

Sistema Informático para la Automatización de los Procesos de Formación e Investigación del Centro de Consultoría de la UCI

Universidad de las Ciencias Informáticas



FACULTAD 15

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

Autores

Denis Boizan Romero

Carlos Ernesto Sanregre Castellanos

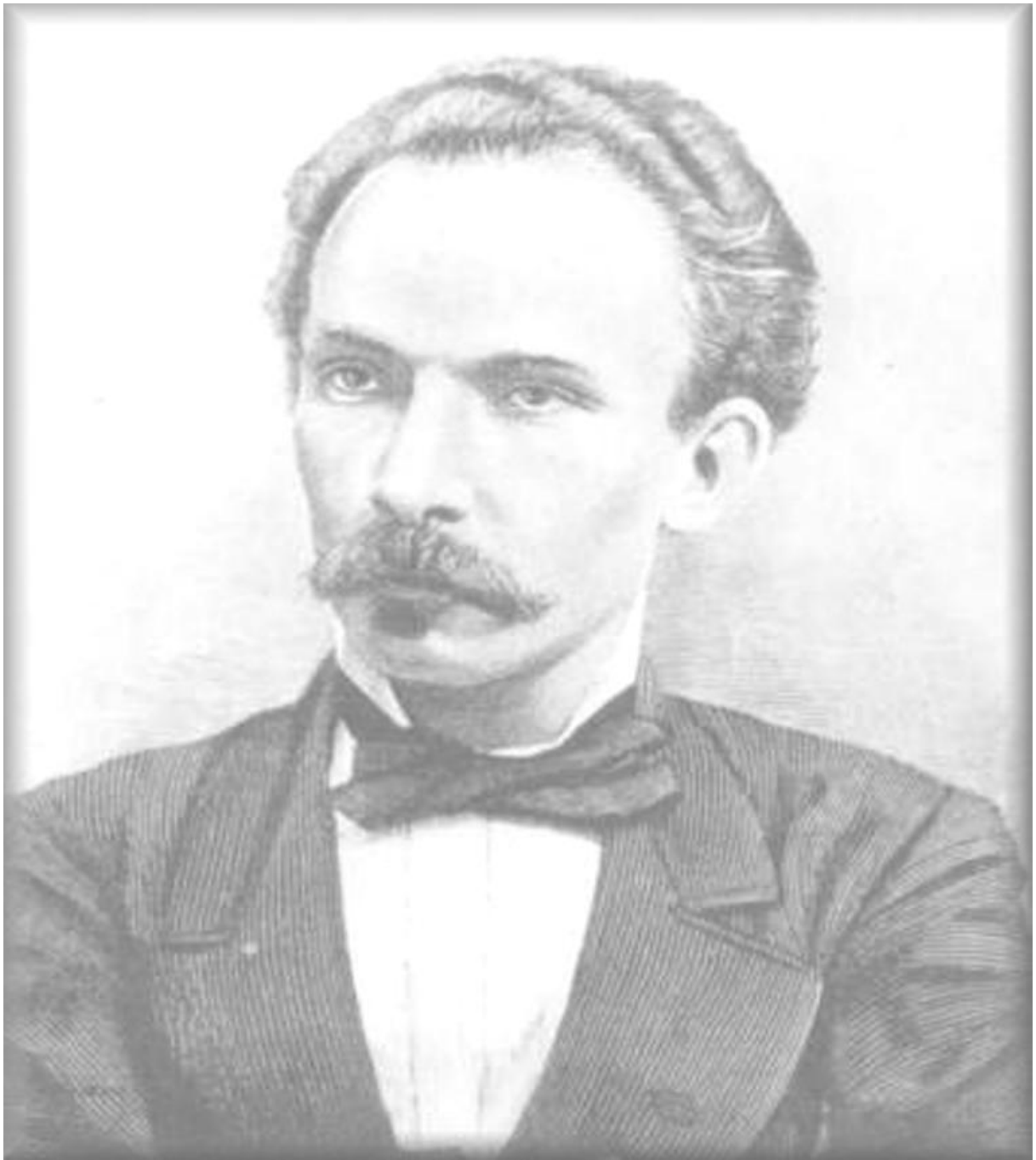
Tutores

Ing. Ailenis Macías Hernández

Ing. Dorisbel Muro Fumero

Ciudad de La Habana, Mayo 2010

Año 52 de la Revolución



**"EL ÚNICO AUTÓGRAFO DIGNO DE UN HOMBRE ES EL QUE DEJA
ESCRITO CON SUS OBRAS."**

DECLARACIÓN DE AUTORÍA

Declaramos que somos autores de este trabajo y autorizamos a la Biblioteca de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con el mismo.

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

Denis Boizan Romero

Firma del Autor

Carlos Ernesto Sanregre Castellanos

Firma del Autor

Ing. Ailenis Macías Hernández

Firma del Tutor

Ing. Dorisbel Muro Fumero

Firma del Tutor

Denis:

En especial a mis padres, por la confianza que siempre depositaron en mí, por su infinito amor, por llevarme a ser todo lo que soy, por respetarme siempre, por contar conmigo para todo, por no defraudarme nunca, por quererlos tanto. A mis hermanitas por estar ahí siempre para mí. A mi sobrino por los momentos tan felices que nos ha dado en la casa. A mis abuelos paternos que tanto han hecho por mí y por los míos, a mis abuelos maternos aunque uno no pudo vivir para verme aquí. A todos los que han creído en mí.

Carlos Ernesto:

En especial a mis padres, que siempre han confiado en mí, por contar siempre conmigo, por darme una buena educación y enseñarme el camino correcto que me llevó a ser la persona que soy, por estar siempre orgullosos de mí y por el amor que he recibido de ellos. A mis hermanas en especial a Daimis, por estar siempre conmigo, apoyándome, cuidándome y queriéndome. A mi abuela materna que siempre ha confiado en mí y que ha sido mi segunda madre. A mi familia por quererme mucho. A todas las personas que siempre creyeron en mí.

AGRADECIMIENTOS

Denis:

A las amistades que he hecho aquí durante estos 5 años, a mi hermano y compañero de tesis Carlos por ayudarnos el uno al otro a salir de esta, a los maestros del ExtJS, el Javico y el Dash. A Dianly por ser una estrella. A 2 megas por ser una amiga en las buenas y malas. A los profesores que fueron realmente finos, los que en verdad se lo merecen. A nuestras tutoras Ailenis y Dorisbel por ayudarnos en la realización de esta larga tarea, a los socios del barrio, los que crecieron conmigo, en fin, a todos con los que he compartido alguna vez en cualquier momento durante mi estancia aquí en la universidad.

Carlos Ernesto:

A mi hermano menor y compañero de tesis Denis, por ayudarnos en los momentos difíciles. A Dianly por ser mi hermano y un tutor especial, por darme siempre buenos consejos y ser un excelente profesor de base de datos. A mis profesores de ExtJS, Javier y Denier, por enseñarme todo lo que sé. A nuestras tutoras Ailenis y Dorisbel por ayudarnos a realizar esta larga tarea. A todos los amigos que he encontrado durante estos 5 años, por compartir siempre conmigo, por estar siempre en las buenas y en las malas. A todas aquellas personas que han compartido conmigo.

Resumen

El Sistema Informático para la Automatización de los Procesos de Formación e Investigación del Centro de Consultoría de la Universidad de las Ciencias Informáticas (UCI) surge debido a la necesidad de gestionar de manera más rápida, segura y eficaz los procesos que se desarrollan actualmente en dicho centro. Fue desarrollado utilizando los lenguajes PHP del lado del servidor con CodeIgniter como framework y JavaScript del lado del cliente. Se utilizó PostgreSQL como gestor de base de datos y ExtJS, una potente librería JavaScript para el diseño de las interfaces y el trabajo con las validaciones. Este sistema informático soluciona los problemas existentes en la gestión de los procesos en esta área de la universidad, que actualmente se desarrolla de forma manual imposibilitando la realización de reportes y actualización de la información con la calidad y rapidez que se necesita. Se esperan como resultados más relevantes la implantación del sistema de forma satisfactoria y las mejoras en los procesos de formación e investigación del Centro de Consultoría de la UCI.

Índice

Resumen	¡Error! Marcador no definido.
Introducción	9
CAPÍTULO 1	¡Error! Marcador no definido.
FUNDAMENTACIÓN TEÓRICA	12
1.1 Introducción.....	¡Error! Marcador no definido.
1.2 Centro de Consultoría.....	12
1.2.1 Investigaciones	13
1.2.2 Superación.....	14
1.2.3 Formación.....	14
1.2.4 Categorización	15
1.3 Sistemas de Gestión.....	15
1.4 Estado del arte	15
1.4.1 Sistemas Internacionales	15
1.4.2 Sistemas Nacionales.....	17
1.4.3 Sistemas en la UCI	18
1.5 Justificación de la necesidad del sistema propuesto.....	18
1.6 Metodologías de Desarrollo de Software	19
1.7 Tendencias y tecnologías actuales.....	20
1.7.1 Lenguajes y plataformas de programación para la Web.....	21
1.7.2 Servidores Web	21
1.7.3 Gestores de Base de Datos	22
1.7.4 Frameworks	22

1.7.5 Otros lenguajes utilizados y arquitectura.....	23
1.7.6 Arquitectura en 3 capas	25
1.8 Fundamentación de la solución propuesta.....	26
1.9 Conclusiones.....	26
CAPÍTULO 2.....	27
CARACTERÍSTICAS DEL SISTEMA	28
2.1 Introducción.....	28
2.2 Requerimientos del sistema.....	28
2.2.1 Requerimientos funcionales	28
2.2.2 Requerimientos no funcionales	31
2.3 Análisis del modelo de diseño propuesto.....	32
2.4 Estándar de código utilizado en la implementación	34
2.4.1 Nomenclatura de las clases	34
2.4.2 Nomenclatura de las clases según el tipo	34
2.4.3 Nomenclatura de los métodos o funciones.....	34
2.4.4 Nomenclatura de las variables	34
2.5 Descripción de la implementación del subsistema Vinculación a la Docencia.....	34
2.6 Descripción de la implementación del subsistema Grupo de Desarrollo	35
2.7 Descripción de clases y operaciones del sistema	36
2.7.1 Clases Modelo	36
2.7.2 Clases Controladoras.....	36
2.8 Integración de los módulos del sistema	39
2.9 Conclusiones.....	40

CAPÍTULO 3	41
VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	41
3.1 Introducción.....	41
3.2 Pruebas de software.....	41
3.3 Niveles de Prueba	41
3.4 Métodos de Prueba	43
3.4.1 Pruebas de Caja Negra o Funcionales.....	43
3.4.2 Pruebas de Caja Blanca o Estructurales	46
3.5 Validación del modelo de diseño propuesto.....	51
3.6 Resultados generales de la implantación del sistema.....	54
3.7 Conclusiones.....	54
Conclusiones generales	55
Recomendaciones	56
Referencias Bibliográficas	57
Anexos	60
Glosario de términos	74

Introducción

El desarrollo de la sociedad actual está estrechamente relacionado con el avance de las Tecnologías de la Información y las Comunicaciones (TIC); al mismo tiempo se hace más dependiente de los medios informáticos, por ello, cada día el uso de las computadoras es más necesario en el mundo que nos rodea, y con ello la informatización de numerosas actividades y procesos que antes se realizaban de forma manual. Cuba no podía quedar fuera de dicho progreso, y el Comandante en Jefe Fidel Castro Ruz tuvo la visión de crear una entidad que garantizara el desarrollo de la informática en el país. Así surge la UCI, creada con el objetivo de lograr avances en la rama de la informática, primera ciudad digital del país como prototipo de una posible ciudad del futuro, donde todos o la mayoría de los procesos que ocurrieran en ella serían controlados por sistemas automatizados que se producirían en la institución. La UCI cuenta con centros productivos vinculados a varias esferas de la sociedad, uno de ellos es el Centro de Consultoría.

El Centro de Consultoría se divide en áreas de procesos que se encargan de su funcionamiento:

- ✓ Gestión del conocimiento: Desarrollo académico y científico.
- ✓ Gestión de recursos humanos y economía.

El Centro de Consultoría tiene un gran impacto como generador de recursos financieros y como formador del capital intelectual. Pretende establecer una cartera de cursos sobre metodología y tecnologías SOA/BPM, también crear un perfil SOA/BPM para la Universidad con al menos 2 salidas: como Analistas de Procesos y Arquitectos.

En estos momentos, en el Centro de Consultoría se trabaja en el análisis y diseño de los procesos de formación e investigación de dicho centro, como parte de una tesis de diploma; este aspecto es esencial para determinar los elementos necesarios, que paralelamente permitirán la implementación de un sistema informático para automatizar estos procesos.

Actualmente el control de los procesos de formación e investigación en el Centro de Consultoría se realiza manualmente, almacenando la información en documentos físicos, resultando en algunos casos engorrosa la gestión de dicha información para cualquier actividad que se desee realizar, afectándose así la agilidad y calidad del proceso. Existen dificultades para la inscripción en los cursos. La búsqueda y recuperación de la información para la realización de los reportes es lenta, no existe visibilidad de las

publicaciones científicas ni de los trabajos presentados en eventos, no se cuenta con un registro de los premios obtenidos en eventos científicos, tampoco se tiene almacenada la información sobre las capacitaciones recibidas ni las ofertadas por el centro.

Debido a lo anteriormente expuesto, surge el siguiente **problema a resolver**: La deficiente gestión de los procesos de formación e investigación en el Centro de Consultoría de la UCI. El presente trabajo tiene como **objeto de estudio**: Los procesos de formación e investigación, y como **campo de acción**: Los procesos de Gestión de tesis, Currículum vitae, Banco de problemas, Cursos y Vinculación a la docencia en el Centro de Consultoría de la UCI.

Se ha planteado como **objetivo general**: Desarrollar un sistema informático que gestione los procesos de formación e investigación en el Centro de Consultoría de la UCI. Como **idea a defender** se tiene que: El desarrollo de un sistema informático que automatice la Gestión de tesis, Currículum vitae, Banco de problemas, Cursos y Vinculación a la docencia garantizará en gran medida que la organización, control, procesamiento y actualización del proceso de formación e investigación del Centro de Consultoría de la UCI se realice de manera más rápida, segura y efectiva.

Las **tareas investigativas** trazadas son:

- Realización del estudio de las principales tecnologías que existen para el desarrollo de aplicaciones Web.
- Análisis de los principales estándares de desarrollo para aplicaciones Web.
- Implementación de una estructura de almacenamiento para los datos.
- Desarrollo de un sistema de búsquedas cruzadas y dinámicas.

Con el desarrollo de este trabajo se espera como **posible resultado** un sistema informático que automatice la gestión de los procesos de formación e investigación del Centro de Consultoría de la UCI. Los métodos científicos utilizados fueron:

Entrevista: Entrevista a los encargados de gestionar los procesos de formación e investigación.

Analítico Sintético: Durante el análisis de los documentos y la información existente acerca de los sistemas de gestión de los procesos de formación e investigación.

Encuesta: Utilizado para obtener el grado de cumplimiento de los requerimientos de la aplicación por parte de otros sistemas de gestión de postgrado que existen en la UCI.

Para realizar una descripción detallada, este documento muestra el resultado de la investigación de la siguiente forma:

El **Capítulo 1** contiene la fundamentación teórica del trabajo. Aborda aspectos importantes del Centro de Consultoría de la UCI, dirigidos principalmente al proceso de formación e investigación. Se valoran algunos sistemas similares que se utilizan en el mundo, en Cuba y en la UCI. Se hace un análisis de las tendencias, tecnologías y las metodologías que se utilizan en la actualidad para el desarrollo de aplicaciones web, señalando cuáles de estas son más apropiadas para el desarrollo del sistema informático que se propone.

El **Capítulo 2** contiene la descripción de las características del sistema informático. Aborda acerca de los requerimientos funcionales y no funcionales. Se describe la estructura de las clases y algunas de las principales funcionalidades que se realizan. Se hace referencia a los principales patrones de desarrollo de software utilizados para el desarrollo de la implementación, los estándares de codificación para la nomenclatura de las clases, los métodos y las variables, así como a los artefactos generados durante esta fase.

El **Capítulo 3** contiene la validación de la solución propuesta. Se realiza en este capítulo lo relacionado con el proceso de pruebas aplicables al sistema informático luego de su terminación: pruebas de caja negra y caja blanca, grafo causa efecto, tamaño operacional de las clases. Por último se muestran los resultados de la implantación del sistema informático.

1.1 Introducción

En este capítulo se aborda la fundamentación teórica del trabajo y se realiza un análisis de los sistemas similares existentes vinculados al campo de acción. Además se presenta un amplio estudio de las tendencias y tecnologías actuales, de las cuales se hace una selección para el desarrollo posterior del sistema informático.

1.2 Centro de Consultoría

El Centro de Consultoría es una institución creada en la UCI con la misión de ofrecer los servicios de consultoría tecnológica a partir del capital intelectual que se desarrolla en esta universidad, como una actividad generadora de recursos financieros para el país. Tiene como objetivos específicos brindar servicios de consultoría en tecnologías informáticas a instituciones nacionales y extranjeras, formar consultores altamente calificados en tecnologías informáticas, comprometidos con los principios de la Revolución Cubana y al servicio de esta, y constituir un centro de referencia en la UCI sobre las tecnologías asociadas a las líneas de consultoría propias (Arquitectura Orientada a Servicios (SOA) y Gestión de Procesos de Negocio (BPM)). Está estructurado de la siguiente forma:

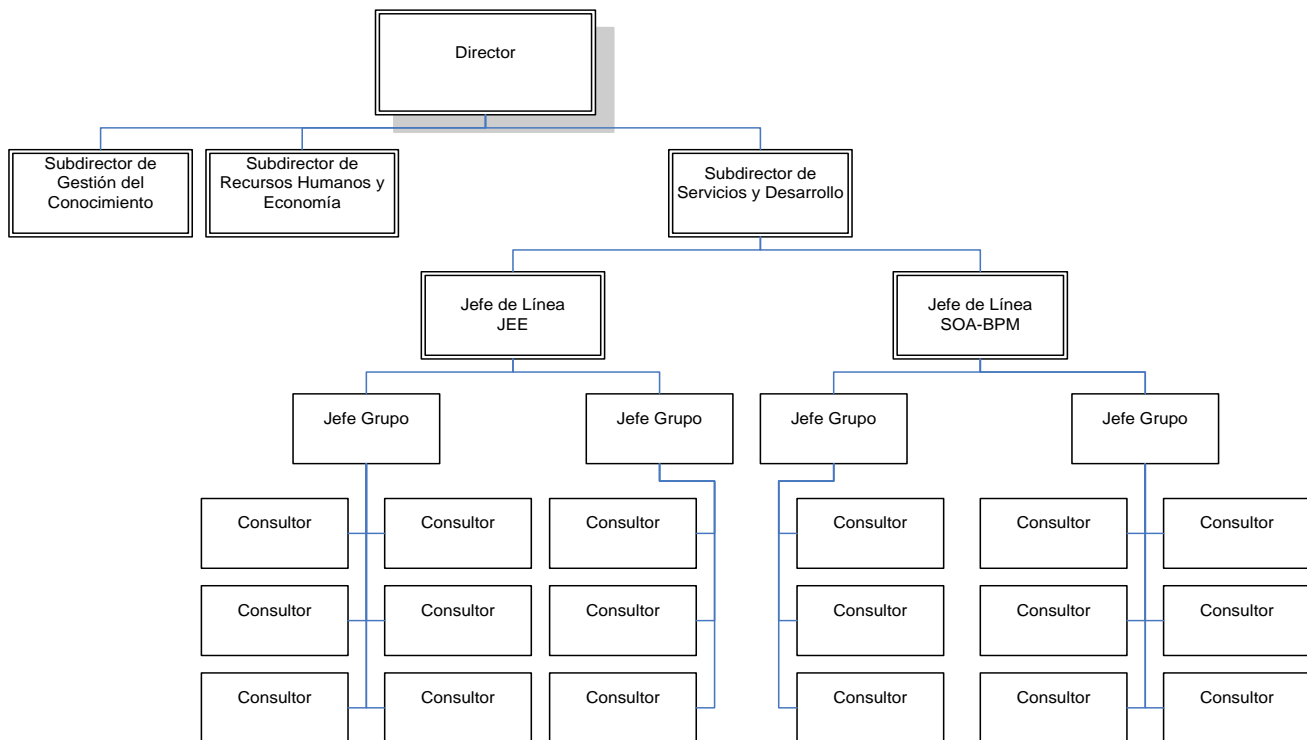


Fig.1.1 Estructura del Centro de Consultoría.

El centro cuenta con distintas áreas de procesos. El área de Gestión del Conocimiento: Desarrollo académico y científico, comprende dentro de sus actividades todo lo relacionado con los procesos de formación e investigación del Centro de Consultoría de la UCI, a los cuales se hará referencia a continuación, destacando principalmente los elementos que constituyen parte del objetivo del trabajo, como son el banco de problemas, las tesis, el curriculum y los cursos.

1.2.1 Investigaciones:

Comité Científico: Crear, organizar y dirigir el funcionamiento del Comité Científico del Centro de Consultoría. Crea y dirige la política científica del centro y aprueba la estrategia de investigación. Mantiene actualizado el **banco de problemas**. Dicho comité tiene la responsabilidad de aprobar los proyectos de investigación que presenten las líneas de consultoría, garantizando la coherencia de los mismos y que tributen a los objetivos estratégicos del centro. Además aprueba los artículos que serán publicados y sugiere la participación en eventos nacionales o extranjeros que promuevan el intercambio y generalización de resultados y visibilidad del Centro de Consultoría.

Proyectos de investigación: Surgen como necesidad de los lineamientos estratégicos de cada línea de consultoría y generan un resultado de valor para el centro. Se proponen al Comité Científico para su aprobación. Una vez aprobados los proyectos deben ser inscritos formalmente. Los proyectos son controlados periódicamente y se exige su correcta ejecución para garantizar la obtención de los resultados esperados. Los proyectos de investigación pueden formalizarse a través de o tributar a trabajos de diploma, tesis de maestría y/o doctorado. Su visibilidad se logra a través de la publicación de artículos y presentación de ponencias en eventos científicos. Los resultados que se obtengan deben ser registrados legalmente.

Trabajos de diploma: Constituir un Comité de **tesis**. Crear Comité de revisión de tesis. Conformar tribunales de tesis. Realizar cortes de tesis. Asegurar que los trabajos de diploma estén orientados a las líneas de investigación aprobadas por el centro. Comunicar a las facultades el resultado de los cortes de tesis y defensa final. Garantizar representatividad ante el comité central de tesis de la UCI y acatar los lineamientos metodológicos que este acuerde. Garantizar la publicación de la tesis.

1.2.2 Superación:

Cursos de postgrado: Coordinar los **cursos de postgrado** que se impartan a los consultores como parte del plan de desarrollo individual. Estos cursos deben estar orientados a satisfacer las necesidades de formación que se exigen para cambios de categoría, créditos lectivos en maestrías, y la producción. La formación del personal debe estar orientada a tres aspectos fundamentales: tecnología, idiomas extranjeros y ciencias empresariales.

Maestría y Doctorado: Promover la incorporación en maestrías de los miembros del Centro de Consultoría y controlar su desempeño en las mismas. Garantizar que las tesis se encuentren en función de las líneas de investigación del centro y que aporten resultados concretos como activos de conocimiento tácito y explícito.

1.2.3 Formación:

Cursos optativos, segundo perfil, especialización: Organizar un segundo perfil, cursos, diplomados o especializaciones afines a las líneas propias del Centro de Consultoría. Conformar el diseño **curricular** de cada una de estas modalidades de formación, preparar las asignaturas, asignar profesores e impartir las clases.

1.2.4 Categorización:

Categorías docentes: Organizar los tribunales de categorización docente. Preparar a los miembros del Centro de Consultoría para los procesos de categorización en función del plan de desarrollo individual.

Certificaciones: Certificar internacionalmente a los especialistas del centro.

1.3 Sistemas de Gestión

Un Sistema de Gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización; propicia el manejo de grandes volúmenes de datos, almacenarlos y transmitirlos en muy poco tiempo. ¿Por qué son necesarios? Las instituciones del siglo XXI se enfrentan a muchos retos significativos, entre ellos: rentabilidad, competitividad, velocidad de los cambios, crecimiento y tecnología. La implementación de un sistema de gestión eficaz puede ayudar a: reducir costos, lograr mejoras continuas y aumentar la satisfacción de clientes y partes interesadas. (1)

1.4 Estado del arte

Actualmente no existe un sistema que automatice los procesos de formación e investigación del Centro de Consultoría de la UCI. Existen varios sistemas que gestionan la formación, en ocasiones integrados con la investigación.

A nivel internacional también se pueden citar ejemplos, que no responden a las políticas y necesidades actuales del Centro de Consultoría. Los procedimientos establecidos en el centro difieren en algunos casos al resto de las áreas en la UCI.

1.4.1 Sistemas Internacionales

Sistema de Administración Académica Postgrado – SAACP

El Sistema de Administración de Postgrados, es un sistema de información desarrollado en Microsoft .Net, que permite apoyar el proceso de gestión académica de los programas a nivel de postgrados que se imparten en la Escuela Superior Politécnica del Litoral (ESPOL).

Cada programa de postgrado, en forma descentralizada puede administrar sus propios procesos de planificación académica, registros de estudiantes e ingreso de resultados del período (calificaciones, asistencia, convalidaciones, etc.) (2)

La aplicación Windows está dirigida a las unidades académicas que dictan los programas de postgrados y administran la siguiente información:

- Parámetros de Configuración.
- Profesores.
- Materias.
- Información de Programas Académicos.
- Información de Aspirantes.
- Información de Estudiantes.
- Proceso de Graduación.
- Reportes varios.

TICBioMed

Es una plataforma de cooperación entre Empresas, Universidades, Proveedores de salud, Fundaciones y Centros tecnológicos de la Región de Murcia para la materialización de la innovación tecnológica en medicina, asistencia sanitaria y biología. (3)

Sus fines fundacionales son:

- Poner en valor la capacidad de innovación tecnológica de sus miembros.
- Fomentar la investigación, desarrollo e innovación en las Tecnologías de la Información y las Comunicaciones (TIC) aplicadas a la Biología y la Medicina.
- Promover la colaboración entre centros de investigación, empresas, administración y profesionales, con el fin de aportar soluciones prácticas a problemas reales.
- Impulsar proyectos TIC bio-sanitarios y la transferencia de tecnología entre los centros de investigación y las empresas. Dar visibilidad a los proyectos e iniciativas de la Red.

- Potenciar una formación de calidad para conseguir profesionales e investigadores de alta calificación.
- Dar una respuesta coordinada a las necesidades comunes de los miembros de la red.

1.4.2 Sistemas Nacionales

El **Departamento de Geociencias del ISPJAE**, acumula una amplia experiencia en la superación postgraduada de los profesionales vinculados con las Ciencias de la Tierra, en el territorio nacional y fuera de sus fronteras geográficas. Desde la década de los 90 y ante la necesidad de los especialistas de capacitarse en el uso de las técnicas de percepción remota y sistemas de información geográfica, su colectivo de Geomática, se trazó la tarea de diseñar y poner en práctica un sistema de superación postgraduada relacionado con estas tecnologías. Este sistema previó, no solamente las necesidades de los geocientistas, sino que además, tuvo en consideración las emanadas de un amplio grupo de especialistas, de las más disímiles profesiones, que han buscado las vías para su capacitación y en muchos casos, su reorientación profesional. (4)

En la **Universidad Central de las Villas**, existe un sistema que permite llevar el control de las actividades de postgrado de un Centro de Educación Superior (CES), tanto a nivel central como desagregado por sus facultades. En su versión actual está ajustado al nuevo Reglamento de Postgrado de la Educación Superior, en vigor desde finales de 2004. Permite controlar la planificación, ejecución, y terminación tanto de las actividades de superación profesional de postgrado (cursos, entrenamientos y diplomados), como al postgrado académico (maestrías, especialidades y doctorados). Se posibilita la planificación estratégica e inmediata de los postgrados, así como el control de las matrículas y evaluaciones. El sistema brinda una serie de informes predeterminados como planificaciones de cursos, listado de matrícula por cursos, actas de examen, informes finales de postgrados concluidos, sábanas de notas de postgrados múltiples, certificados de evaluación de cursos, entrenamientos y diplomados, certificaciones de estudios terminados de maestrías y especialidades. Es posible también llevar el control de los doctorados que se desarrollan en el CES o alguna entidad autorizada.

1.4.3 Sistemas en la UCI

Sistema para la Gestión de Postgrados en la UCI

Centrado en la actualización del análisis y diseño realizado anteriormente en la tesis Análisis y Diseño del Sistema para la Gestión de Postgrados en la UCI, y en la implementación del sistema con el objetivo de mejorar la gestión de la información y minimizar el tiempo de realización de reportes, accediendo a la información en cualquier momento y desde cualquier lugar dentro de la UCI. Brinda especial atención a la autenticación de los usuarios y sus privilegios en el sistema, así como la gestión de algunas actividades dentro de las que se pueden incluir los cursos y la matrícula en estos.

Sistema de Gestión de Tesis Facultad 2

Tiene como objetivo central desarrollar una aplicación web que permita optimizar el trabajo y la documentación referente al Proceso de Tesis de Grado en la Facultad 2, brindando la posibilidad de llevar un control durante el transcurso de cada período sobre cada una de las tesis y otros eventos que se efectúan como parte del proceso, el cual se ejecuta hoy de forma manual, lo que influye negativamente en la calidad y efectividad del mismo. El sistema brinda una amplia información sobre todo el tema referente a las tesis de grado, garantiza la centralización de los documentos generados en función de este proceso, permite publicar y asignar los temas de tesis propuestos a los estudiantes de quinto año y mantener un seguimiento de la evaluación y evolución de dichas tesis.

Control de la Actividad de Postgrado en la Facultad 15

Tiene como principal objetivo realizar un control efectivo de las actividades de postgrado en la Facultad 15. Maneja gran parte de la información relacionada con la autenticación de usuarios y la gestión de los roles que estos desempeñan en el sistema, y prácticamente todo lo relacionado con la gestión de las cursos, maestrías y doctorados. También gestiona la matriculación en las actividades mencionadas anteriormente, y es capaz de brindar los reportes relacionados con dichos cursos y otras actividades desarrolladas que se encuentren vinculadas al postgrado.

1.5 Justificación de la necesidad del sistema propuesto

Se desarrolló una encuesta para obtener información acerca de otros sistemas que gestionen de alguna manera las actividades de postgrado en la UCI. Los resultados obtenidos fueron ponderados en función

del cumplimiento de los requerimientos funcionales del sistema informático que se desea implementar para la Automatización de los Procesos de Formación e Investigación del Centro de Consultoría de la UCI. Se realizó una encuesta sobre 6 sistemas, y los resultados arrojados fueron los siguientes:

- De los sistemas, solamente uno de ellos cumplía con al menos el 50 por ciento de los requerimientos del sistema informático que se desea implementar para el Centro de Consultoría.
- El sistema mencionado anteriormente, único que cumplía con lo señalado en al menos la mitad, algunos de estos requerimientos sólo era capaz de solucionarlos parcialmente, destacando que los procedimientos para darle solución a dicho requerimiento no eran los mismos, omitiendo en ocasiones flujos de eventos que son considerados de importancia para el sistema informático que se desea implementar.
- La mayoría de los sistemas no brinda solución a los requerimientos relacionados con los curriculums, considerados de gran importancia para la aplicación, llegando a representar un 75 por ciento sin darle solución, un 16,75 por ciento de solución total, y 8.25 por ciento de solución parcial.
- Ninguno de los sistemas gestiona la vinculación a la docencia.
- El 66.5 por ciento de los sistemas no da tratamiento al banco de problemas, un 17 por ciento totalmente, y un 16.5 por ciento de manera parcial.
- Más del 58 por ciento de estos sistemas no daba ningún tratamiento a los cursos, el 14 por ciento de forma parcial, y poco más del 25 por ciento de forma total.

Por todo lo antes expuesto, y teniendo en cuenta que no se pueden integrar los sistemas existentes, surge la necesidad de desarrollar un sistema informático que sea capaz de dar solución total a cada uno de los requerimientos que se tuvieron en cuenta para llevar a cabo la encuesta, quedando integrados en un único Sistema Informático para la Automatización de los Procesos de Formación e Investigación del Centro de Consultoría de la UCI.

1.6 Metodologías de Desarrollo de Software

La calidad en el desarrollo y mantenimiento de los sistemas informáticos se ha convertido en uno de los principales objetivos de las organizaciones, debido a que cada vez más los procesos principales dependen de sistemas informáticos para su buen funcionamiento. Una metodología para el desarrollo de

un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación que los desarrolladores de sistemas informáticos deben seguir para garantizar la calidad de un producto informático. (5)

Conocer la metodología que va a guiar el proceso de desarrollo de un software es muy importante, ya que esta permite definir los pasos a seguir durante todo el ciclo de desarrollo. En la actualidad existen diversas metodologías de desarrollo con particularidades específicas, entre las más conocidas están: Rational Unified Process (RUP) y Extreme Programming (XP).

Para el desarrollo de este trabajo se utilizó RUP, uno de los procesos más generales de los existentes actualmente. Unifica completamente al equipo de desarrollo de software y optimiza la productividad de cada uno de los miembros, aprovechando que está fundamentada en un enfoque orientado a modelos de desarrollo basado en componentes, en este caso para el desarrollo de los distintos módulos del trabajo. RUP se caracteriza por ser dirigido por casos, que definen lo que el usuario desea a partir de la captura de requisitos y la modelación del negocio. Es centrado en la arquitectura, característica que brinda una visión completa del sistema, donde se describen los procesos del negocio que son más importantes para comprenderlo, desarrollarlo y producirlo de una forma eficaz. Es iterativo e incremental permitiendo desarrollar en cada fase las iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo.

1.7 Tendencias y tecnologías actuales

Es muy importante para los productores de software estar al tanto de todo lo que ocurre en el mundo que tenga que ver con el desarrollo actual de soluciones informáticas. Se hace necesario conocer las ventajas y posibilidades que brindan las tecnologías a utilizar para la realización de un producto. En este momento existe una marcada tendencia hacia la web, no sólo por los costos reducidos que esto implica, sino por el alto nivel de flexibilidad y agilidad que permite en la implementación de un sistema. El acceso a una página web es independiente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios. Por esta razón, en vez de crear aplicaciones para Windows, GNU/Linux, u otros Sistemas Operativos, la aplicación es escrita sólo una vez y mostrada a todos los clientes a través del acceso a un servidor. En las siguientes secciones se describen las principales tecnologías y herramientas que se utilizan en la actualidad para la construcción de

aplicaciones web, especificando en cada caso la que se utilizará para el desarrollo de la solución propuesta.

1.7.1 Lenguajes y plataformas de programación para la Web

Existen varios lenguajes y plataformas de programación para la web. Algunos de los más conocidos son Java, Python, ASP y PHP.

Para el desarrollo del sistema se utilizará PHP, teniendo en cuenta que es un lenguaje de programación del lado del servidor, gratuito, rápido, seguro, con una gran librería de funciones y mucha documentación. Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web, lo que significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro fácilmente. Tiene una gran capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL y Oracle, y da soporte orientado a objetos, clases y herencia. Por otro lado, se poseen conocimientos anteriores de dicho lenguaje. (6)

1.7.2 Servidores Web

Existen distintos tipos de servidores diseñados para la web. Se pueden citar entre ellos Internet Information Server y Apache. Para el desarrollo del sistema se utilizará Apache.

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Entre sus características destacan:

- Multiplataforma
- Es un servidor web conforme al protocolo HTTP.
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona.
- Se desarrolla de forma abierta.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor. (7)

1.7.3 Gestores de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. (8) Se utilizará como gestor de la base de datos PostgreSQL.

Principales características de este gestor de bases de datos:

1. Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos,
2. Incorpora una estructura de datos array.
3. Permite la declaración de funciones propias.
4. Soporta el uso de índices, reglas y vistas.
5. Incluye herencia entre tablas.
6. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. (9)

1.7.4 Frameworks

Un framework es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable, a la que se le puede añadir las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework web, por tanto, se puede definir como un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. (10)

Ejemplos de framework son Zend, CakePHP, Symfony y CodeIgniter. Para el desarrollo de la aplicación se utilizó CodeIgniter.

CodeIgniter es un entorno de desarrollo abierto que permite crear webs dinámicas con PHP. Su principal objetivo es ayudar a que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero. Entre sus principales características destacan:

- Sistema basado en el Modelo-Vista-Controlador.
- Características completamente equipadas de la base de datos con la ayuda para varias plataformas.
- Ayuda de base de datos.
- Validación de la forma y de los datos.
- Gerencia de sesión.
- Clases de envíos de email. Soporte de agregados, protocolos múltiples (sendmail, smtp, y mail) y más.
- Biblioteca de manipulación de imágenes.

CodeIgniter es liviano, muy rápido y fácil de aprender, y además recomendable para el desarrollo en pequeños equipos de trabajo. Por otro lado, cabe destacar que era el framework del que se poseían mayores conocimientos por estudios previos a la realización de este trabajo. (10)

1.7.5 Otros lenguajes utilizados y arquitectura

HTML: Hyper Text Markup Language, es un lenguaje de programación muy sencillo que se utiliza para crear textos y páginas web. Está compuesto por etiquetas que definen la estructura y el formato del documento que verá el usuario a través de la web. Esas etiquetas son leídas por el navegador o visualizador, es decir el programa que se utiliza para navegar, y que es el que ejecuta las funciones creadas en HTML permitiendo que puedan ser visibles en la máquina.

Una de sus características es que, además del texto, permite que se creen enlaces entre distintas partes del mismo documento o entre distintas fuentes de información a través de hiperenlaces o hipervínculos, e incluso insertar otros elementos como imágenes y sonidos. (11)

CSS: Cascading Style Sheets (Hojas de Estilo en Cascada) es un formato usado en las páginas web para separar el estilo de la estructura. Es una característica del HTML que da más control a los programadores, diseñadores y usuarios de la web sobre cómo se muestran las páginas web. Se pueden entonces aplicar a cualquier página o a todas las páginas en un sitio web particular, lo que hace la codificación mucho más fácil. El término cascading (en cascada) deriva del hecho de que múltiples hojas de estilo se pueden aplicar a una misma página web. Entre las ventajas de utilizar CSS vale mencionar que:

- Se tiene el control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño. (12)

JavaScript: Es un lenguaje interpretado, del lado del cliente, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al contrario de Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia; es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Principales características:

- Es simple, no hace falta tener conocimientos de programación para poder hacer un programa en JavaScript.
- Maneja objetos dentro de una página web y sobre ese objeto se pueden definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario.
- Es dinámico, responde a eventos en tiempo real, lo que permite cambiar totalmente el aspecto de la página al gusto del usuario. (13)

ExtJS

En el mercado actualmente existen múltiples librerías de JavaScript que permiten realizar todo tipo de maravillas en el navegador web. Una de estas librerías, se llama ExtJS y tiene muchas cualidades que la hacen interesante. ExtJS incluye:

- Componentes de interfaz de usuario del alto performance y personalizables.
- Un API fácil de usar.

- Licencia código abierto y comercial.

Principales características:

- La ventana flotante que provee ExtJS es excelente por la forma en la que funciona.
- Existe un balance entre Cliente – Servidor.
- Comunicación asíncrona.
- Eficiencia de la red. (14)

1.7.6 Arquitectura en tres capas

La Arquitectura del Software o Arquitectura Lógica, es el diseño de más alto nivel de la estructura de un sistema. Consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software. Establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. (15)

Una arquitectura de software se selecciona y diseña con base en objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, adaptabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea. La arquitectura de una aplicación es la vista conceptual de la estructura de esta. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como está distribuido este código.

El diseño en tres capas trata de que todo el código no sea puesto sobre las interfaces del usuario del sistema. Con este objetivo se definen tres niveles de profundidad:

1. Capa de presentación, con las interfaces de usuarios.
2. Capa de negocio (lógica del dominio). Surge de los procesos encontrados en el análisis.

3. Capa de acceso a datos. Contiene el código que permite acceder a las fuentes de datos. Esencialmente trabaja sobre 4 operaciones básicas, llamadas CRUD (por Create-Retrieve-Update y Delete), que se realizan sobre cualquier fuente de datos. (15)

CodeIgniter usa el acercamiento Modelo Vista Controlador, permitiendo una buena separación entre la lógica y la presentación.

SOAP: (Simple Object Access Protocol, Protocolo Simple de Acceso a Objetos) es un protocolo de mensajes entre computadores. SOAP especifica el formato de mensaje que accede e invoca a los objetos, más que un objeto en particular. Permite la comunicación y la interoperabilidad entre diversas aplicaciones web desarrolladas bajo tecnologías diferentes.

1.8 Fundamentación de la solución propuesta

Como propuesta de solución al problema y teniendo en cuenta los aspectos tratados en este capítulo, se ha decidido desarrollar un sistema informático basado en la web que automatice los procesos de formación e investigación del Centro de Consultoría de la UCI. Las ventajas que proporcionan las aplicaciones web necesarias para este trabajo son:

- Se facilita el trabajo a distancia.
- No se requieren complicadas combinaciones de hardware y software para utilizar estas aplicaciones, sólo un computador con un buen navegador web.
- Son fáciles de usar.
- Proporciona alta disponibilidad.
- Permiten reunir las diferentes áreas de la empresa.
- Mayor control de datos y mejor seguridad en las diferentes secciones del sitio web.
- Permiten tener un avanzado sistema de consultas, altas, bajas y modificaciones de datos provenientes de cualquier área de la empresa, lo cual mantendrá la información siempre actualizada. (16)

1.9 Conclusiones

En este capítulo se realizó un estudio de las tendencias actuales para el desarrollo de sistemas de gestión de la formación e investigación, demostrando que no existe en estos momentos un sistema que sea capaz de gestionar estos procesos de acuerdo a las políticas del Centro de Consultoría de la UCI y la necesidad de llevar el control de la situación referente al banco de problemas, las tesis, los curriculums y los cursos. Se escogieron las tecnologías apropiadas, en este caso las que por política utiliza el centro para el desarrollo de sistemas informáticos, sentando las bases para la fase de implementación.

2.1 Introducción

Para desarrollar una aplicación es necesario comprender la estructura y dinámica de la organización en la cual se va a implantar para delimitar el problema e identificar las mejoras potenciales, asegurando que los clientes y desarrolladores tengan un entendimiento común de la misma, para lo que se realiza el modelamiento del negocio. Del modelo de negocio se derivan los requerimientos funcionales y no funcionales del sistema que darán solución al problema planteado.

En el presente capítulo se abordarán las características del sistema. Se presentarán los requisitos funcionales y no funcionales. Se hará referencia al diseño de los prototipos de interfaces de usuarios no funcionales y al diseño de la base de datos.

2.2 Requerimientos del sistema

2.2.1 Requerimientos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir.

- R-1-Autenticar usuario
- R-2-Gestionar usuario.
 - ✓ R-2.1-Insertar usuario.
 - ✓ R-2.2-Modificar usuario.
 - ✓ R-2.3-Eliminar usuario.
- R-3-Gestionar rol a usuario.
 - ✓ R-3.1-Asignar rol a usuario.
- R-4-Gestionar perfil de tesis.

- ✓ R-4.1-Insertar perfil de tesis.
- ✓ R-4.2-Modificar perfil de tesis.
- ✓ R-4.3-Eliminar perfil de tesis.
- R-5-Gestionar tribunales de tesis y cronograma.
 - ✓ R-5.1-Insertar tribunales de tesis y cronograma.
 - ✓ R-5.2-Modificar tribunales de tesis y cronograma.
 - ✓ R-5.3-Eliminar tribunales de tesis y cronograma.
- R-6-Gestionar notas de cortes de tesis y señalamientos.
 - ✓ R-6.1-Insertar notas de cortes de tesis y señalamientos.
 - ✓ R-6.2-Modificar notas de cortes de tesis y señalamientos.
- R-7-Gestionar Cursos.
 - ✓ R-7.1-Insertar cursos.
 - ✓ R-7.2-Modificar cursos.
 - ✓ R-7.3-Eliminar cursos.
- R-8-Gestionar grupo de desarrollo.
 - ✓ R-8.1-Insertar grupo de desarrollo.
 - ✓ R-8.2- Modificar grupo de desarrollo.
 - ✓ R-8.3- Eliminar grupo de desarrollo.
- R-9-Gestionar matrícula en los cursos.
 - ✓ R-9.1-Matricular en cursos.

- ✓ R-9.2-Desmatricular de cursos.
- R-10-Evaluar y eliminar estudiante en un curso.
 - ✓ R-10.1-Evaluar estudiante.
 - ✓ R-10.2-Eliminar estudiante.
- R-11-Gestionar vinculación a la docencia.
 - ✓ R-11.1-Insertar vinculación a la docencia.
 - ✓ R-11.2-Modificarvinculación a la docencia.
 - ✓ R-11.3-Eliminarvinculación a la docencia.
- R-12-Gestionar problemas del banco de problemas.
 - ✓ R-12.1-Insertar problema.
 - ✓ R-12.2-Modificar problema.
 - ✓ R-12.3-Eliminar problema.
- R-13-Gestionar currículums.
 - ✓ R-13.1-Insertar currículums.
 - ✓ R-13.2-Modificar currículums.
 - ✓ R-13.3-Eliminar currículums.
- R-14-Realizar Búsquedas.
 - ✓ R-14.1-Buscar Tesis.
 - ✓ R-14.2- Buscar cursos.
 - ✓ R-14.3-Buscar Banco de problemas.

- ✓ R-14.4- Buscar currículums.
- ✓ R-14.5- Buscar formas de vinculación a la docencia.
- ✓ R-14.6-Buscar publicaciones.
- ✓ R-14.7-Buscar persona.

2.2.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como características que hacen al producto atractivo, confiable, rápido y usable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir, una vez se conozca lo que el sistema deba hacer se podrá determinar cómo ha de comportarse, que cualidades debe tener o cuán rápido o grande debe ser. (17)

- **Interfaz:** El sistema informático debe poseer una interfaz web amigable y sencilla, fácil para la interacción del usuario, sin saturación de colores ni imágenes.
- **Usabilidad:** El sistema informático podrá ser utilizado por los usuarios de la universidad.
- **Rendimiento:** El sistema informático debe ser lo más eficiente posible para poder lograr un tiempo de respuesta adecuado. La velocidad de procesamiento de la información debe de ser rápida.
- **Soporte:** El sistema informático debe permitir posteriores modificaciones y actualizaciones a fin de alcanzar mayor funcionalidad o dado que cambien algunos elementos del negocio.
- **Legales:** Es un deber proteger la información por parte de las personas que tienen derecho de administración u otros que pongan en peligro la integridad y seguridad del sistema.
- **Software:** Tener como sistema operativo Linux, Windows 98 o superior, se necesita tener instalado un servidor de aplicaciones web Apache, y haber instalado PosgretSQL.

- **Hardware:** Se requiere un servidor de 512 MB de RAM como mínimo y 100 MB como espacio mínimo en el disco duro. Las computadoras para el proceso de pruebas deben de estar conectadas a una red y tener al menos 128 MB de RAM.
- **Políticos y culturales:** Se deben poner las palabras lo más acorde a su entendimiento.
- **Seguridad**
 - Integridad: La información manejada por el sistema será objeto de cuidadosa protección, será considerada igual a la fuente o autoridad de los datos.
 - Disponibilidad: A los usuarios autorizados se le garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán ni retrasarán a los usuarios para obtener los datos deseados en un momento dado.

2.3 Análisis del modelo de diseño propuesto

Del diseño propuesto se pudo extraer las clases fundamentales que deben ser definidas para que el sistema informático funcione satisfactoriamente, así como los atributos y métodos que deben tener las mismas, logrando de esta manera adentrarse en las especificidades de los requisitos no funcionales, las restricciones de los lenguajes de programación, las características de la programación en capas y las tecnologías de desarrollo. La obtención del diseño, resultó de gran importancia, pues permitió crear una entrada apropiada y un punto de partida para las actividades de implementación.

El diseño propuesto fue creado con los siguientes patrones, que de manera general constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, permitiendo llevar a cabo la implementación del sistema informático bajo estos como el Modelo Vista Controlador (MVC) y los Patrones Generales de Software para Asignación de Responsabilidades (GRASP).

Entre los patrones usados en la solución se encuentran:

- **MVC:** El patrón MVC realiza un diseño que desacopla la vista del modelo, con la finalidad de mejorar la reusabilidad, permitiendo esto que las modificaciones

realizadas en las vistas influyan en menor medida en la lógica de negocio o de datos (Anexo 11).

- **Creador:** El patrón Creador se encarga de identificar la clase responsable de la creación de nuevos objetos. Presente en todas las clases controladoras cuando crean diferentes objetos del modelo según los datos que necesiten.
- **Alta cohesión:** El patrón Alta Cohesión indica que la información que almacena una clase debe ser coherente, de manera que todos sus métodos tengan un comportamiento bien definido. Es utilizado para la construcción del sistema informático debido a que cada clase implementa las funcionalidades que le son correspondidas.
- **Bajo acoplamiento:** El patrón Bajo Acoplamiento, su objetivo en el sistema informático es tratar de mantener las clases lo menos ligadas entre sí, de tal forma que en caso de producirse una modificación en alguna de ellas se tenga la mínima repercusión posible en el resto, potenciando la reutilización y disminuyendo sus dependencias.
- **Experto:** El patrón Experto es el encargado de asignar una responsabilidad al experto en información. La clase controladora delega la responsabilidad a las clases modelos de realizar la acción, por ejemplo al insertar, las modelos son las encargadas de realizar esta función porque conocen toda la información.
- **Controlador:** El patrón Controlador se encarga de gestionar los eventos generados en capas anteriores. Las clases controladoras a partir de dichos eventos toman las decisiones apropiadas, invocando funcionalidades contenidas en capas más profundas como el acceso a datos. Su función es de mediador o intermediario, del controlador del negocio asociado que en este caso son las clases modelos. (18)

Los patrones antes mencionados fueron aplicados a las clases definidas, distribuyendo responsabilidades entre las mismas de forma tal que no existan muchas relaciones y que no sea sobrecargada de métodos una clase en específico, un ejemplo pudiera ser el uso del patrón Alta Cohesión y Bajo Acoplamiento, los cuales así unidos mantienen la complejidad dentro de límites manejables, permiten igualmente obtener clases que pueden ser reutilizadas y que no sean vulnerables a los cambios.

2.4 Estándar de codificación utilizado en la implementación

2.4.1 Nomenclatura de las clases

Los nombres de las clases se escriben la primera letra con mayúsculas y las demás con minúsculas. En caso de ser un nombre compuesto es seguido del primer nombre y tiene la misma estructura que el primer nombre. Ejemplo: Reporte, ReporteVinculación.

2.4.2 Nomenclatura de las clases según el tipo

- Las Controladoras que se encuentran dentro de la carpeta “controller”:

Las clases controladoras comenzarán con C de Controller y luego el nombre de la clase comenzando con mayúscula. Ejemplo: *CVinculación*.

- Las Modelos que se encuentran dentro de la carpeta “model”:

Las clases modelos comenzarán con M de Model y luego el nombre de la clase comenzando con mayúscula. Ejemplo: *MVinculación*.

2.4.3 Nomenclatura de los métodos o funciones

El nombre de los métodos de una clase comienzan con minúsculas, en caso de que sea compuesto seguido del primer nombre comienza el segundo con la primera letra en mayúscula. Deben describir el propósito del mismo. Ejemplos: insertar(), insertarVinculación().

2.4.4 Nomenclatura de las variables

El nombre a emplear para las variables se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto seguido de la primera palabra se escribe la segunda con inicial mayúscula. Ejemplos: var nombre, var nombreProfesor.

2.5 Descripción de la implementación del subsistema Vinculación a la Docencia

El subsistema Vinculación a la Docencia tiene como responsabilidad más importante gestionar la vinculación a la docencia de los profesores del Centro de Consultoría en las distintas áreas de la universidad, y algunos otros casos específicos como las facultades regionales existentes en otras provincias del país. Facilitará la forma de vincular a un profesor a un lugar de los antes mencionados, y

será más rápido saber donde se encuentra trabajando algún profesor en un período dado. Para la implementación de este subsistema se trabajó con 5 tablas, partiendo de la tabla profesor, quien es el que tiene vinculaciones docentes. Las vinculaciones a la docencia pueden ser de 4 formas:

- Impartiendo asignaturas.
- Impartiendo maestrías.
- Impartiendo cursos.
- Tutorando tesis.

Las acciones a llevar a cabo para cada una de las vinculaciones a la docencia que se vayan a desarrollar serán controladas por la clase controladora CVinculación a través de la llamada a los métodos definidos en la clase modelo MVinculación. Para gestionar la vinculación a la docencia de un profesor primeramente se cargan en una tabla todos los profesores que se encuentran registrados en la BD de la aplicación (Anexo 2). Luego se escoge el profesor al que se le va a gestionar su vinculación docente, y en dependencia del tipo de vinculación que sea, se selecciona en la barra de tareas de la interfaz Vinculación a la Docencia (Anexo 3). Por último se podrá gestionar la vinculación del profesor seleccionado (Anexo 4).

2.6 Descripción de la implementación del subsistema Grupo de Desarrollo

El subsistema Grupo de Desarrollo tiene como objetivo controlar toda la información que se maneja en los distintos grupos de desarrollo existentes en el Centro de Consultoría. Permite la gestión primeramente de los grupos, y posteriormente la asignación a un grupo de todos los elementos que pueda contener. Estos elementos podrán ser en este caso profesores, eventos, banco de problemas y publicaciones. Las acciones a llevar a cabo para cada una de las actividades que se vayan a desarrollar serán controladas por la clase controladora CGrupo a través de la llamada a los métodos definidos en la clase modelo MGrupo. Para el desarrollo de este subsistema se utilizaron 8 tablas, partiendo de la tabla grupo de desarrollo, y brindando las posibles acciones a desarrollar. Luego de tener algún grupo registrado se le pueden asignar los diferentes elementos (Anexo 5). Algunas de las posibles acciones a desarrollar es asignar profesores a un grupo (Anexo 6), asignar eventos (Anexo 7) y asignar publicaciones (Anexo 8).

2.7 Descripción de algunas clases y operaciones del sistema

2.7.1 Clases Modelo

Nombre MNombreClase	
Tipo de clase: Model	
Atributo	Tipo
Para cada responsabilidad	
Nombre.	Descripción.
MNombreClase	Constructor de la clase que instancia a las entidades.
insertar(\$parámetros)	Encargada de los datos en la base de datos.
modificar(\$parámetros)	Encargada de modificar los datos de un objeto.
eliminar(\$parámetro)	Encargada de eliminar los datos de un objeto.
cargar(\$limit,\$start)	Encargada de cargar los datos de un objeto de forma lineal.

Tabla 0.1 Generalización de las clases Modelo.

2.7.2 Clases Controladoras

Nombre: CVinculación	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad	
Nombre.	Descripción.
cargarVincCurso	Encargada de cargar los datos de las vinculaciones a cursos existentes.
adicionarVincCurso	Encargada de adicionar una nueva vinculación a un curso.
modificarVincCurso	Encargada de modificar una vinculación a un curso.
eliminarVincCurso	Encargada de eliminar una vinculación a un curso.
cargarVincTesis	Encargada de cargar los datos de las vinculaciones a tesis existentes.
adicionarVincTesis	Encargada de adicionar una nueva vinculación a tesis.

modificarVincTesis	Encargada de modificar una vinculación a una tesis.
eliminarVincTesis	Encargada de eliminar una vinculación a una tesis.
cargarVincAsignaturas	Encargada de cargar los datos de las vinculaciones a asignaturas existentes.
adicionarVincAsignaturas	Encargada de adicionar una vinculación a una asignatura.
modificarVincAsignaturas	Encargada de modificar una vinculación a una asignatura.
eliminarVincAsignaturas	Encargada de eliminar una vinculación a una asignatura.
cargarVincMaestrías	Encargada de cargar los datos de las vinculaciones a maestrías existentes.
adicionarVincMaestrías	Encargada de adicionar una vinculación a una maestría.
modificarVincMaestrías	Encargada de modificar una vinculación a una maestría.
eliminarVincMaestrías	Encargada de eliminar una vinculación a una maestría.

Tabla 0.2 Principales funcionalidades de la clase controladora del módulo Vinculación a la docencia.

Nombre: CGrupo	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad	
Nombre.	Descripción.
cargarGrupos	Encargada de cargar los datos de los grupos existentes.
adicionarGrupo	Encargada de adicionar un nuevo grupo.
modificarGrupo	Encargada de modificar un nuevo grupo.
eliminarGrupo	Encargada de eliminar un nuevo grupo.
complementarGrupo	Encargada de adicionar áreas, objetivos y roles a un grupo.

cargarProfesores	Encargada de cargar los profesores existentes.
asignarProfesores	Encargada de asignar profesores a un grupo.
mostrarAsignados	Encargada de mostrar los profesores asignados a un grupo seleccionado.
cargarEventos	Encargada de mostrar los eventos asignados a un grupo seleccionado.
adicionarEvento	Encargada de adicionar una nuevo evento a un grupo.
modificarEvento	Encargada de modificar una nuevo evento a un grupo.
eliminarEvento	Encargada de eliminar una nuevo evento a un grupo.
cargarBancoProblemas	Encargada de cargar los datos del banco de problemas de un grupo.
adicionarProblema	Encargada de adicionar un problema al banco de problemas de un grupo.
modificarProblema	Encargada de modificar un problema del banco de problemas de un grupo.
eliminarProblema	Encargada de eliminar un problema del banco de problemas de un grupo.
cargarPublicaciones	Encargada de cargar los datos de las publicaciones de un grupo.
adicionarPublicaciones	Encargada de adicionar una publicación a un grupo.
modificarPublicaciones	Encargada de modificar una publicación de un grupo
eliminarPublicaciones	Encargada de eliminar una publicación de un grupo.

Tabla 0.3 Principales funcionalidades de la clase controladora del módulo Grupo de Desarrollo.

Nombre: CBuscar	
Tipo de clase: <i>Controladora</i>	
Atributo	Tipo
Para cada responsabilidad	
Nombre.	Descripción.
cargarPersonas	Encargada de cargar los datos de las personas registradas en la aplicación.

buscarPersonas	Encargada de buscar una o más personas por criterios especificados.
cargarCursos	Encargada de cargar los datos de los cursos existentes.
buscarCursos	Encargada de buscar uno o más cursos por criterios especificados.
cargarTesis	Encargada de cargar los datos de las tesis existentes.
buscarTesis	Encargada de buscar una o más tesis por criterios especificados.
cargarBancoProblemas	Encargada de cargar los datos de los problemas existentes.
buscarBancoProblemas	Encargada de buscar uno o más problemas por criterios especificados.
cargarVincCursos	Encargada de cargar los datos de las vinculaciones a cursos existentes.
buscarVincCursos	Encargada de buscar una o más vinculaciones a cursos por criterios especificados.
cargarVincTesis	Encargada de cargar los datos de las vinculaciones a tesis existentes.
buscarVincTesis	Encargada de buscar una o más vinculaciones a tesis por criterios especificados.
cargarVincMaestrías	Encargada de cargar los datos de las vinculaciones a maestrías existentes.
buscarVincMaestrías	Encargada de buscar una o más vinculaciones a maestrías por criterios especificados
cargarVincAsignaturas	Encargada de mostrar los datos de las vinculaciones a asignaturas existentes.
buscarVincAsignaturas	Encargada de buscar una o más vinculaciones a asignaturas por criterios especificados

Tabla 0.4 Principales funcionalidades de la clase controladora del módulo Búsqueda.

2.8 Integración de los módulos del sistema

La integración de los módulos en el sistema viene dado por los privilegios con los que cuentan los distintos usuarios que se registren. En dependencia del privilegio se tendrá acceso o no a los diferentes módulos, estableciendo permisos específicos para estudiantes, profesores, el responsable de formación e

investigación en el Centro de Consultoría y un administrador general que tendrá total dominio sobre la aplicación. La seguridad del sistema estará dada por los niveles de protección que se le brinde a la información en dependencia del usuario que se autentique, por ejemplo, un estudiante no puede tener acceso a los módulos con los mismos privilegios que un profesor, y así con los demás roles que se hayan definido. Un estudiante mayoritariamente tiene acceso a los reportes provenientes de módulos como Tesis y Búsquedas, y a acciones como ver los cursos existentes, solicitar un curso y luego poder ver su evaluación en dicho curso. Por su parte, el profesor tendría permisos para acceder a los mencionados anteriormente incluyendo también los módulos Curriculum, Grupo de desarrollo y Vinculación a la docencia. El responsable de formación e investigación se encargaría de gestionar la información de todos los módulos, sin el único privilegio de cambiar los roles a los usuarios, de lo que se encargaría el administrador general. La interacción entre los módulos está en dependencia de las acciones que se vayan a desarrollar, partiendo de la autenticación, requisito principal para poder interactuar con el sistema informático, de ahí en adelante no existe un orden lógico para desarrollar alguna actividad en específico, se podrá acceder a las funcionalidades hacer lo que sea permitido de acuerdo al privilegio del usuario que se registre.

2.9 Conclusiones

Durante el desarrollo de este capítulo se explica en detalle la estructura interna del sistema informático que se propone. Fueron relacionados los requerimientos funcionales y no funcionales, los estándares de codificación utilizados en la fase de implementación y se realizó la descripción de algunos de los módulos de la aplicación. Se desarrolló la solución partiendo de la arquitectura definida, quedando listo el escenario para el flujo de trabajo de prueba. Se generaron artefactos como el Modelo Entidad – Relación y el diagrama de despliegue (Anexo 12).

VÁLIDACIÓN DE LA SOLUCIÓN PROPUESTA**3.1 Introducción**

En el desarrollo del software, las posibilidades de errores son innumerables. Pueden darse por una mala especificación de los requisitos funcionales, uso indebido de las estructuras de datos, errores al enlazar módulos, entre otras. Para resolver este problema, se incluyó a nivel metódico un nuevo proceso en la confección de los sistemas informáticos: el proceso de prueba. Este proceso de prueba puede probar la presencia de errores pero no la ausencia de ellos.

3.2 Pruebas de software

En la cadena de valor del desarrollo de un software específico, el proceso de prueba es clave a la hora de detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. Las pruebas de software, son los procesos que permiten verificar y revelar la calidad de un producto de software, son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa. (19)

Un buen caso de prueba es aquel que tiene alta probabilidad de mostrar un error no descubierto hasta entonces. Una prueba tiene éxito si descubre un error no detectado hasta entonces. La prueba no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software. (20)

La fase de pruebas añade al producto final el valor para afirmar que ya se ha alcanzado la calidad requerida. Gran porcentaje de los programas que se desarrollan tienen errores, y es en la fase de pruebas donde se descubren, esa es la importancia fundamental de esta etapa, el objetivo específico de la fase de pruebas es encontrar todos los posibles errores, mientras más mejor. (20)

3.3 Niveles de Prueba

Pruebas unitarias: es una forma de probar el correcto funcionamiento de un módulo de código. Asegura que cada uno de los módulos funcione correctamente por separado. La idea es escribir casos de prueba para cada función no trivial o método en el módulo de forma que cada caso sea independiente del resto.

Estas pruebas aisladas proporcionan cinco ventajas básicas: fomentan el cambio, simplifica la integración, documenta el código, separa la interfaz del código y hace que los errores estén más acotados y sean fáciles de localizar. Es válido destacar que las pruebas unitarias no descubrirán todos los errores del código. Por definición, sólo prueban las unidades por sí solas. Por lo tanto, no descubrirán errores de integración, problemas de rendimiento y otros problemas que afectan a todo el sistema en su conjunto. (21)

Pruebas de integración: A partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces. Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez. Consiste en realizar pruebas para verificar que un gran conjunto de partes del software funcionan juntos. (22)

Pruebas de sistemas: El software ensamblado totalmente con cualquier componente (hardware) que requiera, se prueba para comprobar que se cumplen los requisitos funcionales. Cualquier pieza de software completo, desarrollado o adquirido, puede verse como un sistema que debe probarse, ya sea para decidir acerca de su aceptación, para analizar defectos globales o para estudiar aspectos específicos de su comportamiento, tales como seguridad o rendimiento. A este tipo de pruebas donde se estudia el producto completo se les llama Pruebas de Sistema. (22)

Pruebas de aceptación: Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable al cliente; sino que se realizan sobre el producto terminado e integrado. (22)

Pruebas funcionales: Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Las pruebas funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático. (23)

3.4 Métodos de Prueba

3.4.1 Pruebas de Caja Negra o Funcionales

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene, sin tener en cuenta su funcionamiento interno. Se encargan de saber qué es lo que hace el software, pero sin dar importancia a cómo lo hace.

Algunas técnicas utilizadas en la prueba de caja negra son:

- **Partición de Equivalencia:** Técnica que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. (23)
- **Análisis de Valores Límites:** La experiencia muestra que los casos de prueba que exploran las condiciones límites producen mejor resultado que aquellos que no lo hacen. Las condiciones límites son aquellas que se hallan en los márgenes de la clase de equivalencia, tanto de entrada como de salida. Por ello, se ha desarrollado el análisis de valores límites como técnica de prueba. (23)

Las pautas para desarrollar casos de prueba con esta técnica son:

1. Si una condición de entrada especifica un rango de valores, se diseñarán casos de prueba para los dos límites del rango, y otros dos casos para situaciones justo por debajo y por encima de los extremos.
2. Si una condición de entrada especifica un número de valores, se diseñan dos casos de prueba para los valores mínimo y máximo, además de otros dos casos de prueba para valores justo por encima del máximo y justo por debajo del mínimo.
3. Aplicar las reglas anteriores a los datos de salida.
4. Si la entrada o salida de un programa es un conjunto ordenado, habrá que prestar atención a los elementos primero y último del conjunto.

- **Grafos de Causa-Efecto:** Es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones. (21)

Para realizar este tipo de prueba se usará el requisito Adicionar Publicación. El objetivo de este requisito es asociarle una publicación a un grupo de desarrollo, procedimiento que estará regido por precondiciones y poscondiciones.

Nombre del Requisito	Descripción General	Escenario de Pruebas	Flujo del Escenario
Adicionar Publicación.	El sistema permite adicionar una nueva publicación teniendo en cuenta que se haya seleccionado el grupo al que se le adicionará dicha publicación.	EP 1.1 Adicionar una publicación introduciendo los datos correctamente al presionar el botón Aceptar.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen los datos correctamente. - Se presiona el botón Aceptar. - Se muestra la notificación “La publicación ha sido adicionada satisfactoriamente.” - Se presiona el botón Aceptar.
		EP 1.2 Adicionar una publicación introduciendo datos inválidos al presionar el botón Aceptar.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen los datos inválidos. - Se presiona el botón Aceptar. - Se muestra la notificación “Existen campo(s) con valor(es) erróneos. Corríjalos.” - Se presiona el botón Aceptar.
		EP 1.3 Adicionar una publicación dejando información requerida en blanco al presionar el botón Aceptar.	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen los datos y se deja información requerida en blanco. - Se presiona el botón Aceptar. - Se muestra la notificación “Existen campo(s) con valor(es) erróneos.”

			<p>Corríjalos.”</p> <ul style="list-style-type: none"> - Se presiona el botón Aceptar.
		<p>EP 1.4 Al presionar el botón Cancelar.</p>	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen los datos o no en los campos. - Se presiona el botón Cancelar.
		<p>EP 1.1 Adicionar una publicación introduciendo los datos correctamente al presionar el botón Aplicar.</p>	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen los datos correctamente. - Se presiona el botón Aplicar. - Se muestra la notificación “La publicación ha sido adicionada satisfactoriamente.” - Se presiona el botón Aceptar.
		<p>EP 1.2 Adicionar una publicación introduciendo datos inválidos al presionar el botón Aplicar.</p>	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen los datos inválidos. - Se presiona el botón Aplicar. - Se muestra la notificación “Existen campo(s) con valor(es) erróneos. Corríjalos.” - Se presiona el botón Aceptar.
		<p>EP 1.3 Adicionar una publicación dejando información requerida en blanco al presionar el botón Aplicar.</p>	<ul style="list-style-type: none"> - Se presiona el botón Adicionar. - Se introducen los datos y se deja información requerida en blanco. - Se presiona el botón Aplicar. - Se muestra la notificación “Existen campo(s) con valor(es) erróneos. Corríjalos.” - Se presiona el botón Aceptar.

Tabla 3.1. Escenario de flujos para el requisito Adicionar Publicación.

3.4.2 Pruebas de Caja Blanca o Estructurales

Se denomina cajas blancas a un tipo de prueba de software que se realiza sobre las funciones internas de un módulo. Así como las pruebas de caja negra ejercitan los requisitos funcionales desde el exterior del módulo, las de caja blanca están dirigidas a las funciones internas. Las pruebas de caja blanca se llevan a cabo en primer lugar, sobre un módulo concreto, para luego realizar las de caja negra sobre varios subsistemas. (19)

Algunas técnicas de prueba de Caja Blanca son:

- **Prueba de Condición:** Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.
- **Prueba de Flujo de Datos:** Se selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.
- **Prueba de Bucles:** Es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles.
- **Prueba del Camino Básico:** Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar la misma como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática.

Los pasos que se siguen para aplicar esta técnica son:

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo.
3. Se determina un conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del camino básico.

Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. (24)

Para realizar la prueba es necesario realizar primeramente el análisis de complejidad del algoritmo sobre el que se va a realizar la prueba, con el propósito de calcular los valores de la complejidad ciclomática.

```
function insertarProblema(banco) {
  if (formbanco.getForm().isValid()) { //1
    var t = Ext.MessageBox.show( //2
      msg : 'Adicionando problema...', //2
      progressText : 'Importar...', //2
      width : 300, //2
      wait : true //2
    ); //2
    Ext.Ajax.request( //3
      url : 'index.php/CGrupo/adicionandoProblema', //3
      params : { //3
        problema : tfproblema.getValue(), //3
        idformaresolucion : cbformaresolucion.getValue(), //3
        idestado : cbestadobanco.getValue(), //3
        idgrupo : smgrupo.getSelected().data.idgrupo, //3
        numareaconoinvolucra : cbareainvolucra.getValue(), //3
        numobjafecta : cbobjafecta.getValue() //3
      }, //3
      callback : function(options, success, response) { //4

        msg('Informaci&oacute;n',
          'Problema adicionado satisfactoriamente.');//4
        storebanco.reload( //4
          params : { //4
            idgrupo : smgrupo.getSelected().data.idgrupo, //4
            start : 0, //4
            limit : 17 //4
          } //4
        ); //4
      } //4
    ); //4
  } //4
  else { //5
    msg('Informaci&oacute;n',
      'Existen campos con valores err&oacute;neos. Corr&iacute;jalos.');//5
  } //6
} //7
```

Fig. 3.1 Representación del algoritmo insertarProblema(banco).

Después de este paso, es necesario representar el grafo de flujo asociado, en el cual se representan distintos componentes como:

Nodos: Son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.

Aristas: Son constituídas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aún cuando el nodo no representa la sentencia de un procedimiento.

Nodos predicados: Son los nodos desde donde salen 2 o más aristas.

Regiones: Son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento. (23)

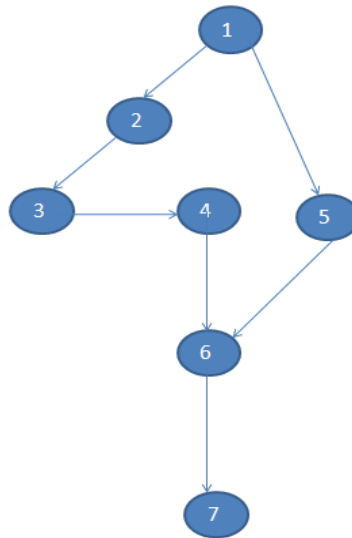


Fig. 3.2 Grafo de flujo asociado al algoritmo insertarProblema(banco).

Seguidamente a la construcción del grafo de flujo se procede a efectuar el cálculo de la complejidad ciclomática del código, el cálculo es necesario efectuarlo mediante tres vías o fórmulas. Para concluir que el cálculo fue correcto es necesario que por las tres vías el resultado sea el mismo, las fórmulas para calcular son las siguientes:

1. $V(G) = (A - N) + 2$

Donde “**A**” es la cantidad de Aristas y “**N**” la cantidad de Nodos.

$$V(G) = (7 - 7) + 2$$

$$V(G) = 2$$

2. $V(G) = P + 1$

Siendo “**P**” la cantidad de Nodos predicados

$$V(G) = 1 + 1$$

$$V(G) = 2$$

3. $V(G) = R$

Donde “**R**” representa la cantidad de regiones en el grafo.

$$V(G) = 2$$

Como el cálculo por las 3 vías arroja el mismo resultado se puede concluir que el algoritmo anteriormente representado tiene una complejidad ciclomática igual a 2. Esto significa que existen a lo sumo 2 caminos por el cual recorrer el grafo. Éste valor representa además el número máximo de pruebas que se le pueden realizar el algoritmo.

Con los datos anteriores se representan los posibles caminos por los que se puede recorrer el grafo:

Camino Básico No. 1: 1 - 2 - 3 - 4 - 6 - 7.

Camino Básico No. 2: 1 - 5 - 6 - 7.

Luego de haber determinado los caminos básicos se procede a ejecutar los casos de prueba para el procedimiento, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

Para realizar los casos de pruebas es necesario cumplir con las siguientes exigencias:

Descripción: Se hace la entrada de datos necesaria, validando que ningún parámetro obligatorio pase nulo al procedimiento o no se entre algún dato erróneo.

Condición de ejecución: Se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.

Entrada: Se muestran los parámetros que entran al procedimiento.

Resultados Esperados: Se expone el resultado que se espera que devuelva el procedimiento.

Caso de prueba para el camino básico # 1:

Descripción: Los datos de entrada cumplirán con los siguientes requisitos: El problema es una cadena de caracteres, los idformaresolución e idestadobanco, el numareaconoinvolucra y numobjafecta se escogen en un combobox, y el idgrupo se selecciona al escoger el grupo al que se le va a adicionar el problema.

Condición de ejecución: El problema es “Asignar tutores a tesis”, id del grupo es igual a 1, id de la forma de resolución es igual a 2, id del estado del banco es igual a 1, el número del área de conocimiento involucrada es igual a 2 al igual que el número del objetivo que afecta.

Entrada: problema: “Asignar tutores a tesis”, idgrupo= 1, idformaresolución= 2, idestadobanco= 1, numareaconoinvolucra= 2, numobjafecta= 2.

Resultados esperados: Se espera insertar un problema al banco de problemas de un grupo de desarrollo.

Se pudo insertar el problema debido a que todos los datos eran correctos.

Caso de prueba para el camino básico # 2:

Descripción: Los datos de entrada cumplirán con los siguientes requisitos: El problema es una cadena de caracteres, los idformaresolución e idestadobanco, el numareaconoinvolucra y numobjafecta se escogen en un combobox, y el idgrupo se selecciona seleccionando el grupo al que se le va a adicionar el problema.

Condición de ejecución: Se deja el campo del problema en blanco, id del grupo es igual a 2, id de la forma de resolución es igual a 1, id del estado del banco es igual a 2, el número del área de conocimiento involucrada es igual a 1 al igual que el número del objetivo que afecta.

Entrada: problema: "", idgrupo= 2, idformaresolución= 1, idestadobanco= 2, numareaconoinvolucra= 1, numobjafecta= 1.

Resultados esperados: Se espera insertar un problema al banco de problemas de un grupo de desarrollo.

No se pudo insertar el problema debido a que existían datos incorrectos, el campo problema que es obligatorio se encontraba en blanco.

3.5 Validación del modelo de diseño propuesto

Son varios los puntos de vista relacionados con la calidad del software. Desde metodologías hasta las distintas normas de calidad, que pueden estar orientados tanto a los procesos de desarrollo como a los productos de software. No es objetivo de este epígrafe abordar sobre los temas de calidad, pero sí desarrollar una evaluación del diseño propuesto del Sistema Informático para la Automatización de los Procesos de Formación e Investigación del Centro de Consultoría de la UCI.

Las métricas de diseño a nivel de componentes se concentran en las características internas de los componentes del software e incluyen entre otras medidas la cohesión, acoplamiento y complejidad, medidas que pueden ayudar al desarrollador de software a juzgar la calidad de un diseño a nivel de componente. (5)

Atributos de calidad que se abarcan:

- **Responsabilidad (Cohesión):** Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- **Complejidad del mantenimiento:** Consiste en el grado de esfuerzo necesario para realizar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.

- **Complejidad de implementación:** Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
- **Reutilización:** Consiste en el grado de reutilización presente en una clase o estructura de clase dentro de un diseño de software.
- **Acoplamiento:** Consiste en el grado de dependencia o interconexión de una clase o estructura de clase con otras, está muy ligada a la característica de Reutilización.
- **Cantidad de pruebas:** Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (unidad) del producto (componente, módulo, clase, conjunto de clases, etc.) diseñado.

Las métricas empleadas para evaluar la calidad del diseño de los componentes y su relación son las siguientes:

- **Tamaño operacional de clase (TOC):** Está dado por el número de métodos asignados a una clase. Los atributos que afecta son: la Responsabilidad, Complejidad de implementación y la Reutilización. De manera que mientras mayor sea el tamaño operacional de clase mayor será la Responsabilidad y Complejidad de implementación, mientras que su Reutilización disminuye. (22)

Ver instrumentos y tabla de resultados en (Anexo 9).

Luego de realizado un análisis profundo en cada una de las clases utilizadas, la evaluación de la métrica reflejó lo siguiente:

Atributos	T.O.C. (%)
Procedimientos	64(tiene entre 10 - 45 Procedimientos)
Responsabilidad	64 (Baja)
Complejidad de Implementación	64 (Baja)
Reutilización	64 (Alta)

Tabla 3.2 Resultados del instrumento de evaluación de la métrica Tamaño Operacional de Clases.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica TOC, se puede concluir que el diseño del Sistema Informático para la Automatización de los Procesos de Formación e Investigación del Centro de Consultoría de la UCI tienen una calidad aceptable teniendo en cuenta que el 64% de las clases incluidas en este componente posee menos cantidad de operaciones que la mitad del valor máximo registrado en las mediciones. Además el 64% de las clases poseen evaluaciones positivas en los atributos de calidad (Responsabilidad, Complejidad de implementación y Reutilización).

- **Relaciones entre clases (RC):** Esta dado por el número de relaciones de uso de una clase. Los atributos que afecta son el Acoplamiento, la Complejidad de mantenimiento, la Reutilización y la Cantidad de pruebas. De manera que mientras mayor sean las relaciones entre las clases mayor será el Acoplamiento, la Complejidad de mantenimiento y la Cantidad de pruebas, mientras que su Reutilización disminuye. (25)

Ver instrumentos y tabla de resultados en (Anexo 10).

Luego de realizado un análisis profundo en cada una de las clases utilizadas, la evaluación de la métrica reflejó lo siguiente:

Atributos	RC (%)
Cantidad de Relaciones	92 (posee 1 Dependencia)
Acoplamiento	92 (Baja)
Complejidad de Mantenimiento	92 (Baja)
Cantidad de Pruebas	92 (Baja)
Reutilización	92 (Alta)

Tabla 3.3 Resultados del instrumento de evaluación de la métrica Relaciones entre Clases.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica RC, se puede concluir que el diseño del Sistema Informático para la Automatización de los Procesos de Formación e Investigación del Centro de Consultoría de la UCI tiene una calidad aceptable teniendo en cuenta que el 92% de las clases incluidas en este poseen sólo una dependencia de otras clases. Además el 92% de las clases poseen bajo acoplamiento con otras, así como los atributos de

calidad Complejidad de mantenimiento, Cantidad de pruebas y Reutilización se comportan satisfactoriamente en un 92% de las clases.

Existen otras métricas como Profundidad de Herencia (PH), Número de Descendientes (ND), Número de operaciones redefinidas para una clase hija (NOR) que se emplean también para medir la calidad del diseño pero, que son enfocadas a las herencia entre clases y en el modelo de diseño propuesto no existe esta relación, por lo que se decidió no reflejarlas en la investigación y hacer un análisis más profundo en las demás métricas, exceptuando la métrica de Volatilidad, encargada de medir el grado de adaptabilidad del diseño propuesto ante los cambios de requisitos o de partes de diseño.

3.6 Resultados generales de la implantación del sistema

La implantación del sistema permitirá que se realicen de manera más rápida y segura todos los procesos de automatización para la formación e investigación del Centro de Consultoría de la UCI. Con este nuevo sistema informático se resolverán los problemas existentes relacionados con la pérdida de información, lentitud en los reportes y la desinformación del personal, y brindará a los usuarios disponibilidad, confidencialidad y seguridad de la información.

3.7 Conclusiones

En este capítulo se realizaron las pruebas básicas al software atendiendo a los distintos tipos de pruebas existentes, valorando también las métricas de diseño a nivel de componentes, permitiendo analizar características internas de los componentes del software que son de gran importancia como la cohesión, acoplamiento y complejidad de implementación. Se demostró la factibilidad del producto con los datos arrojados por estas pruebas, y que el sistema se encuentra en óptimas condiciones para el proceso de implantación en el Centro de Consultoría de la UCI.

Conclusiones generales

Con la finalización del trabajo, se ha confirmado la importancia de la implementación del Sistema Informático para la Automatización de los Procesos de Formación e Investigación del Centro de Consultoría de la UCI, pues permitirá gestionar de forma eficiente y segura el flujo de toda la información que se maneja hoy manualmente en esta área de la universidad.

Se llevó a cabo un estudio detallado del problema existente, estado del arte, y las necesidades reales que existían en el Centro de Consultoría. Se sentaron las bases para la implementación del sistema, demostrando la importancia de entender qué es realmente lo que desea el cliente, y hasta qué punto se puede cumplir. Se utilizaron para el correcto funcionamiento de esta aplicación las metodologías, herramientas y lenguajes acordes a las necesidades del país de migrar al software libre y a las políticas de la UCI.

Además se realizó una validación del diseño mediante la utilización de instrumentos de medición que se inspiraron en métricas para la calidad del diseño. Los resultados arrojados permitieron concluir que el diseño presentaba valores positivos en indicadores de calidad tales como Reutilización, Facilidad de Mantenimiento, Complejidad del Diseño, Complejidad de Implementación, Acoplamiento y Cantidad de pruebas.

Se cumplió el objetivo de la investigación, pues se realizó la implementación y las pruebas al Sistema Informático para la Automatización de los Procesos de Formación e Investigación del Centro de Consultoría de la UCI, cumpliendo con todos los requisitos establecidos.

Recomendaciones

- Cumplir con las políticas de seguridad orientadas a las TIC en la UCI.
- Continuar agregando funcionalidades al Sistema Informático para la Automatización de los Procesos de Formación e Investigación del Centro de Consultoría de la UCI con el objetivo de solucionar problemas existentes más allá de los procesos de formación e investigación.

Referencias Bibliográficas

1. **Group, BSI.** BSI. [Online] BSI Group, 1901. [Cited: Febrero 5, 2010.] <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion>.
2. **FFIS.** Fundación para la Formación e Investigación Sanitarias . [Online] Comunidad Autónoma de la Región de Murcia, Diciembre 10, 2004. [Cited: Febrero 5, 2010.] <http://www.ffis.es>.
3. **Plataforma de cooperación entre empresas.** TICBioMed. [Online] Abril 2002. [Cited: Febrero 5, 2010.] <http://www.ticbiomed.net>.
4. **Sánchez, M. A. Mendoza.** Informatizate.net. [Online] 2004. [Cited : 4 de Marzo de 2010.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
5. **Alvarez, M. A.** DesarrolloWeb.com. [Online] 2001. [Cited : 4 de Marzo de 2010.] <http://www.desarrolloweb.com/articulos/392.php>.
6. **Garbage Collector.** Sistema de Gestión de Basa de Datos. [Online] 2004. [Cited: 6 de MARzo de 2010.] http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
7. **Borrego, D.** WampServer5. [Online] 2009. [Cited: 11 de Febrero de 2010.] <http://www.herramientasparapymes.com/wampserver-5>.
8. **Fher98.** Guatewireless.org. [Online] 2009. [Cited: Marzo 12, 2010.] <http://www.guatewireless.org/articulos/mysql-vs-postgresql/>.
9. **Jordisan.** Jordisan.net. [Online] 2006. [Cited: Marzo 12, 2010.] <http://jordisan.net/blog/2006/que-es-un-framework>.
10. **Google.** Zonlinux.com.ar. [Online] 2009. [Cited: Marzo 15, 2010.] <http://zonlinux.com.ar/codeigniter-un-poderoso-framework-open-source/>.
11. **Alvarez, M. A.** DesarrolloWeb.com. [Online] 2002. [Cited: Marzo 15, 2010.] <http://www.desarrolloweb.com/articulos/que-es-html.html>.
12. **Monteiro, Lázaro.** DesarrolloWeb.com. [Online] 2009. [Cited: Febrero 13, 2010.] <http://www.desarrolloweb.com/articulos/26.php>.
13. **MaestrosdelWeb.** Maestrosdelweb.com. [Online] 2007. [Cited: 16 de Marzo de 2010.] <http://www.maestrosdelweb.com/editorial/que-es-javascript>.

14. **Corzo, G.** DesarrolloWeb.com. [Online] 2008. [Cited: 26 de Marzo de 2010.]
<http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo>.
15. **FeCr_88.** Kernelerror.net. [Online] 2009. [Cited: 16 de Febrero de 2010.]
<http://kernelerror.net/programacion/php/arquitectura-3-capas>.
16. **Emilio Barrios, Johanna Rojas.** Udistrital.edu.co. [Online] 2007. [Cited: Abril 6, 2010.]
<http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas>.
17. **Ambler, Scott W.** Ambyssoft.com. [Online] 2009. [Cited: Abril 6, 2010.]
<http://www.ambyssoft.com/essays/flootSpanish.html>.
18. **Gracia, Joaquín.** Ingenierossoftware.com. [Online] 2003. [Cited: Abril 10, 2010.]
<http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>.
19. **Christopher.** Web2development.blogspot.com. [Online] Mayo 29, 2007. [Cited: Abril 10, 2010.]
<http://web2development.blogspot.com/2007/05/patron-mvc.html>.
20. **Saavedra, J.** El mundo informático. [Online] Mayo 8, 2007. [Cited: Abril 12, 2010.]
<http://jorgesaavedra.wordpress.com/category/patrones-grasp/>.
21. **Pello, Javier.** Softqanetwork.com. [Online] 13 de Julio de 2009. [Cited: 18 de Abril de 2010.]
<http://www.softqanetwork.com/2009/07/requisitos-no-funcionales-nfr/>.
22. **Esencia Humana.** Esenciahumana.com.mx. [Online] 2008. [Cited: Abril 20, 2010.]
<http://www.esenciahumana.com.mx/Servicios/AplicacionesWeb/VentajasBeneficiosAplicaciones.html>.
23. **Vegas, Jesús.** ServidoresWeb. [Online] Marzo 21, 2002. [Cited: Abril 25, 2010.]
<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>.
24. **Cuenca, Carlos Luis.** Desarrolloweb.com. [Online] 20 de Marzo de 2003. [Cited: 26 de Abril de 2010.]
<http://www.desarrolloweb.com/articulos/1112.php>.
25. **territoriopc.** territoriopc. [Online] 2001.
http://www.territoriopc.com/javascript/tutorial_javascript_introduccion.php..
26. **HtmlCastellano.** HtmlCastellano. [Online] 2009. <http://www.programacion.com/html>.
27. **AbartiaTeam.** [Online] 2006. http://www.abartiateam.com/desarrollo-web/200602_uso-de-la-tecnologia-ajax-en-el-desarrollo-web.
28. **Vega, Yanet.** L. S. *Definición del ciclo de vida del proyecto*. Habana : s.n., 2009.

29. **Metodologiasdesistemas.** [Online] 2007. [Cited: 3 1, 2010.] <http://metodologiasdesistemas.blogspot.com/2007/10/que-es-un-orm-object-relational-mapping.html>.
30. SOA. [Online] 2006. [Cited: 29 2, 2010.] <http://arquitecturaorientadaaservicios.blogspot.com/2006/03/pero-que-es-realmente-soa.html>.
31. **Systems Popkin, Software.** Modelado de Sistemas com UML. [Online] 2008. <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.
32. **MP3es.** Spket IDE. [Online] 2009. http://wiki.mp3.es/Es/Spket_IDE.
33. **Somerville, Ian.** *Ingeniería de Software*. s.l. : Addison-Wesley, 2002. 9270-26-0206-8.
34. **Pressman, Roger S.** *Ingeniería de Software, un enfoque práctico*. s.l. : McGraw-Hill Companies, 2002. ISBN: 8448132149.
35. **Javier, Eguíluz Pérez.** librosweb.es. *Introducción a CSS*. [Online] <http://www.librosweb.es/css/>.
36. —. *Introducción a XHTML*. [Online] <http://www.librosweb.es/xhtml>.
37. **HTML.net.** Tutoriales sobre HTML y CSS - Construye tu propio sitio web. [Online] <http://es.html.net/tutorials/css/lesson1.asp>.
38. **Coello, Costa Helkyn R. informatizate.** [Online] Noviembre 2002. http://www.informatizate.net/articulos/dime_como_programas_y_te_dire_quien_eres_23082004.html.
39. **Larman, Craig.** *UML y patrones, Introducción al análisis y diseño orientado a objeto*. 1999.
40. Etapa de pruebas. [Online] Septiembre 16, 2007. <http://fabian-quintosemestre.blogspot.com/2007/09/etapa-de-pruebas.html>.
41. **Escuela Superior Politécnica del Litoral - Guayaquil - Ecuador Campus Gustavo Galindo.** SISTEMA DE ADMINISTRACION ACADEMICA POSTGRADO - SAACP. [Online] 2008. [Cited: 30 de Abril de 2010.] <http://www.csi.espol.edu.ec>.
42. **Miranda, Willy R. Rodríguez.** Superación postgraduada en percepción remota y SIG en el departamento de geociencias de la CUJAE. [Online] enero 2009. [Cited: Abril 25, 2010.] <http://www.mappinginteractivo.com>.
43. **Prestedge, Jessica.** Workflow Patterns. [Online] <http://www.workflowpatterns.com/>.
44. **Almada, Federico.** [Online] 1 22, 2008. <http://www.techtear.com/2008/01/22/zend-studio-for-eclipse-desarrollo-profesional-en-php>.

45. **Celis, Ismael.** ESTADOBETA desarrollo web con estándares. Active Record. [Online]
<http://www.estadobeta.com/2006/05/02/active-record/>.
46. Burbujas en .NET. IoC o el poder de ceder el control. [Online]
<http://geeks.ms/blogs/etomas/archive/2008/10/28/ioc-o-el-poder-de-ceder-el-control.aspx>.
47. **Collera, T. A.** Sistema de Gestión Académica. [Online] 2005. [Cited: Febrero 5, 2010.]
<http://desarrolloweb.com/artic.html>.

Anexos

Anexo 1: Gráficos Encuesta.

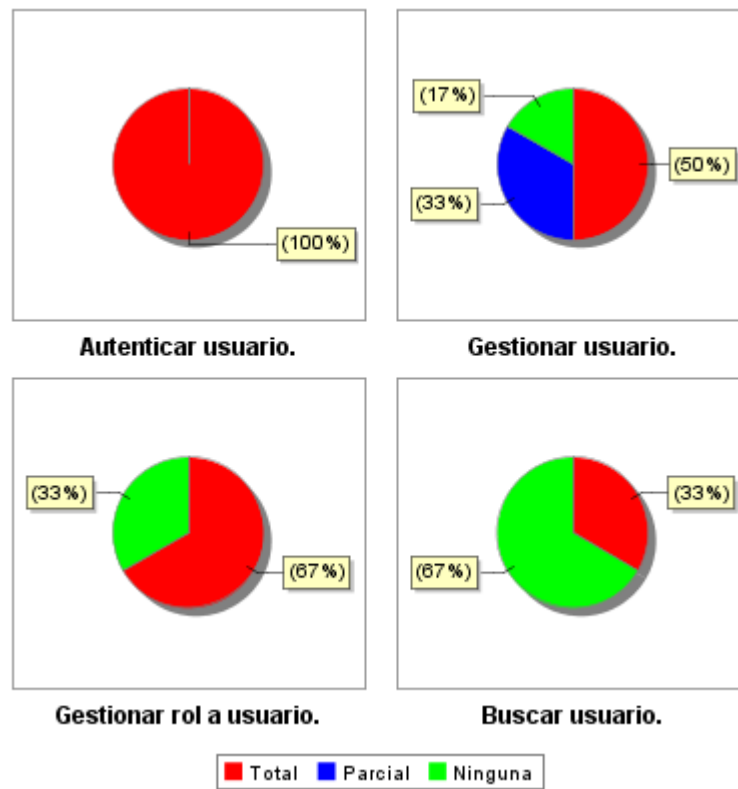


Fig.2. Solución a los requerimientos relacionados con los usuarios.

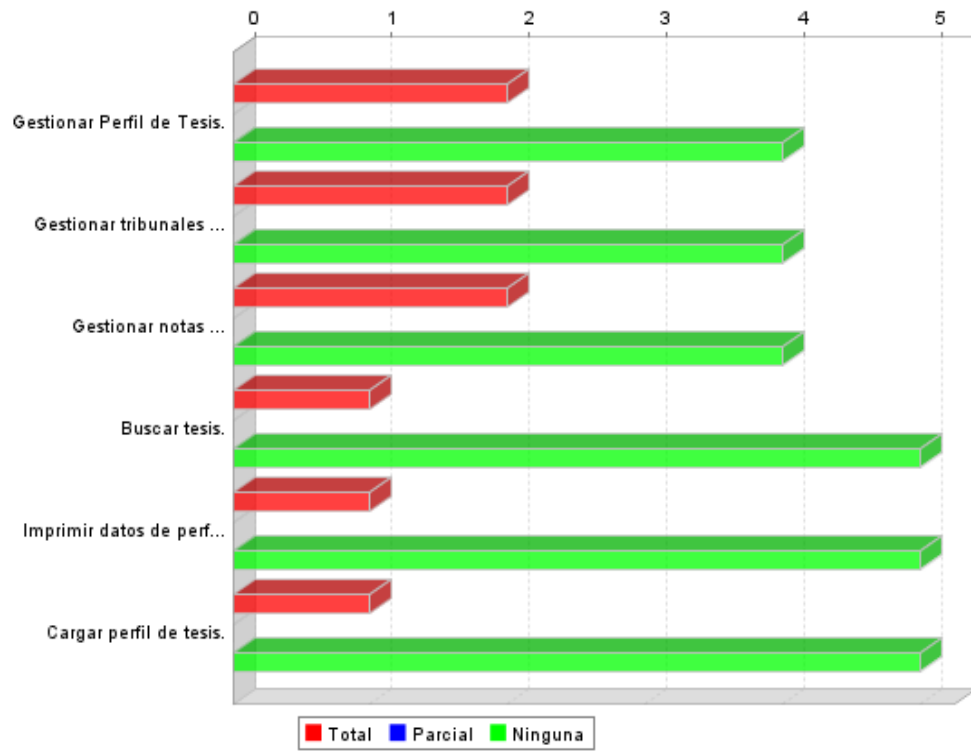


Fig.3. Solución a los requerimientos relacionados con las tesis.

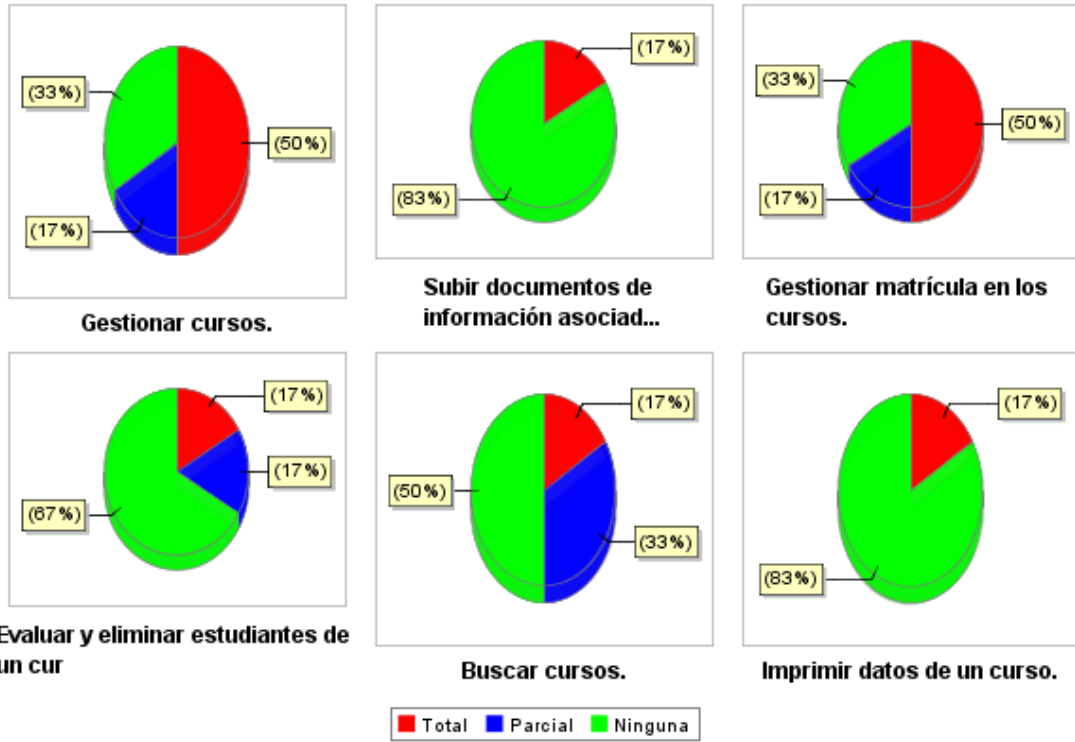


Fig.4. Solución a los requerimientos relacionados con los cursos.

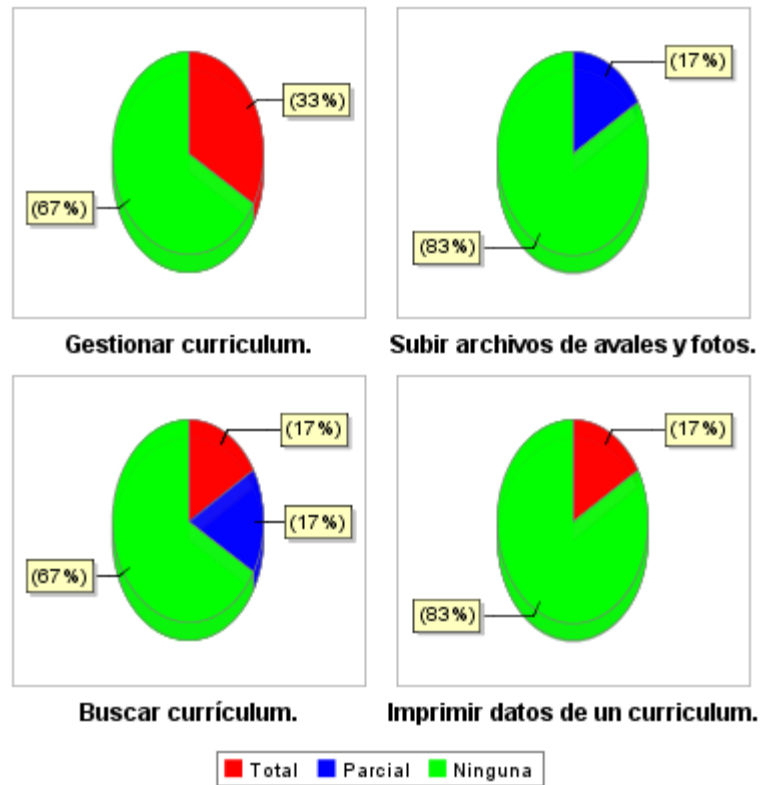


Fig.5. Solución a los requerimientos relacionados con los curriculums.

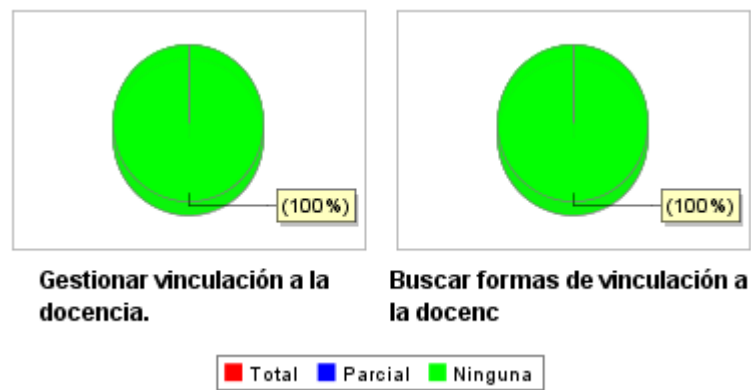


Fig.6. Solución a los requerimientos relacionados con la vinculación a la docencia.

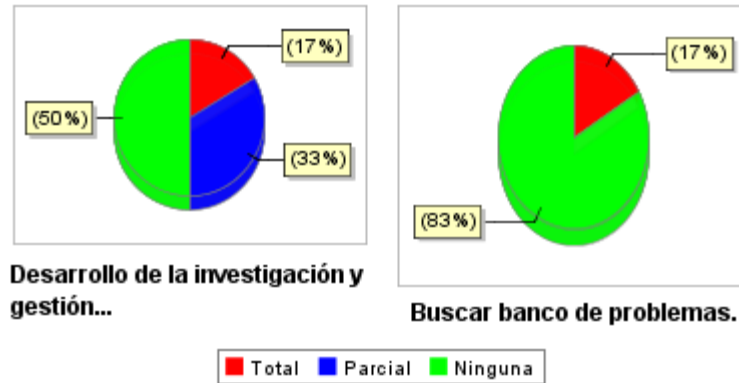


Fig.7. Solución a los requerimientos relacionados con banco de problemas.

Anexo 2: Gestionar Vinculación a la Docencia.

Vinculación a la Docencia			
<input type="radio"/> Cursos <input type="radio"/> Tesis <input checked="" type="radio"/> Asignaturas <input type="radio"/> Maestrías			
Nombre y Apellidos	ID	Teléfono	Dirección
Denís Boizan Romero	dboizan	078372868	Edif 110 Apto 102
Carlos Sanregre Castellanos	csanregre	78945624	Edif 110 Apto 101

Page 1 of 1

Anexo 3: Seleccionar profesor al que gestionar la vinculación.

Vinculación a la Docencia

Curso Tesis Asignaturas Maestrías

Nombre y Apellidos	ID	Teléfono	Dirección
Denis Boizan Romero	dboizan	078372868	Edif 110 Apto 102
Carlos Sanregre Castellanos	csanregre	78945624	Edif 110 Apto 101

Page 1 of 1

Anexo 4: Gestionar vinculación de un profesor.

Vinculación con Asignaturas

Adicionar Modificar Eliminar

Nombre	Dpto	Lugar	Semestre	Curso
Asignatura1	Dpto Asignatura1	Fac 2	1er Semestre	2010/2011
Asignatura 2	Dpto Asignatura 2	Fac 8	2do Semestre	2010/2011

Page 1 of 1

Displaying 1 - 2 of 2

Anexo 5: Gestionar grupo.

The screenshot displays a web application interface titled "Gestión de Grupos". At the top, there is a navigation bar with several icons and labels: "Adicionar" (with a plus icon), "Modificar" (with a pencil icon), "Eliminar" (with a red minus icon), "Complementar Grupo" (with a plus and minus icon), "Profesores" (with a person icon), "Eventos" (with a globe icon), "Banco de Problemas" (with a book icon), and "Publicaciones" (with a document icon). Below this bar is a table with a header row labeled "Nombre" and two data rows: "Grupo1" and "Grupo2". The table is currently empty of data. At the bottom of the interface, there is a pagination control showing "Page 1 of 1" and navigation arrows.

Anexo 6: Asignar profesores a un grupo.

Asignación de Profesores ✕

Asignar al Grupo

Nombre y Apellidos	ID	Teléfono
Denis Boizan Romero	dboizan	078372868
Carlos Sanregre Castellanos	csanregre	78945624




Page 1 of 1 Displaying 1 - 2 of 2

Eliminar

Nombre y Apellidos	ID
Carlos Sanregre Castellanos	csanregre




Page 1 of 1 Displaying 1 - 1 of 2

Anexo 7: Gestionar eventos de un grupo.

Gestión de Eventos			
 Adicionar	 Modificar	 Eliminar	
Nombre trabajo	Personal involucrado	Categoría	Evento
Sitio Web	Jefe de Grupo	Trabajos en eventos internacionales en Cuba	Jornada Científica

Page 1 of 1

Anexo 8: Gestionar publicaciones de un grupo.

Gestión de Publicaciones			
 Adicionar	 Modificar	 Eliminar	
Nombre	Personal involucrado	Estado	Categoría
Publicacion1	Responsable publicacion	Construcción	Publicaciones en la Web of Science Public

Page 1 of 1

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

Atributo	Categoría	Criterio
Responsabilidad	Baja	< =Promedio.
	Media	Entre Promedio y 2* Promedio.
	Alta	> 2* Promedio.
Complejidad de implementación	Baja	< =Promedio.
	Media	Entre Promedio y 2* Promedio.
	Alta	> 2* Promedio.
Reutilización	Baja	> 2* Promedio.
	Media	Entre Promedio y 2* Promedio.
	Alta	< =Promedio.

Rango de valores de para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) relacionados con la métrica TOC.

No	Clases	Procedimientos	Responsabilidad	Complejidad	Reutilización
1	CGrupo	44	Baja	Baja	Alta
2	CVinculación	26	Baja	Baja	Alta
3	CCurriculum	60	Media	Media	Media
4	CCurso	25	Baja	Baja	Alta
5	CTesis	45	Baja	Baja	Alta
6	CAdmin	10	Baja	Baja	Alta
7	CBuscar	20	Baja	Baja	Alta
8	MGrupo	75	Media	Media	Media
9	MVinculación	45	Baja	Baja	Alta
10	MCurriculum	80	Media	Media	Media
11	MCurso	40	Baja	Baja	Alta
12	MTesis	80	Media	Media	Media

1 3	MAdmin	20	Baja	Baja	Alta
1 4	MBuscar	70	Media	Media	Media

Resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).

Anexo 10: Instrumento de medición de la métrica Relación entre Clases (RC).

Atributo	Categoría	Criterio
Acoplamiento	Bajo	1
	Medio	2
	Alto	>2
Complejidad del mantenimiento	Baja	< =Promedio.
	Media	Entre Promedio y 2* Promedio.
	Alta	> 2* Promedio.
Cantidad de pruebas unidad	Baja	< =Promedio.
	Media	Entre Promedio y 2* Promedio.
	Alta	> 2* Promedio.
Reutilización	Baja	> 2* Promedio.
	Media	Entre Promedio y 2* Promedio.
	Alta	< =Promedio.

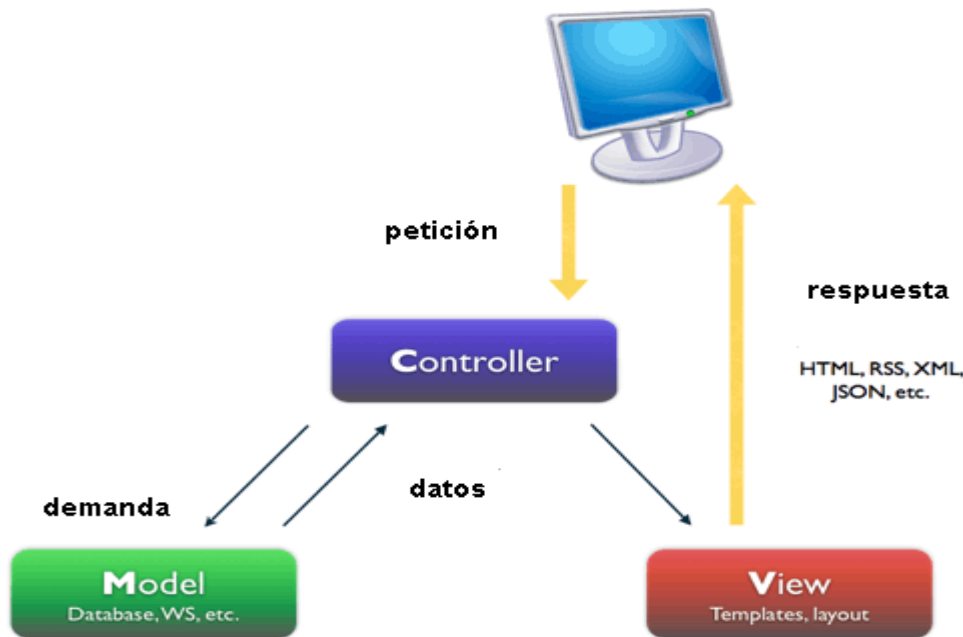
Rango de valores de para la evaluación técnica de los atributos de calidad (Acoplamiento, la Complejidad de mantenimiento, la Reutilización y la Cantidad de pruebas) relacionados con la métrica RC.

No	Clases	Cantidad de Relaciones	Acoplamiento	Complejidad de Mantenimiento	Cantidad de Pruebas	Reutilización
1	CGrupo	1	Bajo	Baja	Baja	Alta
2	CVinculación	1	Bajo	Baja	Baja	Alta
3	CCurriculum	1	Bajo	Baja	Baja	Alta
4	CCurso	1	Bajo	Baja	Baja	Alta

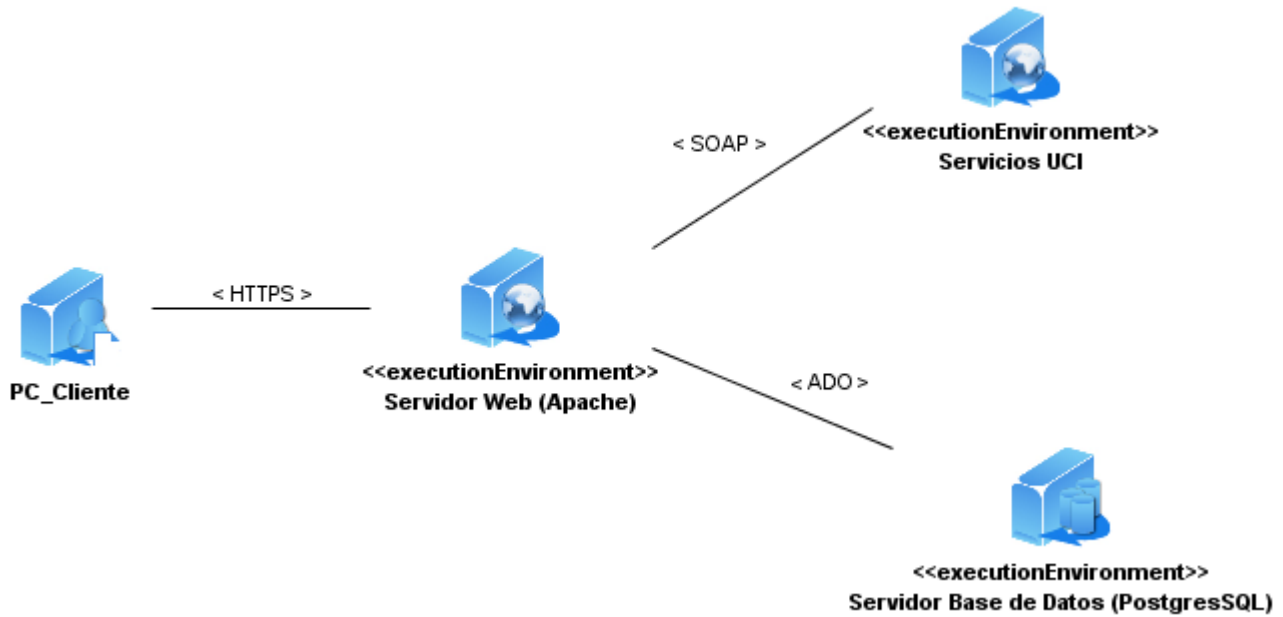
5	CTesis	1	Bajo	Baja	Baja	Alta
6	CAdmin	3	Alto	Alta	Alta	Baja
7	CBuscar	1	Bajo	Baja	Baja	Alta
8	MGrupo	1	Bajo	Baja	Baja	Alta
9	MVinculación	1	Bajo	Baja	Baja	Alta
10	MCurriculum	1	Bajo	Baja	Baja	Alta
11	MCurso	1	Bajo	Baja	Baja	Alta
12	MTesis	1	Bajo	Baja	Baja	Alta
13	MAdmin	1	Bajo	Baja	Baja	Alta
14	MBuscar	1	Bajo	Baja	Baja	Alta

Resultados de la evaluación de la métrica RC y su influencia en los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Cantidad de Pruebas y Reutilización).

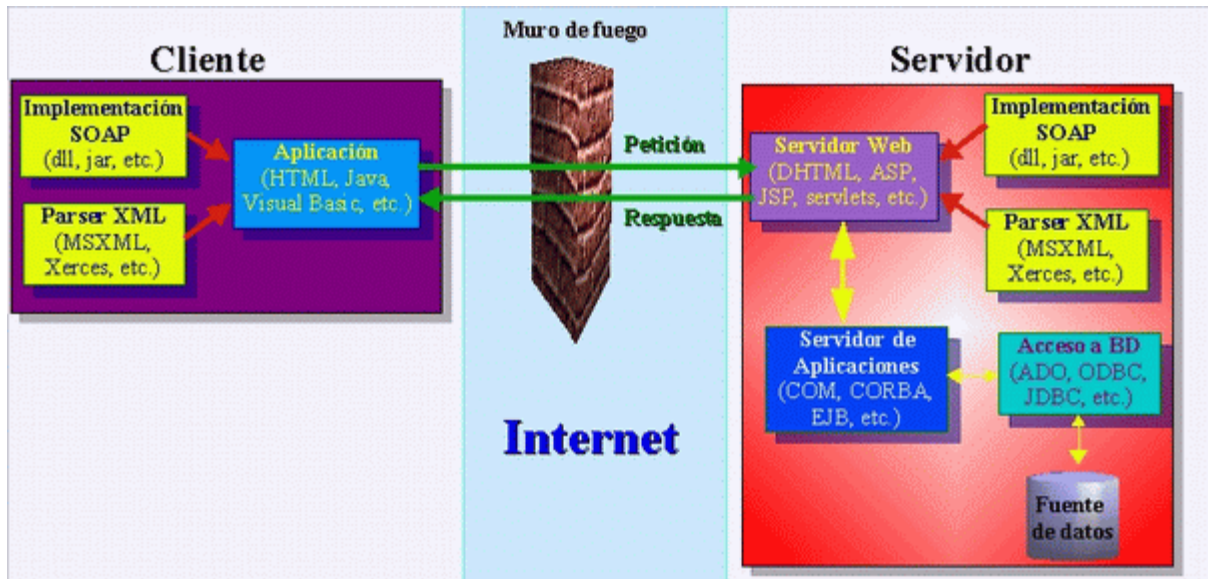
Anexo 11: Estructura del patrón Modelo-Vista-Controlador.



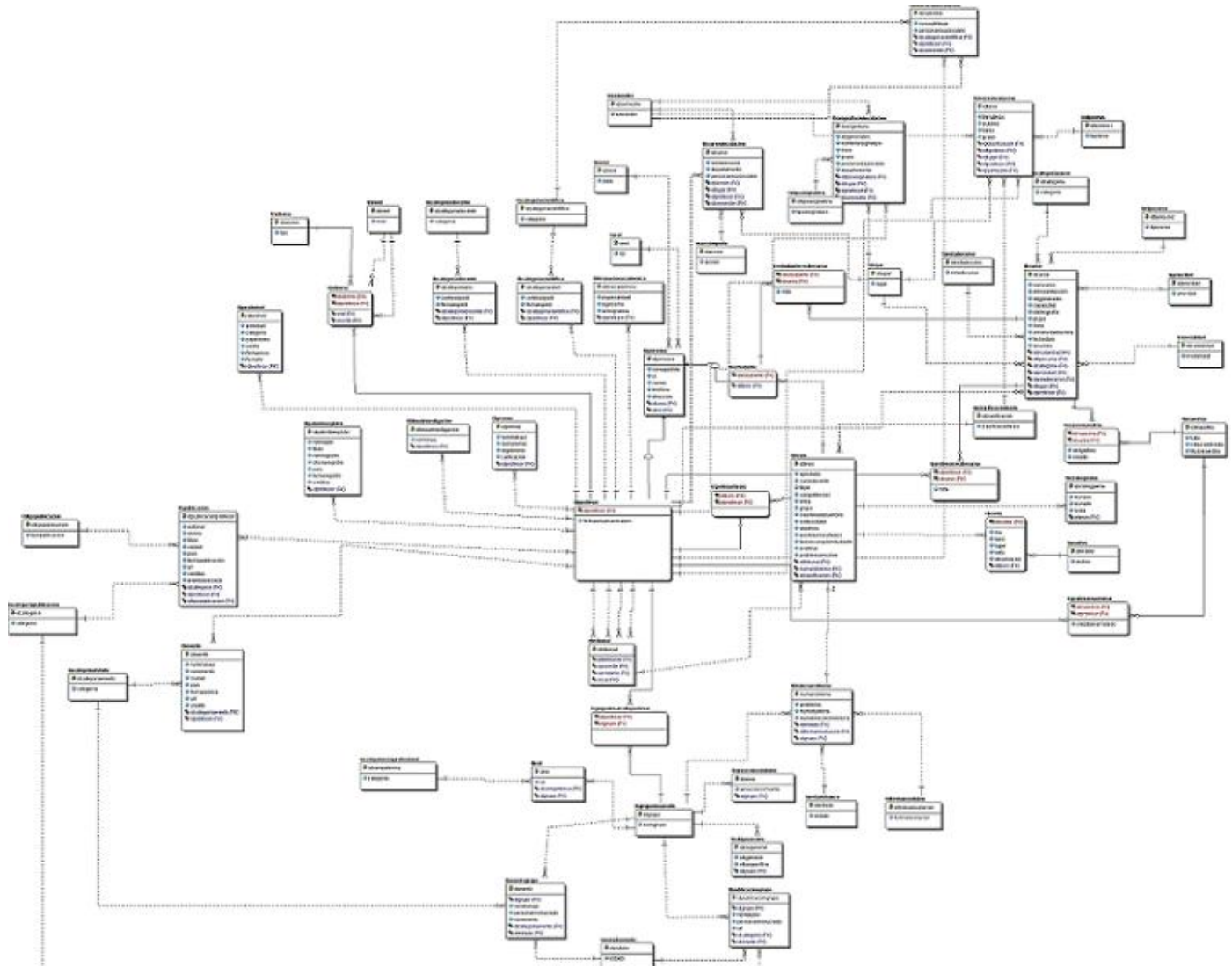
Anexo 12: Diagrama de despliegue.



Anexo 13: Funcionamiento de SOAP.



Anexo 14: Diagrama Modelo Entidad Relación.



Glosario de términos

A

API: Es una interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

AppServer: Es una herramienta Open Source para Windows que facilita la instalación de Apache, MySQL y PHP en la cual estas aplicaciones se configuran en forma automática.

B

BPM: Modelo de Procesos del Negocio.

C

CSS: Hojas de Estilo en Cascada.

CRUD: Create-Retrieve-Update y Delete.

D

DAO: (Data Access Object, Objeto de Acceso a Datos).

E

ESPOL: Escuela Superior Politécnica del Litoral.

F

Framework: Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

G

GRASP: Patrones Generales de Software para Asignación de Responsabilidades.

I

ISPJAE: Instituto Superior Politécnico José Antonio Echeverría.

M

MVC: Modelo Vista Controlador.

O

Open Source (Código abierto): Es el término con el que se conoce al software distribuido y desarrollado libremente.

R:

RC: Relación entre Clases.

RUP: Rational Unified Process.

S

SOA: Arquitectura Orientada a Servicios.

SAACP: Sistema de Administración Académica Postgrado.

SOAP: (Simple Object Access Protocol, Protocolo Simple de Acceso a Objetos).

I

TOC: Tamaño Operacional de las Clases.

X

XP: Extreme Programming.