

Universidad de las Ciencias Informáticas

Facultad 5



**Sistema para el Departamento de Investigación de
Matemática y Física.**

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

Autores: Reinier Chávez La Rosa.
Misael Crespo Padrón

Tutor: Ing. Irina Cancela Nieto.

Co-tutor: Alison Muñoz Capote

Ciudad de la Habana

Julio del 2007

Año 49 de la Revolución

*“No basta con tener buen ingenio lo principal es aplicarlo bien.”
René Descartes*

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los ____ días del mes de julio del año 2007.

Reinier Chávez La Rosa

Misael Crespo Padrón

Ing. Irina Cancela Nieto



AGRADECIMIENTOS

Reinier

A mis padres por ser mi fuente de inspiración, por el apoyo y la confianza que depositan en mí, a ellos debo todo lo que soy y seré en la vida.

A papi (Miguel) por sus sabios consejos y preocupación constante por mi superación, para el todo mi cariño y respeto.

A mi hermano (Diosvany) y su esposa (Dunia) por siempre tenerme presente.

A mi primo (Andrés) por darme todo su apoyo en los momentos más difíciles.

A tía Susa, Adelfa, Yory y a Mary por siempre estar presentes.

A todos mis primos y amigos por hacerme pasar tan buenos momentos.

A Mairita y Ramoncito por soportarme y darme todo su apoyo a lo largo de mi carrera, para ellos todo mi cariño.

A mi gran amigo Johanny por su ejemplo y apoyo constante.

Misael

A mis padres pues usaron su ejemplo para educarme de forma íntegra, por haber velado celosamente por mi futuro, por ser más mis amigos que mis padres, de ustedes estoy orgulloso y les debo todo lo que soy.

A mi hermano por ser siempre el mejor, por enseñarme a no permitir la derrota, por haber sido mi héroe de niño, por comprometerme en seguir su camino (una meta bien grande).

A mi novia por haberse mantenido por casi 6 años soportándome, por haberme hecho tan feliz y tan estable, por haber vivido cada momento de mi vida como si fuera de ella, por haberme tenido tanta fé.

A todos mis compañeros poder contar con ellos siempre.

Reinier Chávez y Misael Crespo

A nuestra tutora (Irina Cancela) por todo su apoyo y dedicación.

A Yunier Velázquez y Dania Souchay por su opinión profesional.

Al vicedecano (Reinier Portelles) por siempre estar dispuesto a brindar su apoyo.

A todos los compañeros que hicieron más amena nuestra estancia en la universidad y en especial al Orlo, Pedry, para Dariel por su ayuda en cuanto al diseño del sistema.

“A mis padres por su cariño y dedicación”

Reinier Chávez.

“A los que considero mi única familia”

Misael Crespo

RESUMEN

Hoy en día con el innegable avance tecnológico de la sociedad y el desarrollo cada vez más elevado de la informática y las comunicaciones no es inocuo privarse de las bondades de la era moderna, es menester del intelecto humano buscar alternativas que mejoren su calidad de trabajo, lo racionalicen y reduzcan sus costos tomando como alternativa viable la automatización de los procesos.

El desarrollo de un Sistema Informático que automatice el trabajo del Departamento de Investigaciones de Matemática y Física Computacional surge con la necesidad de centralizar y hacer accesible desde cualquier lugar de la Universidad toda la información que se manipula en el grupo. Debido a que los profesores de la especialidad de Matemática y Física poseen poco conocimiento de las investigaciones realizadas en el centro, por la inexistencia de un sistema que los agrupe y sirva de apoyo al estudio y superación de los interesados en el tema.

Este trabajo tiene como objetivos fundamentales la publicación de noticias relevantes, relacionadas con temas investigativos de esta ciencia en el ámbito nacional, internacional ó dentro de la propia Universidad, además de darle promoción a concursos, talleres, eventos realizados por el grupo, así como agrupar a todos sus miembros en una comunidad que los identifique a través de la Web, elevando de esta forma el nivel científico-investigativo en el Dpto. de matemática y física.

El principal resultado de esta investigación es la elaboración de la primera versión de un sistema que cumpla las exigencias del grupo del investigativo.

PALABRAS CLAVES

Sistema, automatizar, gestión, información, centralizar, investigación.

Tabla de Contenidos

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1. Antecedentes.....	4
1.2. Conceptos Fundamentales.....	5
1.2.1. ¿Qué es Gestión?.....	5
1.2.2. La Gestión Documental.....	7
1.2.3. ¿Qué es la Investigación Científica?.....	8
1.3. Tendencias y tecnologías actuales.....	9
1.3.1. Desarrollo de Aplicaciones Web.....	9
1.3.2. Política de migración a software libre.....	11
1.3.3. Principales lenguajes de desarrollo Web en la actualidad.....	13
1.3.4. ¿Por qué utilizar PHP y no otro?.....	22
1.3.5. ¿Por qué MySQL?.....	23
1.3.6. ¿Por qué usar DreamWeaver como ID principal en el diseño?.....	23
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	25
2.1. Modelo del Dominio.....	25
2.2. Reglas del negocio.....	27
2.3. Requerimientos funcionales.....	27
2.4. Requerimientos no funcionales.....	28
2.5. Actores del sistema a automatizar.....	30
2.6. Descripción de los casos de uso.....	33
2.7. Casos de Uso Expandidos.....	38
CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	63
3.1. Modelo de Análisis.....	63
3.2. Diseño.....	63
3.2.1. Diagrama de clases persistente.....	63
3.2.2. Diagrama de Clases Web.....	64
3.2.3. Diagramas de Interacción.....	65
3.2.4. Descripción de las clases de diseño.....	65
3.2.5. Diseño de la base de datos.....	79
3.2.6. Descripción de las tablas de la base de datos.....	79
3.2.7. Principios de diseños.....	80
3.2.8. Uso de capa de abstracción a datos.....	81
3.2.9. Patrón de Arquitectura en Capas.....	81
3.2.10. Patrones para asignar responsabilidades.....	82
3.3. Modelo de despliegue.....	83
CONCLUSIONES GENERALES.....	85
RECOMENDACIONES.....	86
BIBLIOGRAFÍA.....	87
ANEXOS.....	88
GLOSARIO DE TÉRMINOS.....	122

INTRODUCCIÓN.

Las tendencias al uso de las nuevas tecnologías de la información han tomado un gran auge en los últimos años por todo lo novedoso y atractivo que aportan a la hora de brindar información de forma rápida y segura a través de la Web. Innumerables son los sistemas para la gestión de la información que existen hoy en día a nivel mundial en las distintas esferas de la sociedad.

La Universidad de las Ciencias Informáticas (UCI) rectora del desarrollo informático en Cuba no está ajena a este desarrollo. En la actualidad cuenta con una gran cantidad de Sistemas Web que tienen como objetivo garantizar el acceso a la información digital disponible, organizar todo el trabajo docente y facilitar el estudio a tiempo completo a través de la red a los estudiantes y profesores. Sin embargo, en el centro existen entidades que abordan temas específicos que no son conocidos por la mayoría de los miembros de la universidad pues no cuentan con un sistema que invite a los interesados a sentirse identificados y aportar ideas que estimulen el avance y logren un mayor desarrollo de dicha entidad. El ejemplo que se trata es el Departamento de Investigaciones de Matemática y Física Computacional, el cual pudiera agrupar a los profesores de la especialidad de Matemática y Física permitiéndoles acceder a trabajos e investigaciones realizadas que sirvan para apoyar su preparación o superación.

Dada la importancia que tiene todo el proceso investigativo en todas las esferas de la sociedad, este trabajo de tesis asume la responsabilidad de automatizar los procesos del Departamento de Investigaciones de Matemática y Física de la UCI y formar una comunidad donde se agrupen todos los profesores de estas especialidades interesados en las distintas líneas de investigación que ofrece ese grupo, siendo este un buen espacio para fomentar la cultura y el conocimiento adquirido en la Universidad en materia de investigaciones. Además de centralizar toda la información concerniente a ese departamento, darle publicidad a los distintos talleres, eventos científicos que se desarrollarán y permitir el acceso a estudiantes y profesores interesados en el tema desde cualquier lugar del campo universitario a las publicaciones realizadas por miembros de este grupo investigativo.

Se parte primeramente de que la **situación problemática** está dada porque actualmente en el departamento de investigación de Matemática y Física Computacional de la Universidad de las Ciencias Informáticas (UCI) la divulgación de talleres, eventos y conferencias se hacen utilizando medios como la intranet, correos electrónicos y carteles, los usuarios carecen de facilidades de acceso a trabajos ya

realizados, lo que impide el avance del grupo investigativo y existe muy poca integración de todos los profesores que pertenecen a dichas ciencias donde ¿Cómo gestionar las actividades del grupo de investigación, el acceso a trabajos realizados, así como la integración de todos los profesores del grupo de investigación de Matemática y Física Computacional mediante un sistema informático? constituye el **problema científico** de esta investigación.

Se define como **objeto de estudio** el proceso de gestión de la información de las investigaciones en la Universidad de las Ciencias Informáticas. Donde el **Campo de acción** estaría delimitado por el proceso de gestión de la información de las investigaciones del departamento de Matemática y Física Computacional de la Universidad de las Ciencias Informáticas.

Se **defiende la idea** de que mediante una aplicación Web para la gestión de toda la información referente al grupo de investigaciones de Matemática y Física Computacional, se podrá dar soporte a todos los usuarios interesados en los trabajos llevados a cabo en las diferentes líneas investigativas, integrar a los profesores de Matemática y Física de las distintas facultades en sus respectivos perfiles, así como la promoción de talleres, eventos y conferencias.

El **objetivo general** es modelar un sistema informático que gestione toda la información referente al grupo de investigación de Matemática y Física Computacional y garantizar el acceso a trabajos investigativos realizados por miembros de esta comunidad.

Planteándose como **Objetivos Específicos** los siguientes puntos:

- Mejorar el proceso de gestión de la información del grupo investigativo.
- Elevar el nivel científico-investigativo en el Dpto. de matemática y física.
- Facilitar el flujo de información entre los miembros del grupo investigativo y usuarios externos.
- Crear un espacio que brinde información actualizada de los últimos acontecimientos científicos en la comunidad.

- Automatizar los procesos que se realizan en el departamento de investigaciones.

Desarrollar una herramienta que permita el acceso a las investigaciones realizadas en las distintas líneas investigativas de este grupo y sirva de guía a los usuarios que tengan cierta inclinación por este tema.

Para dar cumplimiento a los objetivos se han trazado las siguientes **tareas de Investigación**.

- Hacer un estudio preliminar de la situación actual de los grupos investigativos de la universidad.
- Entrevistar al jefe y a los miembros del grupo investigación para recoger toda la información referente al grupo de matemática computacional.
- Obtener los requerimientos del sistema.
- Realizar un análisis de los lenguajes de programación y las herramientas existentes.
- Hacer un análisis y discutir la arquitectura de la aplicación.
- Obtener el modelo de datos más idóneo que permita tener un sistema parametrizable.
- Comprobar las funcionalidades del sistema mediante las pruebas de caja negra.

El presente documento se estructura en resumen, introducción, tres capítulos de contenidos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos donde se incluye todo lo relacionado con el trabajo investigativo realizado.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Introducción

El contenido de este capítulo constituye la base teórica del presente trabajo. En él se describen los principales conceptos relacionados con la modelación del proyecto así como, explicar y describir las tendencias, herramientas y tecnologías actuales con las que se puede dar solución a este problema, abordando también sobre los antecedentes que él brinda en los diferentes niveles escalonados (mundo, país, universidad) y culminando con la selección de la alternativa más flexible para la construcción del sistema.

1.1. Antecedentes

Al presente trabajo se le anteceden disímiles sistemas informáticos por la importancia que le otorgan los organismos a nivel mundial a la automatización de sus procesos la publicación y promoción de sus intereses por el aumento considerable de la calidad de trabajo, la reducción de costos y de tiempo de procesamiento de la información.

Es innegable la relación histórica existente entre el sistema científico-tecnológico y la sociedad en su conjunto, dado que la actividad científica y tecnológica es concebida como estrategia social para la superación de la pobreza y del atraso social. Bajo esta noción, el carácter sustantivo del desarrollo científico-tecnológico para el avance social, debe orientar toda su intención a la construcción de políticas fomentadoras de la ciencia, la investigación y la innovación.

Este trabajo sólo conformará un eslabón más de la gran cadena que está estrechamente unida en el mundo en cuanto a la gestión de información a través de las redes informáticas dándole notables ventajas a los usuarios de desarrollar su intelecto mediante el buen uso de las tecnologías presentes hoy en la era moderna.

Hasta el momento han sido implementados muchos sistemas para gestión de la información referente a actividades investigativas que de una forma u otra responden a las exigencias, cada día más crecientes, de responsabilidad, calidad y rapidez en todo proceso de gestión utilizando fundamentalmente las tecnologías Web.

Es válido destacar que en los países de América Latina, La Universidad constituye el principal camino para el fortalecimiento de las estructuras nacionales de producción científica, ya que en ellas se concentran en gran medida los resultados alcanzados en materia de investigación y desarrollo de conocimientos científicos en el área de las ciencias, así como la infraestructura necesaria y el personal calificado para llevar adelante dicha tarea.

Cada país del continente tiene su propia visión sobre la gestión de las investigaciones y aunque los resultados alcanzados se desconocen a grandes rasgos, debido a que no se dispone de los datos necesarios para evaluar la situación actual, no se puede dejar de reconocer países como Argentina, Brasil, Chile, Colombia, México y Venezuela, que promueven y estimulan el desarrollo de investigaciones a escala nacional.(ROYERO 2002) ,brindando espacios donde usuarios y expertos pueden intercambiar experiencias, dar sugerencias, criterios, opiniones, garantizando una mayor cultura y retroalimentación en materia de conocimiento acerca de las investigaciones en ciencias tan importantes como lo son la Matemática y la Física, además de impartir talleres, conferencias online relacionadas con el tema.

En Cuba y específicamente en la Universidad de las Ciencias Informáticas gracias al gran avance tecnológico existente y las bondades que proporciona la Universidad, se han desarrollado varias herramientas como (Sitios Web, Portales y Plataformas de desarrollo) disponibles a tiempo completo que cumplen con algunos de los requisitos planteados en materia de gestión y divulgación de información pertenecientes a diferentes sectores de la UCI. Pero específicamente el tema de las investigaciones en el área de las matemáticas y la física no está en su estado óptimo en cuanto a organización, divulgación y acceso a trabajos científicos de estas ciencias, aunque ya se viene trabajando en este sentido, se prevé desarrollar una aplicación que centralice toda la información perteneciente al Departamento de Investigaciones de Matemática y Física Computacional así como otras actividades de vital importancia en el mismo.

1.2. Conceptos Fundamentales.

1.2.1. ¿Qué es Gestión?

La gestión no es más que la conducción o manejo de los recursos de un sistema con el fin de lograr los objetivos del mismo. Gestionar es entonces alcanzar propósitos a través de la acción coordinada de personas.

Mejorar la gestión significará aumentar progresivamente la efectividad, equidad y eficiencia de las acciones. La gestión se hace evidente mediante indicadores que reflejan el avance hacia los propósitos establecidos.

La existencia de gestión sugiere procesos de planeación, ejecución, evaluación y ajuste, siempre en constante mejoramiento. Incluye además voluntad y apropiación de las personas involucradas en buscar nuevas formas de organizar el trabajo en procura de mejoras eficaces (alcanzar propósitos) y eficientes (alcanzar propósitos, con la menor cantidad de recursos, incluido el tiempo). (ACADEMIA 2006)

Existen varias acepciones para el término gestión, el diccionario Larousse la define como una palabra del latín *gestio onis*. Acción y efecto de administrar.

En el libro “Indicadores de Gestión”, publicado en 1999, Pág. 24, se define genéricamente la gestión como: “conjunto de decisiones y acciones que llevan al logro de objetivos previamente establecidos”.

Es claro que la gestión trasciende a la acción por sí misma, ya que incluye la formulación de objetivos, la selección, evaluación y determinación de estrategias, el diseño de los planes de acción, la ejecución y el control de los mismos.

En las empresas u organizaciones es común hablar de tres niveles de gestión:

- La gestión estratégica
- La gestión táctica
- La gestión operativa.

Se entiende por gestión **estratégica** el conjunto de decisiones y acciones que llevan a la organización a alcanzar los objetivos corporativos. Tiene directa relación con la formulación, ejecución y control del plan estratégico de la empresa y se basa en la comprensión y administración de la relación e interacción de la empresa con el medio ambiente, es decir con los proveedores y los clientes por una parte y por otra, con los demás agentes presentes en el entorno tales como la competencia, el gobierno, y en general todos aquellos que constituyen la cadena del valor a la que pertenece la organización.

De la gestión estratégica se deriva la gestión **táctica**, ya que los objetivos de ésta se desprenden directamente de la primera gestión y deben estar alineados con ellos. La gestión táctica involucra el ámbito interno de la organización y obedece al óptimo desarrollo de todas sus actividades internas.

Finalmente, se tiene la gestión **operativa**, que no por ser de alcance más limitado es menos importante que la gestión estratégica y la gestión táctica. Los objetivos operativos se derivan directamente de los objetivos tácticos. El nivel operativo involucra cada una de las actividades de la cadena del valor interna, tanto primarias como de apoyo, por tanto es posible hablar de gestión de aprovisionamiento, gestión de la producción, gestión de distribución, gestión de marketing, gestión de servicio, gestión de personal así como la gestión financiera. (JARAMILLO 2001)

1.2.2. La Gestión Documental.

En el contexto actual en que se desenvuelven las diferentes instituciones y organizaciones exige la aplicación de determinados sistemas, métodos, procedimientos y otros instrumentos que respondan a esas expectativas en el área de la gestión de información y documentación.

La gestión documental es el centro de la gestión de información y no es más que un conjunto de actividades específicas que permite coordinar y controlar de forma sistemática, todos aquellos aspectos relacionados con la creación, recepción, organización, almacenamiento, preservación, acceso y difusión de los documentos en una organización.

Los primeros pasos:

Desde la aparición del primer sistema operativo en un ordenador de escritorio en los años ochenta, se adoptó la estructura de carpetas en la interfaz del usuario, esta tendencia prevalece actualmente por íconos que ilustran carpetas y documentos. Los sistemas de gestión documental proveen una estructura navegable con la capacidad de indexar en una base de datos la información asociada, facilitando herramientas de distribución. (OFFICE 2006)

Ventajas de un sistema de gestión documental.

- Acceso de forma simultánea por múltiples usuarios a la misma o diferente documentación.
- Posibilidad de múltiples claves de búsqueda.
- Alta productividad.
- Ventajas competitivas y diferenciación de la información obtenida.
- Posible integración con otras aplicaciones.
- Ahorro en costes de mobiliario, archivadores, carpetas, subcarpetas y grapas.
- Posibilidad de vínculos, revisiones, actualizaciones y relaciones entre ficheros y/o documentos.

- Gestión más integrada de casi toda la documentación de una organización.
- De fácil uso e instalación. (Implantación de Sistemas de Gestión Documental 2006)

1.2.3. ¿Qué es la Investigación Científica?

La Investigación es un proceso que, mediante la aplicación de métodos científicos, procura obtener información relevante, para entender, verificar, corregir o aplicar el conocimiento.

Para obtener algún resultado de manera clara y precisa es necesario aplicar algún tipo de investigación, la investigación está muy ligada a los seres humanos, esta posee una serie de pasos para lograr el objetivo planteado o para llegar a la información solicitada. Tiene como base el método científico y este es el método de estudio sistemático de la naturaleza que incluye las técnicas de observación, reglas para el razonamiento y la predicción, ideas sobre la experimentación planificada y los modos de comunicar los resultados experimentales y teóricos.

El ser humano tiene una tendencia natural a buscar el sentido de las cosas, desde muy niño, pregunta al adulto; y ya joven, se sigue maravillando. De esto se deduce que existen diversos tipos de investigaciones, desde las más elementales y cotidianas por las cuales se busca ampliar el horizonte de los objetos conocidos, hasta la investigación científica con características propias de eficacia superior.

Algunas definiciones:

"...el proceso más formal, sistemático, e intensivo de llevar a cabo un método de análisis científico...es una actividad más sistemática dirigida hacia el descubrimiento del desarrollo de un cuerpo de conocimientos organizados. Se basa sobre el análisis crítico de proposiciones hipotéticas para el propósito de establecer relaciones causa-efecto, que deben ser probadas frente a la realidad objetiva. Este propósito puede ser ya la formulación-teoría o la aplicación-teoría, conduciendo a la predicción y, últimamente, al control de hechos que son consecuencia de acciones o de causas específicas."

"La investigación científica es una investigación crítica, controlada y empírica de fenómenos naturales, guiada por la teoría y la hipótesis acerca de las supuestas relaciones entre dichos fenómenos."

"...la investigación es un procedimiento reflexivo, sistemático, controlado y crítico que tiene por finalidad descubrir o interpretar los hechos y fenómenos, relaciones y leyes de un determinado ámbito de la realidad...una búsqueda de hechos, un camino para conocer la realidad, un procedimiento para conocer verdades parciales, o mejor, para descubrir no falsedades parciales."(GRAJALES 2000)

1.3. Tendencias y tecnologías actuales

Las exigencias de calidad son cada vez más rigurosas, las tendencias hacia los productos cada vez más personalizados, paralelamente a ello la vida del producto es cada vez más corta por lo que para hacer uso de las tendencias y tecnologías actuales se necesitan personas con flexibilidad al continuo cambio y entereza en desafiar la evolución de las tecnologías.

1.3.1. Desarrollo de Aplicaciones Web.

Las aplicaciones Web como parte de las TICS están cada vez más presentes en Internet, brindando un alto grado de disponibilidad de la información, haciéndose cada vez más dinámicas y ajustables a las necesidades de los usuarios, no solo como simples contenedores de información sino como verdaderos elementos integradores de las potencialidades de las aplicaciones de escritorios, multimedia, hipermedia y los conceptos positivos de la Web de las generaciones anteriores.

A pesar de los diversos significados que tiene este término para los diferentes desarrolladores, se puede considerar que una aplicación Web es un sistema donde la entrada del usuario (entrada de datos y navegación) afecta el estado del negocio. Esta definición intenta establecer que una aplicación Web es un sistema de software con estado de negocio y por tanto puede elaborarse utilizando la metodología de trabajo que propone RUP (Rational Unified Process), y utilizando UML (Unified Modeling Language) para los modelos.

En la actualidad y debido al auge de este tipo de aplicaciones se reconocen diferentes tipos de categorías en dependencia del tipo de negocio que modelen:

Informacionales: Periódicos, catálogos, manuales, libros electrónicos.

Interactivas: Formularios de registro, presentación de información personalizada.

Comunitarias: Salas de Chat, mercados, subastas online.

Es importante destacar que las aplicaciones Web tiene una arquitectura Cliente/Servidor, esta no es más que una forma de dividir las responsabilidades de un Sistema de Información separando la interfaz de usuario (Nivel de presentación) de la gestión de la información (Nivel de gestión de datos).

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

En esta arquitectura los clientes pueden realizar funciones como son el manejo de la interfaz de usuario que no es más que la forma en que los usuarios pueden comunicarse con una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, la captura y validación de los datos de entrada o la generación de consultas e informes sobre las bases de datos. Por otra parte los servidores pueden realizar funciones como son la de gestionar los periféricos compartidos, el control de accesos concurrentes a bases de datos compartidas o enlaces de comunicaciones con otras redes de área local.

Otro aspecto de suma importancia a la hora de diseñar una buena aplicación Web es el de la seguridad. Con respecto a este tema cabe destacar que no existe ningún método de seguridad perfecto, incluso en los grandes software y en los brindan mayor protección se han detectados algunos ataques de personas no autorizadas las cuales han utilizados herramientas para burlar la seguridad. Por lo que queda claro que si se desea mantener la integridad de los datos y la información que se brinda hay que poner todo el empeño en implementar una buena seguridad.

Entre las amenazas a las que están expuestas las aplicaciones Web se encuentran:

Manipular datos: Los datos se ven amenazados tanto si están en tránsito como si están almacenados. Ya que a través de la red se pueden interceptar los paquetes que por ella viajan y luego son descifrados por los atacantes.

Denegación de servicio: La disponibilidad y la confiabilidad de las aplicaciones se ven directamente afectadas por los ataques de denegación de servicio, esto hace que algunas de las aplicaciones no sean disponibles incluso para los usuarios autorizados a utilizar esos servicios. Los ataques del DoS se producen cuando un sistema está desbordado por el tráfico hasta el punto de no poder procesar las solicitudes del servidor. Una de las formas para evitar la denegación de servicio es, regular el límite de

ancho de banda y de los recursos para evitar que un sitio Web sea sobrecargado y bloquee todo un servidor. (LLOP 2004)

Ventajas de las aplicaciones Web.

Entre las ventajas más significativas de las aplicaciones Web se encuentran su forma de instalación y distribución, ya que normalmente su instalación consiste en configurar los componentes del lado del servidor sin necesidad de configuración del lado del cliente, mencionando su alto grado de portabilidad y capacidad de ser multiplataforma.

Las aplicaciones basadas en la Web deberían ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en Web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos. Esta es la razón por la cual las aplicaciones basadas en Web deberían tener mucho menos bugs que el software de escritorio descargable tradicional.

Además su principal protocolo de comunicación es http, el cual funciona normalmente desconectado, es decir, el cliente hace una petición al servidor, este la procesa y devuelve el resultado terminando la comunicación entre ellos.

En fin hoy en día ninguna compañía o centro educacional como es el caso de la Universidad de las Ciencias Informáticas (UCI) se plantea prescindir de una Web que los identifique, del mismo modo que no se priva de las comunicaciones telefónicas o de la electricidad. La Web permite una comunicación directa con público de cualquier lugar del mundo, las 24 horas del día.

1.3.2. Política de migración a software libre.

En los inicios de la informática, prácticamente todo el software era libre. Los primeros grandes ordenadores de las universidades en los años sesenta y setenta venían del fabricante con un conjunto de software, su código fuente, y sin ninguna limitación para modificarlo. De este modo, el software estaba sometido a un proceso de mejora continua con las diferentes aportaciones que hacía cada usuario. Eran comunes los casos en que los usuarios creaban programas propios, o mejoraban los que venían de serie

con estos sistemas, y los enviaban al fabricante del hardware para que, si lo consideraba adecuado, los distribuyera en la próxima versión de su sistema.

Este modelo se empezó a romper hacia principios de los años setenta, cuando las empresas observaron que el software podía venderse y empezaron a cerrar el acceso al código fuente. Así, el acceso al código fuente del programa y otras libertades, que hasta entonces eran comunes, fueron desapareciendo progresivamente y, con ellas, las libertades de los usuarios.

En 1983, Richard Stallman, consciente de la pérdida de libertades que se iba experimentando en el mundo del software, empezó un proyecto dirigido a construir un sistema operativo libre. Este proyecto, pocos meses después, lo llevó a crear la Free Software Foundation, una organización que ha sido la principal difusora de la filosofía del software libre durante los últimos veinte años y que organizó el desarrollo del principal software libre de sistemas.

El software libre representa una oportunidad sin precedentes para el desarrollo de lenguas minoritarias en el mundo de las nuevas tecnologías y el desarrollo local. Aprovechar esta oportunidad depende en gran medida de las administraciones, las universidades y las instituciones.

Actualmente en la Universidad de las Ciencias Informáticas, y en el país en general, existe una tendencia hacia la utilización en grado creciente de software libre. Cada vez se promociona más la migración desde los sistemas con licencia comercial que están en poder de unos pocos monopolios de la rama de la informática, hacia aquellos que se denominan “libres”, es decir, aquellos cuyo uso por parte de cualquiera que esté interesado en hacerlo está exento de pago.

Como institución de avanzada en el campo de la informática, la UCI está prácticamente obligada a llevar a cabo, y cuánto antes mejor, esta migración. El presente trabajo parte de esa premisa y se propone la construcción de un sistema que satisfaga las necesidades que lo originaron, haciendo uso de herramientas y tecnologías libres.

1.3.3. Principales lenguajes de desarrollo Web en la actualidad.

Html es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, es el formato estándar de las páginas Web y sus siglas significan HyperText Markup Language (*Lenguaje de Marcas Hipertextuales*).

Producto al auge de Internet se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para Web.(WIKIPEDIA 2007d)

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Sus siglas en inglés significan **eXtensible Markup Language** (lenguaje de marcas extensible).Permite definir la gramática de lenguajes específicos por lo que pudiera decirse que XML no es un lenguaje en particular sino una manera de definir lenguajes.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto y hojas de cálculo. (WIKIPEDIA 2007I)

Las **hojas de estilo en cascada** "Cascading Style Sheets" son una tecnología desarrollada con el fin de separar la presentación de la estructura del HTML. Funciona aplicando reglas de estilo a los elementos HTML, entre las que incluyen, tamaño, color de fondo, color del texto, posición de los elementos, márgenes y tipos de letra. Quedando de esta manera todo lo que tiene que ver con la parte gráfica de la Web, separada completamente de la estructura.

Este lenguaje desarrollado por la W3C, ha venido haciéndose cada vez más importante entre los diseñadores, gracias a la facilidad de uso y a los óptimos y flexibles resultados.

Utilizar CSS facilita un mejor flujo de trabajo, mayor organización del código, menos peso en las páginas, y más flexibilidad a los cambios, además de proporcionar un fácil y más rápido diseño que de la manera antigua.(RAMM 2005)

El lenguaje de programación **JavaScript**: es del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Por su sencillez esta pensado para realizar las acciones con rapidez y cuenta entre sus ventajas con ejecutar instrucciones como respuesta a las

acciones del usuario, por lo que ofrece la posibilidad de crear páginas interactivas además de poner a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente.(ALVAREZ 2007)

El lenguaje **PHP** fue creado en 1994 por Rasmus Lerdorf como un complemento para el lenguaje PERL. Lo incorporó por primera vez en su propia página Web para monitorizar las visitas que recibía. Fue tanta su popularidad que infinidad de usuarios le pidieron poder utilizar estos comandos en sus páginas, por lo que Rasmus decidió lanzar la primera versión completa de PHP, conocida por entonces como Personal Home Page Tools (Herramientas para Páginas Personales), que permitía a los desarrolladores incluir en sus páginas funcionalidades como libros de visitas, foros o contadores.

PHP es un acrónimo recurrente que significa "**PHP Hypertext Pre-processor**" y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+.

Es un lenguaje del lado del servidor que 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; lo cual permite la creación de Aplicaciones Web muy robustas. (WIKIPEDIA 2007h)

ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones Web comercializado por Microsoft. Es usado por programadores para construir sitios Web domésticos, aplicaciones Web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP). (WIKIPEDIA 2007a)

Java Server Pages (JSP), es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas Web con programación en Java.

Dicho lenguaje posibilita crear aplicaciones que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma, es un desarrollo de la compañía Sun Microsystems, y su funcionamiento se basa en scripts.(ALVAREZ)

Python es un lenguaje de programación, lanzado por primera vez en el año 1990, habitualmente comparado con TCL, Perl, Scheme, Java y Ruby. Actualmente, Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. La última versión estable del lenguaje es actualmente (Septiembre de 2006) la 2.5.

Guido van Rossum, más conocido como Guido, fue el creador de Python, un lenguaje de programación de scripting, la "oposición leal" a Perl, lenguaje con el cual mantiene una rivalidad amistosa. Los usuarios de Python consideran a éste mucho más limpio y elegante para programar.

Permite dividir el programa en módulos reutilizables desde otros programas Python. Viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas (o como ejemplos para empezar a aprender Python). También hay módulos incluidos que proporcionan E/S de ficheros, llamadas al sistema, sockets y hasta interfaces a GUI (interfaz gráfica con el usuario) como Tk, GTK, Qt entre otros.

Además de que el hecho de ser un lenguaje interpretado, hace que se ahorre un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. (WIKIPEDIA 2007i)

C# es una versión avanzada de C y de C++ y se ha diseñado especialmente para el entorno .NET. C# es un nuevo lenguaje orientado a objetos empleado por programadores de todo el mundo para desarrollar aplicaciones que se ejecuten en la plataforma .NET. De todas formas, C# no es parte del entorno .NET. C# es parte de Microsoft Visual Studio .NET 7.0. C# es un paso muy importante en la evolución de los lenguajes de programación, y es una solución ideal para las aplicaciones empresariales de alto nivel. Con C# puede desarrollar todo tipo de proyectos de aplicaciones cliente / servidor.(ARORA *et al.* 2002)

Unified Modeling Language UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software.

UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante

del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. (MARTÍNEZ 2006)

Principales Gestores de bases de Datos.

El software de bases de datos ha experimentado un auge extraordinario a raíz de la progresiva informatización de casi la totalidad de las empresas de hoy día, rapidez, efectividad en los procesos y los grandes flujos de información son la necesidad más apremiante a la hora de optimizar servicios y productos.

Ante esta notable demanda de soluciones informáticas han surgido multitud de gestores de bases de datos, estos son programas que permiten manejar la información de modo sencillo y que prestan servicios para el desarrollo y el manejo de bases de datos.

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. (MYSQL 2007)

Postgres es un sistema gestor de bases de datos relacionales orientadas a objetos con cerca de una década de desarrollo, soporta casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python). (PostgreSQL)

Oracle es un sistema de gestión de base de datos (o RDBMS por el acrónimo en inglés de Relational Data base Management System), fabricado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- Soporte de transacciones.
- Estabilidad.

- Escalabilidad.
- Es multiplataforma.

Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux. (WIKIPEDIA 2007g)

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Entre sus características figuran:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase ASE. (WIKIPEDIA 2007f)

Firebird es un sistema de administración de base de datos (o RDBMS) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. Su código fue reescrito de C a C++. Es un desarrollo derivado de interbase apoyado por desarrolladores de todo el mundo que dan

continuidad a lo que se conoce como bases de datos de código abierto. Posee algunas características de interbase aunque en la actualidad toman diferencias debido a que son de desarrollos diferentes.

Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows. Ejecutable pequeño, con requerimientos de hardware bajos. Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros (embedded). Soporte de transacciones y claves foráneas. Es medianamente escalable. Buena seguridad basada en usuarios/roles. (WIKIPEDIA 2007b)

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, y que está contenida en una relativamente pequeña librería en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia de los sistemas de gestión de base de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la librería SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción. (WIKIPEDIA 2007k)

Principales servidores Web.

Servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft). (WIKIPEDIA 2007j)

Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web).

El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl. (WIKIPEDIA 2007e)

IDES de Desarrollo.

Dreamweaver es un editor HTML profesional para diseñar, codificar y desarrollar sitios, páginas y aplicaciones Web. Tanto si desea controlar manualmente el código HTML como si prefiere trabajar en un entorno de edición visual, Dreamweaver le proporciona útiles herramientas que mejorarán su experiencia de creación Web. Las funciones de edición visual de Dreamweaver permiten crear páginas de forma rápida, sin escribir una sola línea de código. No obstante, si prefiere crear el código manualmente, Dreamweaver también incluye numerosas herramientas y funciones relacionadas con la codificación. Además, le ayuda a crear aplicaciones Web dinámicas basadas en bases de datos empleando lenguajes

de servidor como ASP, ASP.NET, ColdFusion Markup Language (CFML), JSP y PHP.(MACROMEDIA 2003)

NuSphere PHPEd es un editor para lenguajes de programación que soporta PHP, Perl, Javascript, SQL y más, y que dispone de plantillas de documentos, código de colores para comandos, cliente de FTP y un servidor Web integrados, entre otras opciones.

Contiene entornos de desarrollo profesional, análisis de errores, ayudas para la localización de estos en el código, publicación segura de código en servidores.

Zend Estudio es un Ambiente de Desarrollo Integrado (IDE) disponible para diseñadores profesionales que abarca todos los componentes de desarrollo necesario para el ciclo de vida de las aplicaciones PHP. Esta desarrollado por los impulsores de la tecnología de servidores PHP. Posee un completamiento de código de gran nivel. Esta escrito completamente en java lo que hace que resulte un poco más lento que otros IDEs pero esto garantiza también que pueda correr en varios sistemas operativos. Zend Studio posee una ayuda abundante además de una referencia completa de funciones predefinidas.(ALVAREZ)

Rational Rose Enterprise Edition es una herramienta desarrollada por Rational, una empresa que se dedica exclusivamente al desarrollo de herramientas relativas al desarrollo software de alto nivel. Cuenta con paquetes dedicados a todo tipo de situaciones, teniendo por tanto desde elementos para el desarrollo software a través de UML, como también puedan ser paquetes para la captura de requisitos, el uso de algoritmos para la eficiencia en el desarrollo de sistemas complejos mediante el control del personal. Esta herramienta también se encarga de la automatización de los sistemas para la posterior generación de código así como para labores de ingeniería inversa (es decir, realización de los diagramas una vez conocido el código). Se puede decir que es una forma de ayuda para la comprensión del sistema y de sus distintos componentes, y lo mejor es que puedes aplicar ingeniería inversa a una multitud de códigos distintos, siempre que obviamente estén orientados a objetos. La clave está en la creación de componentes, los cuales van a contener una serie de archivos dentro de los cuales se encuentran las distintas clases pertenecientes ha dicho componente. Rational Rose presenta una pequeña desventaja, y es que necesita de mucha memoria para que su manejo sea de forma más rápida y eficiente. (*Rational Rose* 2006)

Tecnologías enfocadas a la Web 2.0.

AJAX: (Asynchronous Javascript and XML) traducido como indica el título, no es más que una forma de programar aplicaciones interactivas para Web. A esta evolución de DHTML se le ha denominado Web 2.0. Para ello utiliza XHTML y CSS para formatear la información, DOM (Document Object Model) para interactuar y visualizar dinámicamente la información, se apoya en XML, XSTL para manipular la información mostrada, el objeto XMLHttpRequest (no estándar) y Javascript para actualizar los datos sin necesidad de refrescar la página, y para manipular todas esas tecnologías. Ajax no es una tecnología en si misma, sino un término que se refiere al uso de un grupo de tecnologías (MARCIBARRIOS 2005)

Adobe Flex es la solución de desarrollo de aplicaciones más completa y potente para crear y desarrollar aplicaciones de Internet sofisticadas (RIA, del inglés Rich Internet Applications) en la empresa y en la Web. Permite a las empresas crear aplicaciones personalizadas y ricas en elementos multimedia que mejoran notablemente la experiencia del usuario y que están revolucionando la manera en que las personas interactúan con la Web.

El objetivo de Flex es permitir a los desarrolladores de aplicaciones web construir rápida y fácilmente Aplicaciones de Internet Ricas, también llamadas RIAs. En un modelo multi-capa, las aplicaciones Flex son el nivel de presentación. (ADOBE 2007; WIKIPEDIA 2007c)

Selección de la tecnología a utilizar

Después de haber realizado un estudio minucioso de todas las tecnologías existentes para el desarrollo de sistemas soportados en la Web. Se determinó que algunas se destacaban más que otras por las bondades que podría traer consigo su elección en consecuencia con la aplicación que se pretende desarrollar.

Como lenguaje del lado del servidor se utilizará PHP, MySQL como sistema gestor de base de dato y Apache como servidor Web. Para lograr una mayor interactividad en el sistema se hará uso de AJAX como filosofía de integración entre las tecnologías javascript y xml de una forma asincrónica. Dreamweaver como editor html, NuSpher PhpED como editor de código PHP y finalmente RUP como

metodología de desarrollo, usando PHP Data Objects (PDO) como capa de abstracción de datos puesto que si en determinado momento se desea cambiar de gestor de base de datos esto facilitará el trabajo.

1.3.4. ¿Por qué utilizar PHP y no otro?

Se decide utilizar PHP como script del lado del servidor debido a que cuenta con una serie de ventajas respecto a otros como Microsoft Active Server Pages (ASP), Java Server Pages (JSP), Perl por mencionar algunos, entre las ventajas se pueden citar: Su alto rendimiento, bibliotecas incorporadas para muchas tareas Web habituales, bajo coste, de fácil aprendizaje y uso. A continuación se explicarán en detalles muchas de estas características.

PHP corre en casi todas las plataformas utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT, 2000,XP,) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al sistema operativo. La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C se incluyen Java y Javascript, de hecho mucha de la funcionalidad del PHP se la debe al C en funciones como fread () o strlen ().

Es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código. Presenta interfaces distintas para cada tipo de servidor. Actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THHTTPD. Otra alternativa es configurarlo como módulo CGI. Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. PHP incorpora una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código. Generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.

Otra de las características que hacen a PHP sea uno de los lenguajes más utilizados hoy en día en la Web se debe a que es Open Source, lo cual significa que el usuario no depende de una compañía

específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione. (ALVAREZ)

1.3.5. ¿Por qué MySQL?

Entre las razones por las que se selecciona MySQL como sistema gestor de base de datos se pueden citar: su fácil integración con PHP como lenguaje del lado del servidor, opera en una arquitectura cliente/servidor. Es uno de los sistemas gestores de bases de datos “Open Source” más popular, o sea que puede ser bajado de Internet y usarlo sin tener que pagar, además que cualquiera puede estudiar su código y adecuarlo a las necesidades que requiera.

MySQL es muy rápido, fiable y fácil de usar, surge para manipular bases de datos muy grandes. Es un sistema multiplataforma de base de datos relacionales, lo que da velocidad y flexibilidad, está escrito en C y C++ probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes, cuenta con un sistema de privilegios contraseñas muy seguro que permite la autenticación básica para el acceso al servidor.

1.3.6. ¿Por qué usar DreamWeaver como ID principal en el diseño?

DreamWeaver es una herramienta con una gran flexibilidad y potencia en el diseño Web mundialmente conocida. Permite realizar diseños perfectos en la vista Diseño de la forma más cómoda. Proporciona una potente combinación de herramientas visuales de diseño, funciones de desarrollo de aplicaciones y soporte para la edición del código. Su total compatibilidad con CSS facilitan la visualización, edición y desplazamiento dentro de los archivos, acelerando el flujo de compatibilidad con el navegador. Una de las grandes ventajas que posee es su compatibilidad con las principales tecnologías de desarrollo Web, como son: HTML, XHTML, CSS, XML, JavaScript, Ajax, PHP entre otras.(ADOBE 2007)

Conclusiones Parciales.

En este capítulo se hizo un análisis de la teoría y la tecnología a utilizar en el desarrollo de la propuesta de solución, así como algunos conceptos y tendencias que esta debe adoptar. Se fundamentó la elección del lenguaje de programación, el sistema gestor de bases de datos, el servidor Web, y la metodología de

desarrollo de software así como el uso de otras tecnologías y finalmente se hace una pequeña descripción de las tecnologías que se van a usar en el desarrollo de la aplicación.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.

Introducción.

En el presente capítulo se representa el Modelo de Dominio mediante un diagrama de objeto, pues los procesos del negocio asociados al objeto de estudio no están bien definidos como para elaborar un sistema que se ajuste a las reglas del mismo.

Se destacan los requisitos funcionales y no funcionales que debe tener el sistema que se propone, lo que permite hacer una concepción general de los objetivos del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan.

2.1. Modelo del Dominio.

Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Debido a la sencillez de la estructura y que no están bien definidos los procesos del negocio estudiado, se arriba a la conclusión de que no se necesita un modelado completo del negocio por lo cual se construye un modelo de dominio. Este permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes, desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. El objetivo del modelo de dominio es comprender y describir las clases más importantes dentro del contexto del sistema, en otras palabras el modelado del dominio deberá contribuir a una comprensión del problema que el sistema resuelve en relación a su contexto.

También se hará uso de un glosario de términos para identificar todos los conceptos que se utilizarán en el diagrama.

Administrador: El administrador es un rol que se le concederá a un profesor determinado con el objetivo de darle soporte y administrar el Sistema, teniendo en cuenta que forma parte de los miembros del grupo y puede desempeñarse como tal.

Profesor: Lo constituye cualquier profesor miembro del Grupo de Investigación de Matemática y Física Computacional.

Usuario: Está compuesto por estudiantes, profesores o trabajadores que hagan uso del sistema.

Acta: Hace referencia a las actas de las reuniones del grupo que serán almacenadas en el sistema.

Bibliografía: Se refiere a las bibliografías utilizadas por miembros del grupo en la realización de trabajos investigativos y que se cree pueda ser de gran utilidad en la realización de trabajos posteriores.

Trabajos Científicos: Trabajos realizados por miembros del grupo de investigación.

Documentos: Lo constituye cualquier información que sea referenciada en el sistema.

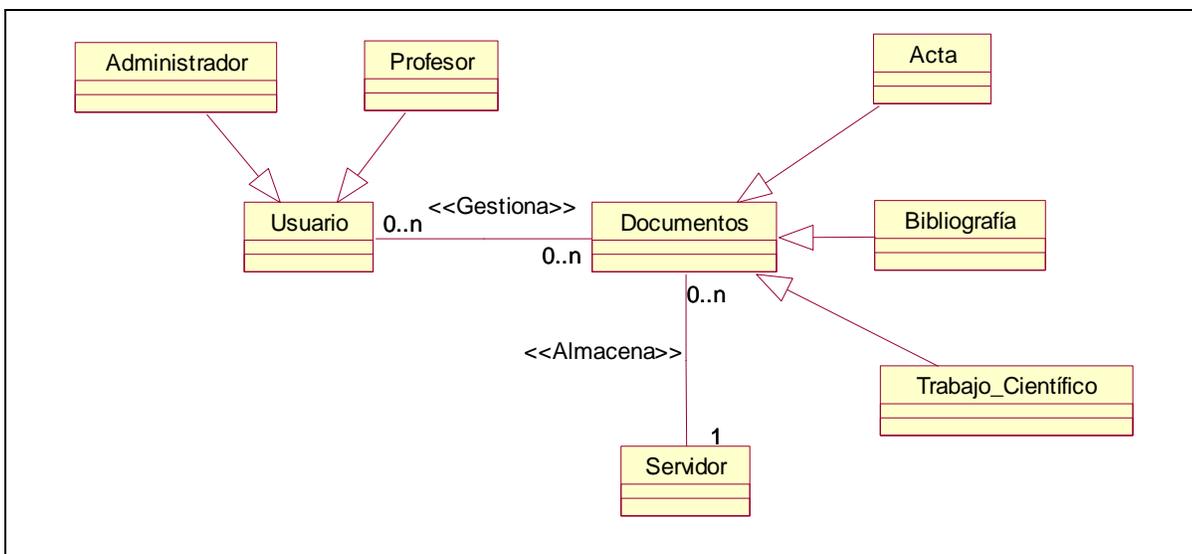


Fig. 2.1 Diagrama del Modelo de Dominio.

2.2. Reglas del negocio.

Si algún profesor perteneciente a alguna de estas Ciencias desea formar parte de este Grupo de Investigación, debe comunicarle su interés al Jefe del Grupo y este conjuntamente con los demás miembros analizará si podrá pertenecer al Grupo o no.

El profesor que desee formar parte del equipo debe cumplir con los requisitos necesarios, además presentar una serie de datos personales como Currículum Vitae, si tiene experiencias anteriores o no en trabajos similares.

Los trabajadores o integrantes del Grupo de investigación tendrán acceso a toda la información del sistema. Pero cada uno tendrá un rol que le permitirá hacer o no ciertas actividades.

Las actas de las reuniones deberán ser revisadas por el Jefe del Grupo antes de las cuarenta y ocho horas pasadas dichas reuniones.

2.3. Requerimientos funcionales.

RF1- Autenticar Usuario.

RF1.1- Verificar que el usuario este registrado en el sistema.

RF1.2- Verificar usuario y contraseña en el dominio UCI.

RF1.3- Verificar usuario y contraseña en el Sistema.

RF2- Gestionar Usuarios del sistema.

RF2.1- Registrar un nuevo usuario al sistema.

RF2.2- Modificar usuario existente.

RF2.3- Establecer privilegios.

RF2.4- Eliminar usuario.

RF3- Confeccionar actas de la reuniones.

RF3.1- Redactar acta de la reunión (Fecha, Hora, Asistencia a reunión, Orden del día, Acuerdos e Inquietudes).

RF3.2- Almacenar acta en la base de datos.

RF3.3- Mostrar Acta.

- Permitir a un profesor imprimir dicha acta.

RF3.4- Eliminar un Acta deseada.

RF4- Gestionar noticias (Curiosidades, Cursos, Seminarios, Reuniones, Talleres).

RF4.1- Publicar noticias.

RF4.2- Modificar noticias.

RF4.3- Eliminar noticias anteriormente publicadas y sus comentarios correspondientes.

RF5- Listar Miembros.

RF5.1- Mostrar un listado de los miembros del grupo de investigación con los siguientes datos (Vínculo al Currículum Vitae, Facultad, email, línea investigativa a la que pertenece).

RF6- Gestionar Actividades del Grupo.

RF6.1- Crear actividad definiendo descripción, fecha de ejecución, responsable.

RF6.2- Modificar datos de la actividad que desee.

RF6.3- Eliminar actividad automáticamente una vez culminada su fecha de cumplimiento.

RF7- Gestionar bibliografía de temas relacionados con investigaciones hechas.

RF7.1- Registrar bibliografía.

RF7.2- Descargar bibliografía para ser usada.

RF7.2- Eliminar bibliografía.

RF8- Gestionar Trabajos Investigativos.

RF8.1- Publicar trabajos investigativos definiendo (título del trabajo, fecha publicación, autor).

RF8.2- Eliminar trabajo anteriormente publicado.

RF9- Consultar información pública (Trabajos Científicos, Actividades, Noticias).

RF9.1- Listar y descargar trabajos investigativos realizados.

RF9.2- Listar noticias ordenadas por fecha de publicación.

RF9.2.1- Permitir agregar comentarios a las noticias deseadas.

RF9.3- Permitir listar actividades y eventos que se desarrollaran en el grupo.

2.4. Requerimientos no funcionales

Requerimiento de software:

- Para la implantación del sistema se requiere de un servidor Linux ó Windows NT.
- Apache para ambas plataformas como servidor Web y MySQL5 como servidor de bases de Datos.

- Los requerimientos en el lado del cliente para la utilización del sistema sólo se limitan a tener disponible un navegador Web compatible o superior con Internet Explorer 4.

Requerimiento de hardware:

- La máquina servidora debe tener como mínimo las siguientes características de hardware: Procesador Pentium 4 2.0 GHz o superior, 512 Mb de memoria RAM (incluye la utilizada por el SO) y 40Gb de capacidad en disco duro.
- Las computadoras situadas en los puestos de trabajo de los usuarios requieren como mínimo un procesador Pentium 2, 64 Mb de memoria RAM. Estas máquinas deben de estar conectadas en red con el servidor.

Apariencia o interfaz externa:

- El software brinda una interfaz simple y de fácil uso para que el usuario no tenga dificultad al interactuar con el sistema.
- La interfaz del sistema se ajusta a los estándares establecidos para el desarrollo de un buen diseño.
- Está diseñada de modo tal que el usuario pueda ir de un punto a otro dentro de ella con gran facilidad.
- Se trata de que la aplicación sea lo más interactiva posible.

Usabilidad:

- El sistema podrá ser usado por cualquier tipo de personas que posean conocimientos básicos en el manejo de la computadora y el ambiente Web en sentido general.

Rendimiento:

- Se debe garantizar que el tiempo de respuesta del sistema ante las solicitudes de los usuarios para cada acción a realizar por estos sea el menor posible, para garantizar el buen funcionamiento del sistema debido a que es muy dinámico e interactivo.

Portabilidad:

- El sistema correrá no solo sobre Windows sino también sobre Linux, permitiendo que el servidor pueda ser cambiado sin importar sistema operativo llevando a cabo esta acción sin necesidad de efectuar cambios significativos.

Requerimientos de Seguridad:

- La información debe ser confidencial, para ello se pretende establecer un sistema de permisos y usuarios para el acceso a la información.
- Para mantener la integridad en el mismo sólo se podrá acceder al sistema después de autenticarse encriptándose la contraseña mediante el algoritmo de encriptación MD5, por lo que viaja hacia el servidor de forma protegida.

2.5. Actores del sistema a automatizar.

Tabla 1. Definición de actores del sistema a automatizar

Nombre del actor	Descripción
Usuario	<p>Es aquel que en determinado momento solicita entrar al sistema y posteriormente se le permite el acceso a determinado contenido, en correspondencia con el rol que éste juegue.</p> <p>Como Usuario tiene acceso a trabajos investigativos realizados con la oportunidad de descargarlos si lo desea, puede agregar comentarios a las distintas noticias que se presentaran, así como consultar la fecha y lugar de las distintas actividades y eventos que en el mismo se realizan.</p> <p>Los requerimientos funcionales asociados con este actor son los pertenecientes al RF1, RF9.</p>
Administrador	<p>Es el encargado por su rol de administrar los usuarios del sistema y asignarles el rol correspondiente que desempeñarán, responsable de las publicaciones tanto de noticias, curiosidades relacionadas con el tema de la investigación como la publicación</p>

	<p>de los diferentes eventos que se llevarán a cabo en el grupo, podrá, cuando lo desee, actualizar, eliminar o publicar una nueva noticia, darle soporte al sistema, además de poder realizar otras actividades como miembro del grupo.</p> <p>Los requerimientos funcionales asociados con este actor son los pertenecientes al RF2, RF4.</p>
Profesor	<p>Tiene la posibilidad de realizar una serie de actividades como listar miembros del grupo, con la posibilidad imprimir dicho listado, gestionar la bibliografía utilizada en trabajos hechos anteriormente. También puede publicar una vez terminados, sus trabajos científicos, eliminar documentos almacenados por él, así como confeccionar y salvar el acta de las reuniones el día que desempeñe el rol de Secretario de Actas, además otras actividades a las que tienen acceso todos los usuarios.</p> <p>Los requerimientos funcionales asociados con este actor son los pertenecientes al RF3, RF5, RF6, RF7, RF8 y RF9.</p>

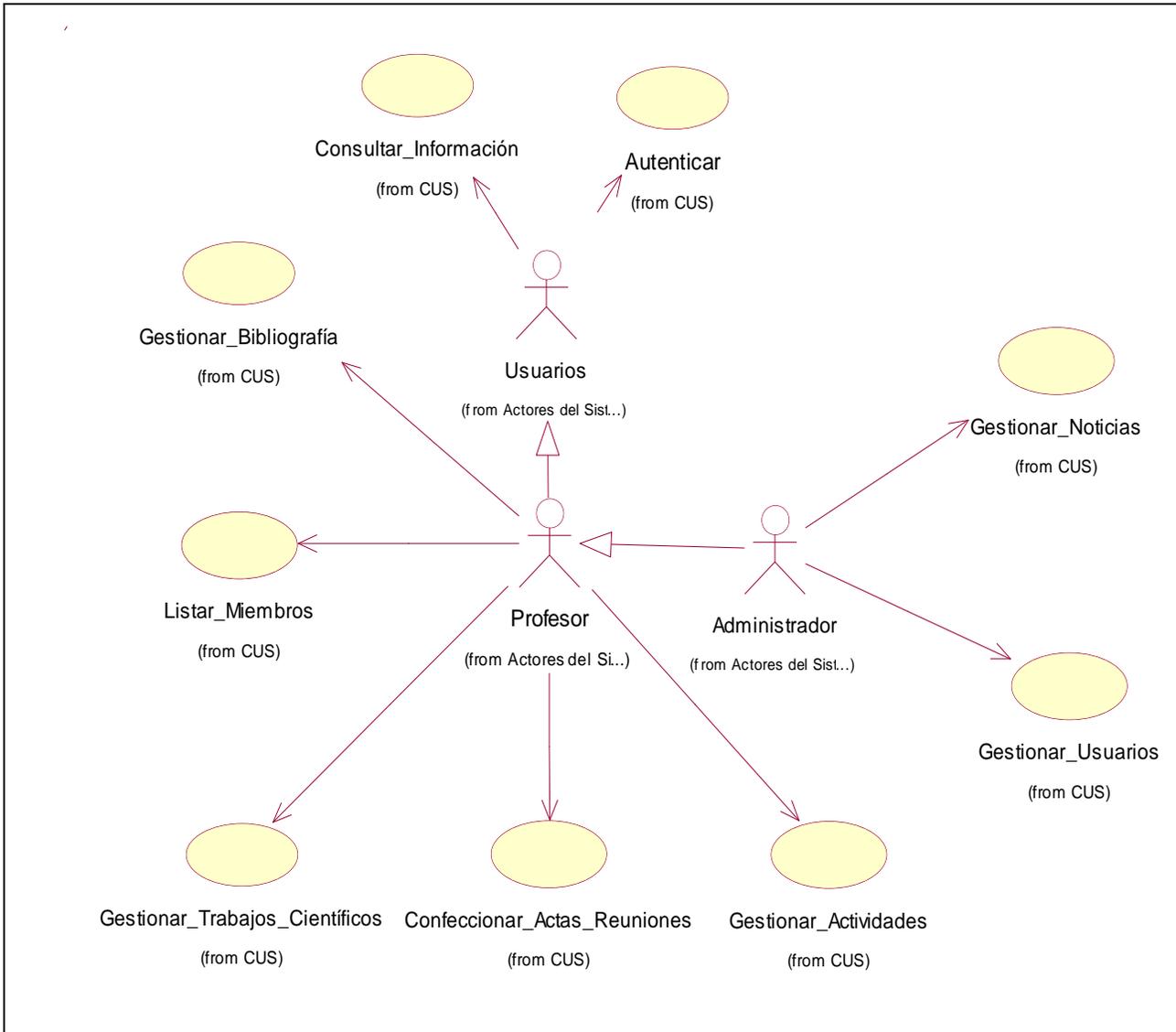


Fig. 2.2 Diagrama de Casos de Uso del Sistema.

2.6. Descripción de los casos de uso

Tabla 2. Gestionar Actividades.

CU-1	Gestionar Actividades
Actor	Profesor
Resumen	El caso de uso se inicia cuando un profesor desea visualizar actividades existentes o introduce alguna actividad en el cronograma, con la posibilidad de modificarlas si en algún momento fuera necesario. Una vez concluida la fecha de cumplimiento de una actividad esta se eliminará automáticamente a una hora determinada.
Referencia	RF6.1, RF6.2, RF6.3
Precondiciones	El usuario debe haberse registrado y tener el rol Profesor o Administrador del Sistema. Debe estar confeccionada la plantilla del Grupo de Investigación
Poscondiciones	Si la acción fue adicionar se adicionó correctamente la actividad, si fue modificar se modificó satisfactoriamente.

Tabla 3. Confeccionar Actas de Reuniones.

CU-2	Confeccionar Actas de Reuniones
Actor	Profesor
Resumen	El caso de uso se inicia cuando al concluir una reunión y quedar redactada correctamente el acta, el profesor que se desempeña en ese momento como secretario procede a almacenar dicha acta en el sistema como es debido o eliminar una ya existente si así lo desea.
Referencia	RF3.1, RF3.2, RF3.3, RF3.4
Precondiciones	El usuario debe haberse registrado y tener el rol Profesor o Administrador del Sistema. Para que el acta pueda ser almacenada tiene que haber sido revisada por el Jefe del Grupo de investigación.
Poscondiciones	Si la acción fue adicionar se adicionó correctamente el acta, si fue eliminar se elimina satisfactoriamente.

Tabla 4. Autenticar Usuario.

CU-3	Autenticar Usuario
Actor	Usuario
Resumen	El caso de uso se inicia cuando el usuario solicita entrar al sistema, introduciendo nombre y contraseña, verificando sus credenciales en el dominio uci.cu para permitirle el acceso, asignándole el rol que se corresponde en el sistema con ese usuario y contraseña o permitirle entrar como usuario. En caso de existir problema con el dominio y no poderse conectar entonces lo intenta en el sistema, consultando su perfil de usuario.
Referencia	RF1.1, RF1.2, RF1.3
Precondiciones	El usuario debe haber sido adicionado previamente al sistema.
Poscondiciones	Se le da acceso al usuario en el sistema, estableciéndose un rol para el mismo.

Tabla 5. Gestionar Noticias.

CU-4	Gestionar Noticias
Actor	Administrador
Resumen	Este caso de uso se inicia cuando uno de los actores concebido como Administrador, inserta una nueva noticias en el sistema, también puede en determinado momento actualizar o eliminar una noticia que haya sido publicada con anterioridad. Al eliminar una noticia se eliminarán todos lo comentarios relacionados con la misma.
Referencia	RF4.1, RF4.2, RF4.3
Precondiciones	El usuario debe haberse registrado y tener el rol Administrador del Sistema.
Poscondiciones	Si la acción fue adicionar se adicionó correctamente la noticia, si fue modificar o eliminar se modificó o eliminó satisfactoriamente.

Tabla 6. Gestionar Usuarios.

CU-5	Gestionar Usuarios
Actor	Administrador
Resumen	Este caso de uso se inicia cuando el Administrador del sistema procede a adicionar un nuevo miembro al grupo, eliminar o modificar los datos de uno existente. Para adicionar un nuevo integrante al Grupo, el Administrador debe introducir correctamente el usuario del dominio para que posteriormente el sistema devuelva todos los datos necesarios de ese usuario que se encuentran en el servidor del Dominio, creando así un nuevo perfil de usuario en el sistema.
Referencia	RF2.1, RF2.2, RF2.3, RF2.4
Precondiciones	El usuario debe haberse registrado y tener el rol Administrador del Sistema. Para poder adicionar un nuevo miembro al Grupo de Investigación tiene que anteriormente haber sido aprobado por el Jefe del Grupo, así como por los demás integrantes del grupo.
Poscondiciones	Si la acción fue eliminar, se eliminará el perfil del usuario de la base de datos, si la acción fue adicionar, se adicionará un nuevo perfil de usuario y si la acción fue modificar, se podrán modificar los datos del perfil deseado.

Tabla 7. Listar Miembros.

CU-6	Listar Miembros
Actor	Profesor
Resumen	Este caso de uso se inicia cuando un profesor decide observar el listado de miembros del Grupo de Investigación, puede obtener este listado por una determinada línea de investigación, de cada miembro se mostrarán sus datos personales como (Vínculo al Currículum Vitae, Facultad, email, línea investigativa a la que pertenece, fecha de ingreso al grupo).

Referencia	RF5.1
Precondiciones	El usuario debe haberse registrado y tener el rol de Profesor o Administrador del Sistema. Debe estar conformada la Plantilla del Grupo.

Tabla 8. Gestionar Bibliografía.

CU-7	Gestionar Bibliografía
Actor	Profesor
Resumen	Este caso de uso se inicia cuando se decide almacenar la bibliografía utilizada en la realización de algunos de los trabajos científicos del grupo que se cree pueda ser de utilidad para trabajos posteriores. También se podrá descargar o eliminar una bibliografía ya existente, pero solo las que fueron almacenadas por él.
Referencia	RF7.1, RF7.2, RF7.3
Precondiciones	El usuario debe haberse loqueado con el rol de profesor o administrador.
Poscondiciones	Si la acción fue adicionar, se adicionó la nueva bibliografía en la base de datos, si la acción fue eliminar, se eliminó la bibliografía ya existente.

Tabla 9. Gestionar Trabajos Científicos.

CU-8	Gestionar Trabajos Científicos
Actor	Profesor
Resumen	Este caso de uso se inicia cuando un profesor después de haber terminado un determinado trabajo investigativo y mostrado a los demás miembros del grupo procede a publicarlo en el sistema para que pueda ser consultado por todo el interesado, también si en algún momento lo desea puede eliminar los trabajos que hayan sido publicados por él.

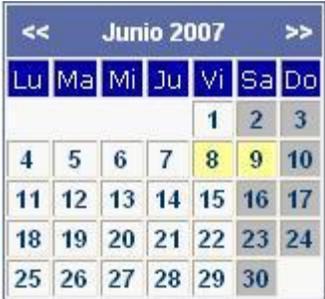
Referencia	RF8.1, RF8.2
Precondiciones	El usuario debe haberse loqueado con el rol de profesor o administrador.
Poscondiciones	Si la acción fue adicionar se adicionó correctamente el trabajo científico, si fue eliminar se elimino dicho trabajo conjuntamente con el fichero que se había subido al servidor.

Tabla 10. Consultar información.

CU-9	Consultar información
Actor	Usuario
Resumen	Este caso de uso se inicia cuando un usuario accede a información que se presenta en el sistema, ya sea para descargar algún trabajo que sea de su interés, observar las noticias publicadas con la oportunidad de dar su opinión al respecto, consultar las distintas actividades del mes. O simplemente desee navegar teniendo en cuenta que tendrá acceso a la información y recursos definidos para el rol que juega en el sistema.
Referencia	RF9.1, RF9.2, RF9.2.1, RF9.3
Precondiciones	Para poder descargar algún trabajo o consultar las distintas actividades del grupo debe haberse registrado en el sistema.

2.7. Casos de Uso Expandidos.

Tabla 11. Gestionar Actividades.

Caso de uso	
CU-1	Gestionar Actividades.
Propósito	Adicionar nuevas actividades del Grupo de Investigación o modificar alguna existente.
Actores: Profesor	
Resumen: El caso de uso se inicia cuando el usuario con el rol de profesor accede a una actividad determinada mediante el calendario que aparece en el lateral derecho, una vez mostrada esta actividad puede insertar una nueva introduciendo todos los datos pertinentes o puede actualizar la actividad seleccionada.	
Referencias	RF 6.1, RF 6.2, RF 6.3
Precondiciones	El usuario debe pertenecer al grupo de profesores.
 <p>Pantalla 1</p>	

	1. Muestra el formulario para insertar los datos de la Actividad (Pantalla 3).
2.Llena el formulario introduciendo los datos correspondiente, Actividad (A), responsable (B), lugar (C), fecha (D), descripción (E) que puede o no tener y presiona el botón registrar (F).	3. Chequea que no haya ningún campo requerido en blanco e inserta la actividad en la base de datos. Comunicándole al actor la correcta o no inserción.

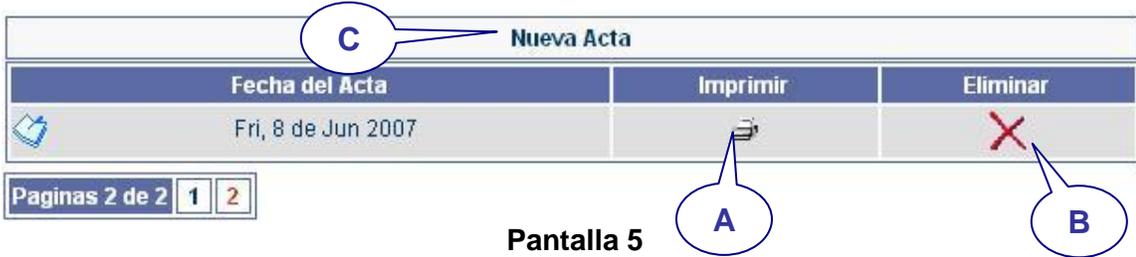
Sección: "Actualizar actividad"

Pantalla 4

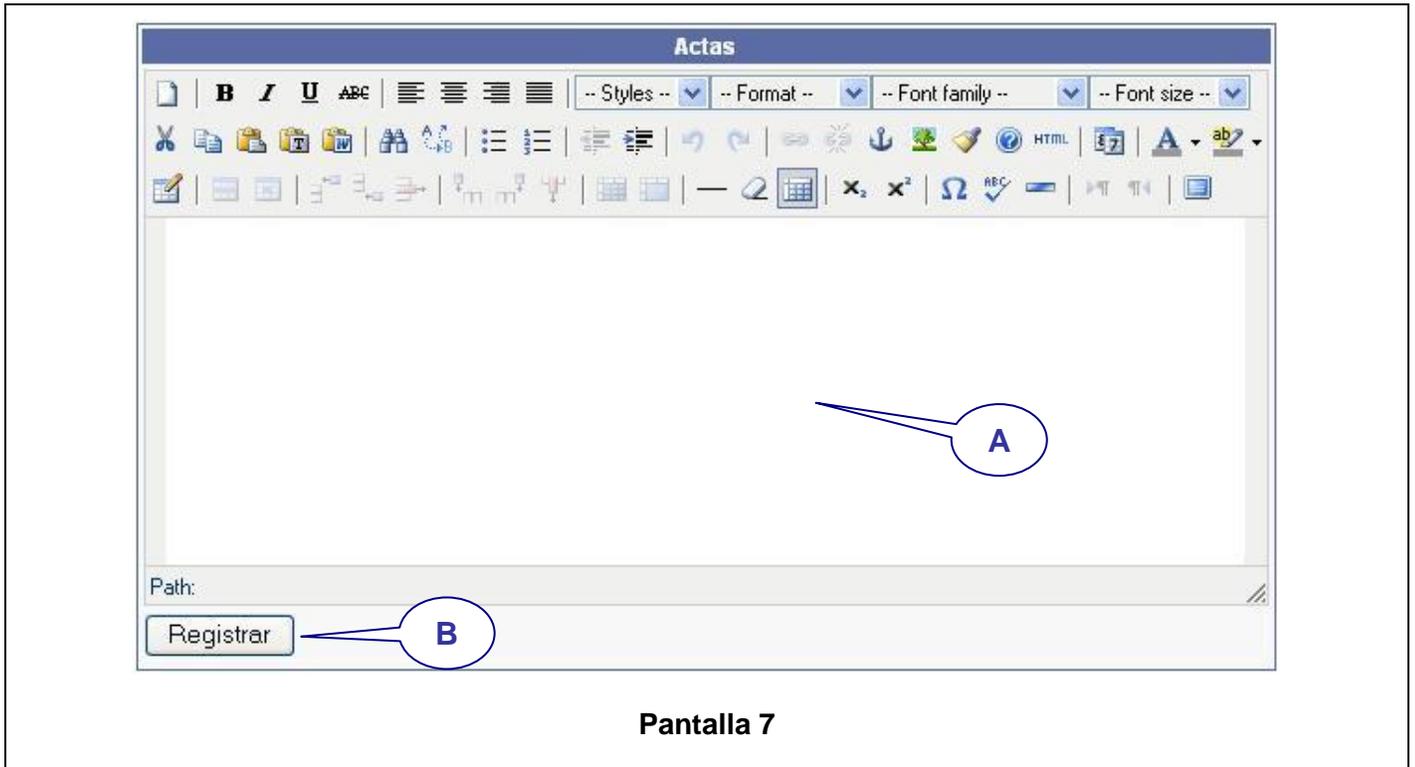
	2. Muestra el formulario con los datos de la actividad que se desea modificar (Pantalla 4).
--	---

3. El actor cambia los datos del campo que desee y posteriormente presiona el botón actualizar (A).	4. Actualiza la actividad seleccionada con los nuevos datos que ha modificado el actor.
---	---

Tabla 12. Confeccionar Actas de Reuniones.

Caso de uso	
CU-2	Confeccionar Actas de Reuniones.
Propósito	Almacenar las actas de las reuniones del Grupo de Investigación.
Actores: Profesor	
Resumen: El caso de uso se inicia cuando el usuario con el rol de profesor accede al sistema y después listar todas actas de reuniones existentes ordenadas por fecha, puede eliminar la que desee, seleccionar una específica con el objetivo de imprimirla o simplemente adicionar una nueva, la fecha se almacenará automáticamente en la base de datos una vez insertada el acta.	
Referencias	RF 3.1, RF 3.2, RF 3.3, RF 3.4
Precondiciones	El usuario debe haberse registrado con el rol de profeso o administrador y para ser almacenada el acta debe haber sido revisada por el Jefe del Grupo.
 <p>Pantalla 5</p>	
Acción del actor	Respuesta del sistema
1. El caso de uso comienza cuando el actor selecciona el vínculo Actas de Reuniones.	2. Muestra todas las Actas Ordenadas por fecha (Pantalla 5).
3. Si el profesor elige eliminar Acta (B).	4. Elimina el acta inmediatamente de la base de datos.
5. a) Si elige Imprimir un Acta específica(A), ver sesión "Imprimir Acta". b) Si elige insertar nueva acta (C), ver sesión	

"Insertar Acta".	
Sesión "Imprimir Acta".	
<p>Acta del Grupo de Investigacion</p>  <p>Pantalla 6</p>	
	1. Muestra una vista de impresión del acta seleccionada con todos sus datos. (Pantalla 6).
2. Si presiona el botón imprimir (A)	3. Imprime el Acta.
4. Si Oprime Cerrar (B).	5. Cierra la ventana vista de impresión.
Sesión "Insertar Acta".	

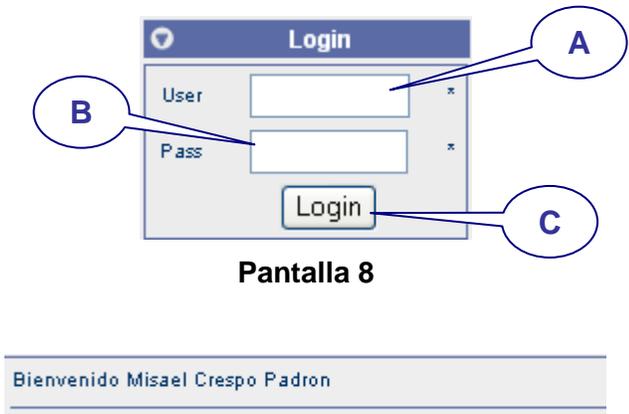


Pantalla 7

	<p>1. Muestra un Editor de texto donde el profesor puede redactar una nueva Acta para su posterior almacenamiento (Pantalla 7).</p>
<p>2. Redacta el acta en el editor (A) y posteriormente oprime el botón registrar (B).</p>	<p>3. Verifica que el editor no esté vacío y almacena el acta en la base de datos.</p>

Tabla 13. Autenticar Usuario.

Caso de uso	
CU-3	Autenticar Usuario
Propósito	Verificar las credenciales del usuario para permitirle el acceso al sistema y asignarle un rol.
Actores: Usuario	

<p>Resumen: El caso de uso se inicia cuando el usuario solicita entrar al sistema, introduciendo nombre y contraseña, verificando sus credenciales en el dominio uci.cu para permitirle el acceso, asignándole el rol que se corresponde en el sistema con ese usuario y contraseña o permitirle entrar como usuario. En caso de existir problema con el dominio y no poderse conectar entonces lo intenta en el sistema, consultando su perfil de usuario.</p>	
Referencias	RF1.1, RF1.2, RF1.3
Precondiciones	Para asignarle un rol tiene que estar registrado en la base de datos del sistema.
 <p>Pantalla 8</p> <p>Pantalla 9</p>	
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el usuario solicita entrar al sistema.	2. Muestra el login (Pantalla 8).
3. Entra usuario (A), contraseña (B) y posteriormente presiona Login (C)	4. Se conecta con el Dominio uci.cu
	5. Verifica las credenciales en el Dominio. - Crea una sesión con el nombre correspondientes a este usuario en el sistema y este le asigna el rol de usuario (Pantalla 9).
Flujos Alternos	
Pantalla 10	

	4. De no poderse conectar con el dominio, informa al usuario que hay problemas con el Dominio. - Muestra un mensaje informando si desea autenticarse por el sistema (Pantalla 10).
4.1 Oprime el botón "Aceptar" (A)	4.2. Verifica las credenciales consultando los perfiles de usuario almacenados en la base de datos.
	5. De no ser correctas las credenciales informa al usuario que el usuario o contraseñas son incorrectos y debe volver a intentarlo (ir al paso 3).

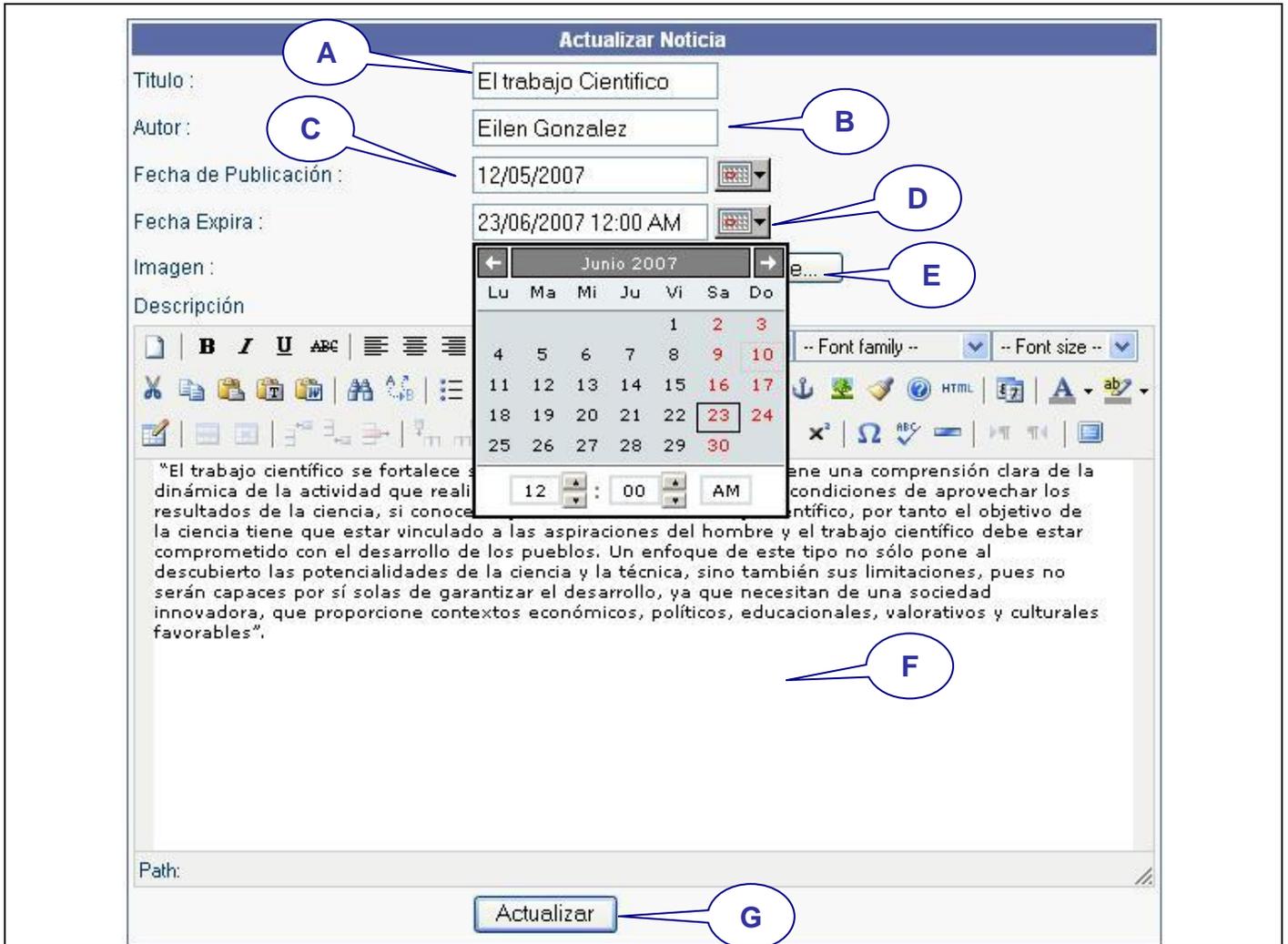
Tabla 14. Gestionar Noticias.

Caso de uso	
CU-4	Gestionar Noticias.
Propósito	Publicar noticias que puedan servir para darle publicidad al grupo, así como otras informaciones de la esfera investigativa que se puedan mostrar a la comunidad.
Actores: Administrador	
Resumen: El caso de uso se inicia cuando uno de los actores concebido como Administrador, inserta una nueva noticia en el sistema, también puede en determinado momento actualizar o eliminar una noticia que haya sido publicada con anterioridad. Al eliminar una noticia se eliminarán todos los comentarios relacionados con la misma.	
Referencias	RF4.1, RF4.2, RF4.3
Precondiciones	Debe estar registrado con el rol de Administrador.



Pantalla 10

Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el Administrador solicita administrar noticias.	2. Muestra el listado de todas las noticias existentes ordenadas por fecha de publicación (Pantalla 10), con las respectivas opciones de eliminar, actualizar e insertar nueva noticia.
3. Si elige Eliminar (A).	4. Elimina la noticia de la base de datos y con ella todos los comentarios asociados.
5. a) Si elige Actualizar (B), ver sesión "Actualizar Noticia". b) Si elige Insertar Nueva Noticia (C), ver sesión "Insertar Nueva Noticia".	
Sesión "Actualizar Noticia".	



Pantalla 11

	<p>1. Muestra el formulario con los datos actuales de la noticia seleccionada (Pantalla11).</p>
<p>2. Actualiza los datos que desee la noticia. Título (A), Autor (B), fecha de publicación (C), fecha en que expira (D), imagen (E), cuerpo de la noticia (F).</p>	<p>3. Actualiza la noticia en la base de datos con los nuevos datos entrados.</p>

-Oprime el botón Actualizar (G).

Sesión "Insertar Nueva Noticia".

The screenshot shows a web form titled "INSERTAR NOTICIA". It contains several input fields and a text area. Callouts A through G point to specific elements: A points to the "Titulo" input field; B points to the "Autor" input field; C points to the "Fechap:" input field which includes a calendar icon; D points to the "Fecha Exp" input field which also includes a calendar icon; E points to the "Browse..." button next to the "imagen" input field; F points to the large text area labeled "Cuerpo:"; and G points to the "Registrar" button at the bottom right of the form. The form also includes a rich text editor toolbar with various icons for text formatting and alignment.

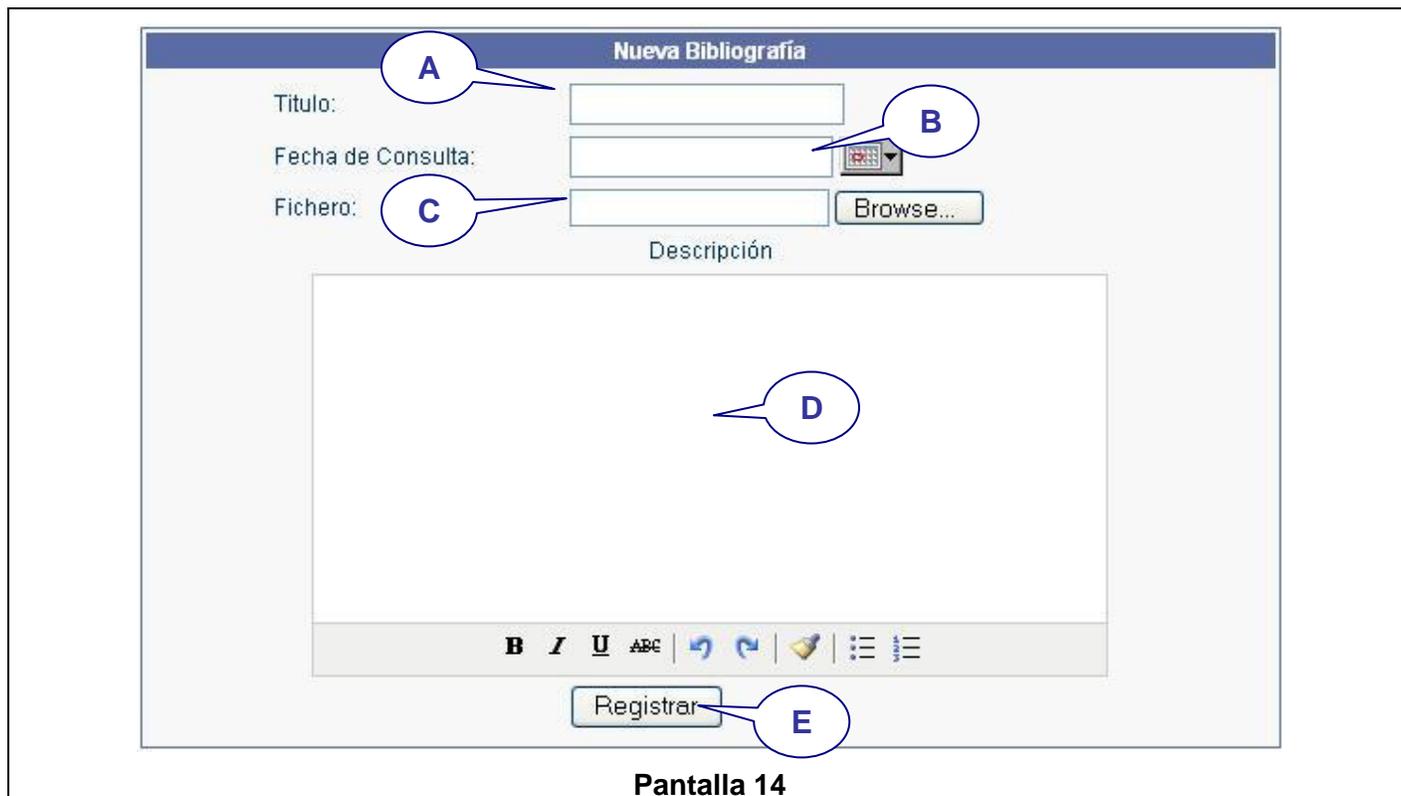
Pantalla 12

	<p>1. Muestra el formulario que permitirá entrar los datos de la nueva noticia (Pantalla 12).</p>
<p>2. Entra los datos de la nueva noticia Título (A), Autor (B), fecha de publicación (C), fecha en que expira (D) que de no ser especificada expirara dos</p>	<p>3. Inserta la noticia en la base de datos e informa al administrador de la correcta inserción o no.</p>

días después de publicada, si tiene imagen (E) la selecciona y por ultimo el cuerpo de la noticia (F). -Una vez llenados los datos oprime el botón Insertar (G).	
---	--

Tabla 15. Gestionar Bibliografía.

Caso de uso	
CU-5	Gestionar Bibliografía.
Propósito	Almacenar las Bibliografías utilizadas en investigaciones ya realizadas y que puedan servir para trabajos posteriores.
Actores: Profesor	
Resumen: El caso de uso se inicia cuando se decide almacenar una bibliografía utilizada en la realización de algunos de los trabajos científicos del Grupo y que pueda ser de utilidad para trabajos posteriores. También se podrá descargar o eliminar una bibliografía ya existente.	
Referencias	RF 7.1, RF 7.2, RF 7.3
Precondiciones	El usuario debe haberse registrado con el rol de profesor o administrador.
<p>Pantalla 13</p>	



Pantalla 14

Acción del actor	Respuesta del sistema
1. El caso de uso comienza cuando el actor selecciona el vínculo Registros Bibliográficos.	2. Muestra todas las Bibliografías registradas ordenadas por fecha (Pantalla 13).
3. Elige eliminar (B) una Bibliografía determinada.	4. Elimina la Bibliografía inmediatamente de la base de datos.
5. Elige descargar una Bibliografía específica y oprime el vínculo de la descripción (A).	6. Descarga la bibliografía del servidor.
7. Oprime el botón Nueva Bibliografía (C)	9. Muestra un formulario permite entrar los datos de dicha bibliografía (Pantalla 14).
10. Entra los datos de la bibliografía, título o título (A), fecha de consultada (B), selecciona el fichero que se subirá al servidor (C), puede añadir una breve descripción de la bibliografía (D). -Una vez llenado estos datos oprime el botón	11. Almacena la nueva bibliografía en la base de datos y sube el fichero al servidor.

Registrar (E).	
----------------	--

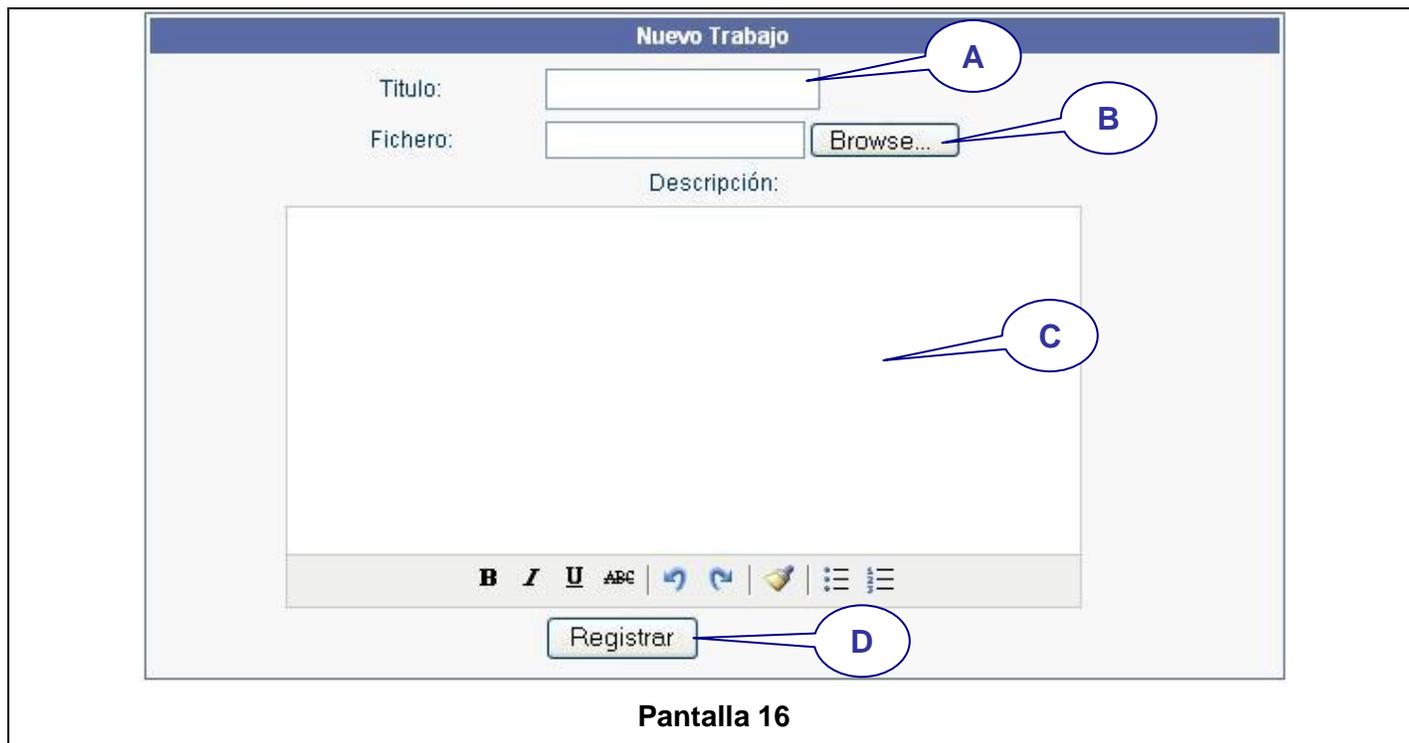
Tabla 16. Gestionar Trabajos Científicos.

Caso de uso	
CU-6	Gestionar trabajos Científicos.
Propósito	Almacenar trabajos investigativos ya realizados por integrantes del grupo a los que tienen acceso cualquier usuario interesado.
Actores: Profesor	
Resumen: Este caso de uso se inicia cuando un profesor después de haber terminado un determinado trabajo investigativo y mostrado a los demás miembros del grupo procede a publicarlo en el sistema para que pueda ser consultado por todo el interesado, también si en algún momento lo desea puede eliminarlo.	
Referencias	RF 8.1, RF 8.2
Precondiciones	El usuario debe haberse registrado con el rol de profesor o administrador.

Nuevo Trabajo			
Trabajo	Autor	Fecha	Eliminar
Geometria	Misael Crespo Padron	Sun, 10 de Jun 2007	
Las Particulas	Reinier Chavez La Rosa	Sun, 10 de Jun 2007	X

Paginas 2 de 2 1 2

Pantalla 15



Pantalla 16

Acción del actor	Respuesta del sistema
1. El caso de uso comienza cuando el actor selecciona el vínculo Trabajos Investigativos.	2. Muestra todos los Trabajos publicados hasta el momento (Pantalla 15).
3. Elige eliminar (A) un Trabajo determinado.	4. Elimina el Trabajo inmediatamente de la base de datos junto al fichero bibliográfico que se encuentra en el servidor.
5. Oprime el botón Nuevo Trabajo (B)	6. Muestra un formulario permite entrar los datos del nuevo trabajo (Pantalla 16).
7. Entra los datos del nuevo trabajo, título (A), selecciona el fichero del trabajo que se subirá al servidor (B), descripción (C). -Una vez llenado estos datos oprime el botón Registrar (D).	8. Almacena el nuevo trabajo en la base de datos y sube al servidor en la dirección adecuada el fichero del trabajo.

Tabla 17. Consultar información pública.

Caso de uso	
CU-7	Consultar información pública
Propósito	Permite a los usuarios acceder a las noticias publicadas, agregarles comentario. Consultar y descargar trabajos realizados por el grupo.
Actores: Usuario	
Resumen: Este caso de uso se inicia cuando un usuario accede a información que se presenta en el sistema, ya sea para descargar algún trabajo que sea de su interés, observar las noticias publicadas con la oportunidad de dar su opinión al respecto, consultar las distintas actividades que se realizarán próximamente en el grupo, para asistir siempre que se pueda y así lo deseen. O simplemente desee navegar teniendo en cuenta que tendrá acceso a la información y recursos definidos para el rol que juega en el sistema.	
Referencias	RF10.1, RF10.2, RF10.2.1, RF10.3
Precondiciones	El usuario debe haberse registrado para poder descargar algún trabajo o consultar las actividades del grupo.
Acción del actor	Respuesta del sistema
1. El caso de uso comienza cuando el usuario accede al sistema para consultar la información que está a su disposición.	2. Muestra las opciones disponibles para los usuarios.
3. a) Si elige Listar Trabajos Investigativos, ver sesión "Listar Trabajos Investigativos". b) Si elige Noticias, ver sesión "Noticias". c) Si desea consultar las actividades que se desarrollaran próximamente accediendo al calendario, ver sesión "Actividad".	
Sesión "Listar Trabajos Investigativos".	

Nuevo Trabajo			
Trabajo	Autor	Fecha	Eliminar
Geometria	Misael Crespo Padron	Sun, 10 de Jun 2007	
Las Particulas	Reinier Chavez La Rosa	Sun, 10 de Jun 2007	X

A

Descripcion

Importante

Pantalla 17

	1. Muestra los trabajos publicados hasta el momento, ordenados por su fecha de publicación (Pantalla 17).
2- Situándose sobre el título del trabajo (A)	3. Muestra una descripción del trabajo.
4. Oprime el vínculo del título (A).	5. Descarga el trabajo del servidor.

Sesión "Noticias".

Resumen: Se mostrará un resumen de las noticias que se encuentran almacenadas en la base de datos siempre que la fecha actual sea mayor que la fecha de publicación y menor que la fecha en que esta expira. El usuario podrá ampliarla si es de su interés, ver y agregar comentarios a una noticia determinada.

Reunion de brigada	rey
 <p>El acto conmemorativo de la Universidad de las Ciencias Informáticas por el 46 Aniversario del Ministerio del Interior fue celebrado en la mañana de este miércoles en la histórica Plaza Mella.</p> <p>Con la presencia ... Ampliar</p> <p>Por: misael el Thu, 7 de Jun 2007</p> <p><input type="checkbox"/> Comentarios:5</p>	 <p>Puro humor criollo, picardía, reflexiones y confesiones, matizaron la velada de la noche de este miércoles. En el lobby del docente cuatro confluyeron las vivencias de un numeroso grupo de trabajadores del diario Juventud Rebelde, ... Ampliar</p> <p>Por: xcxcxc el Thu, 7 de Jun 2007</p> <p><input type="checkbox"/> Comentarios:0</p>

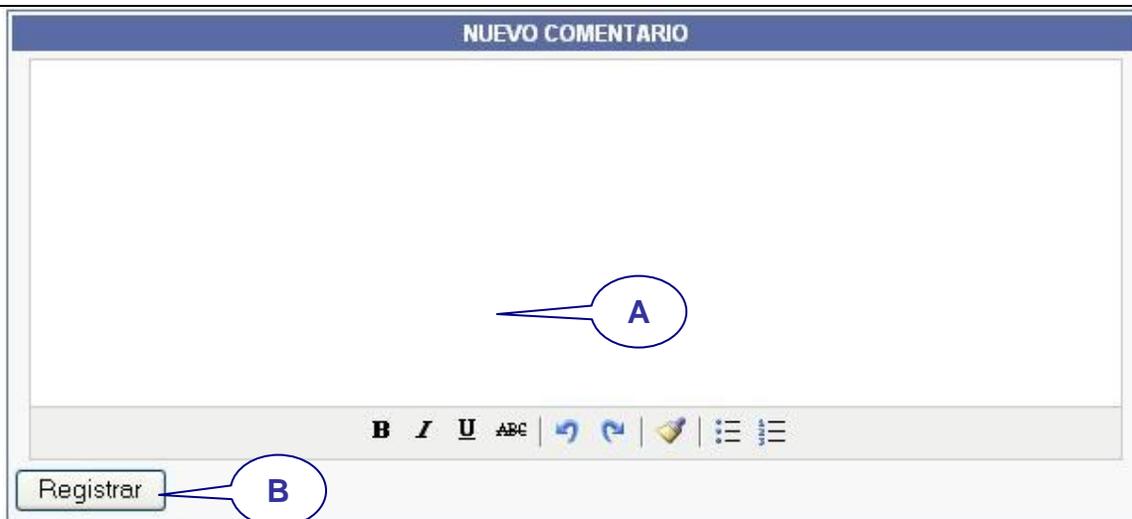
A

B

Pantalla 18



Pantalla 19



Pantalla 20

	<p>1. Muestra un resumen de todas las noticias que están en fecha de publicación (Pantalla 18).</p> <p>- Muestra también la cantidad de comentarios existentes de cada noticia (B).</p>
2. Oprime Ampliar noticia (A).	3. Muestra la noticia completa.
4. Oprime Comentarios (B).	5. Muestra la noticia completa y todos los comentarios asociados a la misma (Pantalla 19).
6. Oprime Insertar Nuevo Comentario (A)	7. Muestra un editor donde se introducirá el nuevo comentario (Pantalla 20).
8. Introduce el texto del comentario (A).	9. Inserta el nuevo comentario correspondiente a la noticia seleccionada.
- Oprime el botón Registrar (B).	

Sesión “Actividades”.



2. Selecciona ver las actividades de un día específico (A) (Pantalla 21). Situándose sobre el calendario.	3. Despliega automáticamente una lista con las actividades de ese día.
4. Selecciona una actividad haciendo clic sobre ella (B)	4. Muestra la actividad completa.

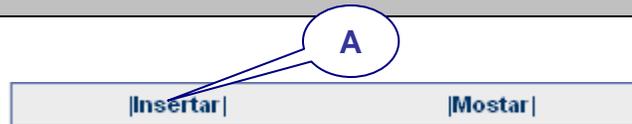
Tabla 18. Gestionar Usuarios del Sistema.

Caso de uso	
CU-8	Gestionar Usuarios del Sistema.
Propósito	Agregar o eliminar los Usuarios del sistema.
Actores: Administrador	
Resumen: El caso de uso se inicia cuando el actor administrador accede a la sección de Gestionar Usuarios, este selecciona alguna de las opciones disponibles, como agregar, eliminar Usuarios o modificar alguno de sus datos.	
Referencias	RF2.1, RF2.2, RF2.3, RF2.4
Precondiciones	El usuario pertenece al grupo de Administradores.
Acción del actor	Respuesta del sistema
1. Accede al menú de gestionar usuarios desde la página de administración del sitio.	2. Muestra las opciones disponibles.

3. Escoge una de las opciones siguientes:

- a) Agregar nuevo Usuario.
- b) Eliminar Usuario.
- c) Actualizar Usuario.

Sección: "Agregar nuevo Usuario"



Pantalla 22



Pantalla 23

The screenshot shows a registration form with the following fields and callouts:

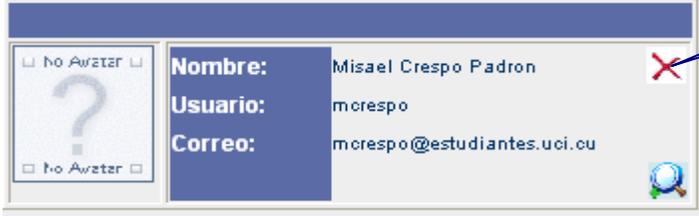
- A:** Points to the 'Nombre y Apellidos' field containing 'Misael Crespo Padron'.
- B:** Points to the 'Usuario' field containing 'm Crespo'.
- C:** Points to the 'Correo' field containing 'm Crespo@estudiantes.uci.cu'.
- D:** Points to the 'Intitucion Donde labora' field.
- E:** Points to the 'Responsabilidad administrativa(Cargo)' field.
- F:** Points to the 'Linea investigativa' dropdown menu showing '<--- Seleccionar --->'.
- G:** Points to the 'Foto' field with a 'Browse...' button.
- H:** Points to the 'Categoria Docente/Investigativa' field.
- I:** Points to the 'Grado Cientifico' field.
- J:** Points to the 'Rol' dropdown menu showing 'Profesor'.
- K:** Points to the 'Curriculun' text area.
- L:** Points to the 'Insertar' button at the bottom of the form.

Pantalla 24

<p>1. El caso de uso comienza cuando el actor selecciona el vínculo de insertar un nuevo usuario al grupo (Pantalla 22).</p>	<p>2. El sistema muestra el formulario para insertar el nuevo usuario (Pantalla 23).</p>
<p>3. Introduce el nombre del usuario del Dominio UCI de la persona que quiere agregar (A). - Oprime el botón verificar usuario (B)</p>	<p>4. Busca el usuario en el Dominio UCI. - Muestra todos los datos del nuevo miembro, extraídos de la base de datos de Dominio, para ser insertados en el Sistema (Pantalla 24).</p>
<p>5. Introduce la institución donde labora (A), Responsabilidad que desempeña (cargo) (B), Línea de investigación a la que pertenecerá (C), y la foto que lo identificará (D), categoría docente –</p>	<p>6. Inserta los datos del nuevo miembro en el sistema.</p>

<p>investigativa(E), Grado científico(F), Selecciona el rol del nuevo miembro(G), Introduce el currículum del nuevo miembro(H)</p> <p>Oprime el botón Insertar (I).</p> <p>-</p>	
--	--

Sección: “Eliminar Usuario”



Pantalla 25

1. Selecciona el vínculo miembros.	2. Muestra el listado de todos los miembros del grupo (Pantalla 25).
------------------------------------	--

3. Oprime eliminar el botón eliminar del que desee (A).	4. Elimina el usuario de la base de datos.
---	--

Sección: “Actualizar Datos de Miembros”



Pantalla 26

1-El caso de uso comienza cuando el actor decide actualizar los datos de un miembro dado del grupo.	2-El sistema muestra una ficha de usuario
3- El actor presiona el campo que desea actualizar (A)	4-El sistema muestra un campo de texto para poder introducir el nuevo dato
5- El actor introduce los datos correspondientes , y sale del campo.	5- El sistema actualiza el o los datos.
<p>Sección: “Actualizar Datos de Miembros”</p> <p>Escenario actualizar rol</p>	
 <p>Pantalla 27</p>	
1-El caso de uso comienza cuando el actor decide actualizar los privilegios de un miembro de grupo	2-El sistema muestra los usuarios del grupo y sus respectivos roles
3-El actor selecciona el usuario al cual se editara	4 El sistema muestra los roles disponibles (B)

el rol (A)	
5- El actor escoge el nuevo rol	5 El sistema actualiza el rol

Conclusiones Parciales

El desarrollo de este capítulo ha permitido una mejor comprensión del contexto a automatizar. Se realizó una descripción de la solución propuesta y se definieron los requisitos funcionales y no funcionales que debe cumplir. Se elaboró el diagrama de casos de uso del sistema, donde se representa cada actor y su relación con cada uno de éstos. Fue elaborada una descripción de todos los casos de uso y posteriormente expandidos. Con el desarrollo de este flujo de trabajo y los artefactos obtenidos a partir de este, se puede pasar al flujo de diseño para comenzar la construcción de la solución de software propuesta, el cual será presentado en el próximo capítulo.

CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

Introducción.

El presente capítulo permite definir una arquitectura candidata del sistema, representar una estructura global del mismo mediante el Modelo de Análisis así como la modelación de los artefactos necesarios para su construcción. Los componentes de la aplicación son tratados como clases y mediante la utilización de UML se podrán representar a través de diagramas de clases con esterotipo Web. Se presenta el modelo de datos que es la base para construir finalmente la base de datos que soportará el trabajo del sistema y el modelo de despliegue donde se representan los nodos en los que se distribuye la aplicación.

3.1. Modelo de Análisis

Este modelo es usado para representar la estructura global del sistema, describe la realización de casos de uso, sirve como una abstracción del Modelo de Diseño y se centra en los requerimientos funcionales. La representación de los diagramas del Modelo de Análisis se muestra en el (Anexo I)

3.2. Diseño.

El diseño tiene el propósito de adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos y tecnologías de interfaz de usuario. Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.

3.2.1. Diagrama de clases persistente.

Este diagrama es una especie de modelo lógico de la base de datos pero contiene las clases persistentes, llámese clases persistentes a las tablas que permanecen en la base de datos. En la figura se muestran las clases persistentes del sistema.

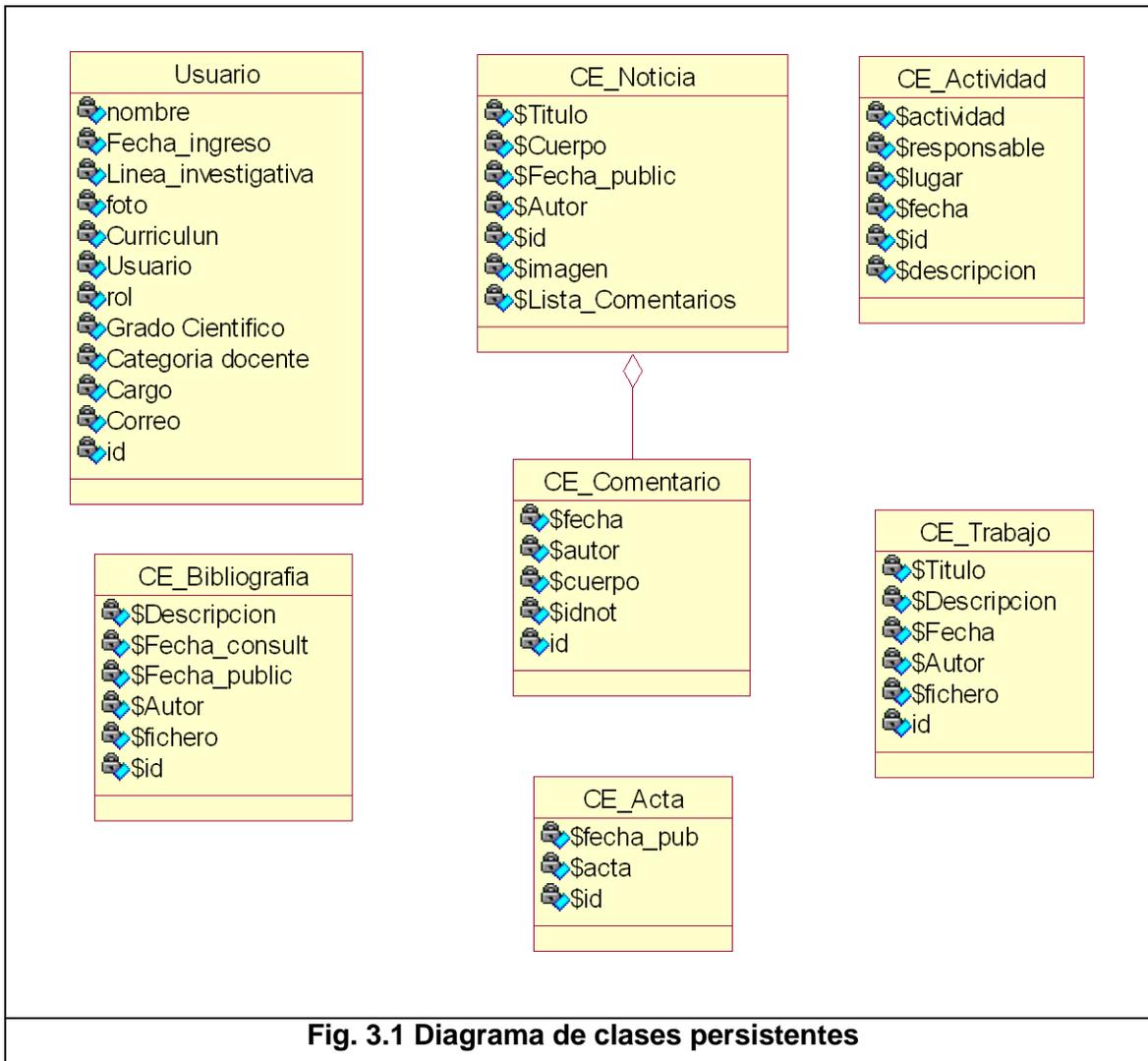


Fig. 3.1 Diagrama de clases persistentes

3.2.2. Diagrama de Clases Web.

El diagrama de clases para las Aplicaciones Web difiere un poco del resto de las aplicaciones que se acostumbra a construir puesto que en ellas son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación. Para obtener un nivel correcto de abstracción y detalle que permita obtener un resultado final es mejor modelar los artefactos del sistema es decir: modelar las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido

dinámico de estas, una vez que estén en el navegador del cliente; estos son los artefactos que se necesitan modelar para que el desarrollador los implemente luego y obtener así el producto final.

Se elaboró un diagrama de clases Web para cada caso de uso de forma tal que se facilite la comprensión de cómo se relacionan los distintos elementos en la realización de cada uno de ellos, ver (Anexo II).

3.2.3. Diagramas de Interacción.

Otro objetivo de esta etapa es la elaboración de los diagramas de interacción que muestran gráficamente como lo objetos se comunican entre si. Esta interacción se puede expresar mediante diagramas de colaboración o diagramas de secuencia. Estos últimos son los utilizados en el presente trabajo de tesis para detallar las secuencias de interacciones ordenadas en el tiempo de los objetos, dichos diagramas se pueden encontrar en el (Anexo III).

3.2.4. Descripción de las clases de diseño.

La descripción de las clases brinda un mayor nivel de detalles de las clases controladoras y entidades con sus atributos y métodos, presentes en el diseño del sistema.

Clases controladoras.

Nombre: CC_GestionarActas	
Tipo de clase (controladora)	
Atributo	Tipo
\$RegistroActa	CE_RegistroActas
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CC_GestionarActas inicializando el objeto \$ RegistroActa.
Nombre:	Adicionar_Acta(\$Acta)
Descripción:	Pasa los datos del acta al registro para que este se encargue de insertarla.
Nombre:	Mostrar_Actas()
Descripción:	Retorna todas las actas almacenadas en la base de datos.
Nombre:	Mostrar_Acta(\$id)
Descripción:	Retorna el acta que se corresponda con el identificador pasado como parámetro.
Nombre:	Eliminar_Acta(\$id)
Descripción:	Pasa el identificador del acta a eliminar al registro para que este proceda a realizar dicha operación.

Nombre: CC_ControladoraActividad	
Tipo de clase (controladora)	
Atributo	Tipo
\$RegistroActividad	CE_RegistroActividad
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CC_GestionarActividad inicializando el objeto \$RegistroActividad.
Nombre:	Adicionar_Activ(\$activ, \$responsable, \$lugar, \$fecha, \$descripcion)
Descripción:	Pasa los datos de la actividad al registro para que este se encargue de insertarla.
Nombre:	Mostrar_Activ(\$id)
Descripción:	Retorna la actividad que se corresponda con el identificador pasado como parámetro.
Nombre:	Mostrar_Fechas()
Descripción:	Retorna un listado de los días (fecha) que hay actividades en la base de datos.
Nombre:	Actualizar_Activ(\$actividad, \$responsable, \$lugar, \$fecha, \$descripcion, \$id)
Descripción:	Pasa los nuevos datos y el identificador de la actividad a actualizar al registro y este se encarga de dicha operación.

Nombre: CC_GestionarBibliografía	
Tipo de clase (controladora)	
Atributo	Tipo
\$RegistroBibliografia	CE_RegistroBibliografia
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CC_GestionarBibliografía inicializando el objeto \$RegistroBibliografia.
Nombre:	Adicionar_Bibliog(\$Autor, \$Fecha_c, \$fichero, \$Descripcion)
Descripción:	Pasa los datos de la bibliografía a insertar al registro que se encargará de esta operación y sube el fichero de la bibliografía al servidor.
Nombre:	Mostrar_Bibliog()
Descripción:	Retorna un listado de todas las bibliografías almacenadas en la base de datos.
Nombre:	Eliminar_Bibliog(\$id)
Descripción:	Pasa el identificador de la bibliografía a eliminar al registro encargado de esta operación y elimina el fichero bibliográfico almacenado en el servidor.

Nombre: CC_GestoraNoticias	
Tipo de clase (controladora)	
Atributo	Tipo
\$RegistroNoticia	CE_RegistroNoticias
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CC_GestoraNoticias inicializando el objeto \$RegistroNoticia.
Nombre:	Adicionar_Not(\$Titulo, \$Autor, \$Cuerpo, \$fecha_p, \$fecha_e, \$imagen)
Descripción:	Pasa los datos la noticia a insertar al registro que se encargara de esta operación, si la noticia tiene asociada una imagen este la sube al servidor en su carpeta correspondiente.
Nombre:	Adicionar_Comentario(\$Cuerpo, \$idnot)
Descripción:	Pasa los datos del comentario a la entidad noticia que se encargará de adicionarlo a su lista de comentarios.
Nombre:	Mostrar_Resumenes()
Descripción:	Retorna un listado con los resúmenes de las noticias existentes en la base de datos.
Nombre:	Actualizar_Noticia(\$Titulo, \$Autor, \$Cuerpo, \$Fecha_p, \$fecha_e, \$imagen, \$id)
Descripción:	Pasa los nuevos datos y el identificador de la noticia a actualizar al registro y este se encarga de dicha operación.
Nombre:	Mostrar_Noticia_Act(\$id)
Descripción:	Retorna la noticia que se corresponda con el identificador pasado como parámetro, para ser actualizada.
Nombre:	Eliminar_Noticia(\$id)
Descripción:	Elimina la noticia que se corresponda con el identificador pasado como parámetro y todos comentarios asociados a ella, si la noticia tiene una imagen asociada también es eliminada del servidor.
Nombre:	Mostrar_comentarios(\$idnot)
Descripción:	Retorna un listado de todos los comentarios al identificador de la noticia pasado como parámetro.

Nombre: CC_GestionarTrabajos	
Tipo de clase (controladora)	
Atributo	Tipo
\$RegistroTrabajo	CE_RegistroTrabajo
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CC_GestionarTrabajo inicializando el objeto \$RegistroTrabajo.

Nombre:	Adicionar_Trabajo(\$Titulo, \$Autor, \$fichero, \$Descripcion)
Descripción:	Pasa los datos del trabajo a insertar al registro que se encargará de esta operación y sube el fichero de la bibliografía al servidor.
Nombre:	Mostrar_Trabajo()
Descripción:	Retorna un listado con todos los trabajos almacenados en la base de datos.
Nombre:	Eliminar_Trabajo()
Descripción:	Pasa el identificador del trabajo a eliminar al registro encargado de esta operación y elimina el fichero del trabajo almacenado en el servidor.

Nombre: CC_Gestion_Usuarios	
Tipo de clase (controladora)	
Atributo	Tipo
\$RegistroUsuarios	CE_Registro_Usuarios
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CC_Gestion_Usuarios inicializando el objeto \$RegistroUsuarios.
Nombre:	Adicionar_Usuario(\$nombre, \$categoria, \$grado, \$intitucion,\$correo, \$cargo,\$curriculun,\$linea,\$usuario,\$rol,\$foto)
Descripción:	Pasa los datos del usuario a insertar al registro que se encargará de esta operación y sube el fichero de la foto al servidor.
Nombre:	Mostrar_Miembros()
Descripción:	Retorna un listado con todos los miembros del grupo de investigaciones
Nombre:	Eliminar_Miembro()
Descripción:	Pasa el identificador del miembro a eliminar al registro encargado de esta operación y elimina el fichero de la foto almacenado en el servidor.
Nombre	Actualizar_Dato()
Descripción:	Pasa el identificador del miembro y dato que vas a actualizar registro encargado de esta operación y actualiza el dato
Nombre	Mostrar_Usuario()
Descripción:	Retorna el listado de los miembros del grupo mostrando su usuario y rol
Nombre	Cambiar_Rol()
Descripción	Pasa el identificador del miembro y el nuevo rol al registro encargado de esta operación y actualiza el rol.
Nombre	Buscar()
Descripción	Pasa un nombre de usuario del dominio uci al registro encargado de hacer esta operación

Nombre: CC_Login	
Tipo de clase (controladora)	
Atributo	Tipo
\$autenticacion	CE_Autenticarse
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CC_Login inicializando el objeto \$dominio
Nombre:	loguearse(\$usuario,\$contraseña)
Descripción:	
Nombre:	Construir_sesion(\$usuario)
Descripción:	

Clases entidades.

Nombre: CE_Autenticarse	
Tipo de clase (entidad)	
Atributo	Tipo
\$Busuario	PDO
\$dominio	Ldap
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CE_Autenticarse inicializando el objeto \$BUusuario y \$dominio.
Nombre:	login(\$usuario,\$contraseña)
Descripción:	Se autentifica utilizando la clase Ldap buscando los usuarios en el dominio
Nombre:	Construir_sesion(\$usuario)
Descripción:	Busca el usuario en la bd y retorna una sesión con su rol

Nombre: CE_Idap	
Tipo de clase (entidad)	
Atributo	Tipo
\$servidor	String
\$usuario	String
\$contrasena	String
Para cada responsabilidad:	
Nombre:	__construct(\$usuario, Servidor,\$contrasena)
Descripción:	Constructor de la clase CE_Idap con sus atributos
Nombre:	login(\$usuario,\$contraseña)
Descripción:	Se autentifica utilizando la clase Ldap buscando los usuarios en dominio
Nombre:	Buscar_usuario(\$usuario)

Descripción:	Busca el usuario en el dominio Uci y devuelve sus datos
--------------	---

Nombre: CE_RegistroActas	
Tipo de clase (entidad)	
Atributo	Tipo
\$BDacta	PDO
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CE_RegistroActas inicializando el objeto \$BDacta.
Nombre:	Insertar_Acta(\$Acta)
Descripción:	Inserta un nuevo trabajo científico en la base de datos con los parámetros pasados.
Nombre:	Muestra_Actas()
Descripción:	Retorna un listado con todas las actas almacenadas en la base de datos.
Nombre:	Elimina_Acta(\$id)
Descripción:	Elimina el acta que se corresponda con el identificador pasado como parámetro, de la base de datos.

Nombre: CE_RegistroUsuarios	
Tipo de clase (entidad)	
Atributo	Tipo
\$BDusuarios	PDO
\$dominio	Ldap
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CE_RegistroActas inicializando el objeto \$BDUsuarios y \$dominio.
Nombre:	Insertar_Usuario(\$Usuario)
Descripción:	Insertar un nuevo usuario en la base de datos con los parámetros pasados.
Nombre:	Muestra_Datos()
Descripción:	Retorna un listado con todos los usuarios almacenados en la base de datos.
Nombre:	Elimina_usuario(\$id)
Descripción:	Elimina el usuario que se corresponda con el identificador pasado como parámetro, de la base de datos.
Nombre:	Buscar_usuario(\$usuario)
Descripción:	Busca el nombre completo y correo de la persona a que pertenezca ese usuario
Nombre:	Actualizar_dato(\$id,\$nuevo_dato)
Descripción:	Le actualiza el dato al usuario que pertenezca dicho id
nombre:	Actualizar_rol((\$id,\$nuevo_rol)
	Le actualiza el rol al usuario que pertenezca dicho id

Nombre: CE_RegistroActividad	
Tipo de clase (entidad)	
Atributo	Tipo
\$BDactividad	PDO
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CE_RegistroActividad inicializando el objeto \$BDactividad.
Nombre:	Formatear_Fecha(\$fecha)
Descripción:	Le da el formato adecuado a la fecha pasada como parámetro para ser almacenada en la base de datos.
Nombre:	Adicionar(\$activ, \$responsable, \$lugar, \$fecha, \$descripcion)
Descripción:	Inserta la nueva actividad en la base de datos con los parámetros pasados.
Nombre:	Devuelve_Actividades(\$id)
Descripción:	Retorna la actividad que se corresponda en la base de datos con el identificador pasado como parámetro.
Nombre:	Devuelve_Fechas()
Descripción:	Retorna un listado de los días (fecha) que hay actividades en la base de datos.
Nombre:	Actualizar(\$actividad, \$responsable, \$lugar, \$fecha, \$descripcion, \$id)
Descripción:	Actualiza en la base de datos la actividad que se corresponda con el identificador pasado como parámetro.

Nombre: CE_RegistroBibliografia	
Tipo de clase (entidad)	
Atributo	Tipo
\$BDBibliografia	PDO
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CE_RegistroBibliografia inicializando el objeto \$BDBibliografia.
Nombre:	Adicionar_Bibliog(\$Autor, \$Fecha_c, \$fichero, \$Descripcion)
Descripción:	Inserta la nueva bibliografía en la base de datos con los parámetros pasados.
Nombre:	Mostrar_Bibliog()
Descripción:	Retorna un listado con las bibliografías almacenadas en la base de datos.
Nombre:	Elimina_Bibliografia(\$id)
Descripción:	Elimina la bibliografía que se corresponda con el identificador pasado como parámetro, de la base de datos.

Nombre: CE_RegistroNoticias	
Tipo de clase (entidad)	
Atributo	Tipo
\$BDnoticia	PDO
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CE_RegistroNoticias inicializando el objeto \$BDnoticia.
Nombre:	cambia_a_mysql(\$fecha)
Descripción:	Le da el formato adecuado a la fecha pasada como parámetro para ser almacenada en la base de datos.
Nombre:	Insertar_Noticia(\$Titulo, \$Autor, \$Cuerpo, \$fecha_pub, \$fecha_exp, \$imagen)
Descripción:	Inserta la nueva noticia en la base de datos con los parámetros pasados.
Nombre:	Devuelve_Resumen_Noticias()
Descripción:	Retorna un listado con los resúmenes de las noticias almacenadas en la base de datos.
Nombre:	Devuelve_Noticia(\$id)
Descripción:	Retorna la noticia que se corresponda con el identificador pasado como parámetro.
Nombre:	Noticia_a_Actualizar(\$id)
Descripción:	Retorna la noticia que se corresponda con el identificador pasado como parámetro, para su posterior actualización.
Nombre:	Actualizar_Noticia(\$Titulo, \$Autor, \$Cuerpo, \$Fecha_p, \$fecha_e, \$imagen, \$id)
Descripción:	Actualiza en la base de datos la noticia que se corresponda con el identificador pasado como parámetro.
Nombre:	Elimina_Noticia(\$id)
Descripción:	Elimina la noticia que se corresponda con el identificador pasado como parámetro, de la base de datos.

Nombre: CE_RegistroTrabajo	
Tipo de clase (entidad)	
Atributo	Tipo
\$BDTrabajo	PDO
Para cada responsabilidad:	
Nombre:	__construct()
Descripción:	Constructor de la clase CE_RegistroTrabajo inicializando el objeto \$BDTrabajo.
Nombre:	Adicionar_Trabajo(\$Titulo, \$Autor, \$fichero, \$Descripcion)
Descripción:	Inserta el nuevo trabajo en la base de datos con los parámetros pasados.

Nombre:	Mostrar_Trabajo()
Descripción:	Retorna un listado con todos los trabajos almacenados en la base de datos.
Nombre:	Elimina_Trabajo(\$id)
Descripción:	Elimina el trabajo que se corresponda con el identificador pasado como parámetro, de la base de datos.

Nombre: CE_Acta	
Tipo de clase (entidad)	
Atributo	Tipo
\$fecha_pub	String
\$acta	String
\$id	Int
Para cada responsabilidad:	
Nombre:	__construct(\$acta, \$fecha_pub, \$id)
Descripción:	Constructor de la clase CE_Acta inicializando los atributos del objeto CE_Acta.
Nombre:	Get_Fecha()
Descripción:	Retorna la fecha en que fue subida el acta.
Nombre:	Get_Acta()
Descripción:	Retorna el cuerpo del acta.
Nombre:	Get_Id()
Descripción:	Retorna el id del acta.

Nombre: CE_Actividad	
Tipo de clase (entidad)	
Atributo	Tipo
\$actividad	String
\$responsable	String
\$lugar	String
\$fecha	String
\$descripcion	String
\$id	Int
Para cada responsabilidad:	
Nombre:	__construct(\$actividad, \$responsable, \$lugar, \$fecha_cump, \$descripcion)
Descripción:	Constructor de la clase CE_Actividad inicializando los atributos del objeto CE_Actividad.
Nombre:	Get_Activ()
Descripción:	Retorna la actividad.
Nombre:	Get_Responsable()

Descripción:	Retorna el responsable de la actividad.
Nombre:	Get_Lugar()
Descripción:	Retorna el lugar donde se dará cumplimiento a la actividad.
Nombre:	Get_Fecha()
Descripción:	Retorna fecha y hora del cumplimiento de la actividad.
Nombre:	Get_Id()
Descripción:	Retorna el identificador de la actividad.
Nombre:	Get_Descripcion()
Descripción:	Retorna la descripción detallada de la actividad.

Nombre: CE_Bibliografia	
Tipo de clase (entidad)	
Atributo	Tipo
\$Descripcion	String
\$Fecha_consultada	String
\$Fecha_publicada	String
\$Autor	String
\$fichero	String
\$id	Int
Para cada responsabilidad:	
Nombre:	__construct(\$Autor, \$Fecha_consult, \$fichero, \$Descripcion)
Descripción:	Constructor de la clase CE_Bibliografia inicializando los atributos del objeto CE_Bibliografia.
Nombre:	Get_Descripcion()
Descripción:	Retorna la descripción.
Nombre:	Get_Autor()
Descripción:	Retorna el autor.
Nombre:	Get_Fecha_public()
Descripción:	Retorna la fecha de publicación de la bibliografía.
Nombre:	Get_Fecha_Cons()
Descripción:	Retorna fecha de consultada de la bibliografía
Nombre:	Get_Id()
Descripción:	Retorna el identificador de la bibliografía.
Nombre:	Get_Fichero()
Descripción:	Retorna la dirección del fichero bibliográfico en el servidor.

Nombre: CE_Noticia	
Tipo de clase (entidad)	
Atributo	Tipo
\$Titulo	String
\$Cuerpo	String

\$Fecha_public	String
\$Autor	String
\$imagen	String
\$id	Int
\$Lista_Comentarios	PDO
Para cada responsabilidad:	
Nombre:	__construct(\$Titulo, \$Autor, \$Cuerpo, \$id, \$fecha_pub, \$fecha_exp, \$imagen)
Descripción:	Constructor de la clase CE_Noticia inicializando los atributos del objeto CE_Noticia.
Nombre:	Get_Titulo()
Descripción:	Retorna el titulo.
Nombre:	Get_Cuerpo()
Descripción:	Retorna el cuerpo de la noticia.
Nombre:	Get_Autor()
Descripción:	Retorna el autor de la noticia.
Nombre:	Get_Fecha_public()
Descripción:	Retorna la fecha de publicación.
Nombre:	Get_Id()
Descripción:	Retorna el identificador de la noticia.
Nombre:	Get_Imagen()
Descripción:	Retorna la dirección de la imagen en el servidor.
Nombre:	Adicionar_Comentario(\$Cuerpo_Comentario, \$idnot)
Descripción:	Inserta el nuevo comentario.
Nombre:	Devolver_Comentarios()
Descripción:	Retorna todos los comentarios asociados a una noticia.
Nombre:	Eliminar_Comentarios(\$idnot)
descripción:	Elimina todos los comentarios asociados a la noticia de la cual se pasa el identificador por parámetros.

Nombre: CE_Trabajo	
Tipo de clase (entidad)	
Atributo	Tipo
\$Titulo	String
\$Descripcion	String
\$Fecha	String
\$Autor	String
\$fichero	String
\$id	Int
Para cada responsabilidad:	
Nombre:	__construct(\$Titulo, \$Autor, \$fichero, \$Descripcion)
Descripción:	Constructor de la clase CE_Trabajo inicializando los atributos del objeto CE_Trabajo.
Nombre:	Get_Titulo()

Descripción:	Retorna el título.
Nombre:	Get_Autor()
Descripción:	Retorna el autor.
Nombre:	Get_Descripcion()
Descripción:	Retorna la descripción.
Nombre:	Get_Fecha()
Descripción:	Retorna la fecha de publicado el trabajo.
Nombre:	Get_Id()
Descripción:	Retorna el identificador del trabajo.
Nombre:	Get_Fichero()
Descripción:	Retorna la dirección del fichero en el servidor.

Nombre: CE_usuario	
Tipo de clase (entidad)	
Atributo	Tipo
\$Nombre	String
\$Correo	String
\$Rol	String
\$Cargo	String
\$Grado_cientifico	String
\$Categoria_Docente	String
\$id	int
\$Foto	String
\$curriculum	String
\$linea_Investigativa	String
\$usuario	String
\$institucion_donde_labora	String
Para cada responsabilidad:	
Nombre:	__construct(\$nombre, \$categoria, \$grado, \$intitucion,\$correo, \$cargo,\$curriculum,\$linea,\$usuario,\$rol,\$foto,\$id)
Descripción:	Constructor de la clase CE_Usuario inicializando los atributos del objeto CE_Usuario.
Nombre:	Get_nombre()
Descripción:	Retorna el nombre.
Nombre:	Get_categoria()
Descripción:	Retorna la categoría.
Nombre:	Get_grado()
Descripción:	Retorna el grado científico.
Nombre:	Get_intitucion()
Descripción:	Retorna la institución a que pertenece.
Nombre:	Get_Id()
Descripción:	Retorna el identificador del usuario.
Nombre:	Get_Imagen()

Descripción:	Retorna la dirección la foto en el servidor.
Nombre:	Get_correo()
Descripción:	Retorna el correo del miembro
Nombre	Get_cargo()
Descripción:	Retorna el cargo del miembro en el grupo
Nombre	Get_Curriculum()
Descripción:	Retorna el currículo del miembro
Nombre	Get_linea()
Descripción:	Retorna la línea investigativa ala que pertenece
Nombre	Get_usuario()
Descripción:	Retorna el nombre usuario del miembro
Nombre	Get_rol()
Descripción:	Retorna el rol del miembro del departamento

Nombre: CE Comentario	
Tipo de clase (entidad)	
Atributo	Tipo
\$fecha	String
\$autor	String
\$cuerpo	String
\$idnot	Int
Para cada responsabilidad:	
Nombre:	__construct(\$cuerpo)
Descripción:	Constructor de la clase CE_Comentario inicializando los atributos del objeto CE_Comentario.
Nombre:	Get_Autor()
Descripción:	Retorna el autor.
Nombre:	Get_Fecha()
Descripción:	Retorna la fecha de publicado el comentario.
Nombre:	Get_Cuerpo()
Descripción:	Retorna el cuerpo del comentario.
Nombre:	Get_Idnot()
Descripción:	Retorna el identificador de la noticia a la que se le asociará el comentario.

Nombre: Fichero	
Tipo de clase (entidad)	
Atributo	Tipo
\$Nomb_fich	String
\$tipo_fich	String
\$tamano_fich	Int
\$DirOrigen	String
\$DirDestino	String

Para cada responsabilidad:	
Nombre:	__construct(\$Nomb_fich, \$tipo_fich, \$tamano_fich, \$DirOrigen)
Descripción:	Constructor de la clase Fichero inicializando los atributos del objeto Fichero.
Nombre:	Upload_Img()
Descripción:	Sube una imagen al servidor.
Nombre:	Upload_Fich()
Descripción:	Sube un fichero al servidor con un tamaño máximo de 15 MB
Nombre:	Delete_Server_Fich(\$direccion)
Descripción:	Elimina un fichero del servidor pasándole la dirección y devuelve true caso de haberse eliminado con éxito, de lo contrario devuelve false.

3.2.7. Principios de diseños.

Estándares en la interfaz de la aplicación:

La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso siendo la consistencia de su interfaz de usuario uno de los aspectos más relevantes de la usabilidad del sistema. El tipo de letra predominante es Times New Roman de tamaño 9. La carga visual es adecuada y el lenguaje de las opciones que se ha utilizado es de fácil comprensión para el usuario. El sistema posee un menú de usuarios que brinda acceso rápido a parte de las opciones.

Las páginas están concebidas para ser perfectamente adaptables a las resoluciones de 1024 x 768 píxeles dado que la mayoría de las computadoras modernas vienen equipadas para trabajar en ella y que esta resolución permite un mejor aprovechamiento del espacio en pantalla, sin disminuir mucho el tamaño de las letras, también se garantiza que el contenido de la información sea perfectamente visible en la resolución de 800 x 600 aunque no sea la más usada en la actualidad.

La interfaz es legible y con colores (bordes de tablas en azul y rellenos en gris) agradables y poco llamativos para no perder concentración. Las páginas Web presentan un formato Standard y sencillo. El diseño de la interfaz está vinculado con la temática que se está exponiendo y está orientado para que el usuario sepa dónde está y qué puede hacer en cada momento.

Tratamiento de excepciones y seguridad del Sistema:

El tratamiento de los errores es un paso fundamental para el buen funcionamiento de un sistema, garantizando la armonía y facilidad de uso de la aplicación.

Para las operaciones de insertar eliminar y/o modificar los registros en la base de datos, se establecerán mecanismos que visualizarán la información para evitar en la medida de lo posible la introducción manual de los datos y minimizar los errores que pueda tener la aplicación, siendo válida toda la información. En el caso de la inserción se implementarán funciones que validarán la entrada de datos y en caso de haber errores, se mostrarán mensajes indicando los errores cometidos en la inserción en el caso de eliminar se implementan funciones que validan todos los datos, luego le da la opción de eliminar, teniendo en cuenta que un miembro del grupo podrá eliminar determinado documento solo si fue subido por él al sistema.

La contraseña es encriptada a través de una función md5 antes de ser almacenada en el sistema para mayor seguridad.

Otros errores pueden ser generados por el Gestor de base de datos, estos se capturan antes de mostrarse al usuario y una vez tratados por el sistema, son mostrados al usuario en un lenguaje afable.

También se debe tener en cuenta que un miembro del grupo de investigaciones de matemática y física, como parte de la seguridad solo podrá acceder a la información que se corresponda con el rol en el sistema.

3.2.8. Uso de capa de abstracción a datos.

En el presente trabajo se hace uso de la capa de abstracción a datos PHP Data Objects (PDO), con el objetivo de abstraer la aplicación del gestor de base de datos utilizado. De esta forma si por determinados motivos se desea cambiar de gestor de base de datos, no sería necesario hacer modificaciones en la capa de datos, solo indicarle el nuevo driver a utilizar, con esto se logra un mayor rendimiento, flexibilidad, simplicidad en la aplicación.

Entre las ventajas que trae consigo el uso de PDO como capa de abstracción a datos se puede mencionar que viene instalado y activado por defecto en PHP5, es una extensión de C lo que lo hace más rápido y eficiente que otras como ADODB, además de brindar gran protección y seguridad mediante las consultas preparadas.

3.2.9. Patrón de Arquitectura en Capas.

Siguiendo la filosofía del modelo actual de desarrollo del software, para la realización del sistema se propone organizar los elementos de la aplicación en componentes independientes buscando alcanzar una mayor efectividad a la hora de administrarlos.

Para ello se seguirán los fundamentos de la programación en múltiples capas ya que esta además de facilitar una administración eficiente de los componentes que la integran. Tal y como plantea la arquitectura de esta tecnología, los componentes primarios de la aplicación serán divididos y programados por separados de forma tal que si una de las capas definidas sufre cambios, no se vean afectados el resto de las capas ni el resultado final del producto.

Se definen tres capas, tal es el caso de: Presentación, Reglas del Negocio y Acceso a Datos.

Presentación: En esta capa se diseña todo lo que constituye la interfaz gráfica y la interacción del usuario con el sistema.

Reglas del Negocio: Contiene todas las subrutinas que implementan la lógica del negocio.

Acceso a Datos: En esta capa se programa todo lo que tiene que ver con el acceso a la base de datos. Esta capa queda encargada de tomar la información de la base de datos dada una petición de la capa de Reglas del Negocio, que a su vez es generada por la capa de presentación.

3.2.10. Patrones para asignar responsabilidades.

Un sistema orientado a objetos se compone de objetos que se envían mensajes entre si para llevar a cabo las operaciones. La calidad de diseño de la interacción entre los objetos y asignación de responsabilidades presentan gran variación. Las decisiones poco aceptadas dan origen a sistemas y componentes frágiles y difíciles de mantener, entender, reutilizar o extender.

Una implementación hábil se funda en los principios que rige un buen diseño orientado a objetos.

En los patrones GRASP se reflejan algunos de estos principios, que se aplican al preparar los diagramas de interacción cuando se asignan responsabilidades.

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns o patrones generales de software para asignar responsabilidades. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente un software orientado a objetos.

Entre los patrones GRASP que se han tenido en cuenta a la hora de realizar los diagramas de interacción se encuentran:

Experto: Es un patrón que se utiliza más que cualquiera a la hora de asignar responsabilidades. Expresan que los objetos hacen cosas de acuerdo a la información con que cuentan. Esto trae como beneficio que se conserve el encapsulamiento, además de favorecer el bajo acoplamiento.

Creador: El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como controlador, se da soporte al bajo acoplamiento.

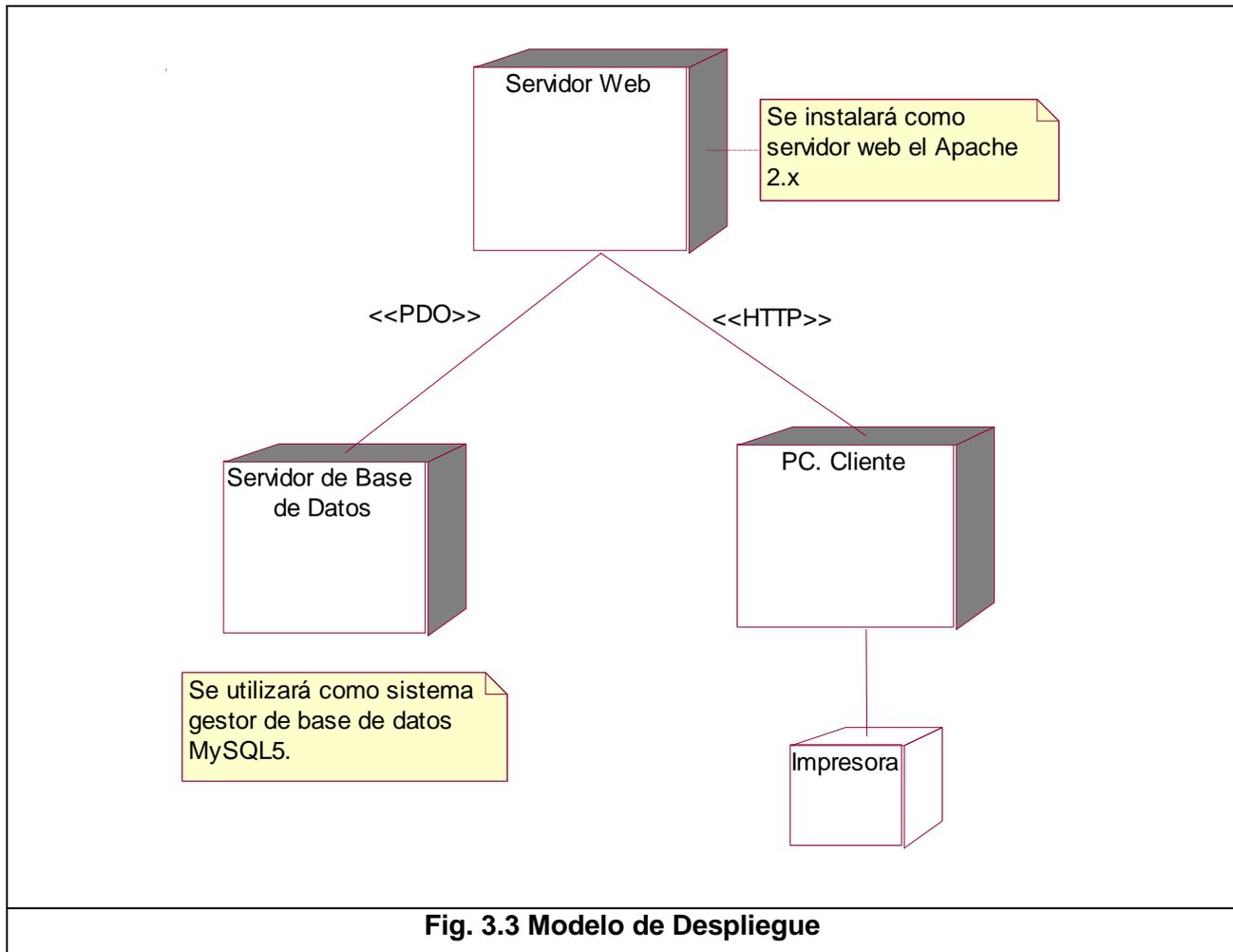
Bajo Acoplamiento: Este patrón soporta el diseño de clases más independientes, que reducen el impacto de los cambios y las hacen más reutilizables.

Alta Cohesión: Es muy útil debido a que una clase con alta cohesión es muy fácil de darle mantenimiento, entenderla y reutilizarla, su alto grado funcional combina con una reducida cantidad de operaciones, lo que simplifica los cambios que puedan ocurrir en la misma.

Controlador: Encargado de manejar los eventos del sistema. (LARMAN 2002)

3.3. Modelo de despliegue.

El modelo de despliegue es un modelo de objetos, que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware. (JACOBSON 2000)



Conclusiones parciales

En este capítulo se modeló detalladamente la solución propuesta a través de los modelos de análisis, diseño y los artefactos necesarios de implementación para posibilitar una mayor comprensión. Se representó mediante diagramas de clases utilizando estereotipos Web la lógica del negocio del sistema, alcanzando de esta forma un alto grado de especificidad en la realización de los casos de uso, para la asignación de responsabilidades se utilizaron patrones de diseño expuestos en este capítulo. Se construyó el modelo de datos que da soporte a la base de datos que usa nuestro sistema, fue realizada la modelación de los nodos en los que será distribuida la aplicación especificando para cada uno de éstos el protocolo de comunicación.

CONCLUSIONES GENERALES.

A modo de conclusión se puede decir que con el desarrollo del Sistema Informático para el Departamento de Matemática y Física Computacional se han obtenido los siguientes resultados:

- Permite el acceso rápido y simultáneo apoyado en su interfaz Web a la información que se manipula en el grupo.
- La solución propuesta utilizando el Modelo de Dominio ha sido acertada, los requerimientos soportan al sistema y los casos de uso satisfacen las necesidades funcionales.
- Se logra una seguridad y protección de los datos consecuente con el nivel de seguridad requerido.
- Después de un estudio previo y mediante esta puesta en práctica, se ha demostrado la eficacia de los lenguajes y tecnologías utilizadas para el desarrollo del sistema.
- Se obtiene la primera iteración de un producto que contribuirá con la organización y control de todas las actividades que se desarrollan en el grupo.

Con el estudio realizado y la propuesta del sistema se le da cumplimiento al objetivo propuesto: Modelar un sistema informático que gestione toda la información referente al grupo de investigación de Matemática y Física Computacional, garantizando el acceso a trabajos investigativos realizados por miembros de esta comunidad.

RECOMENDACIONES.

Una vez concluido el desarrollo de este documento se recomienda:

- Realizar un período de prueba del software con el objetivo de erradicar algunas posibles fallas.
- Continuar el estudio con el objetivo de encontrar nuevas funcionalidades para refinar e implementar una herramienta más completa.

Realizar un estudio de los distintos FrameWork existentes para PHP, analizar sus potencialidades para una posible expansión del sistema.

Realizar una extensión del sistema para el departamento de la especialidad de la facultad 5.

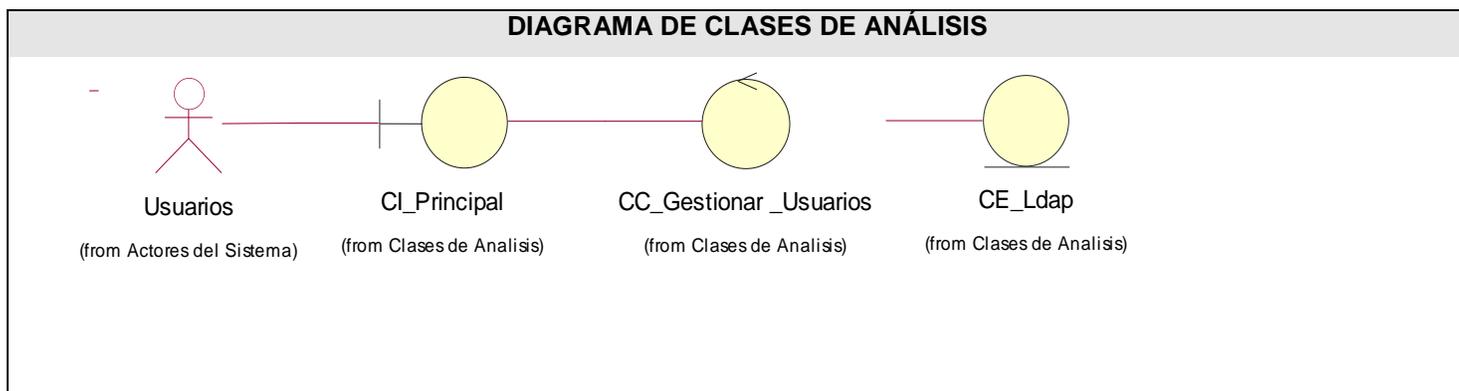
BIBLIOGRAFÍA

- ACADEMIA, R. Que es Gestion, 2006.
- ADOBE Principales funciones de ADOBE DreamWeaver, 2007.
- ALVAREZ, M. A. Índice del manual de lenguajes del web.
---. *Javascript*, 2007.
- ARORA, G.; B. AIASWAMY, *et al.* Programación C#, 2002.
- GRAJALES, T. EL CONCEPTO DE INVESTIGACIÓN, 2000.
Implantación de Sistemas de Gestión Documental, 2006.
- JACOBSON, I. *El Proceso unificado de desarrollo de software*. Pearson Educación, 2000. p. 84-7829-036-2
- JARAMILLO, M. B. Que es Gestion, 2001.
- LARMAN, C. *UML y Patrones*. 2002. p.
- LLOP, J. M. T. Ataque de denegación de servicio, 2004.
- MACROMEDIA, I. *Utilización de Dreamweaver*, 2003.
- MARCIBARRIOS AJAX, 2005.
- MARTÍNEZ, G. M. ingeniería del Software(UML), 2006.
- MYSQL, 2007.
- OFFICE, V. Gestión Documental, 2006.
PostgreSQL.
- RAMM. CSS, 2005.
Rational Rose. 2006.
- ROYERO, J. *Gestión de Sistemas de Investigación en América Latina* 2002.
- WIKIPEDIA ASP.NET, 2007a.
--- Firebird, 2007b.
--- Flex, 2007c.
--- HTML, 2007d.
--- Internet Information Services, 2007e.
--- Microsoft SQL Server, 2007f.
--- Oracle, 2007g.
--- PHP, 2007h.
--- Python, 2007i.
--- Servidor HTTP Apache, 2007j.
--- SQLite, 2007k.
--- XML, 2007l.

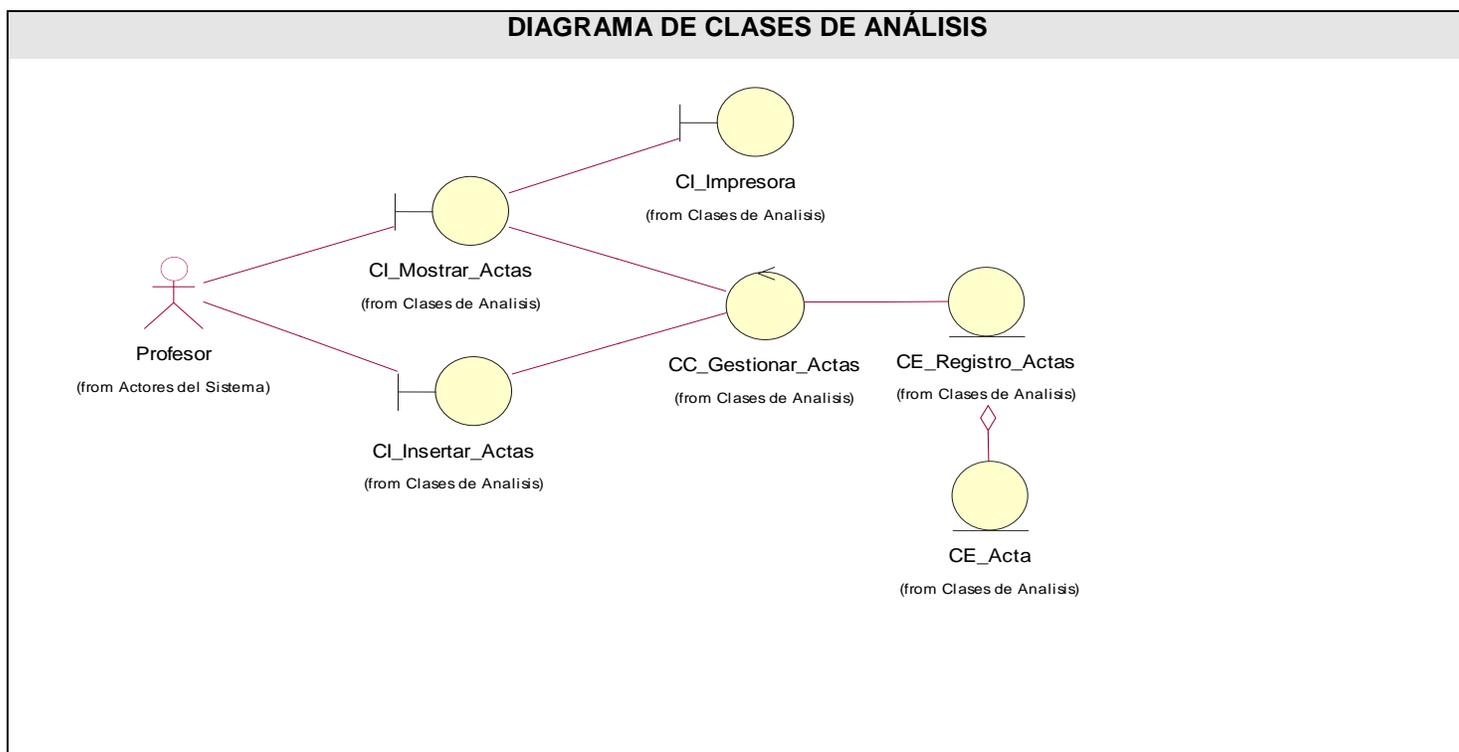
ANEXOS

Anexo 1 Diagramas de Clases del Análisis

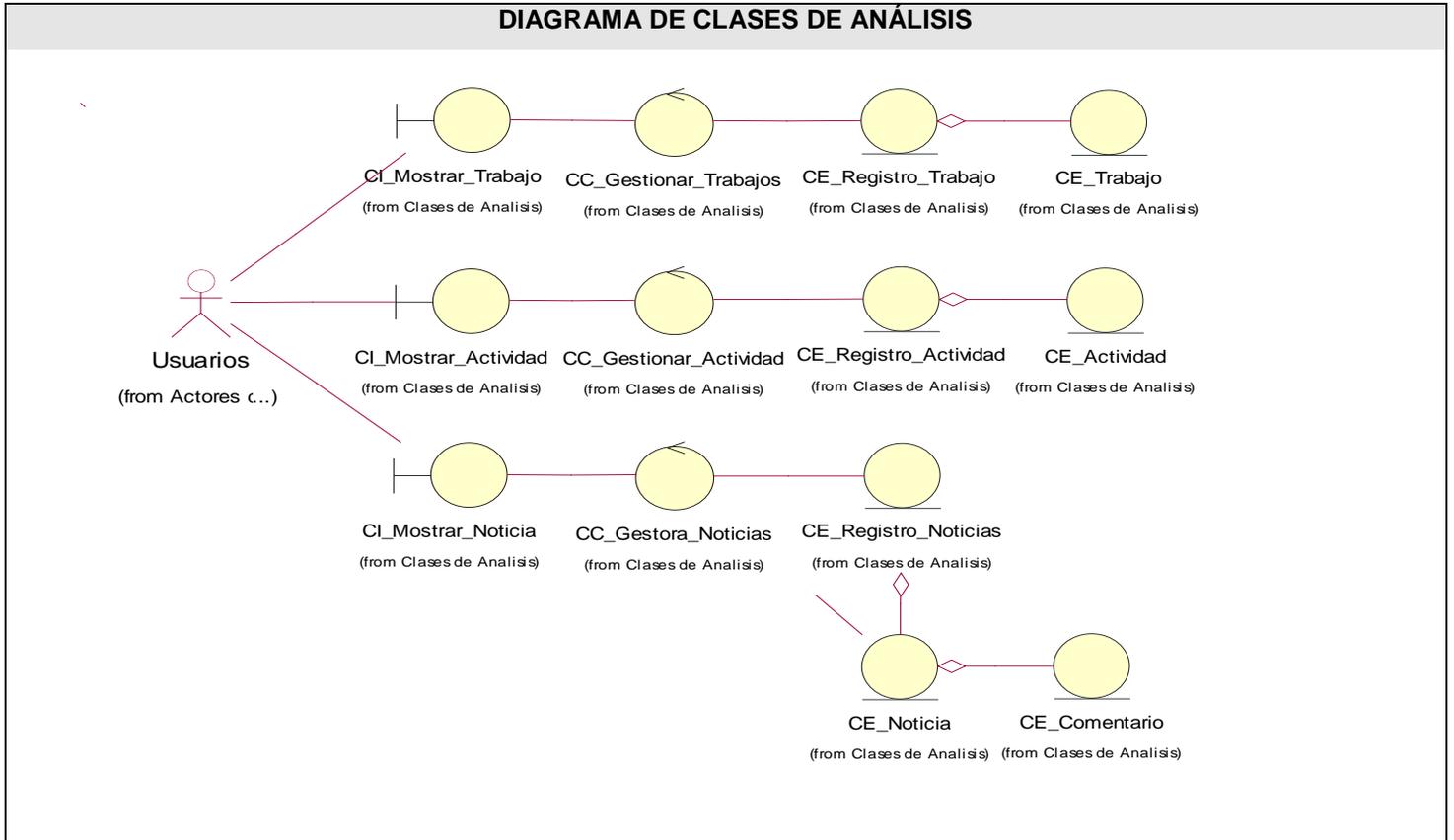
Caso de Uso: Autenticar Usuario.

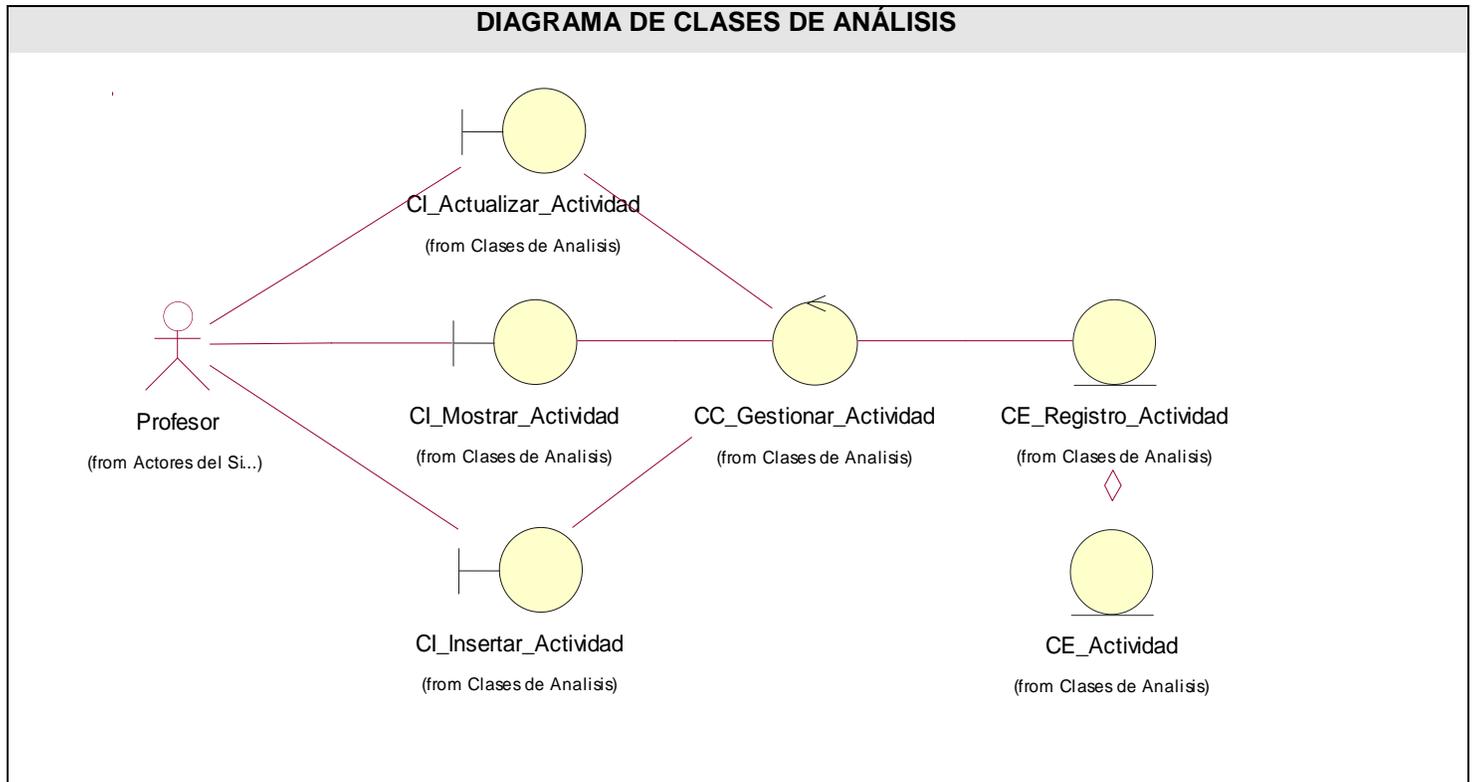


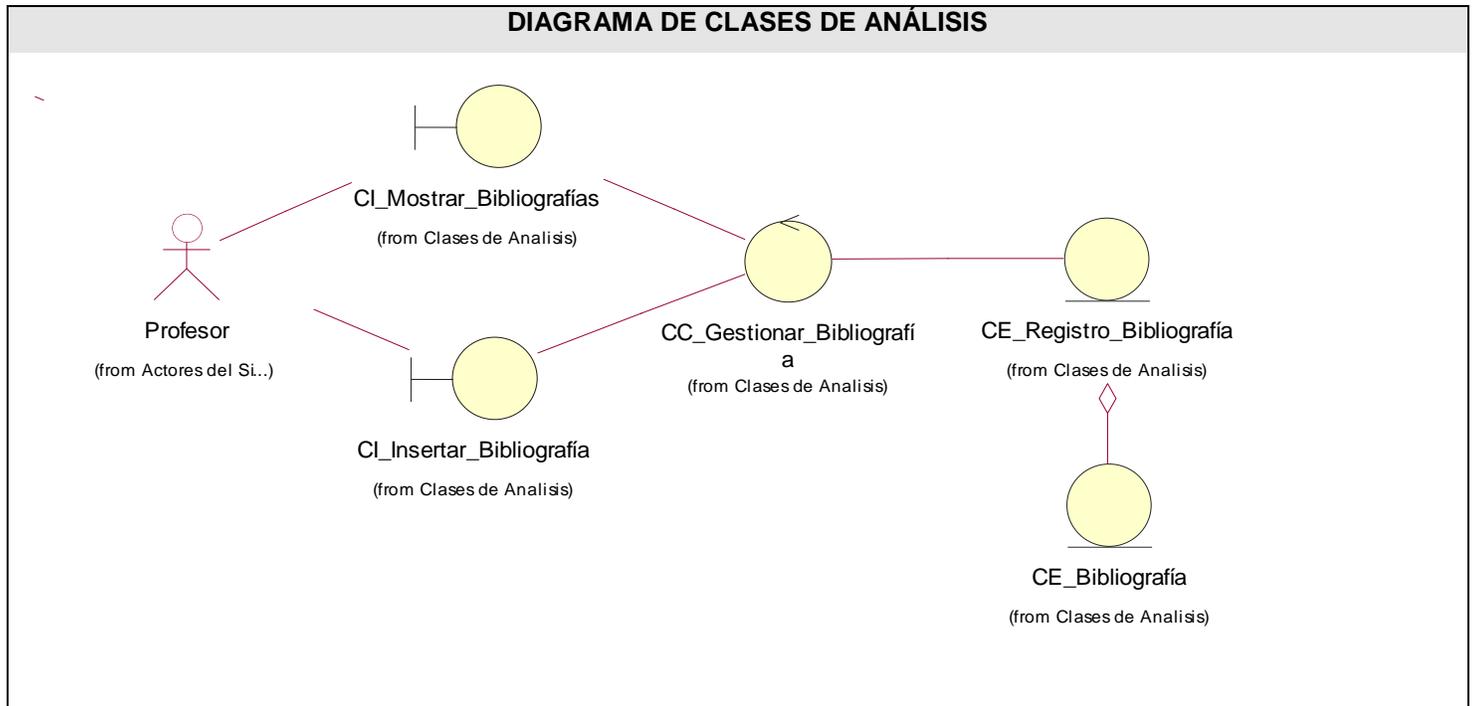
Caso de Uso: Confeccionar Actas de Reuniones.



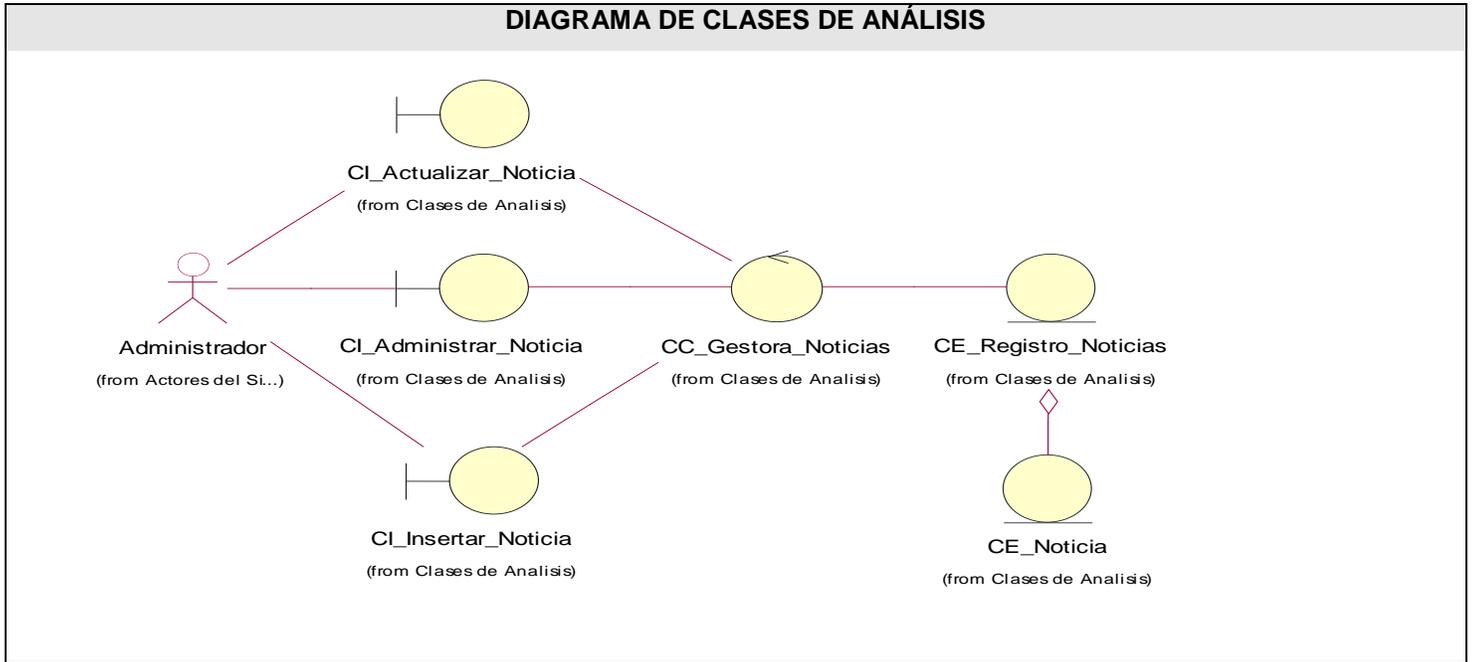
Caso de Uso: Consultar Información.



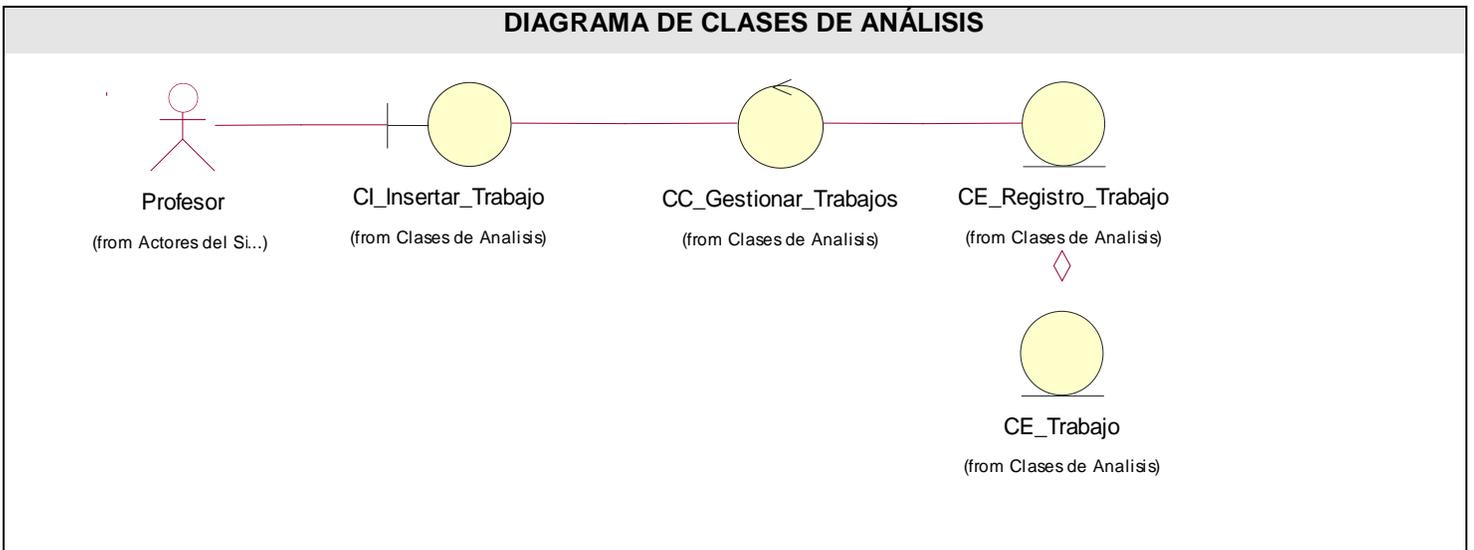
Caso de Uso: Gestionar Actividades.

Caso de Uso: Gestionar Bibliografía.

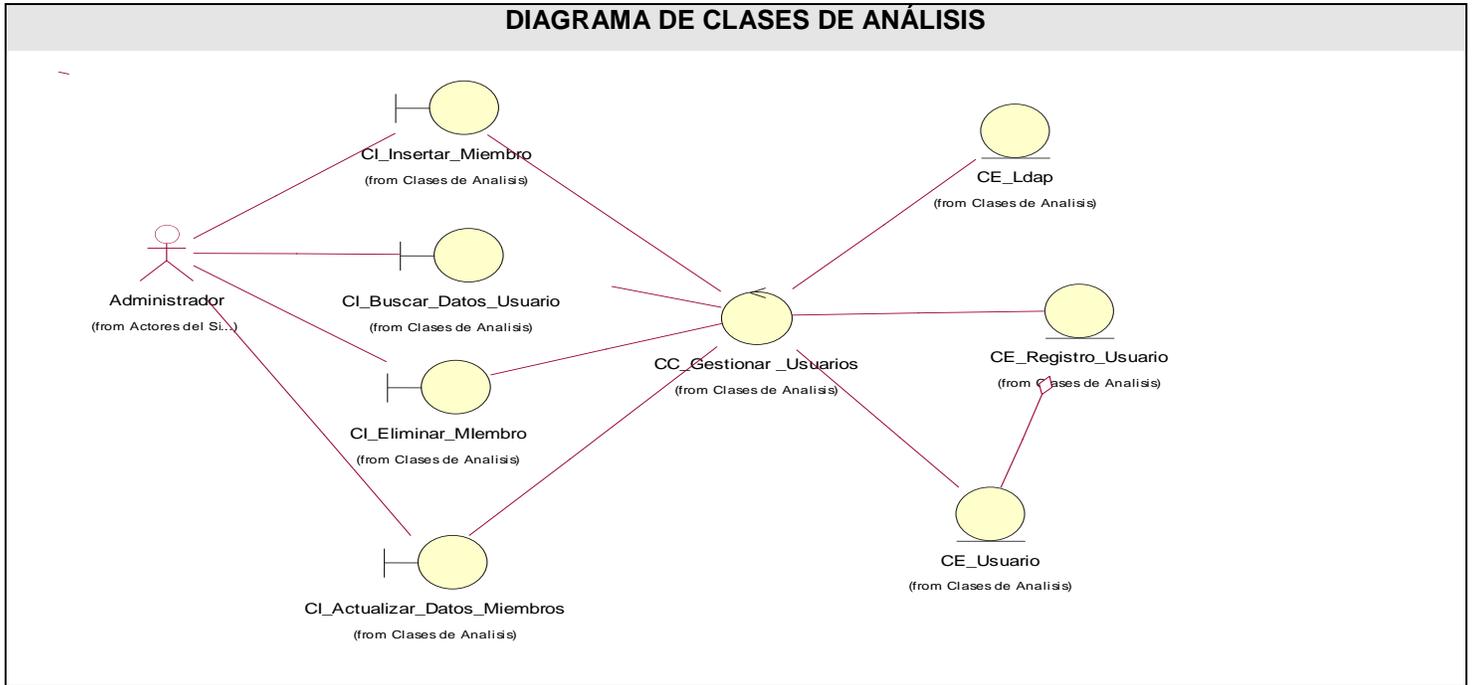
Caso de Uso: Gestionar Noticia.



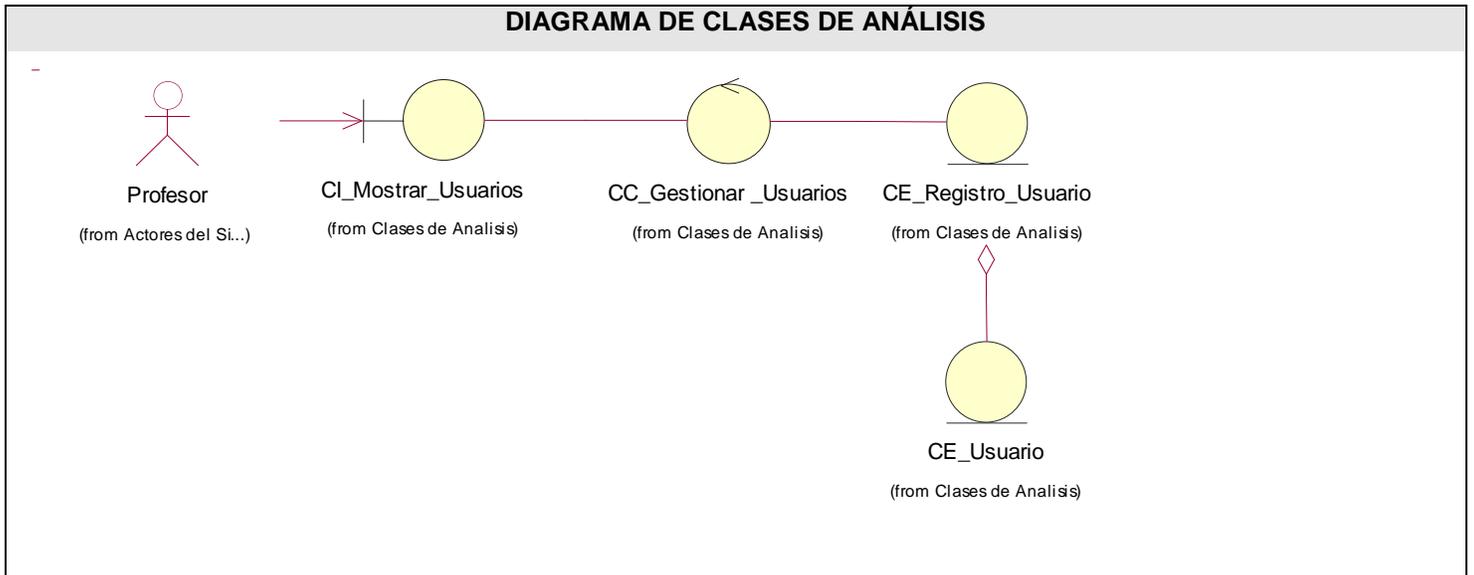
Caso de Uso: Gestionar Trabajos Científicos.



Caso de Uso: Gestionar Trabajos Usuarios.



Caso de Uso: Listar Miembros.



Anexo II Diagramas de Clases del Diseño.

Diagrama de clases del diseño: Autenticar

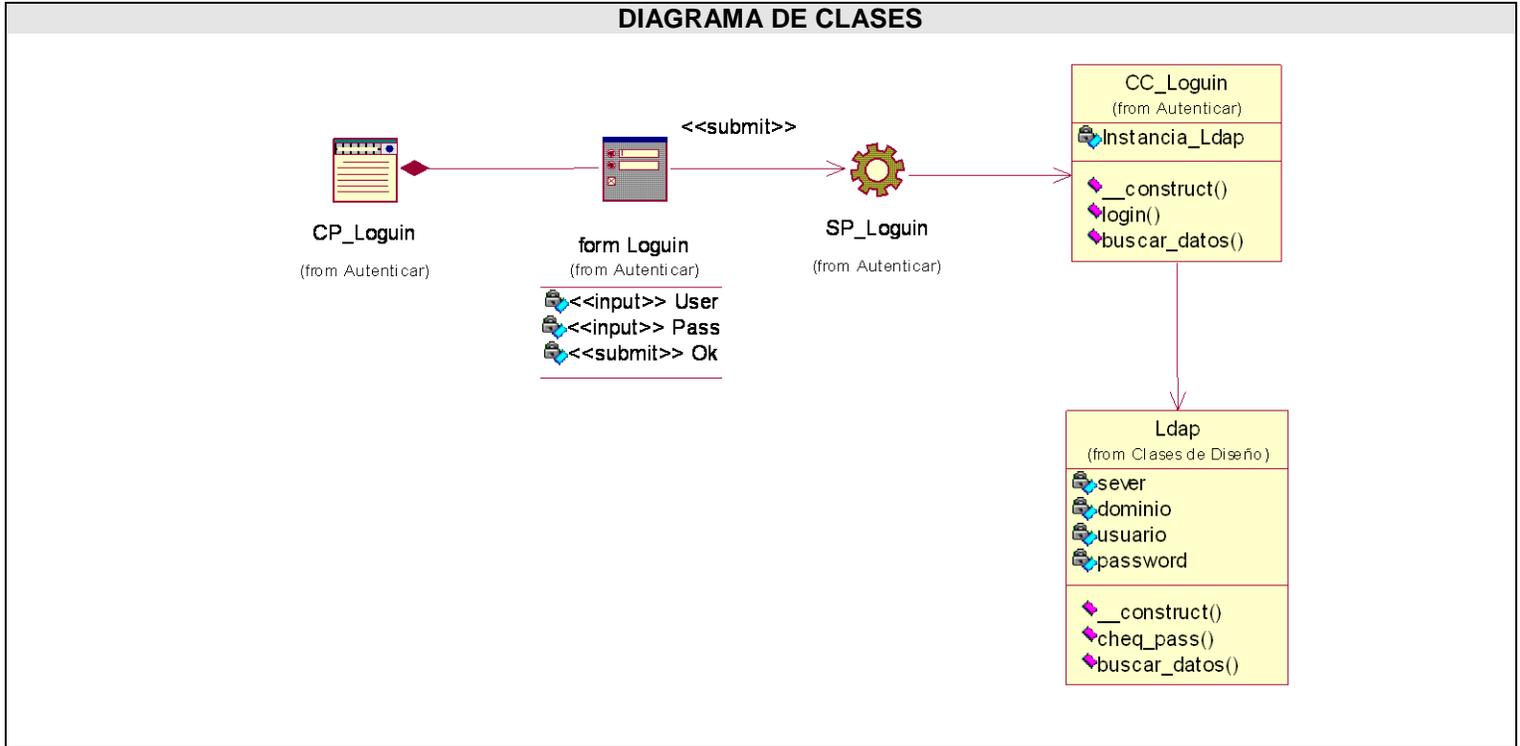


Diagrama de clases del diseño: Confeccionar Actas

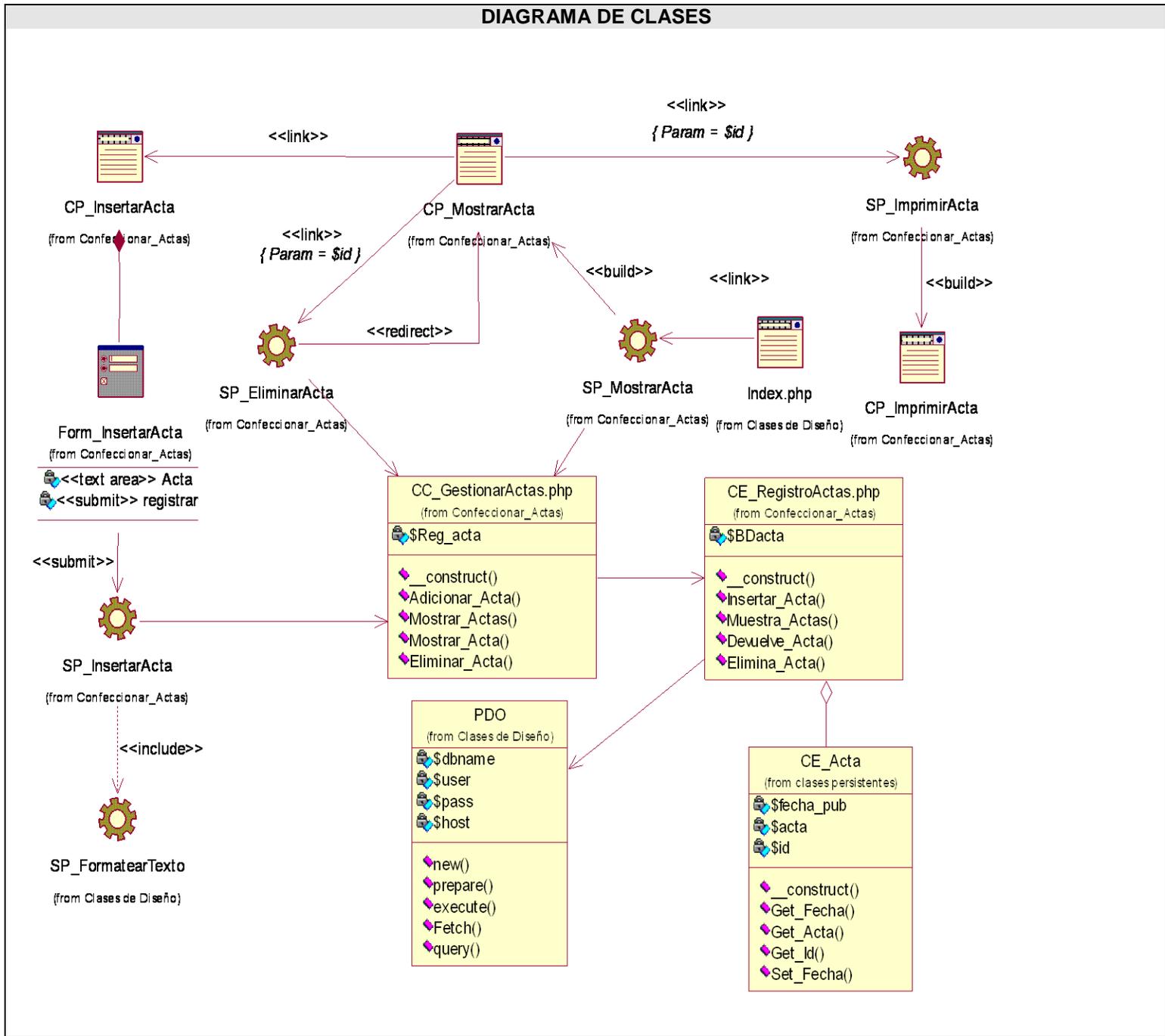


Diagrama de clases del diseño: Consultar Información.

DIAGRAMA DE CLASES

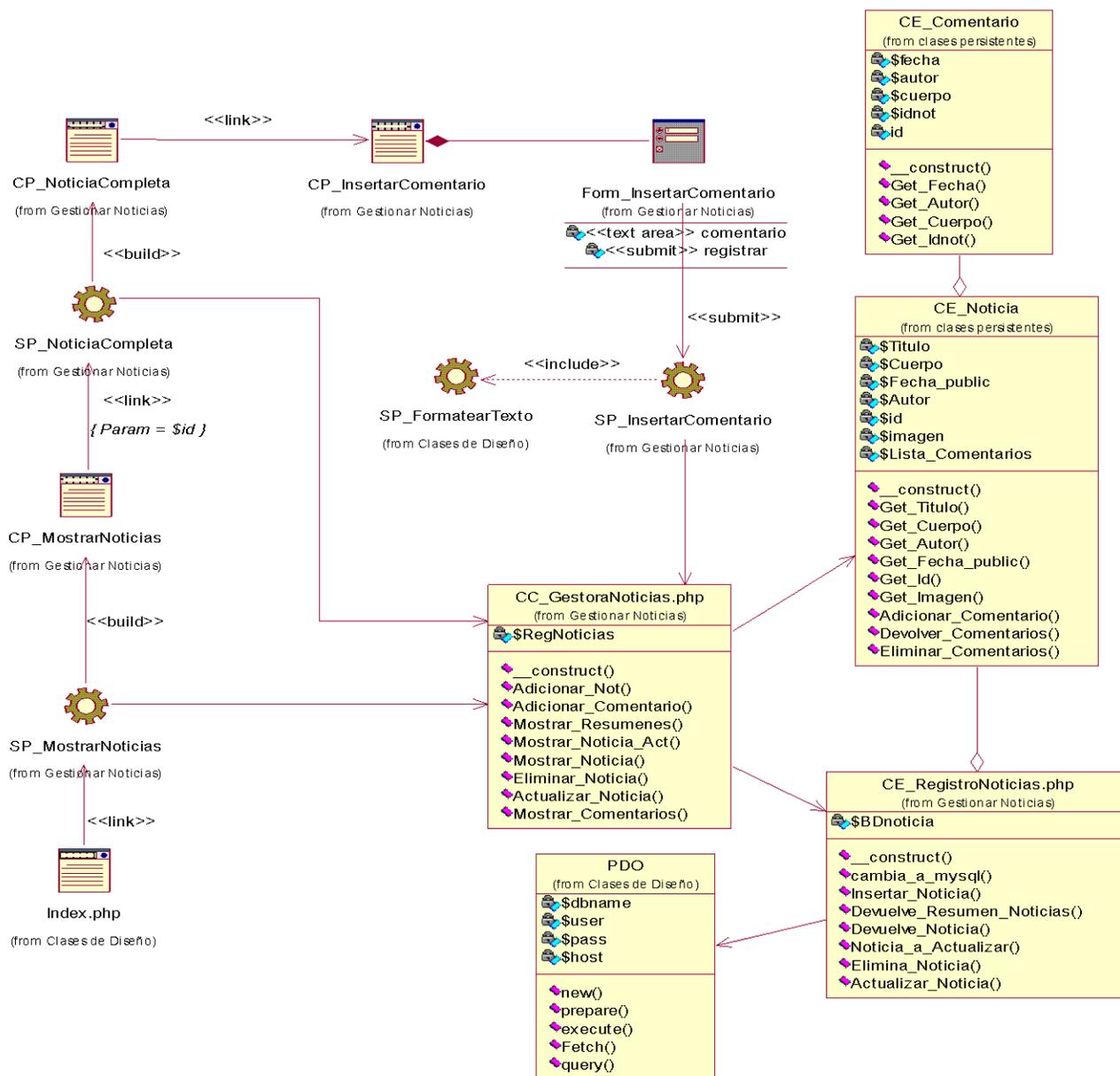


Diagrama de clases del diseño: Gestionar Usuarios.

DIAGRAMA DE CLASES

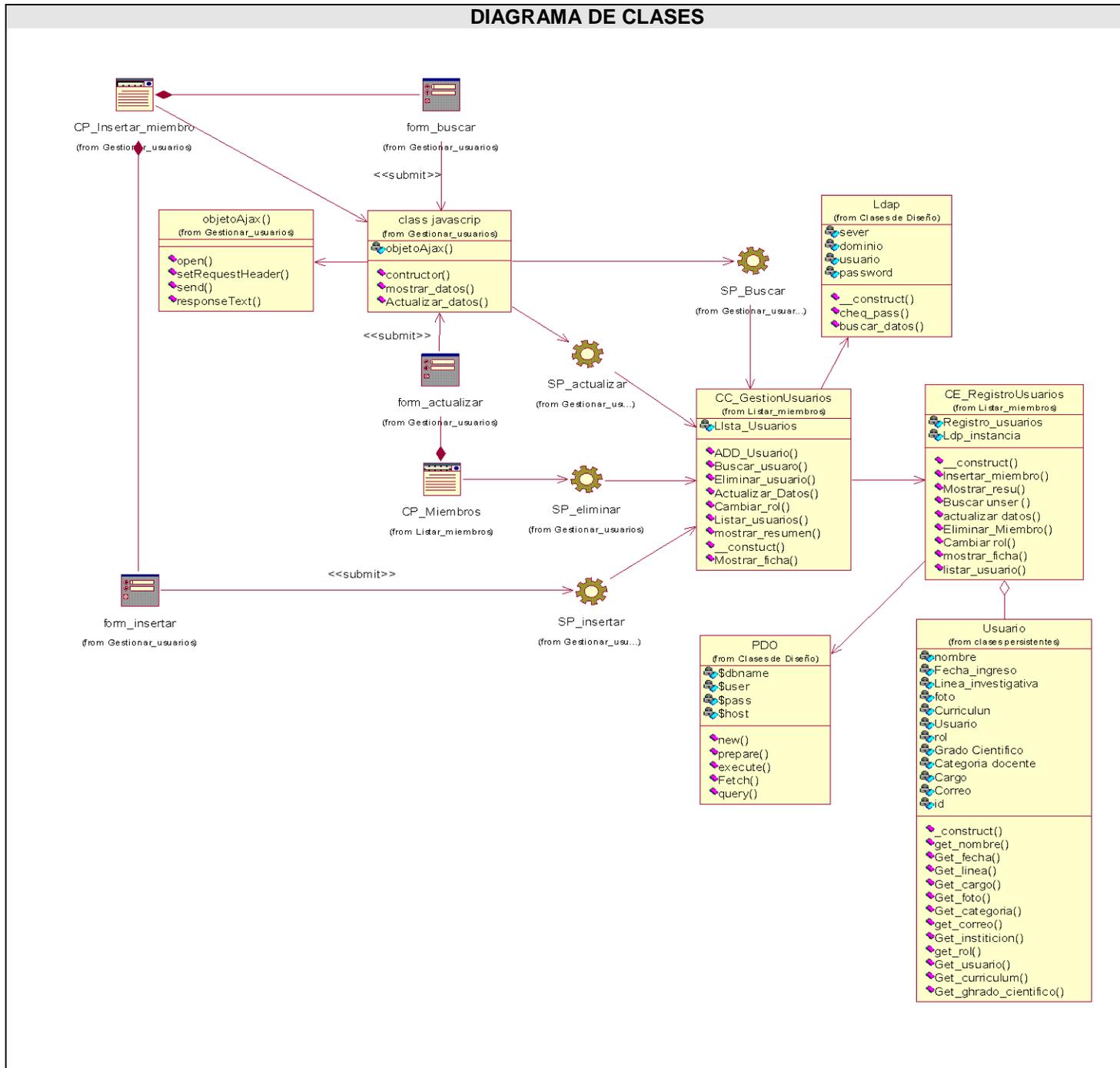


Diagrama de clases del diseño: Gestionar Actividad.

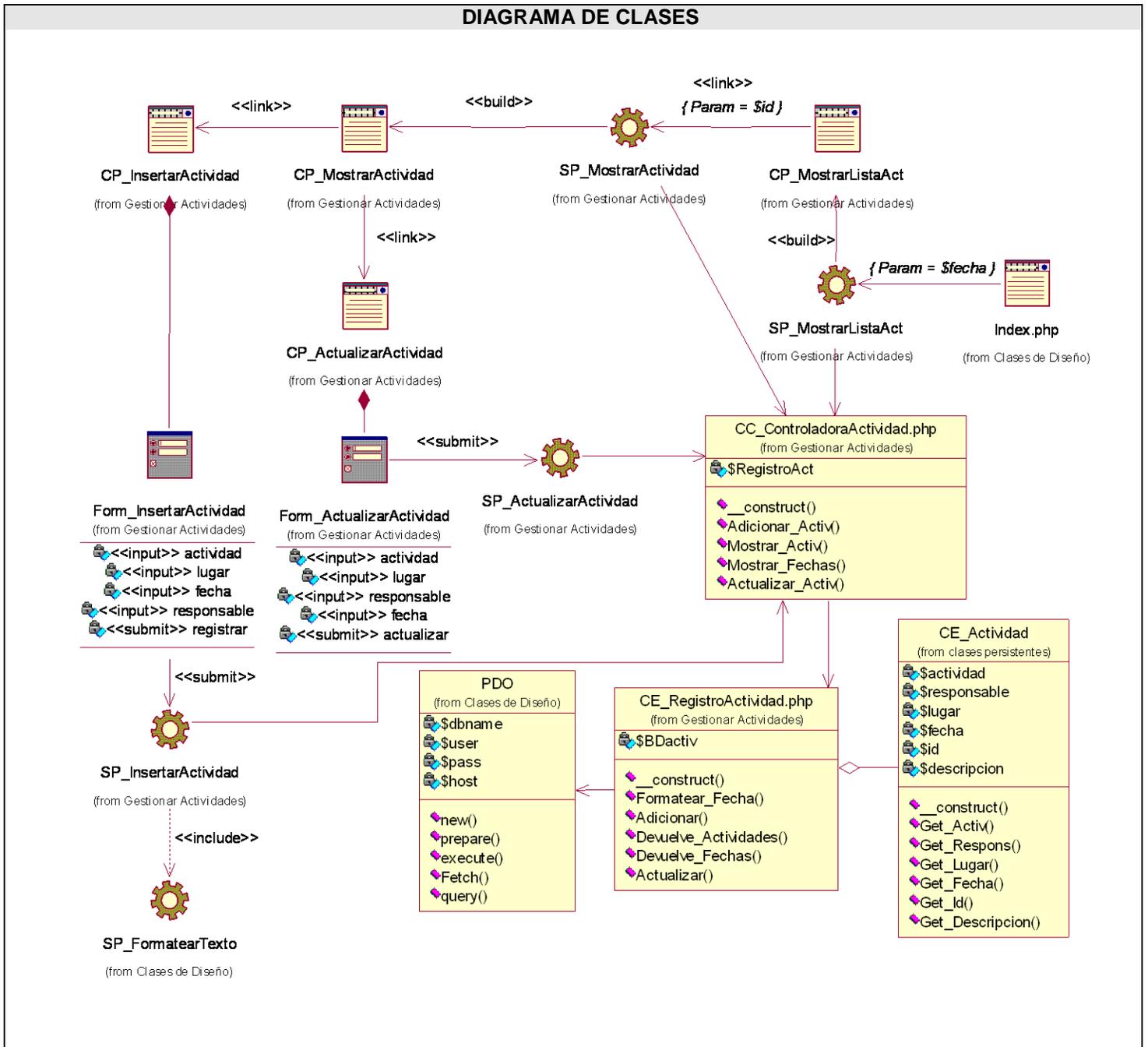


Diagrama de clases del diseño: Gestionar Noticia.

DIAGRAMA DE CLASES

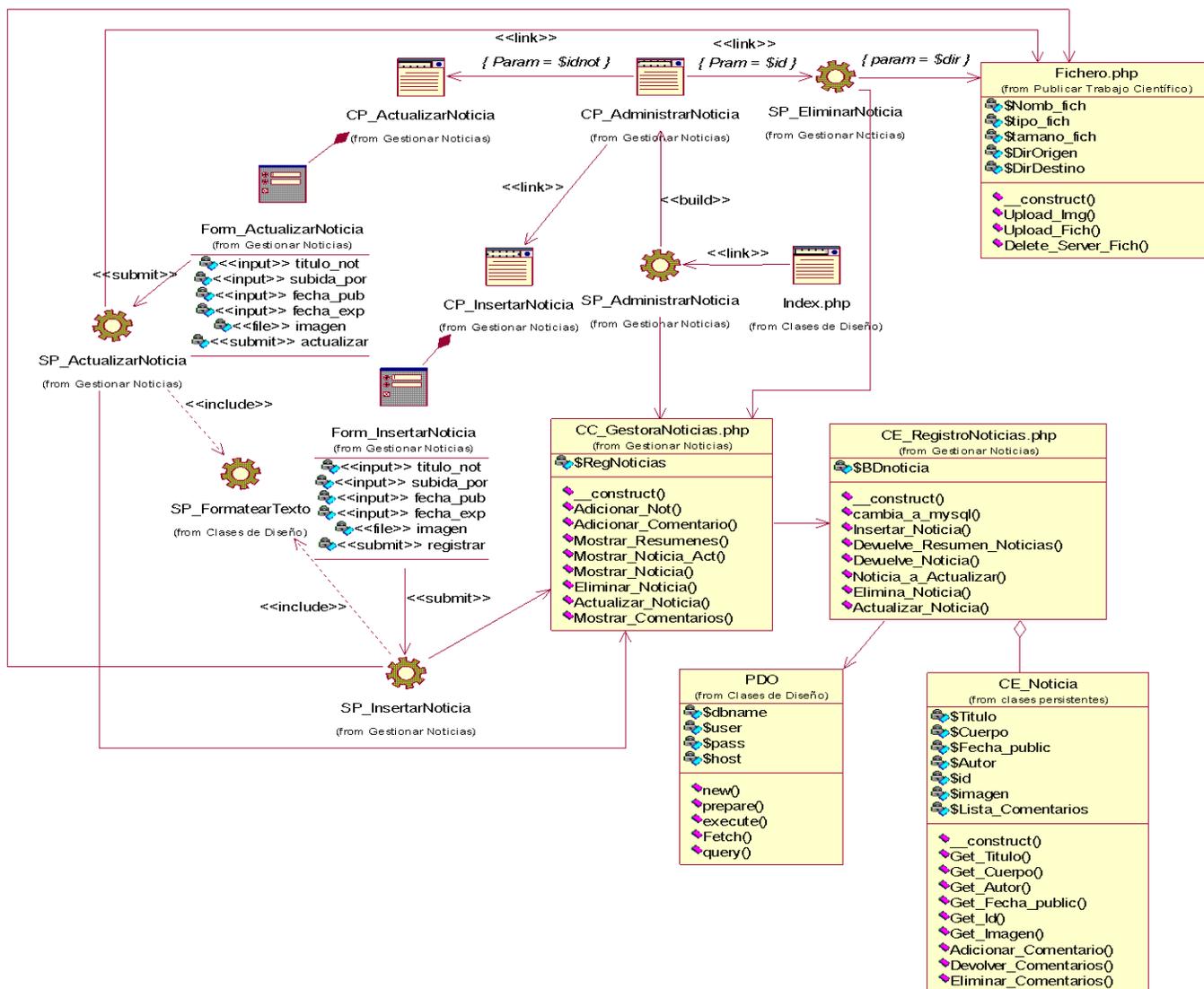
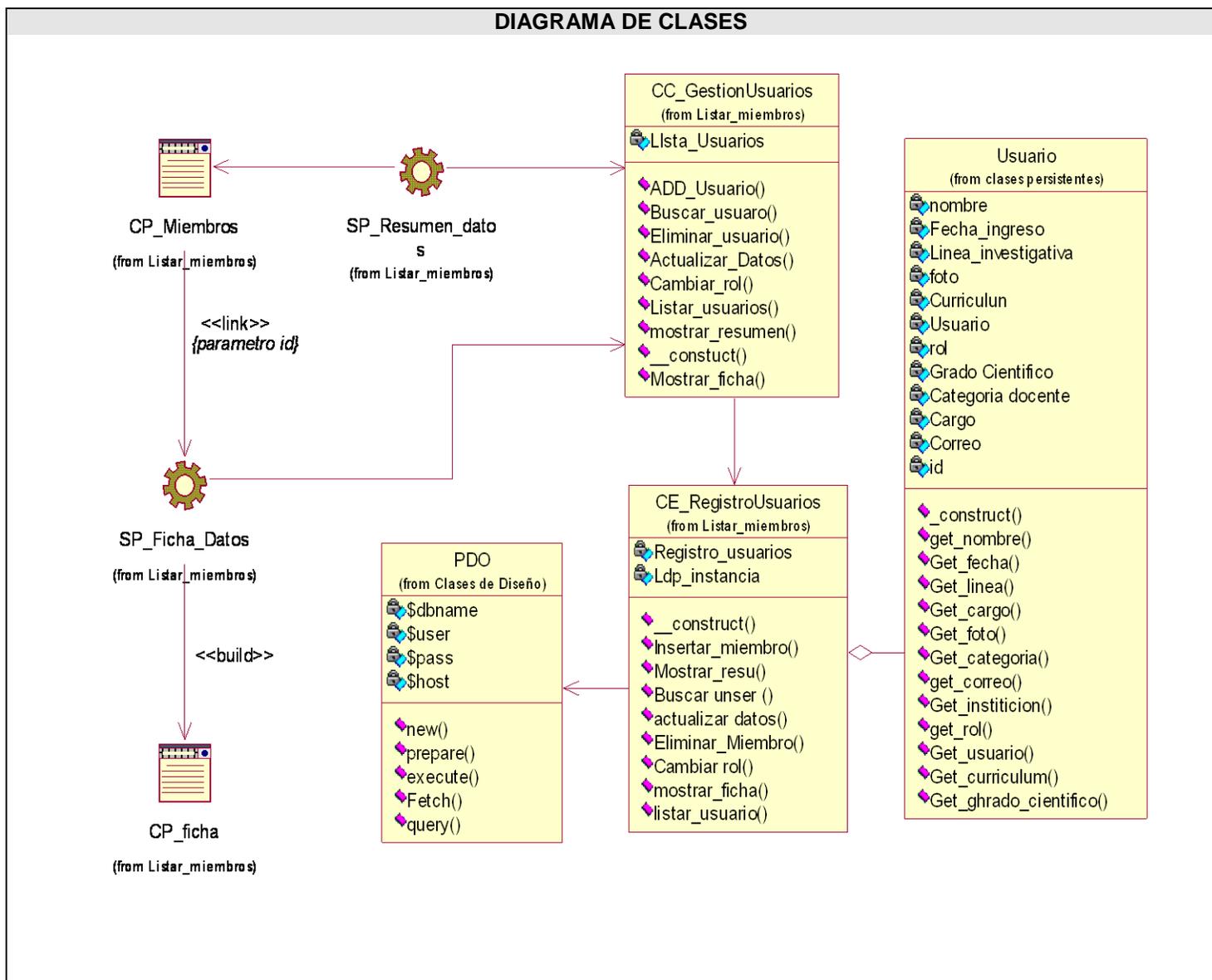


Diagrama de clases del diseño: Listar Miembros.



Anexo III Diagramas de Interacción.

Diagrama de Secuencia: Autenticar

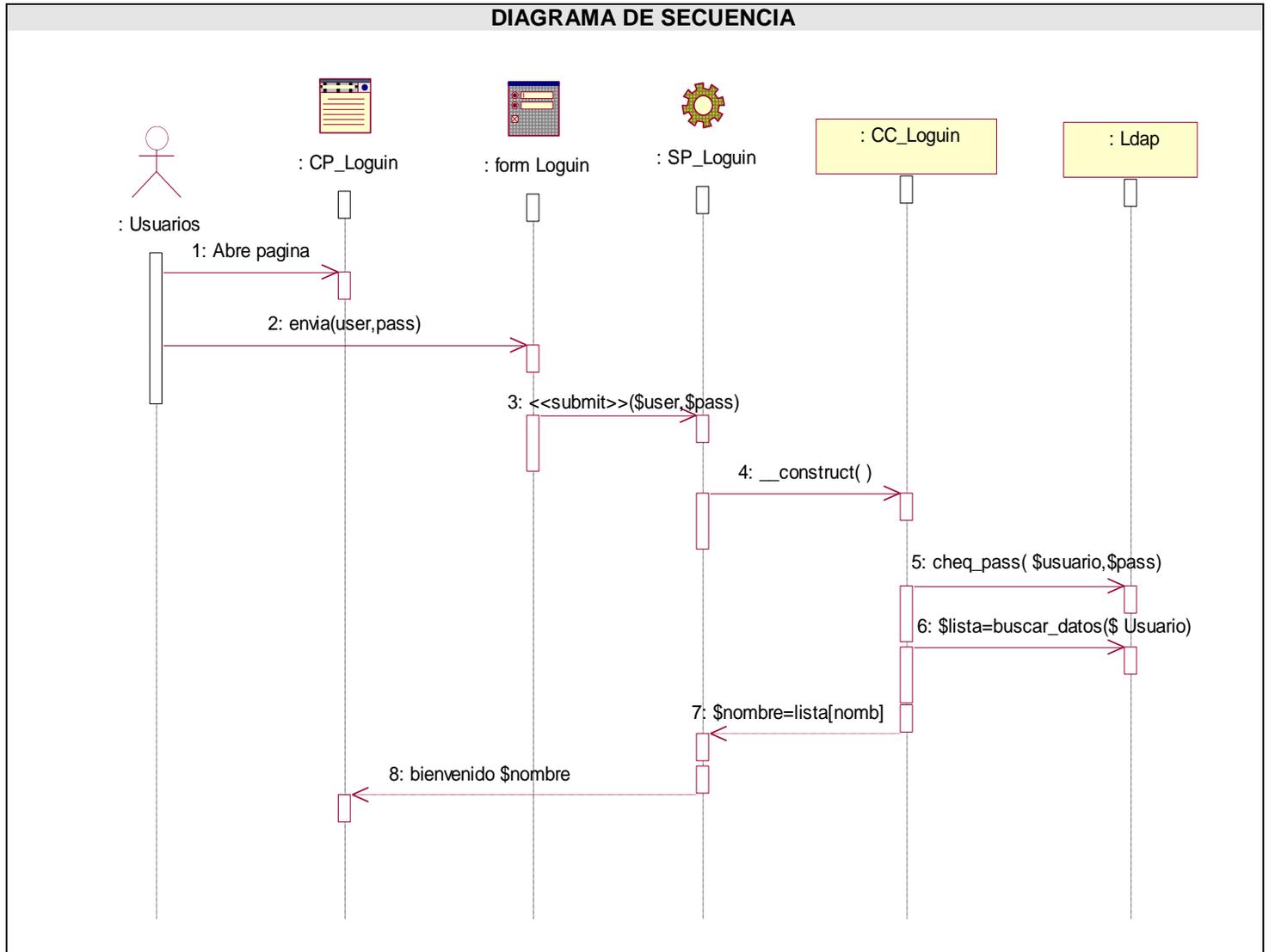


Diagrama de Secuencia: Gestionar Acta ('Insertar Acta').

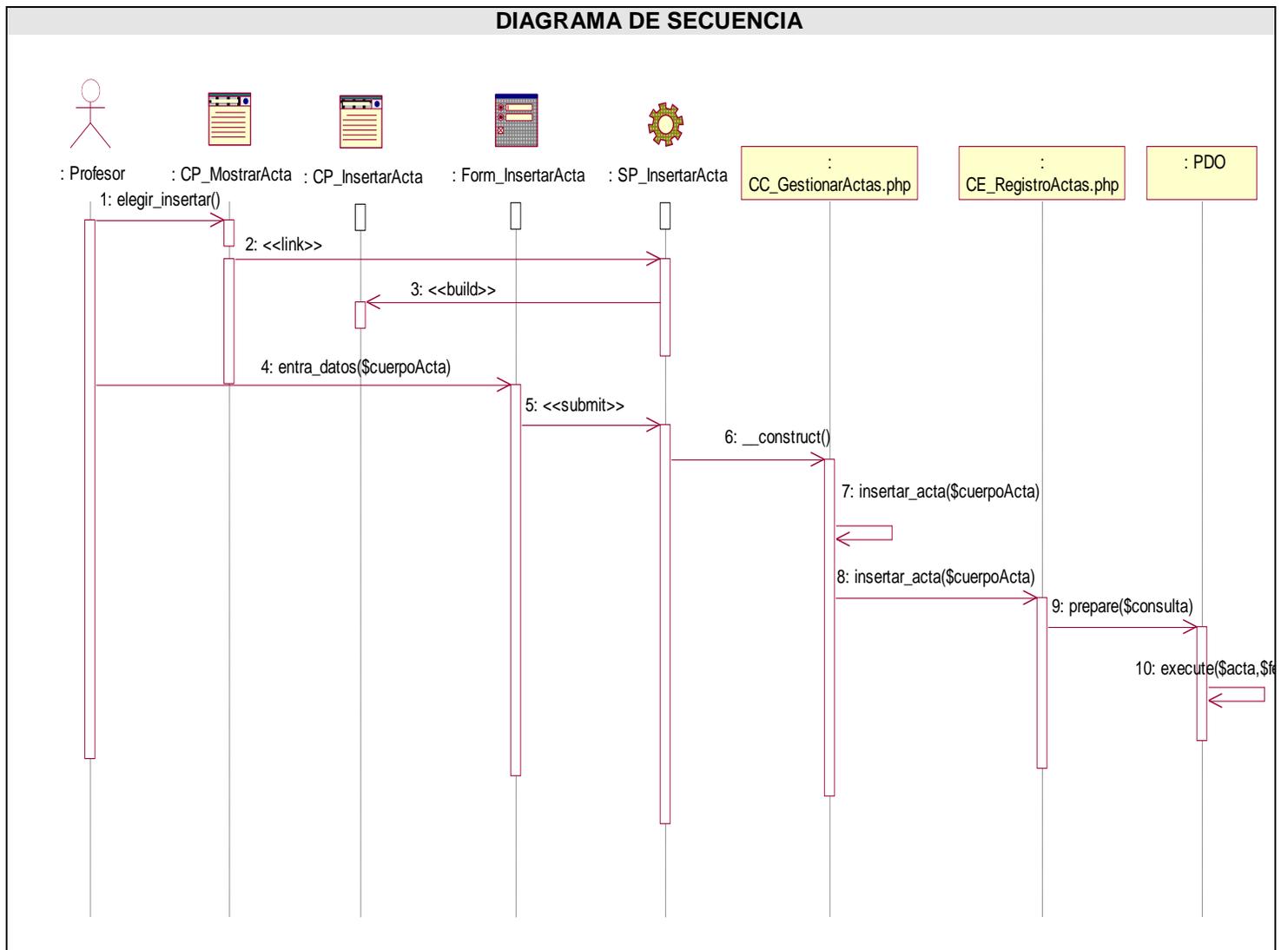


Diagrama de Secuencia: Gestionar Acta ('Eliminar Acta').

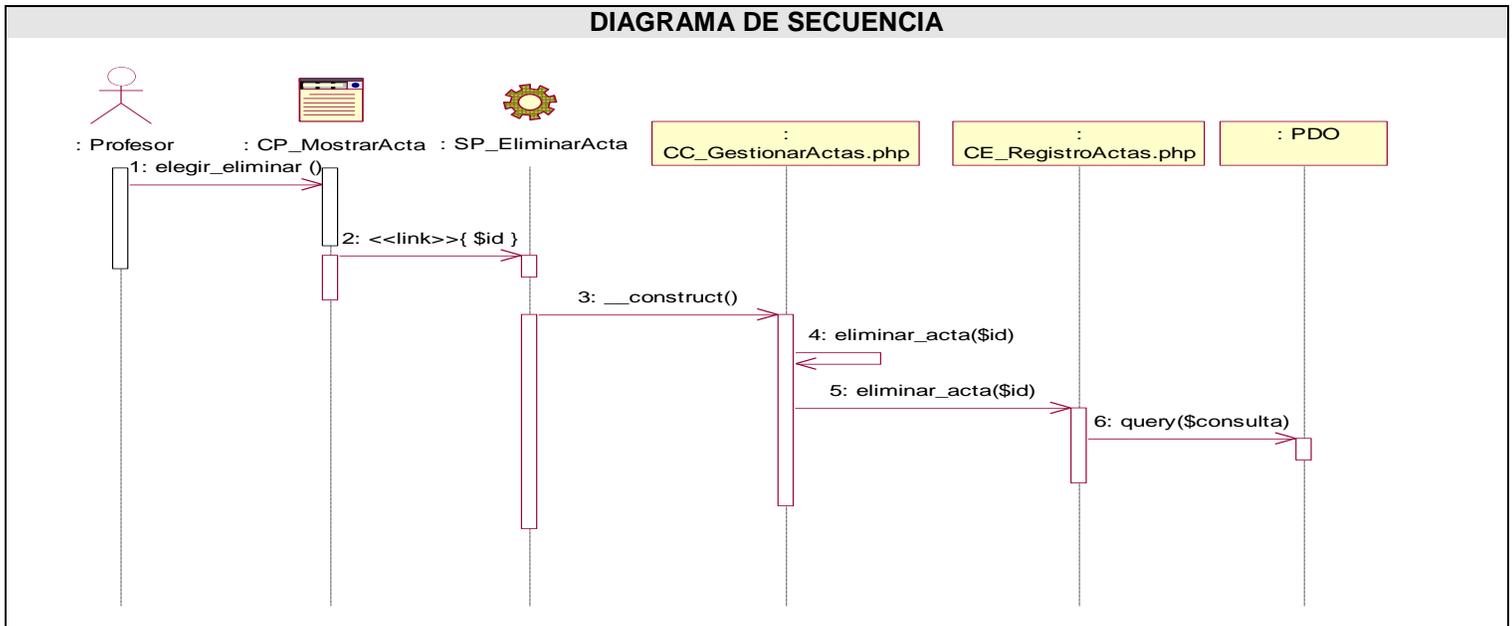


Diagrama de Secuencia: Consultar Información ('Mostrar Noticias').

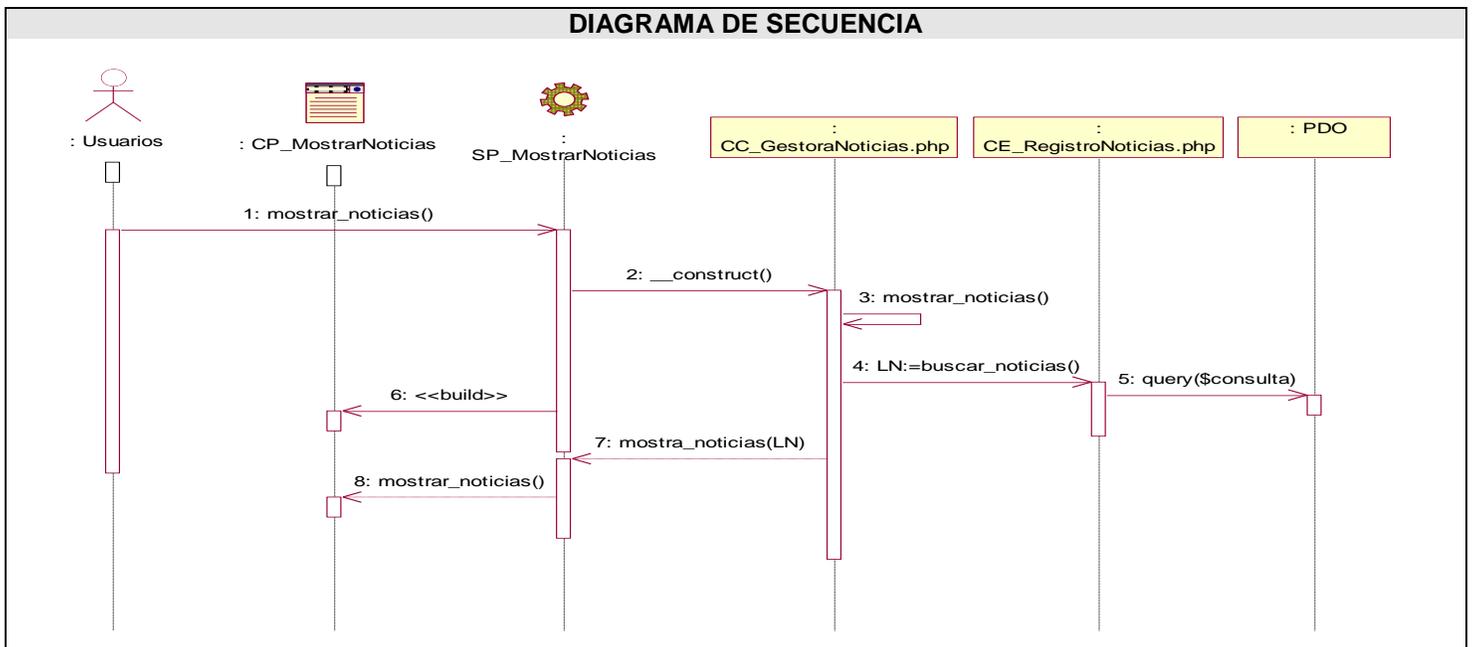


Diagrama de Secuencia: Consultar Información ('Noticia Completa').

DIAGRAMA DE SECUENCIA

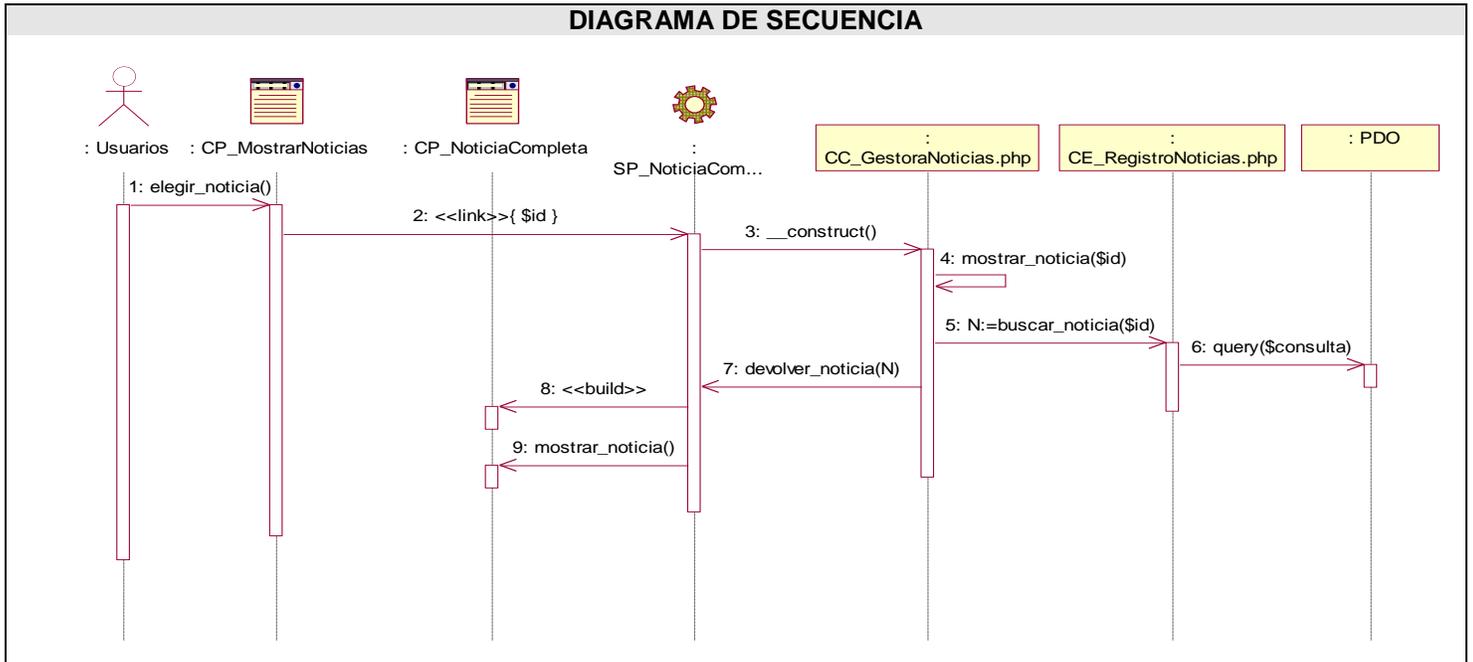


Diagrama de Secuencia: Consultar Información ('Insertar Comentario').

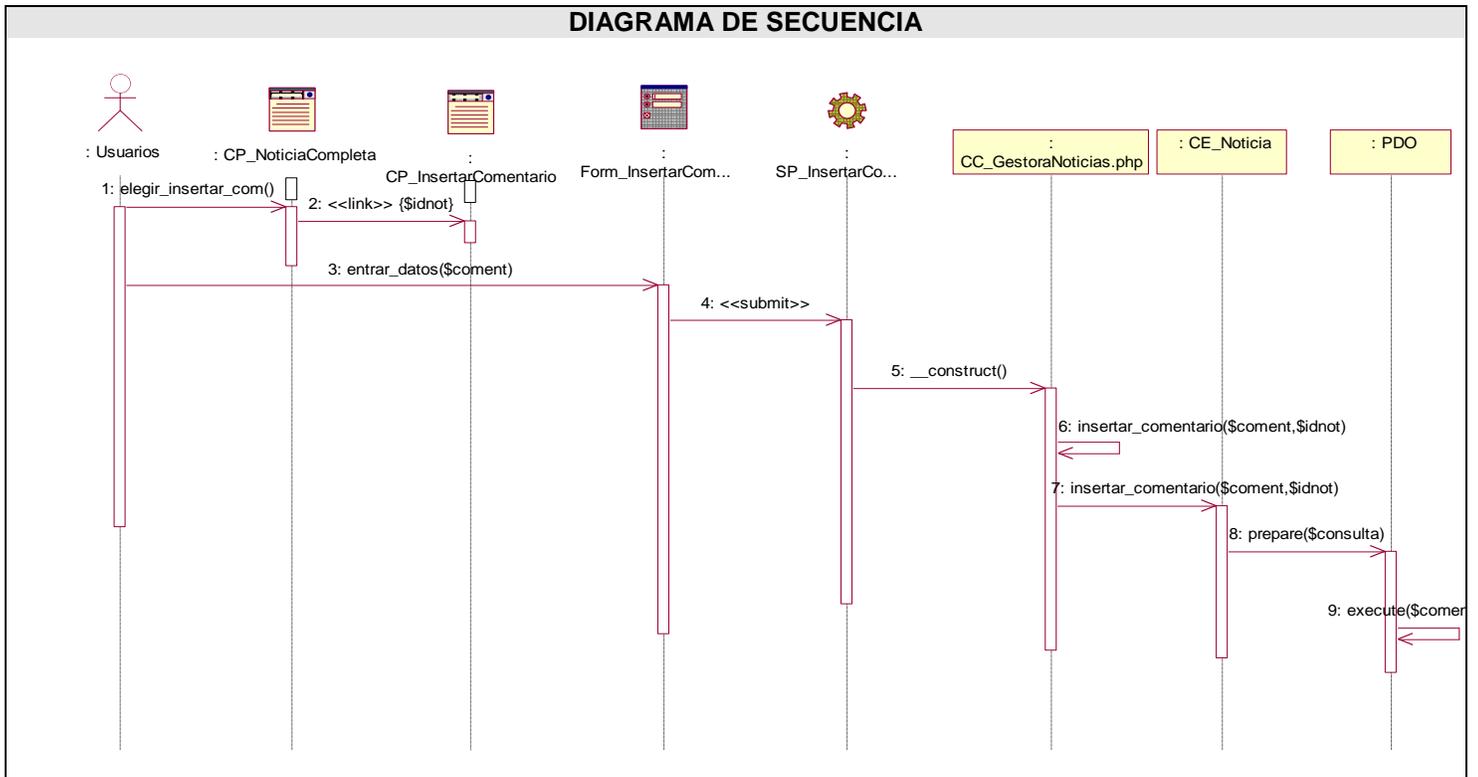


Diagrama de Secuencia: Gestionar Usuarios ('Insertar Usuario').

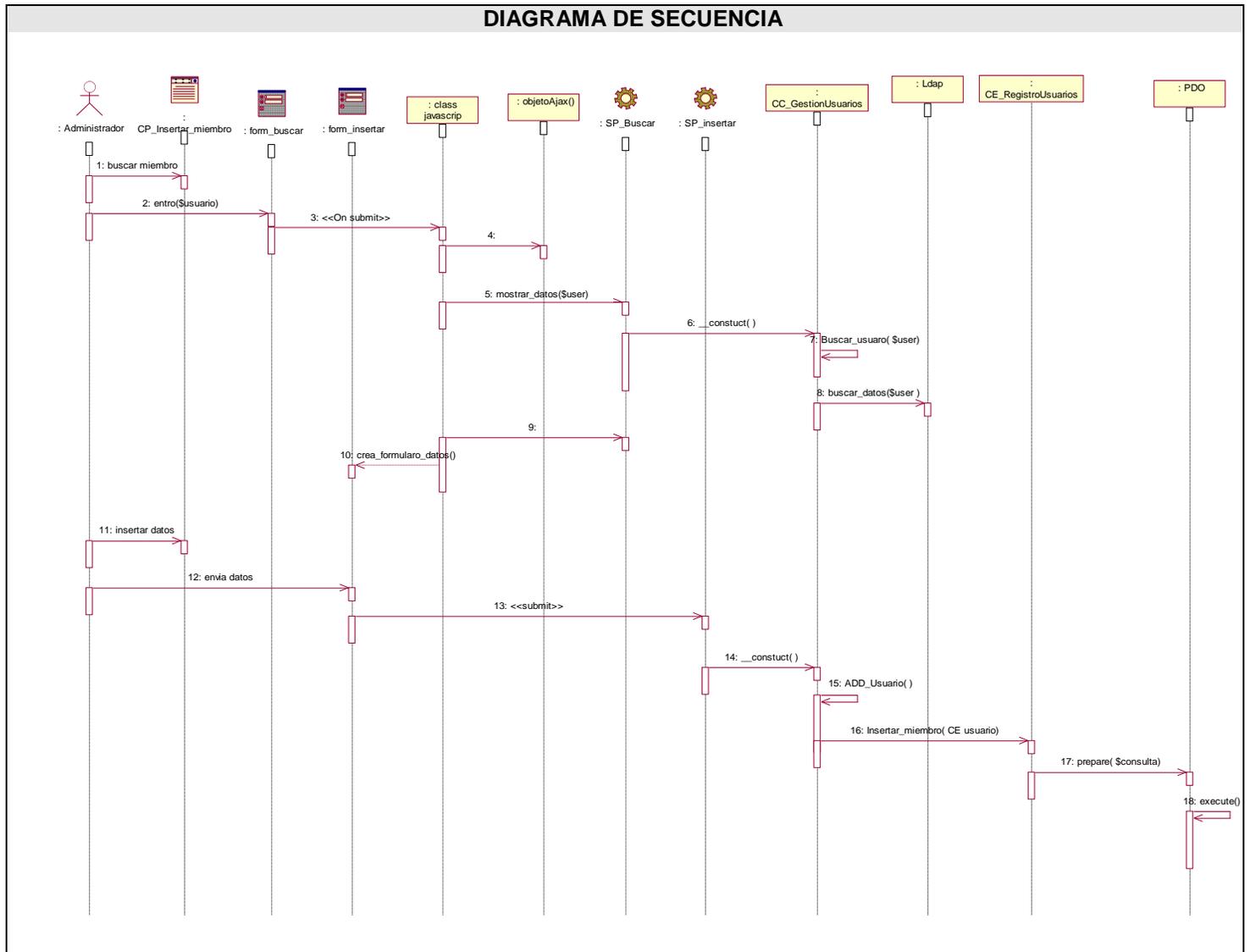


Diagrama de Secuencia: Gestionar Usuarios ('Actualizar Usuario').

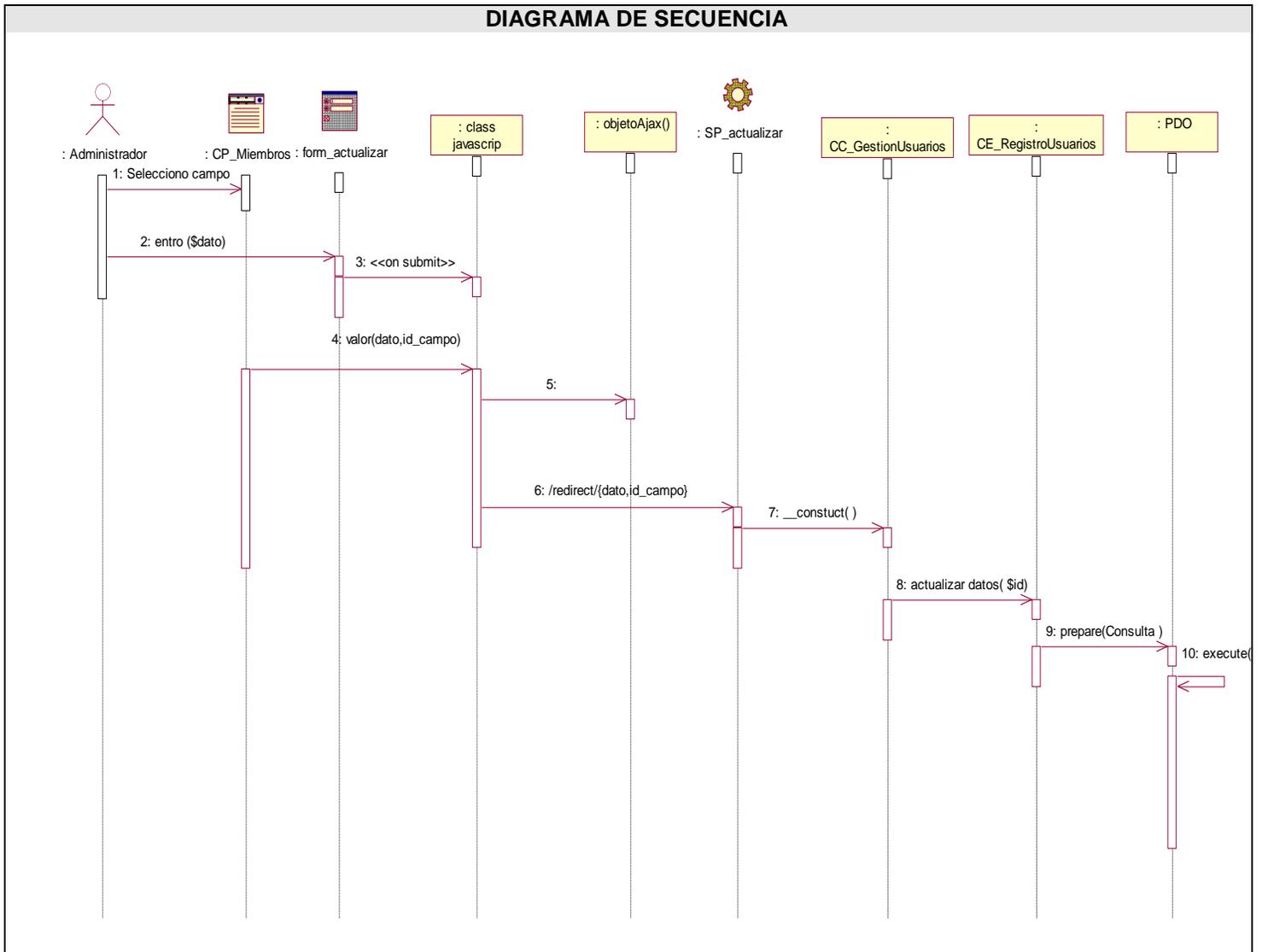


Diagrama de Secuencia: Gestionar Usuarios ('Eliminar Usuario').

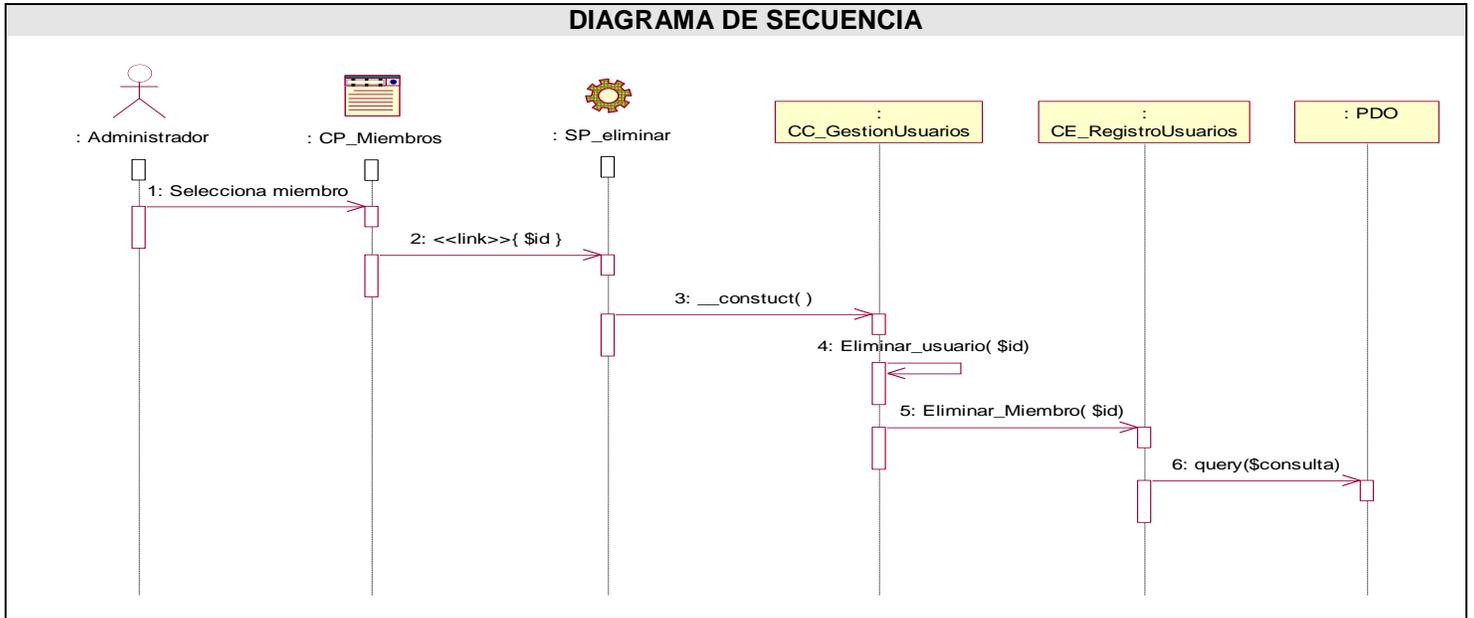


Diagrama de Secuencia: Gestionar Actividades ('Insertar Actividad').

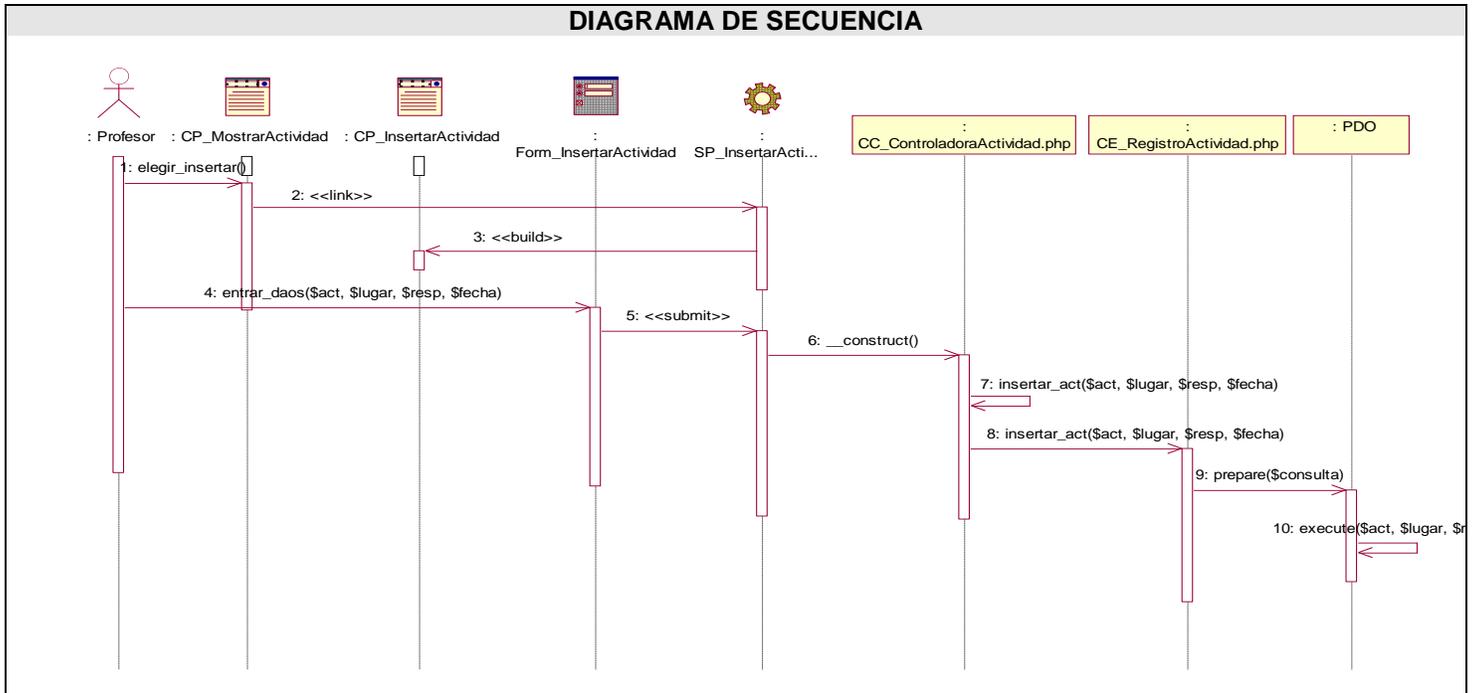


Diagrama de Secuencia: Gestionar Actividades ('Actualizar Actividad').

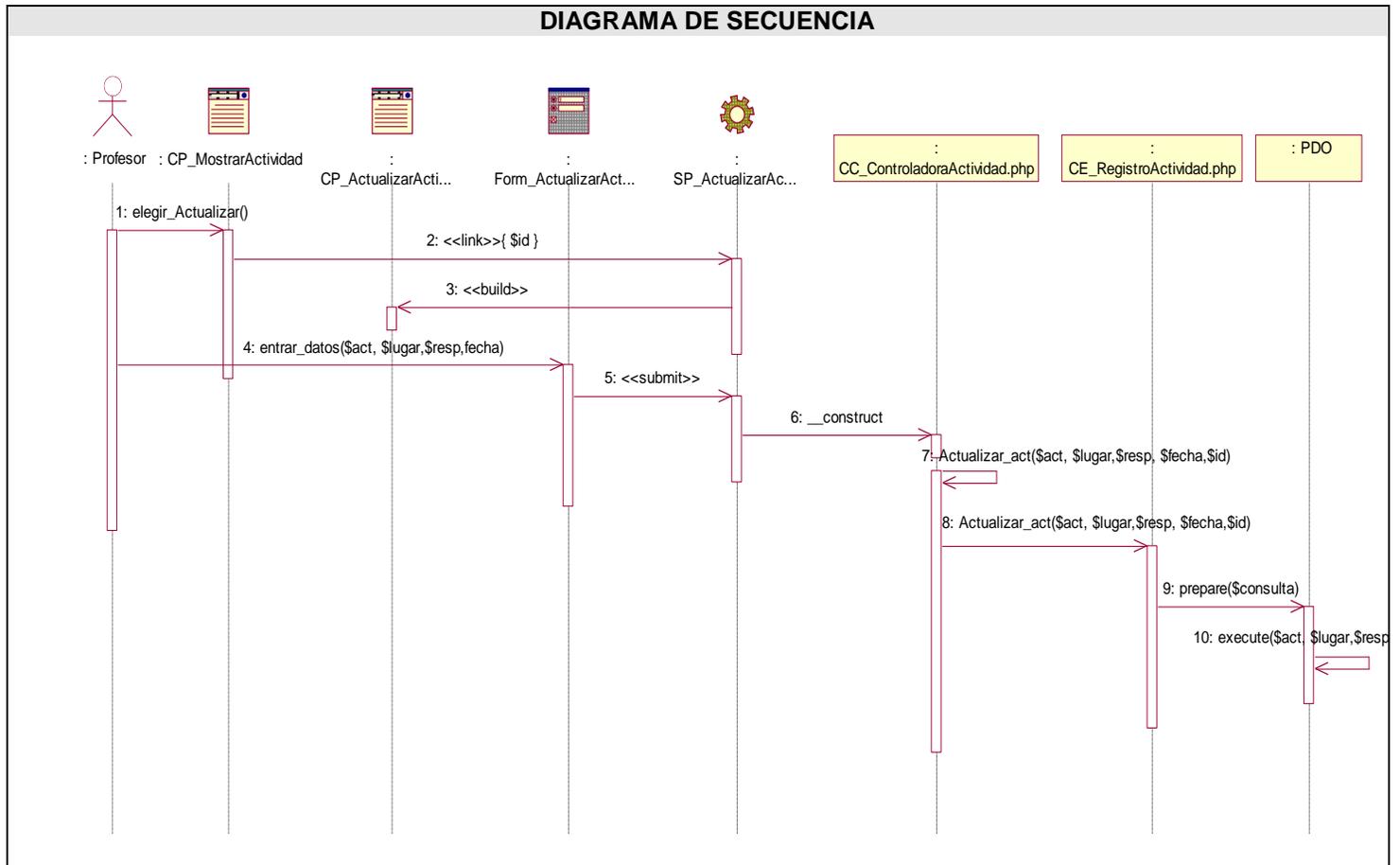


Diagrama de Secuencia: Gestionar Bibliografía ('Insertar Bibliografía').

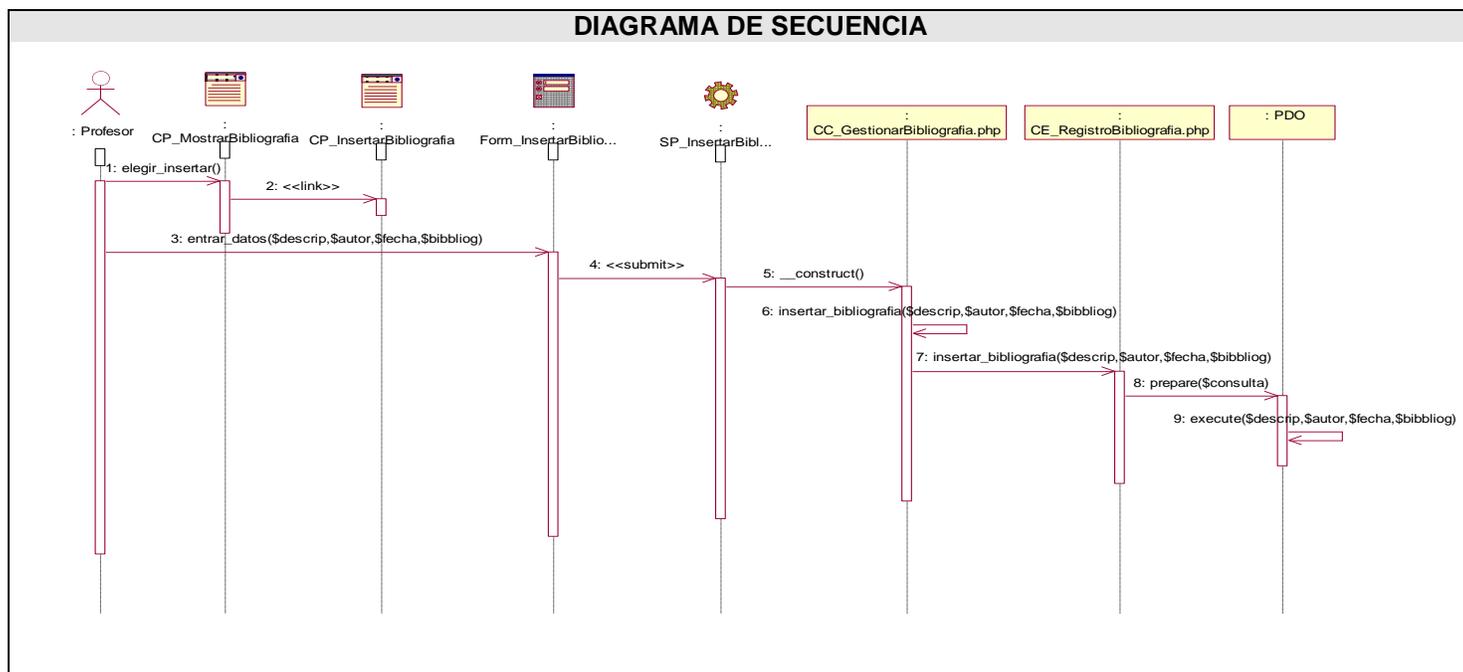


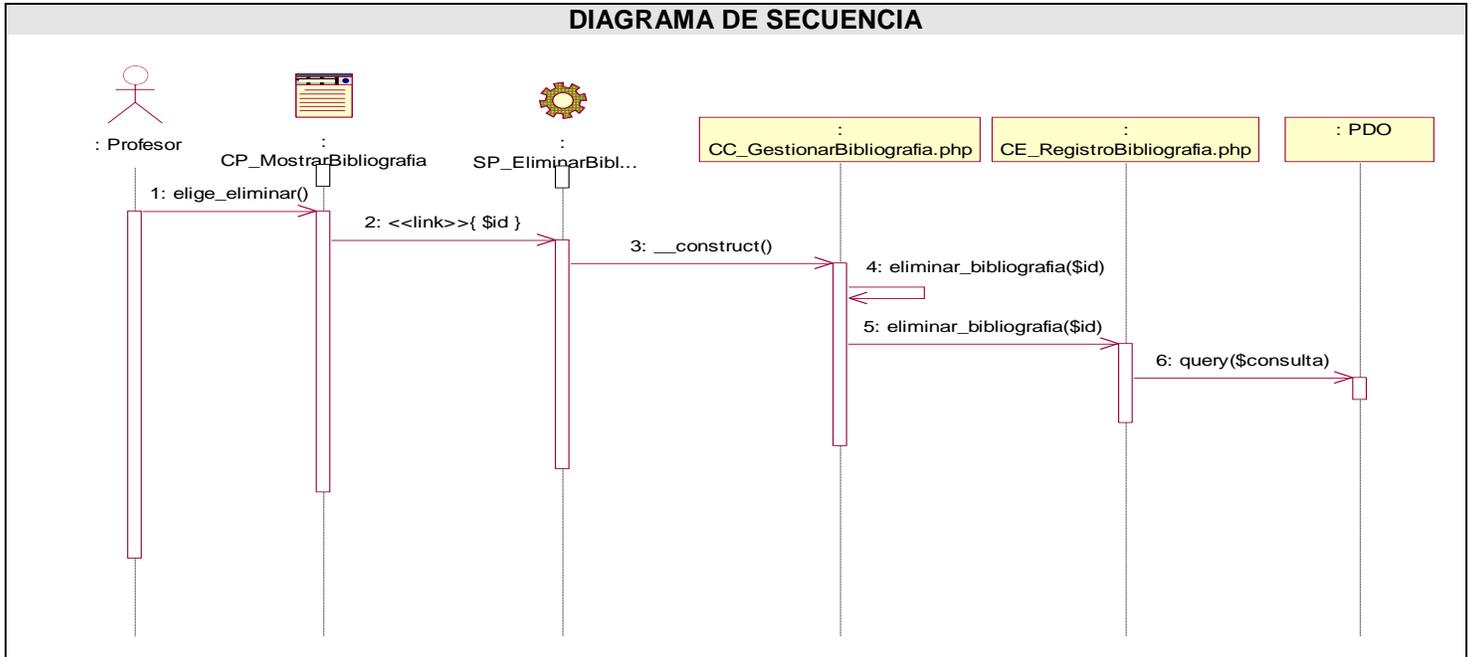
Diagrama de Secuencia: Gestionar Bibliografía ('Eliminar Bibliografía').

Diagrama de Secuencia: Gestionar Noticias ('Insertar Noticia').

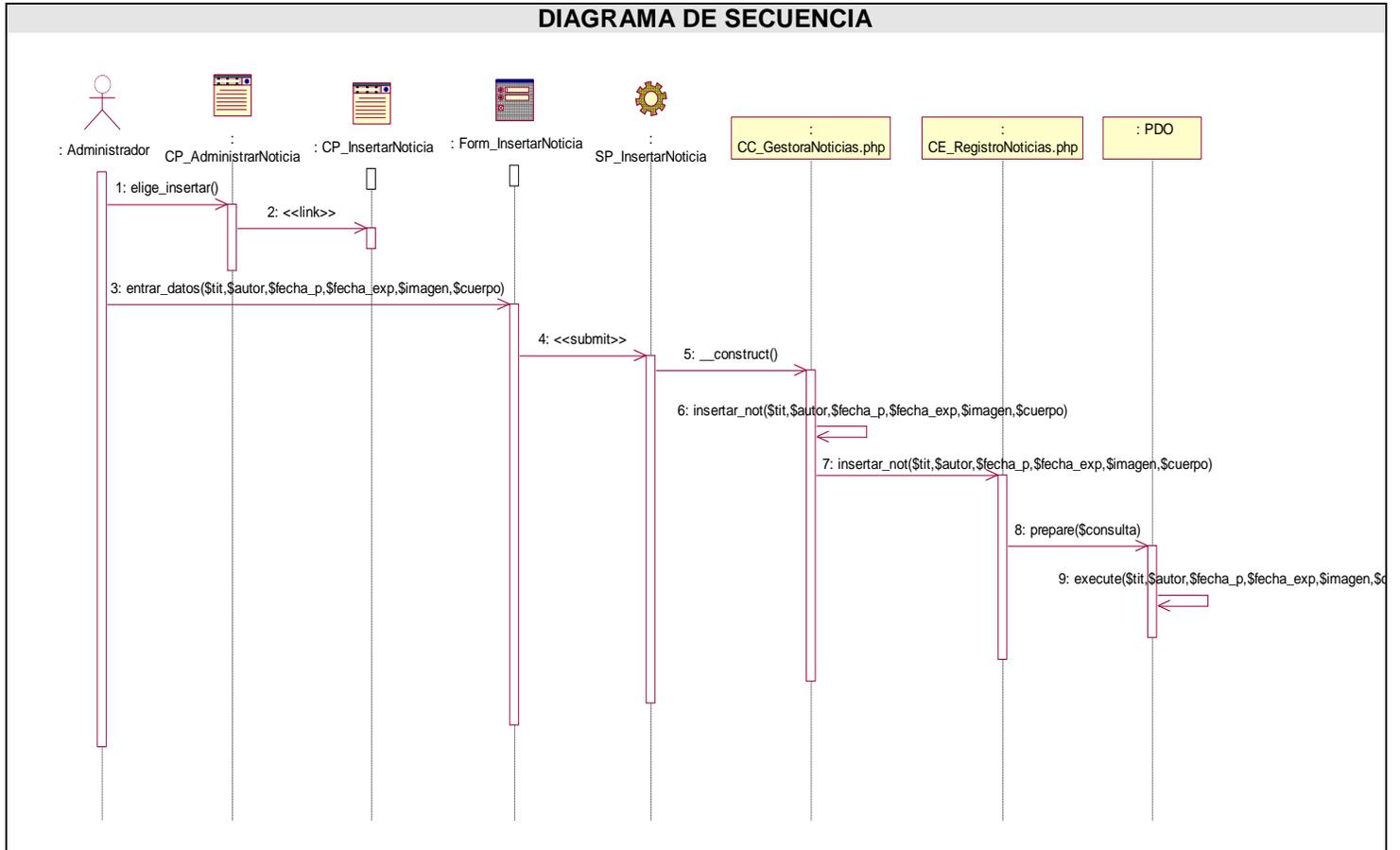


Diagrama de Secuencia: Gestionar Noticias ('Actualizar Noticia').

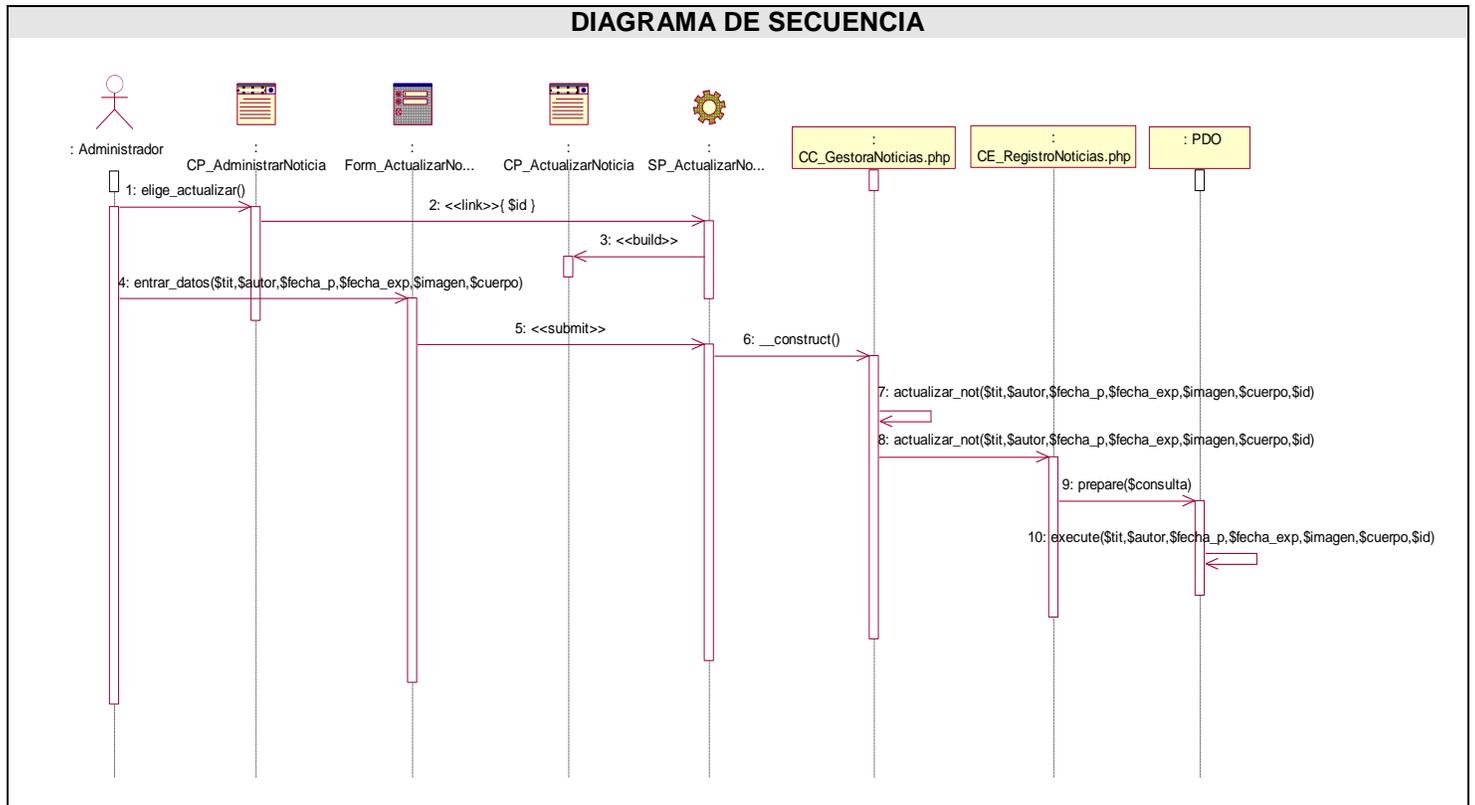


Diagrama de Secuencia: Gestionar Noticias ('Eliminar Noticia').

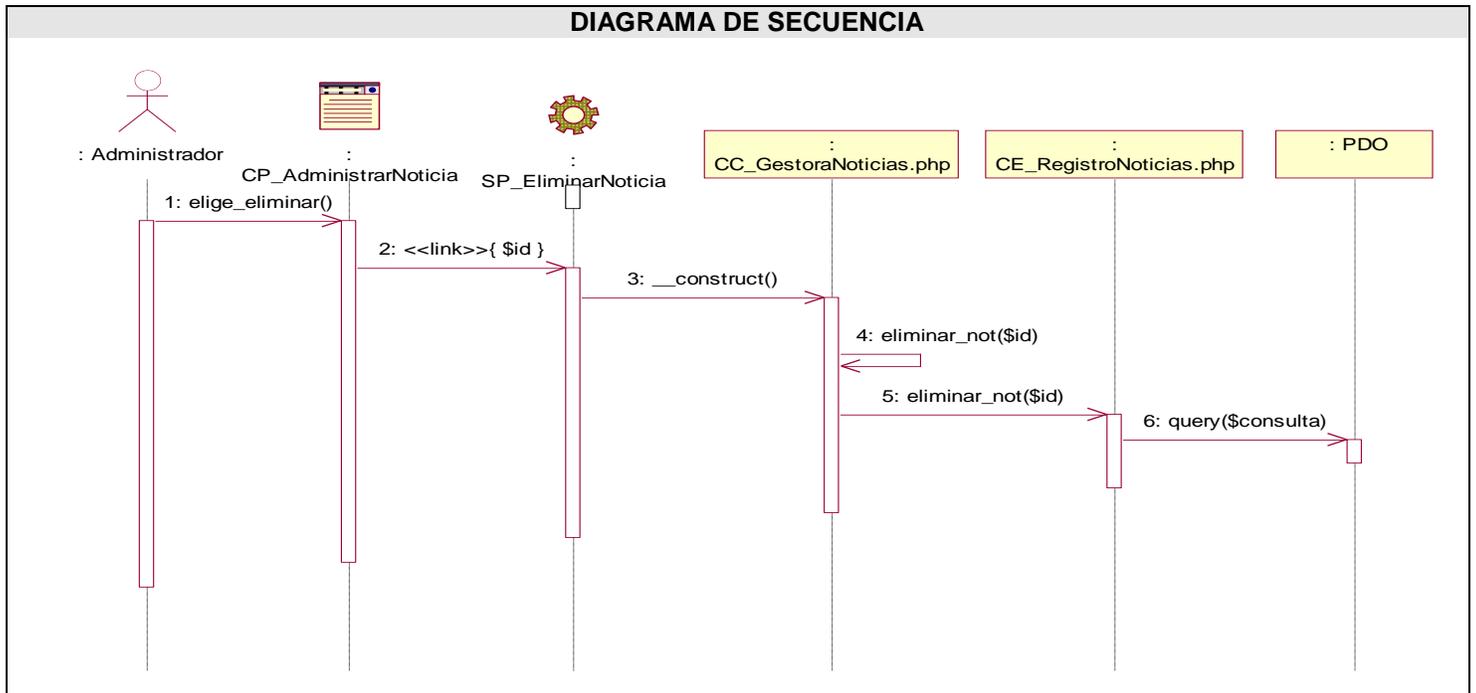


Diagrama de Secuencia: Gestionar Trabajos Científicos ('Insertar Trabajo').

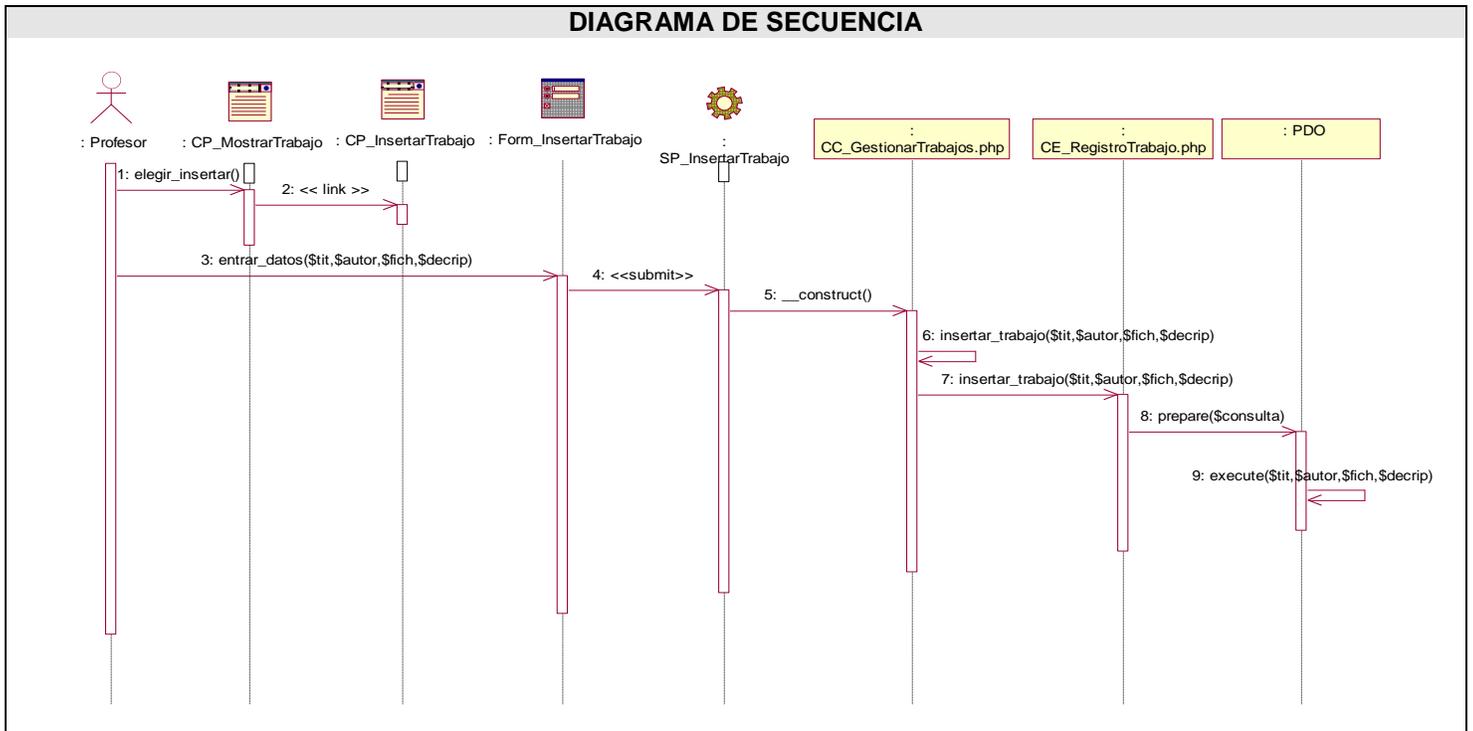


Diagrama de Secuencia: Gestionar Trabajos Científicos ('Eliminar Trabajo').

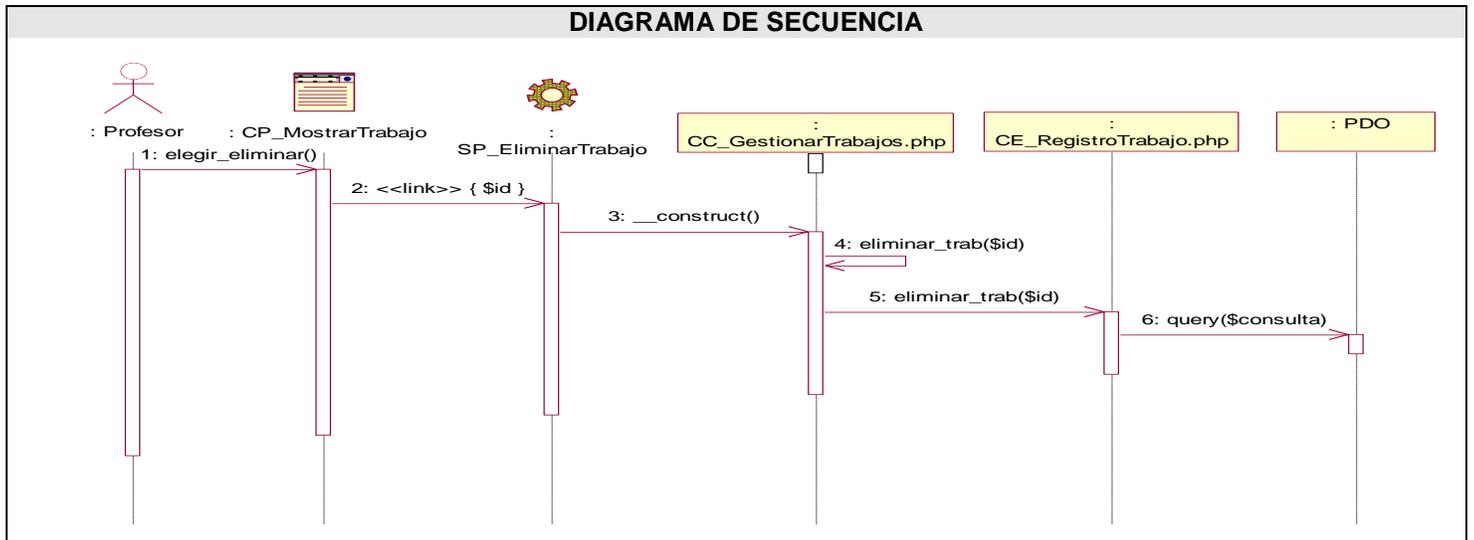


Diagrama de Secuencia: Listar Miembros ('Listar').

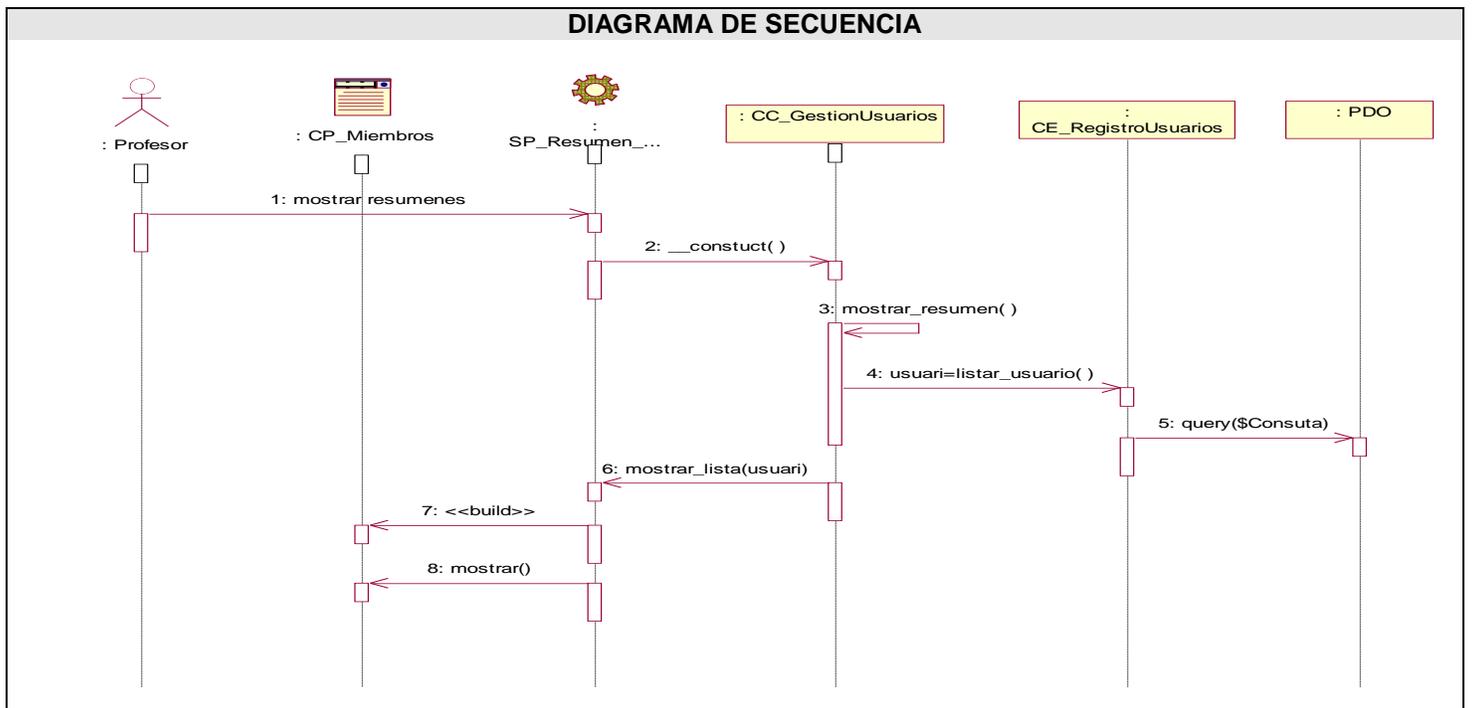
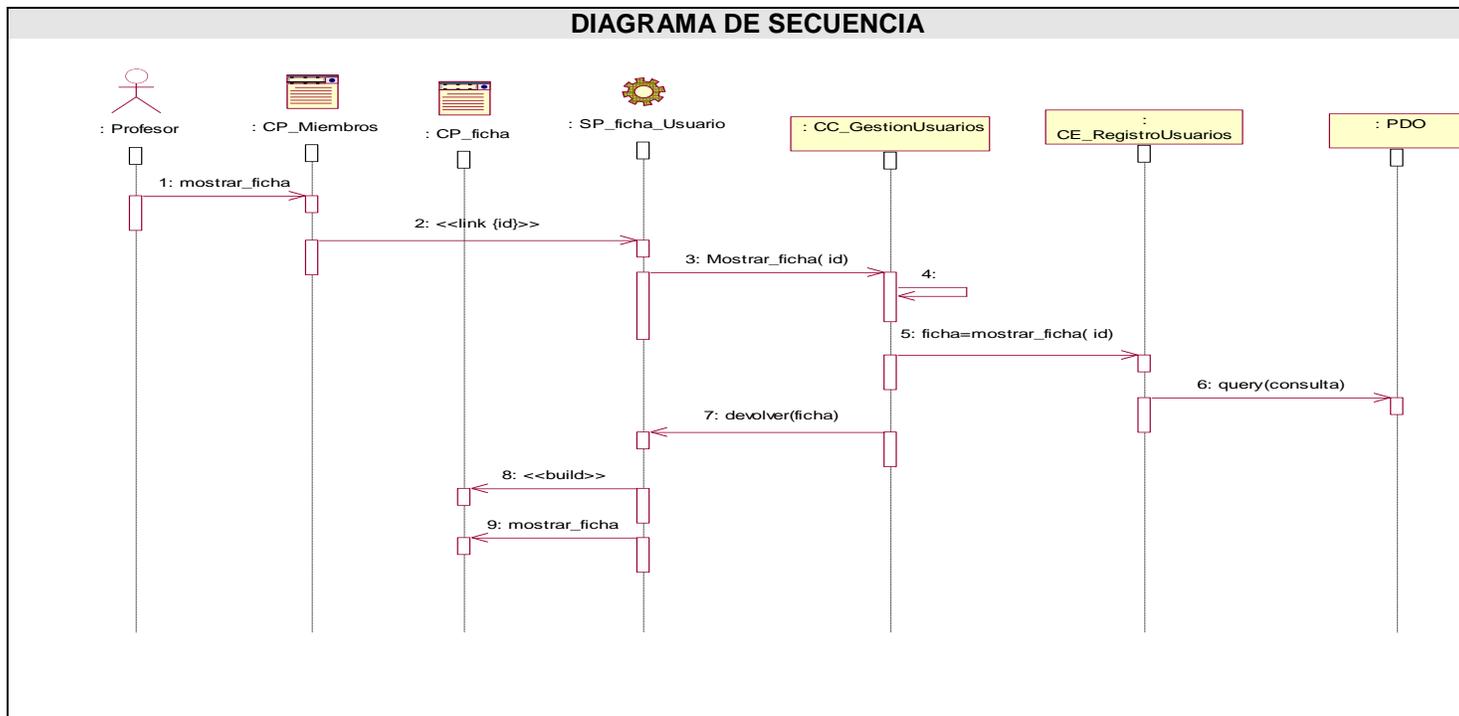


Diagrama de Secuencia: Listar Miembros ('Mostrar Ficha').



Anexo IV descripción de las tablas de la base de datos.

Nombre: T_Actas		
Descripción: En esta tabla se almacena las actas de la reuniones realizadas por el Departamento de Investigaciones		
Atributo	Tipo	Descripción
idacta	int	Este atributo es el que identifica la tabla pk
fecha_pub	date	Aquí se guarda la fecha en que fue subida el acta.
cuerpo	longtext	Este campo contiene el texto del acta
autor	varchar	Aquí se guarda el nombre del miembro que realizo y subió el acta.

Nombre:T_Actividad		
Descripción: Esta tabla contiene todas las actividades planificadas por el grupo mensualmente.		
Atributo	Tipo	Descripción
Id	int	Este atributo es el que identifica la tabla pk
nomb_actividad	Varchar	Se guarda el nombre la actividad
responsable	Varchar	Nombre del responsable de que la actividad se realice.

lugar	Varchar	Nombre del lugar donde se llevara acabo.
fecha	datetime	Fecha que se realizara la actividad
descripción	longtext	Resumen de la actividad.

Nombre: T_Bibliografías		
Descripción: Esta tabla un grupo de bibliografías pertenecientes al grupo para uso común de los integrantes y puestas a disposición del que la necesite		
Atributo	Tipo	Descripción
idbib	Int	Este atributo es el que identifica la tabla pk
autor	varchar	Nombre del profesor que subió la bibliografía.
fecha_consultada	date	Fecha en que fue consultada dicha bibliografía.
fecha_public	date	Fecha en que se subió la bibliografía al sistema.
fichero	varchar	Campo que contiene la dirección del lugar de almacenamiento del fichero.
descripción	varchar	Descripción del tema bibliográfico.

Nombre: T_Comentarios		
Descripción: Esta tabla guarda los comentarios realizados por lo usuarios a cerca de una noticia dada		
Atributo	Tipo	Descripción
idcom	Int	Este atributo es el que identifica la tabla pk
idnot	Int	Identificador que refleja a que noticia pertenece dicho comentario.
usuario	varchar	Nombre del usuario que realizo dicho comentario.
fecha_pub	date	Fecha en que fue publicado dicho comentario.
cuerpo	longtext	El texto del comentario.

Nombre:T_Noticia		
Descripción: Esta tabla contiene las noticias publicadas por el grupo de investigaciones		
Atributo	Tipo	Descripción
idnot	int	Este atributo es el que identifica la tabla pk
titulo	varchar	Titulo de la noticia
autor	varchar	Nombre del autor que la diseño.
cuerpo	longtext	Texto de la noticia.
fecha_pub	Date	Fecha de publicación en el sitio.
fecha_exp	date	Fecha de vencimiento de la noticia.
imagen	varchar	Dirección de almacenamiento de la imagen de la noticia.

Nombre: T_Trabajo		
Descripción: Tabla donde se almacenan todos los trabajos relacionados con el grupo de investigaciones		
Atributo	Tipo	Descripción
Id	int	Este atributo es el que identifica la tabla pk
titulo	varchar	Titulo del trabajo.
autor	varchar	Nombre del profesor que subió el trabajo al sistema

		(Los autores del trabajo se reflejan en el cuerpo del mismo).
fecha	date	Fecha de publicación.
fichero	varchar	Dirección del fichero donde se encuentra el trabajo.
descripción	longtext	Descripción del contenido del trabajo.

Nombre:T_Usuario		
Descripción: Esta tabla contiene la lista de los miembros del departamento de investigaciones con una serie de datos		
Atributo	Tipo	Descripción
nombre	varchar	Nombre con dos apellidos de miembro
rol	Varchar	
foto	Varchar	Dirección de almacenamiento de la imagen del usuario
usuario	Varchar	Usuario del dominio uci
id	int	Este atributo es el que identifica la tabla pk
categoría	Varchar	Categoría docente
grado	Varchar	Grado científico
institución	Varchar	Institución donde labora
correo	Varchar	El Correo de la uci del miembro
cargo	Varchar	Responsabilidad del miembro
curriculun	Text	Su currículum vitae
línea	Varchar	Línea investigativa a la que pertenece

GLOSARIO DE TÉRMINOS.

Bugs: Un bug es un error o un defecto en el software o hardware que hace que un programa funcione incorrectamente, o haga cosas no queridas. A menudo los bugs son causados por conflictos del software cuando las aplicaciones intentan funcionar de forma diferentes.

HTTP (Protocolo de transferencia de hipertexto): Protocolo de Internet que se utiliza para proporcionar información mediante la World Wide Web.

SGBD (Sistema de gestión de base de datos): Son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de bases de Datos y Database Management System, su expresión inglesa.

DOM: El Modelo de Objetos del Documento (DOM) es una interfaz de programación de aplicaciones API para documentos validos HTML y bien construidos XML. Define la estructura lógica de los documentos y el modo en que se accede y manipula.

Con el Modelo de Objetos del Documento, los programadores pueden construir documentos, navegar por su estructura, y añadir, modificar, o eliminar elementos y contenido. Se puede acceder a cualquier cosa que se encuentre en un documento HTML o XML, modificando, borrando o añadiendo utilizando el Modelo de Objetos del Documento.

InterBase: Es un sistema de administración de base de datos relacionales (RDBMS) desarrollada y comercializada por la compañía Borland Software Corporation. Se destaca de otros por su bajo consumo de recursos, su casi nula necesidad de administración y su arquitectura multi-generacional. InterBase corre en plataformas Linux, Microsoft Windows y Sistemas Operativos Solaris.