

Universidad de las Ciencias Informáticas

“Facultad 2”



Título: *“Versión 2.0 del Sistema para la Gestión de la Información asociada a las Investigaciones y Postgrado”*

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

Autor(es): Maipú Delgado Porres
Elmer Yunion Reyes Matos

Tutor(es): Ing. Deivis Ricardo Álvarez Mendoza
Co-tutor: Ing. Albey Alfonso Navarro
Ing. Julio José Coroa Castro

Ciudad de La Habana, Junio del 2011.



"...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de si mismos..."

DECLARACIÓN DE AUTORÍA

Declaramos que Elmer Yunior Reyes Matos y Maipú Delgado Porres somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad (2) para que hagan el uso que estimen pertinente con este trabajo.

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

"[Insertar nombre(s) de autor(es)]"

"[Insertar nombre(s) de tutor(es)]"

Datos de Contacto

Tutor:

Nombre y Apellidos: Ing. Deivis Ricardo Álvarez Mendoza.

Sexo: M. **Institución:** UCI.

Dirección de la institución: Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

Correo electrónico: dalvarezm@uci.cu **Teléfono del trabajo:** 8358150.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2006. **Institución donde se graduó:** Universidad de las Ciencias Informáticas.

Co - tutor:

Nombre y Apellidos: Ing. Albey Alfonso Navarro.

Sexo: M. **Institución:** UCI.

Dirección de la institución: Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

Correo electrónico: aanavarro@uci.cu **Teléfono del trabajo:** 8358110.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2010. **Institución donde se graduó:** Universidad de las Ciencias Informáticas.

Co - tutor:

Nombre y Apellidos: Ing. Julio José Coroa Castro

Sexo: M. **Institución:** UCI.

Dirección de la institución: Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

Correo electrónico: jjcoroa@uci.cu **Teléfono del trabajo:** 8358143.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2010 **Institución donde se graduó:** Universidad de las Ciencias Informáticas.

Agradecimientos

Elmer Junior Reyes Matos

Agradezco a toda mi familia porque siempre han confiado en mí y me han ayudado en todo momento, a todas mis amistades del barrio, Reinier mi hermanito de la DC, a Maikel, Adrian, Marian, Raudel, Domingo, El Rinco, Marco y a todos los demás del piquete. También a todas las buenas amistades que he hecho aquí como Wilfredo, Osmany, Marian, Dianelys, Los Trigueños, la gente del 60 y a todos los que siempre hemos compartido en el aula, muchas gracias a todos.

A mis profesores que han ayudado a mi formación, a nuestro tutor Deivis y cotutores Albey y Julio por tener la paciencia que tienen y habernos soportado en todo momento brindándonos su apoyo que siempre estuvieron allí para cualquier duda o sugerencia, gracias a todos por ser amigos y hermanos.

A mi compañera de tesis Maipú por ser la amiga que es y por siempre estar siempre volviéndome loco con la tesis y sus ataques de histeria, gracias de verdad no podía tener una compañera de tesis mejor.

Les agradezco a todos por haber compartido los minutos buenos y malos, porque esos momentos son los que nos hacen crecer y valorar a las personas que nos rodean. Gracias a todos de verdad.

Maipú Delgado Torres

Le agradezco a toda mi familia, tíos, tías, primos, mi abuela, vecinos, amistades que siempre me han dado su apoyo tanto en los buenos momentos como en los bastante duros que la vida me ha puesto por delante y he tenido que enfrentar.

A mi mamá por darme fuerzas antes de hacer una prueba con la frasecita "Suerte y que la vida te sonría", por darme todo su amor, confianza, eres especial mi mamita.

A mi papá que está padeciendo una enfermedad terrible y por encima de todo eso no ha dejado de darme fuerzas, por hacerme dejar el miedo a un lado en momentos difíciles, por darme su confianza y apoyo, por estar siempre pendiente de mí, por ser el mejor papá del mundo.

Mi hermano y mi cuñada por apoyarme en todo, por brindarme su amor.

A mi novio por tener que soportar mis majaderías, por estar a mi lado cuando más lo he necesitado principalmente en este último año que he pasado en la Universidad que ha sido muy difícil para mí, por su amor, comprensión.

A Álvaro que voy a estar eternamente agradecida por ayudarme tanto en segundo año.

A mis amistades con las que he compartido estos cinco cursos, al 2507 que ha sido un grupo estupendo, los triqueños. A Dianelys, Danae, Malena, Toña, Eutimio que me han ayudado mucho y me han tenido que soportar mis achaques de vieja como me dicen, pero los he tenido ahí dándome fuerzas, gracias.

A mi compañero de tesis Elmer que en realidad no ha terminado loco por falta de tiempo, no lo dejo en paz, pero es un buen amigo, muy comprensivo y buen consejero.

A todos los profesores que he tenido en estos cinco años que han contribuido a mi formación. A nuestro tutor Deivis que es maravilloso y ha dedicado mucho de su tiempo a nosotros, como mismo lo han hecho los cotutores Albey y Julio que siempre nos han apoyado.

Gracias a todos!!!!

Dedicatoria

Elmer Junior Reyes matos

A mis padres, porque creyeron en mí y porque me sacaron adelante, dándome ejemplos dignos de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y porque el orgullo que sienten por mí, fue lo que me hizo ir hasta el final. Ya por ustedes, por lo que valen, porque admiro su fortaleza y por lo que han hecho de mí.

A mi hermana del alma tu no sabes cuanto te quiero por ser como eres amiga y hermana, a mi abuela, gracias por haber fomentado en mí el deseo de superación y el anhelo de triunfo en la vida mil palabras no bastarían para agradecerles su apoyo, su comprensión y sus consejos en los momentos difíciles.

A mi esposa por soportar mis chistes y mis locuras y siempre estar ahí para ayudarme en todo y levantarme los ánimos cuanto más los necesito, gracias mi amor no sabes lo mucho que significas para mí.

Maipú Delgado Torres

El presente trabajo de diploma ha sido el resultado de varios años de estudio, esfuerzo y sacrificio y con mucho amor se lo dedico a toda mi familia que es bastante extensa, especialmente a:

Mis padres por apoyarme siempre desde mis primeros años de estudios, por creer en mí y darme todo su amor, por sacrificarse y dar todo para que en un futuro fuese una profesional y ha llegado ese momento esperado, por darme aliento y fuerzas en los momentos difíciles. Doy mil gracias por tenerlos a ustedes los quiero con la vida.

Mi hermano por haber sido siempre un ejemplo a seguir para mí, darme su apoyo y saber que siempre puedo contar con él, te quiero mucho mi hermanito del alma, siempre vas a ser mi guía.

Por ustedes soy la persona que se ha formado hasta hoy y todo se lo debo al amor sin límites que me han brindado.

Resumen

En la Facultad 2 de la Universidad de la Ciencias Informáticas los procesos relacionados con las investigaciones y postgrados se gestionan de forma conjunta. En el año 2010 se elaboró un software con el objetivo de automatizar estos procesos, para de esta forma ganar en rapidez y organización, el mismo por cuestiones de tiempo solo incluyó funcionalidades básicas no cumpliendo del todo con las necesidades de la facultad. Con la realización de esta nueva versión se propone agregar al software anterior nuevas funcionalidades y mejorar algunas existentes con el objetivo de dar cumplimiento a las necesidades planteadas por el cliente.

Como resultado del trabajo desarrollado se obtuvo una nueva versión del sistema informático que incorpora y mejora posibilidades para: la gestión del currículum del profesor, el seguimiento de los postgrados cursados por profesores, la gestión de eventos y publicaciones, entre otras. El software está soportado sobre el lenguaje PHP y utiliza el gestor de base de datos PostgreSQL para el almacenamiento y gestión de los datos.



ÍNDICE GENERAL

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción	5
1.2 Estado del Arte	5
1.3 Herramientas, y metodología.....	9
1.3.1 Metodología de desarrollo	9
1.3.3 Herramienta CASE	12
1.3.4 Gestores de Base de Datos.....	13
1.3.5 Herramientas de Desarrollo	15
1.3.6 Servidor Web	16
1.4.1 Modelo Cliente Servidor.....	17
1.5 Frameworks	20
1.5.1 Dojo Toolkit	21
1.5.2 CodeIgniter	21
1.7 Conclusiones.....	22
CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA.....	23
2.2 Objeto de estudio	23
2.2.1 Problema y situación problemática.....	23
2.2.2 Objeto de automatización	24
2.3 Propuesta de sistema	24
2.4 Modelo de negocio.....	24
2.4.1 Actores y trabajadores del negocio. Justificación	24
2.4.2 Diagrama de Casos de Uso del Negocio	26
2.4.3 Descripción textual de los Casos de Uso del Negocio	26
2.4.4 Diagramas de actividades.....	26
2.4.5 Modelo de objetos	27
2.5 Requerimientos	27
2.5.1 Listado de los requerimientos funcionales.....	27
2.5.2 Definición de los requerimientos no funcionales.....	29
2.6 Modelo de Casos de Uso del Sistema.....	31
2.6.2 Descripción textual de los Casos de Uso.	33
2.7 Conclusiones.....	33
CAPÍTULO 3: DISEÑO DEL SISTEMA	34
3.1 Introducción	34
3.2 Diseño	34
3.2.1 Diagramas de Interacción.....	34
3.2.1.1 Diagramas de Secuencias del Diseño	34
3.2.2 Diagrama de clases del diseño	35
3.3 Modelo de Datos	38
3.3.1 Modelo Lógico de Datos	38



3.3.2 Modelo Físico de Datos	39
3.4 Arquitectura del Sistema	40
3.4.1 Patrones.....	40
3.4.1.1 Patrón propuesto	40
3.5 Conclusiones.....	41
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	42
4.1 Introducción.	42
4.2 Diagrama de despliegue	42
4.3 Diagrama de componentes	43
4.4 Pruebas.....	44
CAPÍTULO 5: FACTIBILIDAD DEL SISTEMA	46
5.1 Introducción	46
5.2 Método de estimación Puntos por Casos de Uso.....	46
5.3 Análisis de costos y beneficios.	53
5.4 Conclusiones.....	53
CONCLUSIONES GENERALES.....	54
RECOMENDACIONES	55
REFERENCIA BIBLIOGRÁFICA	56
BIBLIOGRAFÍA	58
ANEXOS.....	¡ERROR! MARCADOR NO DEFINIDO.
Anexo 1: Fases y Flujos de RUP	¡Error! Marcador no definido.
Anexo 2: Arquitectura Cliente Servidor.....	¡Error! Marcador no definido.
Anexo 3: Arquitectura Modelo Vista Controlador (MVC)	¡Error! Marcador no definido.
Anexo 4: Descripción textual de los casos de uso del negocio	¡Error! Marcador no definido.
Anexo 5: Diagramas de actividades.....	¡Error! Marcador no definido.
Anexo 6: Modelo de Objetos.....	¡Error! Marcador no definido.
Anexo 7: Listado extendido de los requerimientos funcionales	¡Error! Marcador no definido.
Anexo 8: Descripción textual de los casos de uso del sistema	¡Error! Marcador no definido.
Anexo 9: Diagramas de secuencia del diseño	¡Error! Marcador no definido.
Anexo 10: Diagramas de clases del diseño.....	¡Error! Marcador no definido.
Anexo11: Casos de pruebas.....	¡Error! Marcador no definido.

Índice de Figuras

Figura 1: interfaz web pública.	7
Figura 2: configuración avanzada del codeigniter.....	22
Figura 3: diagrama de cu del negocio, investigación.....	26
Figura 4: diagramas de cu del negocio, postgrado.....	26
Figura 5: diagrama de cu del sistema	31
Figura 6: diagrama secuencia, cu gestionar espacios de publicación, eliminar	35
Figura 7: diagrama secuencia, cu gestionar espacios de publicación, adicionar	34
Figura 8: diagrama de clases del diseño, gestionar evento.....	36
Figura 9: diagrama de clases del diseño, gestionar espacios de publicación.....	37



Figura 10: modelo lógico	38
Figura 11: modelo físico	39
Figura 12: patrón de diseño mvc	41
Figura 13: diagrama de despliegue.....	42
Figura 14: diagrama de componentes, acceso a datos	43
Figura 15: diagrama de componentes, código fuente.....	44

Índice de Tablas

Tabla 1: descripción del actor del negocio. investigación	24
Tabla 2: descripción de los trabajadores del negocio. investigación	25
Tabla 3: descripción del actor del negocio. postgrado.....	25
Tabla 4: descripción de los trabajadores del negocio. postgrado	25
Tabla 5: definición de los actores del sistema.....	32
Tabla 6: cálculo del factor de peso de los actores sin ajustar.	47
Tabla 7: cantidad de transacciones por casos de uso.	47
Tabla 8: cálculo del factor de peso de los casos de uso sin ajustar.....	48
Tabla 9: cálculo del factor de complejidad técnica.	50
Tabla 10: cálculo del factor ambiente.	51
Tabla 11: distribución del esfuerzo estimado entre los flujos de trabajo de <i>rup</i>	52



INTRODUCCIÓN

La Universidad de las Ciencias Informáticas (UCI) se ha convertido en una empresa productora de software. Entre sus objetivos figura el de fortalecer la economía del país y para lograrlo se ha dividido en estructuras productivas especializadas en diversos temas. Como parte de este proceso se cuenta con un amplio claustro de profesores, el mismo está integrado en gran medida por egresados de la misma institución, por lo que es de gran importancia para la Universidad dedicar tiempo para su superación tanto en el quehacer productivo como en la docencia. Para ello se ofrecen cursos de postgrado o diplomados en los cuales pueden matricular. Otra forma de superarse es participando en eventos investigativos relacionados con líneas que tributen a la Facultad. Las investigaciones realizadas por los profesores en muchos casos son presentadas en eventos científicos y publicadas, elevando el prestigio de la Universidad.

Actualmente en la Universidad existen varias facultades, las cuales tienen definidas estrategias para gestionar la información referente a las investigaciones y cursos de postgrado. En la Facultad 2 actualmente existe un sistema para gestionar dichos procesos el cual no ha sido puesto a prueba pero sí responde a algunas de las necesidades de la dirección de la facultad. Dicho sistema tiene implementado las funcionalidades básicas para la gestión de la información de las investigaciones y cursos de postgrado. Algunas de estas no fueron terminadas y faltaron por definir, las cuales permiten dar seguimiento al quehacer científico y de superación de la facultad. Algunas de las deficiencias existentes en el sistema anterior son:

1. La aplicación solamente se limita a realizar las convocatorias y el proceso de aceptación o no de un curso de postgrado o diplomado. No existe un seguimiento para conocer si el profesor terminó o no el curso o diplomado (forma de avalar la terminación del curso).
2. No existe una gestión de las publicaciones de artículos realizados por profesores.
3. Deficiente manejo del currículum del profesor.
4. No brinda la forma de gestionar los locales para impartir cursos de postgrado y por ende realizar una programación adecuada de los mismos.
5. No permite divulgar eventos científicos y tener el conocimiento de quien participa o no.
6. Insuficientes reportes informativos.
7. No permite exportar a diferentes formatos la información contenida.



8. No posee un canal para la subscripción sobre noticias y eventos.
9. El diseño relacionado con los diplomados y cursos de postgrado es ineficiente.

A partir de estas funcionalidades que no están implementadas en el sistema y para dar solución a la necesidad existente, se plantea el siguiente **problema científico**: ¿Cómo mejorar el proceso de gestión de la información referente a la Investigación y Postgrado en la Facultad 2?

De donde se deriva el siguiente **objeto de estudio**: el proceso de gestión de la información referente a la investigación y postgrado.

El **campo de acción** planteado lo constituye el proceso de gestión de la información referente a la investigación y postgrado en la Facultad 2.

Con el propósito de obtener un software que satisfaga las necesidades existentes, se traza el siguiente **objetivo general**: Desarrollar una segunda versión al Sistema para la Gestión de la Información asociada a las Investigaciones y Postgrado agregándole 13 funcionalidades complementarias.

Para resolver el problema planteado se pueden definir las siguientes **tareas de la investigación**:

1. Análisis de los procesos asociados a la gestión y control de la investigación en la Facultad 2.
2. Realización de un estudio sobre aplicaciones similares existentes.
3. Identificación de las características principales asociadas a las aplicaciones web y de escritorio.
4. Realización de un estudio del arte sobre algunas de las metodologías de desarrollo, tecnologías y herramientas a utilizar.
5. Rediseño de la base de datos.
6. Análisis y diseño de nuevas funcionalidades asociadas con el sistema y los procesos de investigaciones y postgrado.
7. Implementación de las nuevas funcionalidades.
8. Realización del estudio de factibilidad del proyecto.



Posibles resultados:

Con el presente trabajo se espera ampliar y mejorar las funcionalidades del sistema para la gestión de la información asociada a las investigaciones y postgrado de manera tal que cumpla con todos los requerimientos que la facultad necesita.

Para el desarrollo del trabajo se utilizarán los siguientes métodos científicos:

Métodos teóricos:

- **Análítico - Sintético:** para el procesamiento de la información del negocio de la organización y arribar a las conclusiones de la investigación, así como para precisar las características del trabajo a realizar.
- **Histórico - Lógico:** Para determinar las tendencias actuales de los sistemas de gestión, de los modelos de desarrollo y las técnicas, lenguajes y herramientas utilizadas durante la investigación.

Métodos empíricos:

- **Observación:** Para la percepción selectiva de las restricciones y propiedades del sistema y para la determinación de la problemática que da origen a la investigación.

El presente trabajo constará de 5 capítulos, a continuación se muestra una breve descripción de cada uno de ellos:

Capítulo 1 “Fundamentación teórica”

Se muestra el resultado de la investigación bibliográfica sobre el objeto de estudio y los sistemas de gestión de la investigación. Se detallan las metodologías y herramientas utilizadas, analizando sus características, ventajas y desventajas.

Capítulo 2 “Características del Sistema”

Incluye la descripción de los procesos del negocio que están automatizados, así como sus actores y trabajadores. Se muestran las reglas que debe cumplir el negocio, así como los modelos de objetos y diagramas de actividades de cada caso de uso. También se identifican los actores que intervienen y las funcionalidades que brinda el sistema, teniendo como resultado los requisitos funcionales y no funcionales del mismo, que darán solución a los problemas existentes.

Capítulo 3 “Diseño del Sistema”



Contiene todo lo relacionado al diseño del sistema, que incluye los diagramas de clases del diseño, y los diagramas de interacción (colaboración o secuencia).

Capítulo 4 “*Implementación y Prueba*”

Contiene la implementación del software que incluye el diagrama de despliegue, los diagramas de componentes y el patrón de arquitectura empleado así como las pruebas basadas en caso de pruebas y de estrés.

Capítulo 5 “*Factibilidad del Sistema*”

Se realiza la estimación del esfuerzo total a través del método de estimación por Puntos de Casos de Uso, así como el costo del proyecto.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el desarrollo de este capítulo se realizará un estudio de los sistemas utilizados para gestionar todo lo relacionado a la investigación o el postgrado a nivel mundial y en nuestro país. Se realizará una valoración de las posibles herramientas y tecnologías que se propusieron a utilizar en la versión anterior, así como los lenguajes de programación y metodologías posibles a utilizar para la realización del análisis, diseño e implementación del problema planteado, de forma tal que quede especificado el marco teórico de la investigación.

1.2 Estado del Arte

A continuación se mostrarán ejemplos de Sistemas de Gestión de la Información.

Estudio de Caso #1.

Sistema de Gestión de la Investigación en la Universidad de Talca, Chile

La Universidad de Talca después de tomar la decisión de transformarse en una Universidad compleja, se involucró activamente en el desarrollo de la investigación y programas de postgrado. Dentro de las actividades que se realizan en esta institución están:

La realización anual de Jornadas de Investigación.

La preparación de memorias anuales por parte de la Dirección de Investigación.

La generación de programas de postgrado, tanto magíster como doctorados.

La creación de cargos de pos doctorados; y la implementación de compromisos de gestión por unidades académicas asociados a incentivos por unidad y por académicos, según cumplimiento de compromisos.

La Universidad de Talca se planteó el desarrollo del SGI con miras a apoyar la investigación que realizan sus académicos por la vía de un sitio web que sirva de punto de encuentro entre la oferta investigativa de la Universidad y la demanda de investigación de la sociedad y las empresas, de manera tal que éstas planteen temas y problemas susceptibles de ser resueltos a través de programas y/o proyectos de investigación.



Inicialmente el sistema se estructuró de forma tal que incluyera una gama de servicios de información de utilidad exclusiva de determinados usuarios en base a su perfil, los que pueden ser ejecutivos, investigadores, visitantes y otros, respecto de programas, proyectos, eventos y productos resultantes de las actividades de investigación desarrolladas.

El sistema es capaz de:

- Mantener actualizados programas y proyectos de investigación, proyectos de tesis y sus consiguientes resultados, sean estos, publicaciones, presentaciones en congresos disciplinarios, organización de eventos, u otros.
- Responder consultas, en especial las asociadas a proyectos y disponibilidad de fondos de parte de los usuarios.
- Actualizar los estados en que se encuentran los proyectos.
- Desplegar indicadores de gestión asociados a las capacidades y resultados de la investigación que se desarrolla en la Universidad de Talca.

Para que el sistema cumpliera con las expectativas planteadas se consideró esencial implementarlo sobre una plataforma web, razón por la cual se planteó la necesidad de desarrollar un sitio web que soporte el SGI. Este sistema cuenta también con una serie de subsistemas integrados a él que lo hacen mucho más complejo, ejemplo de ello son el SGU (Sistema de Gestión de la Universidad), el SAFI (Subsistema de Gestión Administrativo y Financiero), el SAGA (Subsistema de Gestión Académica) .

Inicialmente el sistema fue desarrollado bajo una plataforma tecnológica Oracle 9i y PHP 4.3.6. Después fue presentada la web pública (ver figura 1) en la que se aplicó la metodología basada en el proceso racional unificado (RUP: Rational Unified Process) metodología que enfatiza la adopción de “mejores prácticas” en el desarrollo de aplicaciones, como una manera de reducir el riesgo inherente en la construcción de un nuevo software.

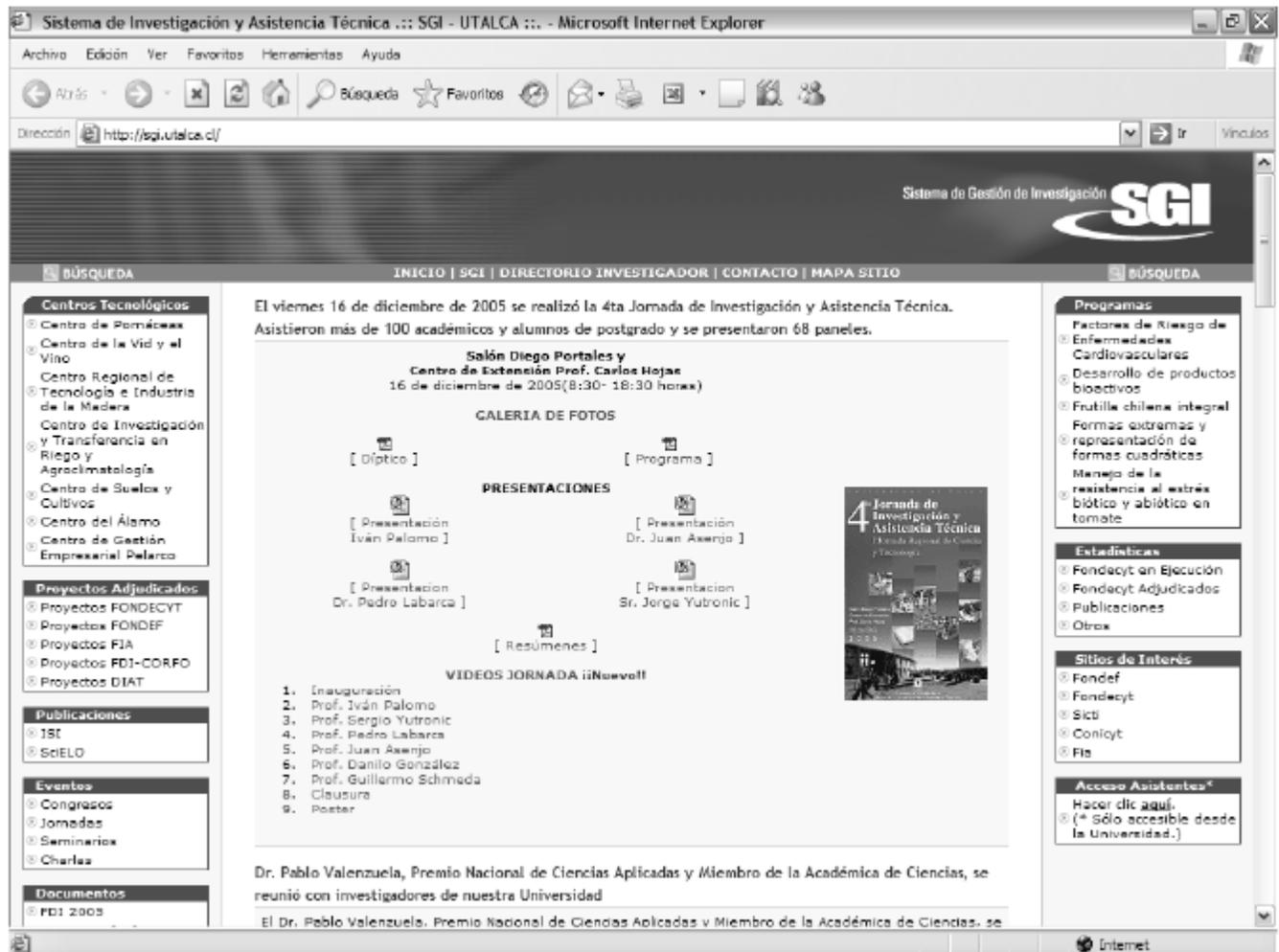


Figura 1: Interfaz web pública.

Para establecer la cooperación con los demás subsistemas el SGI, utilizó Web Services, además usó el estándar de lenguaje XML (Extensible Markup Language) para facilitar el intercambio entre sistemas dispersos y heterogéneos. También se operó sobre una plataforma tecnológica basada en Oracle Workflow versión 2.6 con interface a la Intranet, desde donde se integra al SGI mediante un enlace a una opción que permite postular proyectos nuevos a los fondos disponibles. (1)



Estudio de Caso #2.

Sistema de Gestión de Investigación en línea PGIL

Este sistema pertenece a la División de Gestión de Proyectos de Investigación DGP-CIUP, Bogotá Colombia.

Desde el año 2004 la Comunidad de Investigadores de la Universidad Pedagógica Nacional (CIUP) cuenta con un completo sistema de gestión administrativa de las convocatorias internas para proyectos de Investigación. Dicho sistema ha sido desarrollado según los requerimientos de la División de Gestión de Proyectos.

El mismo, conocido por la comunidad académica como PGIL (Proceso de Gestión de la Investigación en Línea), permite participar en las convocatorias gestionadas por la División. Brinda a todos sus investigadores y personal administrativo un nombre de usuario y contraseña para participar en las distintas convocatorias que se realizan como investigadores, coinvestigadores, monitores, decanos, etc. De esta manera, se manejan los diferentes perfiles para cada participante.

Este sistema va desde la publicación de los términos de referencia y cronograma de la convocatoria, pasando por la inscripción de líneas y grupos de investigación, la conformación de equipos de trabajo, la actualización de las hojas de vida de cada investigador, el diligenciamiento de los proyectos y de las solicitudes de presupuesto y de cargas académicas. También permite el ingreso de diversos tipos de evaluación y la publicación de informes finales del proceso.

Cuenta además con un sistema muy completo de soporte en línea para atender cualquier inquietud que pueda presentarse y con un sistema de reportes para apoyar la gestión interna administrativa y académica de la Comunidad. Para apoyar este proceso de adiestramiento en el uso de la herramienta, se realizan talleres presenciales cada año durante el desarrollo de la Convocatoria, resolviendo inquietudes de los usuarios.

Desde el 2004 el Proceso de Gestión ha dado soporte a todas las convocatorias internas para grupos con proyectos de investigación en educación y pedagogía y lanza una convocatoria entre las Universidades Pedagógica Nacional, Universidad Pedagógica y Tecnológica de Colombia y la Universidad Distrital Francisco José de Caldas.



Se han adiestrado más de 300 proyectos, que involucran a más de 900 usuarios investigadores. Ha generado mejoras significativas en la gestión de los proyectos, los tiempos para la generación de informes y en general, la gestión completa de la investigación. (2)

Un argumento importante para darle continuidad a este sistema es que actualmente en la dirección de Postgrado UCI no existe una aplicación donde se pueda gestionar la información referente a las investigaciones y cursos de postgrado, solo cuentan con una aplicación desktop en la cual la secretaria guarda la información de los profesores que matriculan. Por eso se considera de vital importancia la elaboración de este sistema de gestión de la información referente a la investigación y el postgrado en la facultad para que todos estos procesos que se llevan en documentos Word y Excel actualmente, puedan realizarse de forma automatizada, facilitándole el trabajo al vicedecano de investigación y postgrado.

1.3 Herramientas, y metodología

1.3.1 Metodología de desarrollo

Al realizar un profundo estudio se puede decir que existen diferentes metodologías de desarrollo las cuales son robustas y ágiles, que se pueden aplicar a diferentes proyectos, teniendo en cuenta el flujo de información con que se trabaja en el mismo. Dentro de las robustas encontramos: Rational Unified Process (RUP), Microsoft Solutions Framework (MSF), etc. y dentro de las ágiles tenemos Extreme Programming (XP), SCRUM, Cristal Methodologies, entre otras. A continuación se seleccionó una metodología ágil (XP) y una robusta (RUP) debido a que son las más utilizadas en la facultad y de las que se tiene más conocimiento para establecer una comparación en cuanto a sus principales características y al final llegar a conclusiones parciales del por qué se selecciona una de estas para el desarrollo del trabajo.

1.3.1.1 Extreme Programming (XP)

Extreme Programming (XP) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (3)



1.3.1.2 Proceso Unificado de Desarrollo/Rational Unified Process (RUP)

RUP es uno de los procesos más generales de los existentes actualmente, pues está pensado para adaptarse a cualquier proyecto. Un proyecto que se realice siguiendo RUP se divide en cuatro fases:

- Inicio (puesta en marcha).
- Elaboración (definición, análisis y diseño)
- Construcción (implementación)
- Transición (fin del proyecto y puesta en producción)

En cada fase se ejecutan una o varias iteraciones y dentro de cada una de ellas se seguirá un modelo de cascada o waterfall para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas. (Ver Anexo 1)

RUP define 9 actividades a realizar en cada fase del proyecto, ellas son:

- Modelado del negocio
- Análisis de requisitos
- Análisis y diseño
- Implementación
- Prueba
- Distribución
- Gestión de configuración y cambios
- Gestión de proyectos
- Gestión del entorno

Las características principales que definen a RUP son:

Guiado por casos de uso donde los casos de uso definen lo que el usuario desea a partir de la captura de requisitos y la modelación del negocio.



Centrado en la arquitectura característica que brinda una visión completa del sistema, se describen los procesos del negocio que son más importantes, para comprenderlo, desarrollarlo y producirlo de una forma eficaz.

Iterativo e incremental donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo.

Después de haber analizado las características de una metodología ágil (XP) y una robusta (RUP), y teniendo en cuenta las necesidades del sistema, se valoró que era mucho más factible optar por una metodología robusta ya que en esta el cliente no forma parte del equipo de desarrollo del proyecto, además, teniendo en cuenta el cargo del cliente y sus responsabilidades en la facultad, este no puede permanecer constantemente junto al equipo de desarrollo. RUP nos brinda una mayor documentación a la hora de elaborar el sistema y sus pasos por todos los procesos relacionados al mismo. Además en la versión anterior se utilizó RUP por lo que es más conveniente seguirla utilizando pues los artefactos que se obtienen ya no van a ser los mismos y sería muy trabajoso cambiarle la metodología debido a que no se cuenta con el tiempo suficiente para ello.

1.3.2 Lenguaje de Modelado Unificado/ Unified Modeling Language (UML)

Para fundamentar el análisis del sistema a desarrollar, se utilizará un lenguaje de modelado visual a la hora de realizar los diagramas necesarios a lo largo de todo el proceso de desarrollo, se optó por UML, ya que es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguajes de programación, así como construir modelos por ingeniería inversa a partir de programas existentes. UML no es un lenguaje altamente formal pensado para probar teoremas. Hay varios lenguajes de ese tipo, pero no son fáciles de entender ni de usar para la mayoría de los propósitos. (4)



El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

Los diagramas más comunes de UML son: (4)

- Diagrama de clases
- Diagrama de objetos
- Diagrama de casos de uso
- Diagrama de estados
- Diagrama de secuencias
- Diagrama de actividades
- Diagrama de colaboraciones
- Diagrama de componentes
- Diagrama de distribución

Se decidió continuar utilizando este lenguaje de modelado por las facilidades que aporta, las cuales anteriormente fueron mencionadas.

1.3.3 Herramienta CASE

Las herramientas CASE de modelado con UML permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraerse del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios, más fáciles de entender y modificar.

Visual Paradigm 3.4: es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Además, la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto y permite control de versiones. Cabe destacar igualmente su



robustez, usabilidad y portabilidad. En definitiva, es una herramienta muy a tener en cuenta a la hora de ponerse manos a la obra con un proyecto importante. (5)

1.3.4 Gestores de Base de Datos

Los sistemas de gestión de bases de datos o SGBD (en inglés *database management system*, abreviado *DBMS*) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El hecho general de los SGBD es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

Los SGBD poseen grandes ventajas como son: (6)

1. Proveen facilidades para la manipulación de grandes volúmenes de datos, entre éstas:

- Simplifican la programación de equipos de consistencia.
- Manejando las políticas de respaldo adecuadas, garantizan que los cambios de la base serán siempre consistentes sin importar si hay errores correctamente, etc.
- Organizan los datos con un impacto mínimo en el código de los programas.
- Bajan drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.

2. Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

Existen diferentes gestores de bases de datos como MySQL Server, PostgreSQL, Oracle, DB2, SQLite entre otros.

1.3.4.1 PostgreSQL 8.3

Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS), es extensible pues soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario. Incluye características avanzadas tales como los joins. Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos. Es un sistema Multiplataforma.

Es un sistema de bases de datos de código abierto. Aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas



SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.

Posee alta concurrencia permitiendo que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Características de PostgreSQL

A continuación se muestran una serie de características, a partir de PostgreSQL 8.3. (7)

DBMS Objeto-Relacional

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.

Integridad Referencial

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

API Flexible

La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

Lenguajes Procedurales

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

Cliente/Servidor

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Por las características antes expuestas de cada uno de los gestores de base de datos, se opta por mantener PostgreSQL debido a que sus funcionalidades se adecuan más a las necesidades que se quieren resolver.



1.3.5 Herramientas de Desarrollo

Las herramientas de desarrollo sirven para soportar los diferentes lenguajes de programación para los cuales estén destinadas.

Se tomó como herramientas de desarrollo el Zend Studio en su versión 7.1.0. A continuación se exponen algunas de las características de esta herramienta:

Zend Studio o Zend Development Environment es un completo entorno de desarrollo integrado para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

Junto con su contraparte Zend Platform, son la propuesta de Zend Technologies para el desarrollo de aplicaciones Web utilizando PHP, actuando Zend Studio como la parte cliente y Zend Platform como la parte servidora. Se trata en ambos casos de software comercial, lo cual contrasta con el hecho de que PHP es software libre. (8)

Dentro de sus características posee: (8)

No requiere la instalación previa de PHP ni del entorno de ejecución de Java.

Soporte para PHP 4 y PHP 5.

Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.

PhpDoc integrado.

Inserción automática de paréntesis y corchetes de cierre.

Detección de errores de sintaxis en tiempo real.

Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada (breakpoints), seguimiento de variables y mensajes de error del intérprete de PHP.

Soporte para gestión de grandes proyectos de desarrollo.

Manual de PHP integrado.

Soporte para control de versiones usando CVS o Subversion.



Cliente FTP integrado.

Soporte para navegación en bases de datos y ejecución de consultas SQL.

Zend Studio fue diseñado para usarse con el lenguaje PHP; sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Java Script y XML, por eso el equipo de desarrollo opta por esta herramienta para desarrollar la aplicación que se desea hacer.

1.3.6 Servidor Web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los clientes como por ejemplo los navegadores web, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada). Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar bucle infinitamente. (9)

Como ejemplos de servidores web pueden encontrarse:

Internet Information Server (IIS)

Apache Web Server

AOLServer

Hawkeye

JavaServer

Xitami, entre otros. (9)

Dentro de los ejemplos mencionados anteriormente se tomará el Apache Web Server 2.5.10 para el empleo y desarrollo del sistema.

Apache presenta ventajas tales como: (10)

Modular: Actualmente existen muchos módulos que son adaptables a él, y están disponibles para que sean instalados cuando se necesiten.

Código abierto.

Es configurable y extensible.

Multiplataforma.

Después de seguir un estudio del arte se llegó a la conclusión que era conveniente utilizar este tipo de servidor por las facilidades que brinda.



1.4 Lenguajes de programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. (11)

1.4.1 Modelo Cliente Servidor

La modalidad o arquitectura Cliente/Servidor (Ver Anexo 2) es aquella en la que confluyen una serie de aplicaciones basadas en dos categorías que cumplen funciones diferentes (una requiere servicios y la otra los brinda) pero que a la vez, pueden realizar tanto actividades en forma conjunta como independientemente. Esas dos categorías son justamente cliente y servidor. (12)

Entre las características fundamentales de esta arquitectura encontramos que tanto el cliente como el servidor pueden realizar tareas en forma conjunta como separada ya que el cliente también tiene sus propias aplicaciones, archivos y bases de datos y que además, pueden estar en la misma plataforma o en plataformas diferentes. Por otra parte, el servidor puede brindar varios servicios a la vez, tanto al mismo cliente como a clientes múltiples. (12)

1.4.1.1 Tecnologías del lado del Cliente

Cliente, es aquel que requiere un servicio del servidor. En esta categoría se realizan funciones de software basándose en el hardware pero en caso de no tener la capacidad de procesar los datos necesarios, recurre al servidor y espera a que este le brinde los servicios solicitados. El cliente es una estación de trabajo o computadora que está conectada a una red a través de la cual puede acceder al servidor. (12)

1.4.1.1.1 HTML

El lenguaje para la creación de sitios web es HTML (en inglés, HyperText Markup Language). Lo soporta una gran variedad de navegadores y exploradores, además su compatibilidad con los navegadores más antiguos y también disminuye el costo de las personas que empiezan a aprender a leer y a escribir HTML, siendo un lenguaje fácil de interpretar y de aprender. La mayoría de los documentos que existen en la WWW (en inglés, World Wide Web) se escriben en HTML. (13)

1.4.1.1.2 Java Script

Java Script es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web.



Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario. Es compatible con la mayoría de los navegadores modernos, también es un lenguaje interpretado que no requiere compilación, utilizado principalmente en páginas web. No requiere un tiempo de compilación, los scripts se pueden desarrollar en un período de tiempo relativamente corto. Los programas Java Script tienden a ser pequeños y compactos, no requieren mucha memoria ni tiempo adicional de transmisión. Además, al incluirse dentro de las mismas páginas HTML se reduce el número de accesos independientes a la red. (13)

1.4.1.1.3 CSS

CSS son las siglas de Cascading Style Sheets, en español Hojas de estilo en Cascada. Es una tecnología que permite crear páginas web de una manera más exacta. Gracias a las CSS somos mucho más dueños de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, entre otras. Incluso podemos definir nuestros propios estilos en un archivo externo a nuestras páginas; así, si en algún momento queremos cambiar alguno de ellos, automáticamente se nos actualizarán todas las páginas vinculadas de nuestro sitio. Al emplear el lenguaje de estilo CSS, el archivo de HTML tiene un tamaño menor y los navegadores pueden leer sencillamente un sitio web. (13)

1.4.1.2 Tecnologías del lado del Servidor

Servidor es la máquina desde la que se suministran servicios y que está a la espera del requerimiento del cliente. Una vez hecho, busca la información solicitada y le envía la respuesta al cliente; incluso puede enviar varios servicios a la vez, lo que es posible porque entre ellos están conectados mediante redes. (12)

Existen varios lenguajes de programación para trabajar en las tecnologías del lado del servidor como son ASP, ASP.NET, PHP, JSP, etc. Mencionando algunas de las características de uno de estos lenguajes podemos poner por ejemplo que ASP.NET es un lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia de mercado denominada .NET. Fue creado para desarrollar webs sencillas o grandes aplicaciones. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. (12)

Dentro de las ventajas con las que cuenta ASP.NET están:



Completamente orientado a objetos.

División entre la capa de aplicación o diseño y el código.

Incremento de velocidad de respuesta del servidor.

Dentro de sus principales desventajas está:

Mayor consumo de recursos.

1.4.1.2.1 PHP 5.0

PHP es un lenguaje de scripting embebido en HTML. Mucha de su sintaxis es tomada de C, Java y Perl con un par de características adicionales únicas y específicas de PHP, esto permite a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones. El propósito del lenguaje es permitir que los desarrolladores web escriban páginas generadas dinámicamente con rapidez. PHP significa *PHP: Hypertext Preprocessor* (PHP: Pre-procesador de Hipertexto).

Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP presenta una serie de ventajas tales como: (14)

Es un lenguaje multiplataforma.

Se utiliza en el desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.

El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de interpretar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.



Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL

Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).

Posee una amplia documentación, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.

Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

Permite aplicar técnicas de programación orientada a objetos.

No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

Tiene manejo de excepciones (desde PHP5).

Después de un estudio del arte se llegó a la conclusión que es conveniente utilizar estos lenguajes de programación por las facilidades que brindan.

1.5 Frameworks

Un framework es una estructura conceptual y tecnológica de soporte definida, normalmente, con artefactos de software concretos, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Un framework define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. (15)

La mayoría de frameworks web se encargan de ofrecer una capa de controladores de acuerdo con el patrón Modelo Vista Controlador (Ver Anexo 3), ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación. (16)

Cuando se va a desarrollar un sistema se puede trabajar con varios tipos de frameworks, algunos orientados al cliente que no son más que la interfaz externa que les brindan a los usuarios como son Dojo



Toolkit, Extend JS, etc. y otros orientados al servidor como por ejemplo Symfony y CodeIgniter. Para el desarrollo del sistema se usará Dojo Toolkit y CodeIgniter.

1.5.1 Dojo Toolkit

Dojo es un framework que contiene APIs y widgets (controles) para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Contiene un sistema de empaquetado inteligente, los efectos de UI, drag and drop APIs, widget APIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX.

Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas (bookmarking), y la habilidad de degradar cuando AJAX/Java Script no es completamente soportado en el cliente. Es conocido como "la navaja suiza del ejército de las bibliotecas Java Script". Proporciona una gama más amplia de opciones en una sola biblioteca Java Script y es compatible con navegadores antiguos.

Los complementos de Dojo son componentes preempaquetados de código Java Script, HTML y CSS que pueden ser usados para enriquecer aplicaciones web. Es un framework multiplataforma. (17)

1.5.2 CodeIgniter

CodeIgniter es un framework PHP de gran alcance, construido para programadores de PHP que necesitan una guía sencilla y elegante para crear aplicaciones web. 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. El uso de CodeIgniter se basa en el patrón Modelo-Vista-Controlador (MVC), lo que facilita mucho estructurar la aplicación y su mantenimiento. El manejo y abstracción de base de datos es bueno, e implementa el patrón Active Record. La división en models, views, controllers, libraries, helpers, config, es muy práctica y comprensible. Los mecanismos de ruteo de CodeIgniter favorecen a éste y otras aplicaciones. (18)

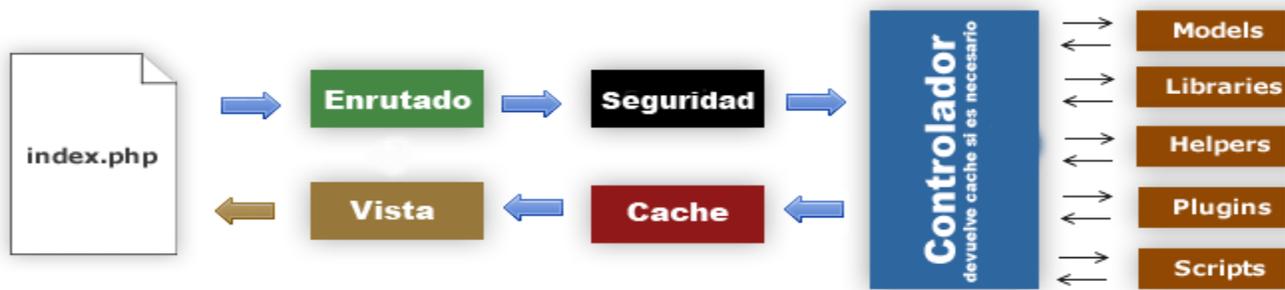


Figura 2: Configuración avanzada del Codelgniter.

1. El **index.php** inicializa el núcleo de Codelgniter.
2. El enrutador **examina la petición HTTP** y determina que se debe de hacer.
3. **Seguridad**, antes de que se cargue el controlador se filtra los datos enviados para que estos sean fiables.
4. El **controlador** comprueba **si existe cache**, en caso afirmativo la devuelve pero al hacer la comprobación en este punto podemos llegar a pasar datos dinámicos a la vista (podríamos conseguir que funcionará un módulo de autenticación aún con la cache habilitada), **si no existe cache** carga el modelo, librerías, plugins, helpers y todos los recursos necesarios para satisfacer la petición.
5. Una vez la **Vista está renderizada es enviada al navegador** si la cache está habilitada se almacena el resultado en cache para la próxima vez que esta URL sea servida.

Es un framework con gran flexibilidad. No fuerza al usuario a utilizar una Base de Datos para un desarrollo. Además, su separación de código PHP y código HTML no está basada en un template en particular, puede usarse incluso sin este. Es un framework Multiplataforma.

1.7 Conclusiones

En este capítulo se detalló con claridad la necesidad existente de la creación de una nueva versión del SGI de la Facultad 2, mejorando las funcionalidades existentes en dicho software y agregándole nuevas. También se hizo un estudio de las herramientas y metodologías utilizadas en la versión anterior y se tomó la decisión de mantener las mismas justificando cada una de estas elecciones.



CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción:

En este capítulo se abordan elementos relacionados con el objeto de estudio que sirven de base fundamental para el desarrollo del Sistema para la Gestión de la Información asociada a las Investigaciones y Postgrado, quedando expuestos los aspectos que tributan a la descripción de la propuesta de solución.

También se muestra el negocio para poder tener una visión clara del proyecto a realizar, se definen además los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos. Se realiza el levantamiento de los nuevos requisitos funcionales y no funcionales que tendrá la aplicación, a partir de ello se realizará el modelado de la misma, exponiendo solamente los casos de uso críticos, o sea, los más importantes para el funcionamiento del sistema.

2.2 Objeto de estudio

2.2.1 Problema y situación problemática

La **situación problemática** radica en que: actualmente en la Universidad existen varias facultades, las cuales tienen definidas estrategias para gestionar la información sobre las investigaciones y los cursos de postgrado o diplomado. En la Facultad 2 actualmente existe un sistema para gestionar dichos procesos el cual no ha sido desplegado ya que solo responde a algunas de las necesidades de la dirección de la facultad. Dicho sistema tiene implementada las funcionalidades básicas para la gestión de la información de las investigaciones, cursos y programas de postgrado, algunas de estas no fueron terminadas y faltan otras funciones que permitan dar seguimiento al quehacer científico y de superación de la facultad como por ejemplo: La aplicación solamente se limita a realizar las convocatorias y el proceso de aceptación o no de un curso de postgrado o diplomado. No existe un seguimiento para conocer si el profesor terminó o no el curso o diplomado (forma de avalar la terminación del curso). No posee una gestión asociada con las publicaciones de artículos de los profesores. Deficiente manejo del currículum del profesor. No brinda la forma de gestionar los locales para impartir cursos de postgrado y por ende realizar una programación adecuada de los locales. No permite divulgar eventos científicos y tener el conocimiento de quien participa o no. Existen insuficientes reportes informativos. No permite exportar a diferentes formatos la información



contenida. No permite tener un canal para la suscripción sobre noticias y eventos. El diseño relacionado con los diplomados y cursos de postgrado es ineficiente. La **propuesta es** ampliar las funcionalidades del Sistema para la Gestión de la Información asociada a las Investigaciones y Postgrado, logrando una mayor rapidez en el manejo de la información referente a los procesos.

2.2.2 Objeto de automatización

Como **objeto de automatización** centramos la atención en dos procesos, estos son:

- Procesos de gestión de la información relacionada a la investigación en la Facultad 2.
- Procesos de gestión de la información relacionada al postgrado en la Facultad 2.

2.3 Propuesta de sistema

Debido a los problemas existentes en la gestión de la información relacionada con la investigación y el postgrado llevados a cabo en la facultad, se decidió mejorar el sistema informático que gestiona estos procesos. El mismo brindará amplias facilidades para almacenar y manejar la información relacionada con ambos procesos. Ofrece la posibilidad de acceder a la información almacenada rápidamente y en el momento que se requiera. Permitirá a los usuarios conectarse y tener acceso a la información de acuerdo con su rol y privilegios dados para consultar, actualizar, adicionar, entre otras. Este sistema contará con varios roles de acuerdo con las responsabilidades que tenga cada rol en el proceso de gestión de información referente a la investigación y el postgrado en la Facultad 2, todos en general pueden consultar la información en el momento que lo requiera, a través de los reportes que ofrezca el sistema.

2.4 Modelo de negocio

2.4.1 Actores y trabajadores del negocio. Justificación

Actores del negocio	Justificación
Profesor	Se encarga de solicitar la creación de un proyecto de investigación.

Tabla 1: Descripción del actor del negocio. Investigación



Trabajadores del negocio	Justificación
Asesor de Investigación	Interesado en controlar la gestión de la información asociada a la investigación de los profesores y dirige la Comisión Científica.
Comisión Científica	Es la encargada de valorar y analizar la Planilla del proyecto de investigación para aceptar o denegar la solicitud del profesor.
Grupo de Investigación	Encargados de revisar la Planilla del proyecto para valorar la aprobación de la misma.

Tabla 2: Descripción de los trabajadores del negocio. Investigación

Actores del negocio	Justificación
Profesor	Solicita inscribirse en alguna de las actividades de postgrado que se estén ofertando.

Tabla 3: Descripción del actor del negocio. Postgrado

Trabajadores del negocio	Justificación
Asesor de Postgrado	Es el encargado de analizar las solicitudes de los profesores para poder aceptar o no su participación en algunas de las actividades de postgrado.
Consejo de Dirección	Encargado de analizar entre todos a los profesores solicitantes y aprobar o no su participación en alguna de las actividades de postgrado.
J' Dpto.	Encargado de analizar a los profesores de su departamento y aprobar o no su participación en alguna de las actividades de postgrado.
Profesor imparte curso	Encargado de impartir un curso de postgrado y de asignarle una nota a los profesores que reciben el curso.

Tabla 4: Descripción de los trabajadores del negocio. Postgrado

2.4.2 Diagrama de Casos de Uso del Negocio

Diagrama de Casos de Uso del Negocio, Investigación

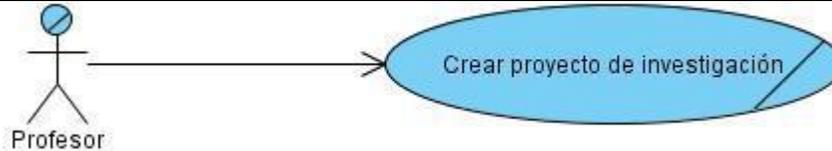


Figura 3: Diagrama de CU del negocio, Investigación.

Diagrama de Casos de Uso del Negocio, Postgrado

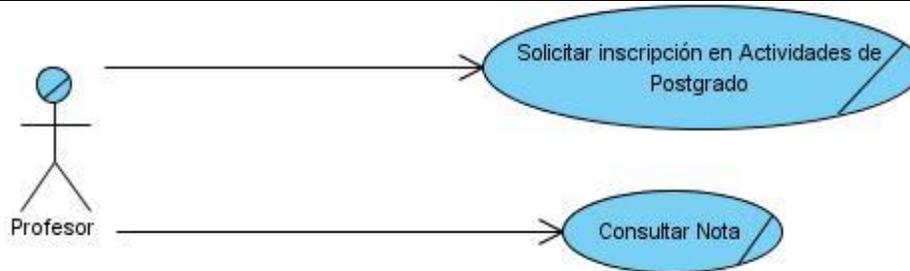


Figura 4: Diagramas de CU del Negocio, Postgrado.

2.4.3 Descripción textual de los Casos de Uso del Negocio

La descripción de los casos de uso del negocio, describen detalladamente la interacción de los actores del negocio con el negocio, (Ver Anexo 4).

2.4.4 Diagramas de actividades

Los diagramas de actividades explican las actividades que ocurren dentro del proceso del negocio, separando las actividades que realizan los actores del negocio de las actividades que realizan los trabajadores del negocio, (Ver Anexo 5).



2.4.5 Modelo de objetos

El modelo de objeto representa las clases entidades del negocio con los trabajadores que interactúan con ellas. (Ver Anexo 6).

2.5 Requerimientos

Los requerimientos son las condiciones o capacidades que tiene que ser alcanzada por un sistema para satisfacer al cliente o usuario final.

Los requerimientos se clasifican en dos tipos: requerimientos funcionales y requerimientos no funcionales. Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir mientras que los requerimientos no funcionales son características o propiedades que el sistema debe tener para ser un producto con mucha aceptación.

2.5.1 Listado de los requerimientos funcionales

- RF-1 Autenticar usuario
- RF-2 Gestionar usuarios
- RF-3 Gestionar roles
- RF-4 Gestionar funcionalidades
- RF-5 Gestionar módulos
- RF-6 Gestionar cursos de postgrado *
- RF-7 Gestionar programas de postgrado
- RF-8 Listar solicitudes por curso de postgrado
- RF-9 Listar solicitudes por programa de postgrado
- RF-10 Gestionar líneas de investigación
- RF-11 Gestionar grupos de investigación
- RF-12 Gestionar líneas temáticas
- RF-13 Gestionar proyectos de investigación
- RF-14 Realizar solicitud de cursos de postgrado
- RF-15 Realizar solicitud de programas de postgrado
- RF-16 Actualizar Curriculum
- RF-17 Listar cursos y programas de postgrado



RF-18 Gestionar eventos +

Es una funcionalidad que tiene el Vicedecano de Investigación y Postgrado para gestionar todo lo relacionado con los eventos y noticias de interés para los profesores.

RF-19 Gestionar espacios de publicación +

Mediante esta funcionalidad se desea propiciar un espacio a los profesores donde puedan adicionar o eliminar sus publicaciones.

RF-20 Generar reportes sobre actividades de un profesor. +

A través de esta funcionalidad se puede conocer todo lo relacionado con el quehacer científico de un profesor.

RF-21 Generar reporte de Balance General +

De esta forma se puede obtener un reporte estadístico en cantidad y porcentaje de la cantidad de profesores existentes en un determinado grado científico y categoría docente.

RF-22 Buscar integrantes por cursos +

Mediante esta funcionalidad se puede obtener los profesores matriculados en un curso determinado.

RF-23 Generar reportes de Postgrado

RF-24 Gestionar notas +

Controlar la asignación de notas a los profesores matriculados en un curso de postgrado y el estado en dicho curso.

RF-25 Exportar. +

Es una opción que permite exportar determinada información en formato Word.

RF-26 Generar reportes avanzados. +

A partir de este requerimiento se puede visualizar los profesores existentes por categoría docente, grado científico y cursos o diplomados.

RF-27 Gestionar locales. +

El propósito de esta funcionalidad es gestionar los locales en los que se va a impartir los cursos de postgrado.

RF-28 Suscribirse a canal RSS. +

Con este canal se quiere mantener informado a los usuarios sobre la publicación de nuevos eventos y noticias.



RF-29 Generar Curriculum. +

Se desea generar el curriculum vitae de un profesor.

RF-30 Gestionar canal de mensajería.+

Brinda una vía de comunicación dentro del sistema.

RF-31 Planificar local+

Con esta funcionalidad se pretende hacer una planificación de los locales donde se va a impartir un curso de postgrado o diplomado, así como ver dicha planificación.

() Significa que no se ha modificado.

(*) Significa las funcionalidades que serán mejoradas.

(+) Significa las nuevas funcionalidades.

El listado extendido de los requerimientos funcionales se muestra en los anexos, (Ver Anexo 7)

2.5.2 Definición de los requerimientos no funcionales

A continuación se presentan los requerimientos no funcionales mínimos que debe tener el sistema para su correcto funcionamiento.

- Rendimiento
 - Se esperan respuestas de manera eficiente en un tiempo reducido de un segundo y un gasto mínimo de recursos de 450 Kb.
- Soporte
 - Se debe realizar un mantenimiento a la aplicación en caso de utilizar nuevas versiones de las herramientas que se empleen, así como un mantenimiento mensual a la base de datos.
- Portabilidad:
 - El sistema será multiplataforma, brindando la posibilidad de desplegarlo sobre diferentes Sistemas Operativos como Linux y Windows.
- Seguridad



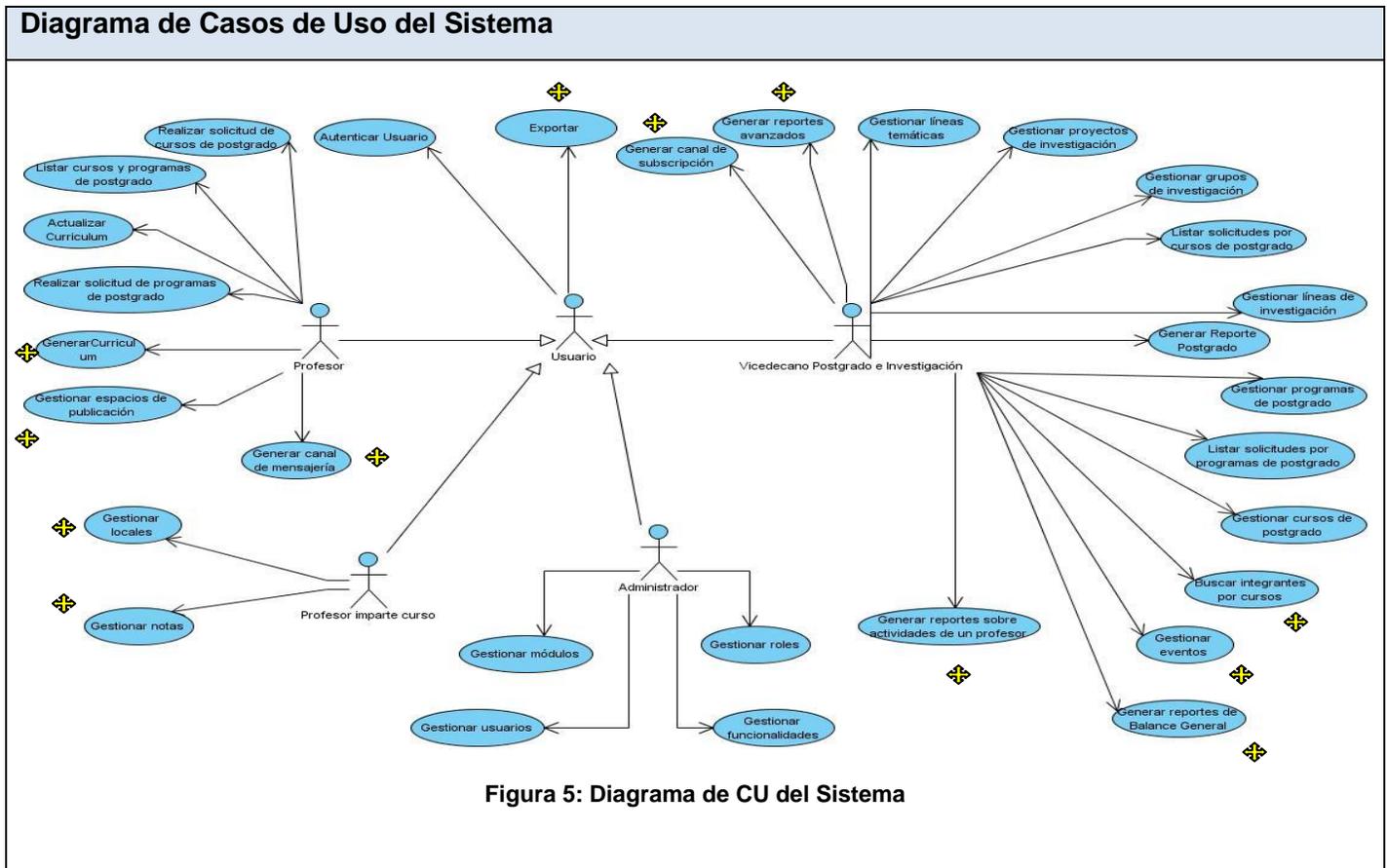
- Disponer de un mecanismo de seguridad basado en el modelo de SAAA (Single Authentication, Authorization and Accounting).
 - El sistema debe contar con técnicas de validación de inyecciones de código SQL, script y Cross-site scripting.
 - **Confiabilidad:** La información manejada por el sistema está protegida de acceso no autorizado.
 - **Integridad:** La información que en el sistema se gestiona debe mantenerse de forma íntegra.
 - **Disponibilidad:** Se utiliza el servicio de autenticación para dar acceso a los usuarios que interactúan con el sistema. Solo los usuarios autorizados tendrán acceso a la información en todo momento.
- Software
 - **Para el cliente:**
 - Sistema Operativo Windows o Linux.
 - Navegador web Internet Explorer superior al 6.0, Mozilla Firefox 2.5 en adelante u otros.
 - **Para el Servidor:**
 - Sistema operativo Windows XP, o Linux en cualquiera de sus distribuciones.
 - Servidor Apache 2.5.10 con módulo PHP disponible.
 - Un servidor de Base de Datos PostgreSQL 7.1.x o versiones superiores.
 - Hardware
 - **Para el Cliente:**
 - Procesador Pentium IV con 256 Mb de memoria RAM como mínimo.



- Para el servidor:
- Procesador Pentium IV a 1GHz de velocidad de procesamiento y 512Mb de memoria RAM o superior para un mejor funcionamiento.
- Requerimientos mínimos de espacio de almacenamiento 80Gb de disco duro.

2.6 Modelo de Casos de Uso del Sistema

El diagrama de casos de uso del sistema muestra la relación existente entre los procesos o casos de uso del sistema y los actores del sistema. Estos casos de uso son resultado de la captura de requisitos funcionales que son los que indican las funcionalidades que debe cumplir el sistema.





2.6.1 Definición de los actores del sistema a automatizar

Actores del sistema	Justificación
Usuario	Es un usuario que generaliza el rol de autenticación al sistema.
Profesor	Es el encargado de realizar las solicitudes y controlar todo lo relacionado a su Curriculum.
Vicedecano Postgrado e Investigación	Es el encargado de gestionar todo lo relacionado a Postgrado e Investigación.
Administrador	Es el encargado de administrar el sistema y gestionar los usuarios
Profesor imparte curso	Gestiona lo relacionado con locales y notas.

Tabla 5: Definición de los actores del sistema

La implementación de los casos de uso se ordenó según la prioridad que se le asignó a cada uno de estos. La tabla que se muestra a continuación representa los casos de uso y el ciclo en el que se realizó cada uno según la prioridad en la primera iteración.

Nombre del Caso de Uso	Ciclo	Justificación
Gestionar eventos	1	Estos casos de usos representan las funcionalidades más importantes a implementar en esta iteración para un mejor funcionamiento del sistema.
Gestionar espacios de publicación		
Generar reportes de Balance General		
Generar reportes sobre actividades de un profesor		
Gestionar notas		
Generar reportes avanzados		
Gestionar locales		



Planificar local		
Gestionar Curriculum		
Buscar integrantes por cursos	2	Estas no constituyen funcionalidades básicas, por lo que se pueden implementar después de haber implementado las anteriores.
Exportar recursos en diferentes formatos		
Generar canal de suscripción		
Gestionar canal de mensajería		

2.6.2 Descripción textual de los Casos de Uso.

La descripción de los casos de uso del sistema, describen de forma detallada la interacción de los actores del sistema con el sistema, (Ver Anexo 8)

2.7 Conclusiones.

En este capítulo se reflejaron las características que definen el negocio y se describieron los nuevos procesos que se le incorporaron al mismo, también se definieron las nuevas actividades que serían objeto de automatización y las que se van a mejorar las cuales conforman los requisitos funcionales que debería tener el sistema para darle solución al problema planteado. También se delimitaron los requisitos no funcionales que el sistema debería cumplir para lograr satisfacer al cliente. Además, se definió qué debería hacer la aplicación para cumplir lo propuesto en los requisitos funcionales a través de la descripción de los casos de uso.

CAPÍTULO 3: DISEÑO DEL SISTEMA

3.1 Introducción

En este capítulo se convierten los requisitos a una especificación que describe cómo implementar el sistema, elaborando así, un diseño que cumpla con todos los nuevos requisitos establecidos. Se muestran los diagramas de clases del diseño de los casos de uso y diagramas de interacción. Así como el diseño de la base de datos, después de haberle incluido las nuevas tablas y campos correspondientes, dentro de éste, el modelo lógico y el modelo físico.

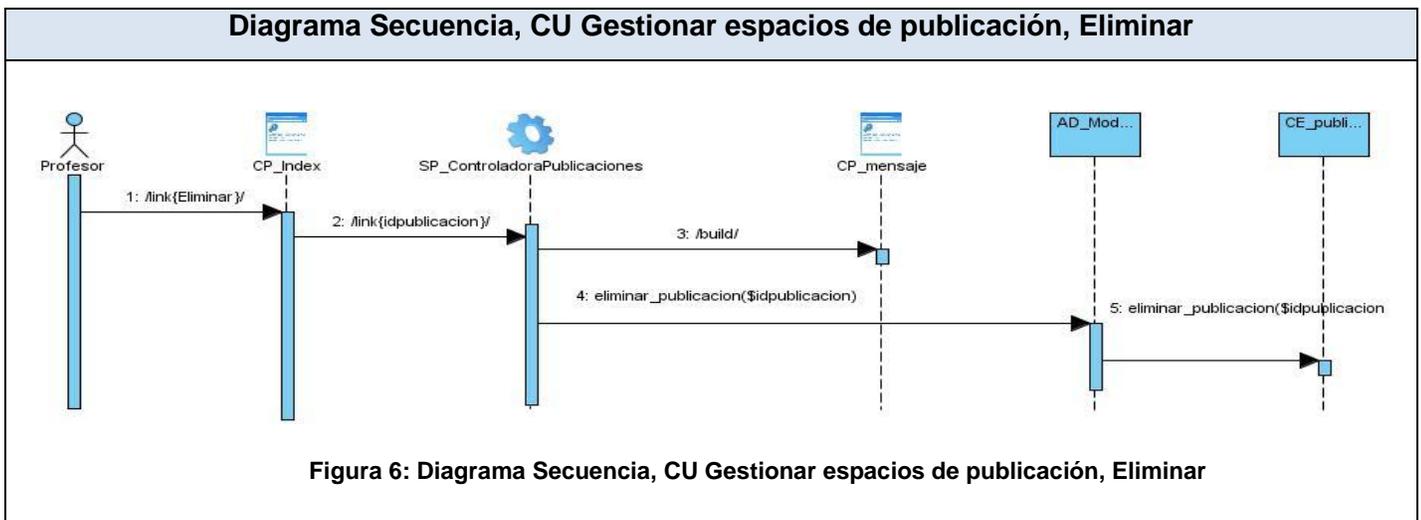
No se realizará el análisis del sistema debido a que RUP propone que, para realizar el proceso de implementación de un software no necesariamente se deben llevar a cabo los procesos del análisis, pues con solo hacer el diseño y tenerlo bien definido, se están analizando los diferentes procesos que se van a informatizar.

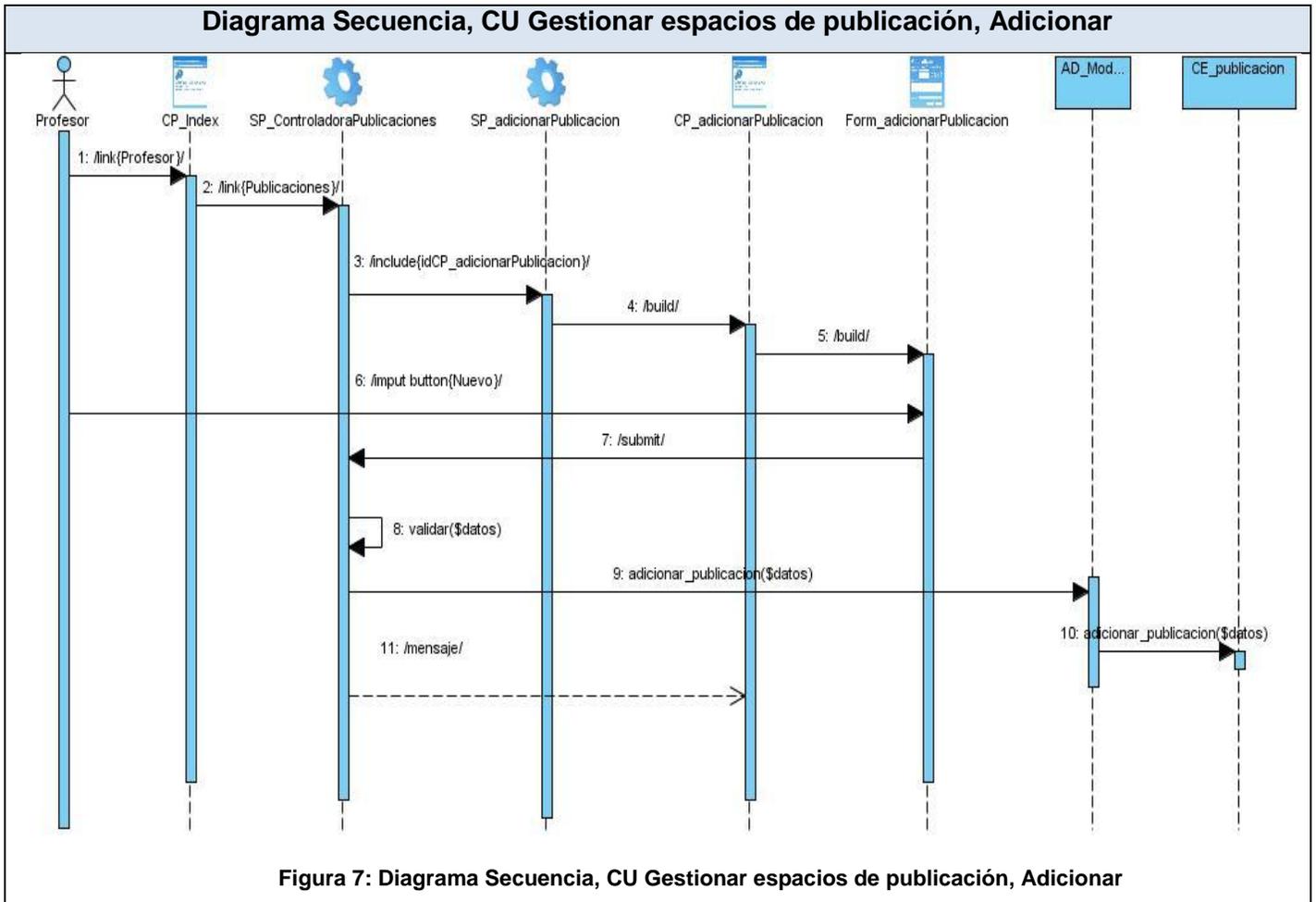
3.2 Diseño

3.2.1 Diagramas de Interacción

3.2.1.1 Diagramas de Secuencias del Diseño

Los diagramas de secuencia, muestran la ordenación temporal de los mensajes modelando la interacción entre los objetos. Los objetos suelen ser instancias con nombre o anónimas de clases, pero también pueden representarse instancias de otros elementos, tales como colaboraciones, componentes y nodos.

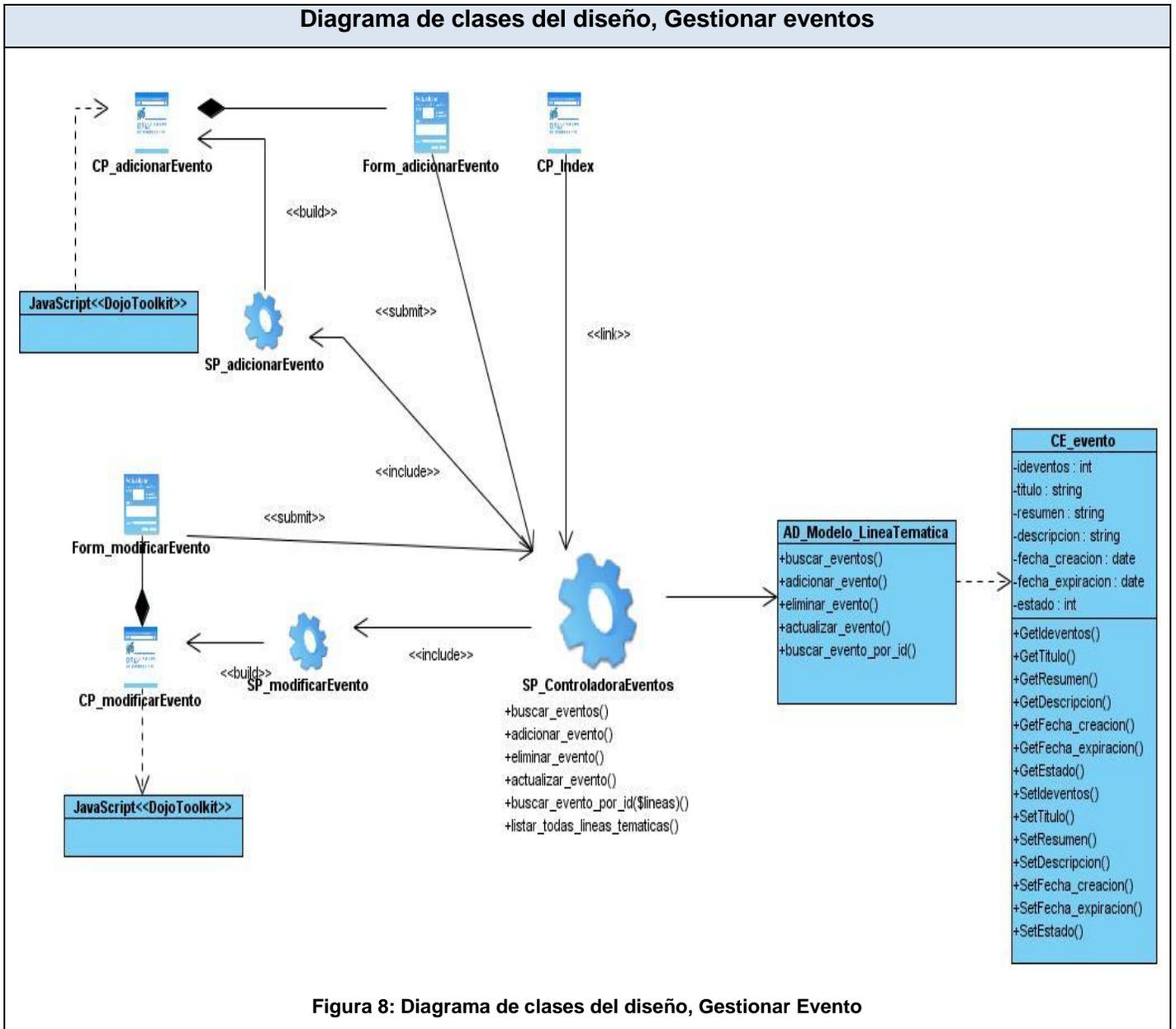


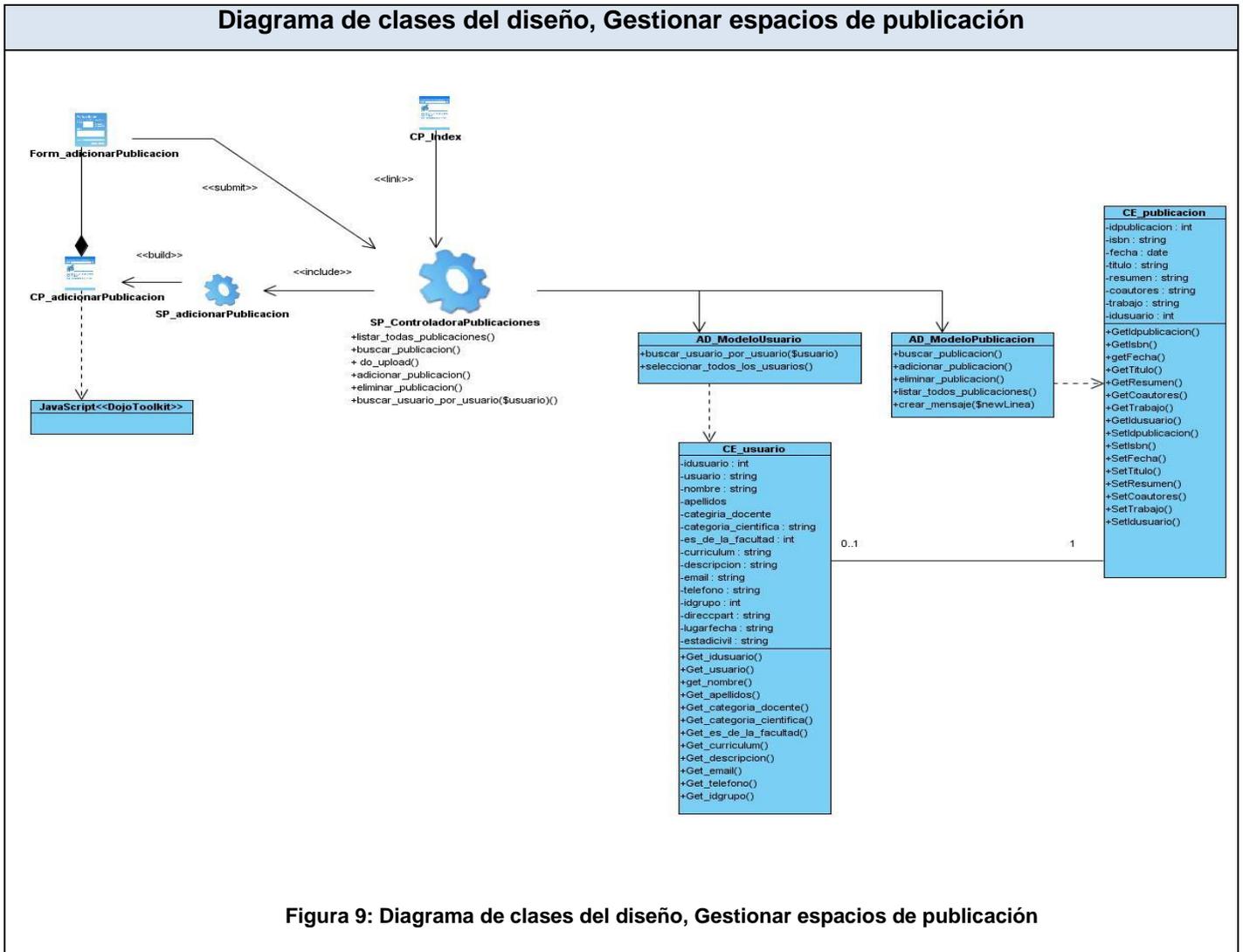


Los demás diagramas se podrán ver en los anexos. (Ver Anexo 9).

3.2.2 Diagrama de clases del diseño

Los diagramas de clase del diseño representan a las clases de diseño y las relaciones entre ellas. Las clases del diseño contienen atributos y métodos. En los distintos diagramas de clases del diseño se representa el patrón de arquitectura Modelo Vista Controlador (MVC).





Los demás diagramas se podrán ver en los anexos (Ver anexo 10)



3.3 Modelo de Datos

3.3.1 Modelo Lógico de Datos

Diagrama de clases persistentes

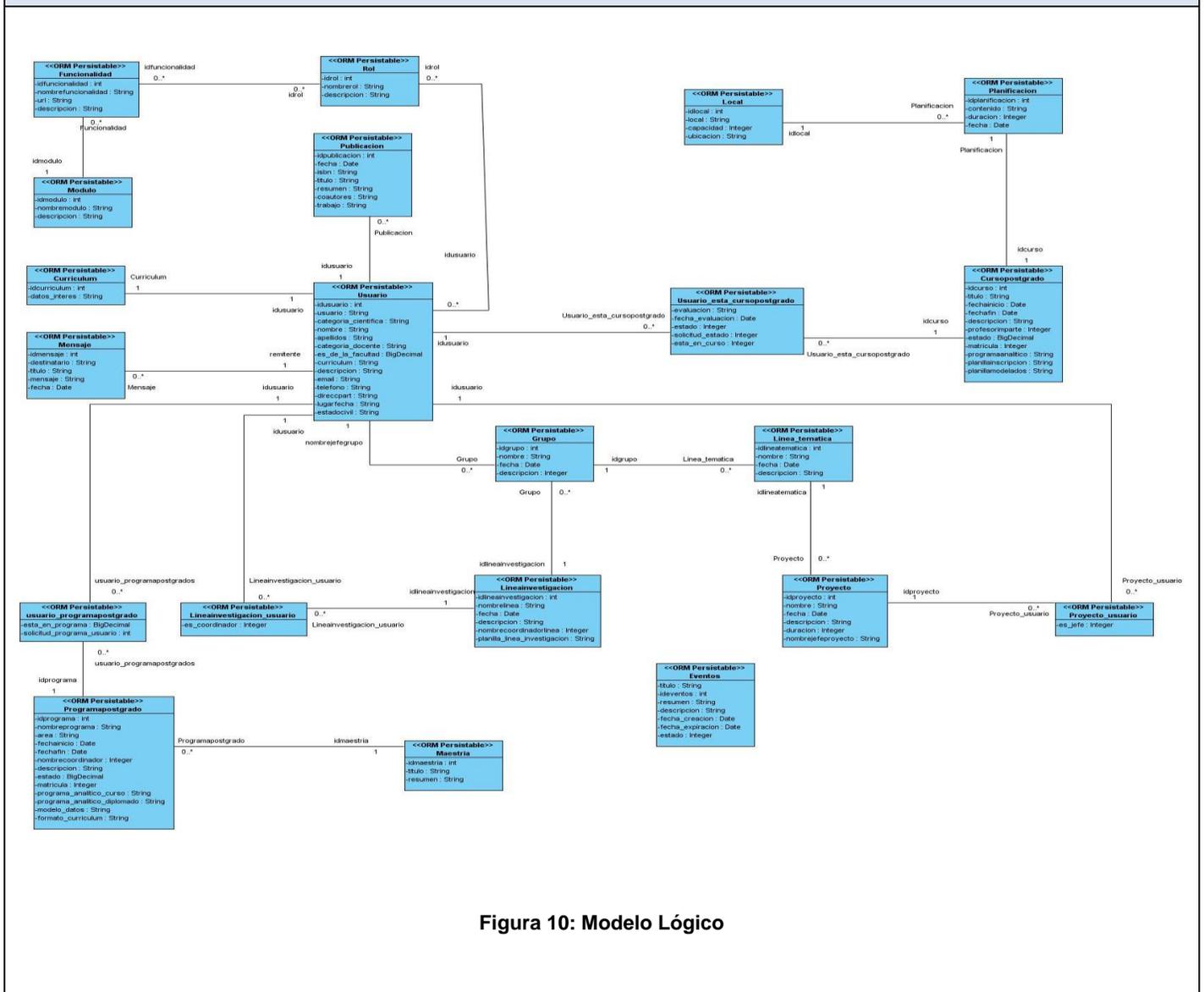


Figura 10: Modelo Lógico

3.3.2 Modelo Físico de Datos

Diagrama Entidad – Relación

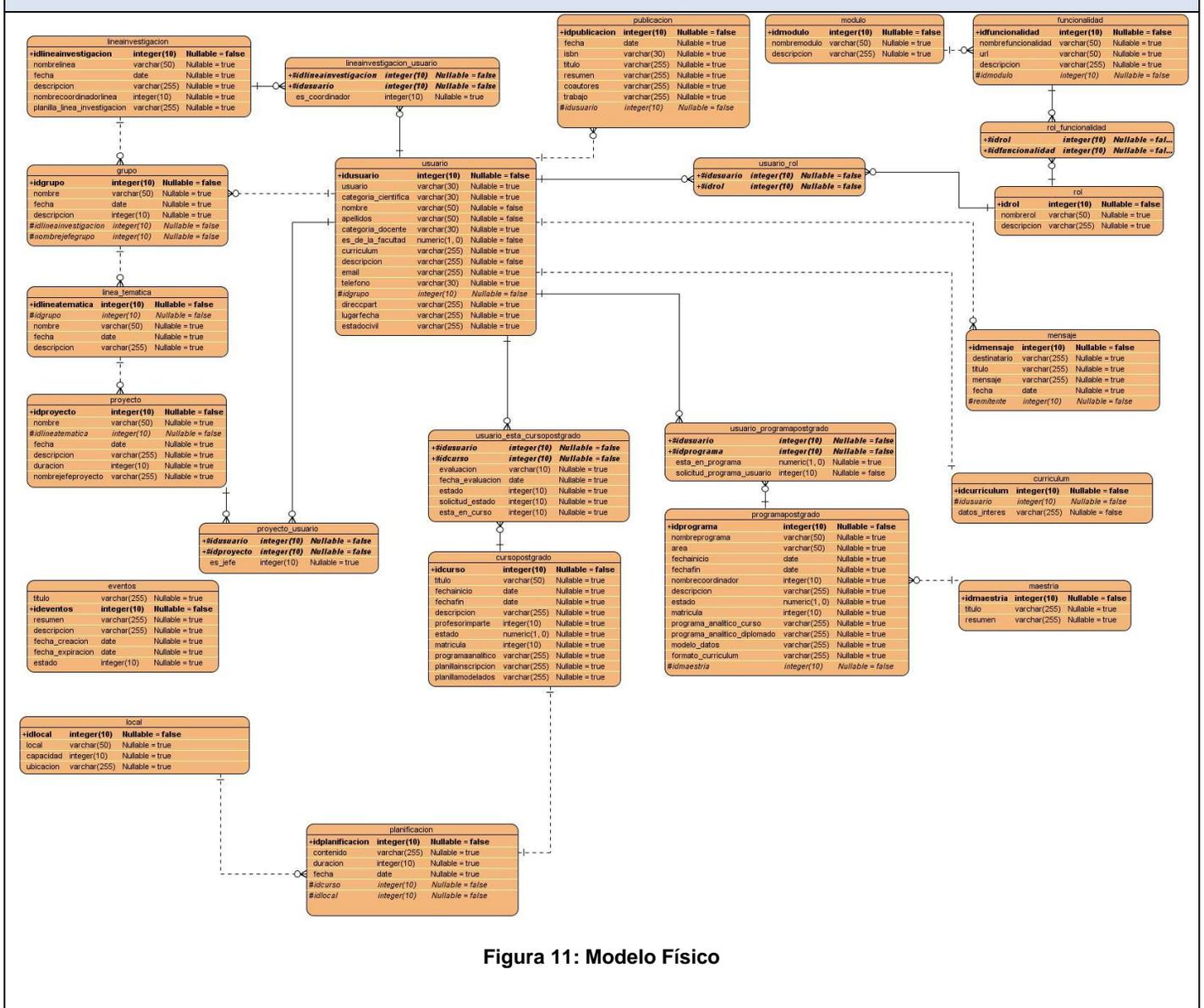


Figura 11: Modelo Físico



Anteriormente se presentó el modelo de datos físico y lógico de forma general. En los anexos se podrá observar de forma más detallada el modelo físico y lógico por cada módulo realizado en el sistema para un mejor entendimiento del mismo. (Ver anexo 12)

3.4 Arquitectura del Sistema

Una arquitectura no es más que un esqueleto o base de una aplicación. Es un diseño que muestra los bloques de construcción de una aplicación (o algún otro sistema de software). Indica la estructura, funcionamiento e interacción entre las partes del software.

Se refiere a un grupo de abstracciones y patrones que nos brindan un esquema de referencia útil para guiarnos en el desarrollo de software dentro de un sistema informático.

La Arquitectura de Software establece los fundamentos para que analistas, diseñadores, programadores, y otros miembros del equipo trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades.

3.4.1 Patrones

Los patrones ayudan a construir la experiencia colectiva de Ingeniería de Software. Son una abstracción de "problema – solución". Se ocupan de problemas recurrentes. Identifican y especifican abstracciones de niveles más altos que componentes o clases individuales y proporcionan vocabulario y entendimiento común.

3.4.1.1 Patrón propuesto

El patrón de diseño **Modelo Vista Controlador** o **MVC** describe una forma, muy utilizada en el Web, de organizar el código de una aplicación separando los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

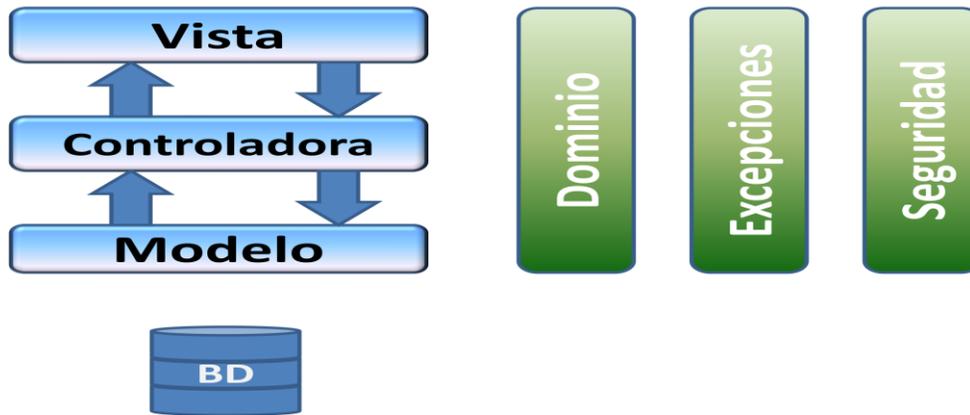


Figura 12: Patrón de Diseño MVC

El **Modelo** se encarga de todo lo que tiene que ver con la persistencia de datos. Guarda y recupera la información del medio persistente que utilizemos, ya sea una base de datos, ficheros de texto, XML, etc.

La **Vista** presenta la información obtenida con el modelo de manera que el usuario la pueda visualizar.

El **Controlador**, dependiendo de la acción solicitada por el usuario, es el que pide al modelo la información necesaria e invoca a la plantilla (de la vista) que corresponda para que la información sea presentada.

El **Dominio** va a ser las clases que retienen o mejor dicho encapsulan la información de una entidad en la base de datos, este interactúa en las tres capas principales (modelo, vista y controladora).

Las **Excepciones**, se encargan del tratamiento de los errores que ocurran, también está presente en las tres capas principales.

La **Seguridad** se concibe en cada una de las capas desde la vista hasta el modelo, y en cada una de estas con sus especificidades.

3.5 Conclusiones

En este capítulo se mostraron los diferentes diagramas del diseño del sistema, quedando conformada la arquitectura del mismo usando el patrón MVC, la cual se manifiesta en los distintos diagramas de clases del diseño, además se muestran algunos diagramas de secuencia, los modelos de datos, tanto físico como lógico, en los que se reflejan las relaciones y las distintas tablas de la base de datos que conforma el sistema.

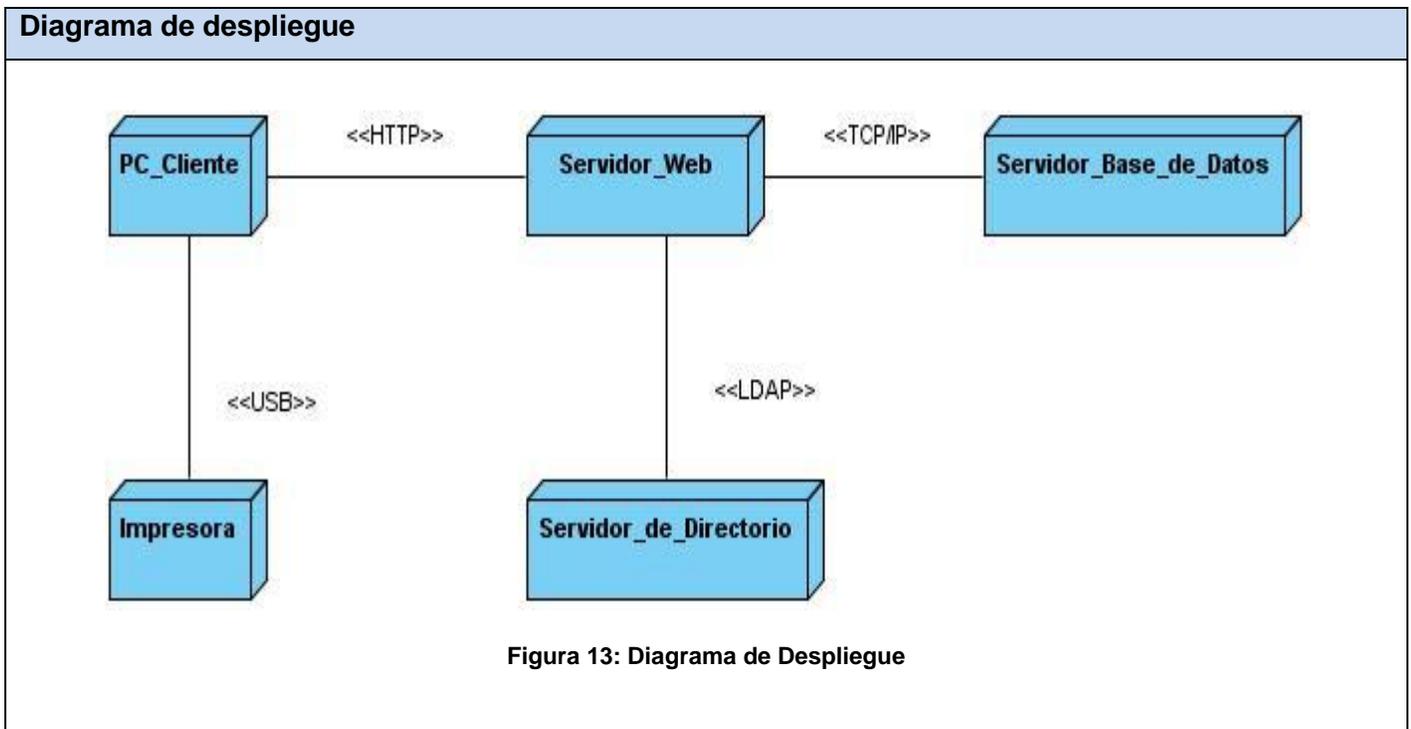
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción.

En el presente capítulo se muestra el diagrama de despliegue formado para efectuar el desarrollo del sistema SGI, se muestra de forma estructurada las conexiones que se establecen entre los servidores web y de base de datos y sus respectivos protocolos a través de los cuales están conectados. También se muestran los diagramas de componentes, todos realizados siguiendo una arquitectura modelo-vista-controlador.

4.2 Diagrama de despliegue

El modelo de despliegue, es usado para modelar la topología del hardware sobre el que se ejecuta el sistema y su distribución física, los distintos dispositivos y procesadores se representan a través de nodos. En este caso el sistema debe disponer de computadoras clientes, un servidor para aplicaciones Web y un servidor para Base de Datos.

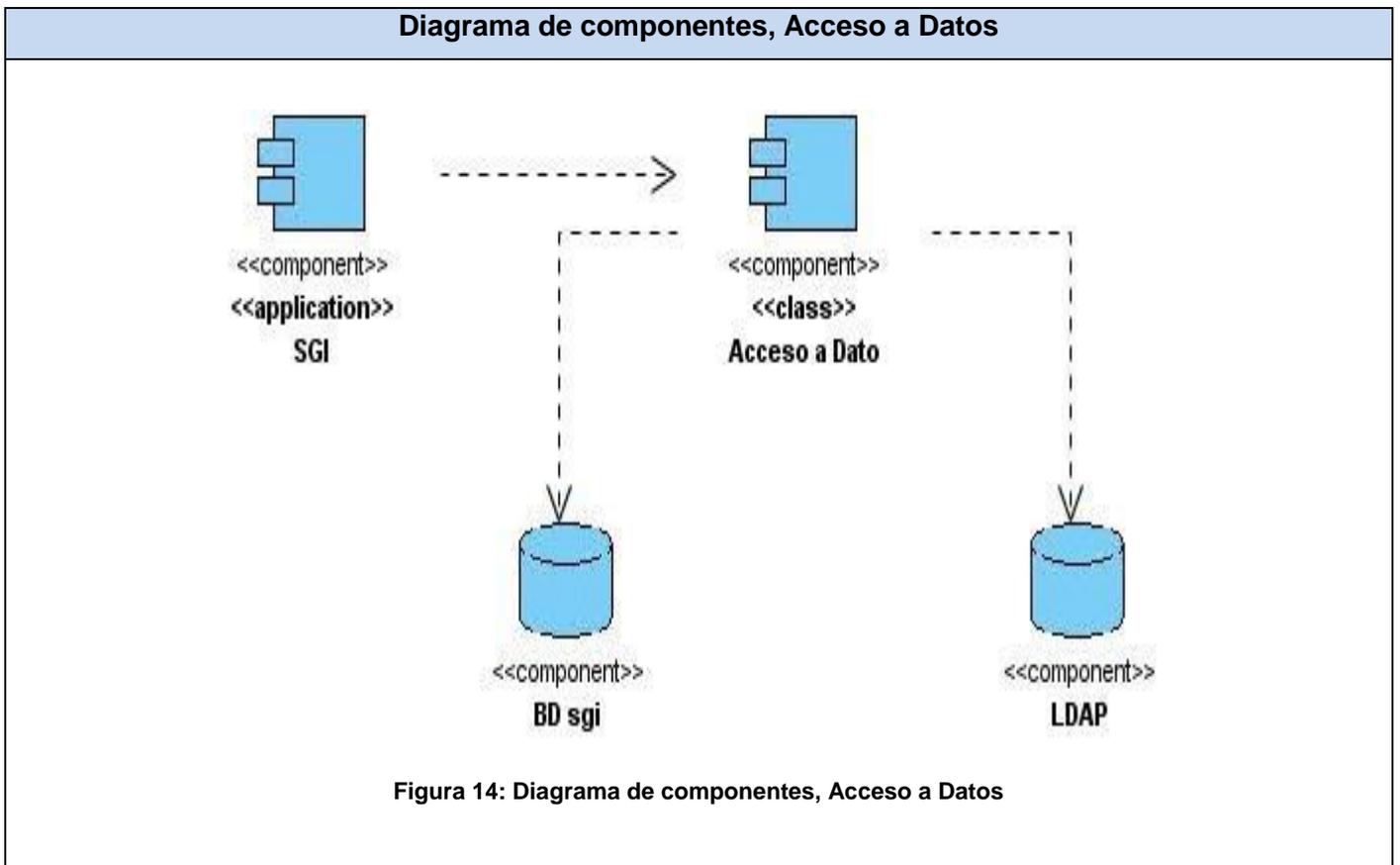


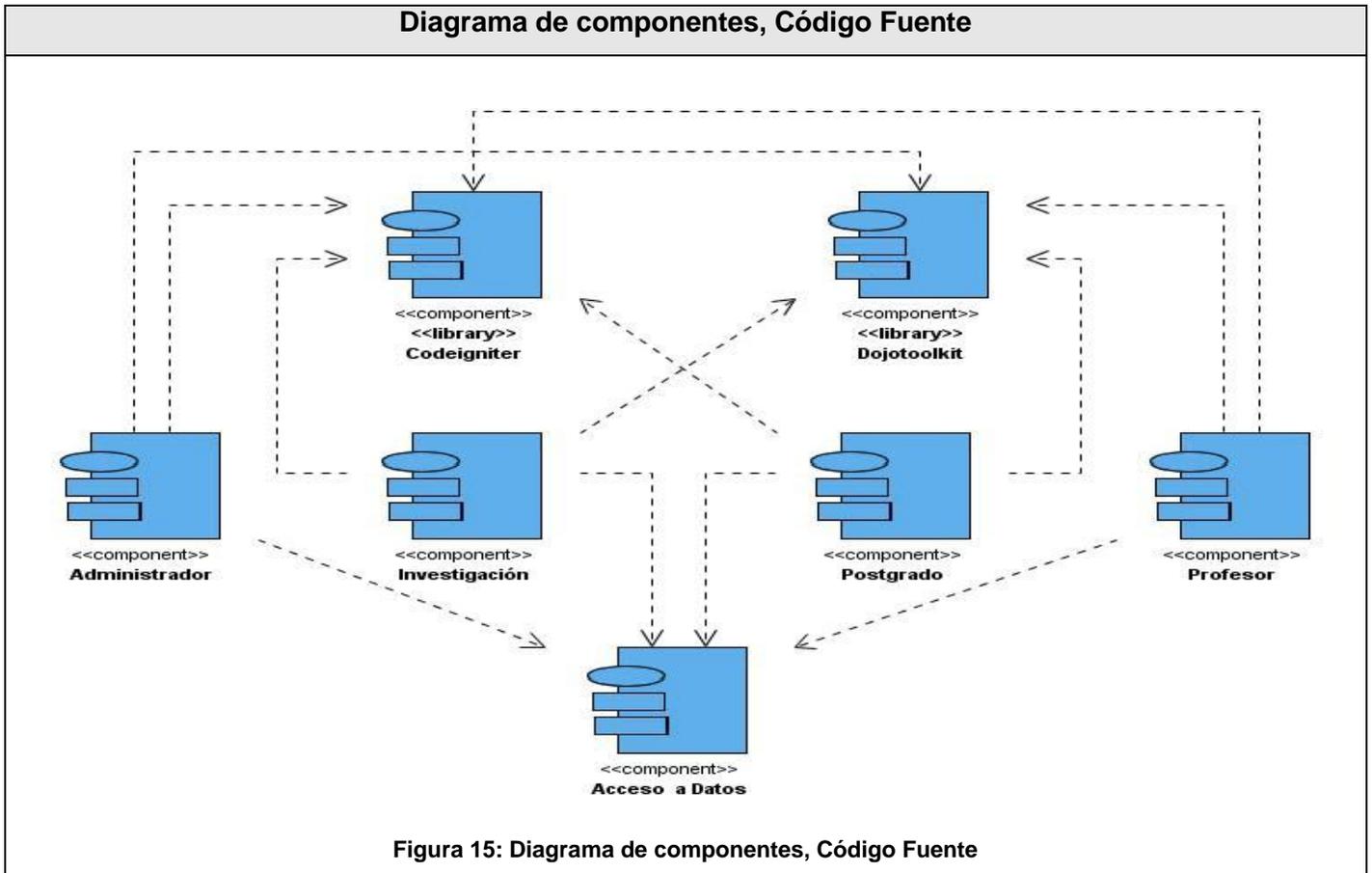
4.3 Diagrama de componentes

El diagrama de componentes se usa para representar a los elementos físicos o componentes y sus relaciones de dependencias. Ellos incluyen código fuente y ejecutables. Los componentes representan todo tipo de elementos de software como por ejemplo: las librerías, archivos, bibliotecas cargadas dinámicamente.

Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

A continuación se muestran el diagrama de componente de acceso a datos, código fuente y componentes web.





4.4 Pruebas

La prueba es un proceso de ejecución de un programa con la intención de descubrir errores. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces. Una prueba tiene éxito si descubre un error no detectado hasta el momento en que se les está realizando. Existen varios métodos de pruebas, en este sistema se utilizó el método de Caja negra y se realizaron Pruebas de Estrés.

Caso de prueba: Conjunto de entradas con datos de prueba, unas condiciones de ejecución, y unos resultados esperados.



Propósito: Identificar y comunicar las condiciones que se llevarán a cabo en la prueba. Los casos de la prueba son necesarios para verificar la aplicación exitosa y aceptable de los requisitos del producto (casos de uso).

Pruebas de caja negra: Pruebas que 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 (no se ve el código).

Prueba de estrés: Esta prueba se utiliza normalmente para evaluar la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que presente anomalías. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

Se realizaron validaciones al código HTML y CSS usando el sitio oficial de la World Wide Web Consortium (W3C).

Las pruebas de Caja Negra se le realizaron a los casos de usos más críticos. Después de haber concluido se obtuvo como resultado dos no conformidades de tipo corrección, a las que se les dio solución inmediatamente.

Con la realización de las pruebas de estrés a través de la herramienta JMeter se detectó que el sistema en situaciones normales soporta la conexión de 250 usuarios conectados concurrentemente. Para un número mayor de usuarios la aplicación comienza a presentar fallas en dar respuestas a las peticiones que se realizan, el tiempo mínimo de respuesta es de 78 milisegundo en situaciones normales aumenta a 547 milisegundo y el riesgo de presentar problemas es de 11.67 %.

Los diseños de casos de prueba se muestran en los anexos. (Ver Anexo 11).

4.5 Conclusiones

En este capítulo se mostró el diagrama de despliegue del sistema así como los diferentes diagramas de componentes, los cuales detallan las relaciones que se establecen entre las clases que contienen las implementaciones y validaciones, siguiendo la arquitectura del modelo-vista-controlador utilizada para desarrollar este sistema logrando un mejor entendimiento del funcionamiento basado ya en su código. Además se le hicieron pruebas al sistema las cuales fueron satisfactorias.



CAPÍTULO 5: FACTIBILIDAD DEL SISTEMA

5.1 Introducción

En este capítulo se realiza la estimación del esfuerzo a través del método de estimación por Puntos de Casos de Uso, que permite al culminar este análisis conocer si es conveniente llevar a cabo o no la realización del proyecto. Es de gran importancia para realizar un estudio previo del período de duración del proyecto que se va a desarrollar, así el equipo de trabajo puede organizar una buena planificación de las horas laborales diarias que deben efectuar para terminar el desarrollo del proyecto.

5.2 Método de estimación Puntos por Casos de Uso

La estimación mediante el análisis de Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de “pesos” a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. A continuación se detallan los pasos a seguir para la aplicación del método de estimación Puntos por Casos de Uso:

Cálculo de Puntos de Casos de Uso sin ajustar.

UUCP = UAW + UUCW

Donde:

UUCP: puntos de casos de uso sin ajustar.

UAW: factor de peso de los actores sin ajustar.

UUCW: factor de peso de los casos de uso sin ajustar.

Para calcular factor de peso de los actores sin ajustar UAW:

Tipo	Descripción	Peso	Cant * peso:
Simple	Otro sistema que interactúa con el SGI mediante una interfaz de programación. (API, Application Programming Interface).	1	0*1
Medio	Otro sistema que interactúa con el SGI mediante un protocolo o una interfaz basada en texto.	2	0*2



Complejo	Una persona que interactúa con el SGI mediante una interfaz gráfica.	3	4*3
Total:			12

Tabla 1: Cálculo del factor de peso de los actores sin ajustar.

Respuesta: El factor de peso de los actores sin ajustar es **UAW=12**.

Para calcular el factor de peso de los casos de uso sin ajustar **UUCW**:

No.	Nombre de Caso de Uso	Cantidad transacciones	Tipo
1	Gestionar eventos	8	Complejo
2	Gestionar espacios de publicación	5	Medio
3	Generar reportes sobre actividades de un profesor	4	Medio
4	Generar reporte de Balance General	3	Simple
5	Buscar integrantes por cursos	3	Simple
6	Gestionar notas	5	Medio
7	Exportar	3	Simple
8	Generar reportes avanzados	8	Complejo
9	Gestionar locales	9	Complejo
10	Subscribir a canal RSS	2	Simple
11	Gnerar Curriculum	8	Complejo
12	Gestionar canal de mensajería	11	Complejo
13	Planificar local	6	Medio

Tabla 2: Cantidad de transacciones por casos de uso.



Tipo	Descripción	Peso	Cant * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5	4*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10	4*10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15	5*15
Total:			135

Tabla 3: Cálculo del factor de peso de los casos de uso sin ajustar.

La cantidad de transacciones se determina a partir de la descripción textual de los casos de uso. Entre más detallada esté la descripción textual, más transacciones se pueden encontrar y la estimación será más exacta.

Respuesta: El factor de peso de los casos de uso sin ajustar es **UUCW= 135**

Luego:

$$\mathbf{UUCP = UAW + UUCW}$$

$$\mathbf{UUCP = 12 + 135}$$

$$\mathbf{UUCP = 147}$$

Cálculo de Puntos de Casos de Uso ajustados.

$$\mathbf{UCP = UUCP * TCF * EF}$$

Donde:

UCP: puntos de casos de uso ajustados.

UUCP: puntos de casos de uso sin ajustar.

TCF: factor de complejidad técnica.

EF: factor de ambiente.

Para calcular factor de complejidad técnica (**TCF**):

$$\mathbf{TCF = 0.6 + 0.01 * \sum (Peso_i * Valor_i)} \text{ (Donde } Valor \text{ es un número del 0 al 5).}$$

Significado de los valores:

0: No presente o sin influencia.

1: Influencia incidental o presencia incidental.

2: Influencia moderada o presencia moderada.



3: Influencia media o presencia media.

4: Influencia significativa o presencia significativa.

5: Fuerte influencia o fuerte presencia.

Factor	Descripción	Peso	Valor	Comentario	Σ (Peso _i * Valor _i)
T1	Sistema distribuido.	2	0	El sistema es distribuido.	0
T2	Objetivos de performance o tiempo de respuesta.	1	4	Se requiere que el sistema tenga un buen rendimiento y una respuesta rápida.	3
T3	Eficiencia del usuario final.	1	2	No hay restricciones de eficiencia.	2
T4	Procesamiento interno complejo.	1	3	No hay cálculos complejos.	1
T5	El código debe ser reutilizable.	1	4	Es reutilizable.	4
T6	Facilidad de instalación.	0.5	3	El sistema debe ser fácil de instalar.	1,5
T7	Facilidad de uso.	0.5	5	Debe ser un sistema amigable.	2,5
T8	Portabilidad.	2	0	Se requiere que el sistema sea portable.	0
T9	Facilidad de cambio.	1	5	Se requiere que sea un sistema flexible ante cambios.	4
T10	Concurrencia.	1	4	Si hay concurrencia.	4
T11	Incluye objetivos especiales de	1	5	El sistema gestiona información cuya	5



	seguridad.			confidencialidad es de carácter limitado.	
T12	Provee acceso directo a terceras partes.	1	3	Provee acceso directo a terceras partes.	2
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	2	No se requieren facilidades especiales de entrenamiento a los usuarios.	2
Total:					30

Tabla 4: Cálculo del Factor de Complejidad Técnica.

$$TCF = 0.6 + 0.01 * 30$$

$$TCF = 0.9$$

Para Calcular EF

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor}_i) \text{ (Donde Valor es un número del 0 al 5)}$$

Factor:	Descripción:	Peso:	Valor:	Comentario:	$\Sigma (\text{Peso}_i * \text{Valor}_i)$:
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	2	El grupo no está familiarizado con el modelo de proyecto.	3
E2	Experiencia en la aplicación.	0.5	3	Experiencia media en trabajo con ese tipo de aplicaciones.	1.5
E3	Experiencia en orientación a objetos.	1	4	La mayoría del grupo ha programado Orientado a Objetos.	4
E4	Capacidad del analista líder.	0.5	2	El analista es un estudiante con conocimientos adquiridos en clases.	1



E5	Motivación.	1	3	El equipo se siente motivado.	3
E6	Estabilidad de los requerimientos.	2	3	Se esperan cambios.	8
E7	Personal part-time.	-1	0	Todo el equipo es full-time.	0
E8	Dificultad del lenguaje de programación.	-1	3	Se usará el lenguaje PHP orientado a objetos, estudiado. No es difícil de aprender.	-3
Total:					17.5

Tabla 5: Cálculo del Factor Ambiente.

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso } i \times \text{Valor asignado } i)$$

$$EF = 1.4 - 0.03 * 17.5$$

$$EF = 0.875$$

Respuesta: El valor del factor de ambiente es 0.875

Luego:

$$UCP = UUCP * TCF * EF$$

$$UCP = 147 * 0.9 * 0.875$$

$$UCP = 115.7$$

Cálculo del Esfuerzo.

$$E = UCP * CF$$

Donde:

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: factor de conversión.

Para calcular Factor de Conversión (CF):

CF = 20 horas-hombre (si Total EF ≤ 2).



CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4).

CF = abandonar o cambiar proyecto (si Total EF \geq 5).

Total EF = Cant. EF < 3 (entre E1 – E6) + Cant. EF > 3 (entre E7 – E8)

Como:

Total EF = 2 + 0

Total EF = 2.

CF = 20 horas-hombre (porque Total EF = 2).

Luego:

E = 115.7 * 20 horas-hombre

E = 2315.25 horas-hombre.

E= 2315

Distribución del Esfuerzo entre las diferentes actividades del módulo.

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10	578.7
Diseño	20	1157.4
Implementación	40	2315
Prueba	15	868
Sobrecarga	15	868
Total	100	5787

Tabla 6: Distribución del esfuerzo estimado entre los flujos de trabajo de RUP.

Beneficios tangibles e intangibles.

El desarrollo del Sistema para la Gestión de la Información asociadas a las Investigaciones y Postgrado de la Facultad 2 de la Universidad de las Ciencias Informáticas aporta para esta última un considerable beneficio económico. El mayor aporte se enfoca en el control de toda la información relacionada a la investigación y el postgrado en la facultad, facilitando así el trabajo al vicedecano de investigación y postgrado y teniendo un mejor control de toda esta documentación.



5.3 Análisis de costos y beneficios.

Salario medio de la fuerza de trabajo: \$100.

Cantidad de trabajadores: 2.

Trabajadores que dedican 12 horas diarias a trabajar en el proyecto: 2

Como se tiene 5787 horas-hombre según lo estimado, se divide este valor entre 12 horas (media de la cantidad de horas que se trabaja en el proyecto entre los 2 trabajadores) para calcular la cantidad de días de trabajo-hombre que se necesitan para realizar el módulo. Al realizar esta operación matemática se obtiene que el módulo de planificación tenga 482.25 días de trabajo-hombre. Este valor se divide entre la cantidad de trabajadores para obtener la cantidad de días de trabajo que tiene cada trabajador para el desarrollo del módulo. De este resultado se concluye que con 2 trabajadores que trabajan 12 horas diarias (media de la cantidad de horas que se trabaja en el proyecto entre los 2 trabajadores), el módulo se terminaría en 241.125 días. Por tanto, finalmente para realizar el módulo se necesitan aproximadamente 241 días (8 meses). El costo final del módulo es:

Costo total del módulo = salario medio * cantidad de trabajadores * duración del proyecto (meses)

Costo total del módulo = \$ 100 * 2 * 8 = \$ 1600.

La creación de software en el mundo tiene un costo muy elevado por los altos precios que implantan las grandes compañías en este sentido. Sistemas similares al que se presenta son codiciados por los usuarios y su costo tiende a ser elevado. Las herramientas que se utilizan en la elaboración del sistema son totalmente libres, no es necesario el pago de licencias, por tanto, no se incurre en gasto alguno en la utilización de las mismas.

Por lo anteriormente planteado, se concluye que la elaboración del Sistema para la Gestión de la Información asociada a las Investigaciones y Postgrado en la Facultad 2, es factible.

5.4 Conclusiones

En este capítulo se realizó un estudio de la factibilidad del sistema que se presenta. Se analizaron los beneficios tangibles e intangibles que proporciona la implantación del mismo en la Universidad de las Ciencias Informáticas, así como, los costos que presuponen herramientas similares con respecto a la que se presenta y la factibilidad de ésta para su implantación en la Universidad. Con la realización de este análisis de costo podemos concluir que el sistema desarrollado es factible.



CONCLUSIONES GENERALES

La investigación desarrollada y los resultados obtenidos permiten a los autores plantear las siguientes conclusiones:

- El análisis de las características de varios sistemas relacionados con la gestión de la información asociada a los procesos de investigación y postgrado, permitió desarrollar una nueva versión del Sistema para la Gestión de la Información asociada a las Investigaciones y Postgrado de la Facultad 2, que se adecua a las necesidades de profesores y asesores de las áreas mencionadas.
- Se desarrollaron artefactos que describen los procesos de ingeniería de software mediante la aplicación de la metodología RUP.
- Se obtuvo una aplicación web a la que se realizaron pruebas de funcionalidad, estrés y rendimiento, cuyos resultados fueron satisfactorios.
- Se utilizaron tecnologías libres para el desarrollo de la aplicación.
- Se realizó un estudio del proyecto para analizar su posible uso en la Facultad 2, con el que se demostró la factibilidad en la implantación del sistema.

Por lo antes expuesto, se considera como principal aporte de este trabajo de diploma el desarrollo de la nueva versión del Sistema para la Gestión de la Información asociada a las Investigaciones y Postgrado. Con la misma se proporciona una mayor organización, centralización y acceso a la información relacionada con las áreas mencionadas, lo que constituye una fortaleza estratégica de la investigación y postgrado dentro de la Facultad 2.



RECOMENDACIONES

Después haber culminado el presente trabajo y ver su resultado se proponen las siguientes recomendaciones:

1. Validar el sistema y que este sea puesto en funcionamiento lo antes posible por parte de los interesados.
2. Agregarle nuevas funcionalidades para mejorar la gestión de la información en el mismo, así como aumentar su valor agregado.
3. Aplicar la solución propuesta al resto de la Universidad de las Ciencias informáticas.
4. Realizar un estudio para adicionarle el módulo de Postgrado al sistema ERP que se desarrolla en la Universidad de las Ciencias Informáticas.



REFERENCIA BIBLIOGRÁFICA

1. Scielo . *Información Tecnológica*. [En línea] [Citado el: 28 de Enero de 2011.] Disponible en:[http://www.scielo.cl/scielo.php?pid=S0718-07642007000100014&script=sci_arttext].
2. Universidad Pedagógica Nacional . *Sistema de Gestión de Investigación en línea PGIL*. [En línea] [Citado el: 30 de Enero de 2011.] Disponible en:[<http://www.pedagogica.edu.co/portal/contenido.php?esquema=894>].
3. ExtremeProgramming. *A gentle introduction*. [En línea] [Citado el: 1 de Febrero de 2011.] Disponible en: [<http://www.extremeprogramming.org>.] .
4. **Alfonso Navarro Albey, Coroa Castro Julio José**. “*Sistema de Gestión de Investigación de la Facultad 2*”. junio 2010. Pág. 11.
5. Boost Productivity with Innovative and Intuitive Technologies. *Visual Paradim como herramienta case*. [En línea] 2 de Abril de 2007. [Citado el: 26 de Noviembre de 2010.] Disponible en: [<http://www.visual-paradigm.com/>].
6. CAVSI. *Sistema gestor de base de datos*. [En línea] [Citado el: 8 de Febrero de 2011.] Disponible en: [<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>].
7. PostgreSQL. [En línea] [Citado el: 4 de Febrero de 2011.] Disponible en:[<http://www.postgresql.org/docs/8.3/static/tutorial-advanced.html>].
8. The PHP Company. *Zend Studio*. [En línea] [Citado el: 16 de Febrero de 2011.] Disponible en: [<http://www.zend.com/en/products/studio/>].
9. **Alfonso Navarro Albey, Coroa Castro Julio José**. “*Sistema de Gestión de Investigación de la Facultad 2*”. junio 2010. Pág. 16.
10. The Apache Software Foundation. *Apache 2.0*. [En línea] 13 de Abril de 2010. [Citado el: 2011 de febrero de 4.] Disponible en:[http://httpd.apache.org/docs/2.0/es/new_features_2_0.html].
11. Lenguajes de programación. *¿Qué es un lenguaje de programación?* [En línea] [Citado el: 8 de febrero de 2011.] Disponible en:[<http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>].
12. MasterMagazine. *Definición de Cliente/Servidor*. [En línea] [Citado el: 8 de febrero de 2011.] Disponible en: [<http://www.mastermagazine.info/termino/4294.php>].



13. Desarrolloweb.com. *Lenguajes del lado servidor o cliente*. [En línea] [Citado el: 8 de febrero de 2011.] Disponible en: [<http://www.desarrolloweb.com/articulos/239.php>].
14. Programación en castellano. *Ventajas del PHP*. [En línea] [Citado el: 12 de febrero de 2011.] Disponible en: [http://www.programacion.com/articulo/por_que_elegir_php_143#porquephp_ventajas].
15. Framework. *What is Framework?* [En línea] [Citado el: 18 de Febrero de 2011.] Disponible en: [<http://www.framework.com/>].
16. **Alfonso Navarro Albey, Coroa Castro Julio José**. “*Sistema de Gestión de Investigación de la Facultad 2*”. junio 2010. Pág. 22.
17. Dojo toolkit. *Documentation*. [En línea] [Citado el: 18 de febrero de 2011.] Disponible en: [<http://dojotoolkit.org/documentation/>].
18. CodeIgniter. [En línea] [Citado el: 16 de febrero de 2011.] Disponible en:[<http://codeigniter.com/>].



BIBLIOGRAFÍA

1. [aut. libro] Craig Larman. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. s.l. : Prentice Hall.
2. **Rumbaugh, James.** *El Lenguaje Unificado de Modelado*.
3. **Jacobson, Ivar.** *El Proceso Unificado de Desarrollo de Software*.
4. **Dueñas, José.** Joé Dueñas Desarrollador Web. *CodeIgniter, un gran framework para PHP*. [En línea] [Citado el: 8 de marzo de 2011.] Disponible en: [<http://www.joseduenas.com/archivos/codeigniter-un-gran-framework-para-php>].
5. Cristalab. *Introducción al framework CodeIgniter*. [En línea] 9 de diciembre de 2008. [Citado el: 11 de marzo de 2011.] Disponible en: [<http://www.cristalab.com/tutoriales/introduccion-al-framework-codeigniter-c65436/>].
6. W3C. *Markup Validation Service*. [En línea]. [Citado el: 11 de mayo de 2011.] Disponible en: [<http://validator.w3.org/>].



GLOSARIO DE TÉRMINOS

- **Framework:** Se definen como un diseño reutilizable de todo o parte de un sistema, representado por un conjunto de componentes abstractos, y la forma en la que dichos componentes interactúan.
- **CASE:** Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software.
- **Caso de Uso:** Es una descripción de un conjunto de secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

