de acceder a los servicios distribuidos en cualquier parte de una red.

Servidor: es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Usualmente la mayoría del trabajo pesado lo hace el servidor, mientras que los procesos del cliente solo se ocupan de la interacción con el usuario a través de interfaces gráficas. Esto permite distribuir físicamente los procesos y los datos de forma segura, reduciendo grandemente el tráfico de la red.

2.8.2 Patrón arquitectónico

El SGI basa su implementación en el patrón Arquitectura en Capas, específicamente en el patrón MVC (*Model View Controller*, en español Modelo Vista Controlador). Este 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. Es usado frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el sistema de gestión de base de datos y el controlador representa la lógica de negocio.

A su vez, el marco de trabajo GUUD utilizado en el centro implementa este patrón de arquitectura (ver figura 5), el cual propone un modelo de diseño a través del cual deben regirse las aplicaciones que se desarrollen en el mismo.

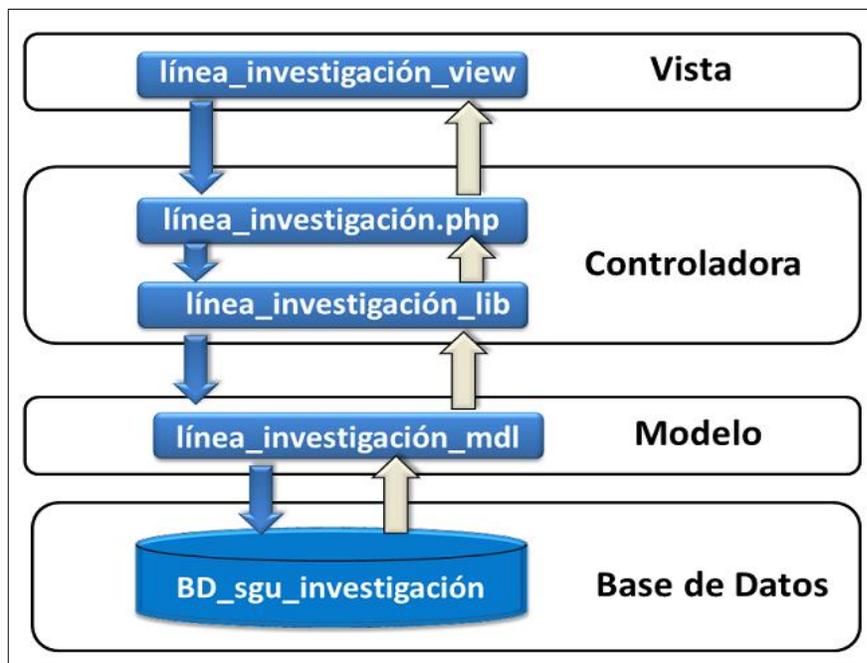


Figura 5: Patrón Modelo-Vista-Controlador en G.U.U.D.

A continuación se describe la utilización de este patrón en el marco de trabajo GUUD para el desarrollo del módulo.

- ✓ **Modelo:** en esta capa se encuentran las clases modelos y las librerías. Las librerías son las encargadas de organizar la lógica del negocio, mientras que las modelos son las que acceden a los datos. Las clases librerías envían los datos a las controladoras, después de obtenerlos de las modelos.
- ✓ **Controlador:** es el intermediario entre la vista y el modelo. Son las clases encargadas de recibir las peticiones del usuario y en vista de satisfacer su demanda, utilizan los métodos que contienen las clases librerías para hacer peticiones de datos a las clases modelos, que son las que tienen acceso a los datos y estas al obtenerlos se los envía a través de las librerías a la controladora, que una vez que los recibe se los entrega a las vistas.



- ✓ **Vista:** son clases que a través de los ficheros JavaScript, estilos CSS y la librería jQuery conforman el código HTML necesario para mostrar la información requerida por el usuario, que obtiene de las clases controladoras.

La estructura de este patrón que se pone de manifiesto, es la siguiente. En el núcleo del sistema existe una carpeta nombrada *base* dentro de la cual aparece la carpeta *application* (aplicación), la cual contiene todos los sistemas por los que se integra el SGU separados cada uno por carpetas. Cada uno de estos sistemas contiene carpetas con los módulos que la componen y dentro de estas, existen cinco carpetas con funciones diferentes.

Por tanto, dentro de la carpeta aplicación se encuentra investigaciones, correspondiente al SGI, en la cual se encuentra estructura_investigativa, compuesta por cinco carpetas, las cuales son:

- ✓ *Config:* incluye la clase *box* donde se define las agrupaciones y funcionalidades que contendrá el módulo.
- ✓ *Controllers:* incluye las clases que contienen las llamadas a los métodos que se encuentran en las librerías obteniendo a través de estas los datos, los organiza y luego son mostrados al usuario.
- ✓ *Libraries:* en esta se encuentran todas las clases librerías por cada uno de las controladoras, estas son las encargadas de la lógica del negocio, sirve de intermediaria entre la modelo y la controladora y para obtener información de diferentes módulos.
- ✓ *Models:* es la encargada de guardar los datos en un medio persistente, contiene las consultas de acceso a datos que encierra una serie de clases con métodos predefinidos para el acceso a datos.
- ✓ *Views:* es la encargada de la presentación de los datos, contiene subcarpetas cada una nombrada con distintas funcionalidades. Incluye a su vez las páginas HTML con las que interactúa el cliente llamadas interfaces.

2.9 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

2.9.1 Patrones de diseño GoF

Los patrones GoF (*Gang of Four*, en español Banda de los Cuatros), describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Permiten



crear grupos de objetos para ayudar a realizar tareas complejas. Algunos de los patrones GoF empleados en el desarrollo del módulo se describen a continuación.

Instancia única (Singleton): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En el módulo todas las clases controladoras son instancias únicas de la clase *My_Controller*.

Mediador (Mediator): define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto. Se aplica en las librerías que funcionan como mediadoras entre las clases controladoras y los modelos o acceso a datos.

Observador (Observer): define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Se aplica en la clase *loader* que es el objeto *load* de las clases controladoras, se encarga de cargar los elementos del marco de trabajo *dígase*, librerías, modelos y se encarga de actualizar la controladora instanciada.

2.9.2 Patrón de diseño GRASP

Los patrones GRASP (*General Responsibility Assignment Software Patterns*, en español Patrones generales de software para asignar responsabilidades) los cuales dan la medida de un refinamiento del diseño. Para el diseño de la solución informática se tuvieron en cuenta los siguientes patrones: experto, creador, bajo acoplamiento, alta cohesión y controlador.

Experto: permite asignar a una clase la información necesaria para cumplir su responsabilidad, de esta forma se asignan las mismas de una forma adecuada y el módulo tiende a ser más fácil de entender, mantener y ampliar, pensando en una futura reutilización de componentes. Se utiliza en las clases librerías, estas cuentan con la información necesaria para desempeñar las responsabilidades sobre los elementos de negocio.

Creador: guía la asignación de responsabilidades de la creación de objetos. Su propósito fundamental es encontrar un creador que deba conectar con el objeto producido en cualquier evento. En la clase *loader* que es el objeto *load* de las clases controladoras, encarga de cargar los elementos del marco de trabajo como librerías, modelos. Es utilizado en el módulo seguridad, en la librería *fabrica_ma_lib*, que se encarga de crear los objetos de los modos de autenticación (*ma*).

Bajo acoplamiento: es el encargado de asignar una responsabilidad de tal modo que no se creen gran cantidad de dependencias de una clase con respecto al resto de las mismas. Así, en caso de producirse una modificación en alguna de ellas, se adquiera la mínima repercusión posible en el resto de las clases,



potenciando la reutilización y disminuyendo la dependencia entre las clases, haciendo que el sistema se vuelva más flexible.

Alta cohesión: plantea que la información que almacena una clase debe de ser coherente y debe estar, en la medida de lo posible, relacionada con la clase. Asigna una responsabilidad de forma tal que la cohesión siga siendo alta. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. Se utilizó para asignar responsabilidades de manera tal que una clase no tenga sobrecarga de funcionalidades y estas estén relacionadas entre sí.

La propia implementación de *CodeIgniter* contiene los patrones bajo acoplamiento y alta cohesión permitiendo el uso de los componentes de forma individual, evidenciando el bajo acoplamiento y así como la dependencia entre ellos o alta cohesión.

Controlador: asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Esto facilita la centralización de actividades (validaciones, seguridad). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Este patrón se evidencia en el uso de las clases controladoras que se implementan con el patrón de arquitectura que establece el marco de trabajo. Es el intermediario entre las capas vistas y librerías, obteniendo a través de las librerías los datos que la capa modelo proporciona y mostrándolo a las vistas garantizando la comunicación entre los eventos externos del sistema en la capa de presentación y los componentes de la capa de negocio.

2.10 Modelo de despliegue

El modelo de despliegue (ver figura 6) provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto. Sobre el modelo de despliegue debe hacerse las siguientes observaciones:

- ✓ Los nodos poseen relaciones que son medios de comunicación entre ellos, tales como TCP/IP, HTTP/HTTPS.
- ✓ Cada nodo representa un recurso de cómputo, puede ser un procesador o un dispositivo hardware.

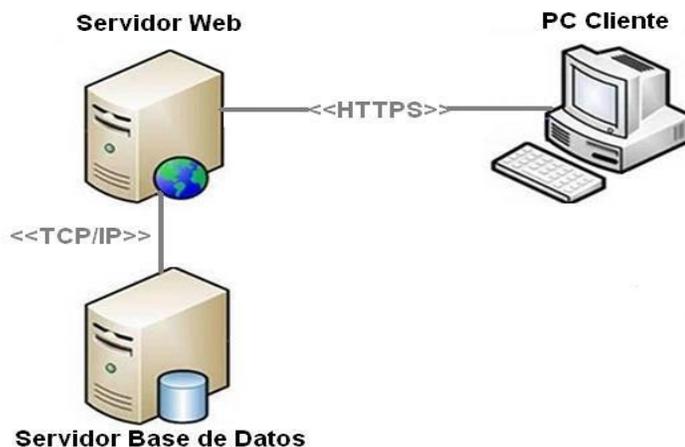


Figura 6: Modelo de despliegue.

PC Cliente: computadora que cuenta con un navegador actualizado y que siga los estándares web (Mozilla Firefox v 2.x o superior). Se recomiendan estaciones de trabajo con sistema operativo GNU/Linux.

Servidor Web: representa una estación donde estará montado el Servidor Apache sobre el cual correrá la aplicación.

Servidor Base de Datos: representa el servidor donde estará el SGBD PostgreSQL que dará respuesta a las peticiones hechas por la aplicación.

2.11 Modelo de datos

Un modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Por lo general, permite describir las estructuras de datos de la base (el tipo de los datos que incluye la base y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregado, borrado, modificación y recuperación de los datos de la base).

Para la construcción del presente sistema se utilizó PostgreSQL como SGBD. La base de datos fue diseñada utilizando la herramienta Visual Paradigm, y cuenta con un total de 30 tablas persistentes que almacenan toda la información del negocio. Se tuvo en cuenta el diseño del modelo lógico y modelo físico de los datos, los cuales se muestran en el anexo 4. Además, en el anexo 5 se incluye una descripción de las tablas de la base de datos. A continuación se muestra el modelo físico de los datos:



Para el diseño del modelo de datos se emplearon los siguientes patrones de base de datos.

Llaves subrogadas: plantea que se genere una llave primaria única para cada entidad, en vez de usar un atributo identificador en el contexto dado. En la solución informática se aplica este patrón ya que cada entidad tiene un atributo único autoincrementar que constituye la llave primaria, esto garantiza que las tablas sean más fáciles de consultar a partir del identificador ya que todos tienen el mismo tipo de dato en cada una de las tablas.

Control de Acceso Basado en Roles (RBAC): permite asignar roles particulares a los usuarios, y a través de los cuales obtienen permisos para ejecutar funciones determinadas en el sistema. En la solución informática se aplica este patrón ya que al usuario se le muestran las opciones a las cuales está autorizado a acceder y podrá realizar cambios en las funcionalidades permitidas teniendo en cuenta el rol que desempeña en el sistema.

Máquinas de estado: representa la ocurrencia del cambio de estado por los que puede atravesar una entidad dada. Un ejemplo de la utilización de este patrón en la solución informática es precisamente el flujo de actividades desde que un investigador realiza la solicitud de un elemento de investigación hasta que termina el proceso dentro del área o en la Universidad.

2.12 Conclusiones parciales

El estudio de los procesos del negocio involucrados en la gestión de la estructura investigativa de la DI permitió identificar el flujo de actividades asociadas a estos. Con la aplicación de las técnicas para identificar requisitos, se detectaron 75 funcionalidades y las características que el sistema debe cumplir. A partir de la especificación de la arquitectura definida en el CENIA, se pudo concebir una propuesta que puede ser integrada al SGU.



Capítulo 3: Implementación y pruebas

3.1 Introducción

Luego de realizar el levantamiento de requisitos en el capítulo anterior así como la validación de los mismos, quedan definidas las funcionalidades que debe poseer el sistema para satisfacer las necesidades del cliente. En el presente capítulo se describen las técnicas de programación utilizadas, así como el estándar de codificación empleado en la realización del software. Se detallan los patrones de diseño empleados en la implementación. Además, se identifican las técnicas utilizadas para la validación de los requisitos y se define la estrategia de pruebas que se realizaron para garantizar el correcto funcionamiento del software.

3.2 Técnicas de programación

Las técnicas de programación constituyen una parte fundamental en el proceso de desarrollo e ingeniería del software dentro del ámbito informático. Cada técnica tiene sus propias características, y distintos métodos de resolución de problemas, así como la implementación de estándares de ciertas compañías o instituciones.

Durante el desarrollo del software se utilizaron como técnicas de programación, la Programación Orientada a Objetos (POO).

3.2.1 Programación Orientada a Objetos

La POO es una técnica que aumenta la velocidad de desarrollo de los programas, gracias a la reutilización de los objetos. El elemento principal de la POO es el objeto, el cual es un conjunto complejo de datos que poseen estructura y forman parte de una organización. Un objeto, contiene los datos bien estructurados y pueden ser visibles o no, dependiendo del programador y las acciones del programa en ese momento. El polimorfismo y la herencia, son unas de sus principales características.

3.3 Estándar de codificación

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación, con vistas a generar un código de alta calidad, es de gran importancia para la calidad del software, y para obtener un buen rendimiento. Además, si se aplica de forma continuada un estándar de codificación bien definido caben muchas posibilidades de que un proyecto de software, se convierta en un sistema fácil de comprender y de mantener.



Indentación, llaves de apertura y cierre y tamaño de líneas: usar una indentación sin tabulaciones, con un equivalente a cuatro espacios, para mantener integridad en las revisiones. El uso de las llaves “{}” será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres, para mantener la legibilidad del código.

Convención de nomenclatura

Variables: se rigen por la nomenclatura *camelCase*. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

Constantes: constantes siempre debe ser todo en mayúsculas, con caracteres de subrayado “_” para separar palabras en caso de nombres compuestos.

Clases: siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con el carácter subrayado “_” y el resto en minúscula.

Funciones: se rigen por la nomenclatura *camelCase*. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

Ficheros: todo siempre en minúscula y en caso de nombres compuestos se usa el carácter subrayado “_”.

- ✓ **Para las vistas:** intuitivo y relacionado con el formulario y/o vista que representa el sufijo *_view*.
- ✓ **Para los modelos:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo *_mdl* o *_base* en caso de ser modelos base.
- ✓ **Para las librerías:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo *_lib*.
- ✓ **Para las controladoras:** con el mismo nombre de la clase que representa.

Estructuras de control: se incluye *if*, *for*, *foreach*, *while*, *switch*, entre las estructuras de control y los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

Si las condiciones son muy largas que sobrepasan el tamaño de las líneas, estas se dividen en varias líneas. En el mejor de los casos cuando la condición es muy extensa, esta se puede dividir en variables y compararlas dentro de la estructuras de control.

Documentación: todos los archivos deben de tener la documentación asociada al mismo.



Buenas prácticas: los valores booleanos y nulos siempre se escriben con mayúscula. Para facilitar la legibilidad del código usar un salto de línea antes de las estructuras de control y definición de las funciones.

3.4 Tratamiento de errores

La informática está muy lejos de ser perfecta y lo demuestran un buen número de mensajes de error que han perseguido y acompañado a los usuarios mientras trabajaban con distintos productos de hardware y software. Para solucionar estos problemas aparece el término tratamiento de errores, elemento que permite identificar, localizar, analizar y eliminar los errores en la implementación de un programa de computadora.

Las aplicaciones web deben poseer un buen mecanismo de tratamiento de errores, enfocándose no solo en las entradas de datos suministradas por el usuario, sino también en cualquier error que pueda ser generado por el comportamiento incorrecto de componentes internos, por ejemplo, errores en las consultas a la base de datos.

En el módulo Estructura Investigativa, los datos introducidos por el usuario son validados tanto del lado del cliente como del lado del servidor. Esto garantiza que cuando se produzca un error en el sistema, este sea capaz de alertar al usuario e impedir que se produzca el envío de datos incorrectos. De esta forma se logra la disponibilidad de la aplicación y la fluidez en la interacción usuario-aplicación.

En el módulo, el usuario introduce los datos mediante las vistas, siendo estos validados en JavaScript. Una vez que se cometa algún error, se muestra un mensaje en rojo encima del campo indicando los valores válidos que deben introducirse. Además se muestran mensajes de confirmación y de información que le indican al usuario la conclusión exitosa de una acción. El tratamiento de estos errores son definiciones que se toman de la arquitectura del SGU.

3.5 Validación de los requisitos funcionales

La validación de requisitos, tiene como misión, demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. Es el proceso de verificarlos en cuanto a validez, consistencia, integridad, realismo y certidumbre.

3.5.1 Técnicas de validación

Revisiones de requerimientos: comprueban consistencia, integridad, verificabilidad, comprensibilidad y rastreabilidad. Proceso manual que involucra a varios lectores, tanto del cliente como del contratista. Puede ser formal o informal. En la revisión formal, el equipo de desarrollo conduce al cliente a través de los requerimientos, explicándole las implicaciones de cada uno.



Construcción de prototipos: representación de aquellos aspectos del software, que serán visibles al usuario. Consiste en construir una maqueta del futuro sistema, a partir de los requisitos recogidos en la especificación.

Generación de casos de pruebas: esta técnica tiene como objetivo comprobar la validez de los requisitos. Consiste en la definición de casos de prueba, que permitan verificar el cumplimiento de los requisitos funcionales.

Se empleó la planilla de Especificación de Requisitos de Software y Criterios para validar requisitos del cliente, definidos en el expediente para los proyectos del CENIA.

3.6 Seguridad

La seguridad en las aplicaciones informáticas es un factor que no debe ser olvidado por los desarrolladores. Con el uso del marco de trabajo que propone el centro se garantiza la seguridad en el módulo haciendo uso de los componentes Traza y Seguridad del núcleo del SGU.

El registro de cada una de las actividades realizadas por el usuario autenticado en el sistema, mediante el almacenamiento de trazas de usuarios, permite mantener un control sobre las acciones realizadas en el sistema.

Se establecen como políticas de seguridad la autenticación de las personas autorizadas con la cuenta del dominio uci.cu. Para garantizar la integridad de la información es utilizado el modelo de control de acceso basado en rol, el cual describe a grupos de usuarios que actúan bajo un conjunto de roles. Por tales motivos fueron definidos varios roles, asignándoles solamente los permisos de acceso a la información de acuerdo al rol que este desempeñe una vez registrado en la aplicación. Entre los roles definidos se encuentran investigador, responsable de área, líder científico, comisión de área, dirección de investigaciones con acceso pleno al sistema, y consejo UCI.

3.7 Proceso de pruebas

Las pruebas de software son actividades en las cuales un sistema o componente es ejecutado bajo condiciones o requerimientos específicos, son un elemento crítico para la garantía de calidad del software y representan una revisión final de las especificaciones del diseño. Verifican además, los resultados de la implementación del sistema y son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un software. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. Las pruebas son aplicadas para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo (32).



Con el objetivo de medir el grado en que el módulo cumple con los requerimientos, se adoptó como nivel de prueba: prueba de sistema, utilizando como tipo de pruebas las pruebas funcionales. Para este caso se utilizó el método de prueba de Caja negra, que se refiere a las pruebas que se llevan a cabo sobre la interfaz del software y que permiten obtener un conjunto de condiciones de entrada, que ejercitan completamente todos los requisitos funcionales del programa.

Una de las técnicas de pruebas que se usan en este método es la partición de equivalencia. Esta divide el campo de entrada de un programa en clases de equivalencia de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran tipos de errores. Los diseños de casos de pruebas permiten guiar el proceso de prueba que se le debe realizar al sistema durante la validación del mismo.

En la tabla 4 se muestra el diseño de caso de prueba que responde al requisito funcional 5. Podrán encontrarse en su totalidad los restantes diseños de casos de pruebas en el expediente de proyecto del SGI.



Tabla 4: Caso de prueba RF5_Crear clasificación de línea.

Escenario	Descripción	Nombre clasificación de línea	Habilitado	Descripción	Respuesta del sistema	Flujo central
EC 5.1 Insertar datos correctamente.	Mediante este escenario se permite insertar o seleccionar los datos correctos para crear una clasificación de línea de investigación	V	V	V	El sistema actualiza el listado y muestra el mensaje <i>"El elemento ha sido creado satisfactoriamente"</i>	<p>El usuario se autentica en el SGU.</p> <p>El sistema muestra la pantalla donde se encuentran los diferentes SGU.</p> <p>El usuario selecciona el sistema Investigaciones.</p> <p>El sistema muestra las diferentes opciones de menú.</p> <p>El usuario selecciona el módulo "Estructura Investigativa".</p> <p>El sistema muestra todas las agrupaciones contenidas del módulo.</p> <p>El usuario selecciona la agrupación "Configuración".</p> <p>El usuario selecciona un bloque de los mostrados "Clasificación de líneas".</p> <p>El sistema muestra un listado con todas las clasificaciones de líneas existentes en el sistema.</p> <p>El usuario selecciona la opción "Crear" en el área de íconos flotantes.</p> <p>El sistema muestra una interfaz para introducir los datos de la nueva clasificación.</p> <p>El usuario introduce los datos correctamente y presiona la opción Aceptar.</p>
		"Líneas de Investigación transversales"	"activo"	"descripción de la clasificación de línea"		
EC 5.2 Cancelar operación	Este escenario permite cancelar la operación de crear una clasificación de línea de investigación	NA	NA	NA	<p>El sistema muestra un mensaje de advertencia <i>"¿Está seguro de realizar la acción?"</i>.</p> <p>- Si el usuario acepta el sistema actualiza y muestra el listado.</p>	<p>El usuario se autentica en el SGU.</p> <p>El sistema muestra la pantalla donde se encuentran los diferentes SGU.</p> <p>El usuario selecciona el sistema Investigaciones.</p> <p>El sistema muestra las diferentes opciones de menú.</p> <p>El usuario selecciona el módulo "Estructura Investigativa".</p> <p>El sistema muestra todas las agrupaciones contenidas del módulo.</p> <p>El usuario selecciona la agrupación "Configuración".</p> <p>El usuario selecciona un bloque de los mostrados "Clasificación de líneas".</p> <p>El sistema muestra un listado con todas las clasificaciones de líneas existentes en el sistema.</p> <p>El usuario selecciona la opción "Crear" en el área de íconos flotantes.</p> <p>El sistema muestra una interfaz para introducir los datos de la nueva clasificación.</p> <p>El usuario introduce o no los datos y presiona la opción Cancelar.</p>



EC 5.3 Insertar datos incorrectamente	Este escenario permite insertar datos incorrectos para crear una clasificación de línea de investigación	I	NA	I	El sistema no le permite continuar entrando caracteres.	El usuario se autentica en el SGU. El sistema muestra la pantalla donde se encuentran los diferentes SGU. El usuario selecciona el sistema Investigaciones. El sistema muestra las diferentes opciones de menú. El usuario selecciona el módulo "Estructura Investigativa". El sistema muestra todas las agrupaciones contenidas del módulo. El usuario selecciona la agrupación "Configuración". El usuario selecciona un bloque de los mostrados "Clasificación de líneas". El sistema muestra un listado con todas las clasificaciones de líneas existentes en el sistema. El usuario selecciona la opción "Crear" en el área de íconos flotantes. El sistema muestra una interfaz para introducir los datos de la nueva clasificación. El usuario introduce los datos incorrectamente.
		"Más de 100 caracteres"		"Más de 1000 caracteres"		
		I	NA	V	El sistema muestra el mensaje en rojo "Entre al menos dos caracteres".	
		"b"		"texto"		
		V	NA	I	El sistema muestra el mensaje en rojo "Entre al menos cinco caracteres".	
"Líneas de investigación específicas (LE)"		"Menos de 5 caracteres"				
EC 5.4 Insertar datos incompletos	Este escenario permite introducir datos incompletos.	I	NA	NA	Si el usuario selecciona el botón Aceptar el sistema muestra un mensaje en rojo: "Campo requerido", asociado al campo que debe ser insertado obligatorio.	El usuario se autentica en el SGU. El sistema muestra la pantalla donde se encuentran los diferentes SGU. El usuario selecciona el sistema Investigaciones. El sistema muestra las diferentes opciones de menú. El usuario selecciona el módulo "Estructura Investigativa". El sistema muestra todas las agrupaciones contenidas del módulo. El usuario selecciona la agrupación "Configuración". El usuario selecciona un bloque de los mostrados "Clasificación de líneas". El sistema muestra un listado con todas las clasificaciones de líneas existentes en el sistema. El usuario selecciona la opción "Crear" en el área de íconos flotantes. El sistema muestra una interfaz para introducir los datos de la nueva clasificación. El usuario no introduce todos los datos necesarios.
		"Vacío"				
EC 5.5 Insertar elementos repetidos	Mediante este escenario se permite insertar elementos	I	V	V	El sistema muestra un mensaje de error: "El elemento ya existe".	El usuario se autentica en el SGU. El sistema muestra la pantalla donde se encuentran los diferentes SGU. El usuario selecciona el sistema Investigaciones. El sistema muestra las diferentes opciones de menú. El usuario selecciona el módulo "Estructura Investigativa".
		"Líneas de investigación vinculadas a las TIC"	"activo"	"texto"		



	que ya están registrados en el sistema.	I	V	V	
		"Líneas de investigación vinculadas a las TIC"	"inactivo"	"texto"	
<p>El sistema muestra todas las agrupaciones contenidas del módulo. El usuario selecciona la agrupación "Configuración". El usuario selecciona un bloque de los mostrados "Clasificación de líneas". El sistema muestra un listado con todas las clasificaciones de líneas existentes en el sistema. El usuario selecciona la opción "Crear" en el área de íconos flotantes. El sistema muestra una interfaz para introducir los datos de la nueva clasificación. El usuario introduce los datos correctamente y presiona el botón Aceptar.</p>					



3.7.1 Pruebas de integración

Las pruebas de integración se concentran en la concepción y la arquitectura del software. Son realizadas para garantizar que los componentes del modelo de implementación funcionen correctamente. La prueba de integración aplicada, es la definida por el Centro Nacional de Calidad de Software (CALISOFT). Este propone un diseño de casos de pruebas específicos para este nivel de prueba. Se especifican los datos del sistema al cual se debe integrar. Cuando el sistema desarrollado funciona correctamente con estos datos, se asume que las pruebas de integración fueron satisfactorias.

Tabla 5: Prueba de integración módulo Personal.

Caso de prueba: Prueba de integración al módulo Personal del Núcleo del SGU
Módulo a integrar: módulo Personal
Número de caso de prueba: Int_1
Condiciones de ejecución: se realiza la petición de los datos.
Descripción de la prueba: comprobar que el módulo Personal proporcione la información de las personas de la Universidad.
Entradas/Pasos de ejecución: el controlador de la agrupación funcional de eventos, realiza una petición a la librería para obtener los datos de las personas gestionadas por el módulo Personal.
Resultado esperado: el módulo Personal brinda la información solicitada.
Evaluación: prueba satisfactoria

Tabla 6: Prueba de integración módulo Estructura y composición.

Caso de prueba: Prueba de integración al módulo Estructura y composición del Núcleo del SGU
Módulo a integrar: módulo Estructura y composición
Número de caso de prueba: Int_2
Condiciones de ejecución: se realiza la petición de los datos.
Descripción de la prueba: comprobar que el módulo Estructura y composición proporcione los datos referentes a la estructura organizativa definida por la Universidad.
Entradas/Pasos de ejecución: el controlador de la agrupación funcional de eventos, realiza una petición a la librería para obtener los datos de las facultades y centros, gestionados por el módulo Estructura y composición.
Resultado esperado: el módulo Estructura y composición brinda la información solicitada.
Evaluación: prueba satisfactoria

Tabla 7: Prueba de integración módulo Traza

Caso de prueba: Prueba de integración al módulo Traza del Núcleo del SGU
Módulo a integrar: módulo Traza
Número de caso de prueba: Int_3



Condiciones de ejecución: el usuario realiza cualquier acción sobre la agrupación funcional.
Descripción de la prueba: comprobar que el componente Traza registra todas la incidencias de un usuario sobre la agrupación funcional de estructura investigativa.
Entradas/Pasos de ejecución: luego de realizada una acción, el sistema registra los datos de la incidencia: nombre de usuario, dirección IP, acción ejecutada, fecha de la ejecución y otros. Después se procede a buscar mediante el registro de trazas, la incidencia deseada.
Resultado esperado: se muestran los datos de la traza registrada.
Evaluación: prueba satisfactoria

Tabla 8: Prueba de integración módulo Seguridad.

Caso de prueba: Prueba de integración al módulo Seguridad del Núcleo del SGU
Módulo a integrar: módulo Seguridad
Número de caso de prueba: Int_4
Condiciones de ejecución: el usuario debe estar autenticado en el sistema.
Descripción de la prueba: comprobar que el módulo Seguridad proporcione el acceso restringido a cada usuario en dependencia del rol que el mismo le asigne.
Entradas/Pasos de ejecución: luego de autenticado, el usuario procede a acceder a las funcionalidades de la solución.
Resultado esperado: el sistema muestra un mensaje de error de acceso denegado y en caso contrario accede a la funcionalidad.
Evaluación: prueba satisfactoria

3.7.2 Resultados de las pruebas

El proceso de pruebas se llevó a cabo en cinco iteraciones. En la primera iteración fueron revisadas 24 funcionalidades, detectándose 20 no conformidades relacionadas con errores ortográficos y de implementación, de las cuales se solucionaron 18, quedando dos por resolver que dependían de funcionalidades de otra iteración. En la segunda iteración fueron revisadas 38 funcionalidades, en las que se registraron 19 no conformidades las cuales fueron corregidas. En la tercera iteración se revisaron 52 funcionalidades, de las cuales se descubrieron cinco no conformidades que se resolvieron en su totalidad. En una cuarta iteración fueron revisadas 65 funcionalidades, donde se corrigieron las dos no conformidades identificadas. En la última iteración realizada se revisaron los 75 requisitos funcionales identificados, en los cuales no se detectaron errores.

En general, de un total de 75 requisitos funcionales obtenidos se detectaron 46 no conformidades distribuidas en las cinco iteraciones de pruebas realizadas, corrigiéndose todas. En la figura 9 se muestra un gráfico que ilustra los resultados obtenidos de las pruebas funcionales realizadas.



Las pruebas de integración del módulo Estructura Investigativa con los módulos del núcleo del SGU, demostraron que la integración es exitosa. Esto permitió comprobar que la obtención de datos de estos módulos se realizó de forma satisfactoria.

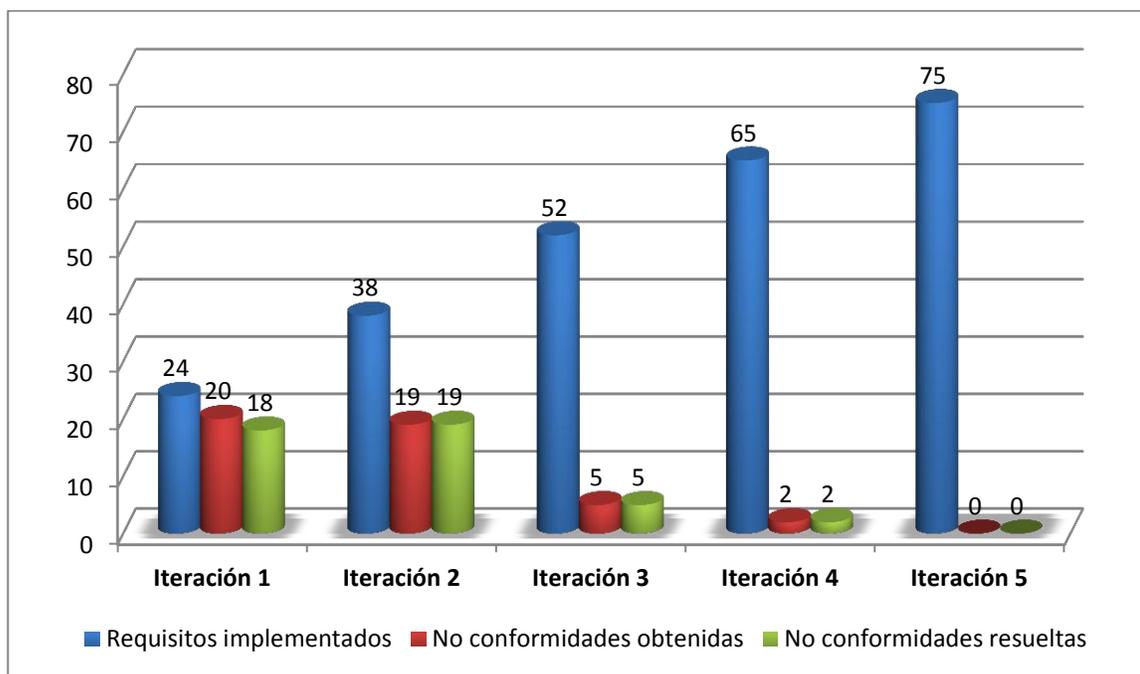


Figura 9: Resultado de las pruebas.

3.8 Conclusiones parciales

Con el uso del estándar de codificación, las definiciones para el tratamiento de errores y las técnicas de programación definidas en la arquitectura del SGU, se logró implementar el módulo que brinda solución a la problemática planteada. La realización del proceso de pruebas para la validación de las funcionalidades permitió corregir los errores encontrados.



Conclusiones

- ✓ El estudio teórico referencial permitió definir el marco conceptual asociado a la investigación y demostró que tanto en el ámbito internacional como nacional, no existe un sistema para la gestión de las líneas, grupos y proyectos de investigación que erradique los problemas planteados anteriormente.
- ✓ El estudio de las pautas definidas en el CENIA para el desarrollo de sistemas de gestión de procesos académicos, permitió diseñar una propuesta que permite administrar el proceso asociado a la gestión de la estructura investigativa en la Universidad de las Ciencias Informáticas y que puede ser integrado al Sistema de Gestión Universitaria.
- ✓ Con el uso del estándar de codificación, las definiciones para el tratamiento de errores y las técnicas de programación definidas en la arquitectura del SGU, se logró implementar el módulo que brinda solución a la problemática planteada.
- ✓ La ejecución de las pruebas definidas permitió validar las funcionalidades identificadas con el cliente.

Por todo lo anteriormente planteado se puede afirmar que se cumplió el objetivo de la investigación.



Recomendaciones

Para darle continuidad a la investigación se recomienda que:

- ✓ El módulo Estructura Investigativa pueda integrarse al componente Notificaciones del Sistema de Gestión Universitaria.
- ✓ Adicionar las funcionalidades gestionar instituciones participantes, controles, reportes y perfiles.



Bibliografía referenciada

1. **VI Congreso del Partido Comunista de Cuba.** Lineamientos de la política económica y social de la revolución cubana. *Juventud Rebelde*. 2011.
2. **Dirección de Investigación.** *Política científica de la Universidad de las Ciencias Informáticas*. [En línea] 05 de noviembre de 2011. [Citado el: 11 de octubre de 2012.] http://investigaciones.uci.cu/downloads.php?cat_id=18.
3. **Agencia de Medio Ambiente.** *Ministerio de Ciencia, Tecnología y Medio Ambiente-Cuba*. [En línea] 2000. [Citado el: 20 de enero de 2013.] <http://www.medioambiente.cu/organigrama.asp#top>.
4. **Investigaciones.** *Investigaciones*. [En línea] UCI, 2010. [Citado el: 4 de diciembre de 2012.] <http://investigaciones.uci.cu/news.php>.
5. **Definición ABC.** *Definición ABC tu diccionario hecho fácil*. [En línea] 2007. [Citado el: 24 de enero de 2013.] <http://www.definicionabc.com/general/estructura.php#ixzz2JxNFehzp>.
6. **Nagy, Jenette.** *Caja de Herramientas Comunitarias*. [En línea] Universidad de Kansa, 2012. [Citado el: 5 de diciembre de 2012.] <http://ctb.ku.edu/es/tablecontents/Seccionprincipal9.1.aspx>.
7. **Dirección de Investigaciones.** *LI_01 Algunas cuestiones a considerar en la organización de líneas de investigación científica de la UCI*. La Habana : s.n., 2011.
8. **Departamento Administrativo de Ciencia Tecnología e Innovación.** Convocatorias COLCIENCIAS 2012. *SRG Sistema General de Regalías. Regalías para la Ciencia, la Tecnología y la Innovación*. [En línea] República de Colombia. [Citado el: 21 de enero de 2013.] <http://www.colciencias.gov.co/>.
9. **Dirección de Investigaciones.** *Manual de Procedimientos. Presentación y aceptación de "Proyectos de Investigación"*. La Habana : s.n., 2013.
10. **Equipo de desarrollo.** *netofficedwins. Net Office Dwins*. [En línea] 10 de febrero de 2007. <http://netoofficedwins.sourceforge.net/.5>.
11. **Clarcats Consulting.** Tecnologías de información, aplicadas con criterio a su negocio. *EGroupWare*. [En línea] 2009. [Citado el: 28 de enero de 2013.] www.clarcats.com/es/productos/egroupware.
12. **Cynertia Consulting.** Sistema de gestión de proyectos. *Cynertia Consulting ¿Te atreves a ganar?* [En línea] 2013. [Citado el: 4 de febrero de 2013.] <http://www.cynertiaconsulting.com/es/consultoria-TIC/sistemas-de-gesti%C3%B3n-de-proyectos>.



13. **Ortiz, Dr. Gustavo.** SIV. *Sistema de Proyectos de Investigación*. [En línea] 1999. <http://www.uccor.edu.ar/sistema-sip/>.
14. **CENDA.** *Gespro*. [En línea] 2010. <http://gespro-help.prod.uci.cu/introduction/#introduccion>.
15. **Sommerville, Ian.** *Ingeniería de software*. Séptima. Madrid : s.n., 2005. ISBN 84-7829-074-5.
16. **Software Engineering Institute.** *CMMI® for Development, Version 1.2*. Engineering Institute Pittsburgh : s.n., 2006. PA 15213-3890.
17. **Pérez Montalván, Deborat et al.** *PROGRAMA DE MEJORA DE LOS PROCESOS PRODUCTIVOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS*. La Habana : s.n.
18. **Rodríguez Andrés, Rolando.** GestioPolis. *Lenguajes, notaciones y herramientas para el modelado y análisis de procesos*. [En línea] 16 de junio de 2008. [Citado el: 15 de febrero de 2013.] <http://www.gestiopolis.com/administracion-estrategia/lenguajes-notaciones-y-herramientas-en-analisis-de-procesos.htm>.
19. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado. Manual de Referencia*. [ed.] Addison Wesley. 2000. ISBN 84-7829-037-08.
20. **Enciclopedia Universal.** Enciclopedia Universal Académica. *Lenguaje de programación*. [En línea] 2012. http://enciclopedia_universal.esacademic.com/74817/Lenguaje_de_programaci%C3%B3n.
21. **Enciclopedia Universal .** Enciclopedia Universal Académica. *PHP*. [En línea] 2012. [Citado el: 11 de febrero de 2013.] <http://www.esacademic.com/dic.nsf/eswiki/3167>.
22. **JavaScript Ya.** *JavaScript Ya*. [En línea] febrero de 2009. [Citado el: 2 de febrero de 2013.] www.javascriptya.com.ar.
23. **BuenasTareas.com.** *Que es un IDE de programación*. [En línea] marzo de 2010. <http://www.buenastareas.com/ensayos/Que-Es-Un-Ide-De-Programacion/163537.html>.
24. **Corredor, Germán.** *Sun libera su IDE NetBeans para Java y PHP*. [En línea] julio de 2009. <http://osum.sun.com/profiles/blogs/sun-libera-su-ide-netbeans-67-1...>
25. **García Vidal, Yanio.** *Documento de Arquitectura de Software. Sistema de Gestión Académica de Pregrado (1.2)*. La Habana : Departamento de Gestión Universitaria, Universidad de las Ciencias Informáticas., 2011.
26. **Caballero, G. M.** SCRIBD. *BIZAGI Process Modeler*. [En línea] 2011. [Citado el: 30 de enero de 2013.] <http://es.scribd.com/doc/60271008/Bizagi-Process-Modeler>.



27. **Lillo, A.** *Nordic Design*. [En línea] 5 de octubre de 2009. [Citado el: 30 de enero de 2013.] <http://www.nordic-design.net/2009/10/05/foreui-otra-herramienta-de-prototipado/>.
28. **Visual Paradigm.** Visual Paradigm for UML. [En línea] 2013. <http://www.visual-paradigm.com>.
29. **Alvarez, Sara.** *Sistemas gestores de bases de datos. Introducción a este concepto y características especiales*. [En línea] 31 de junio de 2007. <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
30. **Consumoteca.** *Servidor Web*. [En línea] 12 de agosto de 2009. [Citado el: 18 de enero de 2013.] <http://www.consumoteca.com/diccionario/servidor-web>.
31. **The Apache Software Foundation.** *The Apache Software Foundation*. [En línea] [Citado el: 2 de febrero de 2013.] <http://www.apache.org/>.
32. **Pressman, R. S.** *Ingeniería De Software, Un Enfoque Práctico*. Quinta. 2002. ISBN 8448132149.



Bibliografía consultada

Agencia de Medio Ambiente. MEDIOAMBIENTE.CU. *Ministerio de Ciencia, Tecnología y Medio Ambiente-Cuba.* [En línea] 2000. [Citado el: 20 de enero de 2013.] <http://www.medioambiente.cu/organigrama.asp#top>.

Alvarez, Sara. desarrolloweb.com. *Arquitectura cliente-servidor.* [En línea] 30 de agosto de 2007. <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.

Alvarez, Sara. desarrolloweb.com. *Sistemas gestores de bases de datos. Introducción a este concepto y características especiales.* [En línea] 31 de junio de 2007. <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.

BuenasTareas.com. *Que es un IDE de programación.* [En línea] marzo de 2010. <http://www.buenastareas.com/ensayos/Que-Es-Un-Ide-De-Programacion/163537.html>.

BuenasTareas.com. *Técnicas de programación.* [En línea] abril de 2010. <http://www.buenastareas.com/ensayos/e-T%C3%A9cnicas-De-Programaci%C3%B3n-Docx/199270.html>.

Caballero, G. M. SCRIBD. *BIZAGI Process Modeler.* [En línea] 2011. [Citado el: 30 de enero de 2013.] <http://es.scribd.com/doc/60271008/Bizagi-Process-Modeler>.

CENDA. Gespro. *Gespro.* [En línea] 2010. <http://gespro-help.prod.uci.cu/>.

Clarcat Consulting. Tecnologías de información, aplicadas con criterio a su negocio. *EGroupWare.* [En línea] 2009. [Citado el: 28 de enero de 2013.] www.clarcat.com/es/productos/egroupware.

Consumoteca. *Servidor Web.* [En línea] 12 de agosto de 2009. [Citado el: 18 de enero de 2013.] <http://www.consumoteca.com/diccionario/servidor-web>.

Corredor, Germán. osum.sun.com. *Sun libera su IDE NetBeans para Java y PHP.* [En línea] julio de 2009. <http://osum.sun.com/profiles/blogs/sun-libera-su-ide-netbeans-67-1...>

Cynertia Consulting. Sistema de gestión de proyectos. *Cynertia Consulting ¿Te atreves a ganar?* [En línea] 2013. [Citado el: 4 de febrero de 2013.] <http://www.cynertiaconsulting.com/es/consultoria-TIC/sistemas-de-gesti%C3%B3n-de-proyectos>.

Definición ABC. *Definición ABC tu diccionario hecho fácil.* [En línea] 2007. [Citado el: 24 de enero de 2013.] <http://www.definicionabc.com/general/estructura.php#ixzz2JxNFehzp>.



Departamento Administrativo de Ciencia Tecnología e Innovación. Convocatorias COLCIENCIAS 2012. *SRG Sistema General de Regalías. Regalías para la Ciencia, la Tecnología y la Innovación.* [En línea] República de Colombia. [Citado el: 21 de enero de 2013.] <http://www.colciencias.gov.co/>.

Dirección de Investigaciones. *LI_01 Algunas cuestiones a considerar en la organización de líneas de investigación científica de la UCI.* LI. La Habana: s.n., 2011.

Dirección de Investigaciones. Manual de Procedimientos. *Presentación y aceptación de "Proyectos de Investigación".* La Habana: s.n., 2013.

Dirección de Investigación. Investigaciones. *Política científica de la Universidad de las Ciencias Informáticas.* [En línea] 05 de noviembre de 2011. [Citado el: 11 de octubre de 2012.] http://investigaciones.uci.cu/downloads.php?cat_id=18.

Enciclopedia Universal. Enciclopedia Universal Académica. *PHP.* [En línea] 2012. [Citado el: 11 de febrero de 2013.] <http://www.esacademic.com/dic.nsf/eswiki/3167>.

Enciclopedia Universal. Enciclopedia Universal Académica. *Lenguaje de programación.* [En línea] 2012. http://enciclopedia_universal.esacademic.com/74817/Lenguaje_de_programaci%C3%B3n.

Equipo de desarrollo. netofficedwins. *Net Office Dwins.* [En línea] 10 de febrero de 2007. <http://netofficedwins.sourceforge.net/5>.

ForeUI. ForeUI Easy-To-Use UI Prototyping Tool. *¿Qué es ForeUI?* [En línea] 2009. [Citado el: 30 de enero de 2013.] <http://www.foreui.com/tour.htm>.

García Vidal, Yanio. Documento de Arquitectura de Software. *Sistema de Gestión Académica de Pregrado (1.2).* La Habana: Departamento de Gestión Universitaria, Universidad de las Ciencias Informáticas., 2011.

Investigaciones. *Investigaciones.* [En línea] UCI, 2010. [Citado el: 4 de diciembre de 2012.] <http://investigaciones.uci.cu/news.php>.

JavaScript Ya. *JavaScript Ya.* [En línea] febrero de 2009. [Citado el: 2 de febrero de 2013.] www.javascriptya.com.ar.

Lillo, A. *Nordic Design.* [En línea] 5 de octubre de 2009. [Citado el: 30 de enero de 2013.] <http://www.nordic-design.net/2009/10/05/foreui-otra-herramienta-de-prototipado/>.

Machin Castillo, Jorge Luis y Corso Bouza, Anay. Biblioteca. *Descripción del proceso de construcción del sistema operativo base de la distribución cubana de GNU/Linux Nova.* [En línea] mayo de 2010.



[Citado el: 5 de diciembre de 2012.] http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_03556_10. TD_03556_10.

Nagy, Jenette. *Caja de Herramientas Comunitarias*. [En línea] Universidad de Kansa, 2012. [Citado el: 5 de diciembre de 2012.] <http://ctb.ku.edu/es/tablecontents/Seccionprincipal9.1.aspx>.

OPENLIBRA. *Programación orientada a Objetos*. [En línea] 2003. <http://www.etnassoft.com/biblioteca/programacion-orientada-a-objetos/>.

Ortiz, Dr. Gustavo. SIV. *Sistema de Proyectos de Investigación*. [En línea] 1999. <http://www.uccor.edu.ar/sistema-sip/>.

Palomo, Ivan F., Veloso, Carlos G. y Schmal, Rodolfo F. SciELO. *Sistema de Gestión de la Investigación en la Universidad de Talca*. [En línea] 2007. <http://www.scielo.cl/>.

Pérez Montalván, Deborat et al. *PROGRAMA DE MEJORA DE LOS PROCESOS PRODUCTIVOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS*. La Habana : s.n.

Pressman, R. S. *Ingeniería De Software, Un Enfoque Práctico*. Quinta. 2002. ISBN 8448132149.

Rodríguez Andrés, Rolando. GestioPolis. *Lenguajes, notaciones y herramientas para el modelado y análisis de procesos*. [En línea] 16 de junio de 2008. [Citado el: 15 de febrero de 2013.] <http://www.gestiopolis.com/administracion-estrategia/lenguajes-notaciones-y-herramientas-en-analisis-de-procesos.htm>.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. *El Lenguaje Unificado de Modelado. Manual de Referencia*. [ed.] Addison Wesley. 2000. ISBN 84-7829-037-08.

Sánchez Barriento, Manuel. QueComoQuien. *Introducción a BPMN*. [En línea] 2 de noviembre de 2008. [Citado el: 25 de enero de 2013.] <http://www.quecomoquien.es/introduccion-a-bpmn.html>.

Sánchez Mariñez, Julio. *El rol de las universidades en el desarrollo científico-tecnológico*. Santo Domingo: s.n., 2008.

Servidor HTTP Apache. Módulos de MultiProcesamiento (MPMs) - Servidor HTTP Apache. *Versión 2.2 del Servidor HTTP Apache*. [En línea] [Citado el: 2013 de enero de 15.] <http://httpd.apache.org/docs/2.2/es/mpm.html#introduction>.

SOFTPEDIA. *Crea prototipos adaptables e interactivos para los sitios web o software, que se ejecutan como DHTML*. [En línea] 9 de junio de 2011. [Citado el: 25 de enero de 2013.] <http://www.softpedia.es/programa-EaSynth-ForeUI-126801.html>.



Software Engineering Institute. *CMMI® for Development, Version 1.2.* Engineering Institute Pittsburgh : s.n., 2006. PA 15213-3890.

Sommerville, Ian. *Ingeniería de software.* Séptima. Madrid : s.n., 2005. ISBN 84-7829-074-5.

THE Agile Alliance. *Manifiesto for Agile Software Dovelopment.* [En línea] febrero de 2001. [Citado el: 12 de febrero de 2013.] <http://agilemanifesto.org/>.

The Apache Software Foundation. *The Apache Software Foundation.* [En línea] [Citado el: 2 de febrero de 2013.] <http://www.apache.org/>.

VI Congreso del Partido Comunista de Cuba. *Lineamientos de la política económica y social de la revolución cubana.* Juventud Rebelde. 2011.

Visual Paradigm. *Visual Paradigm for UML.* [En línea] 2013. <http://www.visual-paradigm.com>.

Wikilibros. *Fundamentos de programación. Herramientas de desarrollo.* [En línea] 2012. http://es.wikibooks.org/wiki/Fundamentos_de_programaci%C3%B3n/Herramientas_de_desarrollo.



Glosario de términos

AJAX (*Asynchronous JavaScript And XML*): Significa JavaScript asíncrono y XML, es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones hacen posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

CASE (*Computer Aided Software Engineering*): Significa Ingeniería de Software Asistida por Computadora, son diversas aplicaciones informáticas destinadas a aumentar la efectividad en el desarrollo de software y sistemas contribuyendo a una reducción de costos de producción.

Clase de equivalencia: Representa un conjunto de estados válidos y no válidos para las condiciones de entrada de un programa.

CMMI (*Capability Maturity Model Integration*): Significa Modelo de Capacidad y Madurez Integrado, es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

Código abierto: en inglés, *Open Source*, es el término con el que se conoce al software distribuido y desarrollado libremente.

CSS (*Cascading Style Sheets*): Significa Hojas de Estilo en Cascada, es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

DOM (*Document Object Model*): Significa Modelo de Objetos de Documento, es un interfaz de programación de aplicaciones para documentos HTML. Define la estructura lógica de los documentos y el modo cómo se accede y manipula un documento.

HTML (*HyperText Markup Language*): Significa Lenguaje de Marcado de Hipertexto, es el lenguaje de marcado predominante para la elaboración de páginas web.

HTML5: Es la versión más nueva del Lenguaje de Marcado de Hipertexto.

HTTP (*Hypertext Transfer Protocol*): Significa Protocolo de Transferencia de Hipertexto, es un protocolo que establece la comunicación entre un cliente y un servidor.

HTTPS (*Hypertext Transfer Protocol Secure*): Significa Protocolo Seguro de Transferencia de Hipertexto, es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP.



IMAP (*Internet Message Access Protocol*): Significa Protocolo de Acceso a Mensajes de Internet, es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

IoC (*Inversion of Control*): Significa Inversión de control, es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos o funciones.

LDAP (*Lightweight Directory Access Protocol*): Significa Protocolo Ligero de Acceso a Directorios, es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

LI02: Es el documento que recoge la información general de una Línea de investigación.

Multiplataforma: es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

PDF (*Portable Document Format*): Significa Formato de Documento Portátil, es un formato de almacenamiento de documentos digitales independiente de plataformas de software o hardware.

Proceso: conjunto de actividades relacionadas lógicamente que producen una salida o resultado.

Sistemas distribuidos: Colección de computadores autónomos conectados entre sí por una red.

Software: Equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

SQL (*Structured Query Language*): Significa Lenguaje de Consulta Estructurado, es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

TCP: Significa Protocolo de Control de Transmisión, es uno de los principales protocolos en las redes TCP/IP, le permite a dos computadoras anfitrionas establecer una conexión e intercambiar flujos de datos, garantiza el despacho de los datos y que los paquetes sean despachados en el mismo orden en que fueron enviados.

Webmail: Es un cliente de correo electrónico, que provee una interfaz web por la cual se puede acceder al correo electrónico.

Widgets: Son pequeños programas que funcionan como aditamentos a una plataforma más importante, ya sea un sistema operativo o un navegador web.

Anexo 1: Sub-procesos del modelo de proceso de negocio

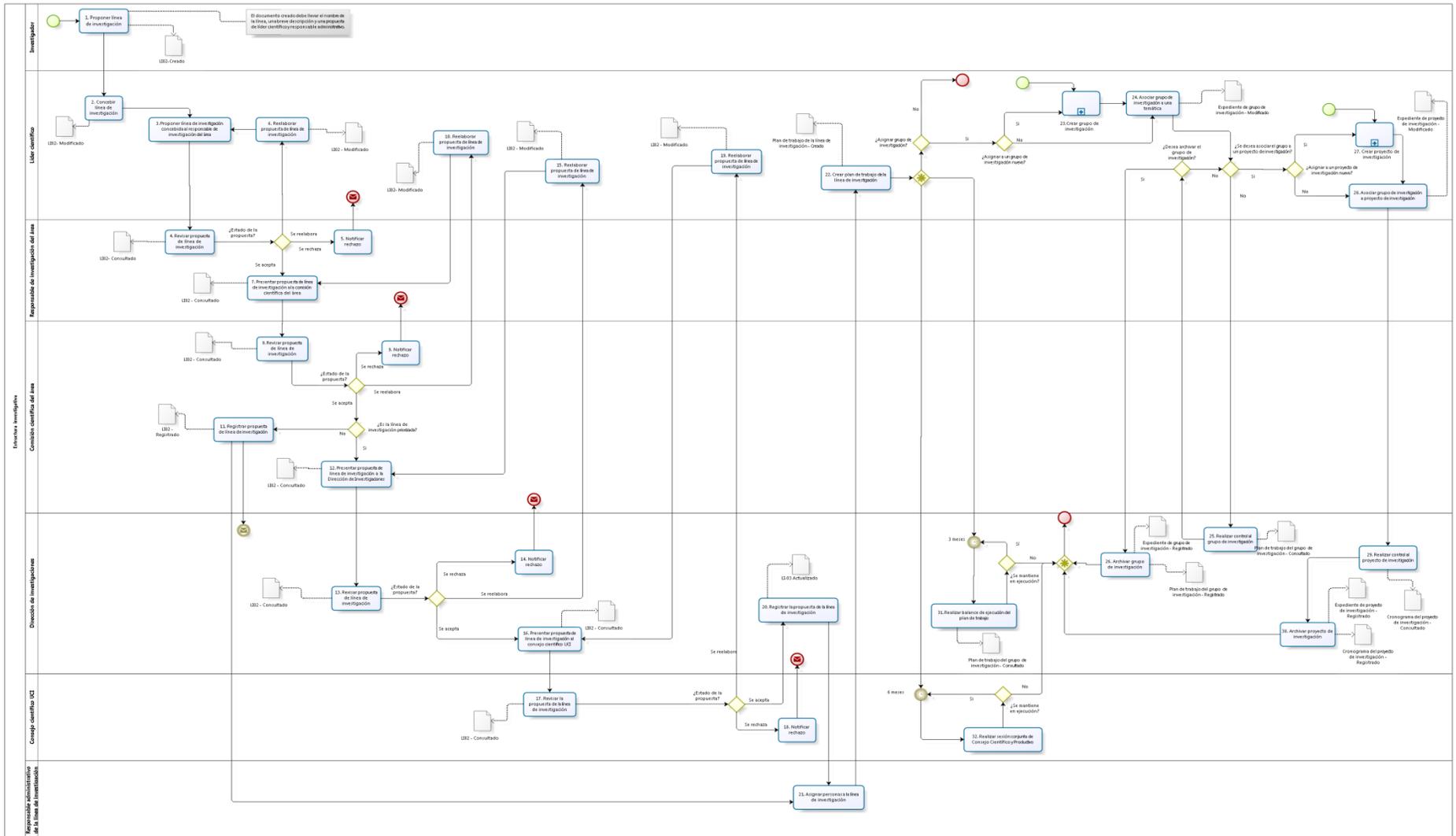


Figura 10: Proceso Estructura investigativa (completo).

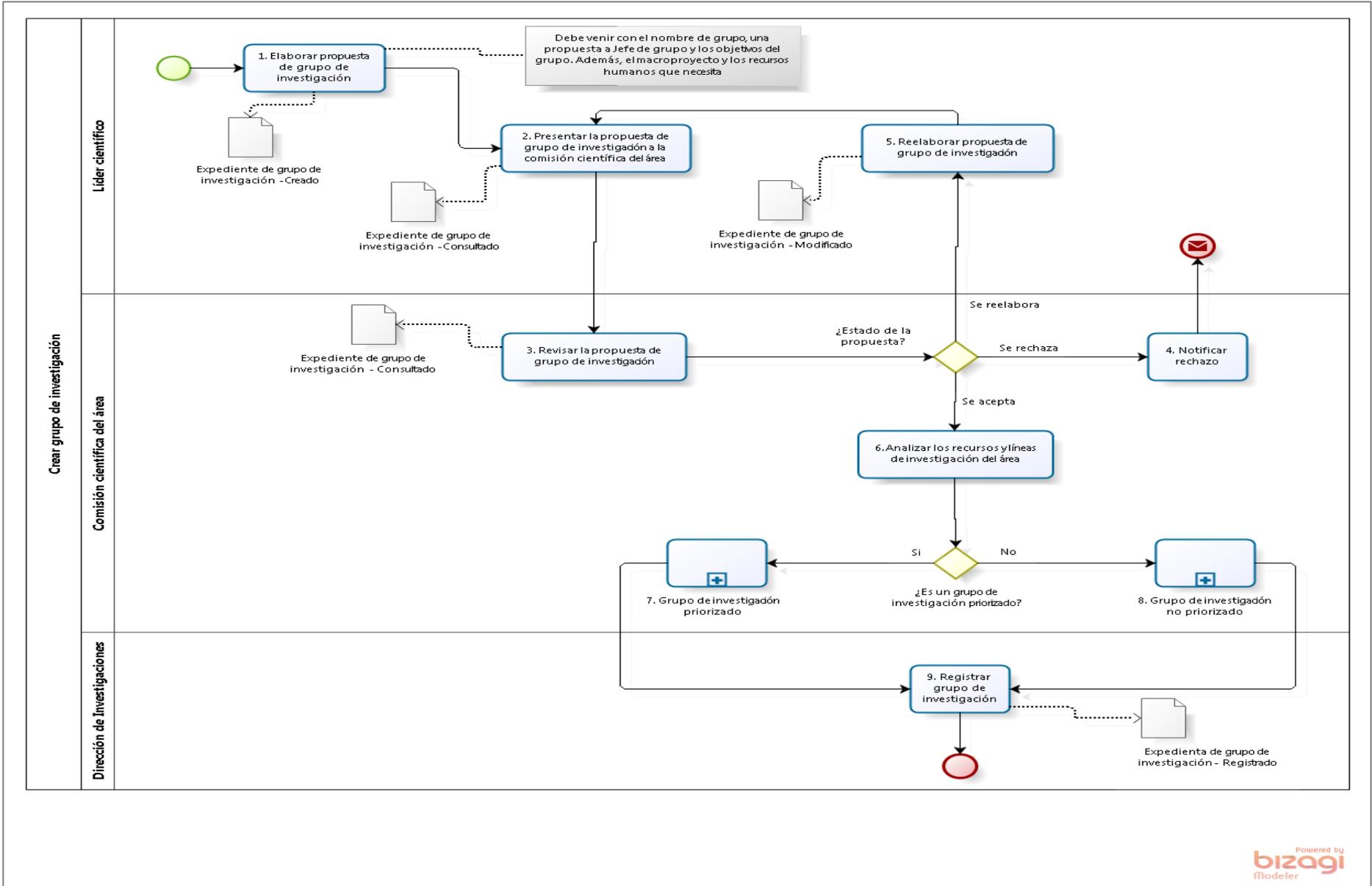


Figura 11: Sub-proceso Crear grupo de investigación.

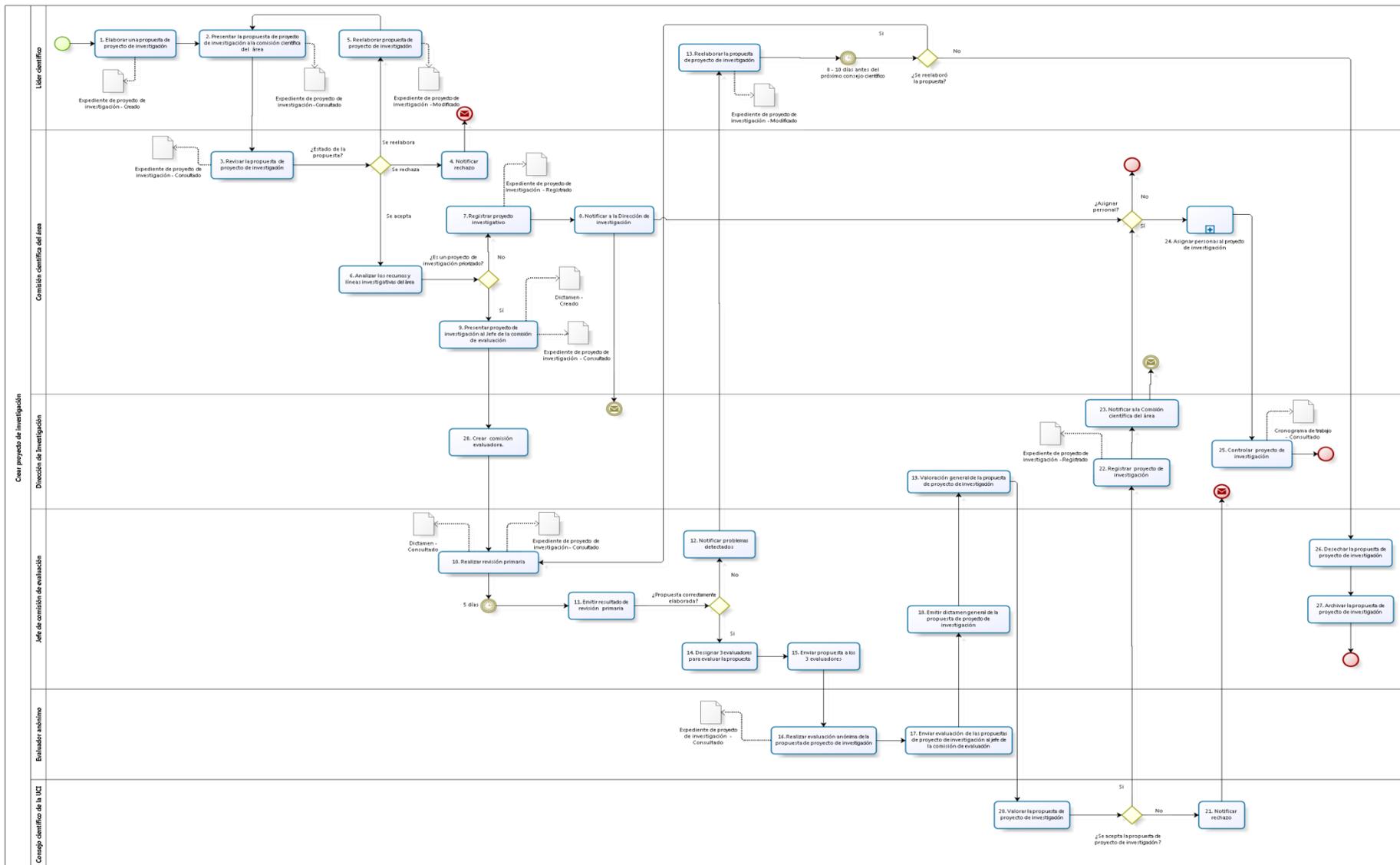


Figura 12: Sub-proceso Crear proyecto de investigación.

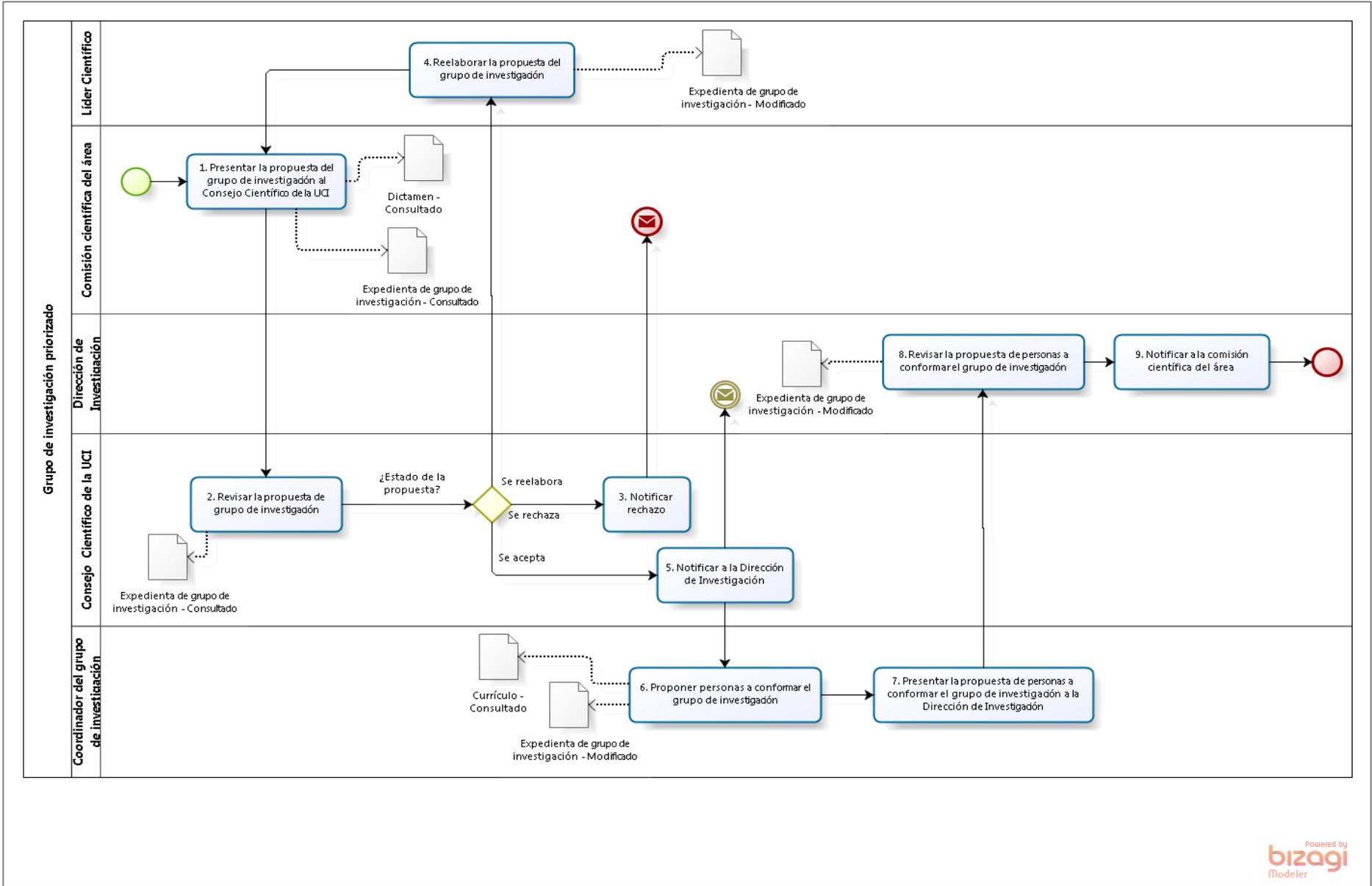


Figura 13: Sub-proceso Grupo de investigación priorizado.

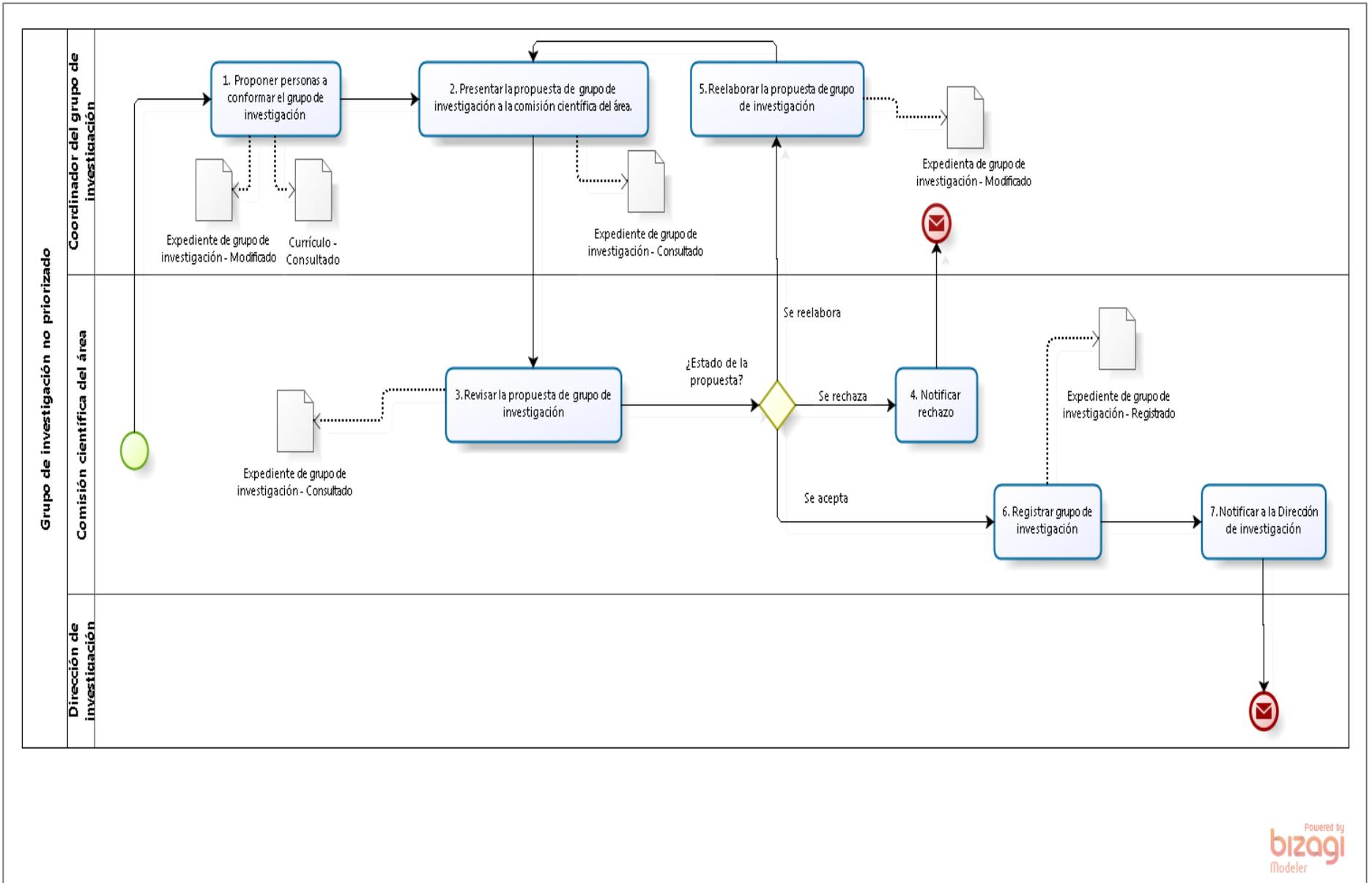


Figura 14: Sub-proceso Grupo de investigación no priorizado.

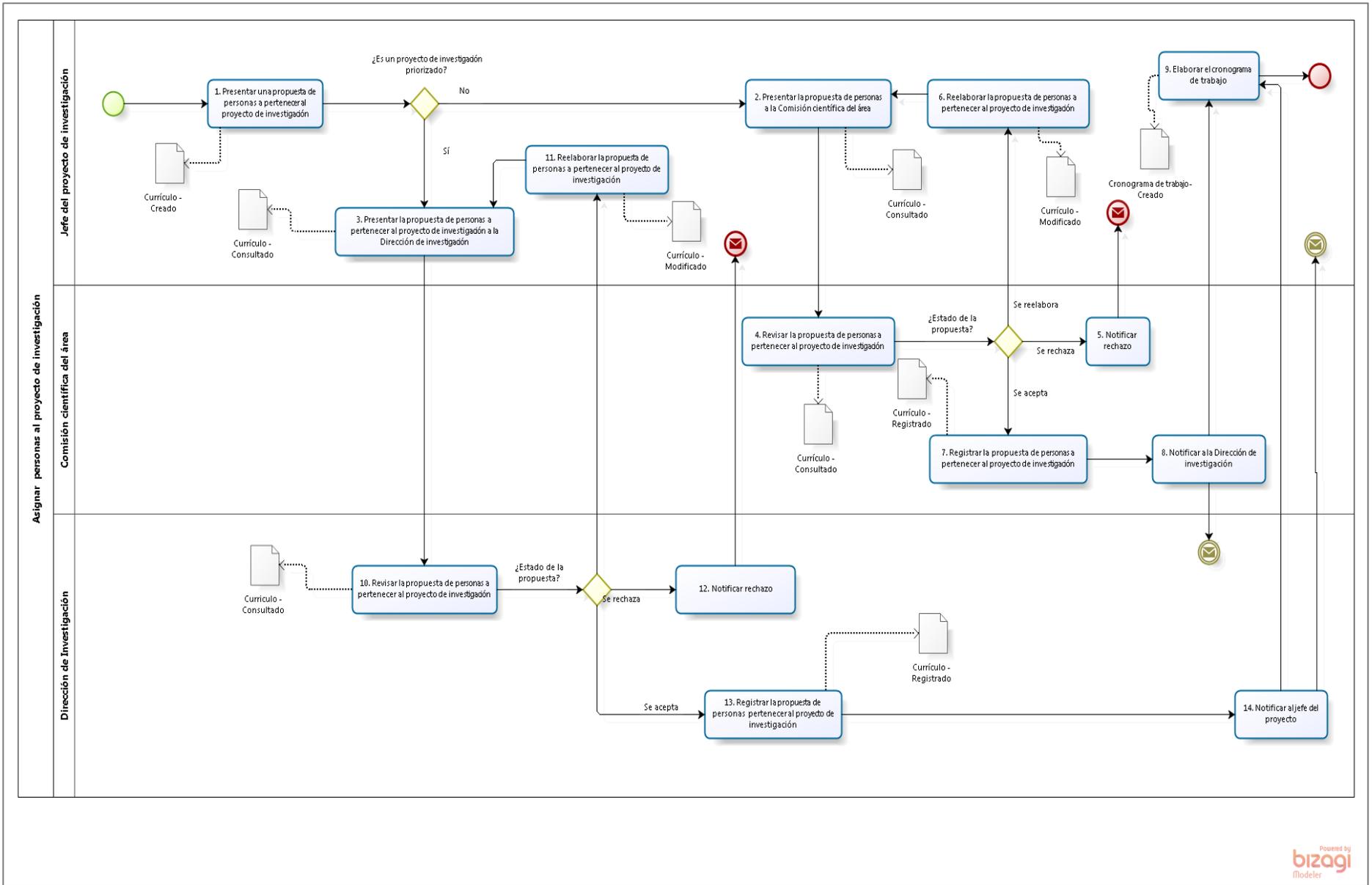


Figura 15: Sub-proceso Asignar personas a un proyecto de investigación.



Anexo 2: Descripción de los requisitos funcionales

RF1_EI_ Crear línea de investigación.

Crear una línea con todos los datos necesarios para la misma. Inicialmente solo tendrá como campos obligatorios, los que se insertan en la propuesta inicial (que inserta cualquier investigador). El resto los completará el líder científico que se designe. Deberá insertarse quién es el responsable administrativo de la línea de investigación. Deberá validar que el nombre del nuevo elemento no esté en uso actualmente o se encuentre entre los archivados.

RF2_EI_ Insertar investigadores a la línea de investigación.

Insertar los recursos humanos a la línea de investigación.

Si se inserta un investigador que ya estuvo en la línea y fue eliminada, se vuelve a recobrar el vínculo con todo el trabajo que había realizado anteriormente.

Deberá validar que no se duplique un investigador insertado en la estructura.

RF3_EI_ Eliminar investigadores de la línea de investigación.

Eliminar a un investigador de una línea de investigación. Deberá mantenerse el registro de que ese investigador trabajó en la línea por tanto tiempo, así como las actividades que realizó.

Cuando se elimina un investigador de la línea, es necesario preguntar si se desea eliminar al investigador de los grupos y proyectos asociados a la línea. Para esto se muestra una lista de los grupos y proyectos a los que está asociado y se podrá seleccionar qué asociaciones eliminar.

Si el investigador es líder investigativo, coordinador de grupo o líder de proyecto, no podrá eliminarse hasta que haya sido relevado del rol.

RF4_EI_ Listar los recursos humanos de la línea de investigación.

Lista todos los investigadores que se encuentran en el grupo de investigación.

RF5_EI_ Crear clasificación de línea de investigación.

Crear una nueva clasificación de línea de investigación. Deberá validar que el nuevo elemento no esté en uso actualmente.

RF6_EI_ Modificar clasificación de línea de investigación.

Modificar una clasificación de línea de investigación.

RF7_EI_ Ver detalles clasificación de línea de investigación.

Mostrar detalles de una clasificación de línea de investigación.

RF8_EI_ Listar las clasificaciones de línea de investigación.

Mostrar un listado con todos los tipos de líneas de investigación.

RF9_EI_ Crear sub-línea de la línea de investigación.



Crear una nueva sub-línea para la línea de investigación. Deberá validar que el nombre del nuevo elemento no esté en uso actualmente o se encuentre entre los archivados.

RF10_EI_ Modificar sub-línea de la línea de investigación.

Modificar una sub-línea de la línea de investigación.

RF11_EI_ Ver detalles sub-línea de la línea de investigación.

Mostrar detalles de una sub-línea de la línea de investigación.

RF12_EI_ Listar las sub-línea de la línea de investigación.

Mostrar un listado con todas las sub-líneas de la línea de investigación.

RF13_EI_ Gestionar macro-proyecto de la línea de investigación.

Gestionar las tareas del macro-proyecto de la línea de investigación. Estas tareas de investigación las inserta solamente el líder científico de la línea. Solamente podrá existir un macro-proyecto para cada línea de investigación.

RF14_EI_ Insertar tarea.

Insertar una tarea a incluir en el macro-proyecto o plan de trabajo de una línea de investigación o de un grupo o proyecto de investigación respectivamente.

RF15_EI_ Cambiar estado de una tarea.

Permite poder realizar directamente el cambio de estado de una tarea incluida en el macro-proyecto o plan de trabajo de una línea de investigación o de un grupo o proyecto de investigación respectivamente. Cuando el estado pase a *Resuelta*, debe darse una evaluación. En cada cambio de estado, pueden insertarse comentarios, notas o documentos.

RF16_EI_ Modificar tarea.

Modificar los datos de una tarea incluida en el macro-proyecto o plan de trabajo de una línea de investigación o de un grupo o proyecto de investigación respectivamente. Incluye poder cambiar el estado de la misma.

RF17_EI_ Archivar tarea.

Archivar una tarea. En caso de que la tarea esté siendo utilizada, se indicará dónde se está utilizando y no se permitirá archivar.

RF18_EI_ Desarchivar tarea.

En caso que sea necesario volver a utilizar una sub-línea archivada, podrá volver a utilizarse.

RF19_EI_ Ver detalles de tareas.

Mostrar detalles de tarea de investigación.

RF20_EI_ Listar tarea.

Mostrar un listado con todas las tareas de investigación

**RF26_EI_ Modificar línea de investigación.**

Modificar los datos de una línea de investigación. También a la funcionalidad que hace referencia para completar la concepción de la línea por parte del líder científico de la misma (el paso después de proponer la línea). Deberá validar que el nombre del nuevo elemento no esté en uso actualmente o se encuentre entre los archivados.

RF27_EI_ Archivar línea de investigación.

Archivar una línea de investigación. Archiva todos los grupos y proyectos de investigación que tienen a esa línea como línea principal.

RF28_EI_ Desarchivar línea de investigación.

En caso que se necesite volver a utilizar una línea de investigación, puede seleccionarse de la lista de líneas de investigación archivadas, y se permite seleccionar que vínculos recobrar con los elementos que tenía anteriormente. Los elementos que estén archivados y tenían asociación con la línea permanecerán archivados.

RF29_EI_ Listar líneas de investigación.

Mostrar una lista de todas las líneas de investigación. Debe poder aplicarse filtros para la búsqueda. Incluir el filtro de líneas priorizadas y de líneas no priorizadas. Incluir el filtro de líneas registradas y no registradas. Incluir el filtro de estados. Incluir el filtro de tiempo de creación.

RF30_EI_ Ver detalles generales de la línea de investigación.

Mostrar datos generales de la línea de investigación, así como los datos generales de los grupos y proyectos que incluye.

RF31_EI_ Valorar propuesta de líneas de investigación.

Se listan las propuestas de las líneas de investigación que ha presentado. En dependencia del rol del usuario, se presentan las propuestas que puede aprobar. Si es el responsable de investigación del área podrá aprobar las que ha realizado un líder científico; si es el presidente de la comisión científica del área aprobará los que haya aprobado el responsable de investigación; si es el consejo científico de la universidad, aprueba los de la comisión científica.

RF32_EI_ Revisar propuesta de línea de investigación.

Muestra los detalles de la línea de investigación propuesta, así como los datos de todos los estados por los que ha pasado y quiénes han aprobado la línea hasta el momento. En este momento es que se podrá rechazar, aprobar o pedir reelaboración de la propuesta. En el caso de que se pida reelaboración, se pondrán comentarios en los campos a modificar. En el caso que se rechace, se insertarán detalles que expliquen el rechazo de la propuesta.

RF33_EI_ Reelaborar propuesta de línea de investigación.

Funciona igual que el modificar, pero la diferencia es que se mostrarán los comentarios que se hicieron a la propuesta.

RF34_EI_ Registrar línea de investigación.



Es cuando se oficializa la línea investigativa. Se muestra la lista de líneas aprobadas por el consejo científico UCI. Es solo a esas líneas registradas que podrán agregarse/asociarse grupos o proyectos. Son solo estas líneas registradas las que pueden aparecer en el LI03.

RF35_EI_ Listar los grupos de investigación asociados a la línea de investigación.

Listar los grupos de investigación asociados a la línea.

RF36EI_ Crear grupo de investigación.

Insertar los datos necesarios para crear un grupo investigativo. Deberá validar que el nombre del nuevo elemento no esté en uso actualmente o se encuentre entre los archivados.

RF37_EI_ Insertar investigadores al grupo de investigación.

Insertar un investigador al grupo de investigación. Deberá validar que no se duplique un investigador insertado en la estructura.

RF38_EI_ Eliminar investigadores del grupo de investigación.

Eliminar a un investigador de un grupo de investigación. Deberá mantenerse el registro de que ese investigador trabajó en el grupo por tanto tiempo, así como las actividades que realizó.

Cuando se elimina un investigador del grupo, es necesario preguntar si se desea eliminar al investigador de los proyectos asociados a al grupo. Para esto se muestra una lista de los proyectos a los que está asociado y se podrá seleccionar de qué asociaciones eliminar.

Si el investigador es coordinador de grupo o líder de proyecto, no podrá eliminarse hasta que haya sido relevado del rol.

RF39_EI_ Listar los recursos humanos del grupo de investigación.

Lista todos los investigadores que se encuentran en el grupo de investigación.

RF40_EI_ Gestionar plan de trabajo del grupo de investigación.

Gestionar las tareas del plan de trabajo del grupo de investigación. Solamente existirá un plan de trabajo por grupo de investigación.

RF42_EI_ Modificar grupo de investigación.

Modificar los datos de un grupo de investigación.

RF43_EI_ Archivar grupo de investigación.

Archivar un grupo de investigación. Archiva todos los proyectos de investigación que tienen a ese grupo como principal.

RF44_EI_ Desarchivar grupo de investigación.

En caso que se necesite volver a utilizar un grupo de investigación, puede seleccionarse de la lista de grupos de investigación archivados, y se vuelve a recobrar el vínculo con todos los elementos tenía anteriormente. Los elementos que estén archivados y tenían asociación con el grupo permanecerán archivados.

**RF45_EI_ Listar grupos de investigación.**

Mostrar una lista de todos los grupos de investigación. Debe poder aplicarse filtros para la búsqueda. Incluir el filtro de grupos priorizados y de grupos no priorizados. Incluir el filtro de grupos registrados y no registrados.

RF46_EI_ Ver detalles generales del grupo de investigación.

Mostrar datos generales del grupo de investigación, así como los datos generales de los proyectos que incluye.

RF47_EI_ Valorar propuesta de grupo de investigación.

Se listan las propuestas de grupos de investigación que ha presentado. En dependencia del rol del usuario, se presentan las propuestas que puede aprobar.

RF48_EI_ Revisar propuesta de grupo de investigación.

Muestra los detalles del grupo de investigación propuesto, así como los datos de todos los estados por los que ha pasado y quiénes han aprobado el grupo hasta el momento. En este momento es que se podrá rechazar, aprobar o pedir reelaboración de la propuesta. En este momento es que se podrá rechazar, aprobar o pedir reelaboración de la propuesta. En el caso de que se pida reelaboración, se pondrán comentarios en los campos a modificar. En el caso que se rechace, se insertarán detalles que expliquen el rechazo de la propuesta.

RF49_EI_ Reelaborar propuesta de grupo de investigación.

Funciona igual que el modificar, pero la diferencia es que se mostrarán los comentarios que se hicieron a la propuesta.

RF50_EI_ Registrar grupo de investigación.

Se oficializa el proyecto investigativo.

RF51_EI_ Listar los proyectos de investigación asociados al grupo de investigación.

Listar los proyectos de investigación asociados al grupo, clasificándolos en los que lo tienen como grupo principal y los que lo tienen como grupo secundario.

RF52_EI_ Crear proyecto de investigación.

Insertar los datos necesarios para crear un proyecto investigativo. Deberá validar que el nombre del nuevo elemento no esté en uso actualmente o se encuentre entre los archivados.

RF53_EI_ Insertar investigadores al proyecto de investigación.

Insertar los recursos humanos al proyecto de investigación. Deberá validar que no se duplique un investigador insertado en la estructura.

RF54_EI_ Eliminar investigadores del proyecto de investigación.

Eliminar a un investigador de un proyecto de investigación. Deberá mantenerse el registro de que ese investigador trabajó en el proyecto por tanto tiempo, así como las actividades que realizó.

Si el investigador es líder de proyecto, no podrá eliminarse hasta que haya sido relevado del rol.

**RF55_EI_ Listar los recursos humanos del proyecto de investigación.**

Lista todos los investigadores que se encuentran en el proyecto de investigación.

RF56_EI_ Gestionar plan de trabajo del proyecto de investigación.

Gestionar las tareas del plan de trabajo del proyecto de investigación. Solamente podrá existir un plan de trabajo por cada proyecto de investigación.

RF57_EI_ Crear recurso material.

Crear un recurso material. Deberá validar que el nombre del nuevo elemento no esté en uso actualmente.

RF58_EI_ Modificar recurso material.

Modificar un recurso material.

RF59_EI_ Ver detalles de recurso material.

Mostrar detalles de un recurso material.

RF60_EI_ Listar los recursos materiales.

Mostrar un listado con todos los recursos materiales.

RF63_EI_ Crear clasificación de proyecto de investigación.

Crear una nueva clasificación de proyecto de investigación. Deberá validar que el nombre del nuevo elemento no esté en uso actualmente.

RF64_EI_ Modificar clasificación de proyecto de investigación.

Modificar una clasificación de proyecto de investigación.

RF65_EI_ Ver detalles de clasificación de proyecto de investigación.

Mostrar detalles de una clasificación de proyecto de investigación.

RF66_EI_ Listar las clasificaciones de proyecto de investigación.

Mostrar un listado con todas las clasificaciones de proyecto de investigación.

RF73_EI_ Crear problema.

Insertar los datos del problema. Deberá validar que el nombre del nuevo elemento no esté en uso actualmente o se encuentre entre los archivados.

RF74_EI_ Modificar problema.

Modificar los datos de un problema.

RF75_EI_ Ver detalles de problema.

Mostrar detalles de problema.

RF76_EI_ Listar problemas.

Se muestran todos los problemas y los elementos a los que se relaciona.

**RF77_EI_ Modificar proyecto de investigación.**

Modificar los datos de un proyecto de investigación.

RF78_EI_ Archivar proyecto de investigación.

Archivar un proyecto de investigación.

RF79_EI_ Desarchivar proyecto de investigación.

En caso que se necesite volver a utilizar un proyecto de investigación, puede seleccionarse de la lista de proyecto de investigación archivados, y se vuelve a recobrar el vínculo con todos los elementos tenía anteriormente. Los elementos que estén archivados y tenían asociación con el proyecto permanecerán archivados.

RF80_EI_ Listar proyecto de investigación.

Mostrar una lista de todos los proyectos de investigación. Debe poder aplicarse filtros para la búsqueda. Incluir el filtro de proyectos registrados y no registrados.

RF81_EI_ Ver detalles generales del proyecto de investigación.

Mostrar datos generales del proyecto de investigación.

RF83_EI_ Revisar propuesta de proyecto de investigación.

Muestra los detalles del proyecto de investigación propuesto, así como los datos de todos los estados por los que ha pasado y quiénes han aprobado el proyecto hasta el momento. En este momento es que se podrá rechazar, aprobar o pedir reelaboración de la propuesta. En el caso de que se pida reelaboración, se pondrán comentarios en los campos a modificar. En el caso que se rechace, se insertarán detalles que expliquen el rechazo de la propuesta. Esta misma funcionalidad será ejecutada por los evaluadores anónimos, pero no podrán rechazar, aprobar ni pedir reelaboración, sino solamente revisar y dar sus comentarios.

RF85_EI_ Valorar propuesta de proyecto de investigación.

Se listan las propuestas de proyectos de investigación que ha presentado. En dependencia del rol del usuario, se presentan las propuestas que puede aprobar.

RF86_EI_ Reelaborar propuesta de proyecto de investigación.

Funciona igual que el modificar, pero la diferencia es que se mostrarán los comentarios que se hicieron a la propuesta.

RF87_EI_ Registrar proyecto de investigación.

Se oficializa el proyecto investigativo.

RF89_EI_ Crear clasificación de resultado.

Crear una nueva clasificación de resultado de un elemento de la estructura investigativa.

RF90_EI_ Modificar clasificación de resultado.

Modificar clasificación de resultado de un elemento de la estructura investigativa.



RF91_EI_ Ver detalles de clasificación de resultado.

Mostrar detalles de clasificación de resultado de un elemento de la estructura investigativa.

RF92_EI_ Listar clasificación de resultados.

Listar los tipos de resultados de los elementos de la estructura investigativa.



Anexo 3: Guía de preguntas

- ¿Cuáles son los problemas existentes en la DI para la gestión de la estructura investigativa?
- ¿Cómo se realiza la gestión de la información relacionada con los elementos de la estructura investigativa?
- ¿Cuáles son las actividades que se realizan en el proceso de gestión de la estructura investigativa?
- ¿Quiénes interactúan en el proceso?
- ¿El nuevo sistema se integrará al Sistema de Gestión Universitaria?
- ¿Algún otro sistema realiza las funcionalidades que se quieren lograr con la nueva solución informática?
- ¿Qué datos son gestionados sobre cada una de los elementos de la estructura investigativa?
- ¿Quién crea los elementos de la estructura investigativa?
- ¿Quiénes tienen permiso para modificar los elementos de la estructura investigativa?
- ¿Quién registra los elementos de la estructura investigativa después de ser aceptados?
- ¿Cómo se desarrolla el proceso de control de los grupos y proyectos en la DI?



Anexo 5: Descripción de las tablas de la base de datos

Para el diseño del modelo de datos se empleó una nomenclatura de las estructuras planteadas, lo cual permite que se establezca una homogenización entre los diseños de las bases de datos, estableciéndose por el nombre de los objetos de la base de datos y la información que se debe almacenar en cada una, teniendo como resultados una mayor consistencia de los datos. Como patrón que los nombres de los diferentes objetos generados en el diseño de la base de datos quedan establecidos de la siguiente forma:

Tabla 9: Nomenclatura de las estructuras.

Objeto	Nomenclatura	Ejemplos
Esquemas	sq_+<texto>	sq_estructura_investigativa
Tablas de entidades	tb_d+<texto>	tb_delemento
Tablas de nomencladores	tb_n+<texto>	tb_nsub_lineas
Tablas de relación	tb_r+<texto>	tb_relemento
Campo de llave primaria	id_+<texto>	id_elemento
Función	fn_+<funcionalidad> <entidad>_<operación>	fn_buscar_persona

A continuación se describen las tablas del modelo físico de datos el cual contiene un total de 30 tablas, entre ellas cinco tablas de datos, 12 tablas de nomencladores y 13 tablas de relación.

Tabla 10: Tablas de datos

Nombre de la tabla	Descripción
tb_delemento	Encargada de guardar toda la información relacionada con los elementos de la estructura investigativa, así como el responsable de cada uno de ellos, el tipo de elemento que representa cada uno y el área en la que estará ese elemento.
tb_dpersona	Contiene la información necesaria para cada uno de los integrantes de cada elemento de la estructura investigativa, de los cuales se almacena si es investigador o si es evaluador anónimo.
tb_dcliente	Tiene la información de los clientes de la estructura investigativa, el cual puede ser una entidad o una persona en específico.
tb_dtarea	Almacena la información de cada una de las tareas asignadas dentro de una estructura investigativa.
tb_dtrazabilidad_elemento	Contiene información sobre los cambios de estados que se les realizan a cada uno de los elementos y quien hizo este cambio de estado.



Tabla 11: Tablas de nomencladores.

Nombre de la tabla	Descripción
tb_nclasificacion_linea tb_nclasificacion_proyecto tb_ncontacto tb_nestado tb_nmodulo tb_nnotas tb_nproblema tb_nrecurso_material tb_nresultado tb_nsub_linea tb_ntipo_elemento tb_nclasificacion_resultado	Estas tablas representan la configuración del sistema, los cuales brindarán información a las tablas de datos para su conformación.

Tabla 12: Tablas de relación.

Nombre de la tabla	Descripción
tb_relemento_relemento	Representa una relación de mucho a mucho recursiva, ya que un elemento puede estar contenido dentro de otro elemento. Ejemplo: una línea de investigación contiene grupos de investigación, y un grupo de investigación contiene proyectos de investigación.
tb_relemento_rresultado	Contiene los elementos y sus respectivos resultados, esta tabla se obtiene de la relación entre tb_delemento y tb_nresultado.
tb_relemento_rsub_linea	Contiene la información de las sub-líneas asociadas a cada una de los elementos de tipo línea de investigación.
tb_rentidad_externa_relemento	Contiene las entidades asociadas a los elementos, se obtiene de la relación de mucho a mucho entre elemento y estructura, esta última se ubica en el núcleo del SGU.
tb_rpolo_cientifico_relemento.	Contiene los polos científicos asociados a cada uno de los elementos de investigación, se obtiene de la relación de mucho a mucho entre elemento y estructura, esta última se ubica en el núcleo del SGU.
tb_rmodulo_restado.	Se crea de la relación entre módulo y estado, esta contiene la información sobre los estados y sus respectivos módulos que van a utilizar estos estados.
tb_rnotas_relemento	Se crea de la relación entre notas y elemento, esta contiene la información sobre las notas que están que se les asignan a cada uno de los elementos en el proceso de revisión.



tb_rpersona_relemento tb_rpersona_rentidad tb_rproblema_relemento tb_rrecurso_material_relemento tb_rtarea_relemento tb_rtarea_rpersona	Estas tablas son utilizadas por diferentes esquemas de la base de datos del sistema de investigaciones, las cuales se obtienen de relaciones de muchos a muchos.
--	--

Para la solución informática se tienen presente otras tablas de datos que no están incluidas en el esquema de sq_estructura_investigativa (ver figura modelo lógico, anexo 5), a continuación se realiza una breve descripción de estas tablas.

Tabla 13: Tablas del núcleo.

Nombre de la tabla	Descripción
tb_dpersona	Brinda la información de las personas que están vinculadas a los elementos de investigación, de los cuales se almacenan información que solo va a ser utilizada por el esquema sq_estructura_investigativa.
tb_destructura	Brinda información de todas las estructuras con las que la universidad mantiene acuerdos, por lo que facilitará el almacenamiento de la información relacionada con estas estructuras que estarán asociadas a los elementos de investigación.
tb_nrol	Permite restringir el acceso a funcionalidades por el rol en que el usuario está autenticado, por lo que solo tendrá acceso a las funcionalidades que al rol se le hayan definido.