

# Universidad de las Ciencias Informáticas



## “Sistema de Gestión de Opiniones de la Facultad Regional de la provincia de Ciego de Ávila”

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

**Autores:** Yadainy Betancourt Rodríguez

Leosdany Sánchez Alba.

**Tutora:** Ing. Yannia Moreira Gamboa.

**Co-Tutor:** Ing. Yoandry Pacheco Águila.

Ciudad de la Habana, julio del 2007

## **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 \_\_\_\_\_ del año\_\_\_\_\_.

---

Yadainy Betancourt Rodríguez

---

Leosdany Sánchez Alba

---

Ing. Yannia Moreira Gamboa





## **Datos de Contacto**

Tutora: Ing. Yannia Moreira Gamboa

Clasificación: Profesional.

Clasificación del área de desarrollo: Dirección de Producción 5.

Síntesis de la Tutora: Ingeniera Informática, dos años de graduada.

Co-Tutor: Lic. Yoandry Pacheco Águila

Clasificación: Profesional.

Síntesis del Co-Tutor: Licenciado en Ciencias de la Computación, dos años de graduado.

## **AGRADECIMIENTOS**

### **De Yadainy:**

Resumir en algunos nombres de personas e instituciones todo el agradecimiento por el alcance de un sueño es una tarea bastante difícil, quizás más que su realización. En este empeño por crecer han estado involucradas muchas personas que han sido el sustento y aliento diario a no cejar en el empeño. En primer término quiero mencionar a mi hermana, a quien amo y para quien trato de ser siempre un ejemplo, espero no fallar en mi empeño Tata. A mi padre a quien no solo agradezco sus locuras, sus consejos y el apoyo, sino, y sobre todas las cosas, el privilegio de ser su hija. A mi hermano Héctor, a mis abuelos Bella y Macho y en especial a mis tíos Belkis y Alfredo, por ayudarme a realizar este sueño, por estar presentes en todo momento y contribuir a la culminación de mis estudios. A Misleidy González Marín y su familia por todo el apoyo y el cariño que me han dado. A Marlon E. Agustino Legón por estar ahí cuando más lo necesité. A Marcial por toda su comprensión y por esa manera tan especial de quererme. A los miembros de " La Cueva ", insustituibles amigos, y compañeros de muchas noches de alegrías, tristezas y días de fiestas. A mis muchos profesores guías en mi preparación. A todos los amigos y amigas que compartieron trabajo, amor, sonrisas, consejos y momentos inolvidables en la FEU. A los trabajadores de la Facultad Regional de Ciego de Ávila con los que tuve el privilegio de compartir los últimos meses de mi vida como estudiante, en especial a Grisel por todo lo que me ha enseñado, por su experiencia, por sus consejos, por sus regaños, por su cariño y por todo el apoyo que me ha dado. A mi tutora Yannia Moreira Gamboa, a Reynier por la ayuda en el diseño y a mi compañero de tesis Leosdany Sánchez Alba que es lo máximo.

**De Leosdany:**

Imposible sería para mí agradecer a todos los que de una manera u otra me han ayudado a la culminación de tantos años de estudio y esfuerzos. La relación tiene que necesariamente comenzar por La Revolución, especialmente por nuestro comandante en jefe Fidel Castro Ruz por permitirme alcanzar lo que una vez soñé. Llegue igualmente este profundo agradecimiento a mi abuelita Rafaela, a mi hermano, por su ejemplo y a mi familia que siempre me apoyó en todo. Agradezco también a todos mis compañeros de estudio, especialmente a los que compartieron conmigo estos últimos meses. A Nilda y a Julito, por dedicarme gran parte de su tiempo. A mis tutores Yannia y Pacheco, por la ayuda que siempre nos brindaron. A mis profesores de la UCI, por todo lo que de ellos he aprendido. No podría dejar de mencionar a todo el colectivo de la Facultad Regional de Ciego de Ávila, en especial a la decana Grisel, por sus consejos y por haber estado siempre pendientes de nosotros y por supuesto, a mi compañera de tesis, Yadainy, por tantas noches de desvelo.

En fin, para no olvidar a nadie, a todos, muchas gracias.

## **DEDICATORIA**

### **De Yadainy:**

A mi madre a quien la vida no le dio la oportunidad de verme graduada, pero ha estado conmigo en cada uno de mis días, este era uno de sus sueños y donde quiera que esté se sentirá orgullosa de verlo cumplido. A mi abuela María y a mi abuelo Osvaldo, por haberme dado su amor incondicional, por convertirse en madre y padre para mí cuando fue necesario, por dedicarme su vida, por tantos años de empeño y esfuerzo en mi educación y por lo orgullosos que se sienten al verme graduada; es para ellos con todo mi corazón. Finalmente quiero dedicar este trabajo a la Revolución, por haber acercado tanto, tan solo a la distancia del empeño, los sueños de muchos jóvenes que, como yo, están convencidos que el futuro se construye con nuestras propias manos.

### **De Leosdany:**

Le dedico este trabajo a ustedes, Mami y Papi: ante todo porque los quiero, aunque ya no se los diga, porque se que padecen cuando yo estoy triste, que sufren cuando yo lo hago, y que comparten mis alegrías en todo momento. A ustedes, porque les debo todo lo que hoy soy, y de que en estos momentos me encuentre preparándome para convertirme en ingeniero, porque siempre están conmigo, en las buenas y en las malas; me educan, me aconsejan, me imparten valores para conducirme correctamente y me ofrecen el sabio consejo en el momento oportuno.

Hoy, me siento muy satisfecho por la virtud de tener unos grandes padres como ustedes y de poder darles el orgullo de verme finalmente graduado.



## RESUMEN

La Universidad de las Ciencias Informáticas (UCI), basada en su estrategia de desarrollo, está enfrascada en el proyecto UCI Ciudad Digital, cuyo objetivo es crear una micro-sociedad completamente informatizada, que pudiera constituir el modelo de sociedad cubana del futuro. Como parte de esta misión se han realizado diferentes proyectos para la informatización de los servicios que se brindan en el centro. Con el surgimiento de las Facultades Regionales (FR) de la Universidad de las Ciencias Informáticas (UCI), se decide transmitir experiencias e implantar aplicaciones que se utilizan para gestionar un grupo de actividades en la sede universitaria, dentro de las cuales se encuentra el Sistema de Gestión de Opiniones (SGO).

El sistema actual presenta deficiencias en algunas de sus funcionalidades básicas; como es el proceso de revisión de las opiniones, la actualización del estado de las mismas, la ausencia de reportes tanto gráficos como estadísticos y la publicación simultánea de varios temas de debates y por este motivo, se decidió desarrollar un nuevo sistema.

El presente trabajo exhibe los resultados de la creación del nuevo Sistema de Gestión de Opiniones que persigue objetivos principales como controlar el proceso de registrar, editar y revisar las opiniones de los estudiantes, profesores y trabajadores de la FR de la UCI, además de brindar información relacionada con los temas, los usuarios y las estadísticas que permitan hacer un estudio del comportamiento de esta actividad y contribuyendo de esta forma a gestionar de manera más eficiente todo el flujo de opiniones que se emiten.

## PALABRAS CLAVES

Sistema de Gestión de Opiniones, opiniones, revisar, editar.

## ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO 1 .....	4
FUNDAMENTACIÓN TEÓRICA .....	4
1.1 INTRODUCCIÓN. ....	4
1.2 APLICACIONES PARA LA GESTIÓN DE OPINIONES. TENDENCIAS ACTUALES. ....	4
1.4 APLICACIONES WEB COMO SOPORTE DE LOS SISTEMAS DE OPINIONES. ....	7
1.5 MODELO CLIENTE SERVIDOR. ....	8
1.6 FUNDAMENTACIÓN DE LA TECNOLOGÍA. ....	10
1.7 CONCLUSIONES. ....	25
CAPÍTULO 2.....	26
CARACTERÍSTICAS DEL SISTEMA .....	26
2.1. INTRODUCCIÓN. ....	26
2.2. OBJETO DE ESTUDIO. ....	26
2.2.1 OBJETIVOS ESTRATÉGICOS DE LA ORGANIZACIÓN. ....	26
2.2.2 FLUJO ACTUAL DE LOS PROCESOS. ....	27
2.2.3 SITUACIÓN PROBLÉMICA. ....	27
2.3 OBJETO DE AUTOMATIZACIÓN. ....	28
2.3.1 SISTEMAS EXISTENTES VINCULADOS CON EL CAMPO DE ACCIÓN. ....	29
2.4 INFORMACIÓN QUE SE MANEJA. ....	30
2.5 PROPUESTA DEL SISTEMA. ....	31
2.6 MODELO DE NEGOCIO. ....	32
2.6.1 DEFINICIÓN DE LOS ACTORES DEL MODELO DE NEGOCIO. ....	32
2.6.2 DEFINICIÓN DE LOS TRABAJADORES DEL MODELO DE NEGOCIO.....	33
2.6.3 CASOS DE USO DEL NEGOCIO.....	33
2.6.4 DIAGRAMA DE CASOS DE USO DEL NEGOCIO. ....	33
2.6.5 CASOS DE USO DEL NEGOCIO Y DIAGRAMA DE ACTIVIDADES.....	34
2.6.6 DIAGRAMA DE CLASES DEL MODELO DE OBJETO DEL NEGOCIO.....	35

2.7. ESPECIFICACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA. ....	35
2.7.1. REQUERIMIENTOS FUNCIONALES.....	35
2.7.2. REQUERIMIENTOS NO FUNCIONALES.....	38
2.8 DEFINICIÓN DE LOS CASOS DE USO.....	40
2.9 CONCLUSIONES.....	40
CAPÍTULO 3.....	41
ANÁLISIS Y DISEÑO DEL SISTEMA.....	41
3.1 INTRODUCCIÓN.....	41
3.2 ANÁLISIS DEL SISTEMA.....	41
3.2.1 MODELO DE CLASES DEL ANÁLISIS.....	41
3.3 ANÁLISIS DE DISEÑO.....	46
3.3.1 DIAGRAMAS DE INTERACCIÓN.....	46
3.3.2 DIAGRAMA DE DISEÑO WEB DEL SISTEMA.....	52
3.3.3 DESCRIPCIÓN DE LAS CLASES.....	62
3.3.4 DISEÑO DE LA BASE DE DATOS.....	72
3.3.4.1 DIAGRAMA ENTIDAD RELACIÓN DE LA BASE DE DATOS.....	72
3.3.4.2 DESCRIPCIÓN DE LAS TABLAS.....	73
3.4 SEGURIDAD.....	78
3.4.1 ENCRIPCIÓN DE CONTRASEÑAS.....	78
3.4.2 CONTRASEÑAS DE UN SOLO USO (OTP).....	79
3.4.4 SEGURIDAD DEL SISTEMA.....	80
3.5 CONCLUSIONES.....	80
CAPÍTULO 4.....	81
IMPLEMENTACIÓN Y PRUEBA.....	81
4.1 INTRODUCCIÓN.....	81
4.3 DIAGRAMA DE COMPONENTES.....	82
4.4 MODELO DE PRUEBA.....	90
4.4.1 NIVELES DE PRUEBA.....	91

4.4.2 TIPOS DE PRUEBA. ....	91
4.4.3 MÉTODOS DE PRUEBA. ....	92
4.4.4 CASO DE USO AUTENTICAR USUARIO. ....	93
4.4.5 CASO DE USO GESTIONAR TEMA.....	94
4.4.6 CASO DE USO INSERTAR OPINIÓN.....	97
4.4.7 CASO DE USO REVISAR OPINIÓN. ....	97
4.4.8 CASO DE USO APROBAR OPINIÓN. ....	98
4.4.9 CASO DE USO GESTIONAR OPINIONES ARCHIVADAS.....	99
4.4.10 CASO DE USO GESTIONAR USUARIO. ....	101
4.4.11 CASO DE USO GESTIONAR REPORTES.....	103
4.4.12 CASO DE USO GESTIONAR LOGS. ....	104
4.4.13 CASO DE USO GESTIONAR ESTADÍSTICAS.....	104
4.5 CONCLUSIONES. ....	105
CONCLUSIONES .....	106
RECOMENDACIONES .....	107
REFERENCIAS BIBLIOGRÁFICAS.....	108
BIBLIOGRAFÍA .....	109
ANEXOS .....	111
GLOSARIO .....	128

## INTRODUCCIÓN

La Revolución Cubana desde sus inicios se ha caracterizado por mantener informada a la población a través de los distintos medios de comunicación del acontecer nacional e internacional. Constituye una de sus principales prácticas, a través de los órganos del Partido, retroalimentarse de los criterios y observaciones del pueblo sobre los diversos temas que acontecen. La Universidad de las Ciencias Informáticas como parte del programa de La Batalla de Ideas es una de las fuentes de retroalimentación con que cuenta el país, pues en esta cursan estudios jóvenes de todos los municipios, permitiendo así conocer desde el recinto universitario opiniones de todos los rincones de la isla.

En el año 2003 la Dirección de la Revolución cree necesario recoger opiniones sobre el suceso de violación del espacio aéreo. De esta forma y depositando plena confianza en los estudiantes, profesores y trabajadores de la UCI se recogen sus opiniones, manteniendo siempre la idiosincrasia que los identifica, y estas son llevadas al Comandante.

Inicialmente las opiniones eran expresadas por escrito, un grupo de compañeros se encargaban de recogerlas por las aulas y apartamentos para después procesar la información. Luego se habilitó una cuenta de correo a la que podían enviar sus criterios. Se organizaron laboratorios con representantes de la FEU, la UJC y el PCC para el procesamiento de las opiniones que se encontraban en esta cuenta.

La solución que se había buscado para facilitar la revisión de las opiniones se tornaba engorrosa. Por esta razón la Dirección del PCC de la Universidad toma la decisión de informatizar este proceso buscando eficiencia a la hora de gestionar las opiniones de la comunidad y recoger la mayor cantidad de criterios en el menor tiempo posible, el resultado obtenido fue una aplicación llamada Sistema de Gestión de Opiniones (SGO); implementada en ASP y como Sistema Gestor de Base de Datos (SGBD), SQL Server.

En la actualidad la Universidad cuenta con un sistema donde los estudiantes, profesores y trabajadores opinan sobre temas de carácter nacional e internacional de interés para la Máxima Dirección del país. Esta aplicación ha permitido organizar y agilizar un poco el trabajo con las opiniones pero presenta dificultades en algunas de sus funcionalidades básicas que frenan el dinamismo con que se deben procesar las mismas.

Con el surgimiento de las Facultades Regionales (FR) de la Universidad de las Ciencias Informáticas (UCI), se decide transmitir experiencias e implantar aplicaciones que gestionan un grupo de actividades en la sede universitaria, dentro de las cuales se encuentra el Sistema de Gestión de Opiniones (SGO).

Debido a la necesidad de mejorar todas las actividades que involucra la gestión de las opiniones, con el objetivo de que lleguen en el tiempo requerido a la Máxima Dirección del país, se decide implementar un nuevo SGO para ser instalado en las FR que rectifique todos los errores que presenta el sistema actual. El nuevo sistema debe presentar una interfaz amigable, de fácil navegación, que cumpla con los estándares de diseño definidos por la Universidad, debe ser capaz de mostrar las estadísticas, tanto en texto plano como mediante gráficas, para su posterior revisión y estudio. Además, se desarrollará en PHP, aprovechando así las características de este lenguaje atractivo y respaldado por una gran comunidad de desarrolladores ya que tiene entre sus principales peculiaridades ser independiente de la plataforma y emplear las técnicas de Programación Orientada a Objetos.

La utilización de código abierto en el Sistema de Gestión de Opiniones parte de la política que ha seguido nuestro país de emigrar a software libre. La Universidad no está ajena a ello, dentro de sus objetivos se encuentra fomentar el estudio, la investigación así como la producción de software sobre soluciones de código abierto teniendo en cuenta las facilidades que brinda de libre distribución: todo el mundo tiene derecho a usarlas, acceder a ellas, aprender e incluso modificarlas.

Actualmente no se cuenta con una aplicación para la Gestión de las Opiniones que cumpla con los requerimientos necesarios del cliente para ser usada en las Facultades Regionales (FR) de la Universidad de las Ciencias Informáticas (UCI), siendo este el problema científico a resolver.

Serán objeto de estudio los procesos que intervienen en la gestión de opiniones cuyo **campo de acción** será el Sistema de Gestión de Opiniones en las Facultades Regionales de la Universidad de las Ciencias Informáticas. Encaminados a cumplir el **objetivo general** de desarrollar un sistema informático que permita gestionar los procesos asociados con la Gestión de Opiniones en las Facultades Regionales, específicamente en la Provincia de Ciego de Ávila.

Para alcanzar el objetivo enunciado anteriormente se han definido los siguientes **objetivos específicos**:

- Realizar el análisis y diseño del sistema.
- Implementar un mecanismo de gestión de reportes dinámicos.
- Implementar un sistema para gestionar las opiniones.

Para cumplir estos objetivos se han llevado a cabo las siguientes tareas de investigación:

- Realización de entrevistas con los directivos del PCC de la UCI para definir los objetivos del trabajo e identificar las necesidades de los usuarios.
- Caracterización del proceso de gestión de opiniones en la Universidad de las Ciencias Informáticas.
- Desarrollo del estado del arte de los sistemas de opiniones.

- Selección y fundamentación de las herramientas para llevar a cabo el proyecto.
- Diseño de la Base de Datos.
- Puesta en marcha de la aplicación Web.
- Desarrollo de las pruebas a la aplicación en el ambiente de trabajo.

Se pretende finalmente obtener un producto a la altura de las exigencias actuales de producción de software en nuestra sociedad, acorde con los estándares de diseño y presentación de aplicaciones Web de la Universidad de la Ciencias Informáticas.

Este trabajo está dividido en cuatro capítulos que contienen toda la información referente a la investigación realizada y al análisis y diseño del sistema a desarrollar, la información está distribuida de la siguiente manera:

**Capítulo I:** Fundamentación Teórica. En este capítulo se tratan las tendencias, técnicas, tecnologías, metodologías y software usados para la solución del problema.

**Capítulo II:** Características del Sistema. En este capítulo se presenta una propuesta del sistema, así como el modelo del negocio, la especificación de los requerimientos del software y definición de los casos de uso del sistema.

**Capítulo III:** Análisis y Diseño del Sistema: En este capítulo se define el modelo de análisis, modelo de clases de análisis, los diagramas de interacción, se describen las clases, se diseña la Base de Datos, se analizan las definiciones de diseño a aplicar, así como el tratamiento de errores, seguridad, interfaz y concepción de la ayuda.

**Capítulo IV:** Implementación y Prueba: En este capítulo se realiza el diagrama de despliegue, el diagrama de componente y las pruebas al software en el ambiente de trabajo.

# Capítulo 1

## Fundamentación Teórica

### **1.1 Introducción.**

Durante el presente capítulo se abordarán las tendencias, técnicas, tecnologías, metodologías y software usados para la solución del problema. Se realizará un breve estudio sobre los diferentes sistemas relacionados con nuestro trabajo en el ámbito internacional y nacional. Como segundo aspecto analizaremos las tecnologías propuestas por la Dirección de Informatización que se utilizarán para la realización de la aplicación Web así como las herramientas de desarrollo y finalmente las metodologías para el análisis y diseño del sistema.

### **1.2 Aplicaciones para la Gestión de Opiniones. Tendencias Actuales.**

Los Sistemas de Opiniones no son más que aplicaciones informáticas que tienen como función principal permitir recoger los criterios de múltiples usuarios sobre diferentes temas, son una vía rápida y fácil para transmitir una idea, criterio o valoración sobre determinada cuestión, desde cualquier lugar, permitiendo conocer en tiempo real y con claridad las ideas expuestas por los usuarios y realizar análisis a partir de las estadísticas y el contenido de las mismas.

Son muchos los soportes disponibles para crear un sistema de opiniones en el mundo. Por lo general están desarrollados en PHP, Perl, ASP o Java. Los datos y la configuración son almacenados por lo general en una base de datos o una serie de archivos de texto. Cada versión provee funciones o capacidades diferentes: los más básicos se limitan a las opiniones sólo con texto, los más avanzados facilitan la inclusión de multimedia, formato del texto, HTML u otros; estos varían dependiendo de los intereses con que se quieran utilizar.

Generalmente se encuentran incluidos en sitios Web de universidades, empresas y muchas otras entidades; donde las opiniones de los usuarios son las bases fundamentales para el perfeccionamiento del trabajo.



En el ámbito nacional como internacional existen numerosos espacios para opinar con fines comerciales, políticos y otros, pero un sistema que permita gestionar opiniones con las características que necesitamos es difícil de encontrar. Los sistemas de opiniones generalmente son aplicaciones sencillas, donde se registran y revisan las opiniones, además de generar pocos reportes estadísticos.

En el mundo existen diferentes vías para obtener opiniones a través de la Web:

**Foros:** Espacio virtual creados en Internet o en una Intranet, donde los usuarios pueden enviar y contestar mensajes que pueden ser leídos por otros usuarios, son también conocidos como foros de mensajes, de opinión o foros de discusión y son aplicaciones Web que brindan soporte a discusiones en línea. [1]

**Encuestas:** Es una herramienta para recolectar información a un grupo de personas mediante la elaboración de un cuestionario. Al hacer el cuestionario hay que formular preguntas que revelen realmente la información deseada. Existen diferentes tipos de encuestas las basadas en entrevistas cara a cara o de profundidad, telefónicas, postales y la que nos interesa abordar:

**Encuestas por Internet:** Este tipo de encuesta consiste en colocar un cuestionario en una página Web o en enviarlo a los correos electrónicos de un panel predefinido. Sus principales ventajas son: la amplia cobertura a la que se puede llegar, el ahorro de tiempo, los bajos costos y la utilización de medios audiovisuales durante la encuesta. Una de sus desventajas es que no siempre se puede verificar la identidad del encuestado. [2]

También existen otros tipos de sistemas de opiniones que no se definen formalmente como foros o encuestas pero si son muy utilizados a nivel mundial. En ocasiones se necesita recoger en los sitios Web determinados criterios sobre uno o varios temas de interés, que necesariamente no tienen que ser publicados en los foros; ni tampoco, regirse por un cuestionario a contestar como en las encuestas. Los formularios de opiniones, sugerencias o críticas, son páginas Web o parte de ellas realizadas atendiendo a las necesidades de las entidades, en las que el usuario completa y estas informaciones se devuelven al servidor para su procesamiento.

A nivel mundial los sitios Web cada día son más utilizados y se perfeccionan constantemente con la ayuda de las opiniones de sus usuarios a través de encuestas, foros o simplemente formularios que almacenan informaciones en bases de datos, con cierto grado de privacidad. En la búsqueda de información y utilizando las facilidades de Internet visitamos algunos sitios que contienen espacios para que los internautas puedan opinar.

## **Opinar**

Es un periódico electrónico de la organización de periodistas en Internet que permite que los usuarios escriban sus opiniones y valoraciones a través de las diferentes secciones del sitio, por ejemplo:

La sección "Denuncia" está concebida para dar cauce a las quejas y denuncias de todo tipo de lectores. Para su publicación, es requisito imprescindible la plena identificación de la persona que envíe su queja o denuncia: nombre y apellidos, documento nacional de identidad o pasaporte, domicilio, y e-mail de contacto, y los únicos datos personales que se publican son el nombre, los apellidos, la ciudad y el país de residencia. Disponible en: <http://www.opinar.net/2006/diciembre/portada.htm>.

## **Dooyoo**

Es un sitio dedicado a la promoción de diferentes productos por ejemplo: Electrodomésticos, Informáticos, Audio, Fotografía, Material de Oficina entre otros. En algunas de sus secciones los usuarios opinan sobre los productos y experiencias en cuanto al uso de los mismos, estas son publicadas facilitando así la promoción de los mismos, además mediante el llenado de formularios se pueden dar valoraciones de los productos y otorgarle una puntuación a partir de categorías definidas. También permite evaluar las opiniones que se han emitido y hasta comentarlas. Disponible en: <http://www.dooyoo.es>.

## **Universidad Carlos III de Madrid**

La Universidad Carlos III de Madrid ha implantado un sistema de atención a sugerencias, quejas, felicitaciones y solicitudes de información. Este es un paso más en la mejora de los servicios desde la consideración en primera línea de la visión de sus usuarios: estudiantes, profesores, personal de administración, servicios y muchos otros ciudadanos.

En Cuba existen distintos sitios que han desarrollado espacios para recoger las opiniones de los usuarios, mediante sus ideas permiten que se lleven a cabo mejoras en el trabajo que realizan o en las funcionalidades que cumplen. Disponible en: <http://www.uc3m.es/uc3m/opina/opinaBiblioteca.html>.

## **Cienfuegos la perla del sur y Magon**

Estos sitios facilitan en sus secciones, opinar sobre la calidad de los portales y votar por la sección que el usuario considera más interesante, como una vía de retroalimentación para llevar a cabo sus mejoras. Disponibles en: <http://www.cienfuegos.cu/> y <http://www.magon.cu/mainpages/home.asp>.

## **Portal de la Ciencia en Matanzas**

Este Portal de la Ciencia en Matanzas tiene en una de sus secciones un foro de discusión que permite el intercambio de ideas y reflexiones relacionadas con el tema en debate, guiado por un moderador.

Disponible en: <http://www.atenas.inf.cu>.

## **Cubasi**

Otra de las formas de gestionar opiniones es a través de las encuestas, este sitio promueve un espacio que permite a los usuarios expresar sus preferencias en cuanto a lo que desean ver en el canal de música. Disponible en: <http://musica.cubasi.cu>.

### **1.4 Aplicaciones Web como soporte de los Sistemas de Opiniones.**

Sobre el servicio Web clásico podemos disponer de aplicaciones Web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Es necesario distinguir entre:

Aplicaciones en el lado del cliente: el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o JavaScript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts).

Aplicaciones en el lado del servidor: el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

**Las aplicaciones Web tienen varias ventajas sobre los programas de software descargables tradicionales:**

- No requieren instalación, pues usan tecnología Web, lo cual nos permite el aprovechamiento de todas las características del Internet.
- Son fáciles de usar (no requieren conocimientos avanzados de computación).
- Alta disponibilidad, ya que puede realizar consultas en cualquier parte del mundo donde tenga acceso a Internet y a cualquier hora.
- Compatibilidad multiplataforma. Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de escritorio. Varias tecnologías incluyendo Java, Flash, ASP y AJAX permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.
- Inmediatez de acceso. Las aplicaciones basadas en Web no necesitan ser descargadas, instaladas o configuradas. Usted accede a su cuenta y están listas para trabajar sin importar cuál es su configuración o su hardware.

- Menos requerimientos de memoria. Las aplicaciones basadas en Web. Al residir y correr en los servidores del proveedor, usan en muchos casos la memoria RAM de las computadoras en las que corren.
- Múltiples usuarios concurrentes. Las aplicaciones basadas en Web pueden realmente ser utilizadas por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta.

#### **Desventajas que presentan las aplicaciones WEB:**

- Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- La interactividad no se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera hasta que se obtiene la reacción del sistema.
- Diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.

## **1.5 Modelo Cliente Servidor.**

La arquitectura cliente-servidor es 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). Es decir dos o más procesos funcionan de forma independiente pero cooperativamente, una aplicación pide datos a la otra, una vez realizada la petición elabora la respuesta y la devuelve a la aplicación demandante.

La aplicación cliente es la que interactúa con el usuario, en esta se realizan las validaciones de los datos de entrada, se realizan las peticiones de datos a las aplicaciones servidoras, se verifica la lógica del negocio y consta de una interfaz gráfica.

La aplicación servidora, normalmente es un sistema gestor que se caracteriza por administrar la Base de Datos, verificar la integridad de la misma, gestionar su seguridad, ejecutar parte de la lógica de la aplicación y responder a las peticiones de las aplicaciones clientes.

La arquitectura tradicional de cliente/servidor también es conocida como arquitectura de dos capas. En este tipo de arquitectura la lógica del negocio reside en la PC cliente pero tiene que estar implementada en el servidor mediante los llamados disparadores o procedimientos almacenados. Los ambientes de dos capas requieren control excesivo de las versiones y demandan esfuerzo de distribución de la aplicación

cuando se les hacen cambios. Esto se debe al hecho de que la mayoría de la aplicación lógica existe en la estación de trabajo del cliente.

La seguridad del sistema en un diseño de dos capas es compleja y a menudo requiere administración de las bases de datos; esto es debido al número de dispositivos con acceso directo al ambiente de esas bases de datos. Además las herramientas del cliente y de la base de datos constantemente están cambiando, lo que genera dependencia a largo plazo de cualquier herramienta, complicándose así el escalamiento futuro o las implementaciones.

La arquitectura de tres capas introduce una capa intermedia al proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. Se instala una interfaz de usuario en la computadora del cliente. La tercera capa generalmente es el sistema de administración de la base de datos que se localiza en un servidor separado, donde son almacenados los datos requeridos por la capa intermedia. En esta última se implementa la lógica del negocio y proporciona un ambiente donde miles de usuarios pueden estar conectados simultáneamente.

### **Ventajas de la arquitectura de tres capas**

- Las llamadas de la interfaz del usuario en la estación de trabajo, al servidor de capa intermedia, son más flexibles que en el diseño de dos capas, ya que la estación sólo necesita transferir parámetros a la capa intermedia.
- Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, la estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC.
- El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular, esto puede reducir los esfuerzos de desarrollo y mantenimiento, así como los costos de migración.
- La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.
- Separando la aplicación de la base de datos, hace más fácil utilizar nuevas tecnologías de agrupamiento y balance de cargas.
- Separando la interfaz del usuario de la aplicación, libera de gran procesamiento a la estación de trabajo y permite que las actualizaciones de la aplicación sean centralizadas en el servidor de aplicaciones.

## **Desventajas de la arquitectura de tres capas**

- Los ambientes de tres capas pueden incrementar el tráfico en la red y requerir más balance de carga y tolerancia a las fallas.
- Los exploradores actuales no son todos iguales. La estandarización entre diferentes proveedores ha sido lenta en desarrollarse. Muchas organizaciones son forzadas a escoger uno en lugar de otro, mientras que cada uno ofrece sus propias y distintas ventajas.

En este trabajo se propone utilizar la arquitectura de tres capas para la realización de la aplicación que dará solución al problema ya que supera en muchos aspectos a la de dos capas, la arquitectura de tres capas soluciona las crecientes necesidades de los clientes, incrementa la habilidad para responder a los cambios, posibilita la reutilización de código, simplifica el mantenimiento y hace más fácil la migración a nuevas plataformas, ofreciendo soluciones viables. Sin duda alguna esta arquitectura cumple con las expectativas de crecimiento de la flexibilidad y de la escalabilidad; así como de una disminución del tiempo de desarrollo.

## **1.6 Fundamentación de la Tecnología.**

Para desarrollar este sistema se ha realizado un estudio de las herramientas y tecnologías que se usaron para su construcción. Hemos detallado en las tendencias actuales, sus características, ventajas, y las novedades de cada una.

### **1.6.1 PHP.**

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado y de alto nivel, especialmente pensado para desarrollos Web y el cual puede ser embebido en páginas HTML. Su sintaxis es prácticamente similar a C, Java y Perl, resulta muy atractivo y está respaldado por una gran comunidad de desarrolladores, debido a la perfecta combinación que proporcionan su simpleza para los programadores inexpertos, y a la vez su poder para los programadores profesionales. La meta de este lenguaje es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil. Su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU.

El código escrito en PHP tiene la peculiaridad de ser independiente de la plataforma, esto quiere decir que una aplicación escrita en PHP puede ser ejecutada en cualquier sistema operativo, gracias a que existen interpretes de PHP para casi o todos en el mercado, incluyendo Linux, muchas variantes Unix, Microsoft

Windows y otros. Además PHP es soportado por la mayoría de servidores Web de hoy día, siendo los principales Apache e Internet Information Services. [3]

Con las primeras dos versiones de PHP, se había conseguido una plataforma potente y estable para la programación de páginas del lado del servidor. Sin embargo, todavía existían puntos negros en el desarrollo de PHP que se han tratado de solucionar con la versión 5, como es la programación orientada a objetos (POO) que, a pesar de que estaba soportada a partir de PHP3, sólo implementaba una parte muy pequeña de las características de este tipo de programación.

El principal objetivo de PHP5 ha sido mejorar los mecanismos de POO. La principal novedad en sus clases es la inclusión de modificadores de control de acceso para implementar la encapsulación, piedra angular en la programación orientada a objetos de la que adolecía PHP4. PHP5 introduce tres palabras claves (public, private y protected) que sustituyen a (var).

#### **Ventajas del lenguaje PHP:**

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- Posee una amplia documentación en su página oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Nos permite crear los formularios para la Web.

#### **Seguridad en PHP.**

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor Web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor Web sea insegura por naturaleza. PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación, PHP nos puede dar la combinación precisa de libertad y seguridad que necesitamos. [4]

PHP fragmenta la seguridad en varios temas: instalación como un binario CGI, instalación como módulo de Apache, seguridad del sistema de archivos, seguridad de Bases de Datos, reporte de errores, uso de Register Globals, datos enviados por el usuario, Magic Quotes, ocultando PHP, estos son abordados punto por punto en sus manuales y en su sitio oficial <www.php.net>.

En nuestro caso uno de los primeros pasos para mejorar en la seguridad del sitio Web es, deshabilitar el reporte de errores y el registro de variables globales para evitar que un crack o cualquier otro usuario obtengan información del código PHP, tal como nombre de variables o funciones, rutas en el disco, etc. y logre modificar su valor a través de algún parámetro pasado por "url".

De acuerdo con estos argumentos, PHP resulta mucho más favorecido y además es uno de los lenguajes definidos, para la arquitectura de los sistemas que conforman la Intranet Universitaria, por el Departamento de Informatización, por tanto se decidió que es el adecuado para implementar la propuesta de sistema de este trabajo, particularmente PHP5.

### **1.6.2 El Lenguaje Procedimental SQL - PL/pgSQL**

Las aplicaciones en red son cada día más numerosas y versátiles. En muchos casos, el esquema básico de operación es una serie de scripts que rigen el comportamiento de una base de datos.

Debido a la diversidad de lenguajes y de bases de datos existentes, la manera de comunicar entre unos y otras sería realmente complicada de gestionar de no ser por la existencia de estándares que nos permiten realizar las operaciones básicas de una forma universal.

El Structured Query Language SQL es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos informática. Como su propio nombre indica, SQL es un lenguaje informático que se puede utilizar para interactuar con una base de datos y más concretamente con un tipo específico llamado base de datos relacional. [5]

SQL es el lenguaje que PostgreSQL y la mayoría de la bases de datos relacionales usan como lenguaje de consulta. Es portable y fácil de aprender. Pero cada sentencia SQL debe ser ejecutada individualmente por el servidor de la base de datos. Esto significa que su aplicación cliente debe enviar cada consulta al servidor de la base de datos, esperar a que sea procesada, recibir los resultados, hacer algunos cálculos y enviar después otras consultas al servidor. Todo esto implica la comunicación entre procesos y puede también implicar una sobrecarga a la red si su cliente se encuentra en una máquina diferente a la del servidor de la base de datos.



PL/pgSQL es un lenguaje procedimental cargable para el sistema de base de datos PostgreSQL que permite agrupar un bloque de cálculos y una serie de consultas dentro del servidor de la base de datos, obteniendo de esta manera el poder de un lenguaje procedimental y la facilidad de uso de SQL, pero ahorrando una gran cantidad de tiempo debido a que no tiene la sobrecarga de la comunicación completa cliente/servidor. También, con PL/pgSQL se puede usar todos los tipos de datos, operadores y funciones de SQL. Todo esto puede aumentar el desempeño de una manera considerable. [6]

### **1.6.3 HTML (HyperText Markup Language).**

HTML es un lenguaje muy sencillo que permite escribir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que permitan la navegación a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido...). La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, Netscape, o Internet Explore). Gracias a Internet y a los navegadores, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos y también de los más fáciles de aprender.

### **1.6.4 JavaScript.**

JavaScript es un lenguaje interpretado que, al contrario de las aplicaciones normales, que son ejecutadas por el sistema operativo, es ejecutado por el navegador que utilizamos para ver las páginas. Eso hace que podamos desarrollar aplicaciones de diversos tipos, desde generadores de HTML, páginas dinámicas, comprobadores de formularios, la manipulación de gráficos, el uso de elementos de las páginas Web, etc., hasta programas que gestionen las capas de una página. Pueden desarrollarse incluso aplicaciones que permitan poder tener capas en una página como si fueran ventanas, y dar la sensación de estar trabajando con una aplicación de Windows.

JavaScript es un lenguaje, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Es un lenguaje de tipo Script, lo que significa que no es un lenguaje compilado, no se ejecuta según se van leyendo sus instrucciones. JavaScript está guiado por eventos, o sea, no se tiene un programa que se ejecute de principio a fin en

cuanto se carga una página Web, sino que, cuando en el navegador suceda algún evento, entonces se ejecutará una acción determinada por una función JavaScript.

### **1.6.5 AJAX.**

AJAX, acrónimo de Asynchronous, JavaScript y XML, es una técnica de desarrollo Web para crear aplicaciones interactivas con interfaces de usuarios mucho más rápidas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. Programar aplicaciones con AJAX nos permite crear interfaces de usuario (UI) mucho más rápidas. AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente. Hoy podemos combinar AJAX, CSS, PHP y JavaScript para crear aplicaciones manteniendo el servidor, la interface de usuario y el diseño gráfico como componentes independientes de una arquitectura cliente-servidor, además de crear interfaces con respuesta más rápida.

#### **Ventajas del uso de AJAX:**

- Ya no se refresca la página constantemente al interactuar con ella.
- El tiempo de espera para una petición se reduce. El usuario al hacer un pedido (request) al servidor, no se envía toda la página.
- Por la misma razón anterior el tráfico al servidor se reduce.

#### **Desventajas que debemos tener en cuenta:**

- Falta de integración con el botón retroceder del navegador. Se debe tener en cuenta esto al intentar guardar funcionalidad con este botón.
- Falta de soporte para todos los navegadores. Aunque esto se va reduciendo, el problema se presenta por la falta de soporte para JavaScript.
- Problemas si el usuario ha deshabilitado el uso de JavaScript en su navegador. Hay que tener esto en cuenta cuando desarrollamos nuestro sitio Web, para enfrentarnos a esta situación.
- Demasiado código AJAX hace lento el navegador. A más AJAX, más uso de código JavaScript del lado del browser, por consiguiente mayor trabajo del browser. El rendimiento del CPU puede verse afectado.

Por las ventajas que ofrece y por conformar la arquitectura para los sistemas anclados a la intranet universitaria definida por el Departamento de Informatización será usada esta técnica de desarrollo Web en la aplicación.

### **1.6.6 XML.**

XML es un lenguaje de metamarcado que ofrece un formato para la descripción de datos estructurados. Esto facilita unas declaraciones de contenido más precisas y unos resultados de búsquedas más significativos en varias plataformas. Además, XML habilitará una nueva generación de aplicaciones, para ver y manipular datos, basadas en Web.

XML ofrece una representación estructural de los datos que se puede implementar ampliamente y es fácil de distribuir. Definido por el World Wide Web Consortium (W3C), XML garantiza que los datos estructurados sean uniformes e independientes de aplicaciones o fabricantes.

Proporciona interoperabilidad mediante un formato basado en estándares, flexible y abierto, con formas nuevas de acceso a las bases de datos existentes y de entregar datos a clientes de Web. Las aplicaciones se pueden generar más rápidamente, su mantenimiento es más sencillo y pueden ofrecer fácilmente varias vistas de los datos estructurados.

#### **Ventajas de XML:**

- Los autores y proveedores pueden diseñar sus propios tipos de documentos usando XML, en vez de limitarse a HTML.
- La información contenida puede ser más “rica” y fácil de usar, porque las habilidades hipertextuales de XML son mayores que las de HTML.
- XML puede dar más y mejores facilidades para la representación en los visualizadores.
- La información será más accesible y reutilizable, porque la flexibilidad de las etiquetas de XML pueden utilizarse sin tener que amoldarse a reglas específicas de un fabricante, como es el caso de HTML.

### **1.6.7 Sistemas Gestores de Base de Datos (SGDB).**

Un Sistema de Gestión de Bases de Datos (SGBD) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la

manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

**Un SGBD tiene los siguientes objetivos específicos:**

- Independencia de los datos y los programas de aplicación.
- Minimizar la redundancia.
- Integración y sincronización de las bases de datos.
- Integridad de los datos.
- Seguridad y protección de los datos.
- Facilidad de manipulación de la información.
- Control centralizado.

Entre los SGBD comúnmente utilizados en el mundo tenemos Oracle, MySQL, Microsoft SQL Server, PostgreSQL, entre otros. Todos los sistemas mencionados anteriormente facilitan el trabajo con la base de datos y tienen características que los diferencian, por ejemplo:

**Oracle:** requiere de una licencia para poderlo utilizar, es decir, es necesario pagar para poder utilizarlo.

**Microsoft SQL Server:** no es multiplataforma, solo puede ser utilizado con el sistema operativo Windows que está patrocinado por la compañía Microsoft.

**MySQL:** soporta un subconjunto de SQL92 menor que el que soporta PostgreSQL.

Como SGBD se seleccionó PostgreSQL por estar aprobado dentro de la arquitectura para los sistemas que conforman la intranet universitaria definida por el Departamento de Informatización.

### **1.6.7.1 PostgreSQL.**

PostgreSQL es un sistema de gestión de bases de datos Objeto-Relacionales (ORDBMS) libre, liberado bajo la licencia BSD (Berkeley Software Distribution) es una alternativa a otros sistemas de bases de datos de código abierto como MySQL, Firebird y MaxDB, así como sistemas propietarios como Oracle y SQLServer. La licencia BSD hace PostgreSQL muy popular con proveedores que desean incluir la base de datos en sus productos sin temor a honorarios, trabas de venta o cambios en los términos de licenciamiento.

El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

**Existen prestaciones y mejoras contribuidas por docenas de desarrolladores. Entre ellas se encuentran:**

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

**Altamente Extensible:** PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

**Soporte SQL Comprensivo:** PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

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

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

**MVCC:** MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Los bloqueos son provocados por usuarios que están escribiendo en la base de datos. Resumiendo, el lector está bloqueado por los escritores que están actualizando registros.

Mediante este sistema MVCC PostgreSQL permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y realizar copias de seguridad en caliente desde pg\_dump mientras la base de datos permanece disponible para consultas.

Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, porque un lector nunca es bloqueado por un escritor. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

## **1.6.8 Servidor de aplicaciones Web Apache.**

Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad. Apache es, el servidor Web más popular del momento, ha

demostrado ser substancialmente más rápido que muchos otros servidores libres y compite de cerca con los mejores servidores comerciales.

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. 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

#### **Ventajas:**

- Es capaz de utilizar otros interpretadores y lenguajes como "TCL", "PHP" y "Python".
- Puede conectarse directamente a una Base de Datos.
- Entre otras, posee diversos módulos que le permiten utilizar una gran gamma de lenguajes y desarrollar funcionalidades avanzadas.

### **1.6.9 Programación Extrema (XP).**

Basándose en la simplicidad, la comunicación y la reutilización del código surge la Programación Extrema (XP), siendo así una metodología muy ligera de desarrollo de software. XP consiste en una programación extrema (rápida). Como requisito para alcanzar el éxito del proyecto se tiene al usuario final como parte del equipo. Es una metodología con reconocido éxito y se utiliza en proyectos con entregas a cortos plazos.

Las características fundamentales de XP son:

- Desarrollo iterativo e incremental, lo que permite pequeñas mejoras, unas tras otras consecutivamente.
- Pruebas unitarias, estas son continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas, recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Permitiendo que el código sea revisado al mismo tiempo que se escribe.
- Frecuente interacción del equipo de programación con el cliente o usuario. Recomendando que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad haciendo entregas frecuentes.

- Refactorización del código, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad.
- Propiedad del código compartida, no se dividen las responsabilidades en el desarrollo de cada módulo. Este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código, es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Lo esencial en este proceso de desarrollo es lograr la comunicación entre desarrolladores y usuarios, la retroalimentación de estos entre ellos y con los usuarios finales y la simplicidad en el código.

### **1.6.10 Rational Unified Process (RUP).**

RUP es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software. El Proceso Unificado es un proceso de desarrollo de software que contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un software. Más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas, para diferentes áreas de aplicación, tipos de organizaciones, niveles de actitud y tamaños de proyecto.

Está basado en componentes, lo cual quiere decir que el software o sistema en construcción está formado por componentes de software interconectados a través de interfaces bien definidas. Utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema. Garantiza la elaboración de todas las fases de un producto de software orientado a objetos. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

#### **Características del Proceso Unificado**

Los verdaderos aspectos definitorios del Proceso Unificado, y que lo convierten en único, se resumen en tres frases clave - dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

**Dirigido por casos de uso:** Teniendo en cuenta que la razón de ser de un sistema es brindar servicios a los usuarios, RUP define caso de uso como el conjunto de acciones que debe realizar un sistema para dar un resultado de valor a un determinado usuario y los utiliza tanto para especificar los requisitos funcionales del sistema, como para guiar todos los demás pasos de su desarrollo, dígame diseño, implementación y prueba.

**Estar centrado en la arquitectura:** La arquitectura es una vista del diseño completo con las características más importantes, dejando a un lado los detalles. Esta no solo incluye las necesidades de los usuarios e inversores, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos, protocolos de red; con los que debe coexistir el sistema. En otras palabras, la arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.

**Ser iterativo e incremental:** La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada uno de estos mini-proyecto se les denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo. En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan, implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costes de un solo incremento, no sacar al mercado un producto en el tiempo previsto, mantener la motivación del equipo pues puede ver avances claros a corto plazo y que el desarrollo pueda adaptarse a los cambios en los requisitos.

### **1.6.11 Lenguaje de Modelado Unificado (UML).**

El Lenguaje de Modelado Unificado UML es una especificación de notación orientada a objetos. Es un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelación visual que permite una abstracción del sistema y sus componentes, que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. [7]

UML también intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos, se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo. [7]

Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otra rama.



### 1.6.12 El patrón Modelo-Vista-Controlador (MVC).

Los patrones de diseño proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifican y describen formas de solucionar problemas que ocurren de forma frecuente ya que se basan en la experiencia acumulada para resolver problemas reiterativos.

Los patrones son soluciones concretas, técnicas que se utilizan en situaciones frecuentes y facilitan la reutilización de código. Al aplicar un patrón, el código resultante no tiene por qué delatar el patrón o patrones que lo inspiró y es difícil reutilizar la implementación de estos ya que solucionan un problema específico.

Dentro de los patrones de comportamiento se encuentra el Modelo-Vista-Controlador (MVC), este será el que se utilizará en el diseño de la aplicación que se propone por las facilidades que ofrece y por estar aprobado dentro de la arquitectura para los sistemas que conforman la intranet universitaria definida por el Departamento de Informatización.

Su principal objetivo es aislar tanto los datos de la aplicación como el estado (modelo) de la misma, del mecanismo utilizado para representar (vista) dicho estado, así como para modularizar esta vista y modelar la transición entre estados del modelo (controlador).

#### **Las aplicaciones MVC se dividen en tres grandes áreas funcionales:**

**Vista:** la presentación de los datos

**Controlador:** el que atenderá las peticiones y componentes para toma de decisiones de la aplicación.

**Modelo:** la lógica del negocio o servicio y los datos asociados con la aplicación.

El propósito del MVC es aislar los cambios. Es una arquitectura preparada para los cambios, que desacopla datos y lógica de negocio de la lógica de presentación, permitiendo la actualización y desarrollo independiente de cada uno de los citados componentes.

El Modelo-Vista-Controlador es útil también cuando es necesario proporcionar múltiples representaciones de los mismos datos, en este contexto el controlador gestiona la información, y múltiples clases Vista ofrecen diferentes representaciones de la misma información, por ejemplo, diferentes diagramas (de barras, de tarta, etc.).

#### **Ventajas:**

- Múltiples vistas del mismo modelo.
- Vistas sincronizadas.
- Flexibilidad para cambiar las vistas y los controladores.
- La aplicación puede soportar distintos tipos de interfaz de usuario.

- Aumenta en gran medida el nivel de reusabilidad de código.
- Facilita una evolución continuada de los sistemas, sin puntos de ruptura, ya que un cambio en un sistema afectará a uno o más componentes pero nunca afectará significativamente al "core" de la aplicación.

**Inconvenientes:**

- Complejidad creciente.
- Cambios innecesarios. Puede ser que no todas las vistas estén interesadas en los cambios.
- Conexión entre la vista y el controlador. Hay que usar los dos a la vez.
- Si cambia la interfaz del modelo, hay que cambiar todas las vistas y todos los controladores.
- Acceso ineficiente a los datos en la vista. Puede necesitar varias llamadas al modelo para actualizar todos sus datos.
- Tanto la vista como el controlador son específicos de una plataforma.
- Algunas herramientas de diseño de interfaces de usuario incorporan parte del procesamiento de eventos entrada. El controlador deja de ser necesario.

**1.6.13 Diseño de la interfaz.**

Para el diseño de las vistas se utilizará Dreamweaver MX, del paquete de aplicaciones de Macromedia. Con Dreamweaver MX se puede diseñar vistas con poco consumo de tiempo solo que el uso de esta herramienta requiere de adiestramiento.

La herramienta Dreamweaver MX posibilita la creación de una interfaz de usuario integrada, que facilita la interacción dada la potencialidad de sus componentes de trabajo, además permite la creación de sitios dinámicos con la posibilidad de inclusión de otras tecnologías como son, ASP, ASP.NET, PHP y JSP para este caso particular.

**1.6.14 Zend Studio.**

Zend Studio es uno de los ambientes de desarrollo integrado o Integrated Development Environment (IDE) disponible para desarrolladores profesionales que agrupa todos los componentes de desarrollo necesarios para el ciclo de desarrollo de aplicaciones PHP. A través de un comprensivo conjunto de herramientas de edición, depurado, análisis, optimización y bases de datos, Zend Studio acelera los ciclos de desarrollo y simplifica los proyectos complejos.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Dispone de una herramienta muy interesante de depuración. Gracias a ella podemos ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las variables del entorno como las cookies, las recibidas por formulario o en la sesión. Podemos colocar puntos de parada de los scripts y realizar las acciones típicas de depuración.

### **1.6.15 Rational Rose.**

Es una herramienta para “modelado visual”, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software. Permite completar una gran parte de las disciplinas (flujos fundamentales) del Proceso Unificado (RUP) e incluye un conjunto de herramientas de ingeniería inversa y generación de código que allanan el camino hasta el producto final. Es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML.

**La metodología RUP divide en 4 fases el desarrollo del software:**

**Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.

**Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.

**Construcción:** El objetivo es obtener la capacidad operacional inicial.

**Transición:** El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, en cada uno, se hace exigente el uso y generación de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

### **1.6.16 Visual Paradigm.**

Esta herramienta acelera el desarrollo de aplicaciones, ya que sirve de puente visual entre arquitectos, analistas y diseñadores de sistemas de información, haciendo el trabajo más fácil y dinámico.

**Visual Paradigm ofrece:**

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio, lo que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas.

### **1.6.17 PgAdmin III.**

PgAdmin III es una aplicación gráfica para gestionar bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets. Es capaz de gestionar versiones a partir de PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas Unix), y puede encriptarse mediante SSL para mayor seguridad.

### **1.6.18 Adobe Photoshop.**

Adobe Photoshop es el software estándar de edición de imágenes profesional y el líder de la gama de productos de edición de imágenes digitales en el mundo. Las innovadoras herramientas creativas permiten conseguir resultados excepcionales. Una adaptabilidad sin precedentes le da la posibilidad de personalizar Photoshop de acuerdo con su método de trabajo. Además, gracias a unos procesos de edición, tratamiento y gestión de archivos más eficaces se puede trabajar con mayor rapidez.

### **1.6.19 MicroOLAP Database Designer for PostgreSQL.**

*MicroOLAP Database Designer for PostgreSQL* es un sistema de desarrollo visual previsto para el diseño de base de datos, la modelación, la creación, la modificación y la ingeniería reversa de una manera fácil y de gran alcance. Presenta una interfaz gráfica que permite construir de forma eficaz la estructura de la base de datos visualmente, representando todas las tablas, referencias entre ellas y el almacenamiento de

procedimientos y otros objetos. Permite generar una base de datos física en un servidor, modificarla según cualquier cambio realizado en el diagrama utilizando rápidas declaraciones ALTER. Acelera no sólo las marcas en el proceso de creación de base de datos sino también simplifica el mantenimiento de la misma. *MicroOLAP Database Designer for PostgreSQL* se desarrolla especialmente para el popular gestor de base de datos PostgreSQL y tiene dentro de sus principales ventajas:

- Ingeniería reversa (importación) de bases de datos existentes desde fuentes múltiples.
- La exportación de diagramas a los formatos gráficos más populares.
- Generación de informes de la estructura de la base de datos.
- Chequeo de los diagramas en busca de errores.
- Combinación de diagramas.
- Capacidad de imprimir el diagrama en páginas múltiples.
- Trabaja bajo Windows 2000, XP, 2003.

## **1.7 Conclusiones.**

Teniendo en cuenta la necesidad de la creación de una nueva aplicación para la informatización del proceso de gestión de opiniones de la Facultad Regional de Ciego de Ávila y el análisis de las potencialidades del uso de las tecnologías que han sido propuestas a lo largo del presente capítulo se llegó a la conclusión de que el sistema utilizará como gestor de bases de datos PostgreSQL (versión 8.1), Apache (versión 2.0.55) como servidor Web, la programación se desarrollará con el lenguaje PHP (versión 5.1.1). Como metodología para el desarrollo del sistema se ha decidido utilizar el Proceso Unificado de Desarrollo (RUP), que a su vez este hará uso del Lenguaje Unificado de Modelado (UML). La decisión de seleccionar esta metodología, además de ser objetivo de la institución, es debido a la consideración de XP como una metodología muy joven. También se tuvo en cuenta para la selección de las tecnologías y herramientas la Arquitectura definida por el Departamento de Informatización para las aplicaciones que conforman la Intranet Universitaria.

# Capítulo 2

## Características del Sistema

### **2.1. Introducción.**

A través de este capítulo se describe el objeto de estudio, el entorno de trabajo en que se desarrolla la aplicación, se realiza la propuesta del sistema partiéndose del análisis de los requerimientos funcionales y no funcionales. Se realiza el modelado del negocio, la definición de los casos de uso, de los actores que intervienen en ellos y se muestra el diagrama resultante de casos de uso.

### **2.2. Objeto de estudio.**

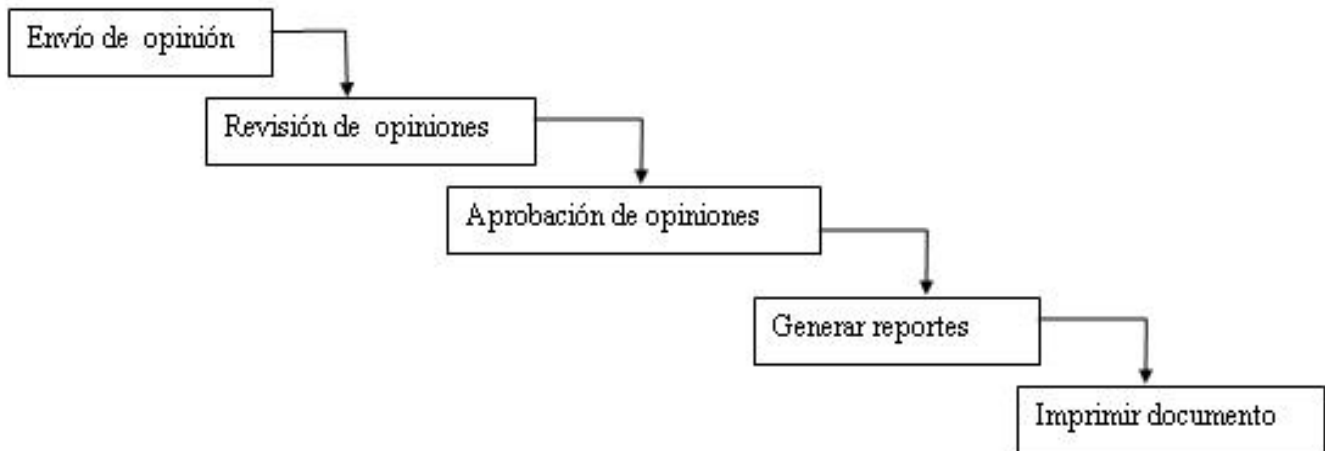
#### **2.2.1. Objetivos estratégicos de la organización.**

La Universidad de las Ciencias Informáticas (UCI) constituye el instituto de este tipo más joven del país, creada con el objetivo de formar personal altamente calificado en Ingeniería Informática, que contribuya en gran medida al desarrollo de ese campo y a la economía del país, y que además, “soporte a la informatización de la sociedad cubana y la competitividad internacional de la industria cubana del software” [8].

En estos momentos se está desarrollando en ella el Proyecto UCI-Ciudad Digital, que tiene como misión “informatizar la UCI y crear una cultura digital, utilizando las tecnologías más novedosas del estado del arte para lograr la integración y funcionamiento armónico de todos los procesos, desarrollando así el prototipo para la informatización de la Sociedad Cubana” [9].

Se han definido diferentes áreas de desarrollo en el proyecto, dentro de las que se encuentra la de Informatización. Esta ha comenzado a automatizar una gran parte de los procesos internos de la universidad, dentro de los cuales se encuentra el Sistema de Gestión de Opiniones.

## 2.2.2 Flujo actual de los procesos.



**Fig1. Esquema del flujo actual de los proceso del SGO.**

## 2.2.3 Situación problemática.

En la actualidad la UCI cuenta con un Sistema de Gestión de Opiniones, que permite a la comunidad universitaria emitir sus criterios acerca de un determinado tema del acontecer nacional e internacional que sea de interés para la Máxima Dirección del país. Esta aplicación ha permitido organizar y agilizar un poco el trabajo con las opiniones pero presenta deficiencias en algunas de sus funcionalidades básicas como la actualización del estado de las opiniones en el proceso de revisión imposibilitando diferenciar las revisadas del resto y provocando que se vuelvan a revisar.

El sistema no brinda la información suficiente para elaborar reportes gráficos o estadísticos, si se desea emitir un reporte que muestre la cantidad de opiniones revisadas aprobadas o archivadas por tipos de usuarios el sistema no puede mostrar esta solicitud. También se hace imposible poder publicar varios temas de debates simultáneamente, esto provoca que los temas de debate tengan un tiempo de publicación muy limitado.

El documento que genera la aplicación con las opiniones aprobadas no cumple con el formato requerido por lo que el mismo es confeccionado a mano, retrasándose de esta manera el envío del mismo a la Dirección del país. Además no posee ninguna documentación que pueda utilizarse para apoyar el análisis de los procesos de negocio y los requerimientos que fueron identificados en sus inicios por lo que se hace imposible la reutilización de su código.

Con el surgimiento de las Facultades Regionales (FR) de la Universidad de las Ciencias Informáticas (UCI), se decide transmitir experiencias e implantar aplicaciones que se utilizan para gestionar un grupo de actividades en la sede universitaria, dentro de las cuales se encuentra el Sistema de Gestión de Opiniones (SGO).

Actualmente no se cuenta con una aplicación para la Gestión de las Opiniones que cumpla con los requerimientos necesarios del cliente para ser usada en las Facultades Regionales (FR) de la Universidad de las Ciencias Informáticas (UCI) siendo este el problema científico a resolver.

Debido a la necesidad de mejorar todas las actividades que involucra la gestión de las opiniones, con el objetivo de que estas lleguen en el tiempo requerido a la Máxima Dirección del país, se decide implementar un nuevo SGO para ser instalado en las Facultades Regionales que rectifique todos los errores que presenta el sistema actual.

## 2.3 OBJETO DE AUTOMATIZACIÓN.

Se desea implementar un nuevo software que informatice los siguientes procesos:

**Autenticar usuario:** Debe permitir la autenticación de los usuarios contra el directorio LDAP para verificar si son usuarios del dominio y según los privilegios asignados, permitir el acceso al sistema.

**Gestionar temas de debates:** Debe permitir la creación, edición y eliminación de temas para opinar. Además el sistema debe registrar el usuario, la fecha, la hora, el IP y el nombre de la PC y la acción que se realiza.

**Emitir Opiniones:** Debe permitir a los usuarios opinar sobre los temas que están publicados. Además el sistema debe registrar el usuario, la fecha, la hora, el IP y el nombre de la PC y la acción que se realiza.

**Revisar Opiniones:** Debe permitir revisar o archivar las opiniones que han sido emitidas por temas publicados, así como corregirlas ortográficamente. Además el sistema debe registrar el usuario, la fecha, la hora, el IP y el nombre de la PC y la acción que se realiza.

**Aprobar las Opiniones:** Debe permitir aprobar, archivar o no aprobar las opiniones que han sido revisadas. En el caso de las aprobadas, estas son las que conformarán el documento final. Además el sistema debe registrar el usuario, la fecha, la hora, el IP y el nombre de la PC y la acción que se realiza.

**Gestionar Opiniones Archivadas:** La operación de archivar se debe realizar cuando: se detecte que una opinión ha sido emitida en más de una ocasión, la valoración emitida es incoherente o no se ajusta al concepto de opinión que se enuncia en el sitio. La informatización de este proceso permitirá editar,



aprobar o eliminar las opiniones que se encuentren archivadas. Además el sistema debe registrar el usuario, la fecha, la hora, el IP y el nombre de la PC y la acción que se realiza.

**Gestionar Reportes:** Debe permitir la creación de reportes estadísticos y de opiniones, brindando la posibilidad de seleccionar de forma dinámica la información que se desee mostrar en los mismos.

Los reportes de opiniones por temas deben ser generados en un documento de texto enriquecido (.rtf), con un formato pre-definido, que contendrá las opiniones a partir de la selección del tipo de información que se quiere mostrar, esta puede ser: opiniones por estados (revisadas, aprobadas o archivadas) y con las siguientes opciones (el usuario, el nombre del emisor, la hora, la fecha, la IP o el nombre de la PC desde la que se emite la opinión), por cada uno de estos estados.

Los reportes estadísticos de opiniones por temas, deben ser generados en un documento de texto enriquecido (.rtf), con un formato pre-definido, que contendrá las estadísticas por opiniones a partir de la selección del tipo de información que se desee mostrar, esta puede ser: opiniones por tipo de usuario o año de estudio según los siguientes estados: emitidas, revisadas, aprobadas o archivadas y total de opiniones por estado o por tipos de usuarios (estudiantes, profesores, trabajadores). Además debe permitir adicionar gráficas de barra, de pastel y de este último también en 3D.

Además el sistema debe registrar el usuario, la fecha, la hora, el IP y el nombre de la PC y la acción que se realiza.

**Gestionar Usuarios:** Debe permitir la asignación de roles a usuarios del dominio y la actualización de estos si se requiere, así como la eliminación de usuarios con roles en la aplicación.

**Graficar Estadísticas:** Debe permitir graficar las estadísticas por año de estudio, por estado (emitidas, revisadas aprobadas, archivadas), por tipo de usuario (estudiantes, profesores, trabajadores) y personalizar las mismas. Además el sistema debe registrar el usuario, la fecha, la hora, el IP y el nombre de la PC y la acción que se realiza.

**Gestionar los Logs:** Debe permitir mostrar y eliminar los Logs de actividades generados.

### **2.3.1. Sistemas existentes vinculados con el campo de acción.**

El sistema para la gestión de las opiniones que se utiliza actualmente en la UCI surgió con la necesidad de lograr una mayor eficiencia en los procesos vinculados a la gestión de opiniones de la comunidad y recoger la mayor cantidad de criterios en el menor tiempo posible, el mismo gestiona las opiniones de los estudiantes, profesores y trabajadores a partir de la publicación de un tema orientado por la Dirección del PCC de la Universidad.

La aplicación permite la autenticación de los usuarios, la emisión de opiniones sobre temas específicos, admitiendo un solo tema publicado y la revisión de las opiniones aunque no de manera eficiente, además crea un reporte en formato: documento de texto enriquecido (.rtf), que no cumple con el formato requerido. También brinda algunas estadísticas predeterminadas. Para su desarrollo se utilizaron las tecnologías: SQL Server como gestor de bases de datos, la programación se desarrolló usando la tecnología ASP, por lo que la aplicación solo puede ser montada sobre la plataforma Windows.

Al interactuar con este sistema se puede notar que existen problemas con la navegabilidad, el ambiente de trabajo resulta en ocasiones restringido, la interfaz requiere ser más agradable, interesante y atractiva a la vista del usuario.

## **2.4 INFORMACIÓN QUE SE MANEJA.**

En el proceso de gestión de las opiniones, al final del flujo de algunos de sus procesos se generan reportes por temas o estadísticos, en formato: documento de texto enriquecido (.rtf),, que manipulan la información referente a las opiniones y los temas de debate. El documento generado según los tipos de reportes tendrá el formato siguiente:

El estilo de la letra será Arial y el tamaño de la fuente: 12, el encabezado de los reportes será:

### **Reporte Generado por el Sistema de Opiniones UCI.**

**Tema:**

**Fecha:**

A partir del formato anterior que es común tanto para los reportes estadísticos como para los de opiniones continúa de la siguiente forma:

#### Reportes de opiniones:

**Opiniones:**

A partir de la selección del tipo de opinión según el estado (revisadas, aprobadas, archivadas) que se desean mostrar, estas se enumerarán y seguido al texto de estas se colocarán las opciones seleccionadas que pueden ser (usuario, nombre, fecha, hora, IP, nombre de la PC).

#### Reportes estadísticos:

**Estadísticas:**

Las estadísticas se mostrarán en tablas que contendrán las informaciones seleccionadas estas pueden ser:

### **Total de Opiniones por Estado o por Tipo de Usuario.**

**Estadísticas por año de estudio:** Opiniones emitidas, revisadas, aprobadas y archivadas por año.

**Estadísticas por Tipo de Usuario:** Opiniones emitidas, revisadas, aprobadas y archivadas por tipo de usuario.

Ejemplo:

### **Estadísticas Generales.**

Total de Opiniones por Estado.

emitidas	revisadas	aprobadas	archivadas

Además se pueden adicionar al documento gráficas de barra, de pastel y de este último también en 3D, que hayan sido graficadas previamente.

## **2.5 PROPUESTA DEL SISTEMA.**

La aplicación que se propone es un nuevo sistema que presenta una interfaz amigable, de fácil navegación y ajustada a los estándares de diseño definidos por la Universidad. El sistema permite la emisión de opiniones así como la revisión y aprobación de las mismas. Brinda la posibilidad de otorgar, modificar y eliminar determinados privilegios asignados a ciertos usuarios del dominio restringiendo de esta forma los niveles de acceso a la aplicación. Es capaz de mostrar la información suficiente para proporcionar estadísticas en texto plano o mediante gráficas para su posterior revisión y estudio.

El nuevo sistema brinda la posibilidad de publicar varios temas de debates simultáneamente de los cuales se podrán generar reportes en un documento con el formato requerido con la información que el usuario desee mostrar de las mismas ya que la aplicación permite seleccionar por estados (revisadas, aprobadas, archivadas), el usuario, el nombre, la fecha, la hora, el IP y el nombre de la maquina desde la que se emite la opinión. Permite graficar las estadísticas por tipos de usuario, por estados, por años de estudio y personalizarlas según el tipo de información que se necesite ilustrar. Además permite mostrar y eliminar los Logs de actividades generados en la aplicación.

A diferencia de la aplicación para la gestión de las opiniones que se encuentra funcionando hoy en la UCI la nueva aplicación para las Facultades Regionales se desarrolló en PHP, aprovechando así las características de este lenguaje atractivo y respaldado por una gran comunidad de desarrolladores ya que

tiene entre sus principales peculiaridades ser independiente de la plataforma y utilizar las técnicas de Programación Orientada a Objetos.

Cuenta con la documentación como complemento para el análisis de los procesos de negocio y los requerimientos que fueron identificados, haciendo posible la reutilización de su código si fuese necesario.

## 2.6 MODELO DE NEGOCIO.

El modelado del negocio es una técnica para comprender las funciones (Proceso) que se desarrollan en el ambiente o entorno en el cual está enmarcado el problema (Negocio).

**Los propósitos que se persiguen al realizarse el modelado del negocio, son:**

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar mejoras posibles.
- Asegurar que los clientes, usuarios finales y desarrolladores tengan una idea común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

### 2.6.1 Definición de los actores del modelo de negocio.

Podemos definir como actor del negocio a una persona, grupo, organización, máquina o sistemas de información externos que interactúan con el negocio para beneficiarse de sus resultados.

Después de todas las entrevistas y estudios necesarios realizados (haciendo una simplificación del negocio) se obtiene un listado de actores y trabajadores del negocio como se muestra a continuación:

**Actores del negocio y su justificación:**

Actores	Justificación
Usuario	Interviene en el proceso de emisión de opiniones con el objetivo de expresar sus valoraciones y criterios relacionados con un tema.
Dirección del PCC	Interviene en el procesamiento de las opiniones con el objetivo de beneficiarse con la obtención de los reportes de las opiniones emitidas y el estudio de los resultados de las estadísticas por facultades y áreas de trabajo.

### 2.6.2 Definición de los trabajadores del modelo de negocio.

Podemos decir que los trabajadores definen la manera de actuar y responsabilidades de un individuo, grupo, sistema automatizado o máquina, que trabajan en conjunto como equipo, ellos son los que realizan las actividades y son propietarios de artefactos (productos tangibles del proyecto).

#### Trabajadores del negocio y su justificación:

Trabajadores	Justificación
Editor	Interviene en el procesamiento de las opiniones siendo el encargado de revisar las opiniones corregir la ortografía y la coherencia en la idea que fue transmitida.
Revisor	Interviene en el procesamiento de las opiniones siendo el encargado de aprobar las opiniones con más calidad y profundidad.
Directivo.	Interviene en el procesamiento de las opiniones siendo el encargado de conformar el reporte final de las opiniones e informar las estadísticas.

### 2.6.3 Casos de Uso del Negocio.

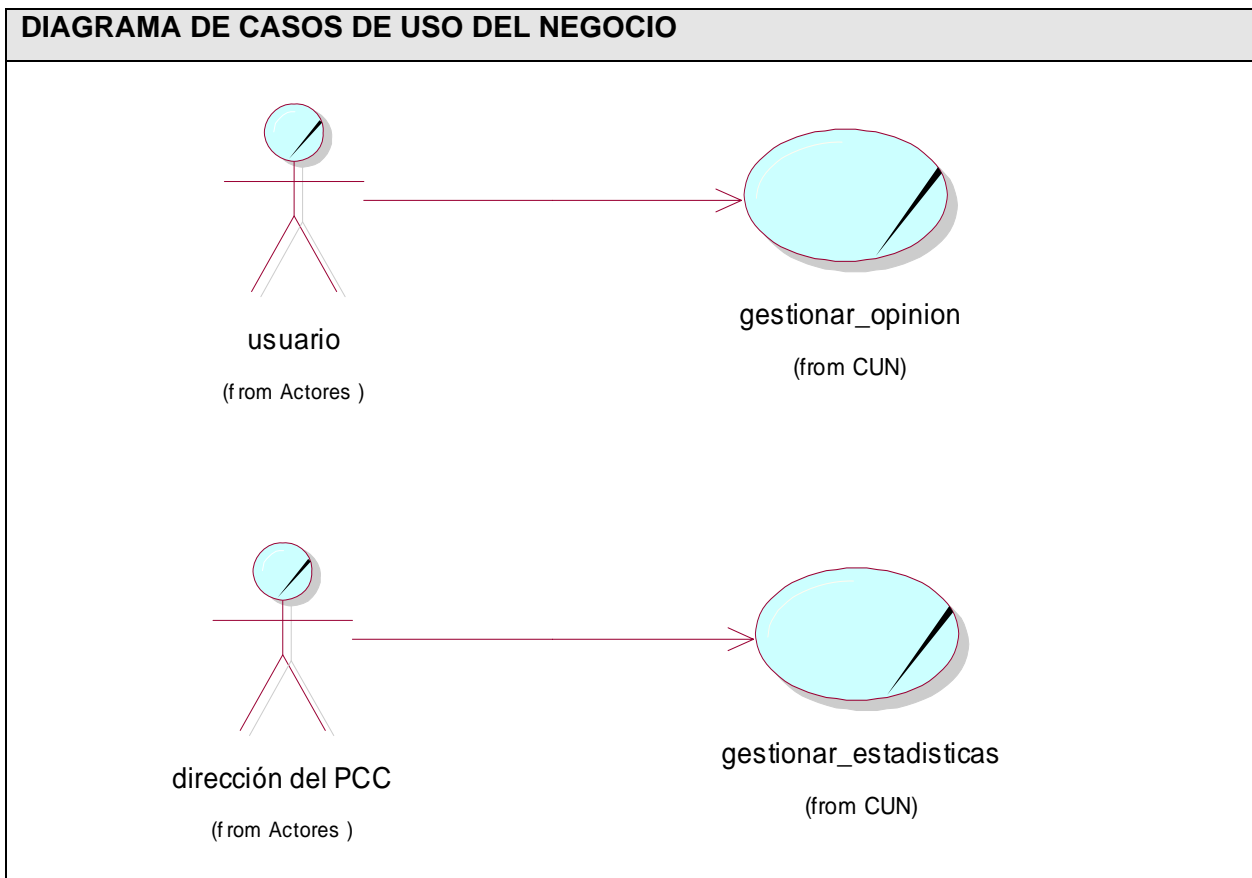
Los casos de uso del negocio son la representación de un proceso del negocio, y se hallan sujetos a un conjunto de reglas del negocio.

**Gestionar opinión:** Permite al usuario dar sus criterios y consideraciones; al editor, revisor y directivo, revisar y aprobar opiniones, así como crear reportes con las estadísticas u opiniones aprobadas a este último.

**Gestionar estadística:** Permite a la Dirección del PCC revisar y analizar las estadísticas así como crear reportes según las necesidades.

### 2.6.4 Diagrama de casos de uso del negocio.

A continuación aparece el diagrama de casos de uso del negocio.



**Figura A1.1: Diagrama del Modelo de Casos de Uso del Negocio.**

### **2.6.5 Casos de uso del negocio y diagrama de actividades.**

El modelo de Casos de Uso del Negocio es un modelo que describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes, es decir, describe las funciones que se realizan en el negocio como tal y su objetivo básico es la clarificación de estos procesos.

Para ello se usan diagramas de actividad que pretenden explicar cómo se construyen los diferentes flujos de trabajo o los procesos dentro de un sistema, y establecer un orden entre ellos generando el cambio de una actividad hacia otra. Un diagrama de actividades generalmente no modela el comportamiento exacto de un sistema de software (como lo hace un diagrama de secuencia), sino los procesos y los flujos a un

muy alto nivel. Además ayudan a identificar que funciones deberá asumir el producto de software, y quienes serán los futuros actores del sistema.

Las actividades generalmente serán realizadas por uno o más casos de uso; y describen el proceso que se desarrolla; tanto el caso de uso como un actor usarán el sistema para realizar toda o parte de una actividad.

La especificación de los casos de uso del negocio con sus diagramas de actividad correspondientes puede verse en el Anexo 1.

### **2.6.6 Diagrama de clases del modelo de objeto del negocio.**

Un modelo de objetos del negocio es un modelo interno al negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y de unidades de trabajo [7].

El diagrama de clases del modelo de objeto de negocio puede verse en el Anexo 2.

## **2.7. ESPECIFICACIÓN DE LOS REQUERIMIENTOS DEL SISTEMA.**

### **2.7.1. Requerimientos funcionales.**

Los requerimientos funcionales no son más que la determinación clara y concisa de qué debe ser capaz de hacer el sistema, éstas se corresponden con opciones que ejecutará el software, operaciones realizadas de forma oculta o condiciones extremas a determinar por el sistema.

A continuación aparecen los requisitos funcionales:

#### **R1. Agregar tema de debate.**

El sistema debe permitir:

- 1.1 Mostrar lista de temas con los que cuenta la UCI para opinar con los campos: tema y fecha de publicación.
- 1.2 Insertar un tema con los campos: tema, fecha de publicación.
- 1.3 Publicar o no publicar determinado tema.
- 1.4 Modificar o eliminar determinado tema.
- 1.5 Registrar el usuario, la fecha, IP donde se inserta el tema.
- 1.6 Permitir la paginación de la lista de temas según el número de registros por página.

## **R2. Agregar opinión dado un tema.**

El sistema debe permitir:

- 2.1 Escoger el tema para opinar.
- 2.2 Insertar opinión.
- 2.3 Registrar el usuario, la fecha, IP donde se opina.
- 2.4 Mostrar el usuario.

## **R3. Revisar opinión dado un tema.**

El sistema debe permitir:

- 3.1. Mostrar lista de opiniones sin revisar por tema con los campos: nombre de la persona que opina, fecha en que se emitió la opinión.
- 3.2. Modificar una opinión.
- 3.3. Registrar el usuario, la fecha, IP donde se revisa o archiva.
- 3.4. Cambiar el estado de la opinión: revisada o archivada, en caso de que se repita.
- 3.5. Permitir la paginación de la lista de opiniones sin revisar según el número de registros por página.
- 3.6. Mostrar lista de opiniones archivadas por tema con los campos: texto de la opinión, nombre de la persona que opina, fecha en que se emitió la opinión, nombre de la persona que archiva la opinión, fecha en que se archiva la opinión.
- 3.7. Mostrar la opinión en revisión por usuario con los campos: texto de la opinión, nombre de la persona que opina, fecha en que se emitió la opinión, nombre de la persona que revisa la opinión.
- 3.8. Permitir la paginación de la lista de opiniones archivadas según el número de registros por página.

## **R4. Recuperar opinión archivada.**

El sistema debe permitir:

- 4.1. Cambiar el estado de la opinión: sin revisar o revisada.

## **R5. Aprobar opinión.**

El sistema debe permitir:

- 5.1. Cambiar el estado de la opinión: aprobada.
- 5.2. Mostrar la opinión revisada para aprobar, por tema con los campos: texto de la opinión, nombre de la persona que opina, fecha en que se emitió la opinión, nombre de la persona que acepta la opinión, fecha en que se acepta la opinión.
- 5.3. Registrar el usuario, la fecha, IP donde se acepta o manda a revisar.



**R6. Generar un documento de texto enriquecido (.rtf) con las opiniones aprobadas por tema y la fecha de entrega.**

El sistema debe permitir:

6.1. Registrar el usuario, la fecha, IP donde se genera el documento.

6.2. Imprimir documento.

**R7. Agregar usuario.**

El sistema debe permitir:

7.1. Mostrar lista de usuarios con los campos: usuario, tipo de usuario.

7.2. Asignar privilegios a un usuario.

7.3. Verificar que una persona esté registrada como usuario del dominio UCI.

7.4. Permitir la paginación de la lista de usuarios según el número de registros por página.

7.5. Modificar o eliminar determinado usuario.

7.6. Registrar el usuario, la fecha, IP donde registra el usuario.

**R8. Graficar reportes dado determinado tema de debate.**

El sistema debe permitir graficar:

8.1. Cantidad de opiniones por año de estudio.

8.2. Cantidad de opiniones revisadas por año de estudio.

8.3. Cantidad de opiniones aprobadas por año de estudio.

8.4. Cantidad de opiniones archivadas por año de estudio.

8.5. Cantidad de opiniones revisadas por estudiantes.

8.6. Cantidad de opiniones revisadas por profesores.

8.7. Cantidad de opiniones revisadas por trabajadores.

8.8. Cantidad de opiniones aprobadas por estudiantes.

8.9. Cantidad de opiniones aprobadas por profesores.

8.10. Cantidad de opiniones aprobadas por trabajadores.

8.10. Cantidad de opiniones archivadas por estudiantes.

8.11. Cantidad de opiniones archivadas por profesores.

8.11. Cantidad de opiniones archivadas por trabajadores.

**R9. Graficar reportes generales por tema de debate.**

El sistema debe mostrar gráficas con las siguientes características:

9.1. Cantidad de opiniones revisadas.

9.2. Cantidad de opiniones aprobadas.

9.3. Cantidad de opiniones archivadas.

**R10. Generar y mostrar reportes generales por tema de debate.**

El sistema debe permitir visualizar reportes con las siguientes características:

10.1. Cantidad de opiniones revisadas por estudiantes, trabajadores y profesores.

10.2. Cantidad de opiniones aprobadas por estudiantes, trabajadores y profesores.

10.3. Cantidad de opiniones archivadas por estudiantes, trabajadores y profesores.

10.4. Cantidad de opiniones emitidas por estudiantes, trabajadores y profesores.

**R11. Obtener documento de texto enriquecido (.rtf) con las estadísticas.**

El sistema debe permitir:

11.1. Seleccionar las estadísticas para generar el documento.

11.2. Imprimir documento.

**R12. Autenticar Usuario.**

El sistema debe permitir:

12.1 Verificar que una persona esté registrada como Persona UCI en el dominio para acceder al sistema.

**R13. Logs de actividades.**

El sistema debe permitir:

13.1 Mostrar los Logs generados por el sistema.

13.2 Eliminar los Logs.

**2.7.2. Requerimientos no funcionales.**

Los requerimientos no funcionales responden a cualidades que el producto debe tener y las características para que este sea atractivo, confiable, usable y seguro.

**Apariencia o interfaz externa:** La interfaz debe ser sencilla, intuitiva, amigable y mantener el formato en páginas similares. Debe poseer una apariencia profesional, legible y simple de usar.

**Requerimientos de portabilidad:** El sistema es multiplataforma, podrá correr sobre Windows, Linux, Unix, etc.

**Requerimientos de software:** Se utiliza Apache con PHP 5, como lenguaje de programación, Postgre como Gestor de Base de Datos y Macromedia Dreamweaver MX como herramienta de diseño.

**Requerimientos de hardware:** Para que los usuarios del sistema puedan utilizarlo es necesario que cada uno tenga una computadora conectada a la red, con acceso a la Intranet de la Facultad Regional. Tanto

los usuarios como la administración, donde se encuentra el servidor, necesitan un navegador para poder acceder a las opciones que brinda el sistema. El sistema requiere de una computadora que haga función de servidor, la cual debe cumplir con las características mínimas siguientes: como sistema operativo puede tener instalado Windows o Linux, la Memoria RAM debe ser de 256 MB y la capacidad del disco duro de 20 GB.

**Requerimientos de Seguridad:**

**Confiabilidad:** La información estará protegida contra accesos no autorizados utilizando para esto mecanismos de autenticación y encriptación que garanticen su confiabilidad.

**Integridad:** La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, por lo que será de vital importancia el uso de políticas de salvadas de la base de datos, mediante las que se realicen seguimientos a cualquier problema que se presente.

**Disponibilidad:** Los usuarios autorizados tendrán acceso a la información contenida en el sistema, siempre que esté publicada.

**Requerimientos de Usabilidad:** El sistema será utilizado por los usuarios (estudiantes, profesores y trabajadores) de la Facultad Regional de la UCI. El personal encargado de la administración, la revisión y la aprobación podrán acceder al mismo cuando esté publicada.

**Requerimientos de Rendimiento:** La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente / Servidor y la velocidad de las consultas en la Base de Datos. El sistema debe implementar varias transacciones por segundo para dar una respuesta rápida al usuario y evitar demoras innecesarias.

**Requerimientos Legales:** La publicación del sitio debe ser aceptada por la Universidad además de cumplir con los requisitos legales que la misma establece. La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL.

**Políticos-Culturales:** Toda modificación que se le realice al sistema deberá ser autorizada por la Dirección de Informatización del centro, conjuntamente con la del PCC.

**Restricciones de diseño:** Se utiliza el Rational Rose Enterprise Edition 2003 como herramienta para la modelación del sistema, el patrón Modelo Vista Controlador (MVC) para el diseño de la aplicación y como Gestor de Base de Datos PostgreSQL.

**Requerimiento de ayuda:** Se deberá brindar una ayuda que facilite a los usuarios el trabajo con la aplicación.

### **Requerimiento de Soporte.**

La base de datos que utilizará el sistema como medio de almacenamiento de la información estará soportada por un gestor de bases de datos PostgreSQL. Se debe lograr la solidez de los datos realizando mantenimientos a la Base de Datos, orientados a la actualización y corrección de la información, a horas donde haya la menor cantidad de usuarios conectados. Se realizarán backups a la Base Datos semanalmente, una vez al año se eliminarán las opiniones archivadas para liberar espacio y cada tres años los temas publicados.

## **2.8 DEFINICIÓN DE LOS CASOS DE USO.**

La definición de los casos de uso del sistema puede verse en el Anexo 2.

## **2.9 CONCLUSIONES.**

Este capítulo ha sido fundamental para determinar y describir el objeto de automatización y la información que se maneja en el sistema propuesto. Se han detallado los procesos de desarrollo de la aplicación, haciendo énfasis en los requerimientos del sistema y en la explicación detallada de los diez casos de usos identificados, siguiendo cada uno de los pasos para obtener un producto con una calidad garantizada. Se ganó claridad en cuanto a la concepción del sistema a construir y se sentaron las bases para las restantes fases del proceso de diseño e implementación del sistema.

# Capítulo 3

## Análisis y Diseño del Sistema

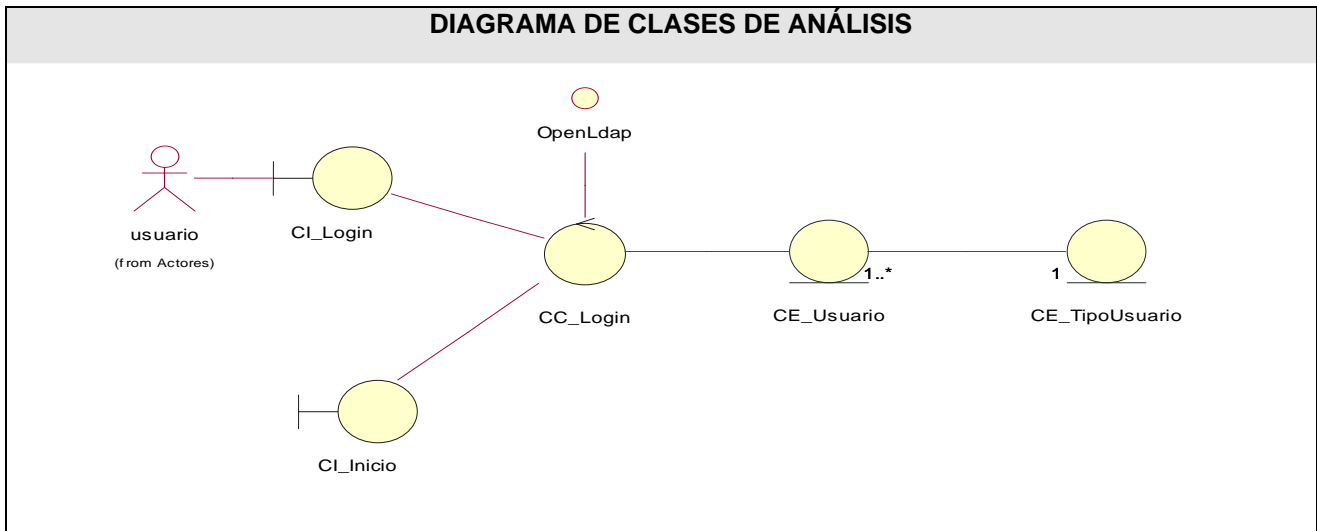
### **3.1 INTRODUCCIÓN.**

El análisis y diseño del sistema son algunos de los mecanismos más importantes dentro del proceso unificado del desarrollo de un software. Se consideran más cercanos al acceso de datos y a la arquitectura física de la aplicación. El diseño es la etapa del proceso de desarrollo donde se decide cómo se llevará a cabo el sistema. A través de esta fase, se toman decisiones estratégicas y tácticas para cumplir los requerimientos funcionales y de calidad de un sistema. En el presente capítulo se define el modelo de análisis, modelo de clases de análisis, los diagramas de interacción, se describen las clases, se diseña la Base de Datos, se analizan las definiciones de diseño a aplicar, así como el tratamiento de errores, seguridad, interfaz y concepción de la ayuda.

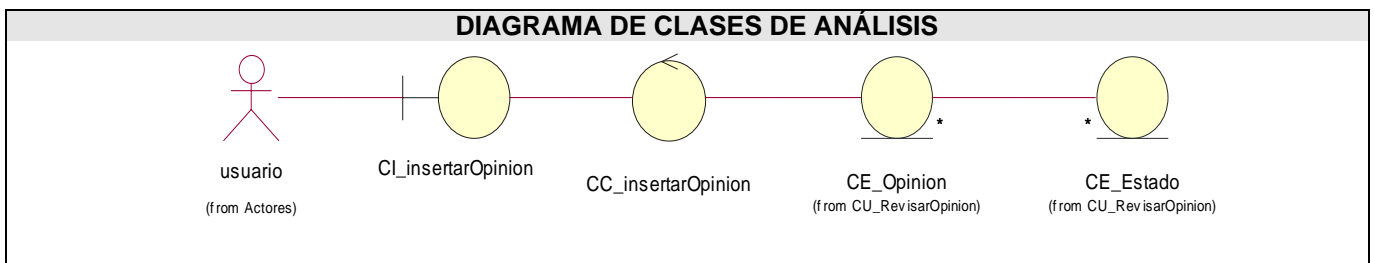
### **3.2 ANÁLISIS DEL SISTEMA.**

#### **3.2.1 Modelo de clases del análisis.**

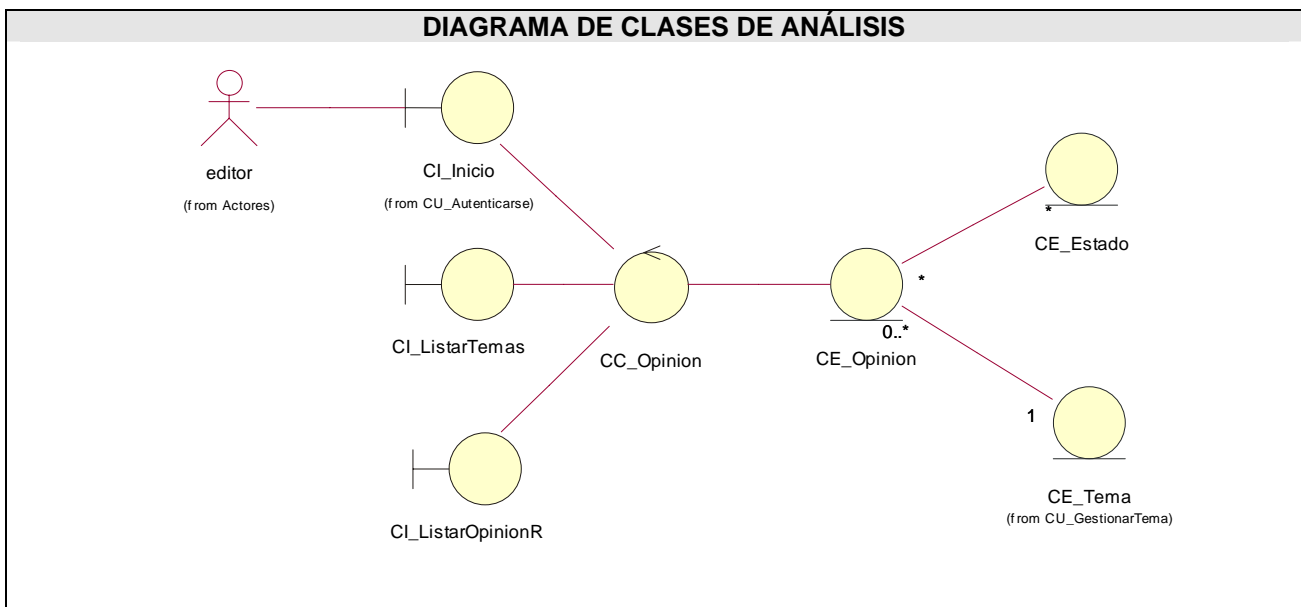
El diagrama de clases es una descripción de los modelos de objetos, contiene clases y las relaciones estructurales y de herencia existentes entre ellas. Este se obtiene como resultado del refinamiento del modelo conceptual y de los diagramas de secuencia. La definición de clase incluye definiciones para atributos y responsabilidades.



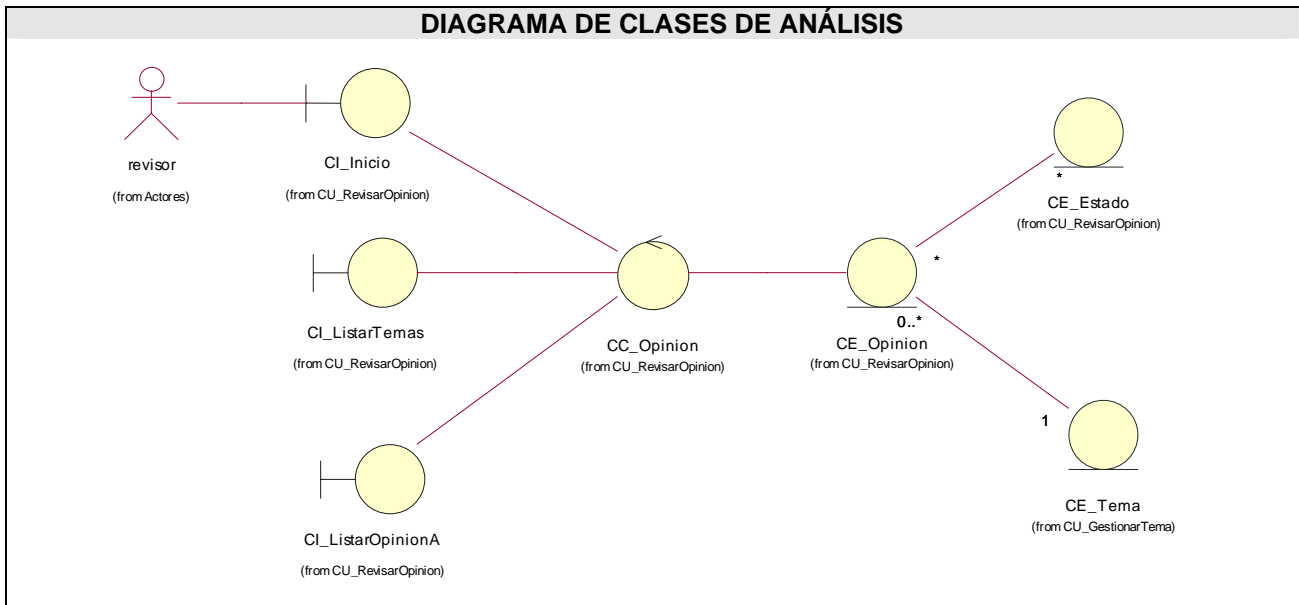
**Figura 3.1: Diagrama de Clases del Análisis CU Autenticarse.**



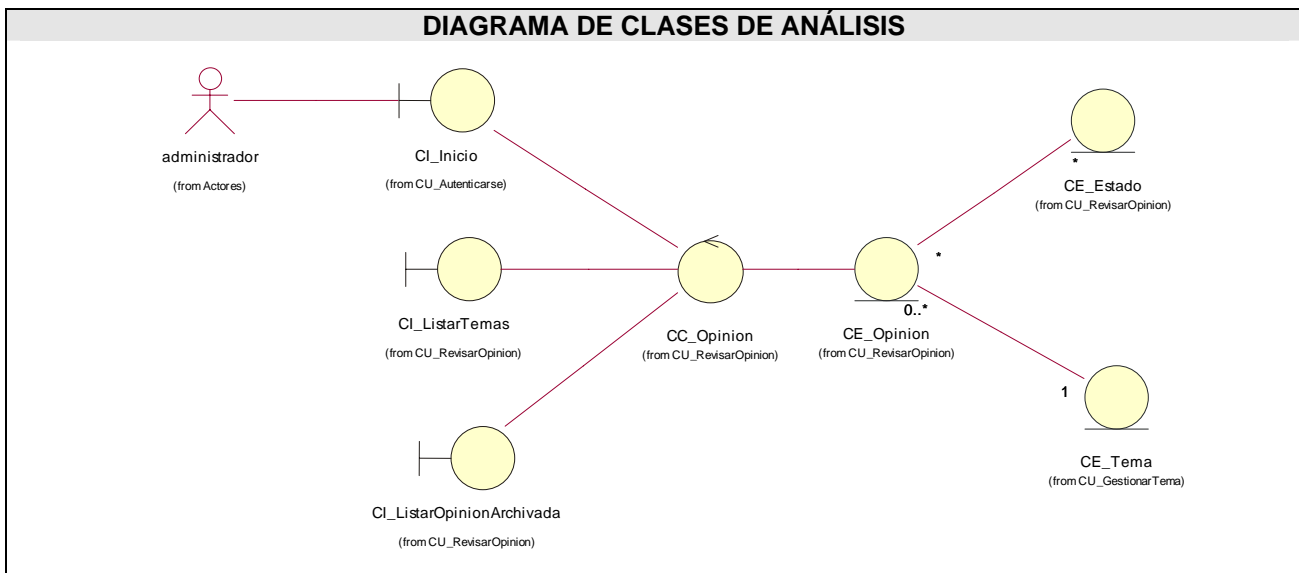
**Figura 3.2: Diagrama de Clases del Análisis CU Insertar Opinión.**



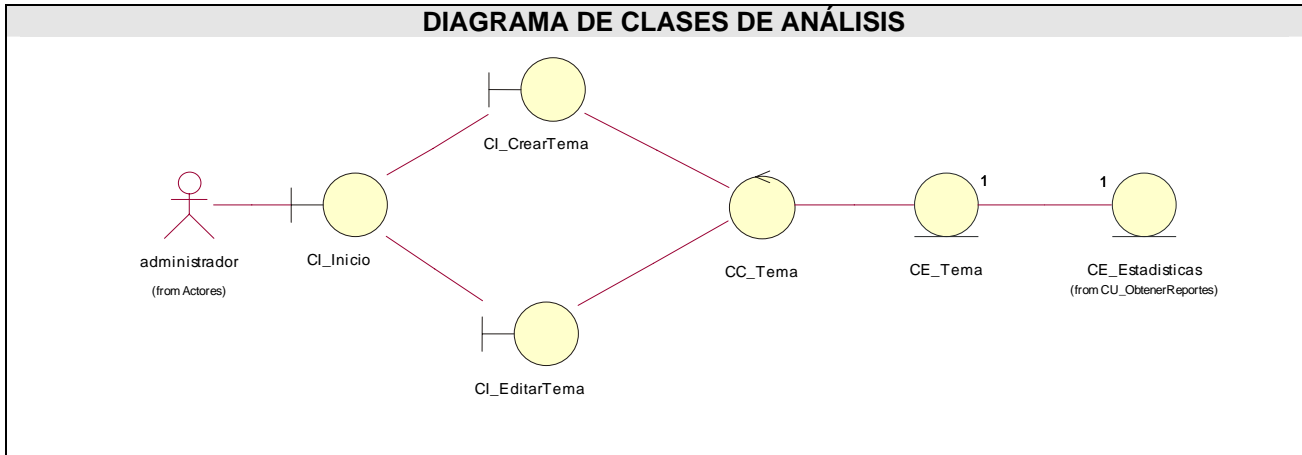
**Figura 3.3: Diagrama de Clases del Análisis CU Revisar Opinión.**



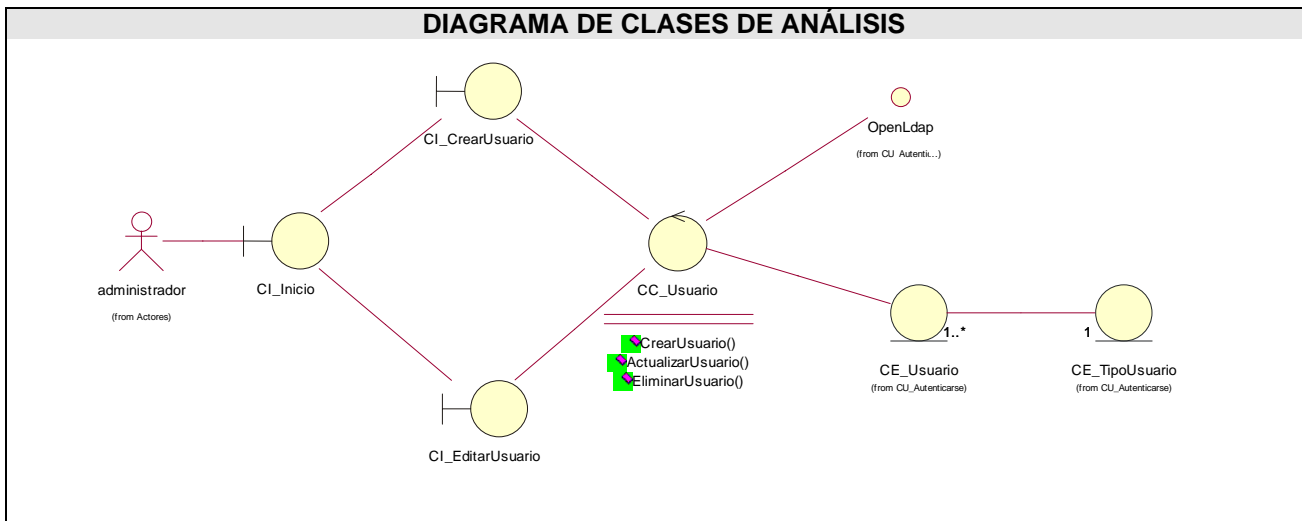
**Figura 3.4: Diagrama de Clases del Análisis CU Aprobar Opinión.**



**Figura 3.5: Diagrama de Clases del Análisis CU Gestionar Opiniones Archivadas.**

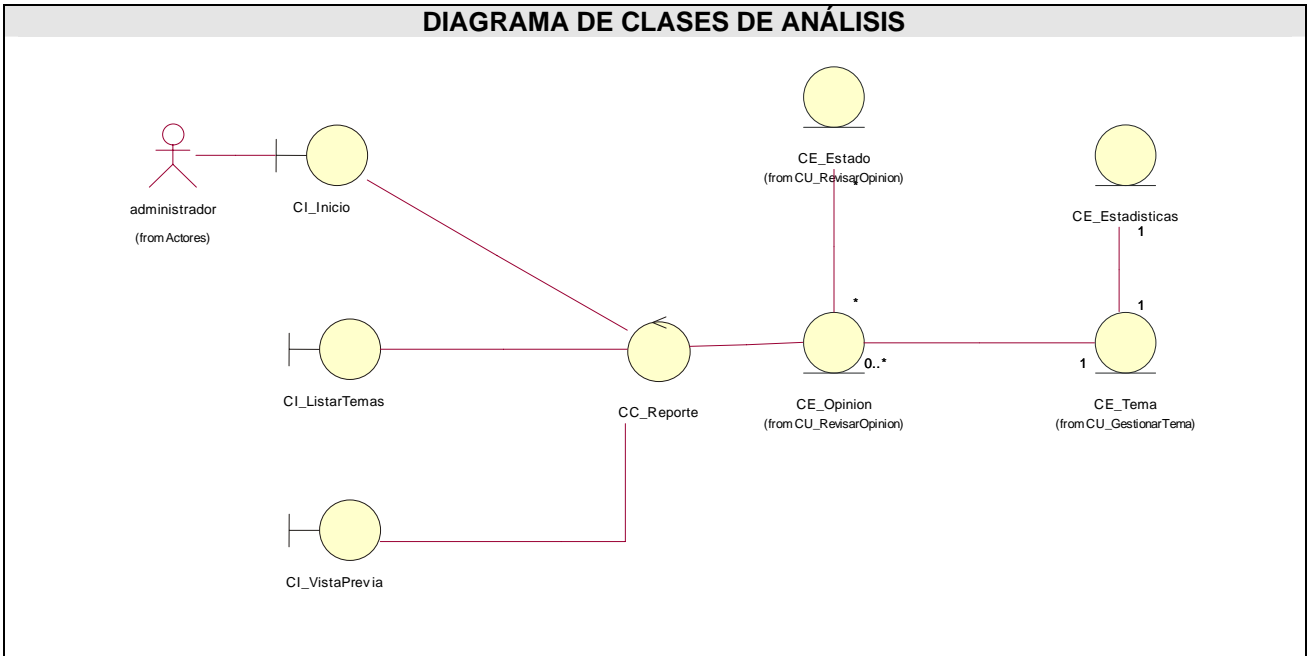


**Figura 3.6: Diagrama de Clases del Análisis CU Gestionar Tema.**

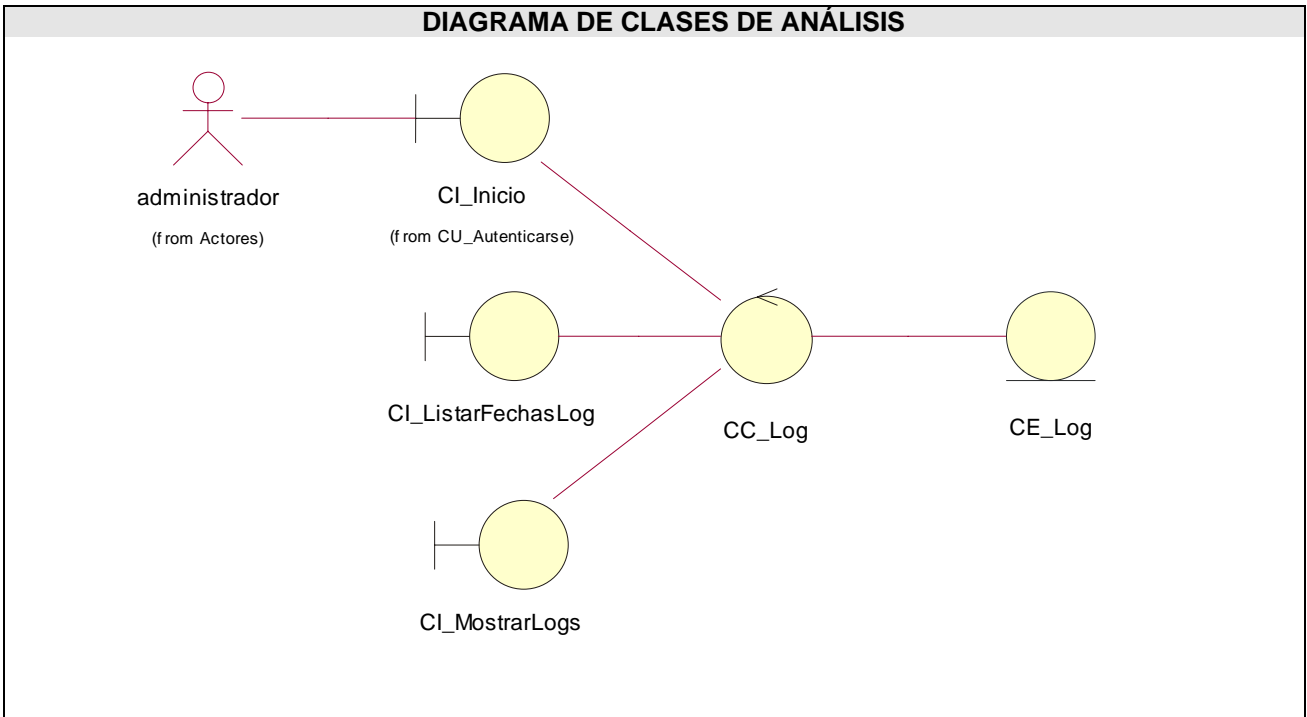


**Figura 3.7: Diagrama de Clases del Análisis CU Gestionar Usuario.**

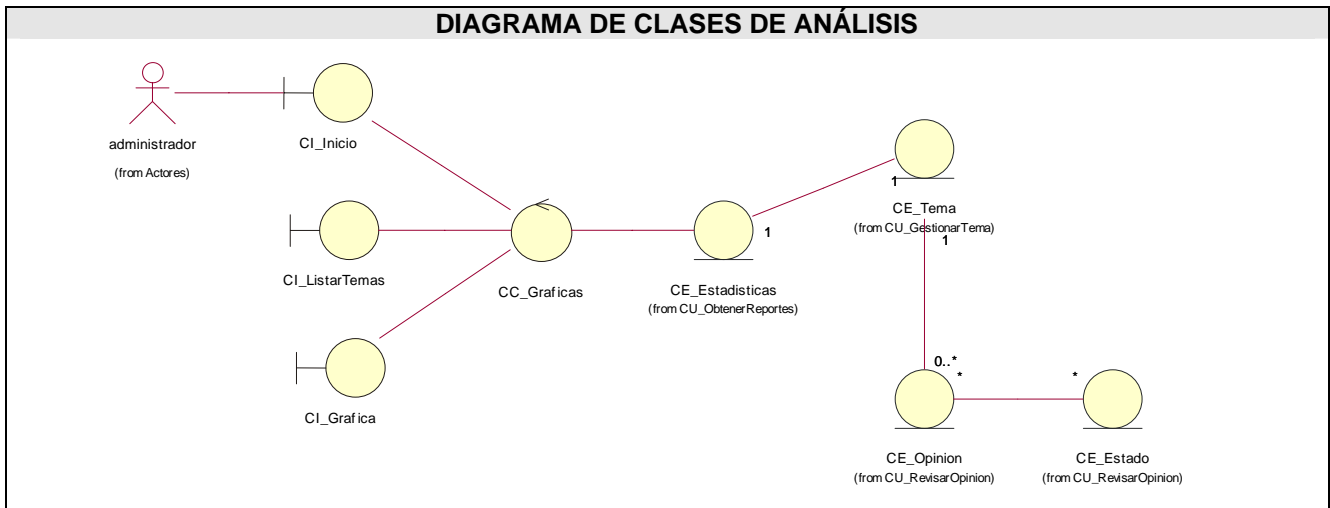




**Figura 3.8: Diagrama de Clases del Análisis CU Obtener Reporte.**



**Figura 3.9: Diagrama de Clases del Análisis CU Gestionar Logs.**



**Figura 3.10: Diagrama de Clases del Análisis CU Graficar Estadísticas.**

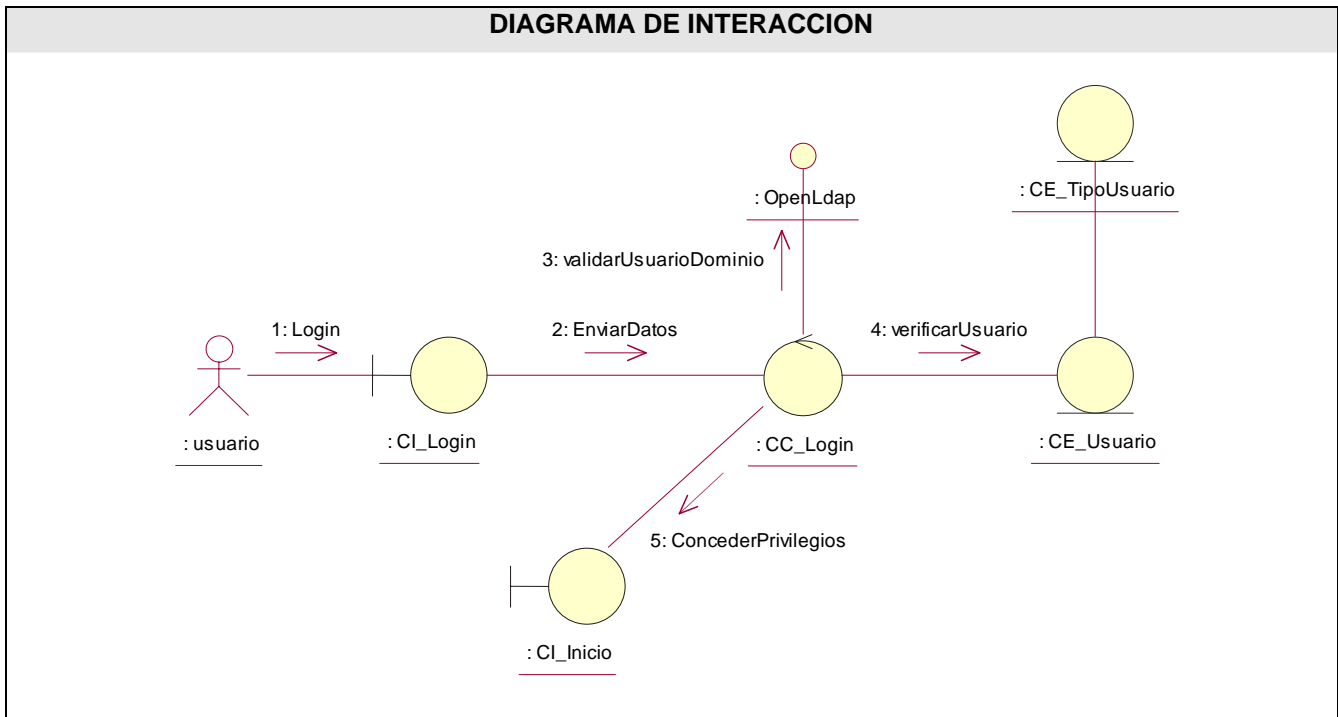
### 3.3 ANÁLISIS DE DISEÑO.

En el presente epígrafe se plasman los resultados de la etapa de diseño del sistema, utilizando UML. Se definen los diagramas de interacción, a partir de las operaciones de los diagramas de secuencia del sistema. Además, se elaboran los diagramas de clases, los cuales se obtienen como resultado del refinamiento del modelo conceptual.

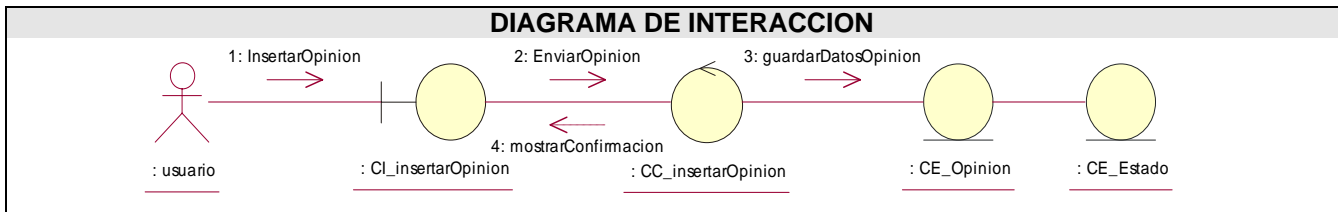
#### 3.3.1 Diagramas de interacción.

Los diagramas de interacción no son más que una descripción del modo en el que cada operación detectada, lleva a cabo sus responsabilidades y modifica el estado del sistema. En UML los diagramas de interacción pueden representarse a través de los Diagramas de Colaboración y/o Diagramas de Secuencia.

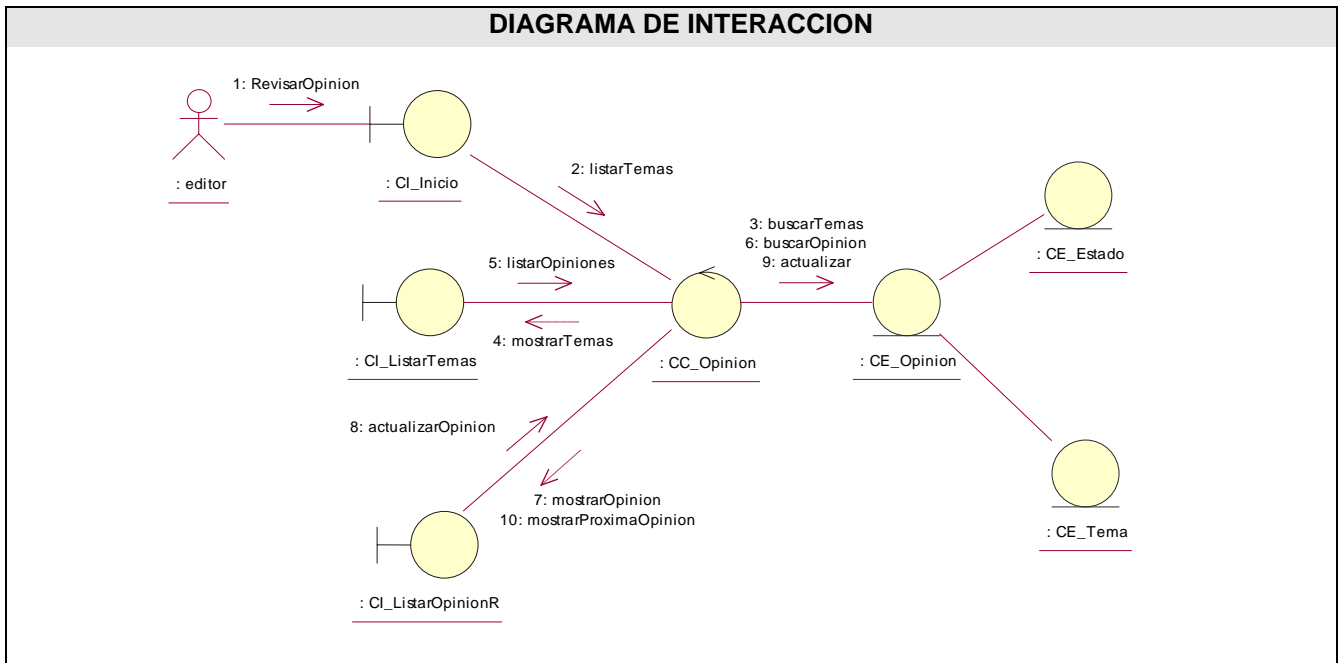
El tipo de diagrama seleccionado para construir los diagramas de interacción fue el de colaboración.



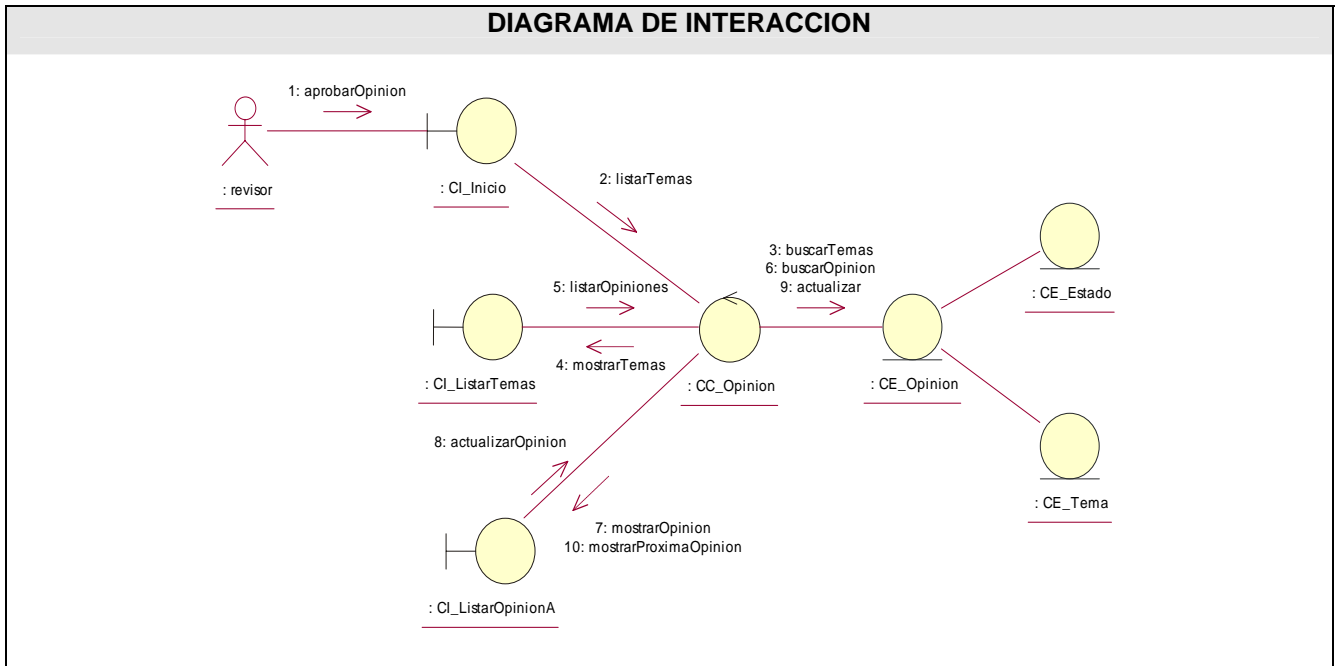
**Figura 3.11: Diagrama de Interacción CU Autenticarse.**



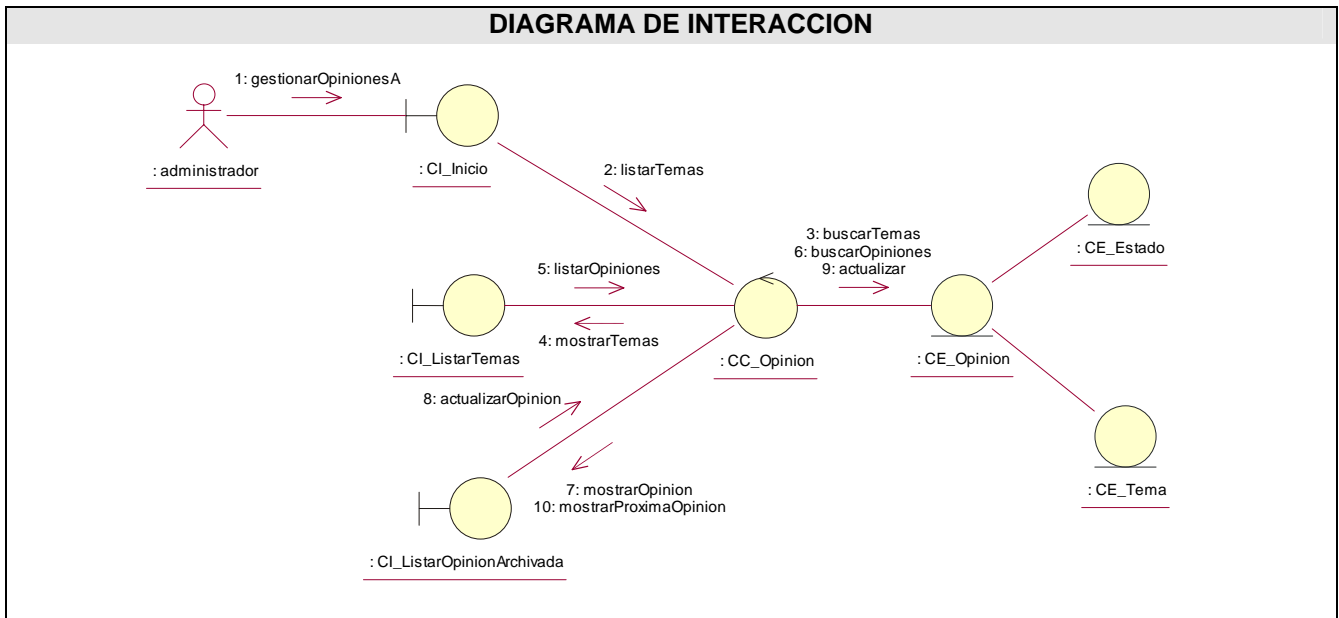
**Figura 3.12: Diagrama de Interacción CU Insertar Opinión.**



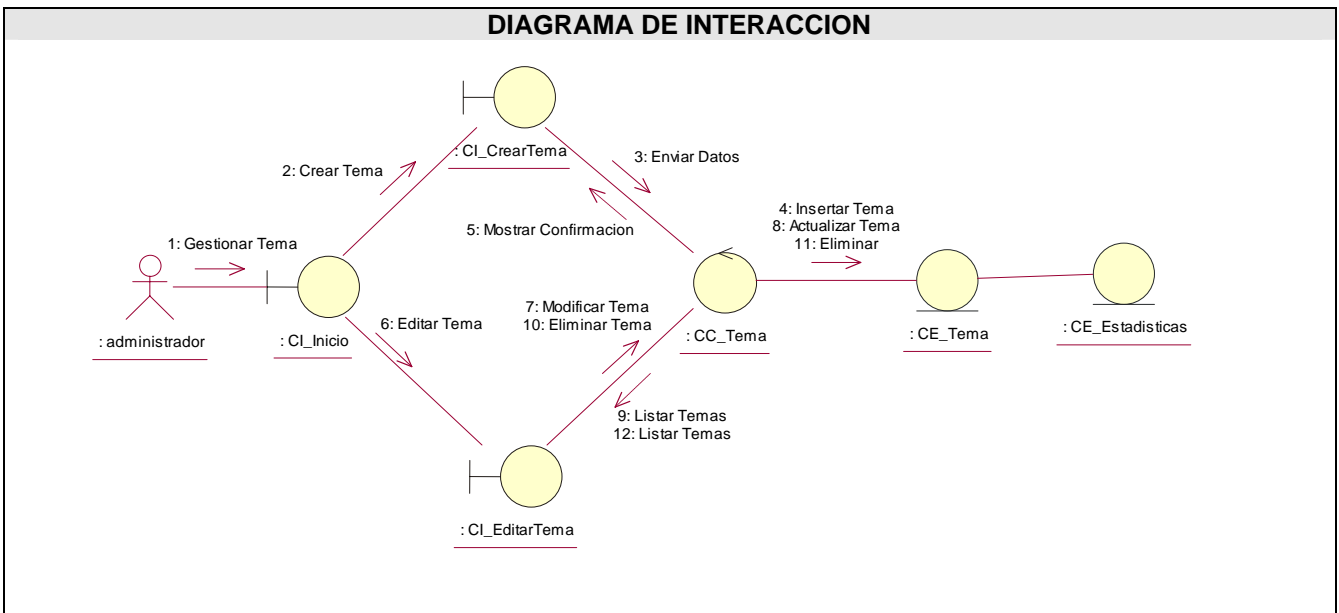
**Figura 3.13: Diagrama de Interacción CU Revisar Opinión.**



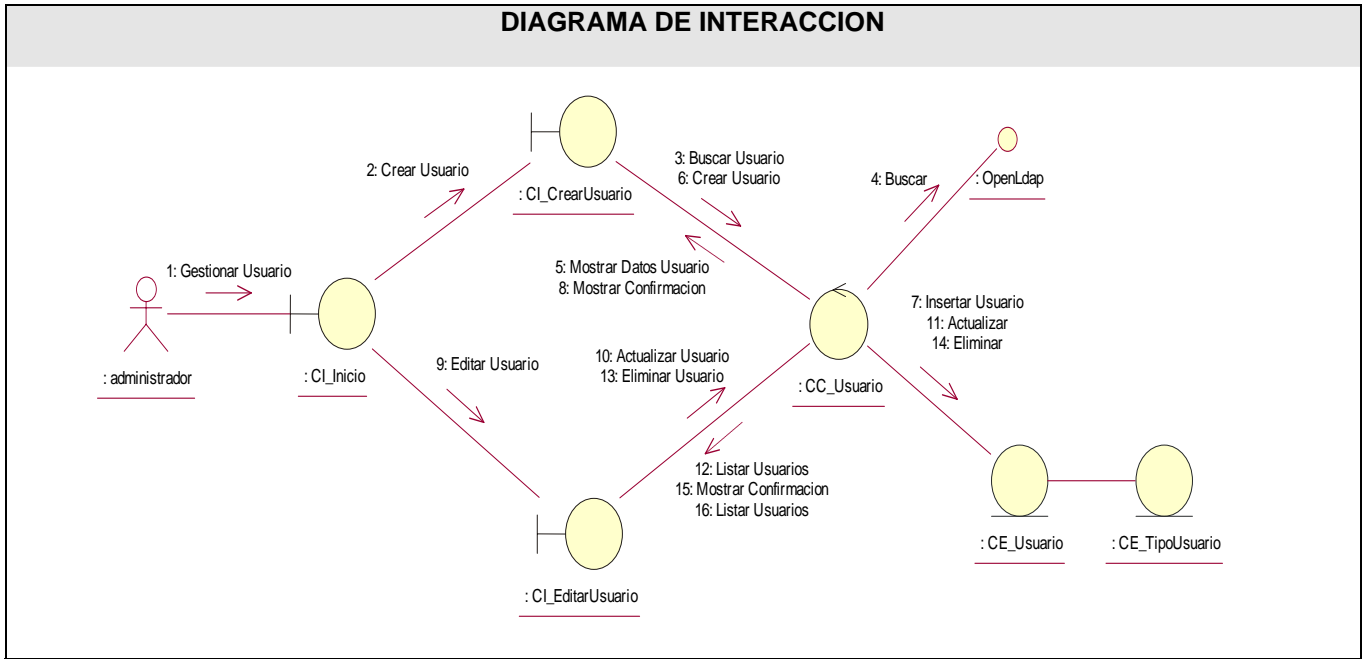
**Figura 3.14: Diagrama de Interacción CU Aprobar Opinión.**



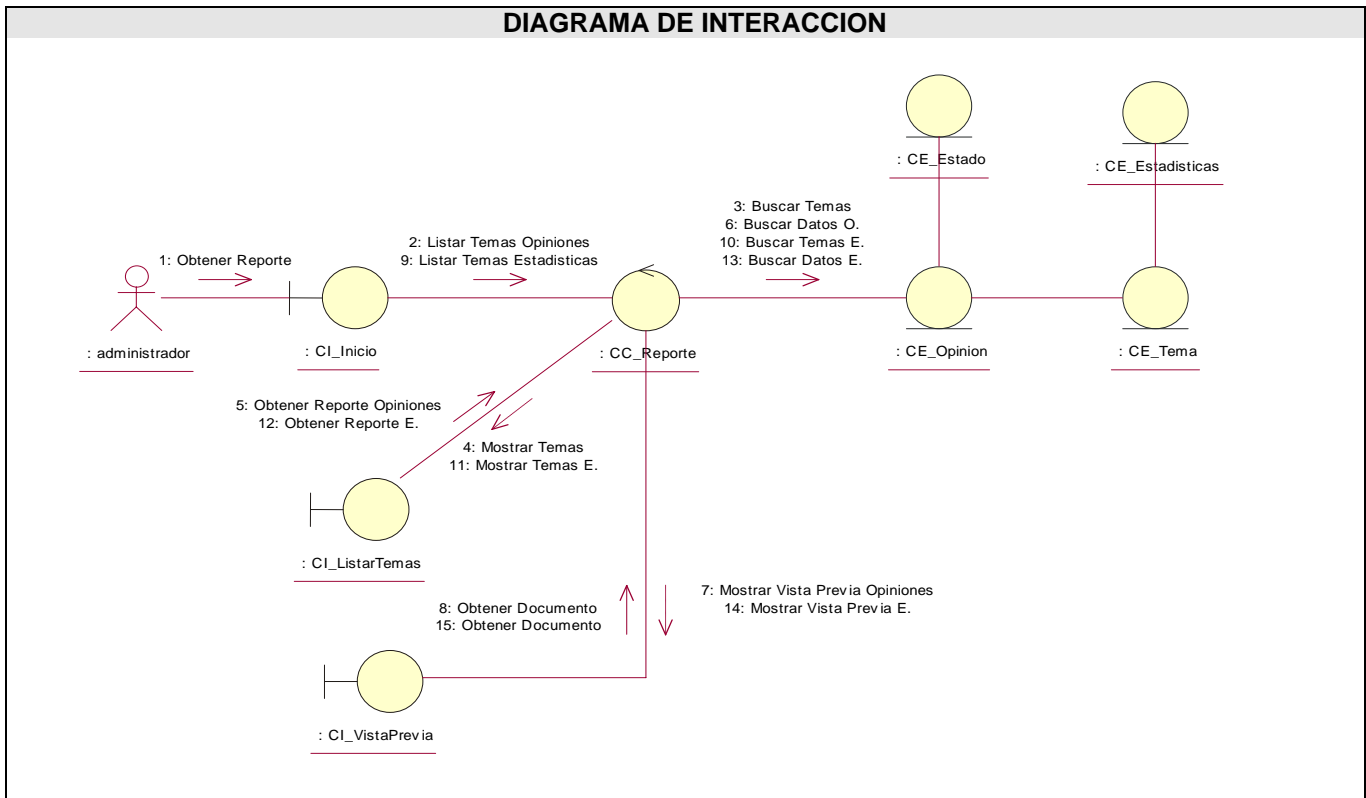
**Figura 3.15: Diagrama de Interacción CU Gestionar Opiniones Archivadas.**



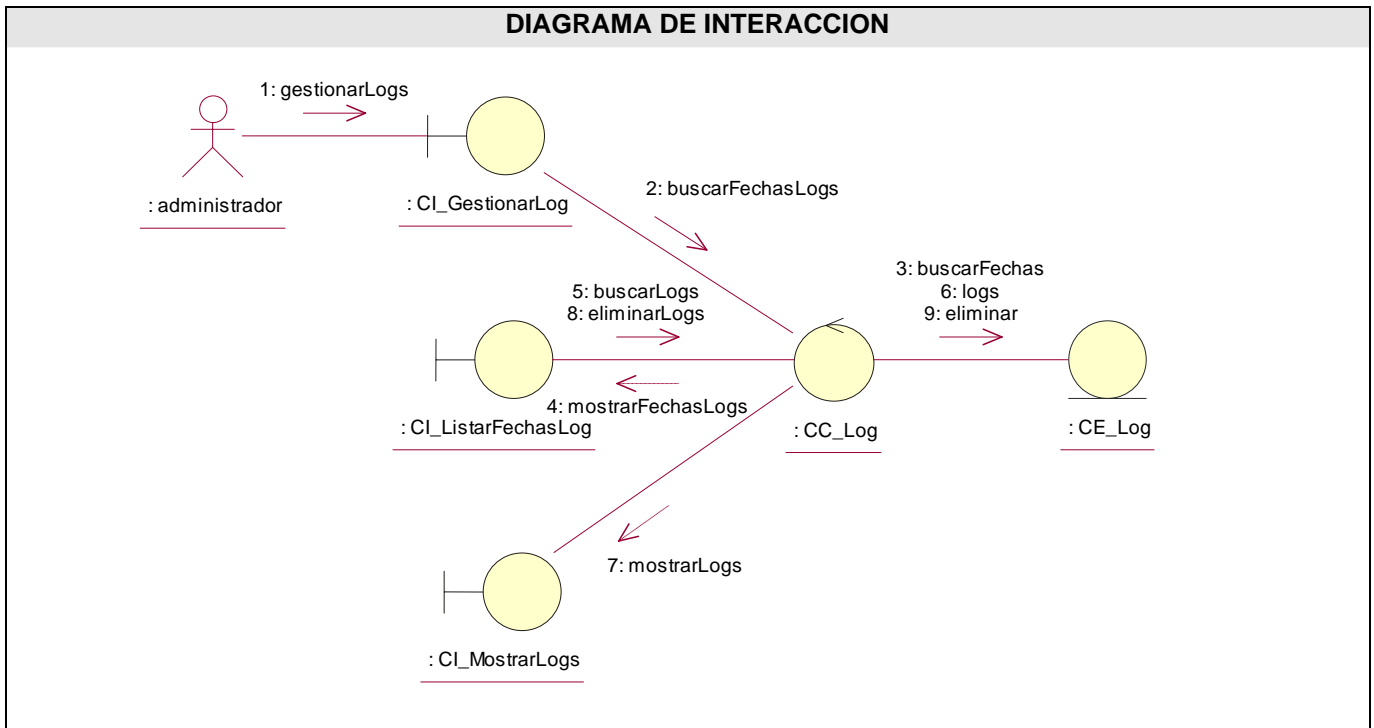
**Figura 3.16: Diagrama de Interacción CU Gestionar Tema.**



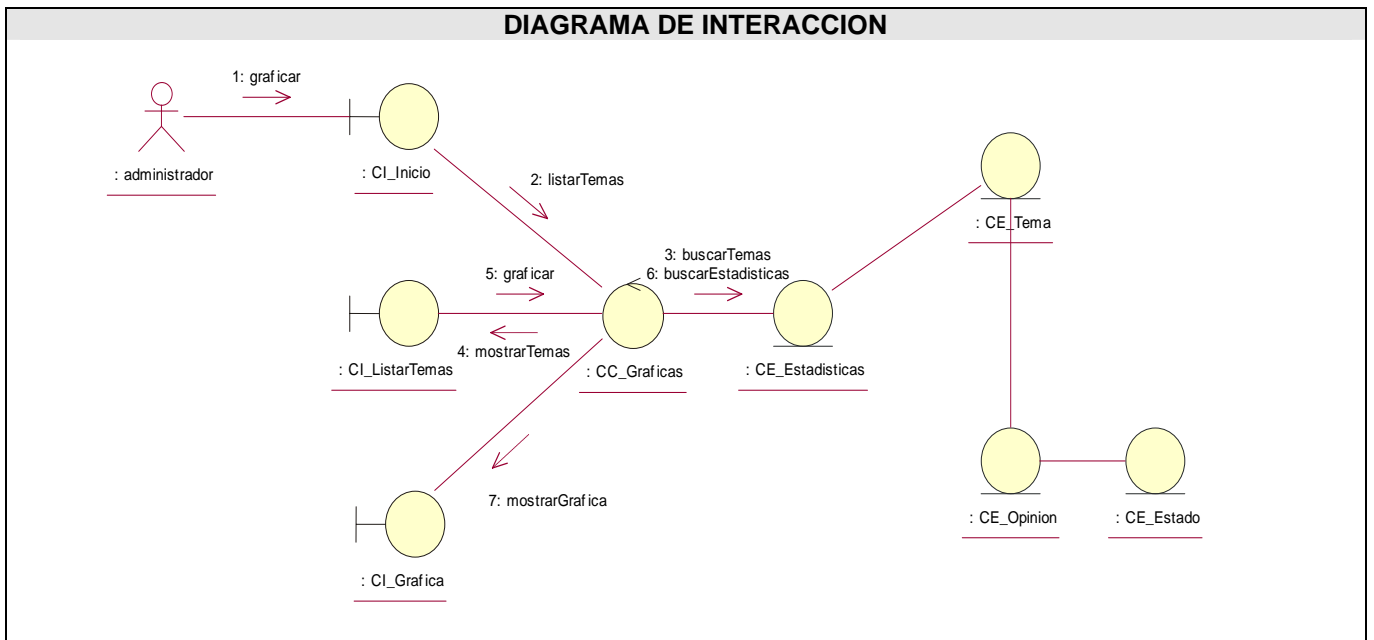
**Figura 3.17: Diagrama de Interacción CU Gestionar Usuario.**



**Figura 3.18: Diagrama de Interacción CU Obtener Reporte.**



**Figura 3.19: Diagrama de Interacción CU Gestionar Logs.**



**Figura 3.20: Diagrama de de Interacción CU Graficar Estadísticas.**

### 3.3.2 Diagrama de diseño Web del sistema.

El diagrama del diseño Web del sistema (diagrama de clases para diseño orientado a objetos) se obtiene como resultado del refinamiento del modelo conceptual y se basa fundamentalmente en los diagramas de interacción.

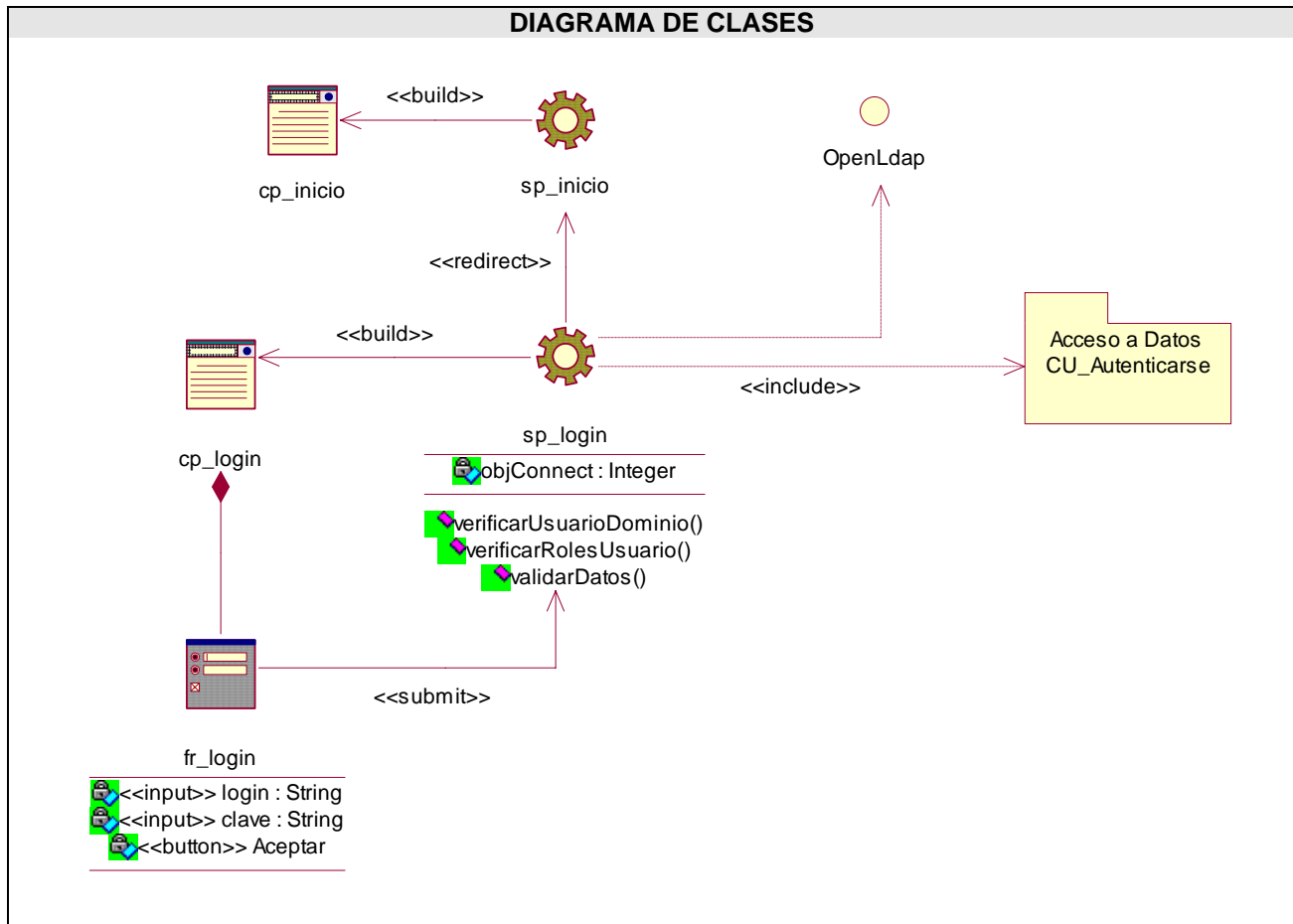


Figura 3.21: Diagrama de Clases del Diseño Web CU Autenticarse.



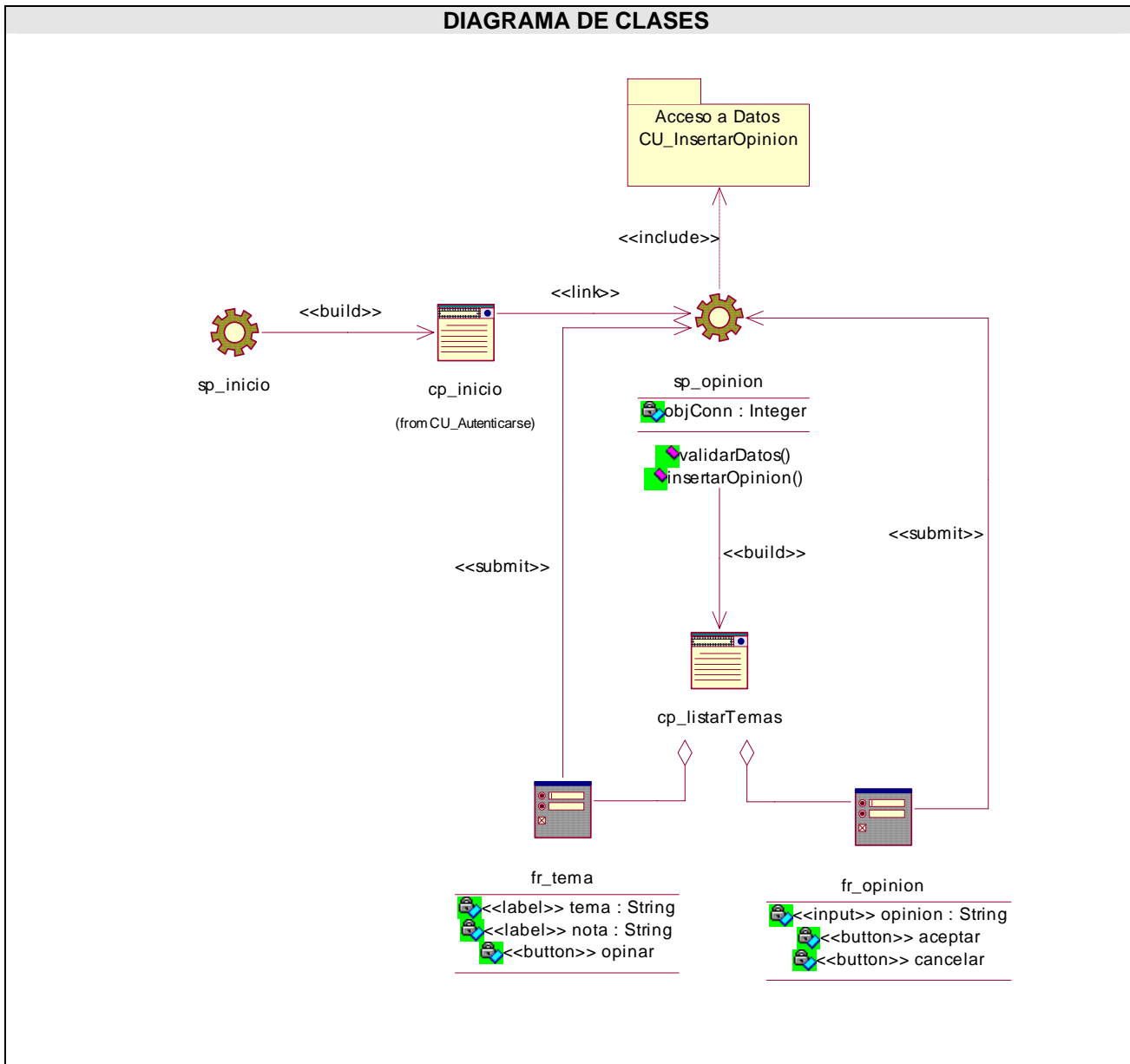


Figura 3.22: Diagrama de Clases del Diseño Web CU Insertar Opinión.

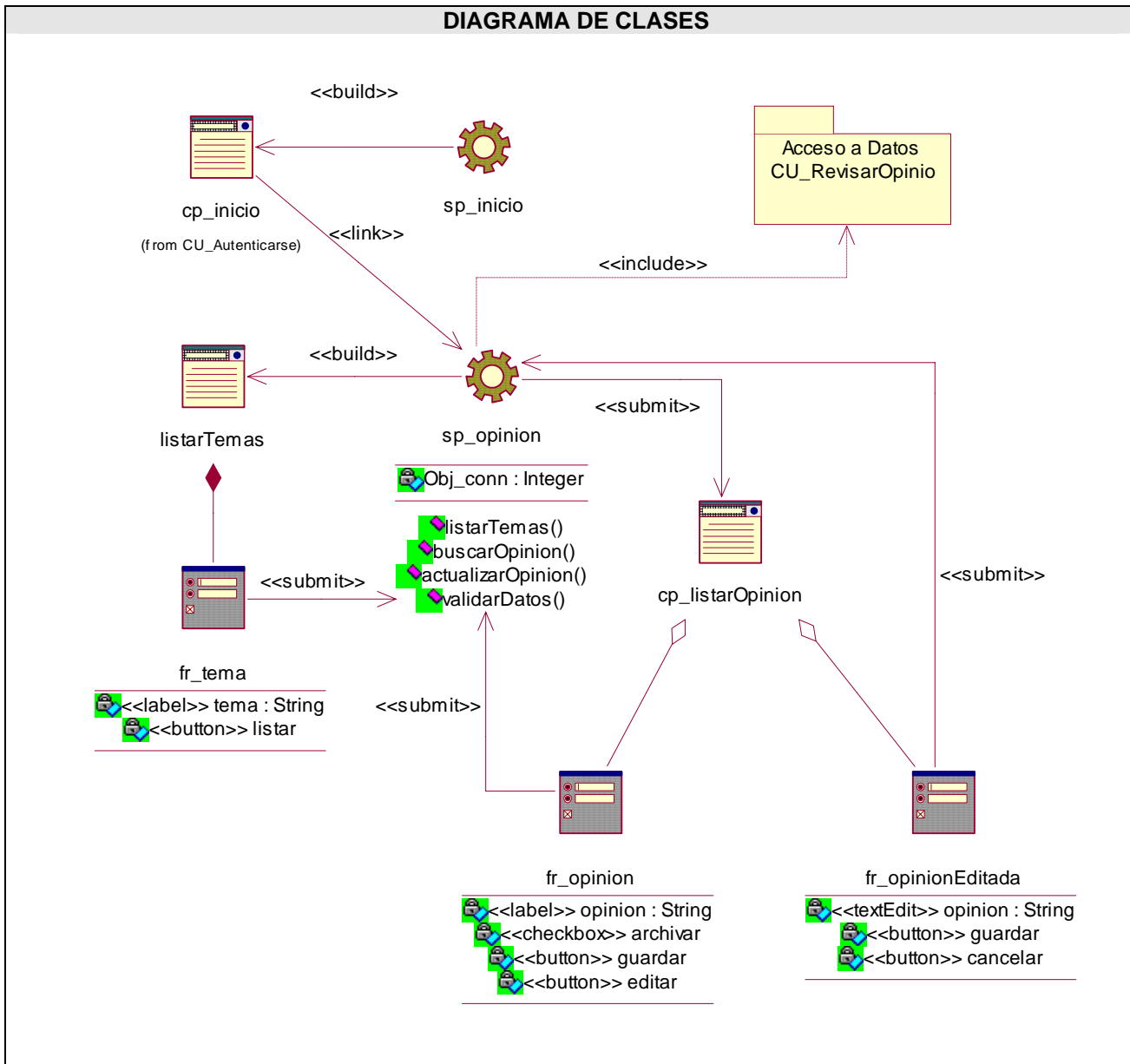


Figura 3.23: Diagrama de Clases del Diseño Web CU Revisar Opinión.

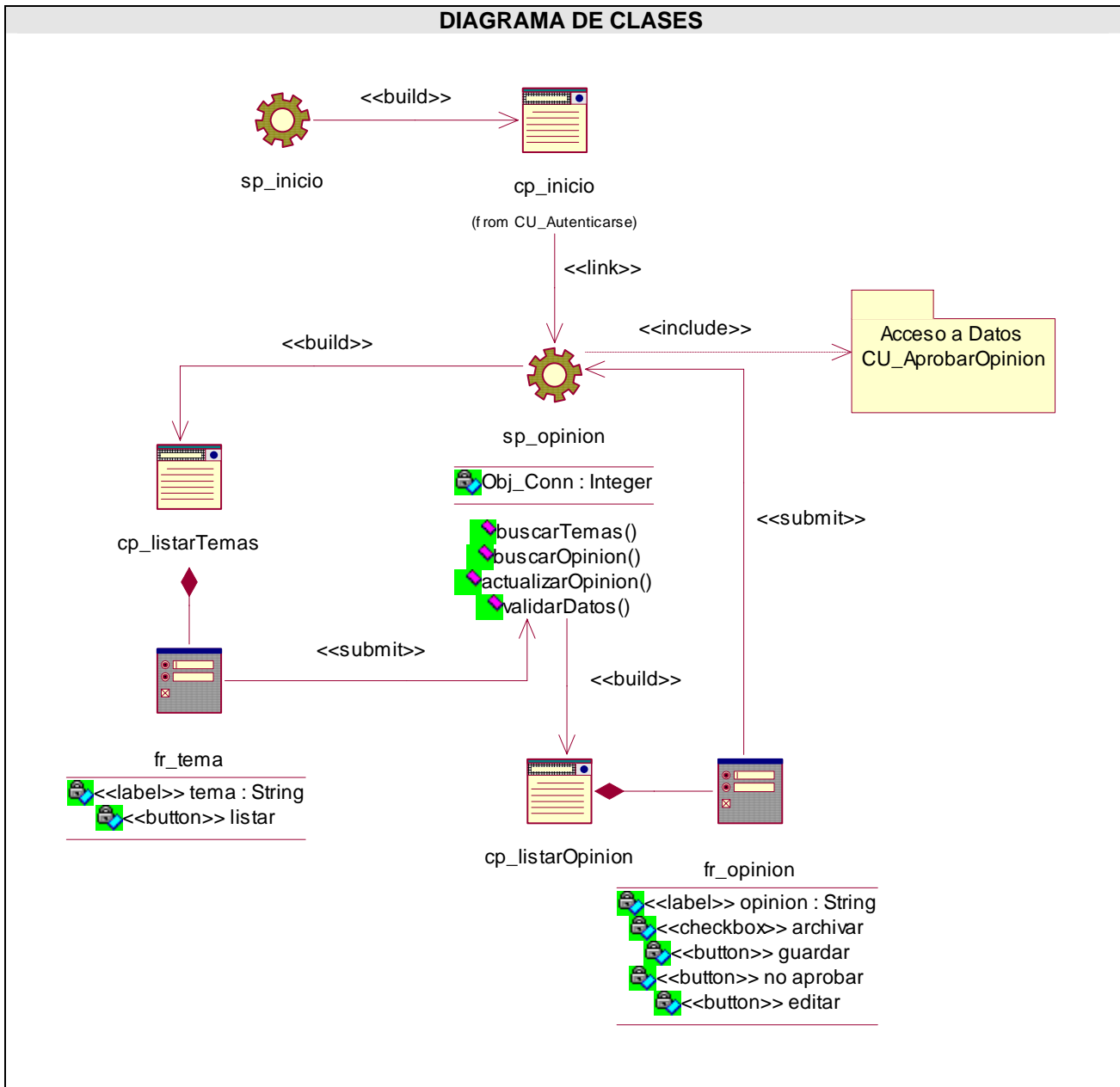
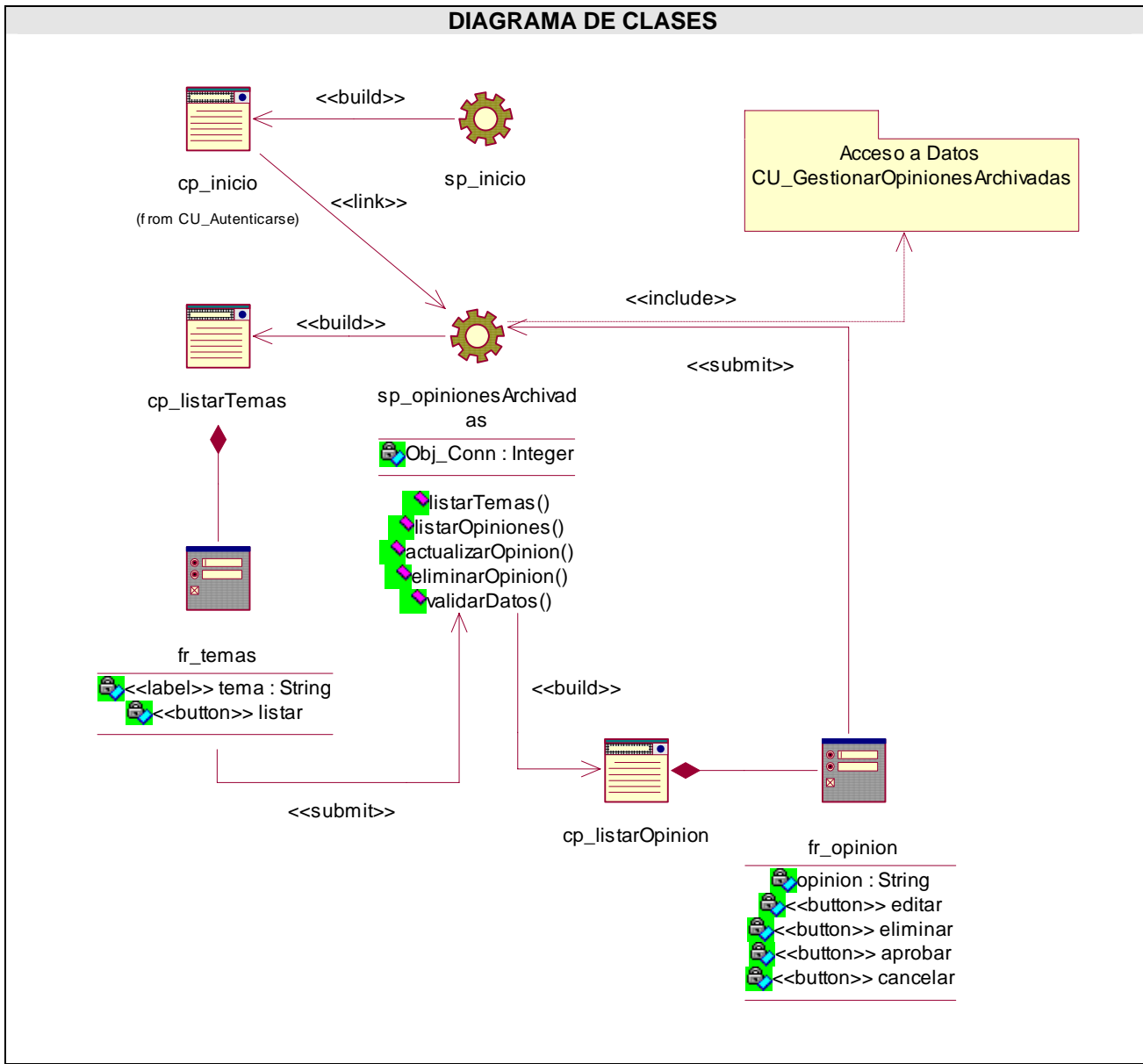


Figura 3.24: Diagrama de Clases del Diseño Web CU Aprobar Opinión.



**Figura 3.25: Diagrama de Clases del Diseño Web CU Gestionar Opiniones Archivadas.**

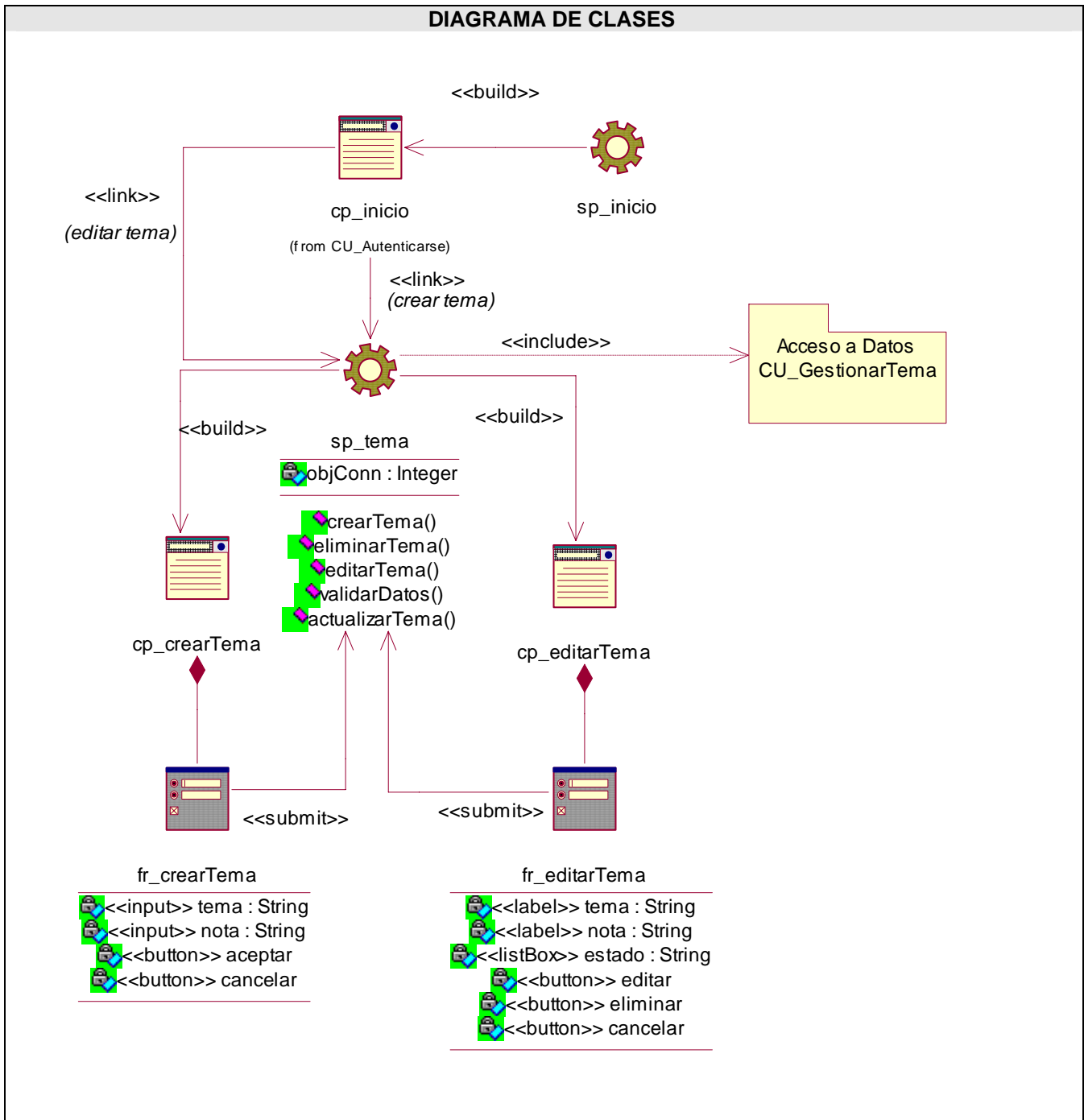


Figura 3.26: Diagrama de Clases del Diseño Web CU Gestionar Tema.

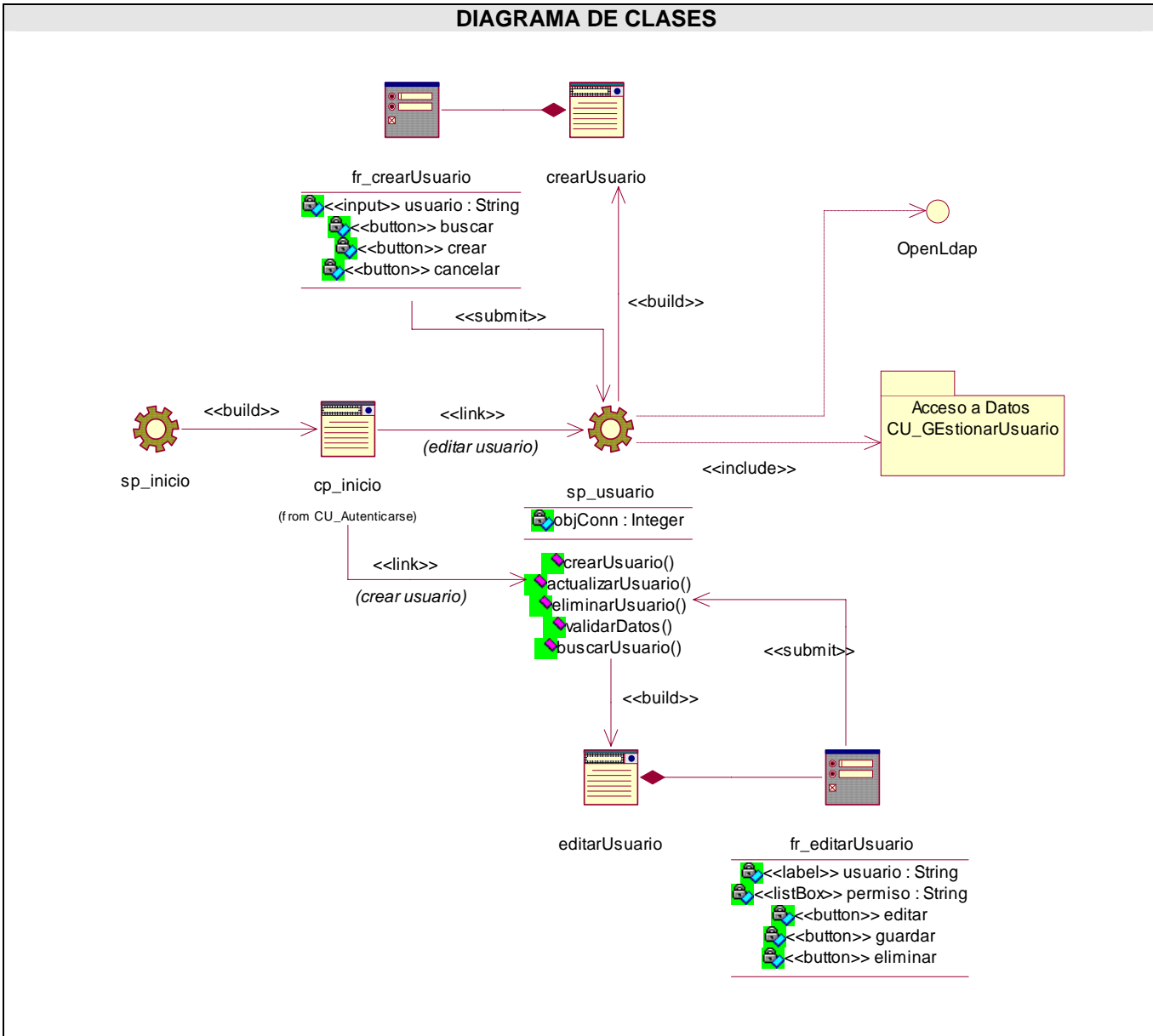


Figura 3.27: Diagrama de Clases del Diseño Web CU Gestionar Usuario.

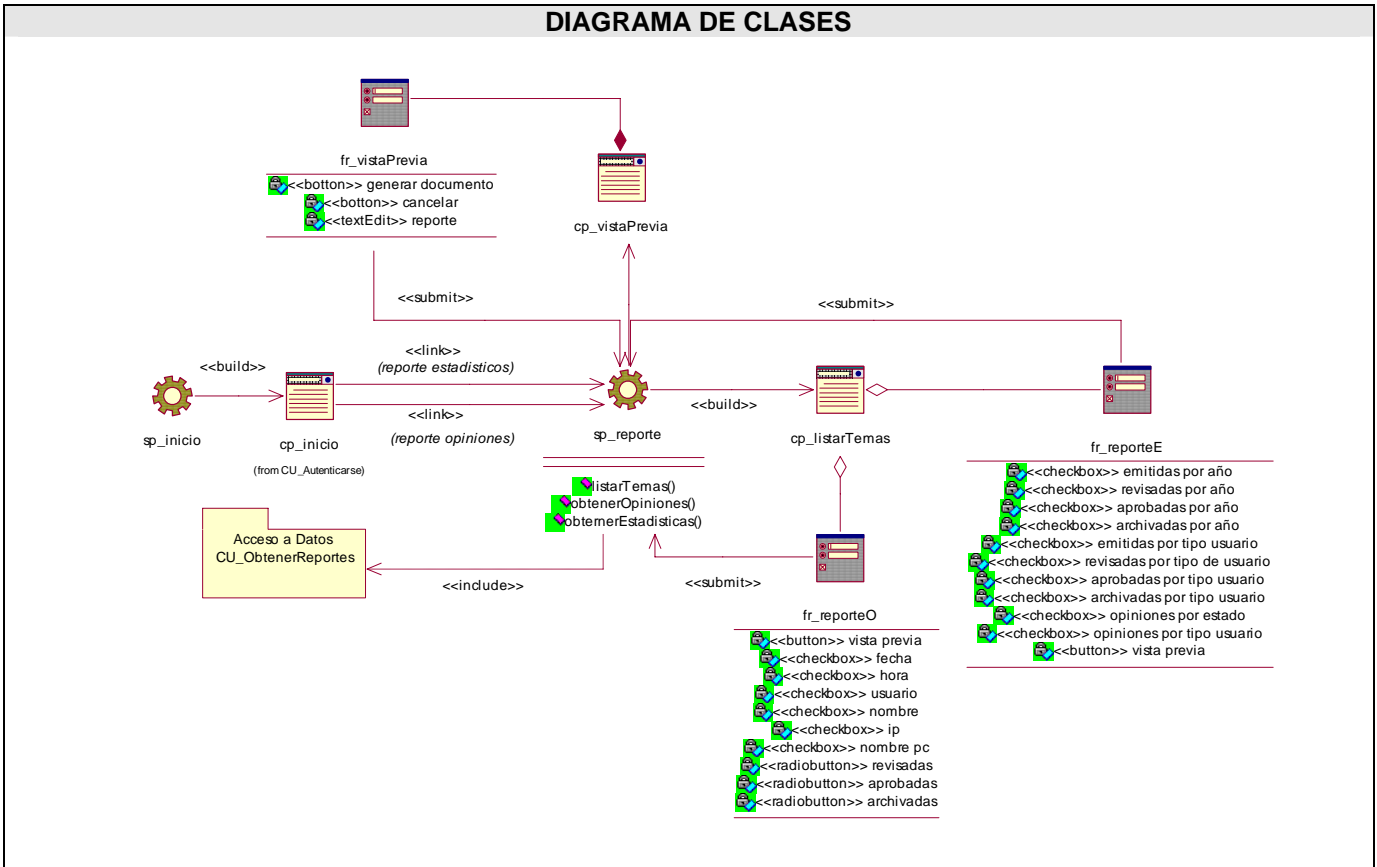
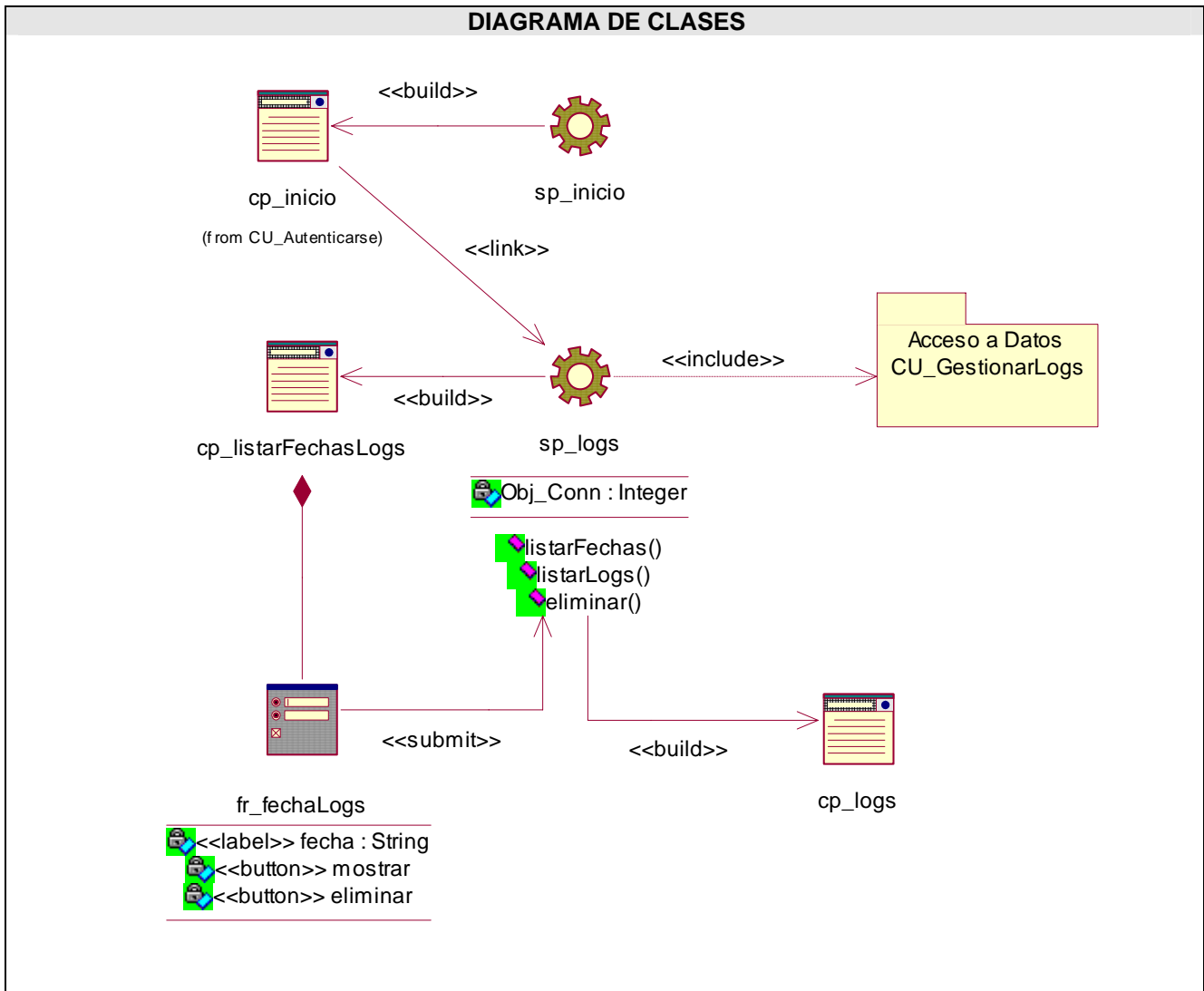


Figura 3.28: Diagrama de Clases del Diseño Web CU Obtener Reporte.



**Figura 3.29: Diagrama de Clases del Diseño Web CU Gestionar Logs.**



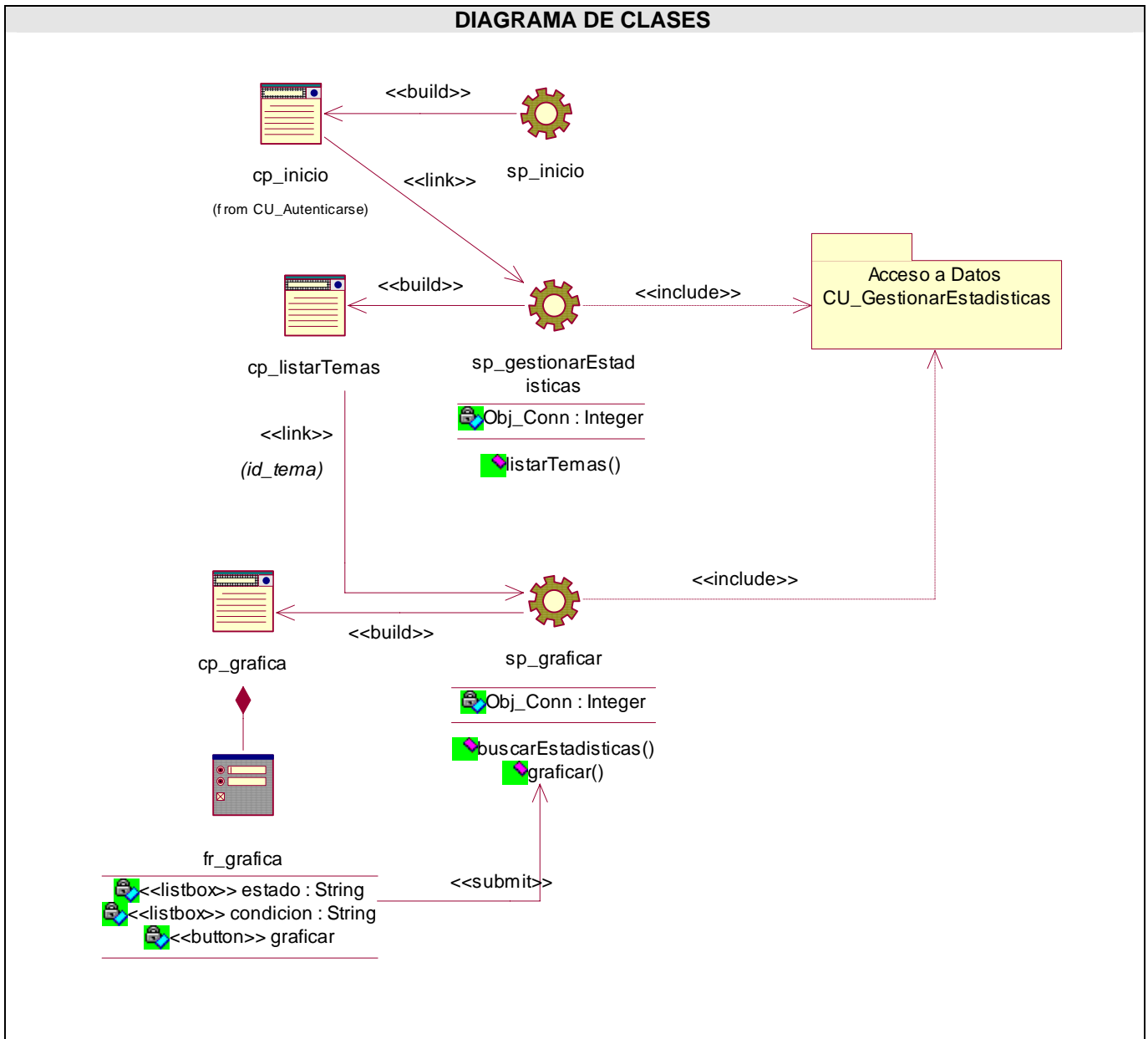


Figura 3.30: Diagrama de Clases del Diseño Web CU Graficar Estadísticas.

### 3.3.3 Descripción de las clases.

<b>Nombre:</b> CC_Opinion	
<b>Tipo de clase :</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
Obj_conn	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	listarTema()
<b>Descripción:</b>	Devuelve una lista con los temas existentes en la Base de Datos.
<b>Nombre:</b>	listarOpiniones()
<b>Descripción:</b>	Se encarga de listar las opiniones que van a ser revisadas una por una.
<b>Nombre:</b>	actualizarOpinion()
<b>Descripción:</b>	Actualiza la opinión en la base de datos.

<b>Nombre:</b> CC_Login	
<b>Tipo de clase :</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
Obj_conn	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	ldap()
<b>Descripción:</b>	Verifica que el usuario es un usuario válido del dominio.
<b>Nombre:</b>	login()
<b>Descripción:</b>	Verifica si el usuario tiene privilegios y concede permisos.

<b>Nombre:</b> CC_Graficas	
<b>Tipo de clase :</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
Obj_conn	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	listarTema()
<b>Descripción:</b>	Devuelve una lista con los temas existentes en la Base de Datos.
<b>Nombre:</b>	buscarEstadisticas()
<b>Descripción:</b>	Obtiene las estadísticas solicitadas de la base de datos.
<b>Nombre:</b>	graficar()
<b>Descripción:</b>	Crea las gráficas a partir de las estadísticas obtenidas.

<b>Nombre:</b> CC_Log	
<b>Tipo de clase :</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
Obj_conn	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	buscarFechasLogs()
<b>Descripción:</b>	Devuelve la lista de las fechas que contienen "logs".
<b>Nombre:</b>	listarLogs()
<b>Descripción:</b>	Lista los "logs" relacionados con la fecha seleccionada.
<b>Nombre:</b>	eliminarLogs()
<b>Descripción:</b>	Elimina de la base de datos los "logs" relacionados con la fecha seleccionada.

<b>Nombre:</b> CC_Tema	
<b>Tipo de clase :</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
Obj_conn	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	crearTema()
<b>Descripción:</b>	Guarda en la base de datos el tema creado.
<b>Nombre:</b>	actualizarTema()
<b>Descripción:</b>	Actualiza en la base de datos las modificaciones realizadas al tema.
<b>Nombre:</b>	eliminarTema()
<b>Descripción:</b>	Elimina de la base de datos el tema y todas las opiniones y estadísticas relacionadas con el mismo.

<b>Nombre:</b> CC_Usuario	
<b>Tipo de clase :</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
Obj_conn	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	crearUsuario()
<b>Descripción:</b>	Verifica que el usuario es un usuario válido del dominio y lo almacena en la base de datos con los permisos otorgados.
<b>Nombre:</b>	actualizarUsuario()
<b>Descripción:</b>	Actualiza los cambios realizados en los permisos del usuario.
<b>Nombre:</b>	eliminarUsuario()
<b>Descripción:</b>	Elimina el usuario seleccionado de la base de datos.

<b>Nombre:</b> CC_InsertarOpinion	
<b>Tipo de clase :</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
Obj_conn	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	listarTema()
<b>Descripción:</b>	Devuelve una lista con los temas existentes en la Base de Datos.
<b>Nombre:</b>	validarDatos()
<b>Descripción:</b>	Valida los datos entrados por el usuario.
<b>Nombre:</b>	insertarOpinion()
<b>Descripción:</b>	Inserta la opinión en la base de datos.

<b>Nombre:</b> CC_Reporte	
<b>Tipo de clase :</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
Obj_conn	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	listarTema()
<b>Descripción:</b>	Devuelve una lista con los temas existentes en la Base de Datos.
<b>Nombre:</b>	buscarEstadisticas()
<b>Descripción:</b>	Se encarga de obtener las estadísticas seleccionadas de la base de datos.
<b>Nombre:</b>	crearReporte()
<b>Descripción:</b>	Crea el reporte seleccionado.
<b>Nombre:</b>	generarDocumento()
<b>Descripción:</b>	Crea un documento en formato "rtf" con el reporte seleccionado.

<b>Nombre:</b> CC_Reporte	
<b>Tipo de clase :</b> Controladora	
<b>Atributo</b>	<b>Tipo</b>
Obj_conn	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	listarTema()
<b>Descripción:</b>	Devuelve una lista con los temas existentes en la Base de Datos.
<b>Nombre:</b>	buscarEstadisticas()
<b>Descripción:</b>	Se encarga de obtener las estadísticas seleccionadas de la base de datos.
<b>Nombre:</b>	crearReporte()
<b>Descripción:</b>	Crea el reporte seleccionado.
<b>Nombre:</b>	generarDocumento()
<b>Descripción:</b>	Crea un documento en formato "rtf" con el reporte seleccionado.

<b>Nombre:</b> CE_Tema	
<b>Tipo de clase :</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id_tema	Integer
nombre	String
nota	String
estado	Integer
fecha	Date
hora	Date
<b>Responsabilidades:</b>	
<b>Nombre:</b>	
<b>Descripción:</b>	

<b>Nombre:</b> CE_Estadística	
<b>Tipo de clase :</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id_tema	Integer
opiniones	Integer
revisadas	Integer
aprobadas	Integer
archivadas	Integer
estudiantes	Integer
profesores	Integer
trabajadores	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	
<b>Descripción:</b>	

<b>Nombre:</b> CE_Opinion	
<b>Tipo de clase :</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id_opinio	Integer
login	String
nombre	String
opinion	String
id_estado	Integer
fecha	Date
hora	Date
ip	String
pc	String
id_tema	Integer
tipo_usuario	String
en_revision	Integer
<b>Responsabilidades:</b>	

<b>Nombre:</b>	
<b>Descripción:</b>	

<b>Nombre:</b> CE_Estado	
<b>Tipo de clase :</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id_estado	Integer
Estado	String
<b>Responsabilidades:</b>	
<b>Nombre:</b>	
<b>Descripción:</b>	

<b>Nombre:</b> CE_Log	
<b>Tipo de clase :</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id_log	Integer
fecha	Date
hora	Date
ip	String
pc	String
login	String
accion	String
<b>Responsabilidades:</b>	
<b>Nombre:</b>	
<b>Descripción:</b>	

<b>Nombre:</b> CE_Usuario	
<b>Tipo de clase :</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
login	String
tipo_usuario	Integer
fecha	Date
hora	Date
id_usuario	Integer
<b>Responsabilidades:</b>	
<b>Nombre:</b>	
<b>Descripción:</b>	

<b>Nombre:</b> CE_TipoUsuario	
<b>Tipo de clase :</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
id_tipo_usuario	Integer
tipo_usuario	String
<b>Responsabilidades:</b>	
<b>Nombre:</b>	
<b>Descripción:</b>	

<b>Nombre:</b> CI_listarTemas	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
tema	<<label>>:String
listar	<<button>>
<b>Responsabilidades:</b>	
<b>Nombre:</b>	
<b>Descripción:</b>	

<b>Nombre:</b> CI_listarOpinion	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
opinion	<<label>>:String
archivar	<<checkbox>>
guardar	<<button>>
no guardar	<<button>>
editar	<<button>>
cancelar	<<button>>
url	String
id	Int
id_tema	Int
<b>Responsabilidades:</b>	
<b>Nombre:</b>	ajax(url,id)
<b>Descripción:</b>	Se encarga de mostrar la opinión de forma editable con el uso de la tecnología "AJAX".

<b>Nombre:</b> CI_login	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
usuario	<<textEdit>>:String
clave	<<textEdit>>String
login	<<button>>
<b>Responsabilidades:</b>	
<b>Nombre:</b>	passw_encrypt()
<b>Descripción:</b>	Se encarga de encriptar la clave antes de enviarla para el servidor.

<b>Nombre:</b> CI_Grafica	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
estado	<<select>>:String
condicion	<<select>>:String
graficar	<<button>>
url	String
id	Int
<b>Responsabilidades:</b>	
<b>Nombre:</b>	ajax(url,id)
<b>Descripción:</b>	Se encarga de mostrar las imágenes graficadas, con el uso de la tecnología "AJAX".

<b>Nombre:</b> CI_listarFechasLogs	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
fecha	<<label>>:String
mostrar	<<button>>
eliminar	<<button>>
<b>Responsabilidades:</b>	
<b>Nombre:</b>	confirmation(form)
<b>Descripción:</b>	Se encarga de preguntar al usuario si realmente desea realizar la operación selecciona, en caso afirmativo se ejecuta, de lo contrario no se realiza ninguna acción.

<b>Nombre:</b> CI_MostrarLogs	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
hora	<<label>>:String
ip	<<label>>:String
pc	<<label>>:String
login	<<label>>:String
accion	<<label>>:String
<b>Responsabilidades:</b>	
<b>Nombre:</b>	
<b>Descripción:</b>	



<b>Nombre:</b> CI_listarOpinionArchivada	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
opinion	<<label>>:String
guardar	<<button>>
eliminar	<<button>>
editar	<<button>>
cancelar	<<button>>
url	String
id	Int
id_tema	Int
<b>Responsabilidades:</b>	
<b>Nombre:</b>	ajax(url,id)
<b>Descripción:</b>	Se encarga de mostrar la opinión de forma editable con el uso de la tecnología "AJAX".
<b>Nombre:</b>	confirmation(form)
<b>Descripción:</b>	Se encarga de preguntar al usuario si realmente desea realizar la operación selecciona, en caso afirmativo se ejecuta, de lo contrario no se realiza ninguna acción.

<b>Nombre:</b> CI_CrearTema	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
tema	<<textEdit>>:String
nota	<<textEdit>>:String
guardar	<<button>>
limpiar	<<button>>
<b>Responsabilidades:</b>	
<b>Nombre:</b>	
<b>Descripción:</b>	

<b>Nombre:</b> CI_vistaPrevia	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
reporte	<<label>>:String
generar documento	<<button>>
cancelar	<<button>>
url	String
<b>Responsabilidades:</b>	
<b>Nombre:</b>	
<b>Descripción:</b>	

<b>Nombre:</b> CI_EditarTema	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
tema	<<textEdit>>:String
nota	<<textEdit>>:String
estado	<<listBox>>:String
actualizar	<<button>>
editar	<<button>>
cancelar	<<button>>
eliminar	<<button>>
url	String
id	Int
<b>Responsabilidades:</b>	
<b>Nombre:</b>	ajax(url,id)
<b>Descripción:</b>	Se encarga de mostrar las propiedades del tema de forma editable con el uso de la tecnología "AJAX"
<b>Nombre:</b>	confirmation(form)
<b>Descripción:</b>	Se encarga de preguntar al usuario si realmente desea realizar la operación selecciona, en caso afirmativo se ejecuta, de lo contrario no se realiza ninguna acción.

<b>Nombre:</b> CI_CrearUsuario	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
usuario	<<textEdit>>:String
nombre	<<textEdit>>:String
permiso	<<listBox>>:String
buscar	<<button>>
crear	<<button>>
cancelar	<<button>>
url	String
id	Int
idu	Int
<b>Responsabilidades:</b>	
<b>Nombre:</b>	ajaxUser(url,id,idu)
<b>Descripción:</b>	Se encarga de buscar el usuario entrado y mostrar sus datos, con el uso de "AJAX".

<b>Nombre:</b> CI_EditarUsuario	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
usuario	<<label>>:String
permiso	<<label>>:String
editar	<<button>>
eliminar	<<button>>
cancelar	<<button>>
actualizar	<<button>>
url	String
id	Int
<b>Responsabilidades:</b>	
<b>Nombre:</b>	ajax(url,id)
<b>Descripción:</b>	Se encarga de mostrar los campos del usuario de forma editable con el uso de la tecnología "AJAX".
<b>Nombre:</b>	confirmation(form)
<b>Descripción:</b>	Se encarga de preguntar al usuario si realmente desea realizar la operación selecciona, en caso afirmativo se ejecuta, de lo contrario no se realiza ninguna acción.

<b>Nombre:</b> CI_insertarOpinion	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
tema	<<label>>:String
nota	<<label>>:String
opiniones	<<label>>:Int
opinar	<<button>>
enviar	<<button>>
cancelar	<<button>>
opinion	<<textEdit>>:String
url	String
id	Int
<b>Responsabilidades:</b>	
<b>Nombre:</b>	ajax(url,id)
<b>Descripción:</b>	Se encarga de mostrar los campos del usuario de forma editable con el uso de la tecnología "AJAX".

<b>Nombre:</b> CI_listarOpinionR	
<b>Tipo de clase:</b> Interfaz	
<b>Atributo</b>	<b>Tipo</b>
opinion	<<label>>:String
archivar	<<checkbox>>
guardar	<<button>>
no guardar	<<button>>
editar	<<button>>
cancelar	<<button>>
url	String
id	Int
id_tema	Int
<b>Responsabilidades:</b>	
<b>Nombre:</b>	ajax(url,id)
<b>Descripción:</b>	Se encarga de mostrar la opinión de forma editable con el uso de la tecnología "AJAX".

### 3.3.4 Diseño de la base de datos.

Con el objetivo de lograr la persistencia de los datos y que estos puedan ser consultados en cualquier momento por cualquier usuario se utilizó, en la concepción de este prototipo, una base de datos relacional. El uso de esta técnica de almacenamiento está muy difundido actualmente debido a las garantías que ofrece en cuanto a la durabilidad de los datos se refiere. La base de datos del sistema es extensa, y se utiliza para almacenar datos referentes a los servicios que brinda el sitio relacionado con las opiniones y los usuarios del sistema.

#### 3.3.4.1 Diagrama Entidad Relación de la Base de Datos.

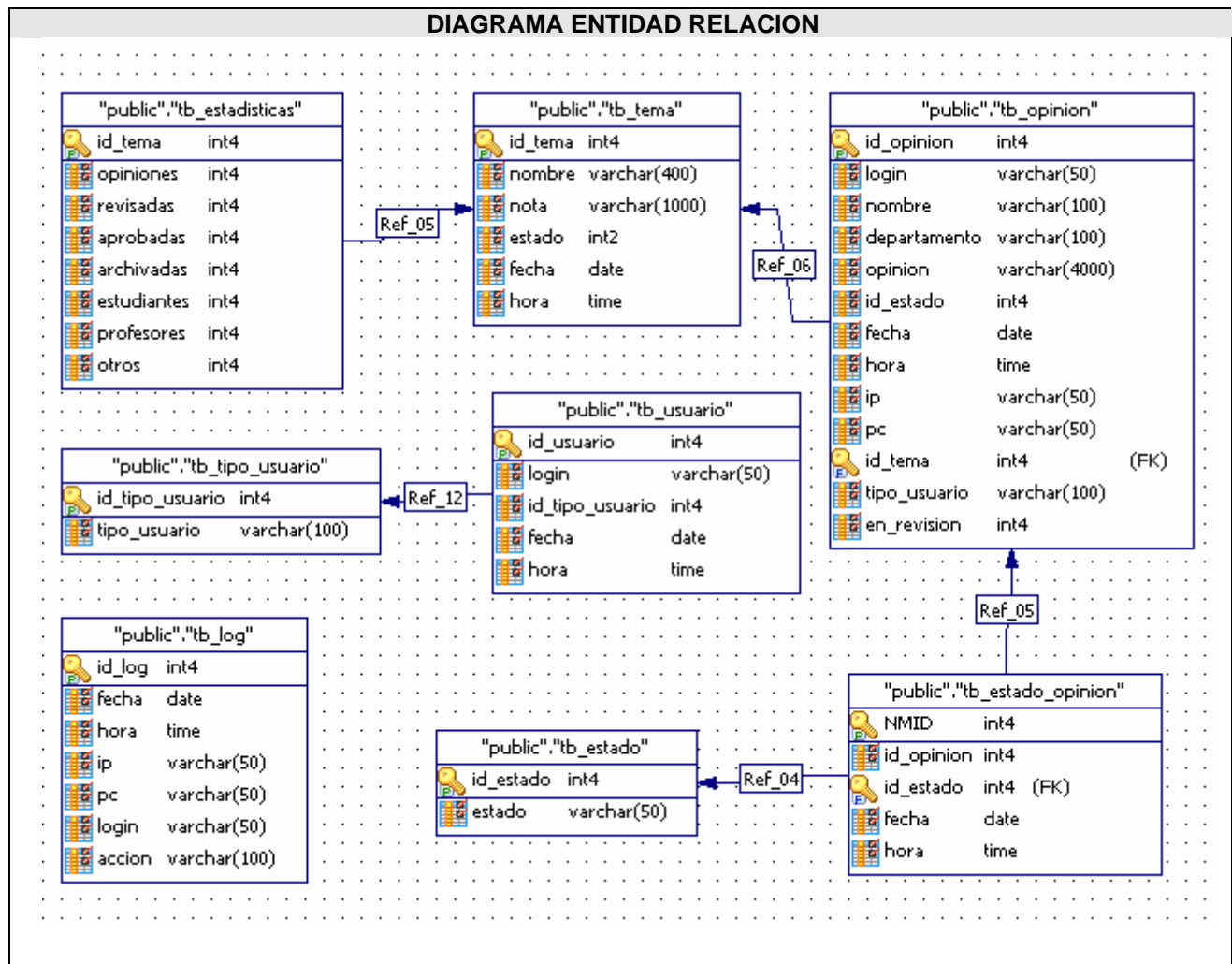


Figura 3.25: Diagrama Entidad Relación de la BD.

### 3.3.4.2 Descripción de las tablas.

Nombre: <b>tb_tema</b>		
Descripción: Contiene los datos de los temas de debate.		
Nombre	Tipo	Descripción
id_tema	Int4	Identificador del tema de debate.
nombre	Varchar(400)	Nombre del tema de debate.
nota	Varchar(1000)	Nota relacionada con el tema de debate.
estado	Int4	Estado en que se encuentra el tema.
fecha	Date	Fecha de creado el tema.
hora	Time	Hora de creado el tema.

<b>Nombre: tb_opinion</b>		
<b>Descripción:</b> Contiene los datos referentes a las opiniones emitidas por los usuarios.		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
Id_opinion	Int4	Identificador de la opinión.
login	Varchar(50)	Nombre de usuario del dominio UCI.
nombre	Varchar(400)	Nombre completo del usuario.
departamento	Varchar(100)	Especifica el año de estudio del usuario para caso de los estudiantes.
opinion	Varchar(4000)	Opinión emitida por el usuario.
id_estado	Int4	Identificador del estado en que se encuentra la opinión
fecha	Date	Fecha de emitida la opinión
hora	Time	Hora de emitida la opinión
ip	Varchar(50)	Dirección ip de la máquina desde donde se emitió la opinión.
pc	Varchar(50)	Nombre de la máquina desde donde se emitió la opinión.
id_tema	Int4	Identificador del tema de debate.
tipo_usuario	Varchar(100)	Tipo de usuario que emite la opinión.
en_revisión	Int4	Estado de la opinión.

<b>Nombre: tb_estadisticas</b>		
<b>Descripción:</b> Contiene las estadísticas relacionadas con los temas de debate y las opiniones.		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
id_tema	Int4	Identificador del tema de debate.
opiniones	Int4	Total de opiniones emitidas.
revisadas	Int4	Total de opiniones revisadas.
aprobadas	Int4	Total de opiniones aprobadas.
archivadas	Int4	Total de opiniones archivadas.
estudiantes	Int4	Total de opiniones emitidas por estudiantes.
profesores	Int4	Total de opiniones emitidas por profesores.
otros	Int4	Total de opiniones emitidas por trabajadores.

<b>Nombre: tb_estado</b>		
<b>Descripción:</b> Contiene los distintos estados de las opiniones.		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
id_estado	Int4	Identificador del estado.
estado	Varchar(50)	Estado de la opinión.

<b>Nombre: tb_usuario</b>		
<b>Descripción:</b> Contiene los nombre de usuario de los administradores del sistema.		
Nombre	Tipo	Descripción
id_usuario	Int4	Identificador del usuario.
login	Varchar(50)	Nombre de usuario del dominio UCI.
id_tipo_usuario	Int4	Identificador del tipo de rol que posee el usuario.
fecha	Date	Fecha de creado el usuario.
hora	Time	Hora de creado el usuario.

<b>Nombre: tb_tipo_usuario</b>		
<b>Descripción:</b> Contiene los tipos de roles de los administradores del sistema.		
Nombre	Tipo	Descripción
id_tipo_usuario	Int4	Identificador del rol.
tipo_usuario	Varchar(50)	Nombre del rol.

<b>Nombre: tb_datos</b>		
<b>Descripción:</b> Contiene los datos relacionados con las visitas al sitio.		
Nombre	Tipo	Descripción
fecha	Date	Fecha de la visita.
visitas	Int4	Total de visitas.
conectados	Int4	Usuarios conectados.

<b>Nombre: tb_log</b>		
<b>Descripción:</b> Contiene los logs generados a partir de las acciones realizadas sobre el sistema.		
Nombre	Tipo	Descripción
id_log	Int4	Identificador del log generado.
fecha	Date	Fecha en que se genero el log.
hora	Date	Hora en que se genero el log.
ip	Varchar(50)	Dirección "ip" del cliente .
pc	Varchar(50)	Nombre de la "pc" cliente.
login	Varchar(50)	Identificador del dominio del usuario.
accion	Varchar(100)	Acción que realiza sobre el sistema.

### 3.3.5 Definiciones de diseño aplicadas.

La interfaz ha de ser lo más uniforme posible utilizando un mismo sistema de colores, con moderación, de forma consistente y razonable tratando de utilizar colores iguales o similares en todas las páginas, con textos concisos y claros sin mezclar muchos tipos de letras y tamaños en cada una. La navegación debe ser lo más rápida posible por lo que la presencia de imágenes innecesarias y todo lo que atente contra la rapidez y la eficiencia, debe ser eliminado.

### 3.3.6 Tratamiento de errores.

Para el tratamiento de errores en el presente sistema, se parte de la idea de que una aplicación bien diseñada debe disminuir la posibilidad de cometer errores.

En el sistema, la validación para el tratamiento de errores será referida principalmente a errores de la interacción del usuario con el sistema que son aquellos en los que se puede incurrir a falta de conocimiento o experiencia en la explotación del sistema.

Se tratará en todo momento de minimizar la posibilidad de ocurrencia de errores de este tipo aprovechando las posibilidades de la interfaz gráfica, es decir se evitará que el usuario juegue un papel más activo en la captación de información para lo cual se le dará la opción de elegir o seleccionar la información, lo cual facilitará la entrada de datos y la rapidez de la misma.

Evidentemente los errores ocurrirán incluso con los usuarios de más habilidad y experiencia, en el caso de los datos que sean adicionados por un usuario del sistema, se hace una validación de estos mediante funciones que garantizan que sean válidos, en caso que ocurra un error, se le presenta una caja de diálogo de mensaje donde se describa el error. Al obtener la confirmación de lectura del mensaje de error por parte del usuario, la caja del diálogo desaparecerá y continuará la ejecución de la aplicación.

A continuación se muestran imágenes de los diferentes mensajes de error que muestra el sistema ante acciones no válidas:

1. Al dejar campos vacíos en la página de entrada al sistema, se muestra el siguiente mensaje de error:



Figura 3.26: Campos usuario y contraseña vacíos.



- Al introducir usuarios o contraseñas no válidas.



**Figura 3.27: Campos usuario y contraseña no válidos.**

- Enviar el campo de tema vacío en el momento de insertar un tema.



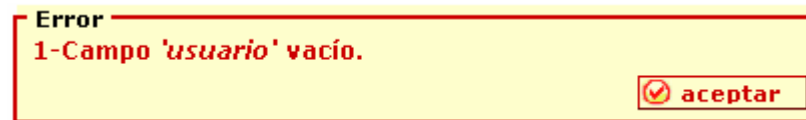
**Figura 3.28: Campo de tema vacío.**

- Enviar el campo de opinión vacío en el momento de insertar una opinión.



**Figura 3.29: Campo de opinión vacío.**

- Enviar el campo de usuario vacío en el momento de otorgar privilegios.



**Figura 3.30: Campo de usuario vacío.**

- Al intentar introducir textos entre comillas.



**Figura 3.31: Entrada de textos entre comillas.**

- Al intentar introducir etiquetas HTML.



**Figura 3.32: Entrada de etiquetas HTML.**

8. Al intentar generar estadísticas sin previa selección de las mismas.



**Figura 3.33: No selección de los campos a mostrar en las estadísticas.**

9. Al intentar generar estadísticas personalizadas sin seleccionar las opciones a graficar.



**Figura 3.34: No selección de los campos a mostrar en las estadísticas personalizadas.**

## **3.4 SEGURIDAD.**

Debido a la importancia y sensibilidad de los datos que se manejarán en la aplicación se hace necesario que se garantice la protección y seguridad de la información para el óptimo funcionamiento del sistema.

Para garantizar que determinada información presente en el sitio sólo se muestre a los usuarios registrados y con autorización previa, se trabaja con variables de sesión, de forma tal que siempre se sabe qué usuario intenta visitar dichas páginas y estas se muestran sólo para aquellos que pueden tener acceso a ellas.

De esta forma, se limitan, se controlan y se protegen, de una manera racional, los datos a los cuales se acceden desde distintos niveles de usuarios, haciendo que la seguridad de la información sea elevada.

### **3.4.1 Encriptación de contraseñas.**

Lo primero a tener en cuenta es la seguridad del servidor, además se debe tratar de evitar, el uso del típico fichero con contraseñas en la forma: usuario contraseña, como una buena práctica de programación. Las contraseñas deben permanecer encriptadas en la máquina que realiza la autenticación, siempre.

Se recomienda el uso de métodos de encriptación como MD5 para almacenar las contraseñas. Este tipo de función al igual que otras existentes son irreversibles es decir: teniendo el número generado por la función hash no existe el proceso inverso para obtener los datos que generaron ese número, por lo que es permisible que pudieran ser leídas por un intruso.

Es necesario destacar que el hecho de enviar las contraseñas encriptadas no es una solución suficiente para garantizar que no sea interceptada por un intruso y luego enviarla simplemente directo al servidor, cuando esta viaje por la red. Ante la disyuntiva de no poder evitar que los datos enviados al servidor se intercepten, utilizamos el método de encriptación de contraseñas de un solo uso OTP.

### **3.4.2 Contraseñas de un solo uso (OTP).**

Este método permite que al ser interceptados los datos de identificación, se pueda disponer de contraseñas que solo fueran válidas para un uso, por lo que los datos enviados ya habrían caducado para cuando el intruso deseara usarlos. La forma para garantizar la renovación de la contraseña, parece poco viable, ya que este proceso se efectuaría por cada acceso y la contraseña debería enviarse por un medio seguro. No obstante, la implementación del sistema de contraseñas de un solo uso permite, empleando el mismo medio inseguro, enviar las nuevas contraseñas, garantizando mayor seguridad.

Para emplear el sistema OTP se necesita definir:

- **Contraseña:** es la contraseña que domina el cliente.
- **Llave temporal:** una contraseña temporal asociada a un instante en el tiempo y a los datos disponibles del cliente.
- **Sesión:** intervalo de tiempo en el cual tiene validez una llave temporal.

Para la implantación del sistema se necesita soporte para encriptación tanto en el cliente como en el servidor. A continuación se explica cómo se utiliza el método de encriptación de contraseñas de un solo uso:

1. El cliente solicita una nueva sesión.
2. El servidor le devuelve una llave temporal (T) que solo es válida para esa sesión.
3. El cliente procesa la contraseña proporcionada por el usuario empleando (T). Ese procesamiento puede incluir desde una concatenación hasta la aplicación de los algoritmos de cifrado. Obteniendo al final del proceso una contraseña encriptada con la llave temporal, siendo este valor lo que se envía al servidor.
4. El servidor debe realizar el proceso contrario que ya ha hecho el cliente, para obtener la contraseña. Para ello procesa la contraseña encriptada enviada por el cliente, creando nuevamente la llave temporal con los datos de este, obteniendo finalmente la contraseña que se utilizará para autenticar con el servidor LDAP.

### **3.4.3 Análisis de las debilidades del algoritmo (envío de datos).**

Cuando el cliente solicita una nueva sesión, el servidor genera (T) empleando la hora del sistema o un número aleatorio, asociando la IP del cliente a ese valor. El servidor no debe guardar (T), sino el valor temporal o aleatorio. Posteriormente en el paso 4 recalculará (T) con la IP del cliente que envía los datos del paso 3. Evitando de esta manera que una máquina solicite sesión y otra distinta conteste. Realmente esto último no es infalible, aunque si proporciona un nivel aceptable de seguridad para muchos casos.

El cliente, en el paso 3, envía la contraseña encriptada por el medio inseguro. Esos datos podrían ser interceptados, pero solo tienen validez para esa sesión, por lo que resultaría completamente inútil para un intruso. Como se puede observar, realmente se emplea una contraseña cada vez, pero de forma automática y transparente para el usuario y el administrador.

### **3.4.4 Seguridad del sistema.**

Para lograr una mayor seguridad e integridad de la información que se trasmite por la red entre el cliente y el servidor, además de las configuraciones del PHP que intervienen en la seguridad de un sitio Web, en la programación se evitó el pase de parámetros confidenciales a través del método "GET" usando la mayor parte del tiempo el método "POST", con métodos de codificación y encriptación en los casos que fuese necesario. Durante el proceso de acceso al sitio es necesario que los usuarios llenen los formularios con el nombre de usuario y contraseña para ser autenticados por el servidor LDAP, estos datos serán encriptados y enviados luego al servidor.

## **3.5 CONCLUSIONES.**

En este capítulo se abordó el análisis y el diseño del sistema a implementar, fue desarrollado el modelo de clases del análisis y se confeccionaron los diagramas de interacción como evolución de la expansión de los casos de uso. Se obtuvo el diagrama de clases del diseño de la aplicación y se describieron las clases teniendo en cuenta sus atributos y responsabilidades, como concepción a las particularidades de la aplicación a implementar y a la tecnología seleccionada, se desarrollaron los diagramas de diseño Web y se realizó la descripción de las páginas en función de sus roles específicos. Se obtuvo el diagrama entidad relación y se realizó la descripción de las tablas teniendo en cuenta la información contenida y sus campos. Además se realizaron algunas consideraciones acerca del tratamiento de errores y el manejo de excepciones con el fin de lograr una aplicación robusta.

# Capítulo 4

## Implementación y Prueba

### 4.1 INTRODUCCIÓN.

En el presente capítulo se desarrollarán los diagramas de despliegue y componentes que conforman lo que se conoce como modelo de implementación al describir los componentes a construir, su organización y dependencia entre nodos físicos en los que funcionará la aplicación. Además se abordarán los temas relacionados con las pruebas aplicadas al software. Estas son un elemento crítico para garantizar la calidad de los sistemas y representan una revisión final de las especificaciones del diseño y la codificación. Existen diferentes niveles de pruebas para diferentes tipos de objetivos y diferentes escenarios, distinguiéndose: la prueba de sistema y de aceptación. Se aplicaron varios tipos de prueba basadas en dimensiones de calidad y el método de prueba utilizado fue el de caja negra.

### 4.2 DIAGRAMA DE DESPLIEGUE.

El diagrama de despliegue permite apreciar de forma visual como se encuentran relacionados físicamente los componentes de la aplicación. En este caso la aplicación se encuentra hospedada en un servidor Web, la misma se comunica con un sistema de gestión de base de datos (Postgre) ubicado en otro servidor.

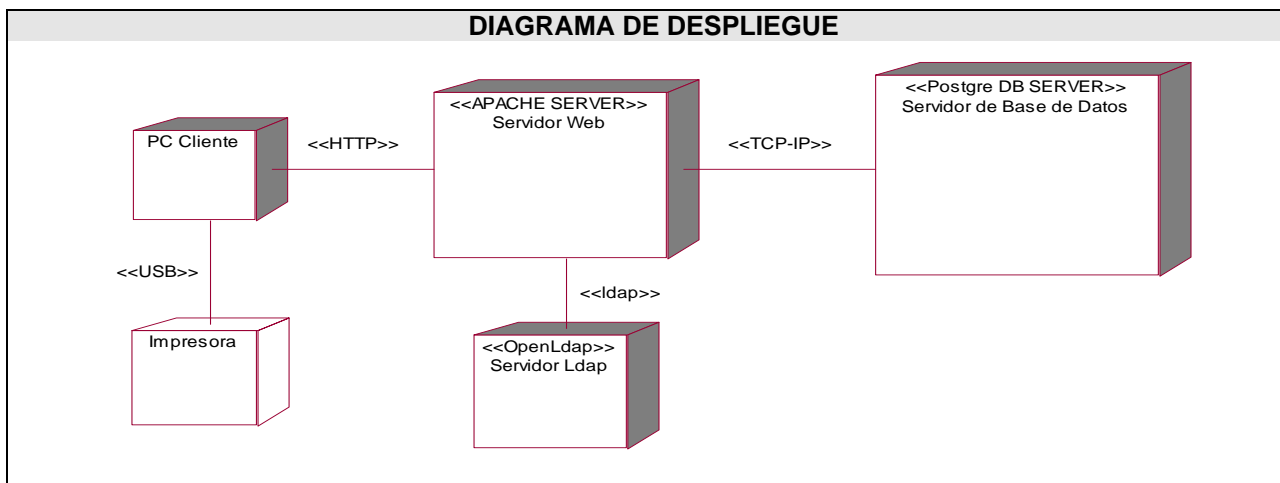


Figura 4.1: Diagrama de Despliegue.

### 4.3 DIAGRAMA DE COMPONENTES.

Los diagramas de componentes contribuyen a un mejor entendimiento del modelo de implementación. Con ellos se representan los componentes lógicos de la aplicación así como las relaciones de dependencia que existen entre ellos.

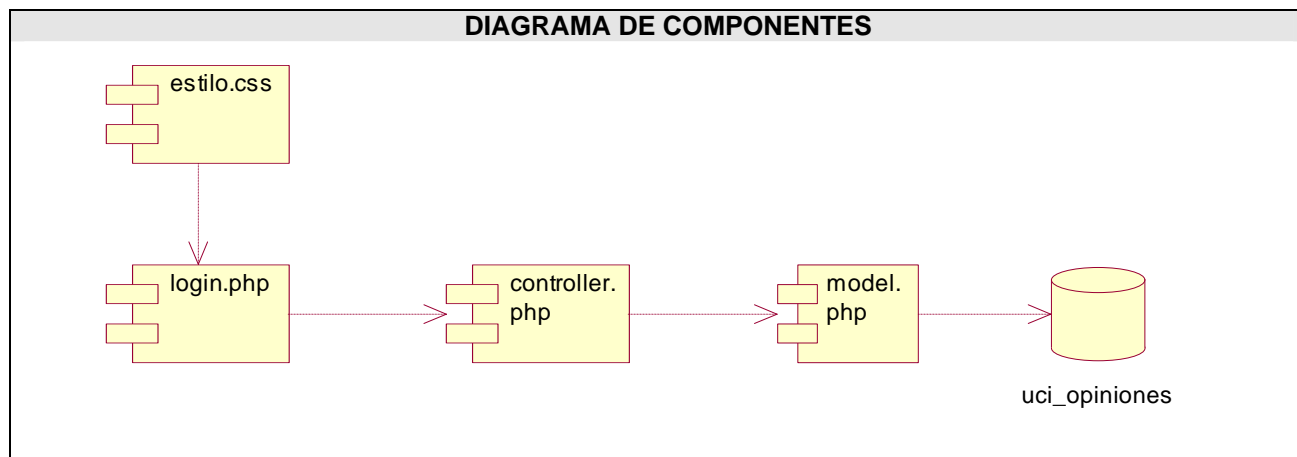


Figura 4.2: Diagrama de Componentes CU Autenticarse.

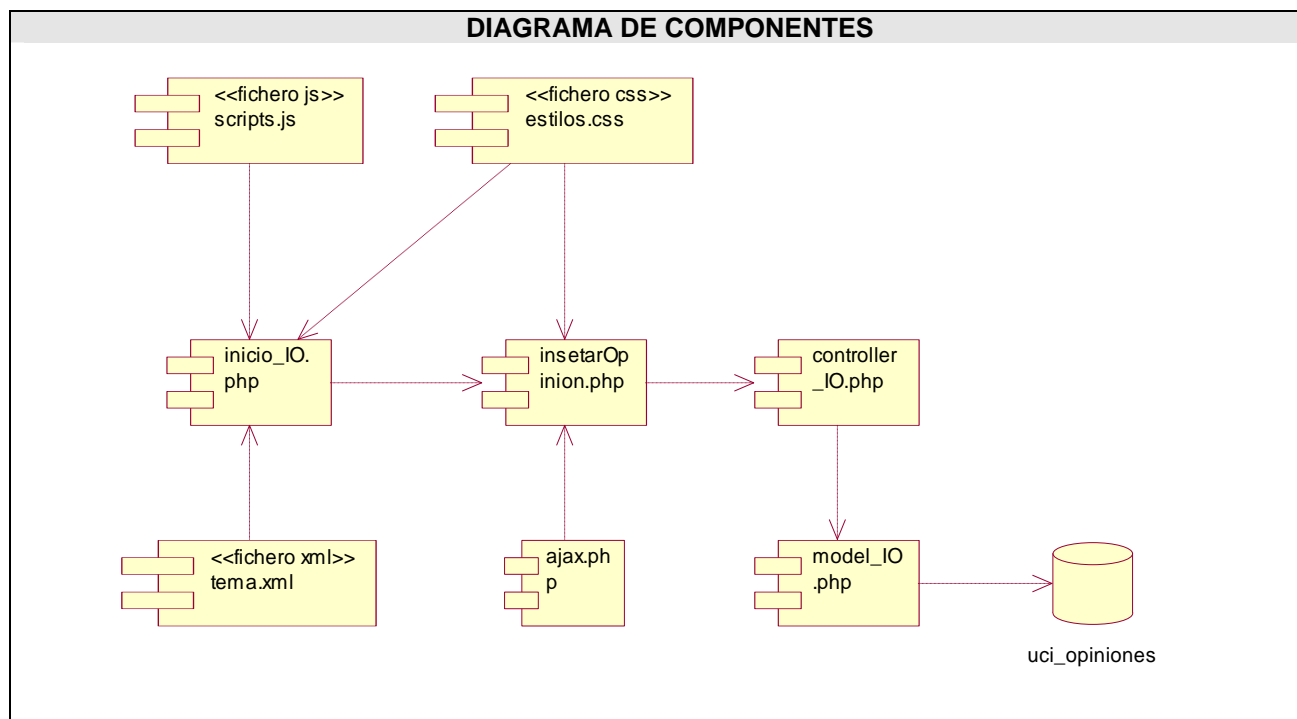
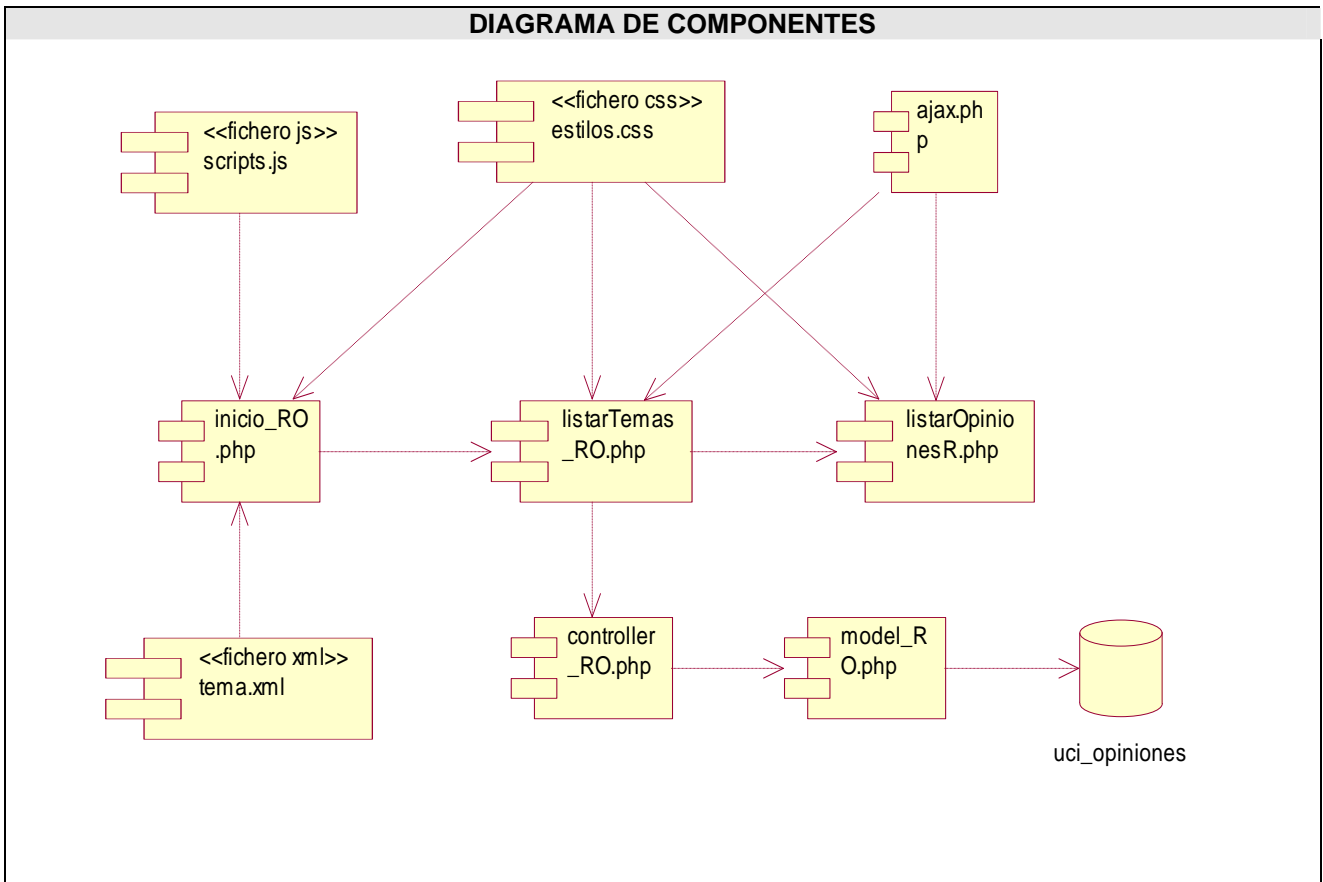
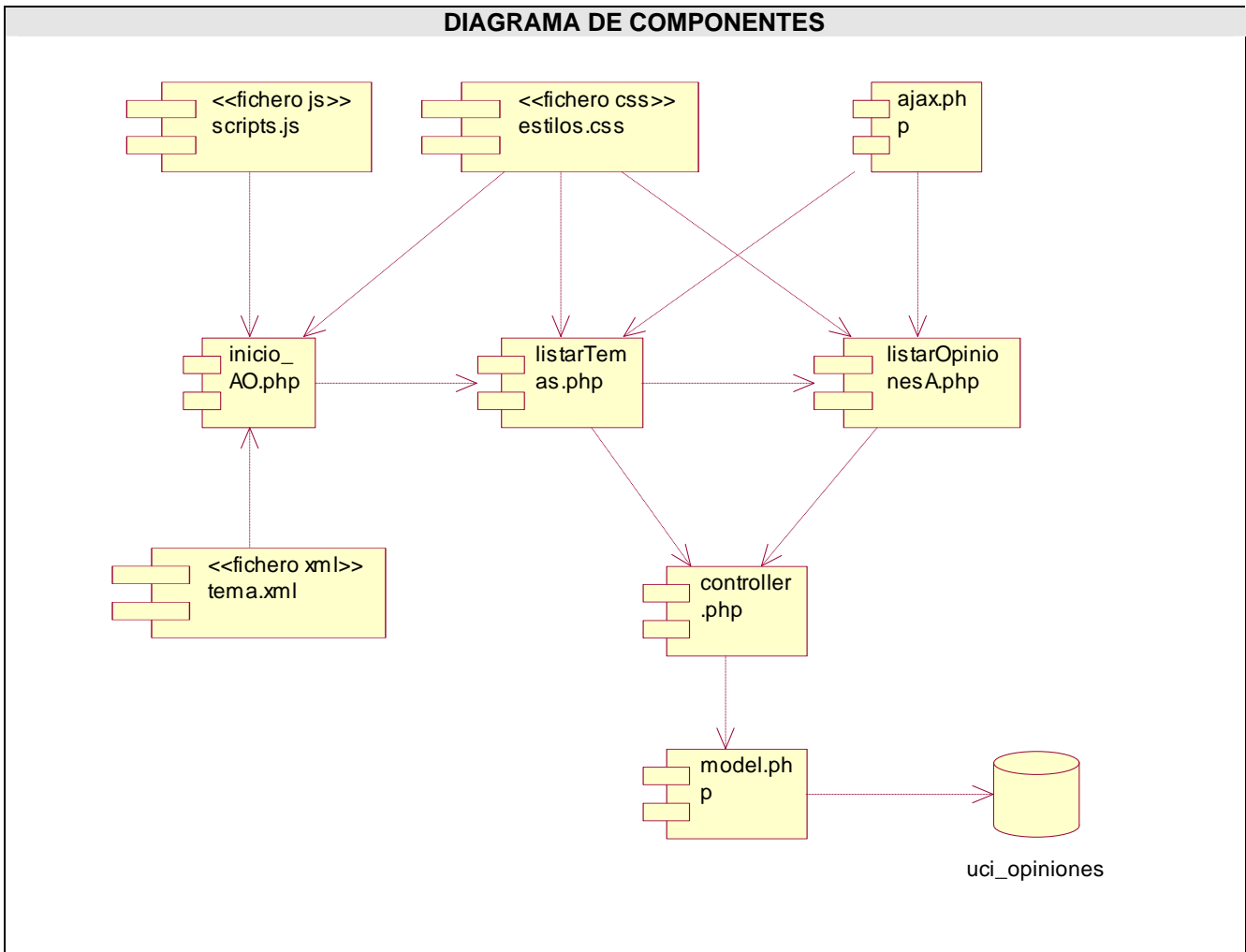


Figura 4.3: Diagrama de Componentes CU Insertar Opinión.

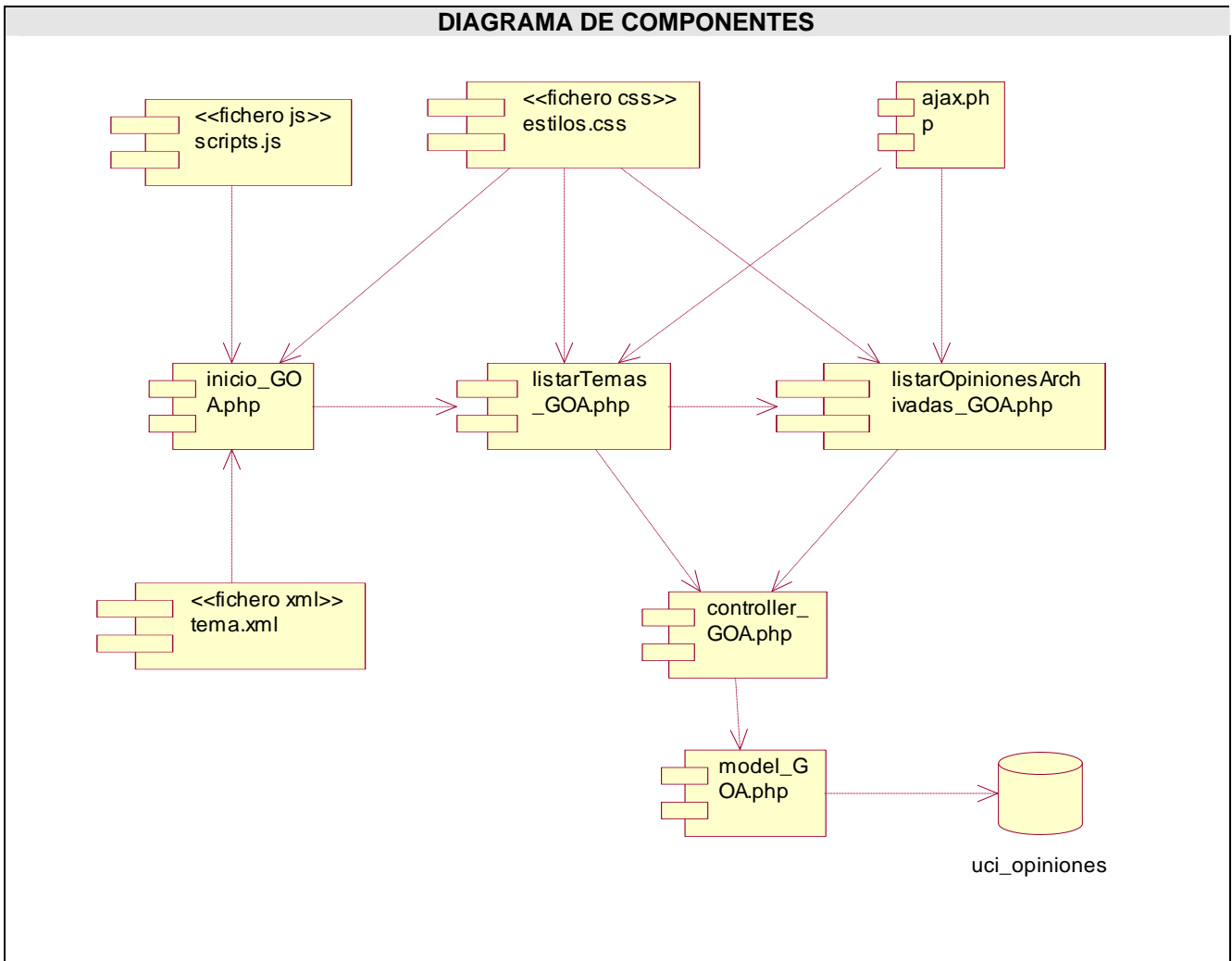


**Figura 4.4: Diagrama de Componentes CU Revisar Opinión.**

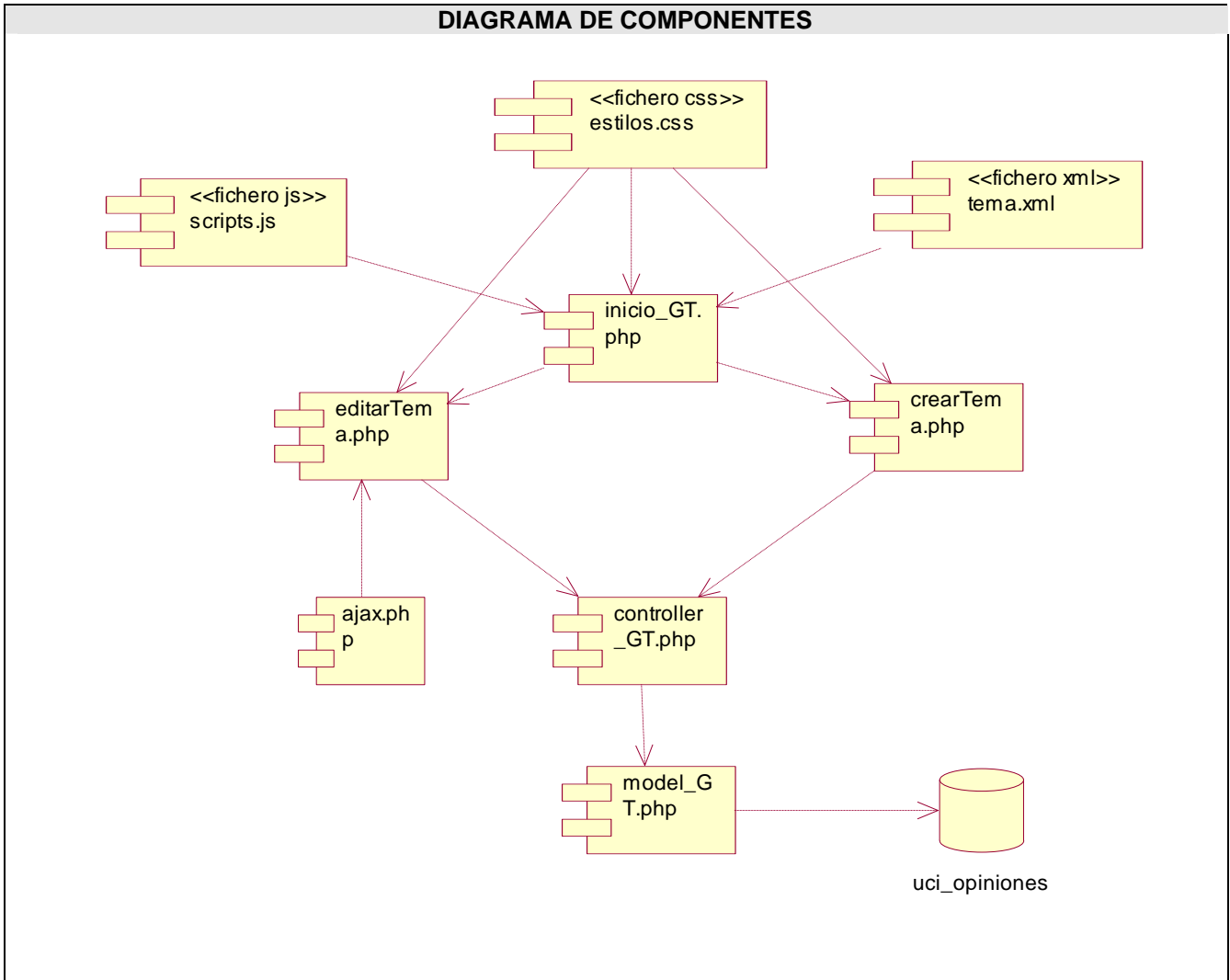


**Figura 4.5: Diagrama de Componentes CU Aprobar Opinión.**

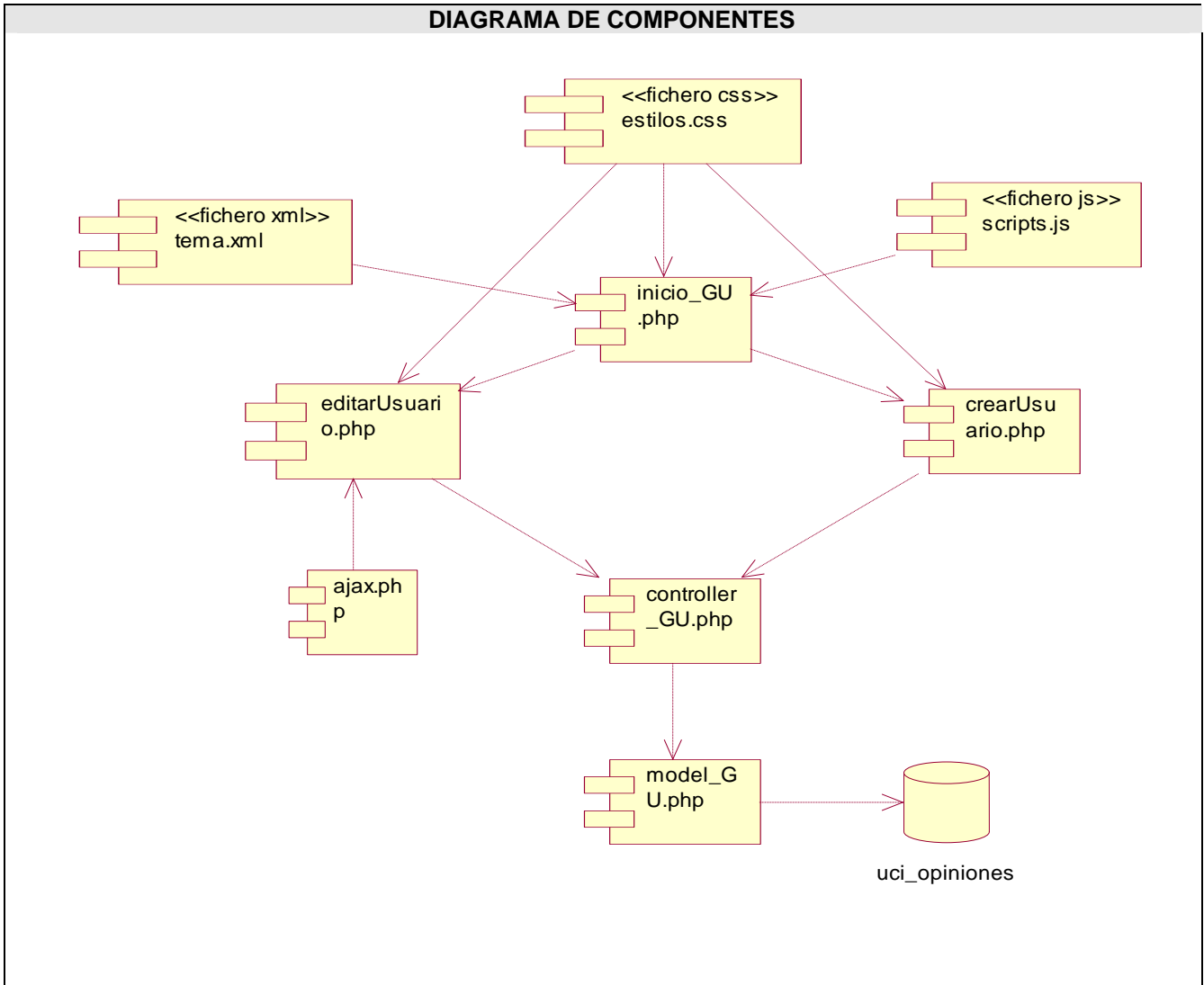




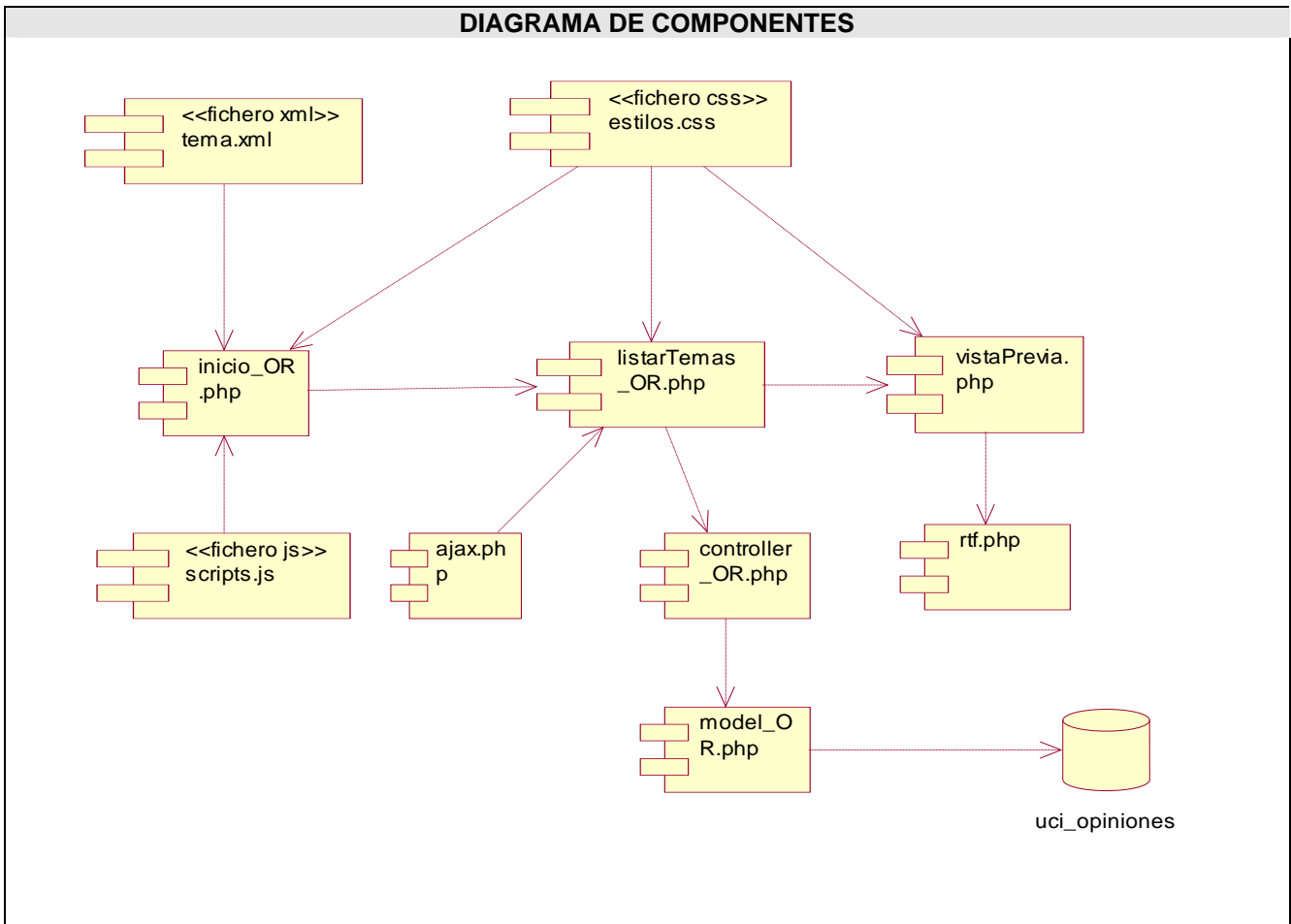
**Figura 4.6: Diagrama de Componentes CU Gestionar Opiniones Archivadas.**



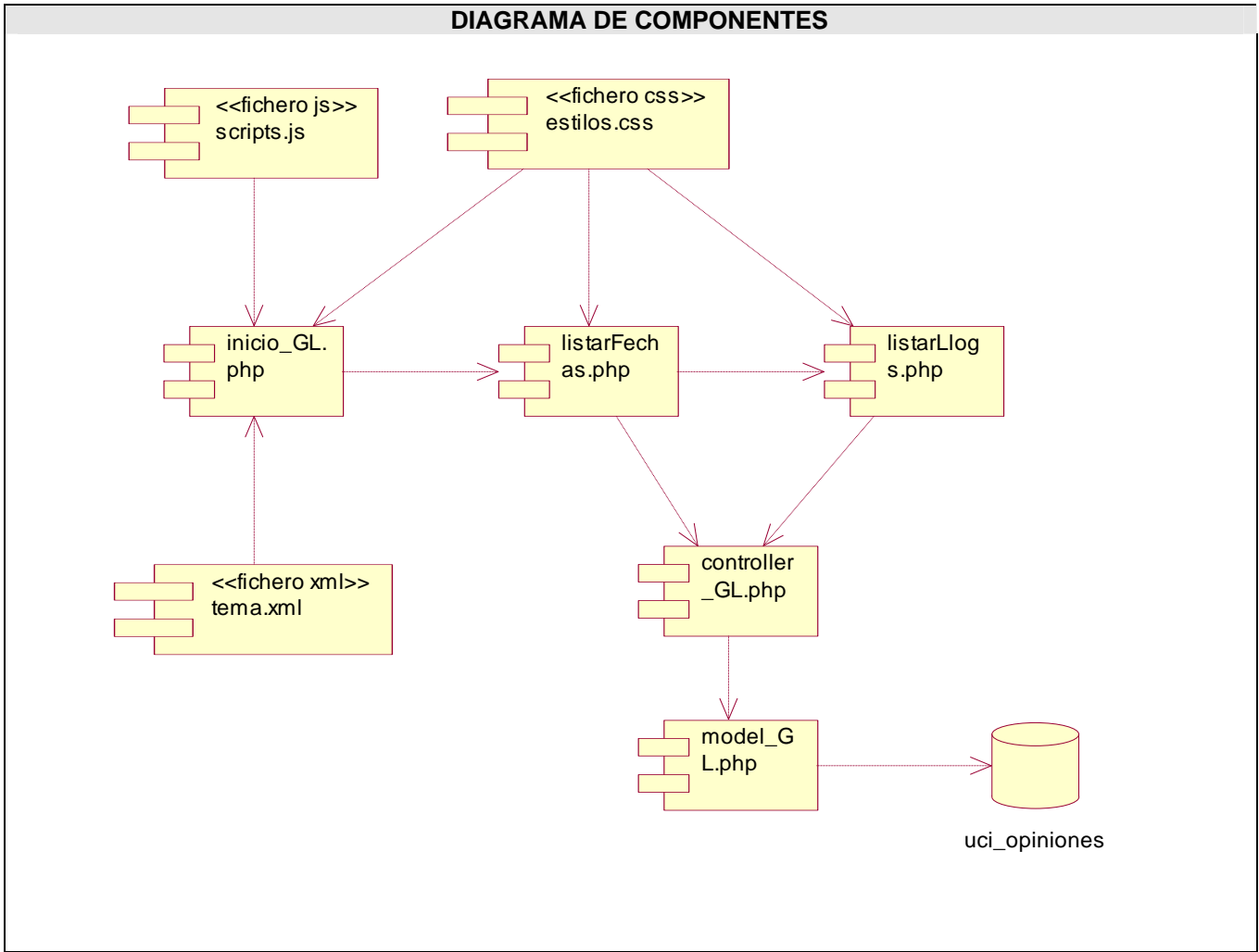
**Figura 4.7: Diagrama de Componentes CU Gestionar Tema.**



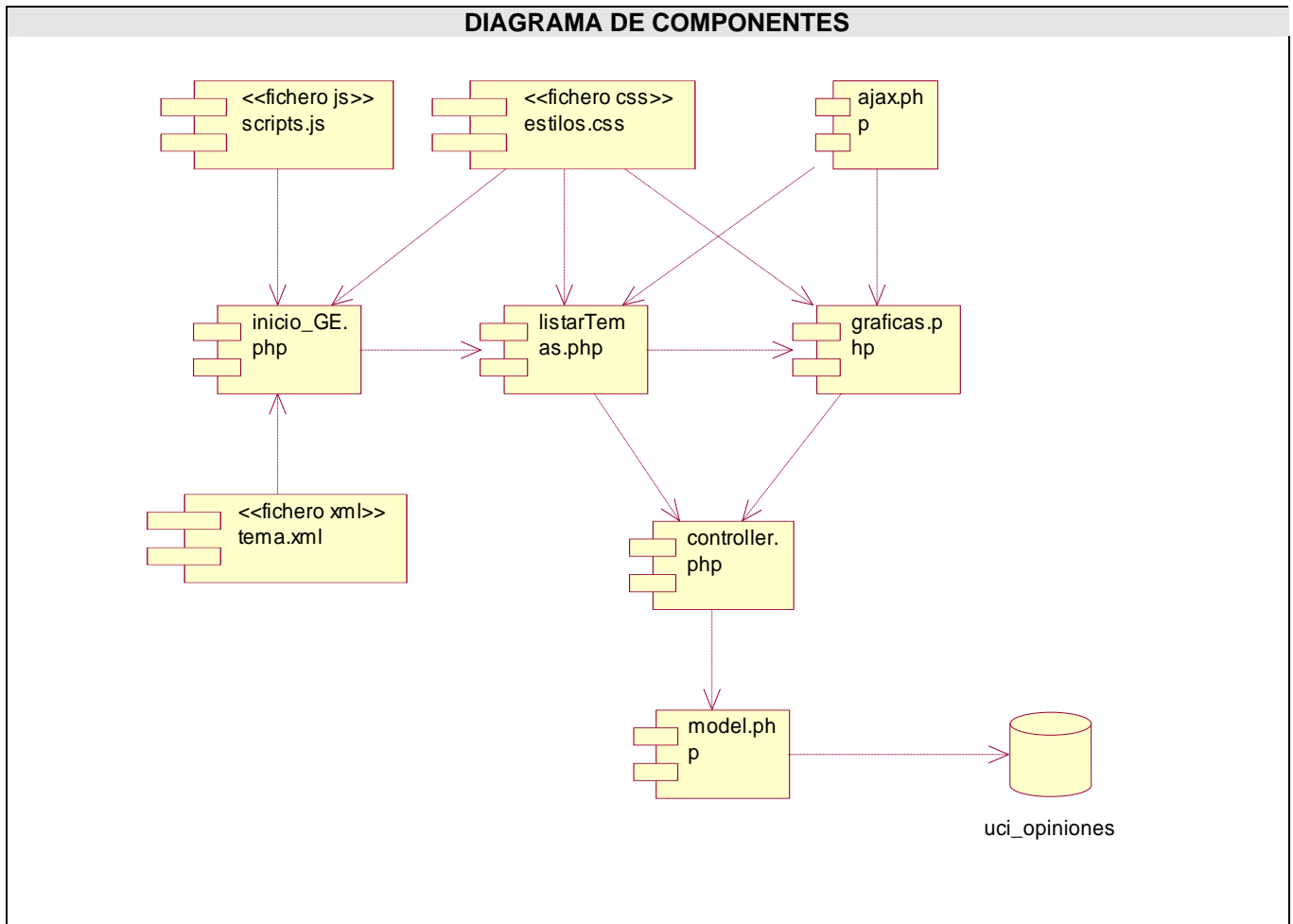
**Figura 4.8: Diagrama de Componentes CU Gestionar Usuario.**



**Figura 4.9: Diagrama de Componentes CU Obtener Reporte.**



**Figura 4.10: Diagrama de Componentes CU Gestionar Logs.**



**Figura 4.11: Diagrama de Componentes CU Graficar Estadísticas.**

## 4.4 MODELO DE PRUEBA.

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que aparezca la falibilidad humana son comunes. Los errores pueden empezar a darse desde el primer momento del proceso en el que los objetivos pueden estar especificados de forma errónea e imperfecta; así en los posteriores pasos del diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad.

*Las pruebas* es una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

#### 4.4.1 Niveles de Prueba.

Las pruebas son aplicadas para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. En nuestro trabajo se distinguen los siguientes niveles de pruebas:

**Prueba de Sistema:** Son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados.

##### Tipos de Pruebas del Sistema

- **Prueba de Recuperación:** Es una prueba del sistema que fuerza el fallo del software de muchas formas y verifica que la recuperación se lleva a cabo apropiadamente.
- **Prueba de Seguridad:** Intenta verificar que los mecanismos de protección incorporados en el sistema lo protegerán, de hecho, de accesos impropios.
- **Prueba de Resistencia:** Está diseñada para enfrentar a los programas con situaciones anormales.
- **Prueba de Rendimiento:** Está diseñada para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado.

**Prueba de aceptación:** Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

#### 4.4.2 Tipos de Prueba.

Cada tipo de prueba tiene un objetivo específico y una técnica que lo soporte. A continuación se muestran los tipos de pruebas basados en dimensiones de calidad que fueron aplicadas:

**Dimensión de calidad/ Riesgo de calidad:** Funcionalidad

**Función:** Pruebas que fijan su atención en la validación de las funciones, métodos, servicios, caso de uso.

**Seguridad:** Centrada en asegurar que los datos o el sistema solamente es accedido por los actores deseados.

**Volumen:** Enfocada en verificar las habilidades de los programas para manejar grandes cantidades de datos, tanto como entrada, salida o residente en la BD.

**Dimensión de calidad/ Riesgo de calidad:** Usabilidad

**Usabilidad:** Prueba enfocada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto y en línea, asistente documentación de usuarios y materiales de entrenamiento.

**Dimensión de calidad/ Riesgo de calidad:** Fiabilidad

**Integridad:** Enfocada a la valoración de la robustez (resistencia a fallos).

**Estructura:** Enfocada a la adherencia, a su diseño y formación. Este tipo de prueba es hecha a las aplicaciones Web asegurando que todos los enlaces están conectados, el contenido deseado es mostrado y no hay contenido huérfano.

**Stress:** Enfocada a evaluar cómo el sistema responde bajo condiciones anormales. (Extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponibles).

**Dimensión de calidad/ Riesgo de calidad:** Performance (Rendimiento)

**Benchmark:** Es un tipo de prueba que compara el rendimiento de un elemento nuevo o desconocido a uno de carga de trabajo de referencia conocido.

**Contención:** Enfocada a la validación de las habilidades del elemento a probar para manejar aceptablemente la demanda de múltiples actores sobre un mismo recurso (registro de recursos, memoria, etc.).

**Carga:** Usada para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga es simular la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales.

**Performance profile:** Enfocadas a monitorear el tiempo en flujo de ejecución, acceso a datos, en llamada a funciones y sistema para identificar y direccionar los cuellos de botellas y procesos ineficientes.

**Dimensión de calidad/ Riesgo de calidad:** Soportabilidad

**Configuración:** Enfocada a asegurar que funciona en diferentes configuraciones de hardware y software. Esta prueba es implementada también como prueba de rendimiento del sistema.

**Instalación:** Enfocada a asegurar la instalación en diferentes configuraciones de hardware y software bajo diferentes condiciones (insuficiente espacio en disco, etc.).

### 4.4.3 Métodos de Prueba.

Los métodos de pruebas fundamentales son: el método de caja negra y de caja blanca. En este trabajo de diploma se utiliza el de caja negra.



La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

#### **Pruebas de caja negra:**

- Verifican las especificaciones funcionales y no consideran la estructura interna del programa.
- Es hecha sin el conocimiento interno del producto.
- No validan funciones ocultas (por ejemplo funciones implementadas pero no descritas en las especificaciones funcionales del diseño) por tanto los errores asociados a ellas no serán encontrados.

Los casos de prueba que se presentan a continuación pretenden demostrar que las funciones del software son operativas, que las entradas se aceptan de la forma adecuada y que se produce el resultado correcto.

### **4.4.4 Caso de Uso Autenticar Usuario.**

#### **Descripción General**

Este caso de uso consiste en validar los usuarios existentes y asignarle los roles correspondientes.

A este Caso de Uso se le realizaron las siguientes pruebas:

1. Autenticar Usuarios

#### **Caso de Prueba 1: Autenticar Usuarios**

Descripción de la Funcionalidad:

Esta funcionalidad permite validar los usuarios existentes en el directorio OpenLdap para otorgarle la entrada al sistema y mostrarle la interfaz correspondiente a su rol.

##### 1.1 Flujo Central:

- El sistema muestra la interfaz “Entrada al Sistema”.
- El usuario introduce los siguientes datos:
  - Usuario
  - Clave
- El sistema muestra la interfaz correspondiente al rol del usuario registrado.

##### 1.2 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<b>Usuario:</b> marcialpt <b>Contraseña:</b> válida		El sistema muestra la interfaz correspondiente al "Usuario".	Satisfactorio	
<b>Usuario:</b> yady <b>Contraseña:</b> válida		El sistema muestra la interfaz correspondiente al "Administrador".	Satisfactorio	
<b>Usuario:</b> lsanchez <b>Contraseña:</b> válida		El sistema muestra la interfaz correspondiente para el "Editor".	Satisfactorio	
<b>Usuario:</b> dsanchez <b>Contraseña:</b> válida		El sistema muestra la interfaz correspondiente para el "Aprobador".	Satisfactorio	
	<b>Usuario:</b> <b>Contraseña:</b>	El sistema muestra un mensaje informando que debe llenar todos los campos.	Satisfactorio	
	<b>Usuario:</b> 123 <b>Contraseña:</b> 123	El sistema muestra un mensaje informando que el usuario no es válido.	Satisfactorio.	

#### 4.4.5 Caso de Uso Gestionar Tema.

##### Descripción General

Este caso de uso consiste en crear temas para opinar además de editarlos, actualizarlos o eliminarlos.

A este Caso de Uso se le realizaron las siguientes pruebas:

1. Crear Tema.
2. Actualizar Tema.
3. Eliminar Tema.

##### Caso de Prueba 1: Crear Tema

###### 1.1 Descripción de la Funcionalidad:

Esta funcionalidad permite crear los temas de debate para que los usuarios puedan luego emitir sus opiniones.

###### 1.2 Flujo Central:

- El sistema muestra la interfaz para "Crear Tema".
- El usuario introduce los siguientes datos:

- Tema
- Nota
- El sistema muestra un mensaje de confirmación señalando que el tema ha sido creado.

### 1.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<b>Tema:</b> Un Tema <b>Nota:</b> Nota del tema.		El sistema muestra un mensaje señalando que el tema ha sido creado.	Satisfactorio	
<b>Tema:</b> Un Tema <b>Nota:</b>		El sistema muestra un mensaje señalando que el tema ha sido creado.	Satisfactorio	
	<b>Tema:</b> <b>Nota:</b>	El sistema muestra un mensaje de error señalando que el campo tema se encuentra vacío.	Satisfactorio	
	<b>Tema:</b> 'un tema' <b>Nota:</b> 'nota'	El sistema muestra un mensaje de error señalando que no está permitido el uso de comillas.	Satisfactorio	
	<b>Tema:</b> <script></script> > <b>Nota:</b> nota	El sistema muestra un mensaje de error señalando que las etiquetas que se están tratando de utilizar no están permitidas.	Satisfactorio	

### Caso de Prueba 2: Actualizar Tema

#### 2.1 Descripción de la Funcionalidad:

Esta funcionalidad permite actualizar los temas de debate para realizarles alguna modificación.

#### 2.2 Flujo Central:

- El sistema muestra la interfaz "Editar Tema".
- El usuario selecciona la opción "Editar":
- El sistema muestra los datos del tema de forma editable.
- El usuario modifica los datos del tema y selecciona la opción "Actualizar".

#### 2.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<b>Tema:</b> Un Tema Nuevo <b>Nota:</b> Nota del tema.		El sistema muestra el tema con los datos modificados.	Satisfactorio	

<b>Estado:</b> activo				
<b>Tema:</b> Un Tema <b>Nota:</b> <b>Estado:</b> inactivo		El sistema muestra el tema con los datos modificados.	Satisfactorio	
	<b>Tema:</b> <b>Nota:</b> <b>Estado:</b> inactivo	El sistema muestra un mensaje de error señalando que el campo tema se encuentra vacío.	Satisfactorio	
	<b>Tema:</b> 'un tema' <b>Nota:</b> 'nota' <b>Estado:</b> inactivo	El sistema muestra un mensaje de error señalando que no está permitido el uso de comillas.	Satisfactorio	
	<b>Tema:</b> <script></script> > <b>Nota:</b> nota <b>Estado:</b> inactivo	El sistema muestra un mensaje de error señalando que las etiquetas que se están tratando de utilizar no están permitidas.	Satisfactorio	

### Caso de Prueba 3: Eliminar Tema

#### 3.1 Descripción de la Funcionalidad:

Esta funcionalidad permite eliminar un tema de debate.

#### 3.2 Flujo Central:

- El sistema muestra la interfaz "Editar Tema".
- El usuario selecciona la opción "Eliminar":
- El sistema muestra un mensaje preguntando al usuario si realmente desea eliminar el tema.
- El usuario selecciona la opción "OK".
- El sistema muestra un mensaje de confirmación señalando que el tema ha sido eliminado.

#### 3.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona la opción "Eliminar"		El sistema elimina el tema seleccionado y muestra un mensaje de confirmación señalando que el tema ha sido eliminado.	Satisfactorio	

## 4.4.6 Caso de Uso Insertar Opinión.

### Descripción General

Este permite a los usuarios insertar opiniones relacionado con un tema de debate.

A este Caso de Uso se le realizaron las siguientes pruebas:

1. Insertar Opinión.

### Caso de Prueba 1: Insertar Opinión

#### 1.1 Descripción de la Funcionalidad:

Esta funcionalidad permite insertar opiniones relacionadas con un tema de debate.

#### 1.2 Flujo Central:

- El sistema muestra la interfaz “Insertar Opinión”.
- El usuario introduce los siguientes datos:
  - Opinión.
- El sistema muestra un mensaje señalando que la opinión ha sido insertada.

#### 1.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<b>Opinión:</b> Opinión Insertada		El sistema muestra un mensaje señalando que la opinión ha sido enviada.	Satisfactorio	
	<b>Opinión:</b> 'Opinión Insertada'	El sistema muestra un mensaje de error señalando que las comillas no están permitidas.	Satisfactorio	
	<b>Opinión:</b>	El sistema muestra un mensaje de error señalando que el campo opinión está vacío.	Satisfactorio	
	<b>Opinión:</b> <script>	El sistema muestra un mensaje de error señalando que las etiquetas utilizadas no están permitidas.	Satisfactorio	

## 4.4.7 Caso de Uso Revisar Opinión.

### Descripción General

Este permite al editor revisar las opiniones emitidas por los usuarios relacionadas con un tema de debate.

A este Caso de Uso se le realizaron las siguientes pruebas:

1. Editar Opinión.

### Caso de Prueba 1: Editar Opinión

### 1.1 Descripción de la Funcionalidad:

Esta funcionalidad permite editar y actualizar las opiniones emitidas por los usuarios.

### 1.2 Flujo Central:

- El editor selecciona la opción “Revisar Opinión” del menú “Editar”.
- El sistema muestra la interfaz “Revisar Opinión”.
- El editor selecciona la opción “Listar” del tema seleccionado.
- El sistema muestra las opiniones emitidas una por una.
- El editor selecciona la opción “Editar” y actualiza la opinión.

### 1.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Opinión: Opinión Editada		El sistema actualiza la opinión.	Satisfactorio	
	Opinión: 'Opinión Insertada'	El sistema muestra un mensaje de error señalando que las comillas no están permitidas.	Satisfactorio	
	Opinión:	El sistema muestra un mensaje de error señalando que el campo opinión está vacío.	Satisfactorio	
	Opinión:<script></script>	El sistema muestra un mensaje de error señalando que las etiquetas utilizadas no están permitidas.	Satisfactorio	

## 4.4.8 Caso de Uso Aprobar Opinión.

### Descripción General

Este permite al revisor aprobar las opiniones emitidas por los usuarios relacionadas con un tema de debate.

A este Caso de Uso se le realizaron las siguientes pruebas:

1. Editar Opinión.

### Caso de Prueba 1: Editar Opinión

#### 1.1 Descripción de la Funcionalidad:

Esta funcionalidad permite editar y actualizar las opiniones revisadas por los usuarios.

#### 1.2 Flujo Central:

- El revisor selecciona la opción “Aprobar Opinión” del menú “Aprobar”.
- El sistema muestra la interfaz “Aprobar Opinión”.
- El revisor selecciona la opción “Listar” del tema seleccionado.

- El sistema muestra las opiniones revisadas una por una.
- El revisor selecciona la opción “Editar” y actualiza la opinión.

### 1.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<b>Opinión:</b> Opinión Editada		El sistema actualiza la opinión.	Satisfactorio	
	<b>Opinión:</b> 'Opinión Insertada'	El sistema muestra un mensaje de error señalando que las comillas no están permitidas.	Satisfactorio	
	<b>Opinión:</b>	El sistema muestra un mensaje de error señalando que el campo opinión está vacío.	Satisfactorio	
	<b>Opinión:</b> <script></script>	El sistema muestra un mensaje de error señalando que las etiquetas utilizadas no están permitidas.	Satisfactorio	

## 4.4.9 Caso de Uso Gestionar Opiniones Archivadas.

### Descripción General

Este permite al administrador aprobar o eliminar las opiniones archivadas durante el proceso de revisión o aprobación de opiniones.

A este Caso de Uso se le realizaron las siguientes pruebas:

1. Editar Opinión.
2. Eliminar Opinión.

### Caso de Prueba 1: Editar Opinión

#### 1.1 Descripción de la Funcionalidad:

Esta funcionalidad permite editar y actualizar las opiniones archivadas.

#### 1.2 Flujo Central:

- El administrador selecciona la opción “Opiniones Archivadas” del menú “Administrar”.
- El sistema muestra la interfaz “Opiniones Archivadas”.
- El administrador selecciona la opción “Listar” del tema seleccionado.
- El sistema muestra las opiniones archivadas una por una.
- El administrador selecciona la opción “Editar” y actualiza la opinión.

#### 1.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<b>Opinión:</b> Opinión Editada		El sistema actualiza la opinión.	Satisfactorio	
	<b>Opinión:</b> 'Opinión Insertada'	El sistema muestra un mensaje de error señalando que las comillas no están permitidas.	Satisfactorio	
	<b>Opinión:</b>	El sistema muestra un mensaje de error señalando que el campo opinión está vacío.	Satisfactorio	
	<b>Opinión:</b> <script>></script>	El sistema muestra un mensaje de error señalando que las etiquetas utilizadas no están permitidas.	Satisfactorio	

## Caso de Prueba 2: Eliminar Opinión

### 1.1 Descripción de la Funcionalidad:

Esta funcionalidad permite eliminar una opinión archivada.

### 1.2 Flujo Central:

- El administrador selecciona la opción "Opiniones Archivadas" del menú "Administrar".
- El sistema muestra la interfaz "Opiniones Archivadas".
- El administrador selecciona la opción listar del tema seleccionado.
- El sistema muestra las opiniones archivadas una por una.
- El administrador selecciona la opción "Eliminar".
- El sistema muestra un mensaje preguntando si realmente desea eliminar la opinión.
- El administrador selecciona la opción "OK".
- El sistema muestra un mensaje de confirmación señalando que la opinión ha sido eliminada.

### 1.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selección de la opción "Eliminar"		El sistema pregunta si desea eliminar la opinión y la elimina en caso de seleccionar "OK".	Satisfactorio	



## 4.4.10 Caso de Uso Gestionar Usuario.

### Descripción General

Este caso de uso consiste en crear usuarios que tendrán diferentes roles para el trabajo con el Sistema de Opiniones, además de editarlos o eliminarlos.

A este Caso de Uso se le realizaron las siguientes pruebas:

1. Crear Usuario.
2. Actualizar Usuario.
3. Eliminar Usuario.

### Caso de Prueba 1: Crear Usuario

#### 1.1 Descripción de la Funcionalidad:

Esta funcionalidad permite crear un usuario que se le asigna un rol específico para el trabajo con la aplicación.

#### 1.2 Flujo Central:

- El administrador selecciona la opción “Crear Usuario” del menú “Administrar”.
- El sistema muestra la interfaz “Crear Usuario”.
- El administrador introduce los siguientes datos:
  - Usuario
- El administrador selecciona la opción “Buscar”.
- El sistema muestra datos del usuario.
- El administrador selecciona la opción “Crear”.
- El sistema muestra un mensaje de confirmación señalando que el usuario ha sido creado.

#### 1.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Usuario: leo		El sistema muestra un mensaje señalando que el usuario ha sido creado.	Satisfactorio	
	Usuario:	El sistema muestra un mensaje señalando que el campo usuario se encuentra vacío.	Satisfactorio	
	Usuario: 'leo'	El sistema muestra un mensaje de error señalando que el campo usuario no es válido.	Satisfactorio	
	Usuario: <script></script>	El sistema muestra un mensaje de error señalando que las etiquetas	Satisfactorio	

	>	que se están tratando de utilizar no están permitidas.		
--	---	--	--	--

## Caso de Prueba 2: Actualizar Usuario

### 2.1 Descripción de la Funcionalidad:

Esta funcionalidad permite actualizar el rol del usuario.

### 2.2 Flujo Central:

- El administrador selecciona la opción “Editar Usuario” del menú “Administrar”.
- El sistema muestra la interfaz “Editar Usuario”.
- El administrador selecciona la opción “Editar”:
- El sistema muestra el rol del usuario de manera que se pueda seleccionar otro rol para asignarle.
- El administrador modifica el rol y selecciona la opción “Guardar”.
- El sistema actualiza y vuelve a mostrar los datos del usuario.

### 2.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Permiso: administrar		El sistema vuelve a mostrar al usuario con los datos modificados.	Satisfactorio	

## Caso de Prueba 3: Eliminar Usuario

### 3.1 Descripción de la Funcionalidad:

Esta funcionalidad permite eliminar un usuario con roles específicos.

### 3.2 Flujo Central:

- El administrador selecciona la opción “Editar Usuario” del menú “Administrar”.
- El sistema muestra la interfaz “Editar Usuario”.
- El administrador selecciona la opción “Eliminar”:
- El sistema muestra un mensaje preguntando si realmente desea eliminar el elemento seleccionado.
- El administrador selecciona la opción “OK”.
- El sistema elimina el usuario y muestra un mensaje de confirmación señalando que el usuario ha sido eliminado.

### 3.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona la opción "Eliminar"		El sistema elimina el usuario seleccionado y muestra un mensaje de confirmación señalando que el usuario ha sido eliminado.	Satisfactorio	

#### 4.4.11 Caso de Uso Gestionar Reportes.

##### Descripción General

Este caso de uso permite al administrador crear reportes con las opiniones, y además reportes estadísticos.

A este Caso de Uso se le realizaron las siguientes pruebas:

1. Crear Reporte Estadístico.

##### Caso de Prueba 1: Crear Reporte Estadístico

###### 1.1 Descripción de la Funcionalidad:

Esta funcionalidad permite obtener reportes estadísticos del tema seleccionado.

###### 1.2 Flujo Central:

- El administrador selecciona la opción "Estadística" del menú "Administrar".
- El sistema muestra la interfaz "Crear Reporte Estadísticos".
- El administrador selecciona las estadísticas del tema seleccionado.
- El administrador selecciona la opción "Vista Previa".
- El sistema muestra en la interfaz "Vista Previa", una vista del reporte.
- El administrador selecciona la opción "Crear Documento".
- El sistema crea un documento en formato "rtf" con el reporte previamente generado.

###### 1.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona las estadísticas.		El sistema muestra en una interfaz la vista previa del reporte.	Satisfactorio	
	No selecciona ninguna estadística	El sistema muestra un mensaje señalando que debe seleccionar al menos una estadística para generar.	Satisfactorio	

#### 4.4.12 Caso de Uso Gestionar Logs.

##### Descripción General

Este caso de uso permite al administrador gestionar los logs generados por el sistema.

A este Caso de Uso se le realizaron las siguientes pruebas:

1 Eliminar Logs.

##### Caso de Prueba 1: Eliminar Logs

1.1 Descripción de la Funcionalidad:

Esta funcionalidad permite eliminar las estadísticas generadas por el sistema.

1.2 Flujo Central:

- El administrador selecciona la opción “Logs” del menú “Administrar”.
- El sistema muestra la interfaz “Logs”.
- El administrador selecciona la opción “Eliminar” del día seleccionado.
- El sistema muestra un mensaje preguntando si realmente desea eliminar el tema seleccionado.
- El administrador selecciona la opción “OK”.
- El sistema muestra un mensaje señalando que los Logs del día seleccionado han sido eliminados y vuelve a listar las fechas.

1.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona la opción “Eliminar”.		El sistema elimina los logs del día seleccionado y vuelve a listar las fechas.	Satisfactorio	

#### 4.4.13 Caso de Uso Gestionar Estadísticas.

##### Descripción General

Este caso de uso permite al administrador graficar estadísticas relacionadas con las opiniones emitidas de un tema específico.

A este Caso de Uso se le realizaron las siguientes pruebas:

1 Personalizar Estadísticas.

##### Caso de Prueba 1: Personalizar Estadísticas

1.1 Descripción de la Funcionalidad:

Esta funcionalidad permite al administrador personalizar las estadísticas para obtener sus gráficas.

#### 1.2 Flujo Central:

- El administrador selecciona la opción “Graficar Estadísticas” del menú “Estadísticas”.
- El sistema muestra la interfaz “Crear Reporte Gráfico”.
- El administrador selecciona la opción “Personalizar Estadística” del tema seleccionado.
- El sistema muestra una interfaz para que el administrador seleccione las opciones a graficar.
- El administrador selecciona las opciones y oprime “Graficar”.
- El sistema muestra la gráfica generada.

#### 1.3 Condiciones de Ejecución:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<b>Estado:</b> emitidas <b>Condición:</b> profesores		El sistema muestra la gráfica generada.	Satisfactorio	
	<b>Estado:</b> emitidas <b>Condición:</b>	El sistema muestra un mensaje de error señalando que debe seleccionar ambas opciones para generar la gráfica.	Satisfactorio	
	<b>Estado:</b> <b>Condición:</b> profesores	El sistema muestra un mensaje de error señalando que debe seleccionar ambas opciones para generar la gráfica.	Satisfactorio	
	<b>Estado:</b> <b>Condición:</b>	El sistema muestra un mensaje de error señalando que debe seleccionar ambas opciones para generar la gráfica.	Satisfactorio	

## 4.5 CONCLUSIONES.

En este capítulo se trataron temas relacionados con la implementación del sistema y las pruebas de caja negra, se desarrolló el diagrama de despliegue y se confeccionaron los diagramas de componentes por casos de uso. El objetivo de las pruebas de software realizadas fue descubrir errores. Para conseguir este objetivo se planificaron y se ejecutaron una serie de pasos que permitieron la revisión de los elementos del software, como parte de estos se diseñaron los casos de pruebas para cada caso de uso siendo satisfactorio el resultado de los mismos.

## CONCLUSIONES

En el período de prueba al que fue sometido el Sistema de Gestión de Opiniones, las corridas del mismo arrojaron como resultado el cumplimiento del objetivo propuesto en este documento. Haciendo posible que la Facultad Regional pueda contar, de manera rápida y eficiente, con las opiniones relacionadas a los temas que se orienten por la dirección de la misma, proceso que, como se había previsto anteriormente, se tornaba difícil y engorroso. Además se alcanzó una definición de los conceptos más significativos en el dominio del problema.

Se considera, que el presente trabajo ha cumplido con todos los objetivos específicos:

- Se realizó el análisis y diseño del sistema.
- Se implementó un Sistema de Gestión de Opiniones.
- Se implementó un mecanismo que permite generar reportes dinámicos.

El estudio realizado sobre las metodologías existentes para el desarrollo del software tanto libre como propietario permitió concluir que RUP es uno de los procesos más generales de los existentes actualmente ya que está pensado para adaptarse a cualquier proyecto, es importante mencionar que algunos autores consideran que dado el carácter general de RUP, todas las otras metodologías son casos particulares de esta por lo que fue seleccionado para realizar el análisis y diseño del sistema propuesto. Para la implementación del sistema y con el objetivo de lograr una mayor extensibilidad y adaptabilidad al entorno de trabajo se utilizaron herramientas libres como PostgreSQL para la gestión de la base de datos y PHP para la programación permitiendo así su implantación en la mayoría de las plataformas. Para mayor funcionalidad del sistema se creó un mecanismo de reportes dinámicos, aspecto novedoso en este trabajo ya que actualmente era limitada la obtención de reportes. Con la implantación del sistema se probaron todas las funcionalidades y los requerimientos del sistema, obteniéndose así resultados satisfactorios y valoraciones positivas por parte de los usuarios finales.

Teniendo en cuenta lo anterior, el Sistema de Gestión de Opiniones de la Facultad Regional de Ciego de Ávila se convierte en una herramienta muy útil para la gestión de los criterios de los estudiantes, profesores y trabajadores de la misma, este facilita y mejora en gran medida el trabajo con las opiniones, los temas de debate, la obtención de estadísticas y reportes, así como la gestión de los usuarios con privilegios de acceso al sistema.

## **RECOMENDACIONES**

Con el objetivo de disponer de una aplicación que abarque todos los procesos que se relacionan con los criterios y opiniones de los usuarios y más segura, se recomienda lo siguiente:

- Utilizar SSL en el servidor de aplicaciones Web Apache, para lograr una mayor seguridad e integridad de los datos que se manejan.
- Implementar un foro de discusión como espacio alternativo para fomentar el trabajo político e ideológico así como la preparación general e integral del los usuarios.
- Implementar una sección para la publicación de noticias relevantes relacionadas con los temas de debate publicados.
- Implementar un espacio para la publicación de artículos enviados por los usuarios relacionados con algunos temas que serán evaluados por editores que según la calidad, el aporte, y la trascendencia de los mismos puedan ser publicados en el sitio.
- Realizar el mantenimiento al software según lo previsto en los requerimientos no funcionales.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Anónimo."Glosario de términos", CampusFormación, 2004. Disponible en: <<http://www.campusformacion.com/glosario.asp>>].
- [2] Thompson, Ivan."Tipos de Encuesta", Promonegocios.net, Julio 2006. Disponible en: <<http://www.promonegocios.net/mercadotecnia/encuestas-tipos.html>>
- [3] Sæther Bakken, Stig. Schmid, Egon. "Manual de Php. PHP Documentation Group ". 2005. <http://www.php.net/docs.php>.
- [4] Pacheco Águila, Yoandry. "Sistema de Manejo de Contenido". Universidad Central de las Villas.2005.No. de páginas 72
- [5] Anónimo. "Tutorial de SQL". (02-2005). <http://www.desarrolloweb.com/manuales/9/>.
- [6] Andrade Fonseca, Roberto. "Manual\_plpgsql". Cap. 'PL/pgSQL - SQLProcedural Language'. (10-2005)
- [7] Booch, G., Rumbaugh, J., Jacobson, I. "El Lenguaje Unificado de Modelado". Addison-Wesley. 1999
- [8] Anónimo. "Documento acerca de la Proyección estratégica de la UCI. " 2003. UCI, Ciudad de la Habana.
- [9] Machado, Alejandro G. "Documento acerca del Proyecto UCI Ciudad Digital".2003. UCI, Ciudad de la Habana.
- [10] Álvarez, Sofía, Hernández Anaisa. "Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML.".2000. La Habana.



## BIBLIOGRAFÍA

1. Antonio, Palos, Juan. "Introducción a J2EE ".2003.  
<http://www.programacion.com/java/tutoriales/J2EE/> [1/11/2006].
2. Larman, Craig. "UML y Patrones, Introducción al análisis y diseño orientado a objetos ".2002. Prentice-Hall. [1/11/2006].
3. Yanes, Rafael. "Revista Latina de Comunicación Social", nº 58, 'Un género entre la opinión y la actualidad'. 2004. <http://www.ull.es/publicaciones/latina/20041858yanes.htm> [3/11/2006].
4. Villariño, Luis. "UML para Web ". 2002. [4/11/2006]
5. Booch, G., Rumbaugh, J., Jacobson, I. "El Lenguaje Unificado de Modelado. Addison-Wesley ". 1999. [4/11/2006]
6. Meyer, B. "Construcción de software orientado a objetos ".1998.Prentice-Hall. [4/11/2006]
7. Joseph, Schmuller. "Aprendiendo UML en 24 horas ".2001.Prentice-Hall. [6/11/2006].
8. Anónimo. "Lenguajes de Programación: Programación Web". <http://lenguajes-de-programacion.com/programacion-web.shtml>. [6/11/2006].
9. Anónimo. "Tutorial de SQL". <http://www.desarrolloweb.com/manuales/9/>. [6/11/2006]
10. Anónimo. "Páginas dinámicas". <http://www.desarrolloweb.com/manuales/7/>. [8/11/2006].
11. Sæther Bakken, Stig. Schmid, Egon. "Manual de Php. PHP Documentation Group ". 2003. <http://www.php.net/docs.php>. [8/11/2006].
12. Tarrillo, Sergio.  
"AJAX".2007.<http://geeks.ms/blogs/sergiotarrillo/archive/2007/01/09/8420.aspx>. [9/1/2007].
13. Anónimo. "XML". <http://www.monografias.com/trabajos6/ixml/ixml.shtml> [18/01/2007].
14. Anónimo. "Servidor de aplicaciones Web Apache". <http://www.apache.org/> [19/01/2007].  
[http://www.osmosislatina.com/aplicaciones/servidor\\_web.htm](http://www.osmosislatina.com/aplicaciones/servidor_web.htm) [19/01/2007].
15. Anónimo. "Aplicaciones Web".  
[http://www.masternewmedia.org/es/aplicaciones\\_web/temas\\_de\\_aplicaciones\\_web/](http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/) [20/01/2007].
16. Anónimo. "Modelo Cliente Servidor". <http://www.oei.eui.upm.es> [18/01/2007].

17. Clases de Ingeniería del Software I, curso 2005-2006, UCI. [15/2/2007]  
<http://www.mailxmail.com/curso/informatica/phpprogramacionweb/capitulo4.htm>.  
[25/2/2007].
18. Martínez, J., Juan. "Contraseñas de un solo tipo". 2002.  
<http://www.usebox.net/jjm/passw/> [25/4/2007].
19. Anónimo."Glosario de términos", CampusFormación, 2004.  
<http://www.campusformacion.com/glosario.asp>. [25/4/2007].
20. Thompson, Ivan."Tipos de Encuesta", Promonegocios.net, Julio 2006.  
<http://www.promonegocios.net/mercadotecnia/encuestas-tipos.html>[25/4/2007].
21. Sæther Bakken, Stig. Schmid, Egon. "Manual de Php. PHP Documentation Group ".  
2005. <http://www.php.net/docs.php>. [28/4/2007].
22. Andrade Fonseca, Roberto. "Manual\_plpgsql". 2005. Cap. 'PL/pgSQL - SQLProcedural Language'. [10-2005) [8/5/2007].
23. Anónimo. "Documento acerca de la Proyección estratégica de la UCI. " 2003. UCI, Ciudad de la Habana. [15/5/2007].
24. Machado, Alejandro G. "Documento acerca del Proyecto UCI Ciudad Digital".2003. UCI, Ciudad de la Habana. [15/5/2007].
25. Álvarez, Sofía, Hernández Anaisa. "Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML." .2000. La Habana. [23/5/2007].
26. Pacheco Águila, Yoandry. "Sistema de Manejo de Contenido". Universidad Central de las Villas.2005.No. de páginas 72. [4/6/2007].
27. Informatización, Dirección. "Arquitectura para los Sistemas que Conforman la Intranet Universitaria".2007. Ciudad de la Habana. [10/5/2007].

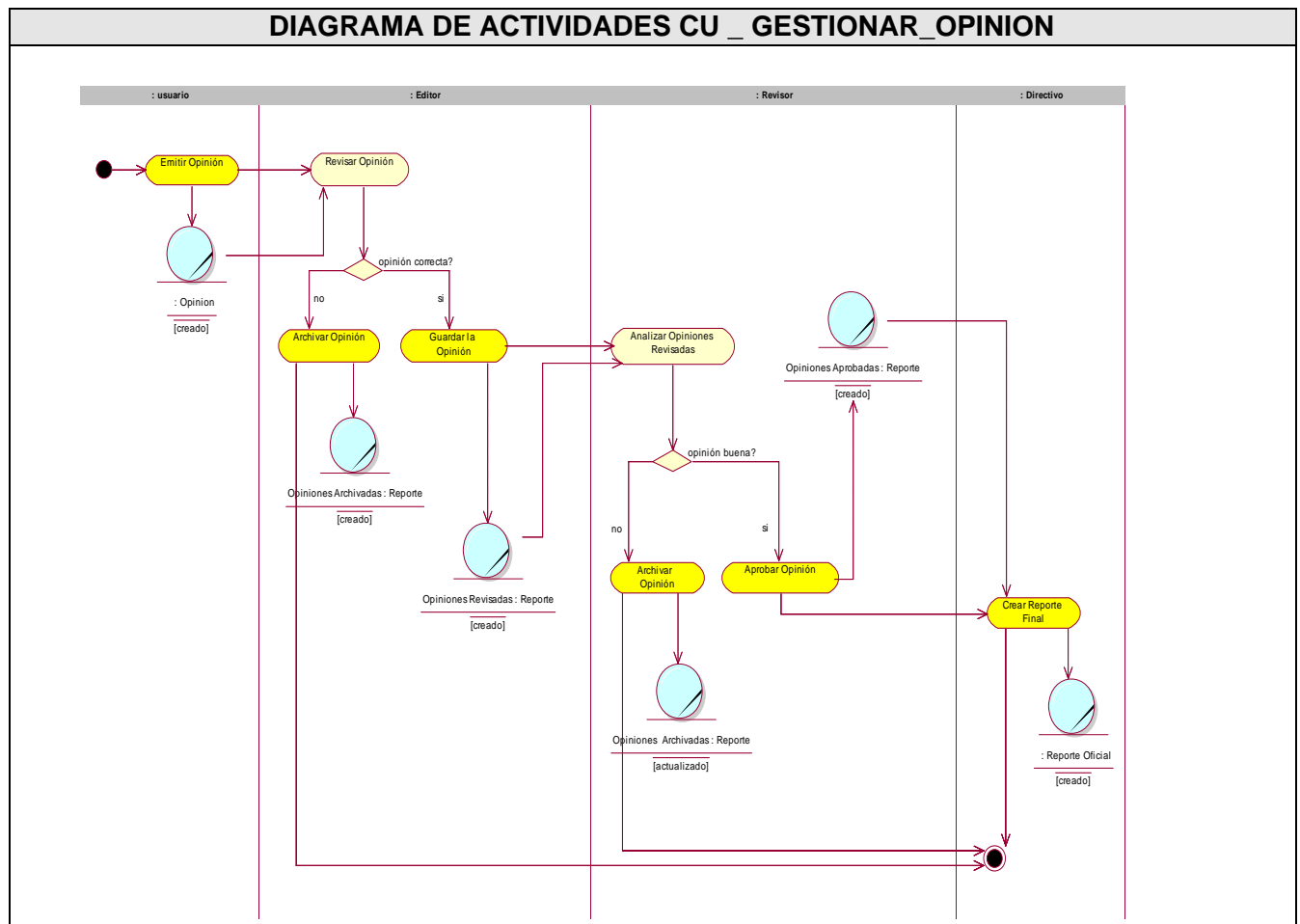
# ANEXOS

## ANEXO 1

### A1.1 Especificación de los casos de uso del negocio con su diagrama de actividad correspondiente.

<b>Caso de uso del negocio:</b>	Gestionar Opinión
<b>Actores del negocio:</b>	Usuario
<b>Propósito:</b>	Este caso de uso garantiza que cuando un usuario emite una opinión esta sea revisada y aprobada o archivada según la calidad que presente, para crear el documento oficial que se envía al Consejo de Estado.
<b>Casos de uso asociados:</b>	-
<b>Resumen:</b> El proceso comienza cuando un usuario emite una opinión de los temas que se orientan por la Dirección del PCC, un directivo se encarga de recoger estas opiniones ya sean emitidas por correo o en formato duro y se revisan, aprueban o archivan. Un especialista crea el documento final con el resumen de las opiniones aprobadas que es enviado al Consejo de Estado.	
<b>Flujo de trabajo</b>	
<b>Acción del actor</b>	<b>Respuesta del negocio</b>
1. El usuario emite una opinión del tema publicado por correo electrónico o en formato duro.	2. El editor recibe la opinión con los datos personales del usuario. 3. El editor revisa la opinión y crea un reporte con las opiniones revisadas. 4. El editor envía el reporte con las opiniones revisadas al revisor. 5. El revisor aprueba las opiniones y crea un reporte con las aprobadas. 6. El revisor envía el reporte con las opiniones aprobadas al directivo. 7. El directivo crea reporte oficial.
<b>Prioridad:</b>	Crítico
<b>Mejoras:</b>	La automatización de este proceso permite al especialista tomar los datos del usuario interno de la UCI con un identificador del mismo. Las opiniones de los usuarios así como las informaciones referentes a los temas quedan guardadas en una base de datos para ser

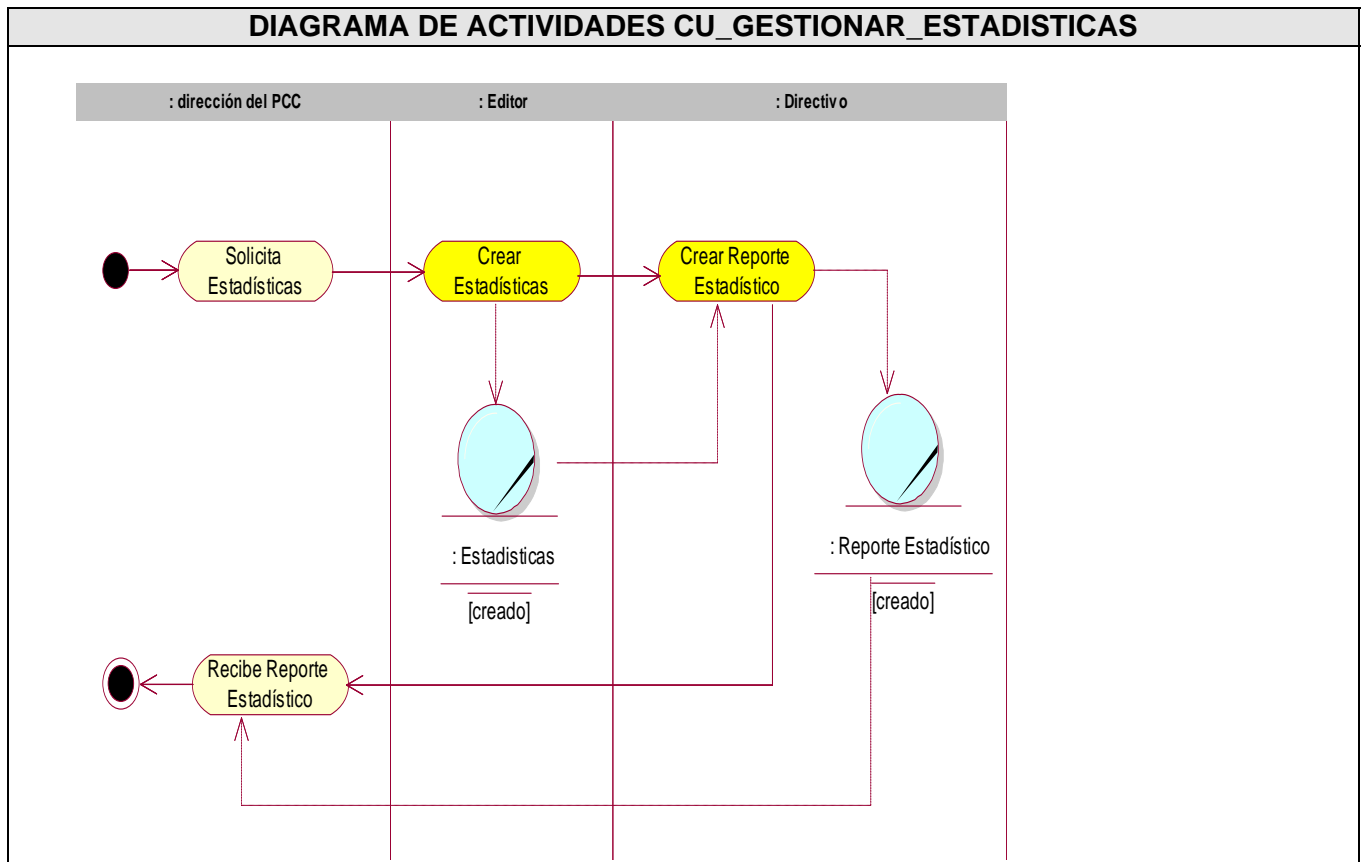
	consultado en caso de que se requiera. Agiliza el proceso de revisión de las opiniones y creación del documento haciendo que sea menos engorroso. También daría la posibilidad de obtener estadísticas y facilitaría el trabajo con las mismas.
<b>Cursos alternos:</b>	
<b>1.a. Emitir opinión en papel.</b>	
1.a.1 El Usuario emite una opinión del tema publicado en un papel.	1.a.2 El directivo correspondiente recoge las opiniones emitidas en papel con los datos personales del usuario y las entrega al Revisor.
<b>3.a. La opinión no es correcta.</b>	
	3.a.1. El Editor revisa la opinión y determina que no es correcta.
	3.a.2. El Editor crea un reporte donde archiva la opinión.
<b>5.a. La opinión no es aprobada.</b>	
	5.a.1. El Revisor no aprueba la opinión.
	5.a.2. El Revisor actualiza el reporte de las opiniones archivadas.



**Figura A1.1: Diagrama de Actividades del Caso de Uso Gestionar Opinión.**

<b>Caso de uso del negocio:</b>	Gestionar Estadística
<b>Actores del negocio:</b>	Dirección del PCC
<b>Propósito:</b>	Este caso de uso garantiza que la Dirección del PCC pueda obtener estadísticas relacionadas con las opiniones emitidas.
<b>Casos de uso asociados:</b>	-
<b>Resumen:</b>	El proceso comienza cuando la Dirección del PCC solicita al editor las estadísticas relacionadas con las opiniones emitidas. El directivo recibe las estadísticas y crea entonces el reporte estadístico final que es enviado a la Dirección del PCC.

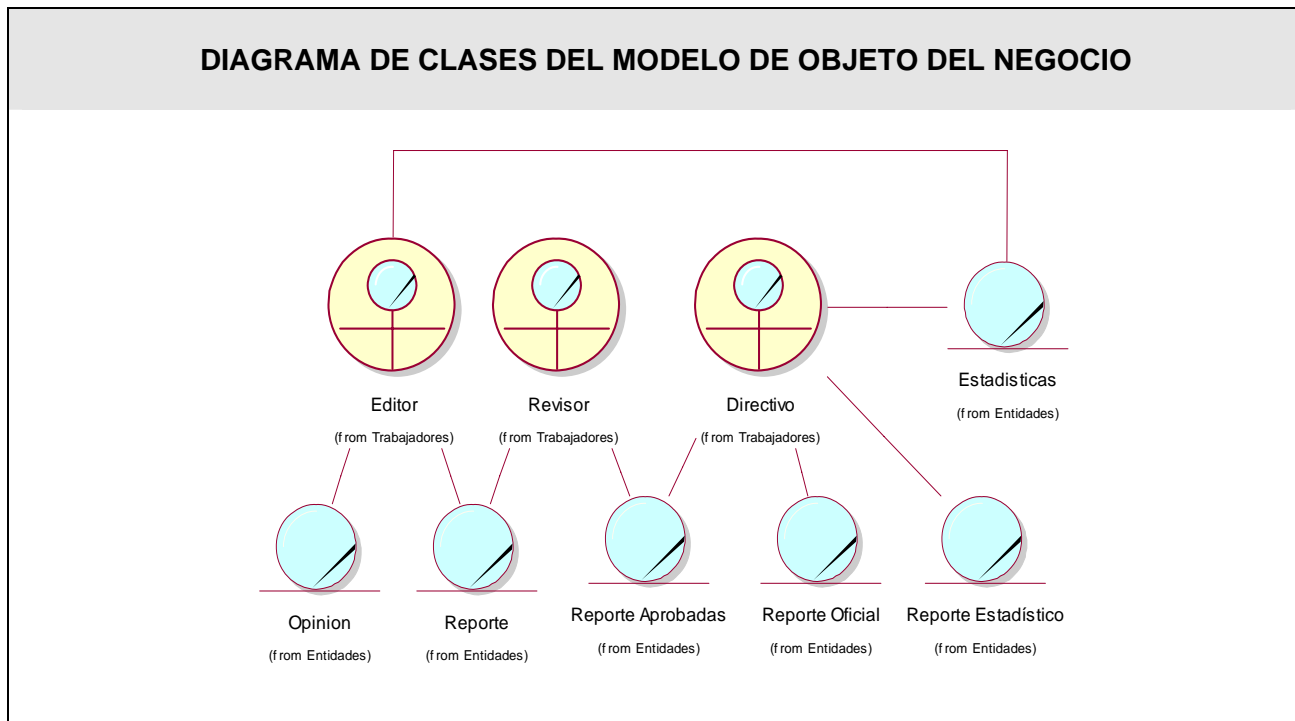
Flujo de trabajo	
Acción del actor	Respuesta del negocio
1. La Dirección del PCC solicita las estadísticas.	2. El Editor crea un reporte con las estadísticas. 3. El Editor envía este reporte al directivo. 4. El directivo crea el reporte final con las estadísticas y las envía a la Dirección del PCC.
<b>Prioridad:</b>	Opcional
<b>Mejoras:</b>	La automatización de este proceso permitirá obtener diferentes tipos de reportes estadísticos y en menor tiempo, incluyendo gráficas de barras o pastel.
<b>Cursos alternos:</b>	



**Figura A1.2: Diagrama de Actividades del Caso de Uso Gestionar Estadísticas.**

## ANEXO 2

### A2.1 Diagrama de Clases del Modelo de Objeto de Negocio.



**Figura A2.1: Diagrama del Modelo de Objetos del Negocio**

### ANEXO 3.

#### A3 Descripción de los actores del sistema.

Basándose en que los casos de uso son una secuencia de acciones que obtienen resultados de valor para un actor y un actor representa cualquier cosa que interactúe con el sistema que puede ser un humano, un software o hardware, [7] se definieron los actores involucrados en los casos de uso:

Actores	Justificación
Usuario	Es la persona que interactúa con el sistema para emitir una opinión.
Editor	Es la persona que interactúa con el sistema para revisar o archivar una opinión.
Revisor	Es la persona que interactúa con el sistema para aprobar o archivar una opinión.
Administrador	Es la persona que interactúa con el sistema para gestionar una opinión, roles a usuarios, temas de debate, generar reportes, los Logs de actividades y graficar estadísticas.

#### A3.1 Descripción de casos de uso del sistema.

Partiendo de que un caso de uso es un documento narrativo que describe la secuencia de eventos de un actor (agente externo) que utiliza un sistema para completar un proceso [10]. Se realizó la definición general de los casos de uso.

Nombre del CU	Autenticarse.	
Actores	Usuario (inicia).	
Objetivo	Acceder al sistema.	
Resumen	El caso de uso se inicia cuando el Usuario solicita entrar al sistema.	
Referencias	R12	
Precondiciones	Usuario válido del dominio. Roles definidos.	
Poscondiciones	Se otorgan los privilegios de acceso.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Usuario solicita entrar al sistema.	2. El sistema muestra una interfaz para que el usuario ingrese su nombre de usuario y contraseña.	
3. El Usuario ingresa su nombre usuario y contraseña, y selecciona la opción "Aceptar".	4. El sistema valida los datos entrados.	
	4.1 El sistema verifica que el usuario es un usuario válido del dominio.	



	4.2 El sistema verifica los permisos del usuario y otorga privilegios.
	4.3 El sistema muestra la interfaz de la página de inicio.
Curso alternativo	
4a. Datos Incorrectos.	
	4a.1. El sistema muestra un mensaje de error señalando los campos que son incorrectos.
4a.2. El Usuario selecciona la opción "Aceptar".	4a.3. Retorna al paso 2.
4b. Usuario no válido.	
	4b.1. El sistema muestra un mensaje de error señalando que el usuario o la contraseña son incorrectos.
4b.2 El Usuario selecciona la opción "Aceptar".	4b.3. Retorna al paso 2.
Prioridad: Crítico	

Nombre del CU	Insertar Opinión.
Actores	Usuario (inicia).
Objetivo	Permitir a los usuarios emitir opiniones sobre los temas publicados.
Resumen	El caso de uso se inicia cuando el Usuario solicita al sistema insertar una opinión.
Referencias	R2
Precondiciones	Usuario autenticado en el sistema. Debe existir al menos un tema de debate activo.
Poscondiciones	Se inserta la opinión y se actualiza.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción "Insertar Opinión".	2. El sistema muestra una interfaz con los temas activos para opinar.
3. El Usuario selecciona la opción "Opinar" de un tema activo.	4. El sistema despliega y muestra un campo para escribir la opinión.
5. El Usuario escribe la opinión y selecciona la opción "Enviar".	6. El sistema valida los datos entrados por el usuario.
	6.1 El sistema almacena en la base de datos la opinión con algunos datos del usuario que la emitió.
	6.2. El sistema muestra un mensaje de confirmación informando al usuario que su opinión ha sido enviada.
7. El Usuario selecciona la opción "Aceptar".	8. Retorna al paso 2. Culmina el caso de uso.
Curso alternativo	
2a. No hay temas activos.	
	2a.1. El sistema muestra un mensaje confirmando que no hay temas disponibles para opinar.
5a. Operación cancelada.	
5a.1. El Usuario selecciona la opción "Cancelar".	5a.2. Retorna al paso 2.

6a. Datos Incorrectos.	
	6a.1. El sistema muestra un mensaje de error señalando que los datos son incorrectos.
6a.2. El Usuario selecciona la opción "Aceptar".	6a.3. Retorna al paso 2.
Prioridad: Crítico	

Nombre del CU	Revisar Opinión.
Actores	Revisor (inicia).
Objetivo	Permitir al Editor revisar las opiniones emitidas por los usuarios.
Resumen	El caso de uso se inicia cuando el Editor solicita al sistema revisar las opiniones emitidas.
Referencias	R3
Precondiciones	Editor autenticado en el sistema. Deben existir opiniones sin revisar.
Poscondiciones	Se modifica el estado de la opinión o la opinión y se actualiza la base de datos.

#### Curso Normal de los Eventos

Acciones del Actor	Respuesta del Sistema
1. El Editor selecciona la opción "Revisar Opinión".	2. El sistema muestra una interfaz con la lista de los temas existentes en la base de datos.
3. El Editor selecciona la opción "Listar" del tema seleccionado.	4. El sistema muestra una interfaz donde va presentando una por una de las opiniones emitidas por los usuarios. Actualiza la opinión seleccionada, marcándola como "en revisión".
	4.1. Editar opinión, ver sección "Editar Opinión".
5. El Editor selecciona la opción "Guardar".	6. El sistema actualiza el estado de la opinión en la base de datos, y la marca como revisada.
	6.1 Retorna al paso 4. Culmina el caso de uso.

#### Curso alternativo

2a. No hay temas.	
	2a.1. El sistema muestra un mensaje confirmando que no hay temas creados.

4a. No hay opiniones del tema.	
	4a.1. El sistema muestra un mensaje de confirmación señalando que no hay opiniones por revisar del tema seleccionado.

4a.2. El Editor selecciona la opción "Aceptar".	4a.3. Retorna al paso 2.
---	--------------------------

#### 4b. Archivar la opinión.

4b.1 El Editor marca la opción "archivar opinión", y selecciona "Guardar".	4b.2. El sistema actualiza el estado de la opinión, y la marca como archivada.
--	--

	4b.3. Retorna al paso 4.
--	--------------------------

#### Sección: Editar la opinión

7. El Editor selecciona la opción "Editar".	8. El sistema muestra la opinión de forma editable.
---	---

9. El Editor revisa la opinión y selecciona la opción "Guardar".	10. El sistema valida los datos entrados por el Editor.
--	---

	10.1. El sistema actualiza la opinión.
--	--

	10.2 Retorna al paso 6.
Curso alternativo	
10a. Datos incorrectos.	
	10a.1. El sistema muestra un mensaje de error señalando que los datos entrados son incorrectos.
10a.2. El Editor selecciona la opción "Aceptar".	10a.3. El sistema actualiza el estado de la opinión. Retorna al paso 2.
8a. Operación cancelada.	
8a.1. El Editor selecciona la opción "Cancelar".	8a.2. Retorna al paso 4.
Prioridad: Crítico	

Nombre del CU	Aprobar Opinión.
Actores	Revisor (inicia).
Objetivo	Permitir al Revisor seleccionar las opiniones que serán utilizadas para enviar en el reporte de las opiniones.
Resumen	El caso de uso se inicia cuando el Revisor solicita al sistema aprobar las opiniones.
Referencias	R5
Precondiciones	Revisor autenticado en el sistema. Las opiniones tienen que estar revisadas.
Poscondiciones	Se modifica el estado de la opinión o la opinión y se actualiza la base de datos.

Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Revisor selecciona la opción "Aprobar Opinión".	2. El sistema muestra una interfaz con la lista de los temas existentes en la base de datos.
3. El Revisor selecciona la opción "Listar" del tema seleccionado.	4. El sistema muestra una interfaz donde va presentando una por una de las opiniones revisadas. Actualiza la opinión seleccionada en la base de datos, marcándola como "en revisión".
	4.1. Editar opinión, ver sección "Editar Opinión".
5. El Revisor selecciona la opción "Guardar".	6. El sistema actualiza el estado de la opinión y la marca como aprobada.
	6.1. Retorna al paso 4. Culmina el caso de uso.

Curso alternativo	
2a. No hay temas.	
	2a.1. El sistema muestra un mensaje confirmando que no hay temas creados.
4a. No hay opiniones revisadas del tema.	
	4a.1. El sistema muestra un mensaje de confirmación señalando que no hay opiniones por aprobar del tema seleccionado.
4a.2. El Revisor selecciona la opción "Aceptar".	4a.3. Retorna al paso 2.
4b. Archivar la opinión.	
4b.1 El Revisor marca la opción "archivar opinión", y selecciona "Guardar".	4b.2. El sistema actualiza el estado de la opinión, y la marca como archivada.
	4b.3. Retorna al paso 4.

4c. No aprobar la opinión.	
4c.1. El Revisor selecciona la opción “no aprobar”.	4c.2. El sistema actualiza la opinión y la marca como “no aprobada”.
	4c.3. Retorna al paso 4.
Sección: Editar la opinión	
7. El Revisor selecciona la opción “Editar”.	8. El sistema muestra la opinión de forma editable.
9. El Revisor revisa la opinión y selecciona la opción “Guardar”.	10. El sistema valida los datos entrados por el aprobador.
	10.1. El sistema actualiza la opinión en la base de datos.
	10.2 Retorna al paso 6.
Curso alternativo	
10a. Datos incorrectos.	
	10a.1. El sistema muestra un mensaje de error señalando que los datos entrados son incorrectos.
10a.2. El Revisor selecciona la opción “Aceptar”.	10a.3. El sistema actualiza el estado de la opinión. Retorna al paso 2.
8a. Operación cancelada.	
8a.1. El Revisor selecciona la opción “Cancelar”.	8a.2. Retorna al paso 4.
Prioridad: Crítico	

Nombre del CU	Gestionar opiniones archivadas.	
Actores	Administrador (inicia).	
Objetivo	Permitir al administrador revisar nuevamente las opiniones que han sido archivadas.	
Resumen	El caso de uso se inicia cuando el Administrador solicita al sistema revisar las opiniones archivadas.	
Referencias	R4	
Precondiciones	Administrador autenticado en el sistema. Deben existir opiniones archivadas.	
Poscondiciones	Se modifica la opinión o se elimina, y se actualiza.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Administrador selecciona la opción “Opiniones Archivadas”.	2. El sistema muestra una interfaz con la lista de los temas existentes que poseen opiniones archivadas.	
3. El Administrador selecciona la opción “Listar” del tema seleccionado.	4. El sistema muestra una interfaz donde va presentando una por una las opiniones archivadas. Actualiza la opinión seleccionada, marcándola como “en revisión”.	
5. El Administrador selecciona la opción “Editar”.	6. El sistema muestra la opinión de forma editable.	
7. El Administrador edita la opinión y selecciona la opción “Guardar”.	8. El sistema valida los datos entrados por el Administrador.	

	8.1. El sistema actualiza la opinión y se almacena como “aprobada”.
	8.2. El sistema actualiza el estado de la opinión.
	8.3. Retorna al paso 4.
Curso alterno	
2a. No hay temas con opiniones archivadas.	
	2a.1. El sistema muestra un mensaje confirmando que no hay opiniones archivadas.
4a. Eliminar opinión.	
4a.1. El Administrador selecciona la opción “Eliminar”.	4a.2. El sistema elimina la opinión.
	4a.3. El sistema muestra un mensaje de confirmación señalando que la opinión ha sido eliminada.
4a.4. El Administrador selecciona la opción “Aceptar”.	4a.5. Retorna al paso 4.
4b. No hay más opiniones archivadas del tema.	
	4b.1. El sistema muestra un mensaje de confirmación señalando que no hay más opiniones archivadas del tema seleccionado.
4b.2. El Administrador selecciona la opción “Aceptar”.	4b.3. Retorna al paso 2.
8a. Datos incorrectos.	
	8a.1. El sistema muestra un mensaje de error señalando que los datos entrados son incorrectos.
8a.2. El Administrador selecciona la opción “Aceptar”.	8a.3. Retorna al paso 4.
Prioridad: Secundario	

Nombre del CU	Gestionar Tema.	
Actores	Administrador (inicia).	
Objetivo	Permitir al Administrador gestionar los temas para opinar.	
Resumen	El caso de uso se inicia cuando el Administrador solicita al sistema la gestión de de temas, permitiéndole agregar nuevos tema, modificarlos o eliminarlos.	
Referencias	R1	
Precondiciones	Administrador autenticado en el sistema.	
Poscondiciones	Se crea el tema y se actualiza.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Administrador selecciona la opción “Temas”.	2. El sistema despliega un menú mostrándole dos opciones.	
	2.1. Crear tema, ver sección “Crear Tema”.	
	2.2. Editar tema, ver sección “Editar Tema”.	
Sección: Crear Tema		
3. El Administrador selecciona la opción “Crear Tema”.	4. El sistema muestra una interfaz para crear el nuevo tema.	
5. El Administrador introduce los datos del nuevo tema y selecciona la opción “Guardar”.	6. El sistema valida los datos entrados por el administrador.	

	6.1. El sistema muestra un mensaje de confirmación señalando que el tema ha sido creado.
7. El Administrador selecciona la opción "Aceptar".	8. Retorna al paso 4. Finaliza el caso de uso.
Curso Alterno	
4a. Cancelar operación.	
4a.1. El Administrador selecciona la opción "Cancelar".	4a.2. El sistema resetea los datos entrados en el formulario.
	4b.3. Retorna al paso 4.
6a. Datos incorrectos.	
	6a.1. El sistema muestra un mensaje de error señalando los campos con errores.
6a.2. El Administrador selecciona la opción "Aceptar".	6a.3. Retorna al paso 4.
Sección: Editar Tema	
3. El Administrador selecciona la opción "Editar Tema".	4. El sistema muestra una interfaz con los temas existentes.
5. El Administrador selecciona la opción "Editar" del tema seleccionado.	6. El sistema muestra los datos del tema de forma editable.
7. El Administrador modifica los datos del tema, selecciona la opción "Actualizar".	8. El sistema valida los datos entrados.
	8.1. Retorna al paso 4. Finaliza el caso de uso.
Curso Alterno	
4a. No hay temas creados.	
	4a.1. El sistema muestra un mensaje señalando que no hay temas creados.
4b. Eliminar tema.	
4b1. El Administrador selecciona la opción "Eliminar".	4b.2. El sistema elimina el tema de la base de datos, y todas las opiniones y estadísticas relacionada con el mismo.
	4b.3. El sistema muestra un mensaje de confirmación señalando que el tema ha sido eliminado.
4b.4. El Administrador selecciona la opción "Aceptar".	4b.5. Retorna al paso 4.
6a. Cancelar la operación.	
6a.1 El Administrador selecciona la opción "Cancelar".	6a.2. El sistema vuelve a mostrar los datos del tema sin campos editables, y sin modificar.
Prioridad: Crítico	

Nombre del CU	Gestionar Usuario.
Actores	Administrador (inicia).
Objetivo	Permitir al Administrador gestionar los usuarios que trabajarán con el sistema.
Resumen	El caso de uso se inicia cuando el Administrador solicita al sistema la gestión de de los usuarios, permitiéndole agregar nuevos usuarios, modificarlos o eliminarlos.
Referencias	R7
Precondiciones	Administrador autenticado en el sistema. Roles definidos.

Poscondiciones	Se asigna los privilegios al usuario y se actualiza la base de datos.	
Curso Normal de los Eventos		
Acciones del Actor		Respuesta del Sistema
1. El Administrador selecciona la opción "Usuario".		2. El sistema despliega un menú mostrándole dos opciones.
		2.1. Asignar rol a usuario, ver sección "Crear Usuario".
		2.2. Editar usuario, ver sección "Editar Usuario".
Sección: Crear Usuario		
3. El Administrador selecciona la opción "Crear Usuario".		4. El sistema muestra una interfaz para crear el nuevo usuario.
5. El Administrador introduce el usuario del dominio del nuevo usuario que desea crear, selecciona la opción "Buscar".		6. El sistema valida los datos entrados por el administrador.
		6.1. El sistema verifica que es un usuario válido.
		6.2. El sistema muestra los datos relacionados con el nuevo usuario.
7. El Administrador otorga el permiso al nuevo usuario y selecciona la opción "Crear".		8. El sistema verifica que el usuario no existe.
		8.1. El sistema almacena el nuevo usuario.
		8.2. El sistema muestra un mensaje de confirmación señalando que el usuario ya ha sido creado.
9. El Administrador selecciona la opción "Aceptar".		10. Retorna al paso 4. Finaliza el caso de uso.
Curso Alterno		
6a. Datos incorrectos.		
		6a.1. El sistema muestra un mensaje de error señalando que los datos introducidos son incorrectos.
6a.2. El Administrador selecciona la opción "Aceptar".		6b.3. Retorna al paso 4.
6b. Usuario no válido.		
		6b.1. El sistema muestra un mensaje de error señalando que el usuario no es válido.
6b.2. El Administrador selecciona la opción "Aceptar".		6b.3. Retorna al paso 4.
6c. Cancelar la operación.		
6c.1. El Administrador selecciona la opción "Cancelar".		6c.2. Retorna al paso 4.
8a. El usuario ya existe.		
		8a.1. El sistema muestra un mensaje de error señalando que el usuario ya existe.
8a.2. El Administrador selecciona la opción "Aceptar".		8a.3. Retorna al paso 4.
Sección: Editar Usuario		
11. El Administrador selecciona la opción "Editar Usuario".		12. El sistema muestra una interfaz con los usuarios existentes en la base de datos.

13. El Administrador selecciona la opción "Editar" del usuario seleccionado.	14. El sistema muestra el permiso del usuario de forma editable.
15. El Administrador modifica los permisos del usuario, selecciona la opción "Guardar".	16. El sistema actualiza los datos del usuario.
	16.1. Retorna al paso 12 Finaliza el caso de uso.
Curso Alterno	
13a. Eliminar usuario.	
13a.1. El Administrador selecciona lo opción "Eliminar" del usuario seleccionado.	13a.2. El sistema elimina el usuario de la base de datos.
	13a.3. El sistema muestra un mensaje de confirmación señalando que el usuario ha sido eliminado.
13a.4. El Administrador selecciona la opción "Aceptar".	13a.5. Retorna al paso 12.
15a. Cancelar operación.	
15a1. El Administrador selecciona la opción "Cancelar".	15a.2. El sistema muestra de nuevo los datos del usuario sin campos editables.
Prioridad: Crítico	

Nombre del CU	Gestionar Reporte.	
Actores	Administrador (inicia).	
Objetivo	Permitir al Administrador crear reportes de opiniones y estadísticos.	
Resumen	El caso de uso se inicia cuando el Administrador solicita al sistema crear reporte, seleccionando los temas que poseen opiniones para la obtención de los mismos.	
Referencias	R6, R10, R11	
Precondiciones	Administrador autenticado en el sistema. Deben existir opiniones.	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Administrador selecciona la opción "Reportes".	2. El sistema despliega un menú mostrándole las opciones "Opiniones" y "Estadísticas".	
	2.1. Crear reporte de opiniones, ver sección "Opiniones".	
	2.2. Crear reporte de estadísticas, ver sección "Estadísticas".	
Sección: Opiniones		
3. El Administrador selecciona la opción "Opiniones".	4. El sistema muestra una interfaz con los temas que poseen opiniones.	
5. El Administrador selecciona las opciones que desea incluir en el reporte y oprime "Vista Previa".	6. El sistema muestra una interfaz con una vista previa del reporte.	
7. El Administrador selecciona la opción "Crear Documento".	8. El sistema genera el reporte en formato "rtf".	
Curso Alterno		



4a. No hay opiniones.	
	4a.1. El sistema muestra un mensaje de confirmación señalando que no existen opiniones.
6a. Cancelar Operación.	
6a.1. El Administrador selecciona la opción "Cerrar".	6a.2. Retorna al paso 4.
Sección: Estadísticas	
9. El Administrador selecciona la opción "Estadísticas".	10. El sistema muestra una interfaz con los temas que poseen opiniones.
11. El Administrador selecciona los tipos de estadísticas que desee incluir y oprime "Vista Previa".	12. El sistema muestra un interfaz con una vista previa del reporte.
13. El Administrador selecciona "Generar documento".	14. El sistema genera el reporte en formato "rtf".
Curso Alterno:	
10a. No hay opiniones.	
	10a.1. El sistema muestra un mensaje de confirmación señalando que no existen opiniones.
13a. Cancelar operación.	
13a.1. El Administrador selecciona la opción "Cancelar".	14a.2. Retorna al paso 10.
Prioridad: Crítico	

Nombre del CU	Graficar Estadísticas.	
Actores	Administrador (inicia).	
Objetivo	Permitir al Administrador crear gráficas con las estadísticas.	
Resumen	El caso de uso se inicia cuando el Administrador solicita al sistema graficar estadísticas, seleccionando los temas que poseen opiniones para generar las gráficas.	
Referencias	R8, R9	
Precondiciones	Administrador autenticado en el sistema. Debe existir opiniones.	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Administrador selecciona la opción "Graficar Estadísticas".	2. El sistema muestra una interfaz con los temas que poseen opiniones.	
3. El Administrador despliega un menú del tema seleccionado y elige la opción "Personalizar Estadísticas".	4. El sistema muestra una interfaz con las opciones para seleccionar la estadística.	
5. El Administrador selecciona las opciones para la estadística y oprime "Graficar"	6. El sistema muestra la gráfica creada.	
Curso Alterno		

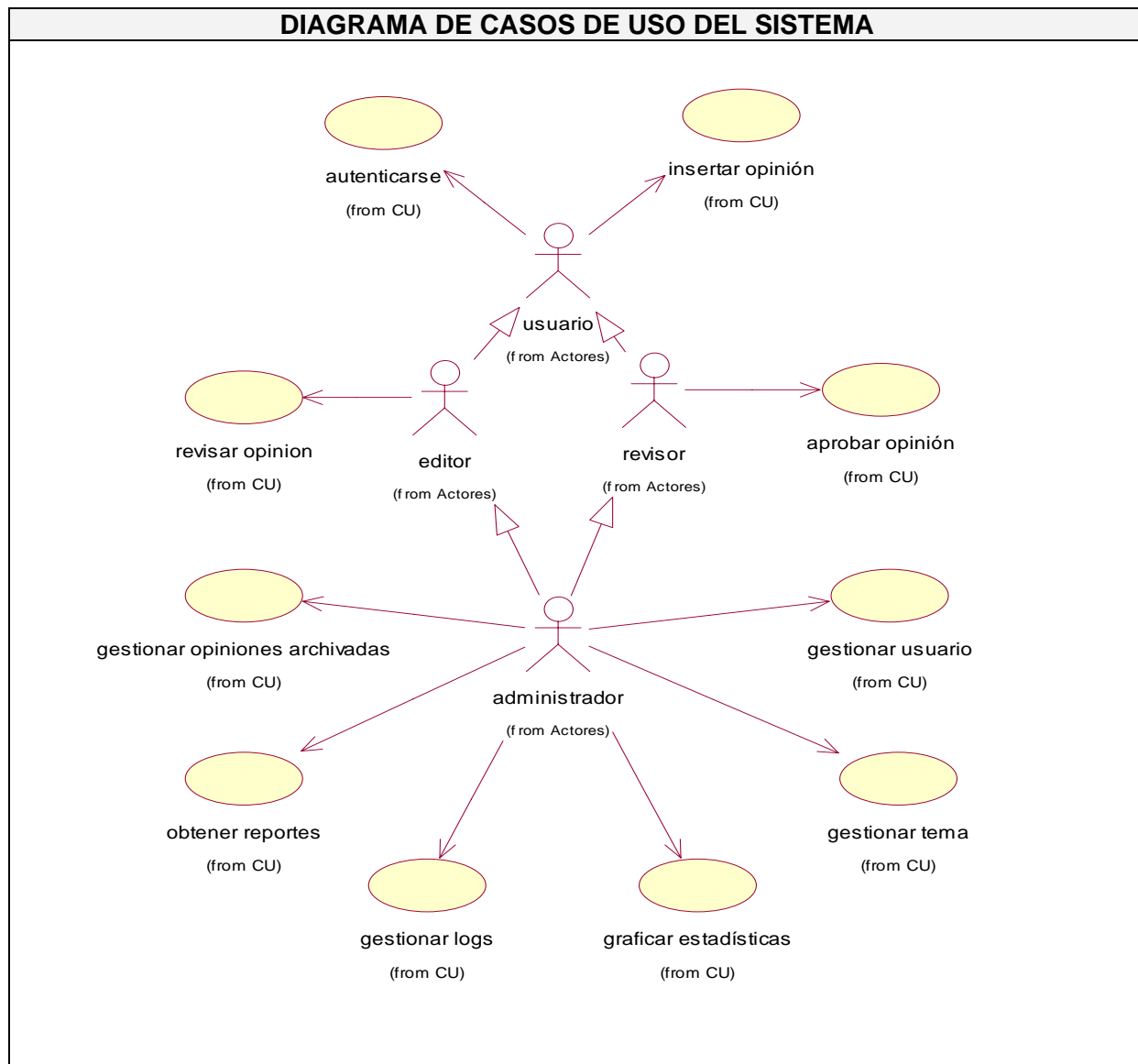
5a. Selecciona opción "Opiniones por año"	
5a.1. El Administrador selecciona la opción "Opiniones por Año".	5a.2. El sistema muestra en una interfaz la gráfica generada.
Curso Alterno	
5b. Selecciona la opción "Opiniones por Tipo de Usuario"	
5b.1. El Administrador selecciona la opción "Opiniones por tipo de usuario"	5b.2. El sistema muestra en una interfaz la gráfica generada.
Curso Alterno	
5c. Selecciona la opción "Opiniones por Estado"	
5c.1. El Administrador selecciona la opción "Opiniones por Estado".	5c.2. El sistema muestra en una interfaz la gráfica generada.
Curso Alterno	
6a. No elige las opciones.	
	4a.1. El sistema muestra un mensaje de error señalando que debe seleccionar las opciones a graficar.
Curso Alterno	
6b. No hay opiniones.	
	6b.1. El sistema muestra un mensaje señalando que no hay opiniones para la estadística seleccionada.
6b.2. El Administrador selecciona "Aceptar".	6b.3. Retorna al paso 4.
Prioridad: Crítico	

Nombre del CU	Gestionar Logs.	
Actores	Administrador (inicia).	
Objetivo	Permitir al Administrador revisar los Logs del sistema.	
Resumen	El caso de uso se inicia cuando el Administrador solicita al sistema revisar los Logs.	
Referencias	R13	
Precondiciones		
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Administrador selecciona la opción "Revisar Logs".	2. El sistema muestra una interfaz donde muestra los días que poseen Logs.	
3. El Administrador selecciona el día y oprime la opción "Mostrar".	4. El sistema muestra una interfaz con los Logs generados en el día seleccionado.	
Curso alternativo		
3a. Eliminar Logs.		
3a.1. El Administrador selecciona la opción "Eliminar".	4a.1. El sistema elimina los Logs relacionados con el día seleccionado.	
	4a.2. El sistema muestra un mensaje de confirmación señalando que los Logs han sido eliminados.	

Prioridad: Crítico

### A3.3 Diagramas de casos de uso

Basándose en que un diagrama de casos de uso representa un conjunto de casos de uso para un sistema, los actores y la relación entre casos de uso y actores, se realizó el diagrama de casos de uso que se muestra a continuación.



**Figura A3.1: Diagrama del Modelo de Casos de Uso del Sistema.**

## GLOSARIO

### A

**AJAX:** acrónimo de *Asynchronous JavaScript And XML* (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas.

**ALTER:** este comando permite modificar la estructura de un objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.

**Apache:** es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

**ASP:** acrónimo en inglés de *Active Server Pages (ASP)* es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente.

### B

**Bases de Datos:** es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.

**BSD:** es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Pertenece al grupo de licencias de software Libre.

### C

**C:** lenguaje de programación creado en 1969 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell.

**C++:** lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C.

**CASE:** (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

**CGI:** *Common Gateway Interface* (en castellano «Interfaz Común de Pasarela», abreviado CGI) es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web.

**CPU:** unidad central de proceso (UCP) en inglés CPU ó simplemente procesador. Es el componente en una computadora digital que interpreta las instrucciones y procesa los datos contenidos en los programas de computadora.

**CSS:** hojas de estilo en cascada (*Cascading Style Sheets*, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

## D

**Dreamweaver MX:** es un editor de páginas web, creado por Adobe (anteriormente por Macromedia).

## F

**Firebird:** es un sistema de administración de base de datos relacional de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000.

**Flash:** se refiere tanto al programa de edición multimedia como a Macromedia Flash Player, escrito y distribuido por Adobe.

## G

**GET:** método utilizado para enviar formularios HTML a través de la web.

**GNU/GPL:** (*General Public License* o licencia pública general) es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

## H

**Hash:** se refiere a una función o método para generar claves o llaves que representen de manera casi unívoca a un documento, registro, archivo, etc.

**HTML:** acrónimo inglés de *HyperText Markup Language* (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

**Http:** protocolo de transferencia de hipertexto (*HTTP*, *HyperText Transfer Protocol*) es el protocolo usado en cada transacción de la Web (WWW).

**Hyperlinks:** también llamado enlace, vínculo, hipervínculo o liga, es un elemento de un documento electrónico que hace referencia a otro recurso, por ejemplo, otro documento o un punto específico del mismo o de otro documento.

## I

**IDE:** en inglés *Integrated Development Environment* (*IDE*) es un programa compuesto por un conjunto de herramientas para un programador.

**Internet Explorer:** es un navegador de Internet producido por Microsoft para su plataforma Windows y más tarde para Apple Macintosh y Solaris Unix.

**IP:** es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (*Internet Protocol*), que corresponde al nivel de red o nivel 3 del modelo de referencia OSI.

## J

**Java:** es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990.

**JavaScript:** es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

**JSP:** es una tecnología para crear aplicaciones web, su funcionamiento se basa en scripts, que utilizan una variante del lenguaje java.

## L

**Linux:** es la denominación de un sistema operativo tipo-Unix y el nombre de un núcleo.

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

**OpenLDAP:** Se trata de una implementación libre del protocolo que soporta múltiples esquemas por lo que puede utilizarse para conectarse a cualquier otro LDAP.

## M

**Magic Quotes:** comillas mágicas, es una funcionalidad que se encarga automáticamente de añadir slashes a las comillas dobles y simples de la información que llega a una página vía parámetros GET, POST o a los que se guardan en cookies.

**MaxDB:** es un sistema de administración de bases de datos adquirido por la compañía SAP para usarse como un repositorio de datos para las aplicaciones de SAP.

**MD5:** (acrónimo de *Message-Digest Algorithm 5*, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.

**Microsoft Internet Information Services:** IIS, es una serie de servicios para los ordenadores que funcionan con Windows.

**Microsoft Windows:** es un sistema operativo con interfaz gráfica para computadoras personales cuyo propietario es la empresa Microsoft.

**Mosaic:** fue el segundo navegador gráfico disponible para visualizar páginas web.

**MVC:** *Modelo Vista Controlador (Model View Controller)*, es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

**MVCC:** acrónimo inglés de *Multi-Version Concurrency Control*, Acceso Concurrente Multiversión.

**MySQL:** es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

## N

**Navegador:** navegador web o explorador web (del inglés, *navigator* o *browser*) es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet.

Netscape:

## O

**Open Source:** código abierto es el término con el que se conoce al software distribuido y desarrollado libremente.

**Oracle:** es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), se considera a Oracle como uno de los sistemas de bases de datos más completos,

**OTP:** es el acrónimo en inglés de *One-Time Password*, que puede traducirse como contraseña de un solo uso.

## P

**PC:** en inglés *Personal Computer*, computadora personal.

**Perl:** Lenguaje Práctico para la Extracción e Informe, es un lenguaje de programación diseñado por Larry Wall creado en 1987.

**PHP:** acrónimo recursivo que significa "*PHP Hypertext Pre-processor*" (inicialmente PHP Tools, o, *Personal Home Page* Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

**PL/pgSQL:** (*Procedural Language/PostgreSQL Structured Query Language*) es un lenguaje imperativo provisto por el gestor de base de datos PostgreSQL.

**POO:** *Programación Orientada a Objetos* un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan *estado* (es decir, datos), *comportamiento* (esto es, procedimientos o *métodos*) e identidad (propiedad del objeto que lo diferencia del resto).

**POST:** método utilizado para enviar formularios *HTML* a través de la Web.

**Postgre:** es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD.

**Python:** es un lenguaje de programación creado por Guido van Rossum en el año 1990.

## R

**RAM:** es el acrónimo inglés de *Random Access Memory* Module (memoria de acceso aleatorio ó memoria de acceso directo).

**RUP:** Proceso Unificado de Rational (RUP, el original inglés *Rational Unified Process*) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

## S

**SGBD:** Sistema de gestor de Base de Datos: son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

**SGO :** Sistema de Gestión de Opiniones.

**SQL Server:** es un sistema de gestión de bases de datos relacionales.

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

**SSL:** *Secure Sockets Layer* es un protocolo criptográfico que proporciona comunicaciones seguras en Internet.

## T

**TCP/IP:** es un conjunto de protocolos de red que implementa la pila de protocolos en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

## U

**UML:** Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es un lenguaje de modelado de sistemas de software.

**Unix:** es un sistema operativo portable, multitarea y multiusuario.

**Unix Domain Sockets:** protocolo de comunicación utilizado en sistemas UNIX.

## W

**Web:** es un sistema de documentos de hipertexto enlazados y accesibles a través de Internet.

## X

**XML:** Sigla en inglés de *Extensible Markup Language*, lenguaje de marcas extensible.

## Z



**Zend:** Compañía líder de infraestructuras para web; está reconocida internacionalmente como la autoridad actualmente en PHP. Sus fundadores son los diseñadores del PHP v.4 en adelante, actualmente es una compañía líder dentro de la comunidad Open Source.