

Universidad de las Ciencias Informáticas

FACULTAD 6



Título: Sistema de Gestión de Opiniones.

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autores: José Luis Vidal Cardoso.

Yaquerleydis Cutiño Nápoles.

Tutor: Ing. Yannia Moreira Gamboa

Ciudad de La Habana, Cuba
Junio 2007

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al área de informatización de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Jose Luis Vidal Cardoso

Yaquerleydis Cutiño Nápoles

Yannia Moreira Gamboa

Datos de Contacto

"[insertar breve curriculum e información de contacto del tutor]"

"[insertar breve curriculum e información de contacto del asesor]"

"[insertar breve curriculum e información de contacto del consultante]"

Agradecimientos

Sin duda alguna esta es la parte más difícil de redactar, quisiéramos poder mencionar a todos los que de una forma u otra pusieron su granito de arena para hacer realidad este sueño, pero son tantos que realmente no alcanzaría esta página para mencionarlos.

Agradecerle:

A la revolución cubana y a nuestro Comandante en Jefe Fidel Castro por darnos la posibilidad de estudiar en la Universidad de las Ciencias Informáticas, de realizar nuestros sueños y contribuir al desarrollo de nuestro país.

A nuestra tutora Yánnia Moreira Gamboa por su ayuda, dedicación y paciencia en todos los momentos.

A los profesores que participaron en nuestra formación profesional.

A nuestros compañeros los Jimaguas Daniel y David, Dogma, Duniel, Guillermo, Cansino, Renier, Danay y Jani por su ayuda incondicional en la realización de este trabajo.

Yo Yaquerleydis Cutiño Nápoles:

A los que proporcionan sentido a mi vida: a ti Abita que eres mi ejemplo a seguir, a mis tres tesoros y fuente de inspiración Abasito, Yuky y Leydita, a mi mamita adorada Idaelsis y a mi papito Abat. A mis segundos padres Mari y Carlos, que aunque él no esté entre nosotros en vida sigue en mi pensamiento y en mi corazón. A mis mejores amigas y hermanas Aime, Joann y Mayi, a mis abuelitos Erada e Iraldo.

A mi novio, compañero de tesis y amigo Jose, a todos mis tíos y primos (no me alcanzaría la hoja para mencionarlos). Yoha gracias por ser tan especial, a todas las personas queridas que en un momento determinado me han apoyado en mi vida profesional y personal.

Yo José Luis Vidal Cardoso:

A mis seres más queridos mis padres, por comprenderme, guiarme, ayudarme y "aguantarme" a ustedes Celia y Andrés son a quienes más les tengo que agradecer, les agradezco por existir y a ustedes le debo todo lo que soy un millón de gracias.

Al resto de mi familia por su apoyo incondicional, mis abuelitos Armando y Gladis, mis hermanos en especial a mi hermanita del alma Damaris y mi sobrinita Yuliet, a mi cuñado Amirís, sin excepción a todos mis tíos y primos.

A mi novia, amiga y compañera de tesis yaquerleydis por haber tenido tanta paciencia conmigo en los momentos difíciles cuando me ponía insoportable.

En general a las amistades que nos dieron fuerzas para seguir adelante por malos que fuesen los tiempos, los que siempre estuvieron ahí deseándonos éxitos. A todos los que en algún momento nos preguntaron ¿Y la tesis?

Muchas gracias de todo ♥.

Resumen

Título: Sistema de Gestión de Opiniones.

La Universidad de las Ciencias Informáticas (UCI) está orientada a desarrollar proyectos que informaticen diferentes servicios y requiere de un sistema que gestione todo el proceso de las opiniones que serán enviadas al Consejo de Estado, con la mayor rapidez posible. En la actualidad la UCI cuenta con un sistema que no responde a algunas de las funcionalidades básicas; como es el proceso de revisión de las opiniones, la actualización del estado de las opiniones, la ausencia de reportes estadísticos y la publicación simultánea de varios temas de debates. Teniendo en cuenta esta necesidad, el objetivo de esta investigación es implementar una aplicación Web que facilite el proceso de gestionar las opiniones. La puesta en marcha de este sistema permitirá dar solución a los problemas actuales existentes en la gestión de opiniones, permitiendo mejorar la rapidez y calidad de los servicios.

Índice

Introducción	1
Capítulo 1: Fundamentación teórica	4
1.1 ¿Qué es un Sistema de Opinión?	4
1.1.2 Tipos de Sistema de Opinión.....	4
1.2 Tendencias actuales de los Sistemas de Opiniones.....	5
1.3 Tecnologías actuales	6
1.3.1 Aplicaciones web.....	6
1.3.2 Modelo Cliente/Servidor	8
1.3.3 Tecnologías del lado del cliente	10
1.3.4 Tecnología del lado del servidor.....	11
1.4 Sistema Gestor de Bases de Datos	12
1.4.1 PostgreSQL	14
1.5 Servidor Web.....	15
1.6 Metodologías de desarrollo	16
1.7 Herramientas de desarrollo	18
1.7.1 Macromedia Dreamweaver.....	18
1.7.2 Zend Studio	19
1.7.3 Case Studio	20
1.7.4 PGAdmin III	20
1.7.5 EMS Postgre SQL Manager	21
Capítulo 2: Características del sistema	22
2.1 Problema y situación problemática	22
2.1.1 Objetivos estratégicos de la organización	22
2.1.2 Flujo actual de los procesos	22

2.1.3 Análisis crítico.....	23
2.2 Objeto de automatización.....	23
2.2.1 Procesos a automatizar	23
2.2.2 Sistema automatizado vinculado con el campo de acción	24
2.3 Información que se maneja	24
2.4 Propuesta de sistema.....	24
2.5 Modelo de dominio	26
2.5.1 Glosario de términos.....	26
2.5.2 Modelo de objeto del dominio	27
2.5.3 Reglas del Negocio.....	27
2.6 Especificación de los requerimientos de software	27
2.6.1 Requerimientos funcionales	28
2.7 Casos de uso	31
2.7.1 Definición de los actores.....	31
2.7.2 Definición de los casos de uso	32
2.7.3 Diagrama de casos de uso del sistema.....	34
2.7.4 Descripciones de casos de uso	35
Capítulo 3: Análisis y diseño del sistema.....	36
3.1 Modelo de análisis.....	36
3.1.1 Modelo de clases de análisis	36
3.2 Modelo de diseño	36
3.2.1 Diagramas de Interacción.....	37
3.2.1.1 Diagrama de Colaboración	37
3.2.2 Diagrama de clases del diseño Web	37
3.2.3 Descripción de las clases del diseño	38

3.3 Diseño de la Base de Datos	38
3.3.1 Diagrama Entidad Relación	38
3.4 Definiciones de diseño que se apliquen	39
3.5 Tratamiento de errores	39
3.6 Seguridad	39
3.7 Interfaz	40
3.8 Concepción de la ayuda	40
Capítulo 4: Implementación	41
4.1 Modelo de implementación	41
4.1.2 Diagrama de despliegue	41
4.1.3 Diagrama de componentes	42
Conclusiones	46
Recomendaciones	47
Bibliografías	48
Referencias Bibliográficas	48
Bibliografías	49
Anexos	51
Glosario:	90

Introducción

El acelerado avance en los conocimientos científicos y sus aplicaciones tecnológicas en el mundo actual, constituyen la base fundamental para el desarrollo económico y social de los pueblos. A pesar de las dificultades que causa el bloqueo, Cuba se ha mantenido inmerso en esta revolución científico técnica con el objetivo de cumplir un plan estratégico para lograr informatizar todo el país.

La Universidad de las Ciencias Informáticas surge como resultado de este proceso, institución nacida al calor de la batalla de ideas; con una matrícula conformada por jóvenes de todas las regiones de la nación, donde la tarea fundamental está encaminada a desarrollar proyectos que informaticen diferentes servicios.

En el año 2003 ocurre la violación del espacio aéreo cubano; a partir de este suceso surge la idea por la máxima Dirección del país de recoger las opiniones del pueblo sobre este hecho. Estudiantes, profesores y trabajadores de la UCI se convierten en los principales protagonistas de esta tarea.

La recogida de opiniones sobre diversos temas de interés, que posteriormente eran trasladadas hasta el consejo de estado donde nuestro comandante las analizaba para saber cuál era el sentir de su pueblo con respecto a los diferentes sucesos de los que se opinaban, se convirtió en una tarea más de la Universidad.

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

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

En la actualidad la UCI cuenta con un SGO donde los estudiantes, profesores y trabajadores opinan sobre determinados temas nacional e internacional que son de interés a la Dirección del país. Esta aplicación había logrado organizar un poco el trabajo pero presenta dificultades en algunas de sus funcionalidades básicas que frenan el dinamismo con que se deben procesar las opiniones. 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 y al Comandante, se decide implementar un nuevo SGO.

El problema de investigación surge con la necesidad de solucionar lo antes expuesto, se formula de la siguiente forma: ¿Cómo gestionar de forma eficiente las opiniones emitidas en la UCI?

El objeto de investigación: sistemas de gestión de la información y su campo de acción: sistema de gestión de opiniones en la Universidad de las Ciencias Informáticas.

Se ha propuesto como objetivo general: implementar una aplicación Web que mejore el proceso de gestión de opiniones.

Por otra parte existen una serie de objetivos específicos trazados en función de dar cumplimiento al objetivo general del trabajo:

- Analizar y diseñar la aplicación.
- Implementar la aplicación.

Para alcanzar dichos objetivos se planteó desarrollar las siguientes tareas:

- Estudio de los sistemas de opiniones que existen.
- Entrevistas con los clientes para identificar las necesidades.
- Estudio de las diferentes metodologías de desarrollo.
- Levantamiento de los requisitos funcionales.
- Análisis y diseño del sistema.
- Implementación de la aplicación Web.

Como resultado de este trabajo se espera disponer de un sistema informatizado que mejore la gestión de las opiniones.

El presente documento se divide en varios capítulos:

El Capítulo 1 Fundamentación Teórica: expone los fundamentos teóricos generales basando en los distintos sistemas de opiniones existentes a nivel nacional e internacional que sirven de punto de partida

para analizar el problema a resolver. Se definen conceptos, se hace un estudio de las tecnologías utilizadas en la actualidad para el desarrollo de aplicaciones Web y se describe la metodología a seguir.

El Capítulo 2 Características del Sistema: brinda una descripción general del objeto de estudio, el problema a tratar y la situación problemática, mostrando los objetivos estratégicos de la organización, el flujo actual de los procesos involucrados en el campo de acción y el análisis crítico de cómo se ejecutan actualmente esos procesos. Se describen los procesos que serán objeto de automatización y los sistemas automatizados que existen en la UCI. La información que se maneja en la aplicación, la propuesta del sistema, la descripción del modelo de dominio, especificación de los requisitos del software y definición de los casos de uso.

El Capítulo 3 Análisis y Diseño del Sistema, en el análisis se realiza la definición del modelo de análisis y el modelo de clases de análisis, donde se genera el diagrama de clases del análisis. En la parte del diseño, se representa el diagrama de interacción y el diagrama de de clases del diseño Web, descripción de las clases. El diseño de la base de datos, su diagrama de entidad relación y descripción de tablas. Además definiciones de diseño que se apliquen, tratamiento de errores, seguridad, interfaz y concepción de la ayuda.

El Capítulo 4 Implementación, muestra las definiciones de los diferentes modelos utilizados en la solución del problema. Se realizan los diagramas de despliegue y de componentes.

Capítulo 1: Fundamentación teórica

Introducción

El presente capítulo tiene como objetivo exponer los fundamentos teóricos generales que sirven de punto de partida para analizar el problema a resolver basándonos en los distintos sistemas de opiniones existentes a nivel nacional e internacional. Se definen conceptos que serán de utilidad en la comprensión del mismo, se hace un estudio de las tecnologías utilizadas en la actualidad para desarrollar aplicaciones Web y las metodologías de desarrollo.

1.1 ¿Qué es un Sistema de Opinión?

Una opinión no es más que el parecer o pensamiento que cada persona tiene acerca de una situación o asunto, donde se explica con claridad el porqué de las ideas expuestas, siendo conciso y preciso.

Los Sistemas de Opiniones (SO) no son más que aplicaciones informáticas que tienen como función principal permitir recoger los criterios de múltiples usuarios sobre diferentes temas; que en algunos casos dependiendo del tipo de sistema se almacenan en bases de datos y estos pueden o no, ser públicos. 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. Todos los SO utilizan formularios que varían dependiendo de los intereses con que se quieran utilizar, estos sirven para facilitar la introducción de datos mediante un conjunto de campos, que una vez rellenos van a reflejarse en la base de datos y posteriormente pueden ser revisados y modificados.

1.1.2 Tipos de Sistema de Opinión

En el mundo existen diferentes vías para obtener las opiniones de usuarios a través de la Web.

Foros: "Espacio virtual creados en Internet o en una Intranet en 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 una aplicación web que le da soporte a discusiones en línea." [1]

Encuestas: Es una herramienta para recolectar información de 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 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. Sus desventajas son: No siempre se puede verificar la identidad del encuestado y la interrogante que deja la muestra en cuanto a su representatividad del universo.” [2]

También existen otros tipos de sistemas de opiniones que aunque no tengan una definición formal como foros o encuestas son muy utilizados a nivel mundial. En ocasiones las entidades necesitan que en los sitios Web sea recogida determinado criterio de usuarios sobre uno o varios temas de interés, que no necesariamente debe ser pública como en los foros; ni tampoco, se rige por un cuestionario a responder como en las encuestas. Los formularios de opiniones, sugerencias o críticas, son una página Web o parte de ella realizada atendiendo a la necesidad de la entidad, en la que el usuario completa y devuelve al servidor para su procesamiento.

1.2 Tendencias actuales de los Sistemas de Opiniones

A nivel mundial en Internet 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 tan solo recopilan informaciones en bases de datos, con cierto grado de privacidad.

Como resultado de las búsquedas por Internet encontramos diferentes ejemplos de sistema de opiniones.

<http://www.uc3m.es/uc3m/opina/opinaBiblioteca.html>: 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.

<http://www.opinionyviaje.com/php/obkurl.php>: En el sitio OpinionyViaje.com se pueden enviar opiniones y comentarios mediante dos sistemas: Uno es por email (contactar@opinionyviaje.com) y otro es el proceso de llenar formulario.

En Cuba también en distintos sitios se han desarrollado espacio para recoger las opiniones de los diferentes usuarios, que con sus ideas hacen que el trabajo sea mejor cada día.

<http://www.uo.edu.cu/sugerencias.php>: En el portal de la universidad de oriente, encontramos una muestra de cómo recogen las sugerencias de sus usuarios como una vía de retroalimentación para llevar a cabo sus mejoras.

<http://www.uh.cu/infogral/contact.html>: En la Web oficial de la universidad de la Habana, también se preocupan por las opiniones de sus usuarios, motivos por el cual tienen un link donde cada usuario puede opinar acerca de su sitio, dan sugerencias, entre otras cosas.

En nuestro país siempre ha sido de interés la opinión del pueblo a la máxima dirección del estado, para así trazar una política hacia como enfrentar las necesidades y los problemas que aquejan. El Comandante siempre ha mantenido la unidad, con una constante correspondencia con el pueblo.

Conociendo que el programa de la Ciencia Informática es una necesidad del mundo moderno en sus diferentes utilidades en los diversos sectores de la economía y en otras esferas; dándole un empleo eficiente en la tramitación de información, con lo cual se contribuye a mantener informado al comandante del estado de opinión en un momento determinado. Estudiantes y profesores han desarrollado un sistema que recoge opiniones.

<http://intranet.uci.cu/opiniones/procOpinion.asp>: La aplicación Sistema de opiniones es habilitada en la intranet de la UCI al tratarse un suceso relevante a nivel nacional o internacional en las mesas redondas, en algún acto político u otro tipo de evento. Son recogidas las opiniones de los estudiantes, profesores y trabajadores del centro, después de gestionar todo el proceso que encierran las opiniones se llevan al Consejo de Estado.

1.3 Tecnologías actuales

A continuación se hace un estudio de las tecnologías actuales en las que se basa la solución del problema existente en el presente trabajo, así como sus principales características.

1.3.1 Aplicaciones web

“Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet.” [3]

Las aplicaciones Web pueden considerarse como un sitio Web al que se accede de manera normal a través de un navegador, pero dotado de interactividad para la gestión, una base de datos y contraseñas de entrada, sin necesidad de instalar en cada equipo ningún otro programa específico. Debido a estos bajos requerimientos, el software puede utilizarse incluso desde ordenadores obsoletos y poco potentes. Al albergarse la aplicación en un servidor remoto, el funcionamiento y los valiosos datos que contiene son totalmente independientes del ordenador utilizado para la gestión. Así, la normal operación de la aplicación es inmune a una avería de hardware, virus informáticos, o cualquier otro problema local.

La rápida evolución de Internet hace que el entorno web haya experimentado grandes cambios en los últimos años; responden a nuevas necesidades y tendencias, se enfrentan a procesos de renovación y cambios en las organizaciones. La decisión de elegir el modelo de una aplicación Web es debido a que es la opción que más se adapta a nuestras necesidades.

Ventajas de las aplicaciones web:

La plataforma Web aporta interesantes ventajas frente al software de escritorio, que se convierte en una interesante alternativa:

- En las aplicaciones Web tanto el software como los datos se alojan en una máquina diferente centralizada, para trabajar de manera estable, continua e ininterrumpida y no necesitan ser descargadas, instaladas o configuradas. Por el contrario, en el esquema tradicional de aplicaciones de escritorio, el usuario tiene que instalar el software en su ordenador y este almacena la totalidad de los datos generados durante su uso. La información se administra desde una única máquina que posee tanto la propia aplicación como toda la información sensible involucrada. Todos los datos dependen de esta única máquina y de su normal operación.
- Las aplicaciones Web tienen compatibilidad multiplataforma por diseño e interoperabilidad. Se puede conectar desde cualquier versión de Windows y otros sistemas operativos como GNU/Linux, Solaris, Symbian a diferencia de las aplicaciones de escritorio, que sólo pueden funcionar bajo el sistema para el que fueron diseñadas.
- En las aplicaciones de escritorio sólo el usuario frente al ordenador puede utilizar la aplicación a diferencia de las aplicaciones Web pueden haber varios usuarios conectados al sistema simultáneamente, desde ordenadores distintos y en ubicaciones geográficas separadas.

1.3.2 Modelo Cliente/Servidor

La arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura.

La arquitectura tradicional de cliente/servidor también es conocida como arquitectura de dos capas. Requiere una interfaz de usuario que se instala y corre en una estación de trabajo y envía solicitudes a un servidor para ejecutar operaciones complejas.

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 los datos requeridos por la capa intermedia son almacenados.

Generalmente se utilizan los modelos cliente-servidor simple, que están compuestos por dos capas: capa cliente (ambiente de trabajo del usuario por lo que tiene la interfaz de la aplicación) y capa servidora (contiene la base de datos). El procesamiento es compartido entre estos dos ambientes por lo que son usados en exceso los procedimientos almacenados y los disparadores.

En una tercera capa se implementa la lógica del negocio y esta proporciona un ambiente donde miles de usuarios pueden estar conectados simultáneamente, el SGBD no tiene que resolver él solo la comunicación con los clientes.

Ventajas del sistema de dos capas

- El desarrollo de aplicaciones en un ambiente de dos capas es mucho más rápido que en ambientes anteriores, pero no es necesariamente más rápido que con el nuevo ambiente de tres capas.
- Las herramientas para el desarrollo con dos capas son robustas y evaluadas. Las técnicas de prototipo se emplean fácilmente.

- Las soluciones de dos capas trabajan bien en ambientes no dinámicos estables, pero no se ejecutan bien en organizaciones rápidamente cambiantes.

Desventajas del sistema de dos capas

- 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.
- Las herramientas del cliente y de la base de datos, utilizadas en diseños de dos capas, constantemente están cambiando. La dependencia a largo plazo de cualquier herramienta, puede complicar el escalamiento futuro o las implementaciones.

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 y basadas Web

- 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 que supera en muchos aspectos a la de dos capas, para la realización de la aplicación que dará solución al problema. 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 del desempeño de la flexibilidad y escalabilidad; así como de una disminución del tiempo de desarrollo de la administración y de los costos a largo plazo.

1.3.3 Tecnologías del lado del cliente

HyperText Markup Language (HTML)

HTML: acrónimo inglés de HyperText Markup Language (lenguaje de marcas hipertextuales), lenguaje diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Internet Explorer, Firefox o Netscape, 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. [4]

JavaScript

JavaScript: es un lenguaje de programación interpretado por el navegador que se utiliza para controlar su apariencia, manipular los eventos que ocurran en su ventana y se inserta dentro del código HTML. A

través de Javascript se pueden conseguir interesantes efectos en las páginas web, validar formularios, abrir y cerrar ventanas, cambiar dinámicamente el aspecto y los contenidos de una página.

JavaScript no requiere compilación, es utilizado principalmente en páginas Web con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al contrario de Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas Web. [5]

1.3.4 Tecnología del lado del servidor

Professional Home Page Tools (PHP)

PHP: acrónimo de "Hypertext Preprocessor", es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. [6]

Características más importantes de PHP:

- PHP tiene soporte para gestores de bases de datos tales como: Adabas D, dbm, dBase, filePro, Hyperwave, Informix, Interbase, Microsoft SQL Server, mSQL, MySQL, ODBC, Oracle, PostgreSQL, Solid, Sybase lo que permite la creación de aplicaciones web muy robustas. Además el soporte para acceso a servidores IMAP, envío de correo con SMTP, acceso a servidores de FTP, acceso a SNMP para gestión de redes y equipos, generación dinámica de gráficos y documentos PDF, análisis de documentos XML, corrector de ortografía y generación de datos en WDDX (Intercambio Web de Datos Distribuidos).
- PHP es multiplataforma; las páginas funcionan tanto en Sistemas Windows, Mac OS X, Unix o Linux. Otra de las ventajas de esta tecnología es que pertenece al grupo de Proyectos de Código abierto, por lo que se puede utilizar con licencia GPL de código abierto.
- PHP tiene una de las comunidades más grandes en Internet, por lo que cuenta con mucha documentación, artículos, noticias, y recursos. Ofrece una solución simple y universal para las

paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y actualizar, a diferencia del código de otros lenguajes.

- PHP goza de la ayuda de un grupo grande de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades.

Mejoras de PHP5 con respecto a PHP 4.

PHP 5 solucionó los problemas encontrados en PHP 4. Los cambios estructurales que se realizaron mejoran muchas áreas en el lenguaje, como es la programación orientada a objetos (OOP) y la integración nativa con el Zend Engine.

La principal novedad en las clases de PHP5 es la inclusión de modificadores de control de acceso para implementar la encapsulación, punto fundamental en la programación orientada a objetos de la que padecía PHP4.

PHP5 introduce tres palabras clave (`public`, `private` y `protected`) que sustituyen a `var` en la definición de variables miembro (atributos) de la clase y que preceden a la definición de funciones miembro (métodos).

Los diseñadores de PHP5 han realizado un cambio radical en el tratamiento de las variables objeto: en PHP5 todas las variables que nombran objetos son en realidad referencias. No hay que usar el operador `&` ni en las asignaciones, ni en el paso de parámetros que son objetos, ahorrándose con ello gran cantidad de potenciales errores. [7]

Php5 es la última versión del lenguaje de servidores open source más utilizado alrededor del mundo, con el cuál es posible gestionar todas las actividades del lado del servidor y conectarse con bases de datos. Y de acuerdo con las mejoras antes mencionadas, para implementar la propuesta de sistema de este trabajo fue seleccionado PHP5.

1.4 Sistema Gestor de Bases de Datos

Los Sistemas de gestión 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. Se compone de un lenguaje de definición de datos, manipulación de datos y consulta. En los textos que tratan este tema se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema

Gestor de Bases de Datos y DataBase Management System. Fundamentalmente el propósito general de los sistemas de gestión de base de datos es manejar de manera clara, sencilla y ordenada un conjunto de información. [8]

Características de los SGBD:

- Los SGBD definen varios niveles de abstracción de la información, ya que no muestran a los usuarios detalles acerca del almacenamiento físico de los datos.
- Tienen independencia de los datos por la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- Un buen diseño de una base de datos logrará evitar la aparición de información repetida. Lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias mínima.
- Los SGBD deben garantizar que la información se encuentre asegurada, frente a ataques que deseen manipular o destruir la información; o simplemente torpezas de usuarios.
- Los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- Garantizan la integridad y validez de los datos almacenados, protege los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- Los SGBD deben proporcionar respaldo y recuperación, al realizar de forma eficientes copias de seguridad de la información almacenada y de restaurar a partir de estas copias los datos que se hayan podido perder.
- El acceso de forma simultánea de muchas personas a una base de datos, para recuperar información o almacenar es muy frecuente; por lo que debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

1.4.1 PostgreSQL

PostgreSQL fue uno de los primeros en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. Es un sistema objeto-relacional, incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. [9]

PostgreSQL es el gestor de base de datos de código abierto más avanzado hoy en día, ofrece control de concurrencia multi-versión, soporta casi toda la sintaxis SQL incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario, cuenta con un amplio conjunto de enlaces con lenguajes de programación.

Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Principales características del gestor de bases de datos PostgreSQL:

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...) y cadenas de bits.
- También permite la creación de tipos propios.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, entre otras.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.

- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Lo mejor de PostgreSQL según las opiniones más comunes en Internet:

- Posee gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, es capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).
- Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz y ofreciendo soluciones en campos en las que MySQL no podría.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, comparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.

Mejoras en las nuevas versiones.

Rendimiento: mejora el rendimiento en alrededor de un 20% en pruebas de sistemas OLTP (procesamiento en línea de transacciones) de gran escala. Los cambios incluyen ordenamientos en memoria y en disco más rápido, mejor escalabilidad en sistemas multi-procesador, mejor optimización de consultas sobre datos particionados, cargas masivas más rápidas y outer joins considerablemente acelerados.

Construcción en línea de índices: la construcción de índices puede ocurrir mientras las aplicaciones escriben a las tablas de la base de datos, permitiendo el afinamiento de rendimiento sin afectar la disponibilidad.

1.5 Servidor Web

Servidor Web Apache

El Proyecto apache es un esfuerzo de desarrollo de software colaborativo apuntado a crear un una aplicación de código de fuente libremente disponible de un servidor HTTP (Web) robusto, con calidad

comercial. Es importante señalar que más del 62 % de los Servidores Web de Internet utilizan este Software lo que indica la calidad y seguridad del mismo.

Principales características del Servidor Web Apache:

- Autenticación Implícita.
- Soporte de varios Módulos de terceras partes.
- Capacidad de servir protocolos SSL a través del Módulo Mod_SSL.
- Facilidades para la creación de Servidores Virtuales.
- Supremacía en el Mercado de Servidores Web de Internet.

1.6 Metodologías de desarrollo

En la actualidad existen varias metodologías para el desarrollo de software que tienen como características generales: que no pueden aplicarse a todo tipo de proyectos, están orientadas en función de los nuevos principios de desarrollo del software y pueden ser ajustables de acuerdo a las características del proyecto. A continuación se abordan las principales características de las más conocidas:

Extreme Programming (XP)

Es una metodología de desarrollo de software utilizada para proyectos de corto plazo, pocos integrantes en el equipo de trabajo y cuyo plazo de entrega sea breve. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Microsoft Solution Framework (MSF)

Es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. Se adapta a proyectos de cualquier dimensión y de cualquier tecnología. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

Rational Unified Process (RUP)

RUP unifica los mejores elementos de las metodologías más conocidas, está preparada para desarrollar grandes y complejos proyectos, es Orientado a Objetos y utiliza el UML como lenguaje de representación visual. Esta metodología está guiada por Casos de Uso, iterativo e incremental y centrada en la Arquitectura.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Lo más importante antes de elegir la metodología que se usa para la implementación de un software, es determinar el alcance que tendrá y luego ver cual es la que más se acomoda a la aplicación. Haciendo un análisis de las metodologías estudiadas, se determinó emplear en este trabajo RUP.

El propósito de explicar el flujo de Modelamiento de Negocio empleando RUP se deriva de su fácil y entendible trazabilidad hacia los demás flujos. Por su gran flexibilidad se puede aplicar a cualquier tipo de proyecto, ya sean grandes o pequeños, obviamente empleando un grado variable de artefactos en cada flujo, además se pueden omitir diversos diagramas y modelos en algunos casos.

El Proceso Unificado de Desarrollo y el Lenguaje Unificado de Modelado no constituyen el único proceso, ni la única notación existente actualmente para hacer software; pero ambos son estándares internacionales muy bien aceptados.

En la comunidad del software UML ha ejercido un gran impacto, tanto a nivel de desarrollo como de investigación. Su éxito y utilización ha sido enorme en todo el mundo para construir aplicaciones de todos los tamaños. UML es un estándar de cualquier industria que requiera la construcción de modelos como condición previa para el diseño y posterior construcción de prototipos.

Unified Modeling Lenguaje (UML):

El UML o Lenguaje de Modelación Unificado es un lenguaje gráfico para especificar, construir, visualizar y documentar las partes o artefactos. Pueden ser artefactos: un modelo, una descripción que comprende el desarrollo de software que se basen en el enfoque Orientado a Objetos, utilizándose también en el diseño Web. UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos. [6]

Herramientas:

La creación de una herramienta CASE (ingeniería de software asistida por computador) denominada Rational CASE fue otro gran aporte, en este caso no sólo conceptual sino práctica en forma de herramientas.

Rational:

La Corporación Rational ofrece un Proceso Unificado (RUP) para el desarrollo de los proyectos de software, desde la etapa de Ingeniería de Requerimientos hasta la de pruebas. Para cada una de estas etapas existe una herramienta de ayuda en la administración de los proyectos.

Rational Suite es la solución integrada y completa del ciclo de vida de un proyecto, reúne las mejores prácticas, herramientas y servicios.

- Unifica equipos de funcionalidades cruzadas en un entorno Windows a través del flujo de trabajo y las integraciones de productos clave.
- Incluye la plataforma de unificación de equipos de IBM Rational para gestionar el desarrollo durante el ciclo de vida. Acelera el desarrollo a través del modelado visual y las funciones de generación de código e ingeniería inversa.
- Encuentra y elimina errores de ejecución, pérdidas de memoria y cuestiones de rendimiento.
- Incluye las mejores prácticas, herramientas líderes del mercado y procesos que se pueden configurar.
- Proporciona todas las herramientas que necesita su equipo en un solo paquete.
- Ofrece acceso a foros de discusión y documentación específicas de productos y activos que se pueden volver a utilizar como un miembro de la comunidad de desarrollo online de IBM Rational.

[10]

1.7 Herramientas de desarrollo

1.7.1 Macromedia Dreamweaver

Macromedia Dreamweaver es un software diseñado para desarrollar páginas web profesionales. Hace fácil la creación de complejas páginas Web dinámicas, con la conocida técnica de arrastrar y soltar,

permitiendo que los diseñadores puedan crear entornos Web y animaciones sofisticadas sin tener que escribir una sola línea de código.

Novedades de Macromedia Dreamweaver 8.0 sobre Dreamweaver MX 2004.

- Con Dreamweaver 8 se introducen datos en formato XML fácil y cómodamente de otras páginas con sólo introducir la fuente, arrastrar y colocar los campos.
- La versión 8 ha mejorado con respecto a la compatibilidad y manejo de estilos de cascada, donde ahora se accede a la configuración de cada uno de los estilos con la facilidad de modificar sus propiedades. Añade una barra de herramientas que proporciona la reproducción inmediata de los estilos para diferentes medios como pantalla, impresora entre otros.
- Se puede seguir trabajando con los archivos mientras el programa realiza actualizaciones y transferencia. Su sincronización hace posible una mejor gestión de cambios, además de permitir en uso de bloqueo/desbloqueo de archivos para que estos no se sobre escriban.
- Los usuarios con problemas visuales podrán acceder a una interfaz mejorada con opción de Aumento de la pantalla. Además de la inclusión de información visual gracias a las guías que permitirán la medición píxel a píxel de todos los elementos.
- Se añadió una barra de herramientas que hace más accesible el código al permitirnos la navegación por etiquetas y su contracción. Una de las nuevas novedades es la posibilidad de añadir comentarios con un sólo clic.
- Se le añadió compatibilidad en esta versión con PHP5, Coldfusion MX 7 y Video Flash. [11]

1.7.2 Zend Studio

Zend Studio constituye un ambiente de desarrollo integrado que agrupa todos los componentes 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, 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 la interfaz de edición y la ayuda, en la parte del servidor se desarrolla la herramienta de depuración que instala Apache y el módulo PHP.

Zend a lanzado con relativa facilidad y rapidez, versiones con nuevos rasgos del producto para Windows, Linux y MacOS que se diferencian de las anteriores.

ZendStudioClient

El Zend Estudio Cliente es un editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos. Esta plataforma de Zend puede trabajar de manera autónoma; actúa como el componente del servidor y permite la depuración de código, tiene mayor aumento de fiabilidad y actuación de su aplicación antes del despliegue. Es compatible con el descargo de PHP 5. Se conecta directamente a las bases de datos profesionales ampliamente usadas como MySQL, Oracle, Microsoft SQL Servidor, PostgreSQL, entre otros. [12]

1.7.3 Case Studio

Case Studio posibilita diseñar base de datos; tanto a personas con pocas experiencias como a profesionales, dado que su intuitiva interfaz gráfica hace fácil las tareas más complicadas. Mediante esta herramienta se realizan diagramas de flujo con muy poco tiempo y esfuerzo. El programa dispone de parches de idioma para un gran número de países. Soporta Oracle, MySQL, MS SQL, MaxDB, Firebird, PostgreSQL y otros sistemas. Permite generar rápidamente diagramas gráficos de las bases de datos relacionales, simplificando el trabajo del programador. Su principal características, es su potente sistema de ingeniería inversa, que permite identificar y estructurar bases de datos ya existentes para poder trabajar con ellas sin problemas.

1.7.4 PGAdmin III

Es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

PGAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL. 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 Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix) y puede encriptarse mediante SSL para mayor seguridad. [13]

1.7.5 EMS Postgre SQL Manager

El SQL Manager para PostgreSQL es una herramienta de gran alcance para la administración y el desarrollo de las base de datos en PostgreSQL. El manejador de PostgreSQL trabaja con cualquiera de las versiones de PostgreSQL hasta 8.2 y la característica más importante es que incluye los nombres de los argumentos en funciones. Ofrece diversidad de herramientas de gran alcance para los usuarios experimentados, tales como, diseñador visual de la base de datos y constructor visual de consultas sql que satisfacen todas las necesidades.

Conclusiones

El presente trabajo tiene como propósito desarrollar una aplicación para la informatización del proceso de gestionar las opiniones llamado, Sistema de Gestión de Opiniones. Teniendo en cuenta la política de informatización de la UCI para desarrollar la aplicación se utilizará: como SGBD PostgreSQL 8.1, como servidor web, Apache 2.0.55 y como lenguaje de programación PHP 5.1.1.

Dentro de las herramientas de desarrollo escogidas están: Dreamweaver 8.0 para el diseño de las páginas web, como manejador de BD pgAdmin III y EMS PostgreSQL Manager 3, el ZendStudioClient 5.0.0 como editor de php. 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) y como herramientas CASE se seleccionaron Rational Suite 2003 y para modelar las BD el Case Studio 2.18 que permite diseñar y exportar ficheros sql para Postgre.

Capítulo 2: Características del sistema

Introducción

El presente capítulo brinda una descripción general del objeto de estudio, el problema y situación problemática relacionado con el SGO existente en la UCI, información que se maneja en la aplicación, la propuesta del sistema, la descripción del dominio y los requerimientos funcionales del sistema.

El objeto de estudio de esta investigación es sistemas de gestión de la información, basadas en las opiniones recogidas en la UCI.

2.1 Problema y situación problemática

2.1.1 Objetivos estratégicos de la organización

La Universidad de Ciencias Informáticas está encaminada a la formación de profesionales que sean capaces de desarrollar software. También se realizan investigaciones en diferentes campos como la Bioinformática, los Equipos Médicos, Multimedia, entre otros; haciendo uso de las nuevas tecnologías de la información.

2.1.2 Flujo actual de los procesos

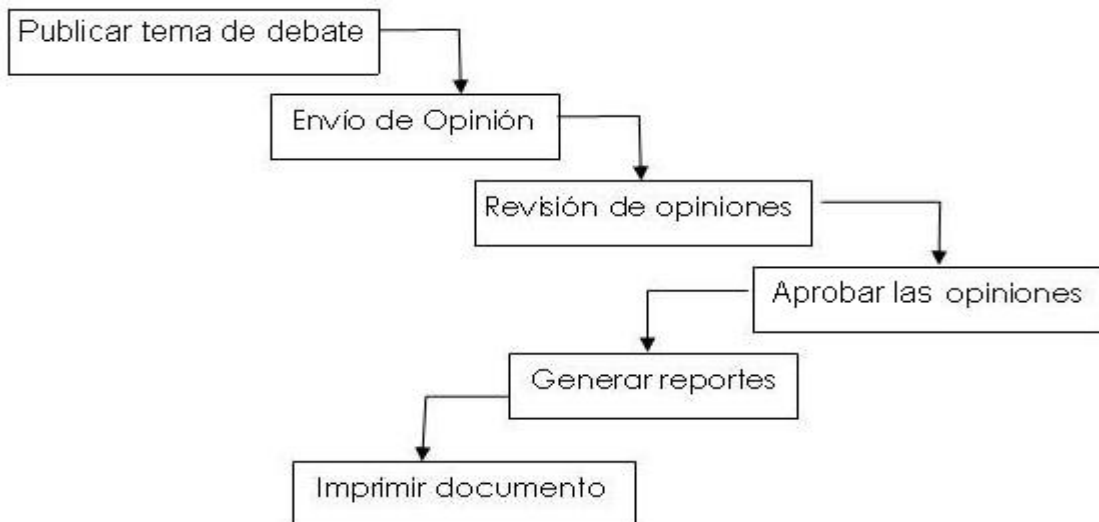


Figura1: Esquema del flujo actual de los procesos del SGO.

2.1.3 Análisis crítico

En la actualidad la UCI cuenta con un SGO dirigido por el PCC, donde los estudiantes, profesores y trabajadores emiten sus opiniones acerca de un determinado tema nacional e internacional que es de interés a la máxima dirección del país. Este sistema presenta dificultades en algunas funcionalidades básicas.

En el proceso de revisión no se actualiza el estado de las opiniones, imposibilitando diferenciar las revisadas del resto y provocando que se vuelvan a revisar. Los encargados de revisar las opiniones están obligados a copiarlas para un documento, por lo que el proceso se torna engorroso y ocasiona demoras. Esto sucede también con las opiniones aprobadas.

La no existencia de reportes estadísticos dificulta obtener resultados concretos. Se hace imposible poder publicar varios temas de debates simultáneamente, esto provoca que el encargado de realizar esta tarea se vea obligado a considerar por terminado un tema para publicar otro.

El sistema no registra las actividades que realizan el editor, revisor y administrador. No posee documentación alguna, por lo que se hace imposible reutilizar su código, no se genera el documento con las opiniones aprobadas con el formato establecido. La interfaz no es amigable por lo que se hace difícil su uso.

Debido a la necesidad de mejorar todas las actividades involucradas en 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 que satisfaga las necesidades de los clientes. Dando lugar al problema: ¿Cómo gestionar de forma eficiente las opiniones emitidas en la UCI?

2.2 Objeto de automatización

2.2.1 Procesos a automatizar

- Administrar los temas de debates, donde se podrán publicar, modificar y terminar.
- Mostrar al usuario los diferentes temas de debates publicados, con el propósito de que pueda seleccionar el tema del cual va a emitir su opinión.
- El proceso de emitir opiniones se realiza por los estudiantes, profesores y trabajadores de la UCI.
- El proceso de revisión de las opiniones se realiza por usuarios con privilegios de editor; donde tiene la posibilidad de corregir la ortografía y archivar aquella opinión que no tenga coherencia. Se genera un listado con las opiniones revisadas.

- Las opiniones revisadas deben ser aprobadas o archivadas por usuarios con privilegios de revisor, se genera un listado de opiniones aprobadas.
- El proceso de recuperar las opiniones archivadas se realiza por el administrador cuando necesitan completar el listado de opiniones aprobadas, la opinión cambia el estado a sin revisar.
- La gestión de los usuarios es realizada por el administrador, que se encarga de asignar y actualizar rol a los usuarios registrados.
- El administrador se encarga de generar documentos con las opiniones aprobadas por temas.
- Generar reportes estadísticos:

El reporte de opiniones por programa, muestra la cantidad de opiniones emitidas, cantidad de opiniones revisadas, aprobadas y archivadas.

El reporte de opiniones por facultades, muestra la cantidad de opiniones emitidas, cantidad de opiniones revisadas, aprobadas y archivadas.

- El administrador se encarga de generar los logs del sistema para mantener un control de las actividades que realizan los usuarios.

2.2.2 Sistema automatizado vinculado con el campo de acción

La UCI cuenta con un Sistema de Gestión de Opiniones que permite registrar las opiniones emitidas por los usuarios.

2.3 Información que se maneja

El Sistema de Gestión de Opiniones es una aplicación que debe ser capaz de generar un documento pdf con un formato establecido arial 10, que contiene un listado de opiniones aprobadas. En la cual se maneja la información referente a las opiniones que emiten estudiantes, profesores y trabajadores de la UCI. Esta información es pública para las personas encargadas de procesar estas opiniones y el Consejo de Estado.

2.4 Propuesta de sistema

La aplicación que se propone facilitará la gestión de las opiniones. Dispondrá de una organización de los eventos a desarrollar en la gestión de las opiniones. Mostrará una interfaz gráfica amigable y fácil de comprender a los usuarios, contendrá un menú que permite que el trabajo sea más organizado.

El sistema requiere autenticación, los usuarios deben existir en el dominio UCI y los permisos dependen del rol asignado, contribuyendo a la seguridad.

La aplicación permitirá registrar, publicar, modificar y terminar los programas. El sistema permitirá que cualquier usuario emita su opinión acerca de cualquier tema publicado. Las opiniones que están registradas pasan a ser revisadas por los encargados de editar. También realiza el cambio de estado de las opiniones, cuando se revisan, aprueban o archivan. El proceso de aprobar se realizará por el encargado de revisar, este podrá ver el texto de las opiniones pero no podrá modificar, solo podrá aceptar y archivar las que tengan algún error. El administrador podrá recuperar las archivadas seleccionando la que desee.

Permitirá que se gestionen los usuarios por parte del administrador, asignando y actualizando rol. Se podrá generar un documento pdf con las opiniones aprobadas por temas, se obtendrán reportes estadísticos y se mostrarán los logs del sistema.

2.4.1 Comparación de otras soluciones con la propuesta

El sistema existente en la UCI solo podía publicar un tema de debate, el encargado de realizar esta tarea se ve obligado a considerar por terminado un tema para publicar otro. La solución que se propone permitirá que sean publicados varios temas de debates, además permitirá que sea terminado cualquier tema en cualquier momento.

El sistema existente en la UCI no era capaz de actualizar el estado de las opiniones, imposibilitando diferenciar las revisadas del resto y provocando que se vuelvan a revisar. En la aplicación que se propone permitirá actualizar el estado de las opiniones: sin revisar, en revisión, revisadas, aprobadas y archivadas.

En el sistema actual para revisar las opiniones los encargados están obligados a copiarlas para un documento, por lo que el proceso se torna engorroso y ocasiona demoras. En el sistema que se propone será capaz de registrar las opiniones revisadas.

La aplicación actual no tiene reportes estadísticos, mientras que la propuesta cuenta con información suficiente para mostrar los reportes estadísticos más importantes.

El sistema actual no tiene forma de controlar las actividades que realizan los usuarios, en la aplicación que se propone se controlan todas las actividades que realizan los usuarios en el sistema. Registrando el usuario y el IP de la PC, la URL donde realizó la actividad, la fecha y la hora.

El sistema existente no posee documentación alguna, por lo que se hace imposible reutilizar su código, la propuesta contará con la documentación necesaria.

2.5 Modelo de dominio

Un Modelo del Dominio captura los tipos más importantes de objetos; los que representan elementos reales o los eventos que suceden en el entorno en el que trabaja el sistema.

¿Por qué Modelo del dominio?

Debido a la poca estructuración de los procesos del negocio que tienen que ver con el objeto de estudio y para poder entender el contexto en que se emplaza el sistema se necesita definir conceptos que se agrupan en un modelo de dominio, donde el objetivo fundamental de esta modelación es la comprensión y descripción de las clases más importantes en el sistema.

2.5.1 Glosario de términos

Son los términos comunes o conceptos que se utilizan para describir el sistema.

Personas: Estudiantes, profesores y trabajadores que pueden pertenecer al PCC, UJC, CTC y FEU.

Opiniones: Criterio o parecer que puede ser emitido por cualquier persona sobre un tema determinado.

Documento Final Opiniones: Es el documento que se organiza con un listado de opiniones aprobadas, que deben ser impresas para ser llevadas al Consejo de Estado.

Reportes: Informaciones que se organiza a partir de las opiniones y los programas, mostrando cantidades.

Encargado de Administrar: Persona encargada de gestionar todos los eventos relacionados con las opiniones, programas y usuarios.

Encargado de Editar: Persona encargada de editar las opiniones.

Encargado de Revisar: Persona encargada de revisar las opiniones.

Programas: Son los temas de debates orientados por el Consejo de Estado.

2.5.2 Modelo de dominio

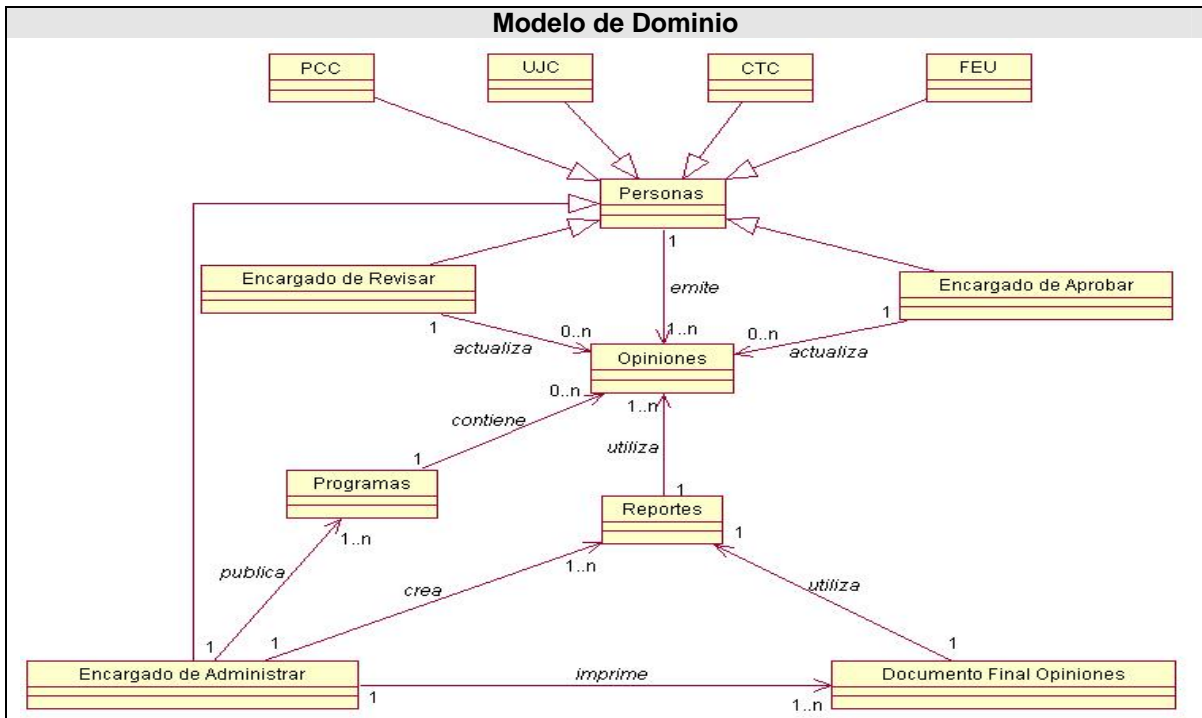


Figura 2: Modelo de dominio.

2.5.3 Reglas del Negocio

Las reglas del negocio permiten definir los límites lógicos de las aplicaciones. En ellas las aplicaciones reflejan las restricciones que existen, evitando la realización de acciones no válidas. A continuación se enumeran las reglas definidas para el sistema:

1. La dirección del PCC de la Universidad determina el programa a publicar.
2. Los encargados de editar no deben cambiar el contexto de las opiniones durante el proceso de revisión.
3. La dirección del PCC de la Universidad es responsable de aprobar las opiniones.
4. El documento pdf que se confecciona con las opiniones revisadas debe tener formato: fuente arial 10, interlineado 1.5, alineación justificada.

2.6 Especificación de los requerimientos de software

Los requerimientos del software constituyen las condiciones indispensables que debe poseer cualquier sistema para su correcto funcionamiento.

2.6.1 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, expresan una especificación detallada de las características del sistema. No alteran la funcionalidad del producto, esto quiere decir que se mantienen invariables sin importarle con que propiedades o cualidades se relacionen.

1. Autenticar.
 - 1.1 Verificar que el usuario pertenece al dominio UCI.
 - 1.2 Verificar el nivel de acceso de los usuarios.
2. Gestionar programas de debate.
 - 2.1 Adicionar y/o publicar el tema.
 - 2.2 Mostrar un listado con los programas de debates no publicados.
 - 2.3 Publicar tema.
 - 2.4 Modificar tema
 - 2.5 Mostrar un listado con los programas de debates publicados.
 - 2.6 Terminar tema.
3. Adicionar una opinión dado un programa.
 - 3.1 Emitir opinión.
4. Cambiar el estado de las opiniones: sin revisar, en revisión, revisada, aprobada y archivada.
5. Revisar opinión.
 - 5.1 Mostrar listado de temas de debates con opiniones sin revisar.
 - 5.2 Modificar opinión.
 - 5.3 Cambiar el estado de la opinión.
6. Aprobar opiniones.
 - 6.1 Mostrar listado de las opiniones revisadas por temas de debates.
 - 6.2 Archivar determinada opinión.
7. Generar documento.

- 7.1 Mostrar listado de temas de debates con opiniones aprobadas.
- 7.2 Generar el listado con las opiniones aprobadas por temas.
- 7.3 Generar documento con las opiniones aprobadas por tema.
- 8. Recuperar opiniones archivadas.
 - 8.1 Mostrar listado de los temas de debates con opiniones archivadas.
 - 8.2 Mostrar listado de opiniones archivadas por temas.
- 9. Gestionar usuario.
 - 8.1 Asignar rol al usuario.
 - 8.2 Actualizar rol al usuario.
 - 8.3 Mostrar listado de usuarios.
- 10. Generar reportes dado un programa.
 - 10.1 Mostrar cantidad general de opiniones emitidas, revisadas, aprobadas, archivadas.
 - 10.2 Mostrar cantidades de opiniones emitidas, revisadas, aprobadas y archivadas por áreas.
- 11. Registrar usuario, fecha, hora, acción y el IP de la PC desde donde se gestionan programas, se agrega la opinión, se gestionan las opiniones, se gestiona usuarios, se realizan reportes.
- 12. Log de sistema.
 - 12.1 Mostrar acciones realizadas por los usuarios.

2.6.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el sistema debe tener; características que hacen al producto atractivo, usable, rápido o confiable.

- Apariencia o interfaz externa: el sistema debe poseer apariencia profesional, tiene que ser rápido, muy legible y simple de usar con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones. No debe contener muchas imágenes que demoren las respuestas al usuario. Los eventos se muestran en un menú sencillo para el rápido desenvolvimiento de los usuarios. Los usuarios pocos familiarizados con el sistema deben percibir sin problemas las salidas del mismo.

- Usabilidad: el sistema será utilizado por todo tipo de usuarios, incluyendo personas con pocos conocimientos en el uso de las computadoras. Permitirá a los usuarios tener control en todo momento del sitio web, evitando la sobrecarga de información.
- Rendimiento: el sistema propuesto debe garantizar que la respuesta a solicitudes de usuarios sea en un período de tiempo mínimo, adecuado a la rapidez con que los encargados gestionen las opiniones. La disponibilidad de trabajo en red contra el servidor es constante. El sistema deberá ser estable y confiable.
- Soporte: se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra.
- Portabilidad: el sistema debe correr sobre una plataforma Web, codificada en php. Debido al sistema gestor de bases de datos que se utilizará PostgreSQL, el sistema será compatible con los sistemas operativos Windows, Linux y Unix.
- Seguridad: el sistema se debe encargar de controlar los diferentes niveles de acceso, e identificar al usuario antes de que pueda realizar cualquier acción. Debe garantizar que la información sea vista únicamente por quien tiene derecho a verla. Se hacen validaciones de la información tanto en el cliente como en el servidor.

El sistema estará montado en el nodo central de la UCI que brinda servicios de hosting, garantizando un alto grado de seguridad. Se encargan de mantener un control estricto del responsable y solo permiten el acceso al personal autorizado. Protegen contra descargas y picos eléctricos, mantienen cuidadosa protección contra la corrupción y estados inconsistentes, tienen políticas de salvaguarda que permiten hacer un seguimiento de cualquier problema que se presente.

PostgreSQL se encarga de hacer copias de seguridad de las bases de datos del Sistema, manipula la concurrencia. Tiene la característica de recuperación puntual, permitiendo ejecutar una restauración completa de los datos desde los archivos de registro de transacciones que se actualizan de manera automática y continua. Además de tener puntos de salvaguarda: donde partes específicas de una transacción sobre la BD, sean objeto de un roll back sin necesidad de abortar toda la operación.

Cada base de datos será accedida por sólo una aplicación. Las bases de datos podrán replicarse a otras para efectuar mecanismos de balance de carga y resguardo de los datos.[15]

- **Políticos-Culturales:** la modificación del funcionamiento establecido en los requerimientos será realizada por la Dirección de Informatización.
- **Requerimientos Legales:** el sistema cumple con las normas y requisitos legales establecidos en la UCI. La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL. La publicación del sitio debe ser aceptada por la Universidad.
- **Requerimientos de confiabilidad:** la información será mostrada solo a los encargados de procesarla y al Consejo de Estado.
- **Requerimiento de Software:** para el funcionamiento del sistema en el servidor será necesario el SO Linux o Unix, en sus versiones de servidores. El funcionamiento del sistema en las terminales cliente requieren de el SO Windows 95 o superior, Linux o Unix.
- **Requerimiento de Hardware:** el producto requiere de una computadora que haga función de un servidor, la cual debe cumplir con requerimientos mínimos de una PC, con procesador Pentium II o superior. Para que los usuarios puedan utilizar el sistema, es necesario que cada uno tenga una computadora conectada a una red, teniendo acceso a un navegador.
- **Ayuda y documentación en línea:** el sistema mostrará ayuda en línea para que los usuarios con poca experiencia acudan al mismo mediante una documentación apropiada.

2.7 Casos de uso

2.7.1 Definición de los actores

Podríamos definir un actor como el rol que asume una persona, sistema o entidad que interactúa con el sistema que se está construyendo. Tiene la propiedad de ser externo al sistema y representan a los usuarios del sistema.

Tabla 1: Actores y justificación.

Actores	Justificación
Usuario	Representan a los (estudiantes, profesores o trabajadores) que interactúan con el sistema, pueden ser administrador, editor, revisor o usuario normal.
Administrador	Representa al administrador del sistema, tiene acceso a todas las funcionalidades que se realizan. El Administrador Hereda de los

	actores Editor y Revisor.
Editor	Revisa y/o archiva las opiniones que están sin revisar en el sistema. El Editor hereda del actor Usuario.
Revisor	Aprueba y/o archiva las opiniones que están revisadas. El Revisor hereda del actor Usuario.

2.7.2 Definición de los casos de uso

Un caso de uso es la descripción de un conjunto de acciones que un sistema ejecuta y que produce un determinado resultado que es de interés para un actor particular. Los casos de uso permiten modelar completa y consistentemente el sistema. A continuación los casos de uso definidos.

Tabla 2: Caso de uso autenticar.

CU-1	Autenticar
Actor	Usuario (inicia)
Descripción	El caso de uso se inicia cuando el usuario introduce los datos, estos son verificados con Ldap para probar su validez y el sistema comprueba si el usuario tiene asignado algún rol. El caso de uso termina cuando el usuario obtiene los permisos y accede al sistema.
Referencia	RF1.

Tabla 3: Caso de uso gestionar programas.

CU-2	Gestionar programas
Actor	Administrador (inicia).
Descripción	El caso de uso inicia cuando el Administrador requiere agregar, publicar, modificarlo o terminar un programa. El sistema actualiza los datos según la acción realizada. El caso de uso termina cuando se agrega, publica, modifica o elimina un usuario.
Referencia	RF 2, RF 11.

Tabla 4: Caso de uso insertar opinión.

CU-3	Registrar Opinión
Actor	Usuario (inicia)
Descripción	El caso de uso se inicia cuando el usuario escoge el tema del cual va a opinar, introduce en el sistema una o varias opiniones acerca de un

	programa determinado. El caso de uso termina cuando el usuario envía la opinión.
Referencia	RF 3, RF 11.

Tabla 5: Caso de uso revisar opinión.

CU-4	Revisar opinión
Actor	Editor (Inicia)
Descripción	El caso de uso inicia cuando el Editor selecciona revisar la opinión, el sistema permite revisar, archivar o cancelar. Almacena y actualiza los datos. El caso de uso termina cuando el editor termina de revisar las opiniones.
Referencia	RF 5, RF 4, RF 11.

Tabla 6: Caso de uso Aprobar opinión.

CU-5	Aprobar opinión
Actor	Revisor (Inicia)
Descripción	El caso de uso inicia cuando el Revisor selecciona aprobar opiniones, el sistema permite aprobar, archivar y cancelar. Verifica y almacena los datos. El caso de uso termina cuando son aprobadas las opiniones.
Referencia	RF 6, RF 4, RF 11.

Tabla 7: Caso de uso generar documento.

CU-6	Generar Documento
Actor	Administrador
Descripción	El Administrador selecciona generar documento, el sistema muestra un listado de programas para seleccionar uno, le permite listar de ese programa el listado de las opiniones aprobadas. El caso de uso termina cuando se genera el documento pdf.
Referencia	RF 7, RF 11.

Tabla 8: Caso de uso recuperar archivadas.

CU-7	Recuperar archivadas
Actor	Administrador (Inicia)
Descripción	El Administrador selecciona recuperar archivadas, el sistema muestra un listado de opiniones archivadas y permite actualizar el estado de la opinión,

	verifica y almacena los datos.
Referencia	RF 8, RF 4, RF 11.

Tabla 9: Caso de uso gestionar usuarios.

CU-8	Gestionar Usuarios
Actor	Administrador (inicia).
Descripción	El caso de uso inicia cuando el Administrador asigna o actualiza rol a un usuario. Verifica que tengan el formato correcto, almacena y actualiza los datos. El caso de uso termina cuando se asigna o actualiza rol a un usuario.
Referencia	RF 9, RF 11.

Tabla 10: Caso de uso generar reportes.

CU-9	Generar Reportes
Actor	Administrador (inicia).
Descripción	El caso de uso inicia cuando el Administrador solicita los reportes estadísticos de un determinado programa. El caso de uso concluye cuando se muestran los reportes.
Referencia	RF 10

Tabla 11: Caso de uso logs del sistema.

CU-10	Logs del sistema
Actor	Administrador (inicia).
Descripción	El caso de uso inicia cuando el administrador solicita los logs de actividades del sistema. El caso de uso termina cuando el sistema muestra los logs.
Referencia	RF 12, RF 11.

2.7.3 Diagrama de casos de uso del sistema

El modelo de los casos de uso del sistema representa un esquema donde se recogen las funcionalidades del negocio que se automatiza y determina cómo será utilizado desde la perspectiva del usuario (Actor) pues se construye sobre la base de sus necesidades. A través de este modelo se puede establecer comunicación, más o menos fluida en dependencia de la claridad del mismo, con los usuarios finales y clientes expertos del sistema en desarrollo, e informarles acerca de su comportamiento futuro. [14]

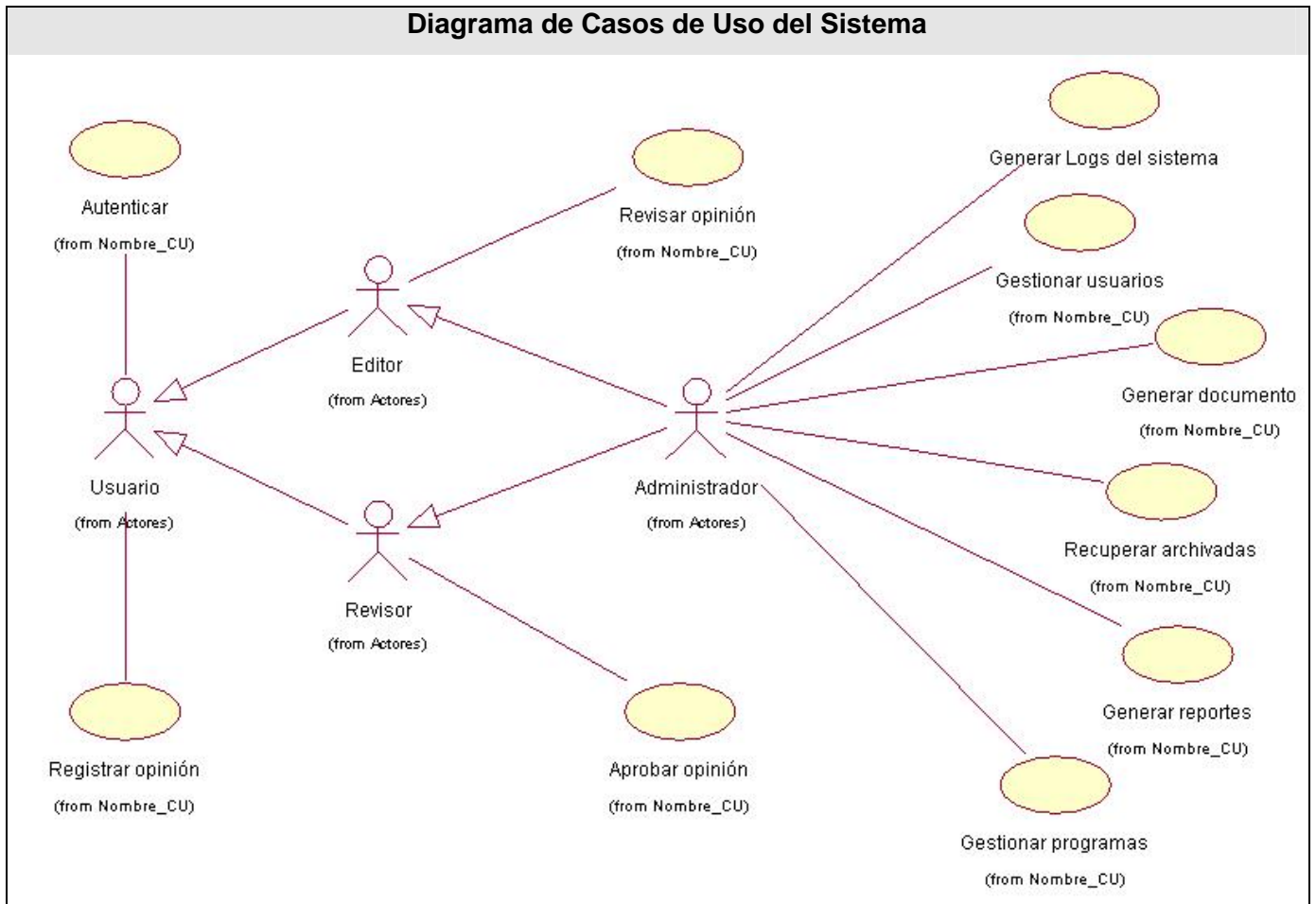


Figura 3: Diagrama de casos de uso del sistema.

2.7.4 Descripciones de casos de uso

Se procede mediante la expansión de los casos de uso a la descripción, refinando los requerimientos funcionales, detallando los flujos del usuario y el sistema. Ver anexo 1.

Conclusiones

Este capítulo ha brindado una clara definición de los requerimientos que debe cumplir el sistema. Se determinaron los actores y los casos de uso del sistema. El empleo del diagrama de casos de usos del sistema propuesto permitió una adecuada modelación de los requerimientos, demostrando su importancia en esta etapa. Se ganó claridad en cuanto a la concesión del sistema a construir y se sentaron las bases para el análisis y diseño.

Capítulo 3: Análisis y diseño del sistema

Introducción

En el presente capítulo se procede al análisis y diseño del sistema. En el análisis se realiza la definición del modelo de análisis y el modelo de clases de análisis. En el diseño se realizan los diagramas de interacción, descripción de las clases, diseño de la base de datos en el diagrama de entidad relación y descripción de tablas, utilizando para el modelado el Lenguaje Unificado de Modelación. Además definiciones de diseño que se apliquen, tratamiento de errores, seguridad, interfaz y concepción de la ayuda. A través de esta fase, se toman decisiones estratégicas y tácticas para cumplir los requerimientos funcionales del sistema.

3.1 Modelo de análisis

El modelo de análisis se basa en un modelo de objetos conceptual, describe la realización de los requerimientos funcionales en el lenguaje técnico del desarrollador, no precisa como se implementa la solución. Se compone por paquete de análisis, que a su vez contienen las clases del análisis y realización de casos de uso del análisis, objetos organizados en paquetes que colaboran.

3.1.1 Modelo de clases de análisis

Las clases de análisis se centran en los requerimientos funcionales, son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Las operaciones y atributos son poco detallados, siempre encajan en tres estereotipos básicos: Clase Interfaz, Clase Controladora y Clase Entidad; se utilizan las asociaciones entre ellas, la multiplicidad o cardinalidad para cada asociación. Ver anexo 2.

3.2 Modelo de diseño

El diseño es un refinamiento del análisis, un modelo físico que crea una entrada apropiada y un punto de partida para la implementación, que tiene en cuenta los requerimientos no funcionales. Está compuesto por varios paquetes de diseños, que a su vez están integrados por clases del diseño, especificación de los casos de usos del diseño e interfaz.

El propósito de esta actividad, es transformar el modelo de clases lógico, que proviene del análisis, en un modelo de clases de diseño donde encuentren soporte todos los requerimientos.

3.2.1 Diagramas de Interacción

Los diagramas de interacción muestran las interacciones entre objetos, mediante transferencia de mensajes entre objetos o subsistemas. En UML los diagramas de interacción pueden representarse a través de los Diagramas de Colaboración y/o los Diagramas de Secuencia.

3.2.1.1 Diagrama de Colaboración

El diagrama de colaboración destaca la organización de los objetos que participan en una interacción y describe cada realización de los cursos alternos que pueden tomar los casos de uso definidos. En estos diagramas se destaca la relación estructural entre los objetos que interactúan, tienen caminos que indican como se enlaza un objeto a otro y los mensajes cuentan con el número de secuencia, para indicar el orden temporal de un mensaje.

La realización del diagrama de colaboración corresponde a cada caso de uso, en algunos casos el actor demanda sólo una operación. Puede dar la impresión de que hay un diagrama de colaboración de sistema para cada operación, pero esto no quiere decir que sea así, sino que cada diagrama de colaboración de sistema mostrará los eventos que envía un actor al sistema para un caso de uso, ver anexo 3.

3.2.1.2 Diagramas de Secuencia

El Diagrama de Secuencia destaca el orden temporal de los mensajes, es una forma alternativa al diagrama de colaboración de mostrar un escenario. Muestra la comunicación entre objetos en una secuencia de tiempo, los objetos tienen líneas de vida que representan existencia a lo largo de un período de tiempo y presentan un foco de control que representa el período de tiempo durante el cual un objeto ejecuta una acción.

3.2.2 Diagrama de clases del diseño Web

El diagrama de clases del diseño es una representación más concreta que el diagrama de clases del análisis. Representa la parte estática del sistema, las clases y sus relaciones. El diagrama del diseño Web del sistema para diseño orientado a objetos se obtiene como resultado del refinamiento del modelo conceptual y se basa fundamentalmente en los diagramas de interacción. Es una abstracción a la implementación del sistema. Los diagramas de clases se pueden ver en el anexo 4.

3.2.3 Descripción de las clases del diseño

En la descripción de las clases del diseño, se utilizan la sintaxis del lenguaje de programación empleado. Se especifica la visibilidad de los atributos y operaciones, referencias entre objetos que tienen significado en la implementación. Se especifican los métodos, manteniendo correspondencia directa con los métodos en la implementación, ver anexo 5.

3.3 Diseño de la Base de Datos

El diseño de la base de datos: tiene el propósito de identificar las clases de diseño que son persistentes en una base de datos y diseñar la estructura correspondiente de la base de datos. Las tablas de la base de datos se originan a partir del modelo conceptual. La base de datos del sistema se utiliza para almacenar información referente a los programas, las opiniones y los usuarios. Descripción de las tablas de las Base de datos, ver anexo 6.

3.3.1 Diagrama Entidad Relación

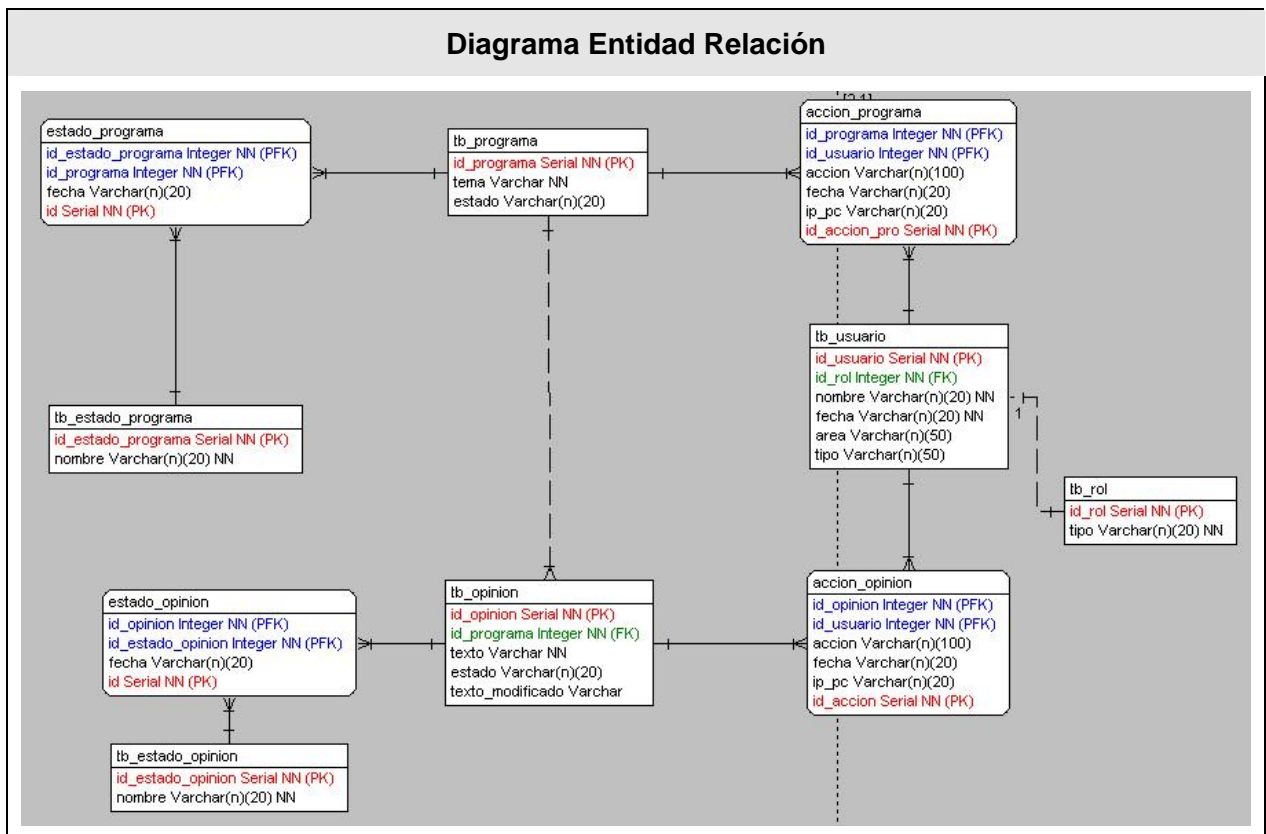


Figura 38: Diagrama entidad relación de la base de datos.

3.4 Definiciones de diseño que se apliquen

Los sistemas de computación cada vez hacen más evidente la necesidad de un diseño de interfaz eficaz que facilite la comunicación con el usuario. Los Analistas de Sistemas, no solo deben considerar que debe hacer el sistema y como se va a realizar, sino también deben centrarse como lograr un producto que cumpla con los requerimientos y que se adapte a los usuarios.

La interfaz de la aplicación ha de ser uniforme; 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 los tipos de letra. Evitar el uso de términos técnicos, propiciando el uso de términos populares.

La navegación debe ser rápida, por lo que debe ser eliminado la presencia de imágenes y funciones innecesarias. Posee un menú que organiza los vínculos a todas las páginas.

3.5 Tratamiento de errores

Se tomarán en cuenta los errores cometido por usuarios durante la interacción con el sistema. Tratando en todo momento de minimizar la posibilidad de que ocurran equivocaciones de este tipo, aprovechando las posibilidades de la interfaz gráfica; se evitará que el usuario asuma un papel más activo en la captura de información, para lo cual se le dará la opción de seleccionar la información a través de vínculos o menú de selección, lo cual facilitará la entrada de datos.

La información que requiera ser introducida por el usuario en el sistema, contará con la validación de la información mediante funciones Javascripts que garantizan que sean válidos y se comprueba que el cuadro de texto no esté vacío si es obligatorio llenarlo. 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 de diálogo desaparecerá y continuará la ejecución de la aplicación.

3.6 Seguridad

En la aplicación se trabaja con variables de sesión; la cual permite restringir el acceso de los usuarios no autorizados a determinada información presente en el sitio. De esta forma, se limita, se controla y se protege, de una manera racional, los datos a los cuales se acceden desde distintos niveles de privilegios. El SGBD PostgreSQL, se encarga de hacer copias de las bases de datos del Sistema lo que permite que la seguridad de la información sea elevada.

3.7 Interfaz

La confección de una interfaz amigable, sencilla y consistente favorece la usabilidad de la aplicación haciéndola entendible para cualquier tipo de usuario. La estabilidad y uniformidad del diseño para ubicar al usuario y hacerlo sentir parte del mismo. El uso de una misma tipografía, forma y estilo en todas sus partes. Se consideró el formato de imágenes de compresión favorables (gif y jpg).

3.8 Concepción de la ayuda

La aplicación contará con nombres sugerentes en todas las páginas que permitirá que el usuario pueda orientarse dentro de las diferentes funcionalidades con que cuenta la aplicación. Para facilitar la manipulación y funcionamiento del sistema, este cuenta con una ayuda disponible en el menú; presenta descripciones breves de las funcionalidades de cada operación, mediante preguntas frecuentes.

Conclusiones

En el capítulo se plantearon definiciones del modelo de análisis y modelo del diseño. Se realizaron diagramas de clases de análisis, diagrama de secuencia, diagrama de clases del diseño, diagrama Entidad Relación de la BD, diagrama de despliegue y diagrama de componentes. Se describieron las clases utilizadas en el diseño y las tablas de la base de datos. También se explicó la definición del diseño que se aplicó, la forma de tratar los errores, se definió la seguridad del sistema, la interfaz y la concepción de la ayuda.

Capítulo 4: Implementación

Introducción

En el presente capítulo se muestran las definiciones de los diferentes modelos utilizados en la implementación para la solución del problema. Se realizan los diagramas de despliegue y de componentes.

4.1 Modelo de implementación

El modelo de implementación describe los resultados del modelo de diseño, como se implementan en términos de componentes y como se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Los diagramas de despliegue y componentes conforman lo que se conoce como un modelo de implementación, al describir los componentes a construir, su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

4.1.2 Diagrama de despliegue

El Diagrama de Despliegue es un modelo de objetos que describe la distribución física del sistema en términos de como se distribuye la funcionalidad entre los nodos de cómputo, muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software.

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad de computación de algún tipo, desde un sensor a un mainframe. Las instancias de componentes software pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz). [16]

La aplicación estará ubicada en los servidores nodos de la UCI, donde tendrán un control total sobre la aplicación: para el servidor de las páginas Web estará instalado Apache y en el servidor de la base de datos, PostgreSQL.

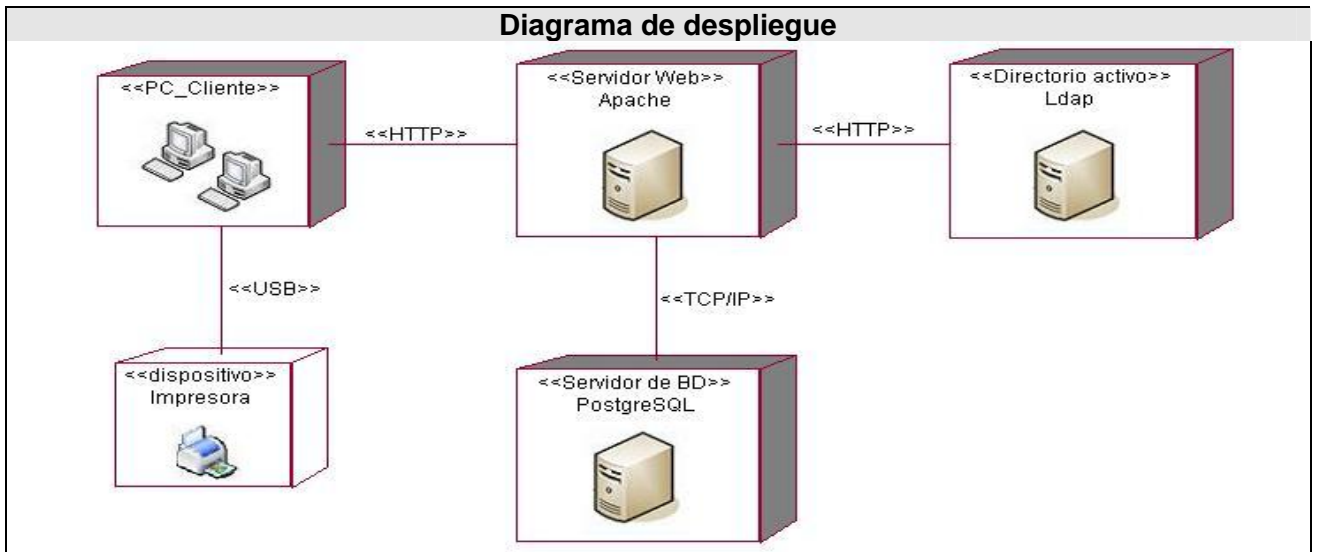


Figura 39: Diagrama de despliegue.

4.1.3 Diagrama de componentes

El diagrama de componentes se representa como un grafo de componentes software unidos por medio de relaciones de dependencia. Muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. Modela la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean éstos componentes de código fuente, librerías, binarios o ejecutables. Cada diagrama describe un apartado del sistema.

Lo que distingue a un diagrama de componentes de otros diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones, también puede contener paquetes utilizados para agrupar elementos del modelo.

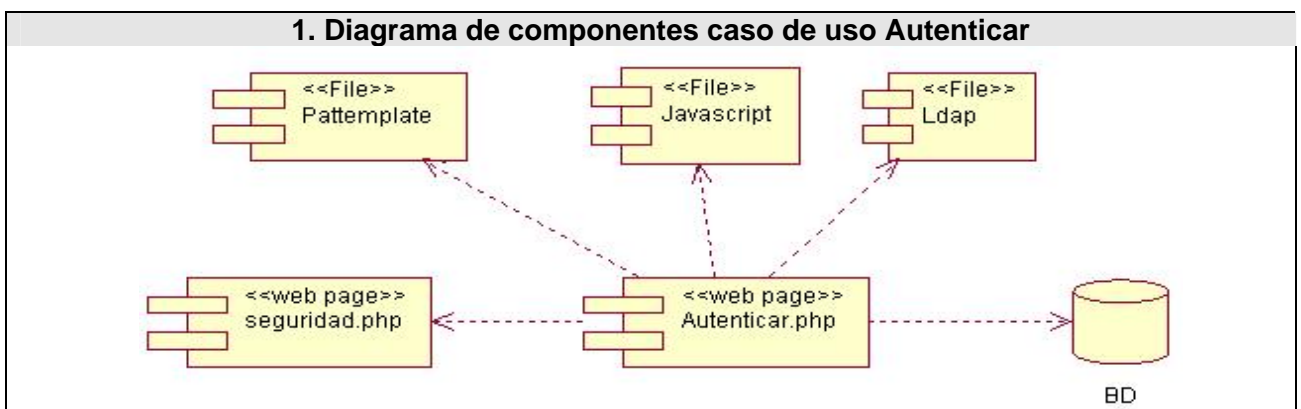


Figura 40: Diagrama de componente caso de uso autenticar.

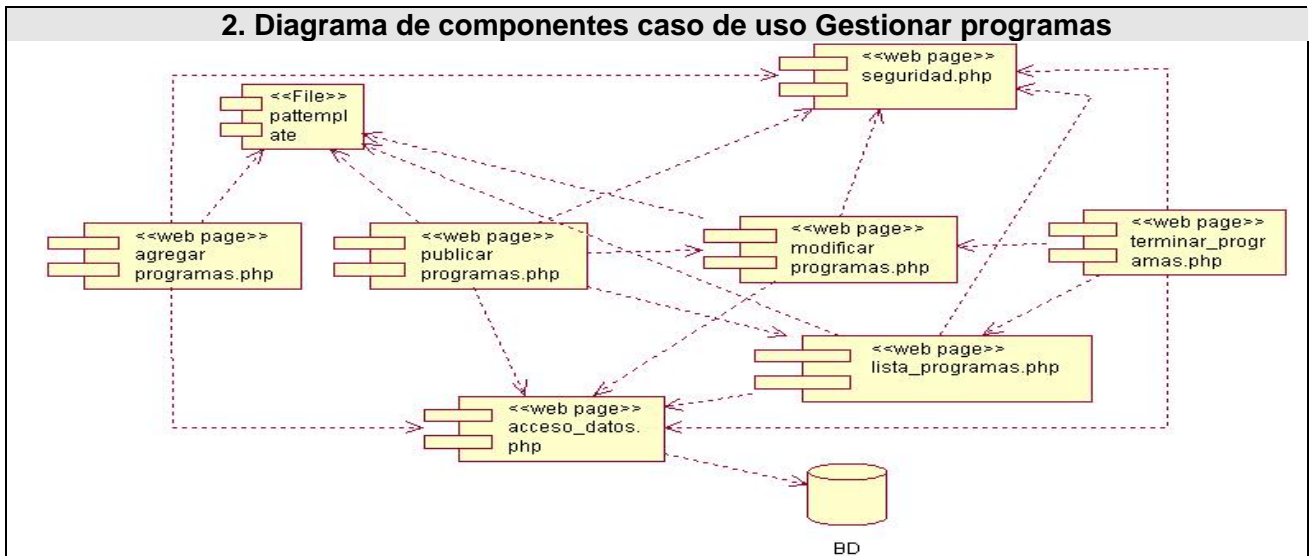


Figura 41: Diagrama de componente caso de uso gestionar programas.

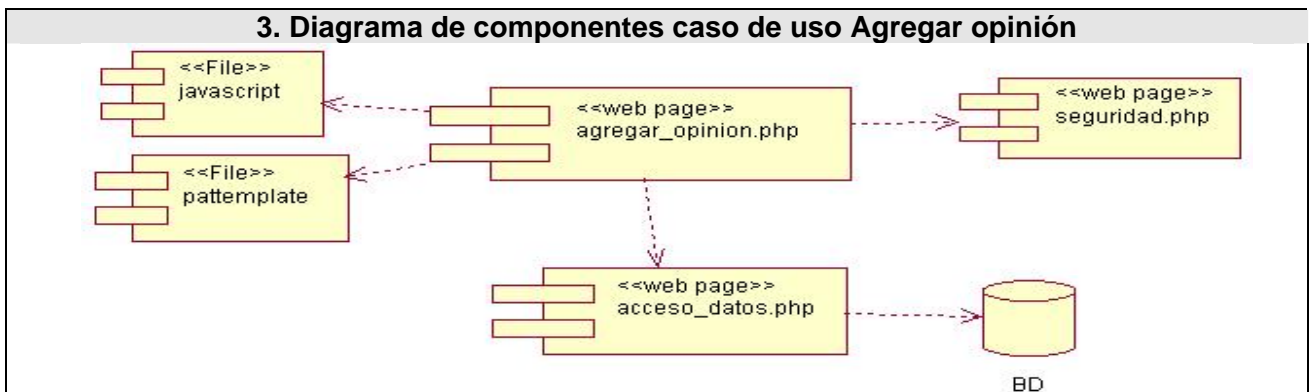


Figura 42: Diagrama de componente caso de uso insertar opinión.

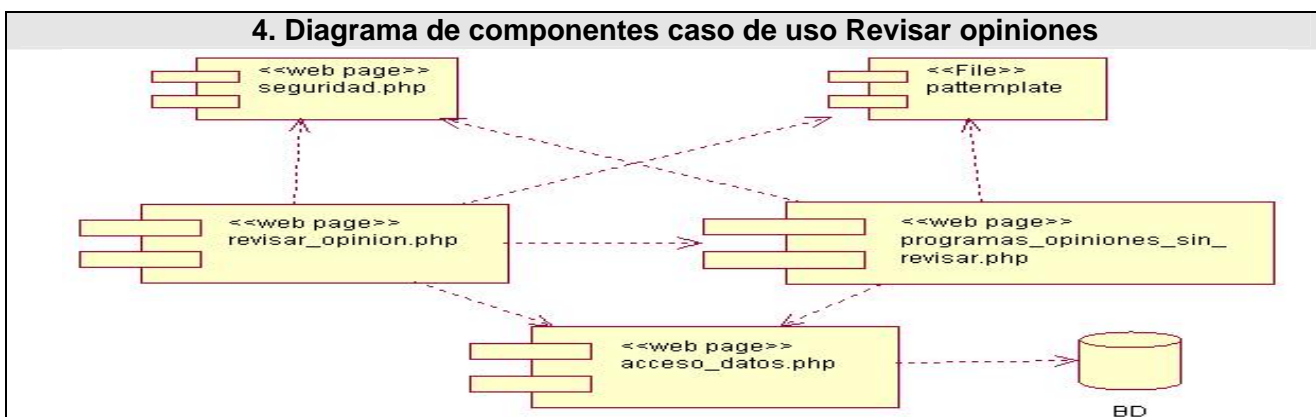


Figura 43: Diagrama de componente caso de uso revisar opiniones.

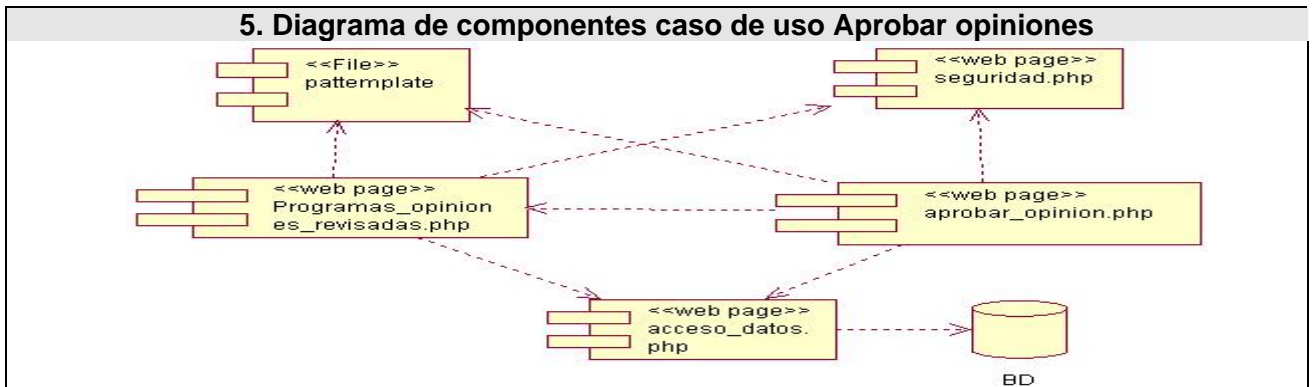


Figura 44: Diagrama de componente caso de uso aprobar opiniones.

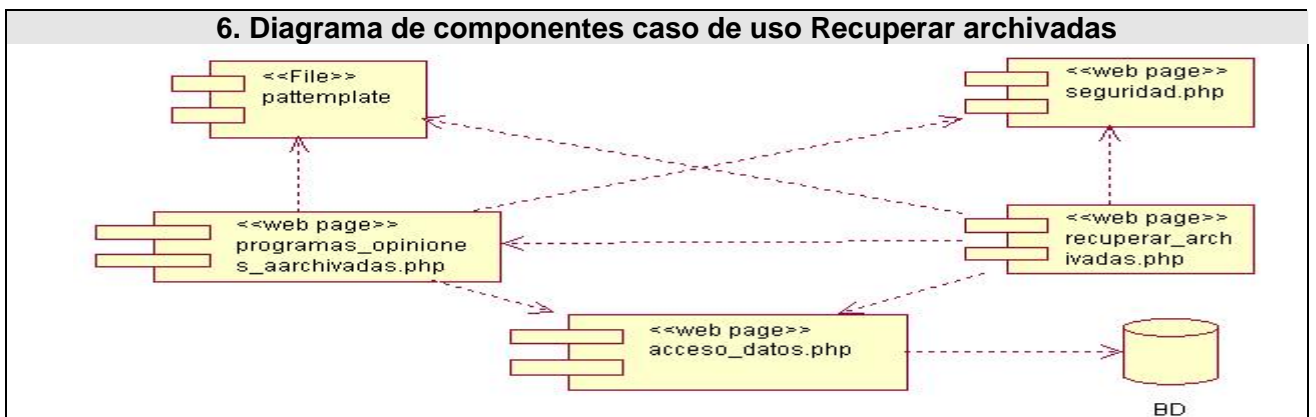


Figura 45: Diagrama de componente caso de uso recuperar archivadas.

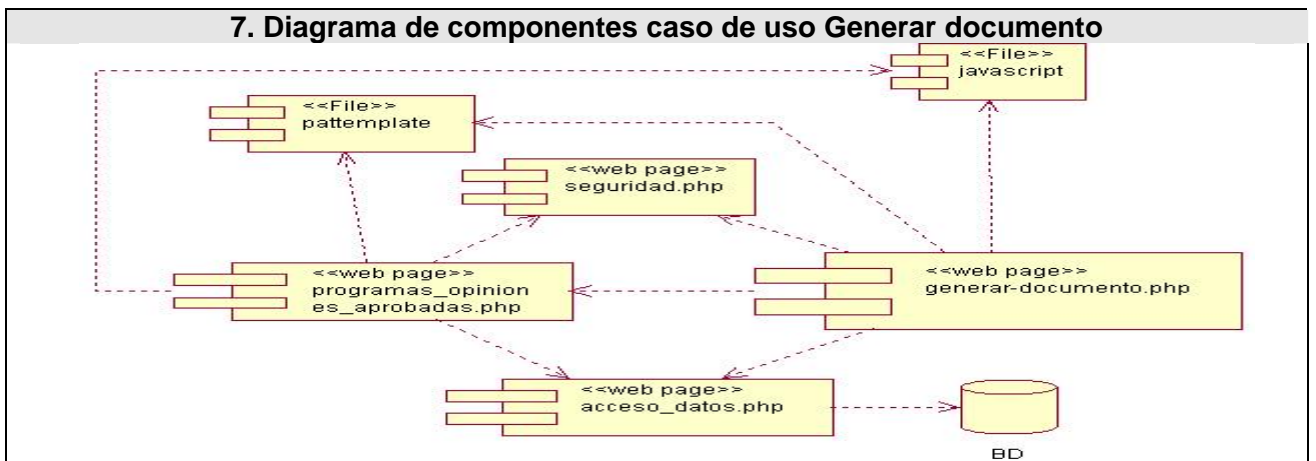


Figura 46: Diagrama de componente caso de uso gestionar documento.

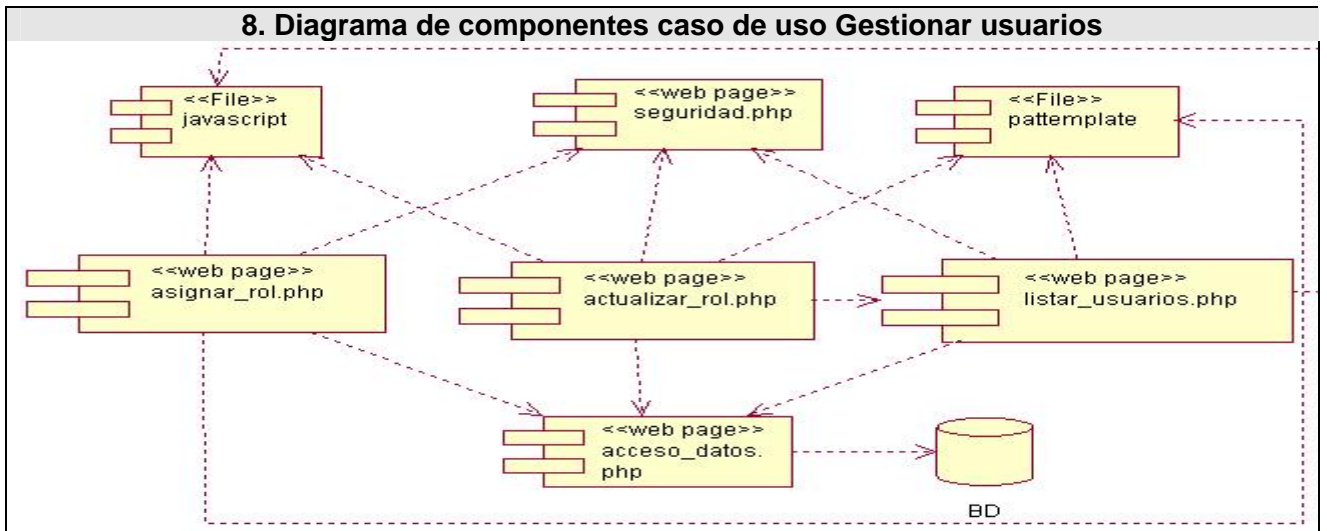


Figura 47: Diagrama de componente caso de uso gestionar usuarios.

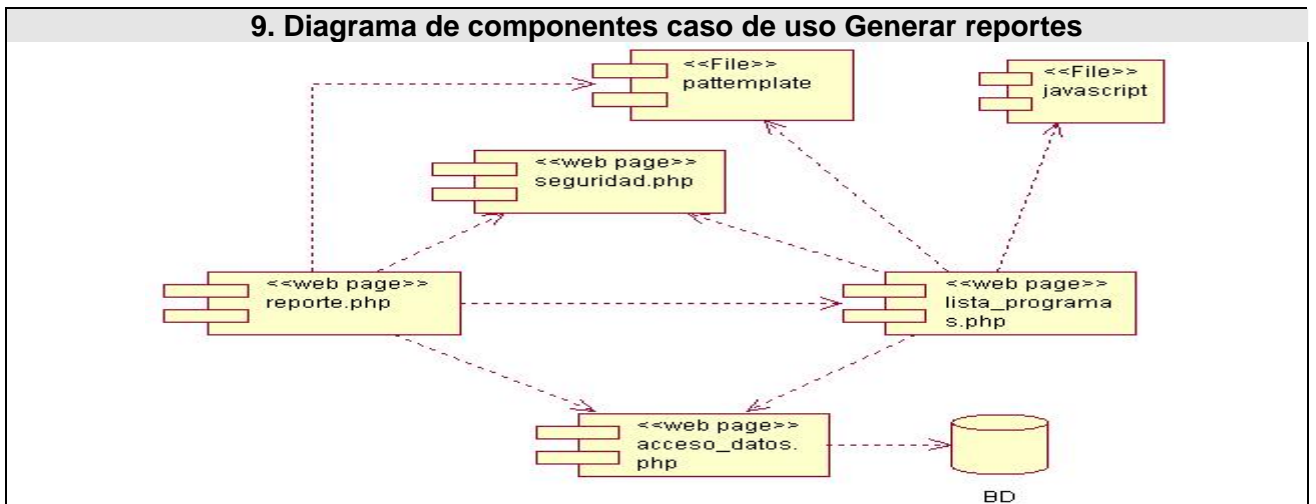


Figura 48: Diagrama de componente caso de uso generar reportes.

Conclusiones

En el capítulo se plantea la definición de modelo de implementación. Se realizaron diagramas, el de despliegue y los de componentes.

Conclusiones

El punto de partida para la investigación, lo constituyó el estudio realizado sobre el sistema de opiniones que existe en la UCI y las deficiencias mostradas a la hora de gestionar opiniones. Teniendo en cuenta que los objetivos trazados y las tareas propuestas quedaron satisfechas con la realización del análisis, diseño e implementación de la aplicación Web.

Se dio solución a los problemas existentes en la gestión de opiniones; se actualizan los estados de las opiniones, permite publicar varios temas de debates simultáneamente, genera documentos pdf, se registran las actividades que realizan los usuarios que interactúan con el sistema y muestra reportes estadísticos por tema de debate.

La puesta en marcha de este sistema permitirá ofrecer mejoras en el proceso de emisión y revisión de las opiniones que se gestionan en la UCI, garantizando de forma rápida y eficiente el control, desempeño y calidad de los servicios que presta.

Recomendaciones

En busca de un mayor enriquecimiento funcional de la aplicación propuesta, se recomienda:

- Perfeccionar el manual de usuario.
- Realizar pruebas a la aplicación.
- Graficar los reportes estadísticos que se muestra en la aplicación.
- Agregarle un corrector ortográfico.
- Continuar con la investigación para garantizar mejores versiones del sistema.

Bibliografías

Referencias Bibliográficas

- [1] "Glosario de términos", CampusFormación, 2004. Disponible en: <<http://www.campusformacion.com/glosario.asp>> [Fecha de consultada 3 noviembre 2006].
- [2] Thompson, Ivan."Tipos de Encuesta", Promonegocios.net, Julio 2006. Disponible en: <<http://www.promonegocios.net/mercadotecnia/encuestas-tipos.html>> [Fecha de consultada 3 noviembre 2006].
- [3] *Aplicación Web*. Disponible en: <http://es.wikipedia.org/wiki/Aplicacion_web> [Fecha de consultada 6 noviembre 2006].
- [4] *HTML*. Disponible en: <<http://es.wikipedia.org/wiki/HTML>> [Fecha de consultada 15 enero 2007].
- [5] *JavaScript*. Disponible en: <<http://es.wikipedia.org/wiki/JavaScript>> [Fecha de consultada 15 enero 2007].
- [6] Román, M. Sitio Web de Soporte a Clientes de la empresa: Telecable Internacional. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico "José Antonio Echeverría" (ISPJ), Ciudad de la Habana, junio 2004. [Fecha de consultada 23 febrero 2007].
- [7] Cantero, J. *Un vistazo a PHP5 [I]*, julio 2004. Disponible en: <<http://libertonia.escomposlinux.org/story/2004/7/15/115328/134>> [Fecha de consultada 23 febrero 2007].
- [8] *Sistema de gestión de base de datos*. Disponible en: <http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos> [Fecha de consultada 25 febrero 2007]
- [9] Pecos,D. *PostGreSQL vs. MySQL*. Disponible en: <http://www.netpecos.org/docs/mysql_postgres/index.html> [Fecha de consultada 25 febrero 2007].
- [10] *Rational Siute*, IBM Rational Software. [Fecha de consultada 25 febrero 2007].
Disponible en:<<http://www.neovalia.es/imagenes/ficheros/ibm/RATIONAL.doc>>
- [11] *Novedades de DreamWeaver 8*, Todo Dreamweaver, 2005. Disponible en: <<http://www.todo-dreamweaver.com/tutorial-reamweaver/otros/novedades-dreamweaver-ocho.html>> [Fecha de consultada 25 febrero 2007].

[12] *Excelente y robusta plataforma de desarrollo PHP*, Zend Studio Client 3.0.2. Madrid, España. Disponible en: < <http://software.elpais.com/ie/21058-Zend-Studio-Client> > [Fecha de consultada 25 febrero 2007].

[13] *PGAdmin III*. [Fecha de consultada 25 febrero 2007]. Disponible en: <http://www.guia-ubuntu.org/index.php?title=PgAdmin_III >

[14] *Jacobson Ivar, Booch Grady, Rumbaugh James. El proceso unificado de desarrollo del software*. Addison Wesley editorial de Pearson Education.sa, 1999. [Fecha de consultada 25 mayo 2007].

[15] Dirección de Informatización, *Arquitectura para los sistemas que conforma la red universitaria*. 9 mayo 2007 [Consultada 25 mayo 2007]

[16] *Diagrama de Despliegue*, febrero 2007 [Fecha de consultada 17 mayo 2007].

Disponible en: <http://es.wikipedia.org/wiki/Diagrama_de_despliegue >

Bibliografías

- *Yanes, Rafael. Un género entre la opinión y la actualidad. Revista Latina de Comunicación Social, diciembre de 2004, nº 58. [Consultada 3 noviembre 2006].*

Disponible en: < <http://www.ull.es/publicaciones/latina/20041858yanes.htm> >

- *Mandrake, Revisión Rápida de PHP5 integrado con Zend, septiembre 2004. Disponible en: <<http://www.venezolano.web.ve/archives/230-Revision-rapida-de-PHP5-integrado-con-Zend.html>>.* [Fecha de consultada 25 febrero 2007].

- *Gil Martín, Manuel Alejandro. Arquitectura para los sistemas que conforman la intranet universitaria, 9 mayo 2007. [Fecha de consulta 17 de mayo de 2007] Disponible en: <http://uddi.uci.cu>*

- *Clases de Ingeniería del Software I, curso 2005-2006, UCI. [Fecha de consultada 15 febrero 2007].*

- *Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Primera Edición por Prentice Hall, Hispanoamericana S.A. 1999.*

- *Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda Edición por Prentice Hall.*

- *Adobe Dreamweaver. [Fecha de consultada 25 febrero 2007]. Disponible en: <http://es.wikipedia.org/wiki/Dreamweaver>.*

- *Herramienta CASE*. Disponible en: <http://es.wikipedia.org/wiki/CASE> [Fecha de consultada 25 febrero 2007].
- *Web dinámicas con PHP*. [Fecha de consultada 25 febrero 2007]. Disponible en: <http://www.mailxmail.com/curso/informatica/phpprogramacionweb/capitulo4.htm>
- *PHP*. Disponible en: <http://es.wikipedia.org/wiki/PHP> [Fecha de consultada 23 febrero 2007].
- *López, D. PHP4 y PHP5: ¿Cuál elegir? ¿Migrar o no Migrar? El advenimiento de PHP6*, 2 agosto 2006. [Fecha de consultada 23 febrero 2007]. Disponible en <http://www.maestrosdelweb.com/editorial/php4y5/>
- *ActionScript 2.0 con PHP 5, 2006*. [Fecha de consultada 23 febrero 2007].
Disponible en: <http://www.ciberaula.com/curso/actionscript2php5/>
- *SQL*. [Fecha de consultada 23 febrero 2007]. Disponible en: http://es.wikipedia.org/wiki/SQL#Caracter.C3.ADsticas_generales
- *MySQL*, febrero 2007. [Fecha de consultada 25 febrero 2007]. Disponible en: <http://es.wikipedia.org/wiki/MySQL>
- *Diseño del sistema de información, pág 23*. [Fecha de consultada 4 mayo 2007]. Disponible en: <http://www.csi.map.es/csi/metrica3/dsiproc.pdf>
- *Franco Navarro, J. UML en acción. Modelando Aplicaciones Web. Centro de Referencia de Ingeniería de Software, perteneciente al Centro de Estudio de Ingeniería de Sistemas, del ISPJAE. Mayo 2005, Ciudad de la Habana.*
- *Fernandez Vilas, Ana. Diagrama de Despliegue, marzo 2003*. [Fecha de consultada 17 mayo 2007]. Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>

Anexos

Anexo 1: Descripción detallada de los casos de uso

Tabla 12: Descripción del caso de uso autenticar.

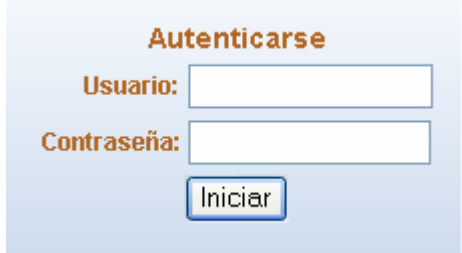
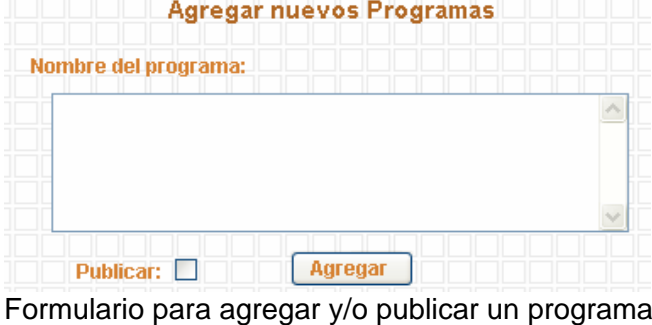
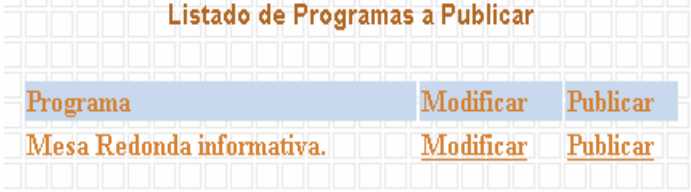
Caso de Uso 1	Autenticar	
Actores	Usuario (inicia)	
Propósito	Permitir el acceso al sistema de forma segura.	
Resumen	El caso de uso se inicia cuando el usuario introduce los datos (usuario y contraseña) que se le piden para acceder a la aplicación, estos son verificados con Ldap para probar su validez y el sistema comprueba si el usuario tiene asignado algún rol. El caso de uso termina cuando el usuario obtiene los permisos de acceder al sistema según los roles definidos.	
Referencias	RF1	
Precondiciones	Existan los usuarios.	
Postcondiciones	Los roles se encuentren definidos correctamente.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El usuario entra Usuario y Contraseña.	2. El sistema consulta el Ldap, busca el usuario y compara la contraseña. En caso de ser correcto, se busca el usuario en el sistema para comprobar si tiene asignado algún rol y se le asignan los permisos.	
Flujo Alternativo		
Acciones del Actor	Respuesta del Sistema	
	2.1 En caso de no existir se envía un mensaje de aviso. Retornar a 1	
Prototipo de interfaz	 <p>Formulario para la autenticación de los usuarios.</p>	
Prioridad	Crítico	

Tabla 13: Descripción del caso de uso gestionar programas.

Caso de Uso 2	Gestionar programas
Actores	Administrador (inicia).
Propósito	Permitir al Administrador la posibilidad de agregar, publicar, terminar y modificar los datos correspondientes a los programas.
Resumen	El caso de uso inicia cuando el Administrador requiere agregar, publicar, modificarlo o terminar un programa. Si selecciona agregar programa, el sistema solicita el nombre del programa a introducir y si lo quiere publicar y guardar; verifica que tengan el formato correcto, almacena y actualiza los datos. Si

	selecciona publicar, el sistema muestra un listado de los programas que no están publicados y selecciona al que desea publicar el nombre, lo publica, verifica y almacena los datos si todo es correcto. Si selecciona modificar, el sistema muestra un listado de los temas publicados y no publicados existentes y selecciona al que desea modificar los datos, modifica, verifica y almacena los datos si todo es correcto. Si selecciona terminar, el sistema muestra un listado de los programas existentes y selecciona al que desea terminar, el sistema actualiza los datos. El caso de uso termina cuando se agrega, publica, modifica o elimina un usuario.
Referencias	RF2, RF 11.
Precondiciones	-El Responsable de Administración ha sido autenticado en el sistema.
Postcondiciones	-Se publica el tema y se actualiza los datos de los programas. -Los estados se encuentren definidos correctamente.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Administrador selecciona Programas del menú.	2. El sistema despliega en el menú las opciones referentes a las operaciones sobre los programas. Las opciones pueden ser: a) Agregar (Ir a sección Agregar_programa). b) Publicar (Ir a sección Publicar_programa). c) Modificar (Ir a sección Modificar_programa). d) Terminar (Ir a sección Terminar_programa).
Sección Agregar	
3. El Administrador selecciona del menú programa la opción agregar.	4. El sistema solicita el nombre del programa a insertar.
5. El Administrador introduce los datos solicitados.	
6. a) El Administrador puede solicitar la opción Publicar programa (mediante un checkbox). b) El Administrador puede no solicitar la opción Publicar programa.	
7. El Administrador solicita guardar el programa.	8. El sistema verifica que los datos estén correctos y que el formulario no esté vacío.
	9. En caso de estar correcta la información, el sistema verifica: a) Si el Administrador solicitó la opción publicar. El sistema guarda y publica los programas. b) Si el Administrador no solicita publicar, el sistema guarda los programas. El sistema actualiza el estado de los programas.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
	El sistema verifica que los datos no son correctos o no está lleno el formulario. El sistema muestra una ventana señalando el error. Retornar a 5

Prototipo de interfaz	
Sección Publicar	
3. El Administrador solicita la opción publicar programas del menú.	4. El sistema muestra un listado con los programas existentes en la base de datos.
5. El Administrador localiza el nombre del programa que desea realizar un cambio. a) Solicita modificar el programa escogido. b) Solicita publicar el programa escogido.	6. a) En caso de solicitar una modificación, el sistema le permite modificar el nombre del programa sin publicar seleccionado y actualiza el listado de programas publicados. El sistema modifica y actualiza los programas sin publicar. b) En caso de solicitar publicar un programa, el sistema le permite publicar el nombre del programa seleccionado y actualiza el estado de los programas.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
Prototipo de interfaz	
Tabla para modificar y publicar un programa.	
Sección Terminar	
3. El Administrador solicita la opción terminar programas del menú.	4. El sistema muestra un listado con los programas publicados.
5. El Administrador solicita realizar una acción sobre un programa: a) Modificar b) Terminar	6. a) En caso de solicitar una modificación, el sistema le permite modificar el nombre del programa publicado seleccionado y actualiza el listado de programas publicados. b) En caso de solicitar terminar, el sistema termina el programa publicado y actualiza el estado de los programas.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema

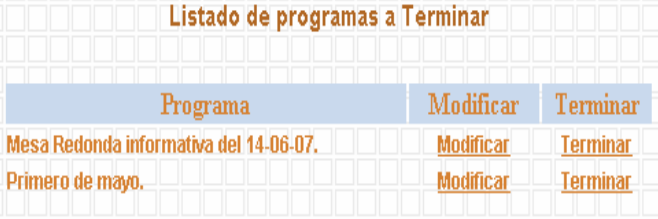
Prototipo de interfaz	 <p>Tabla para modificar y terminar un programa.</p>
Prioridad	Crítico

Tabla 14: Descripción del caso de uso registrar opinión.

Caso de Uso 3	Registrar Opinión
Actores	Usuarios (inicia)
Propósito	Emitir opinión acerca de un programa.
Resumen	El Caso de Uso se inicia cuando el usuario escoge el tema del cual va a opinar, introduce en el sistema una o varias opiniones acerca de un programa determinado. El caso de uso termina cuando el usuario envía la opinión.
Referencia	RF3, RF11
Precondiciones	Que exista al menos un tema
Postcondiciones	Se registra la opinión en la tabla de opiniones y se actualiza.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El usuario solicita opinar en el menú.	2. El sistema muestra un listado de los programas publicados.
3. El usuario escoge del listado un programa publicado.	4. El sistema muestra en una ventana el programa solicitado, con un área de texto para opinar.
5. El usuario emite la opinión.	
6. El usuario solicita enviar la opinión a través de un botón.	7. El sistema verifica que el formulario no esté vacío.
	8. El sistema almacena, actualiza los datos y el estado.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
6.1 El usuario solicita no enviar la opinión. Retorna 3	
	7.1 En caso de que el formulario esté vacío. Se envía un mensaje solicitando que inserte la opinión. Retornar a 5.

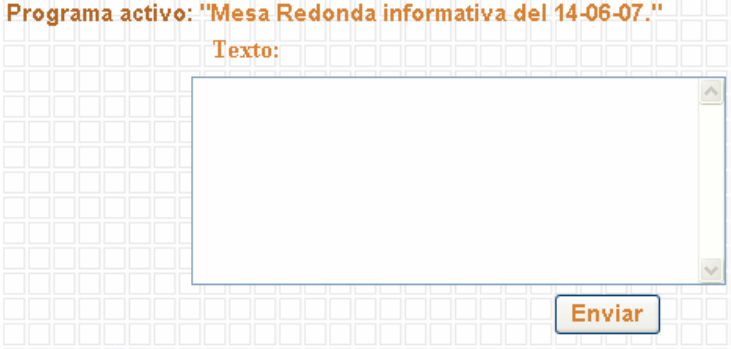
Prototipo de interfaz	 <p>Programa activo: "Mesa Redonda informativa del 14-06-07." Texto: <input type="text"/> <input type="button" value="Enviar"/></p>
Prioridad	Crítico

Tabla 15: Descripción del caso de uso revisar opinión.

Caso de Uso 4	Revisar opinión	
Actores	Editor (Inicia)	
Propósito	Posibilita editar, archivar, cancelar las opiniones, actualiza las opiniones y los estados.	
Resumen	El caso de uso inicia cuando el Editor selecciona revisar la opinión dado un programa, el sistema le va a mostrar una región editable que carga el texto de la opinión; permitiendo revisar, archivar o cancelar, luego el sistema almacena y actualiza las opiniones.	
Referencias	RF 5, RF 4, RF 11.	
Precondiciones	Que existan opiniones sin editar.	
Postcondiciones	Se modifica el estado de la opinión y se actualiza las opiniones.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Editor solicita editar opiniones del menú.	2. El sistema muestra un listado de programas terminados y publicados que tienen opiniones sin editar.	
3. El Editor solicita un programa determinado para editar opiniones.	4. El sistema muestra en una región editable el texto de las opiniones del tema seleccionado de una en una.	
5. El Editor solicita las opciones referentes a las operaciones de editar: a) Actualizar b) Archivar c) Cancelar	6. En caso de que el Editor solicite: a) Actualizar: el sistema actualiza el estado de la opinión revisada. b) Archivar: el sistema actualiza el estado de la opinión archivada. c) Cancelar: el sistema actualiza el estado de la opinión, pasa de en edición a no editada. Sale a la página principal.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	

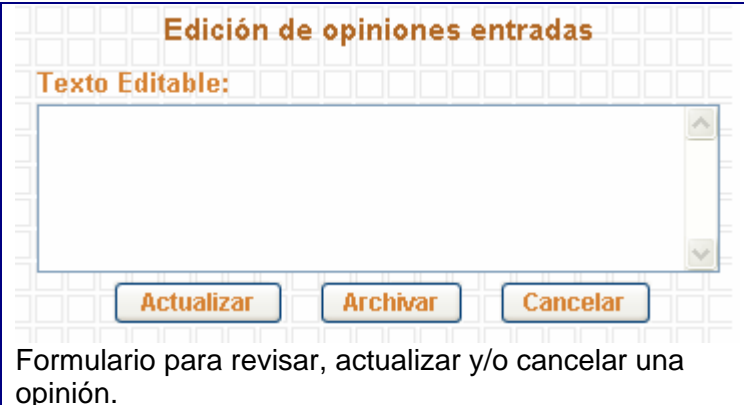
Prototipo interfaz	
Prioridad	Crítico

Tabla 16: Descripción del caso de uso aprobar opinión.

Caso de Uso 5	Aprobar opinión.
Actores	Revisor (Inicia)
Propósito	Posibilita aprobar, archivar y cancelar las opiniones, actualiza los estados.
Resumen	El caso de uso inicia cuando el Revisor selecciona aprobar, el sistema muestra el texto de la opinión; permitiendo aprobar, archivar y cancelar. Verifica y almacena los datos si todo es correcto.
Referencias	RF 6, RF 4, RF 11.
Precondiciones	Que existan opiniones editadas.
Postcondiciones	Se modifica el estado de la opinión y se actualizan las opiniones.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Revisor solicita seleccionar la opción aprobar opinión del menú.	2. El sistema muestra un listado de programas terminados y publicados que tienen opiniones editadas.
3. El Revisor solicita un programa determinado para aprobar las opiniones.	4. El sistema muestra el texto de las opiniones editadas del tema seleccionado de una en una.
5. El Editor solicita las opciones referentes a las operaciones de editar: a) Aprobar b) Archivar c) Cancelar	6. En caso de que el Administrador o Editor soliciten: a) Aprobar: el sistema actualiza el estado de la opinión aprobada. b) Archivar: el sistema actualiza el estado de la opinión archivada. c) Cancelar: el sistema actualiza el estado de la opinión y sale a la página principal.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema

Prototipo interfaz	<p style="text-align: center;">Aprobar opiniones revisadas</p> <p>Texto a Aprobar:</p> <div style="border: 1px solid gray; height: 60px; width: 100%;"></div> <p style="text-align: center;"> <input type="button" value="Aprobar"/> <input type="button" value="Archivar"/> <input type="button" value="Cancelar"/> </p> <p>Formulario para aprobar, actualizar y/o cancelar una opinión.</p>
Prioridad	Crítico

Tabla 17: Descripción del caso de uso generar documento.

Caso de Uso 6	Generar Documento	
Actores	Administrador	
Propósito	Posibilita generar un documento con un listado de opiniones aprobadas por tema.	
Resumen	Si el Administrador selecciona generar documento, el sistema le muestra un listado de programas para que seleccione uno, le permite listar de ese programa el listado de las opiniones aprobadas. El caso de uso termina cuando se genera el documento pdf.	
Referencias	RF7, RF 11.	
Precondiciones	Que existan opiniones aprobadas.	
Postcondiciones		
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Administrador selecciona la opción Generar Documento del menú opiniones.	2. El sistema muestra un listado de programas terminados y publicados, que cuentan con opiniones aprobadas.	
3. El Administrador localiza el programa del cual va a generar el listado.		
4. El Administrador solicita listar las opiniones aprobadas.	5. El sistema muestra un listado con las opiniones aprobadas de ese tema.	
6. El Administrador solicita generar documento pdf.	7. El sistema generar el documento en pdf y sale de la aplicación.	
Flujo alterno		
Acción del Actor	Respuesta del Sistema	
Prototipo interfaz	<p style="text-align: center;">Listado de las Opiniones Aprobadas</p> <p>Texto de opinión</p> <p>probando</p> <p>probando caracteres especiales(ñóá).</p> <p>dfgdsfg, sdfhbsdf ño.</p> <p style="text-align: center;"><input type="button" value="Generar PDF"/></p>	

	Muestra el listado de los textos de las opiniones para genera el documento.
Prioridad	Crítico

Tabla 18: Descripción del caso de uso recuperar archivadas.


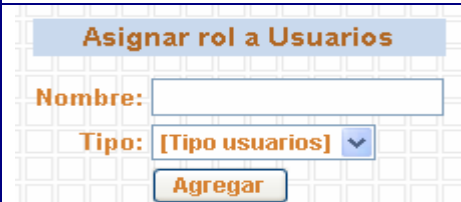
Caso de Uso 7	Recuperar archivadas
Actores	Administrador (Inicia)
Propósito	Posibilita consultar las opiniones archivadas por tema dando la posibilidad de poderlas editar, actualizando los estados de las opiniones.
Resumen	Si el Administrador selecciona recuperar archivadas, el sistema muestra un listado de opiniones archivadas, permite actualizar el estado, verifica y almacena los datos.
Referencias	RF 8, RF 4, RF 11.
Precondiciones	Que existan opiniones archivadas en la BD.
Postcondiciones	Se modifica el estado de la opinión y se actualizan las opiniones.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción recuperar archivadas del menú opiniones.	2. El sistema muestra un listado con las opiniones archivadas existentes en la base de datos.
3. El Administrador localiza las opiniones archivadas que desea modificar el estado.	
4. El Administrador solicita editar las opiniones archivadas.	5. El sistema actualiza el estado de la opinión, pasa de archivadas a sin revisar.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
Prototipo interfaz	 <p>Un listado con los textos de las opiniones archivadas para actualizar el estado (recuperar.)</p>
Prioridad:	Secundario

Tabla 19: Descripción del caso de uso gestionar usuarios.

Caso de Uso 8	Gestionar Usuarios
Actores	Administrador (inicia).
Propósito	Brindar la posibilidad de asignar rol y actualizar rol de los usuarios que interactúan con el sistema.
Resumen	El caso de uso inicia cuando el Administrador asigna o actualiza el rol a un usuario. Si selecciona asignar rol el sistema solicita el usuario del dominio a introducir y el rol a desarrollar, que será el que le dará los privilegios y accesos; verifica que tengan el formato correcto, almacena y actualiza los datos. Si selecciona actualizar rol, el sistema muestra un listado de los usuarios con su rol y selecciona al que desea actualizar, el sistema actualiza los datos. El caso de uso termina cuando se asigna o actualiza el rol a un usuario.

Referencias	RF 9, RF 11.
Precondiciones	El Responsable de Administración ha sido autenticado en el sistema. Los roles se encuentren definidos correctamente.
Postcondiciones	Se determinan los privilegios de usuarios y se actualicen los datos.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Administrador solicita usuarios del menú.	2. El sistema despliega en el menú las opciones referentes a las operaciones sobre los usuarios. Las opciones pueden ser: a) Asignar rol (Ir a sección Inserta usuarios). b) Actualizar rol (Ir a sección gestiona usuarios).
Sección Asignar rol	
3. El Administrador solicita la opción Asignar rol del menú usuario.	4. El sistema solicita el usuario del dominio y el rol. Según el rol, será el acceso que tenga el usuario.
5. El Administrador introduce los datos solicitados.	
6. El Administrador asigna el rol al usuario.	7. El sistema verifica que los datos proporcionados sean correctos.
	8. En caso de estar correctos el sistema almacena y actualiza los datos.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
	7.1 El sistema comprueba que los datos no son correctos. Se indica en una ventana el error para que sea corregido. Retornar a 5
Prototipo interfaz	 <p>Formulario para asignar rol a los usuarios.</p>
Sección Actualizar rol	
3. El Administrador solicita la opción Actualizar rol del menú usuario.	4. El sistema muestra un listado con los usuarios y los roles correspondiente.
5. El Administrador localiza el usuario que desea actualizar.	
6. El Administrador solicita actualizar.	7. El sistema verifica y solicita seguridad para actualizar el rol del usuario.
8. El Administrador procede a aceptar la solicitud de actualizar el rol.	9. El sistema actualiza los datos.
Flujo alternativo	
Acción del Actor	Respuesta del Sistema

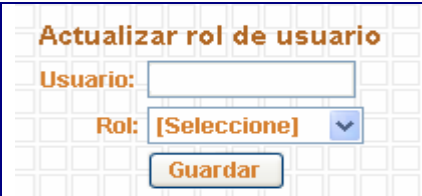
Prototipo interfaz	 <p>Formulario para actualizar rol a los usuarios.</p>
Prioridad:	Crítico

Tabla 20: Descripción del caso de uso generar reportes.

Caso de Uso 9	Generar Reportes
Actores	Administrador (inicia).
Propósito	Mostrar al Administrador los reportes estadísticos generales por programas y así obtener la información almacenada en la base de datos.
Resumen	El caso de uso inicia cuando el administrador solicita los reportes generales de un determinado programa. Selecciona un programa determinado y el sistema genera reportes estadísticos acerca de las opiniones enviadas de ese programa. El caso de uso concluye cuando la aplicación muestra los resultados de la búsqueda.
Referencias	RF 10
Precondiciones	Existan reportes registrados. Se encuentren definidos correctamente los tipos de reportes.
Postcondiciones	El Administrador obtiene los resultados generados.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Administrador selecciona en el menú Reportes Estadísticos.	2. El sistema muestra un listado de los programas que han sido publicados en el sistema y tienen opiniones, además de un listado de con tipos de reporte definidos.
3. El administrador selecciona el programa deseado y el tipo de reporte deseado. El tipo de reporte puede ser: a) General de opiniones. b) Opiniones por áreas.	
4. El administrador solicita mostrar los reportes.	5. El sistema verifica que se haya seleccionado un programa y el tipo de reporte.
	6. El administrador seleccionó el tipo de reporte: a) El sistema muestra una interfaz con los reportes estadísticos acerca de las opiniones generales por tema. b) El sistema muestra una interfaz con los reportes estadísticos acerca de las opiniones por facultades y estudiantes por tema.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
	5.1 El sistema comprueba de que no hizo completa la selección. El sistema muestra un mensaje de error.



	Retorna al 3.
Prototipo interfaz	 <p>Formulario para seleccionar el tipo de reporte</p>
Prioridad	Secundario

Tabla 21: Descripción del caso de uso Log del Sistema.

CU-10	Log del Sistema
Actor	Administrador (inicia).
Propósito	Mostrar al Administrador los logs de actividades registrados en el sistema.
Resumen	El caso de uso inicia cuando el administrador solicita los logs de programas o de opiniones. El sistema muestra todas las actividades que realiza el actor sobre la gestión de los programas y de las opiniones. El caso de uso finaliza cuando el sistema muestra los logs del sistema.
Referencias	RF 12, RF 11.
Precondiciones	Existan opiniones y programas registrados.
Postcondiciones	El Administrador obtiene el listado de logs.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Administrador solicita en el menú Logs del sistema. a) Logs de programas. b) Logs de opiniones.	2. El sistema muestra en un listado las actividades que realizó el actor. a) El sistema muestra una interfaz con el listado de actividades realizadas sobre los programas. b) El sistema muestra una interfaz con el listado de actividades realizadas sobre las opiniones.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
Prototipo interfaz	 <p>Tabla para recoger los logs del sistema usuarios, acción, fecha e ip de la PC.</p>
Prioridad	Secundario

Anexo 2: Diagrama de Clases del Análisis



Figura 4: Diagrama de clases del análisis autenticar.

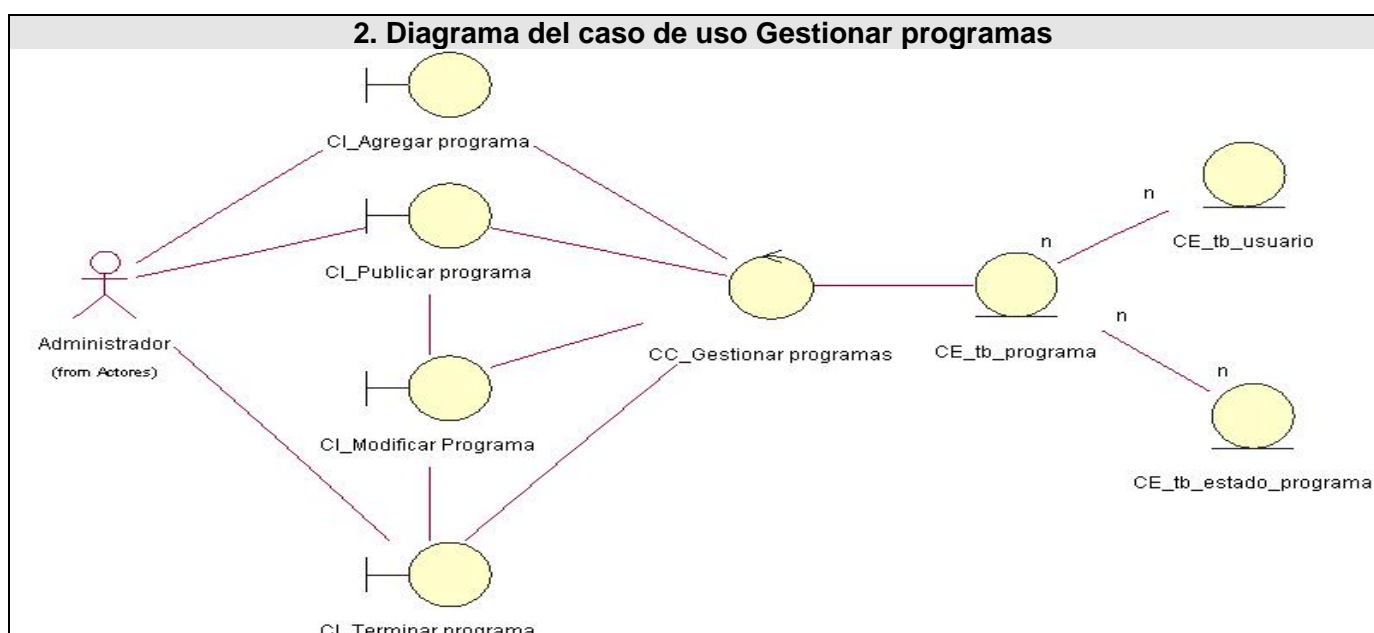


Figura 5: Diagrama de clases del análisis gestionar programas.

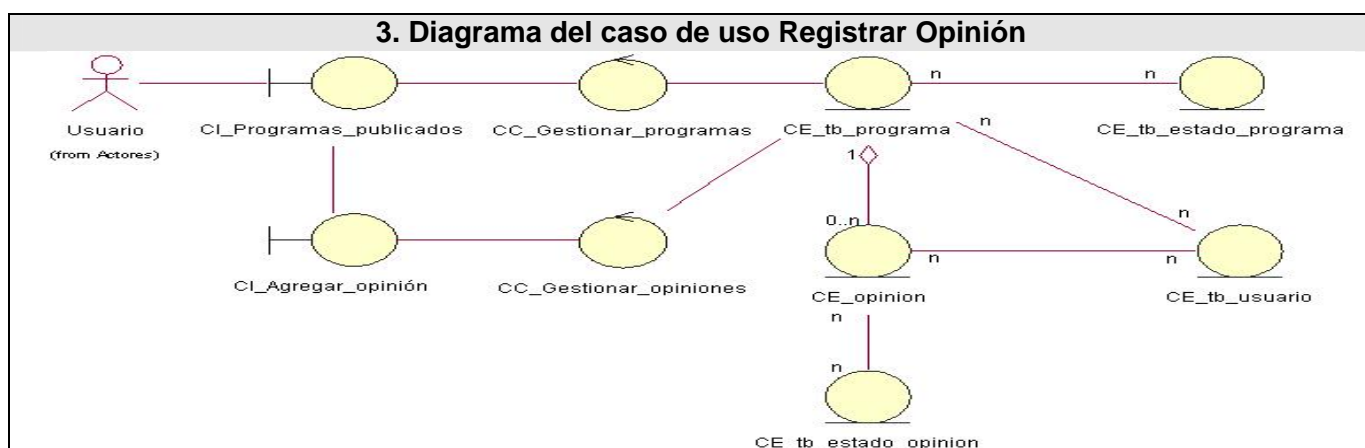


Figura 6: Diagrama de clases del análisis insertar opinión.

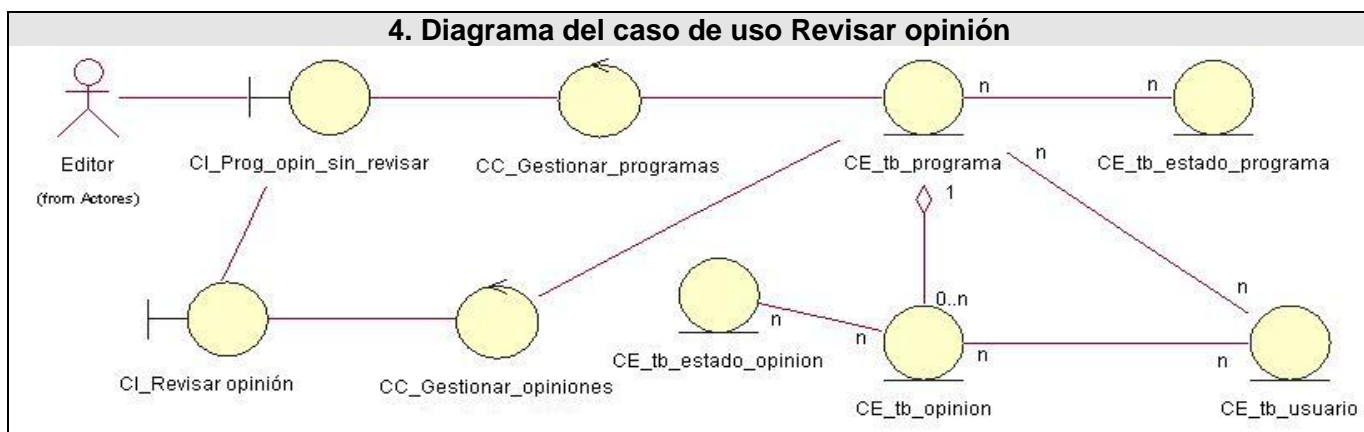


Figura 7: Diagrama de clases del análisis revisar opinión.

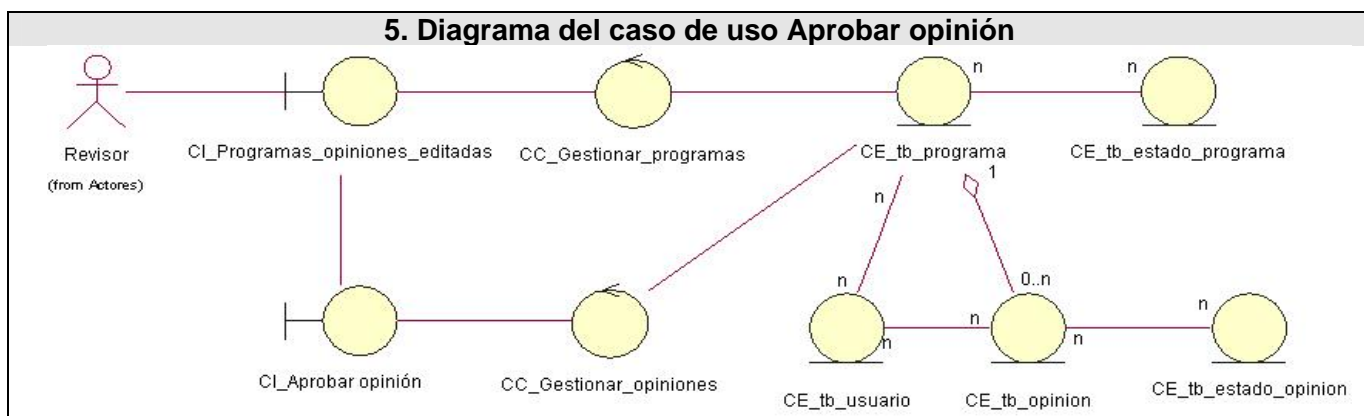


Figura 8: Diagrama de clases del análisis aprobar opinión.

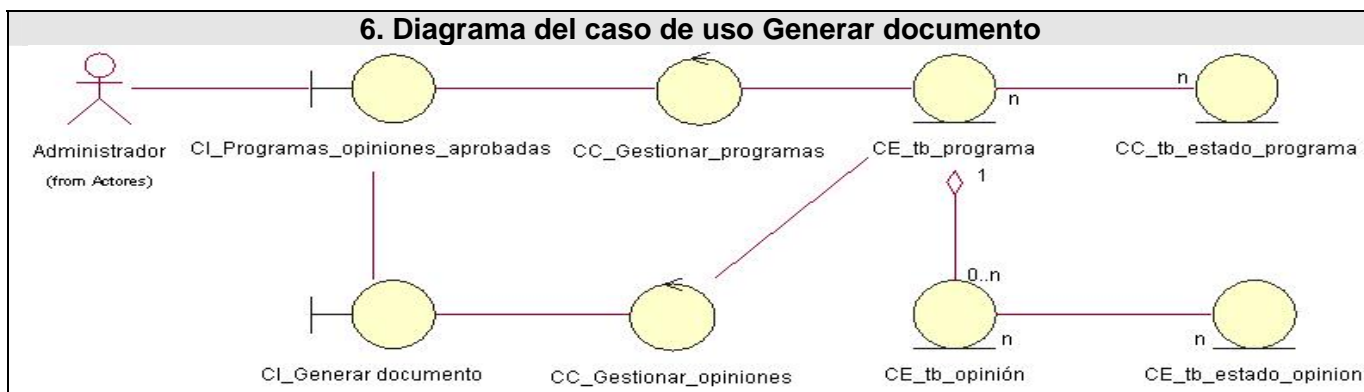


Figura 9: Diagrama de clases del análisis generar documento.

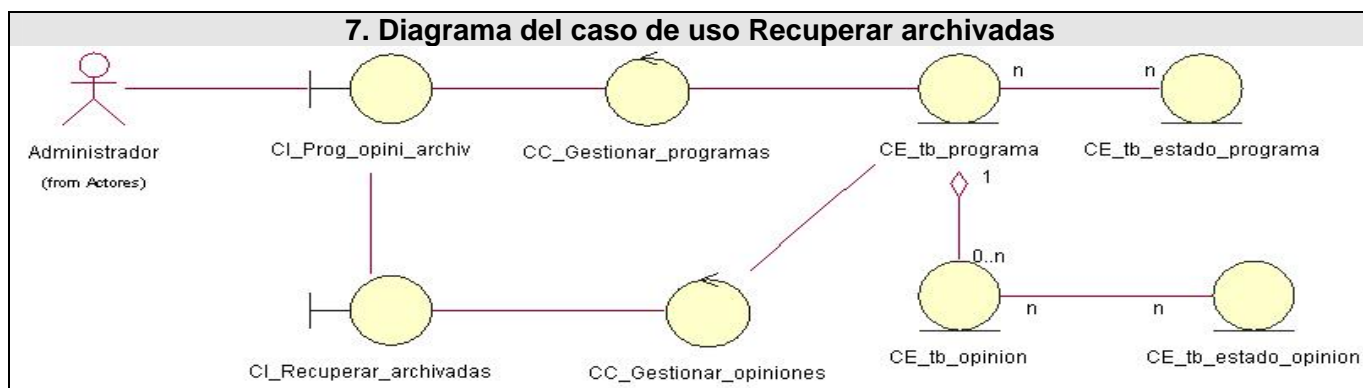


Figura 10: Diagrama de clases del análisis recuperar archivadas.

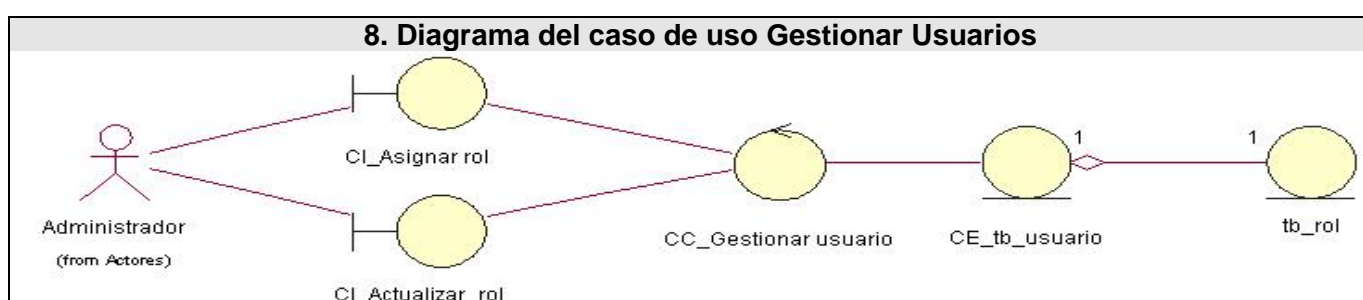


Figura 11: Diagrama de clases del análisis gestionar usuarios.

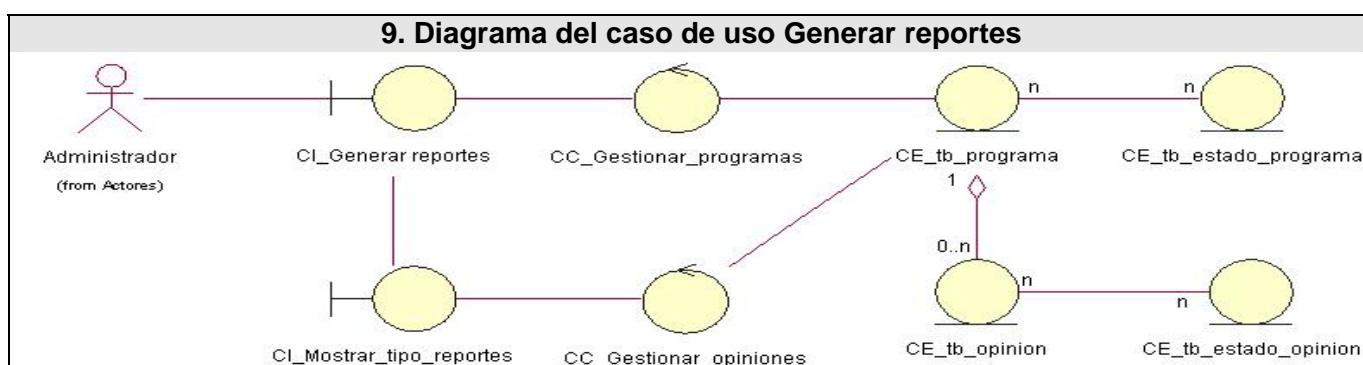


Figura 12: Diagrama de clases del análisis generar reportes.

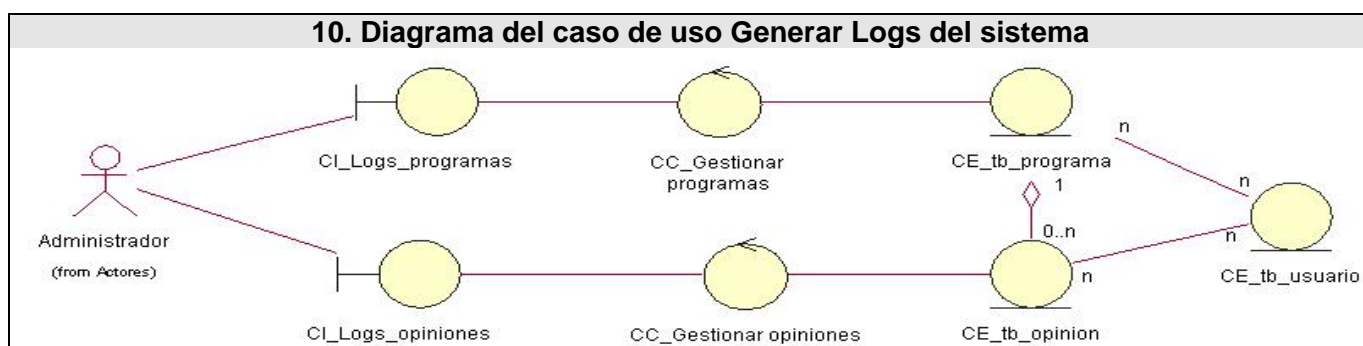


Figura 13: Diagrama de clases del análisis generar logs del sistema.

Anexo 3: Diagrama de colaboración

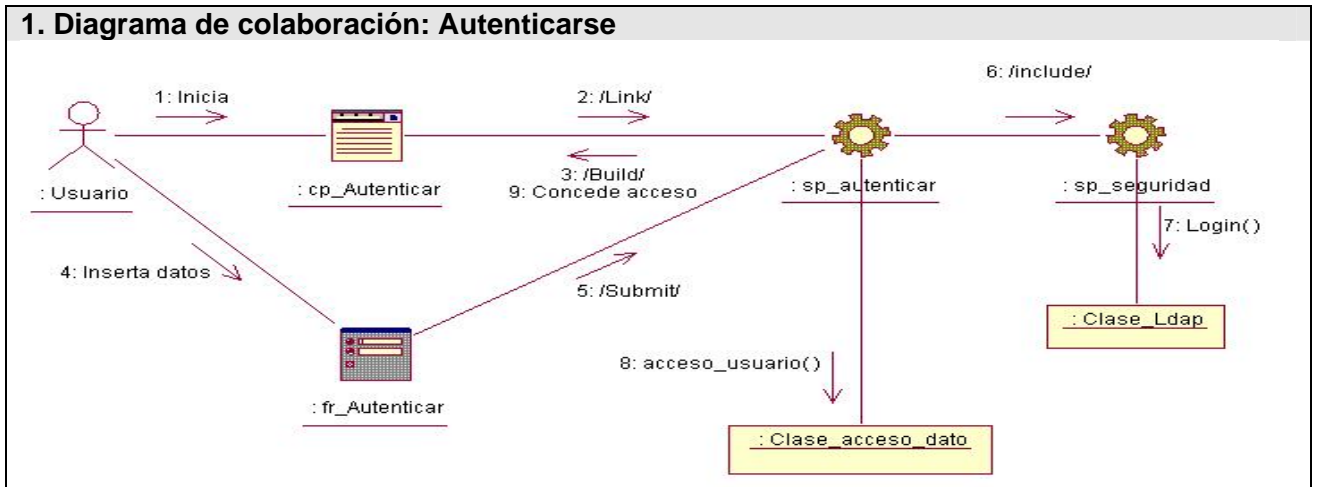


Figura 14: Diagrama de colaboración del escenario autenticarse.

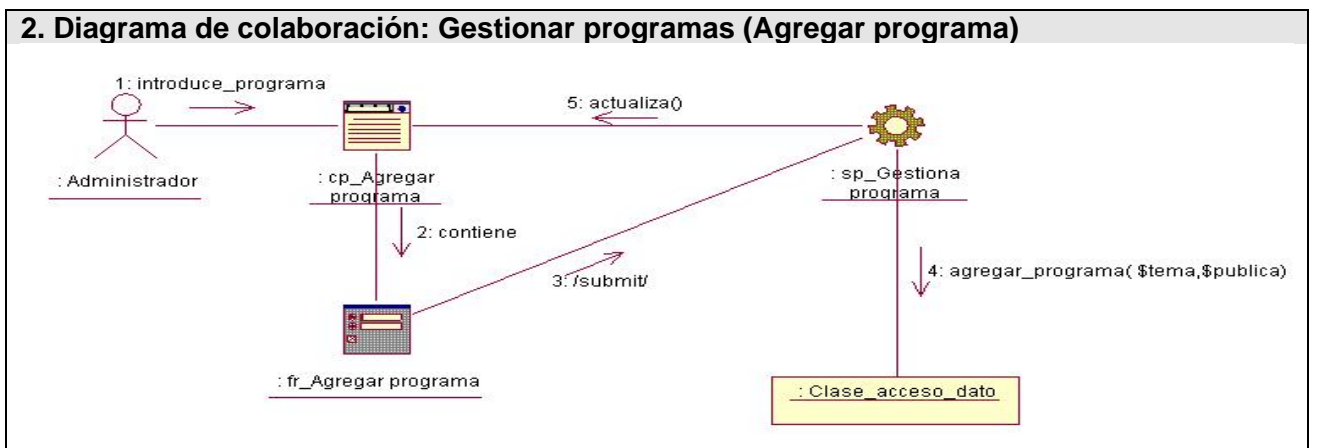


Figura 15: Diagrama de colaboración del escenario agregar programa.

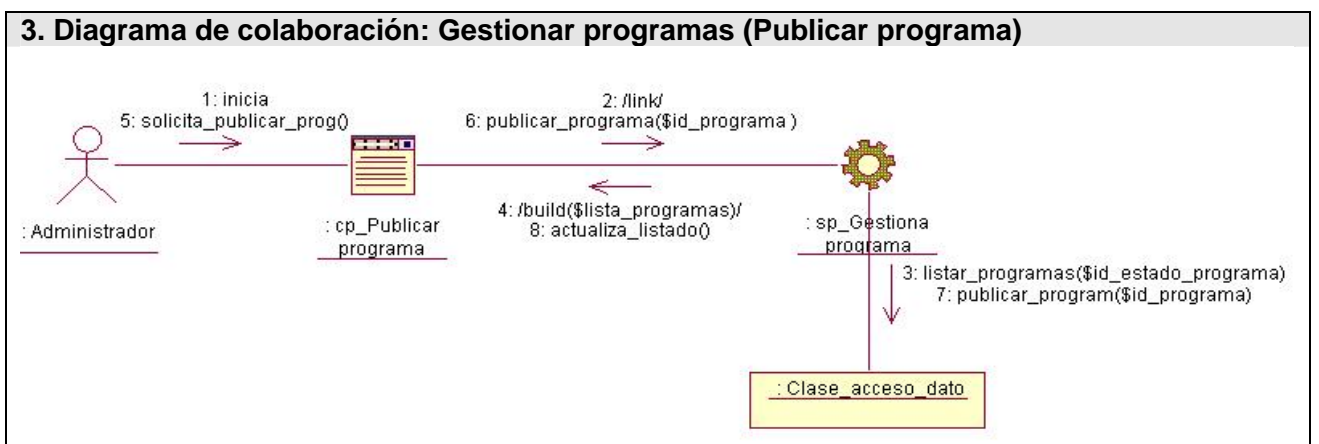


Figura 16: Diagrama de colaboración del escenario publicar programa.

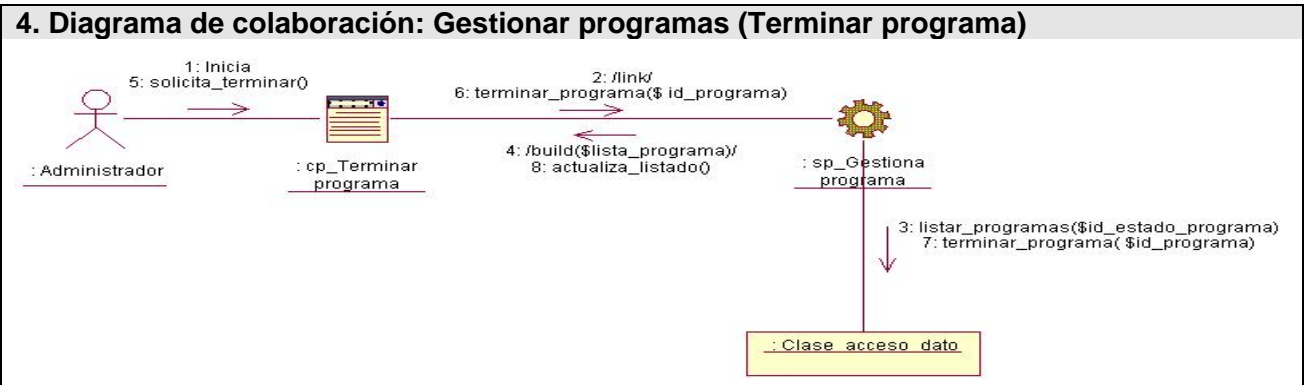


Figura 17: Diagrama de colaboración del escenario terminar programa.

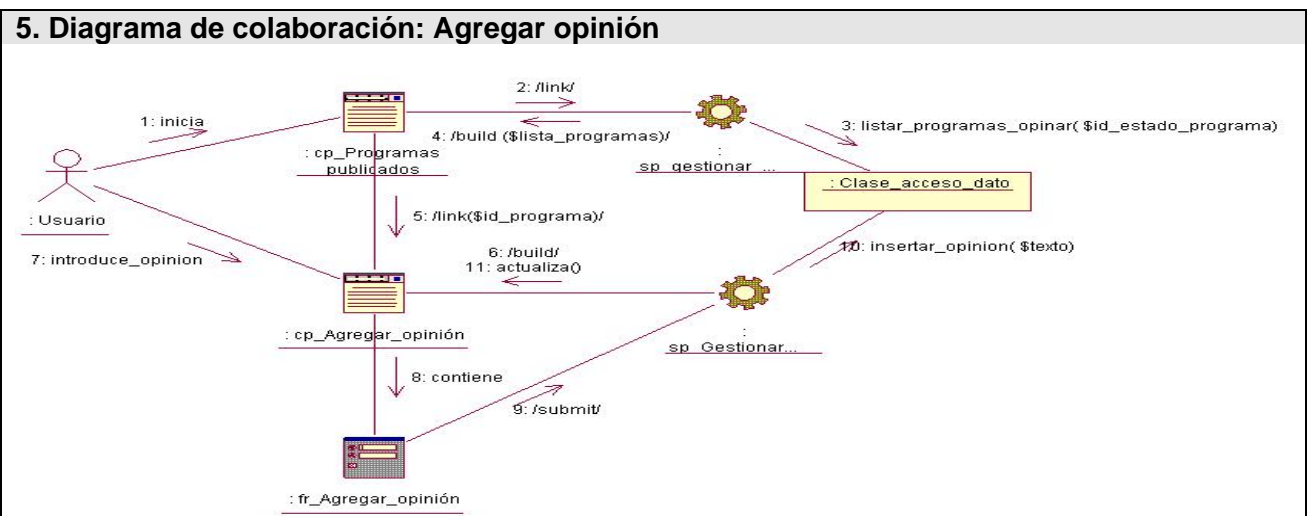


Figura 18: Diagrama de colaboración del escenario agregar opinión.

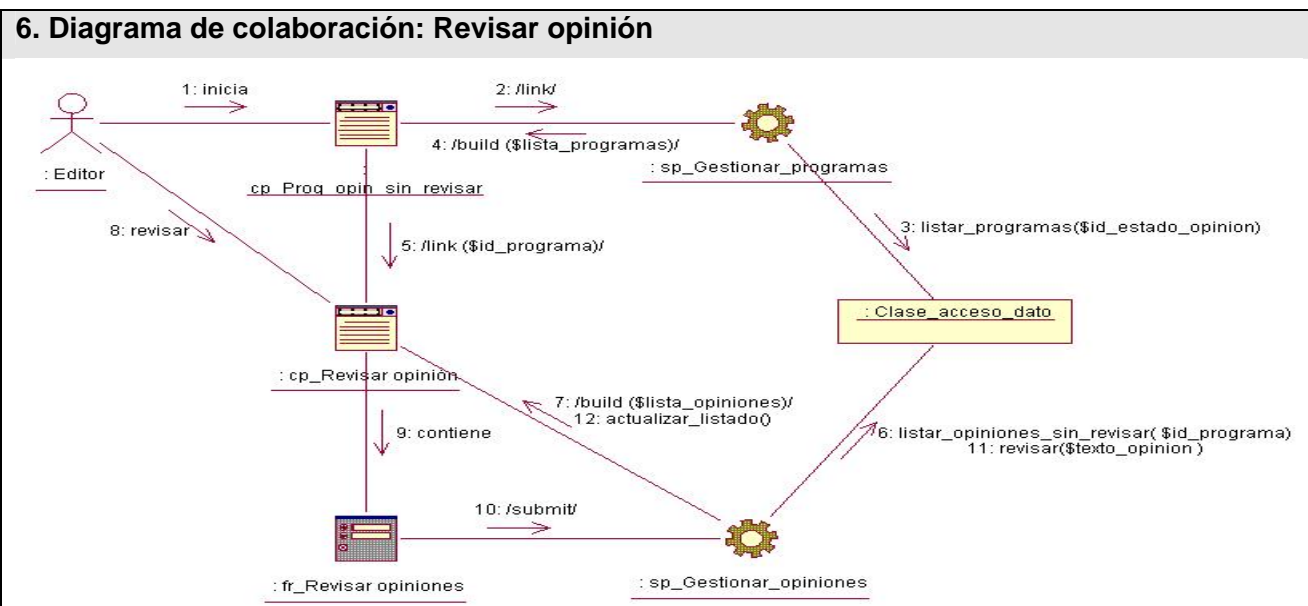


Figura 19: Diagrama de colaboración del escenario revisar opinión.

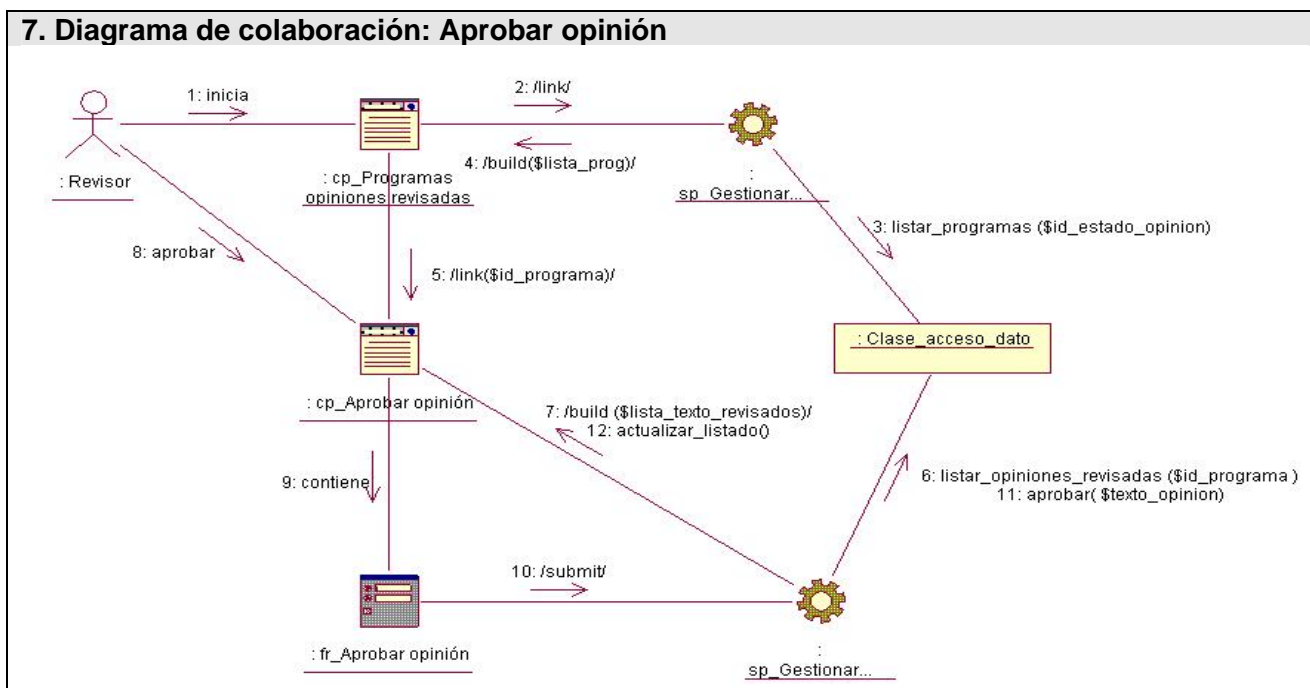


Figura 20: Diagrama de colaboración del escenario aprobar opinión

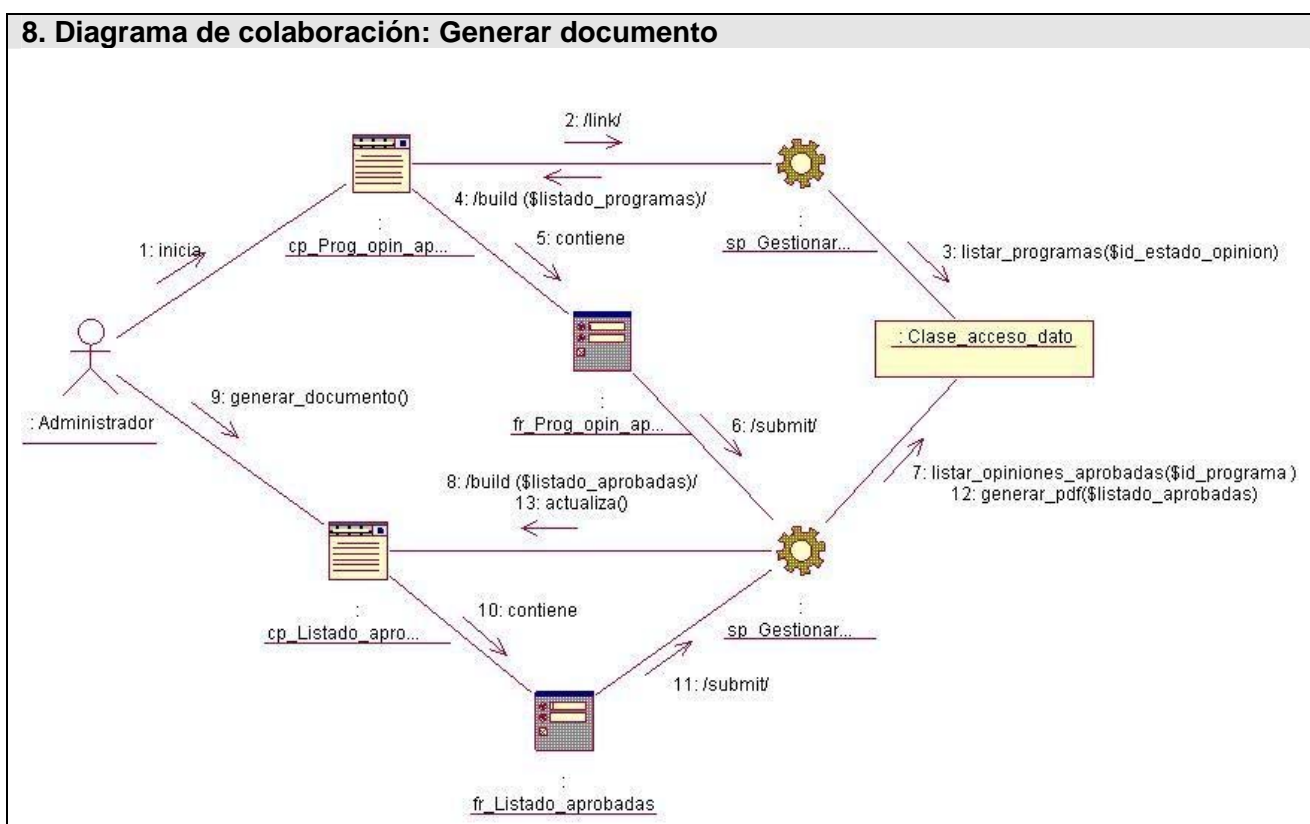


Figura 21: Diagrama de colaboración del escenario generar documento.

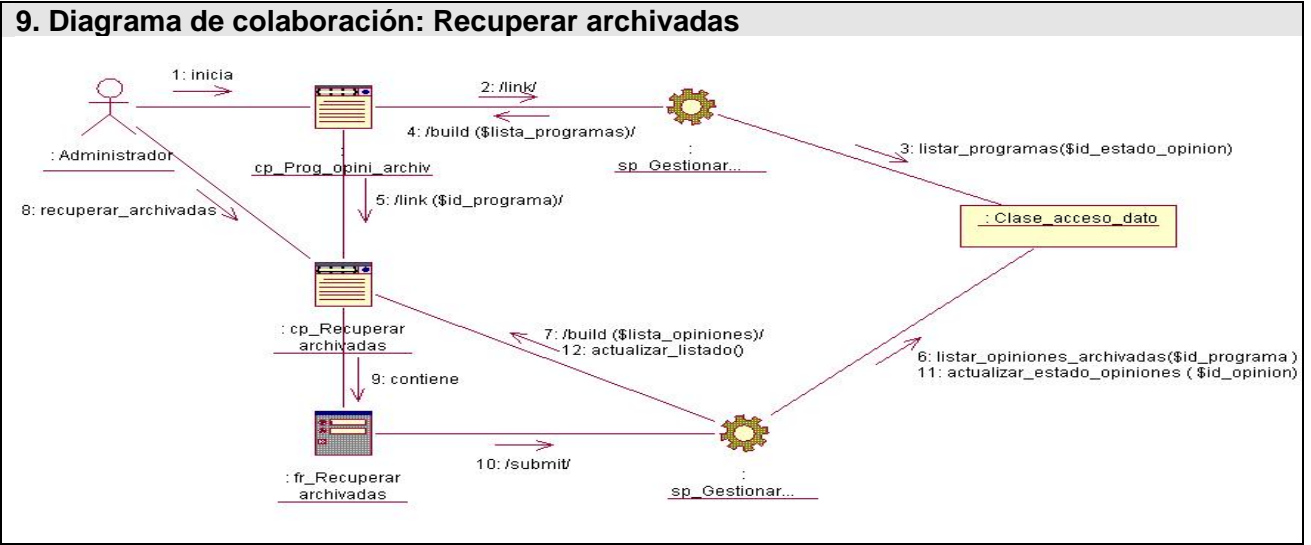


Figura 22: Diagrama de colaboración del escenario recuperar archivadas.

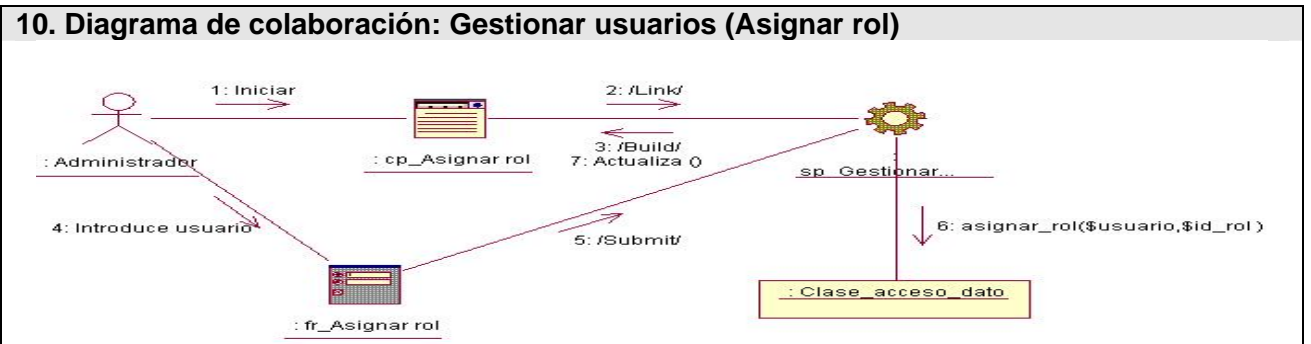


Figura 23: Diagrama de colaboración del escenario asignar rol.

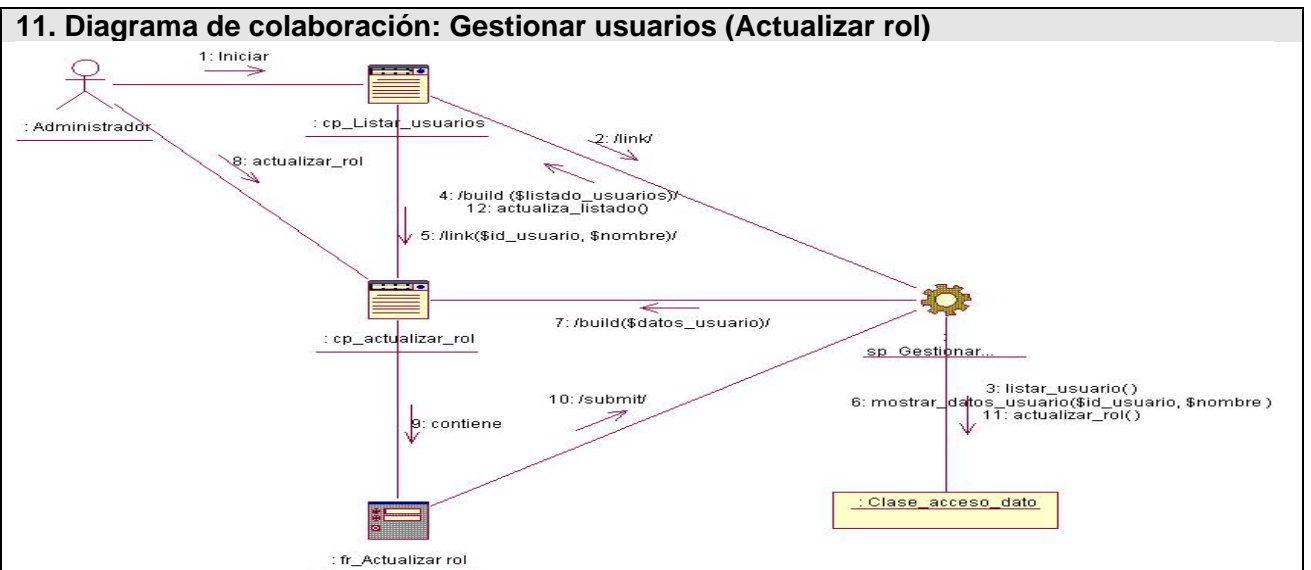


Figura 24: Diagrama de colaboración del escenario actualizar rol.

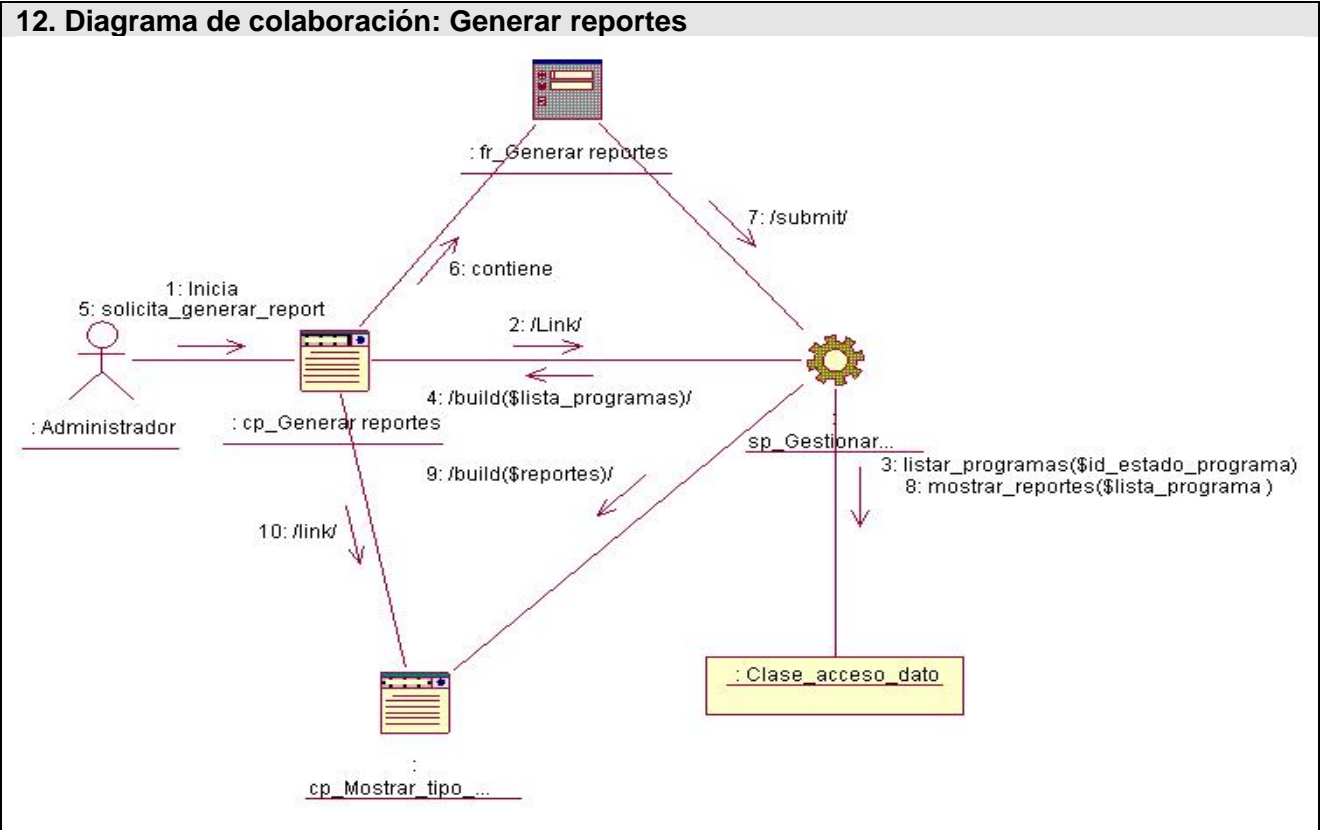


Figura 25: Diagrama de colaboración del escenario generar reportes.

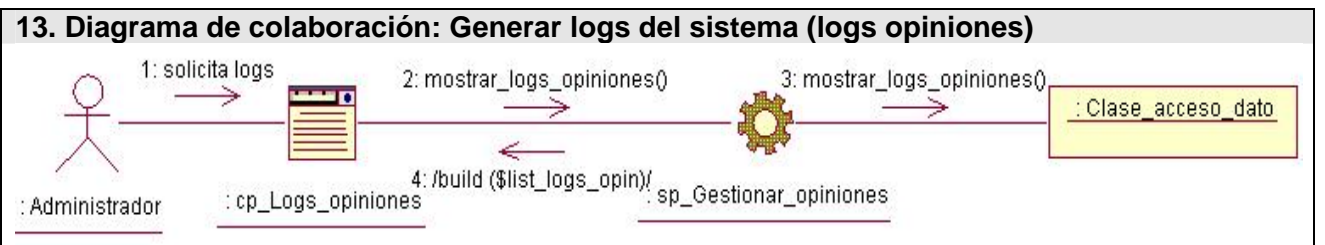


Figura 26: Diagrama de colaboración del escenario generar logs de opiniones.

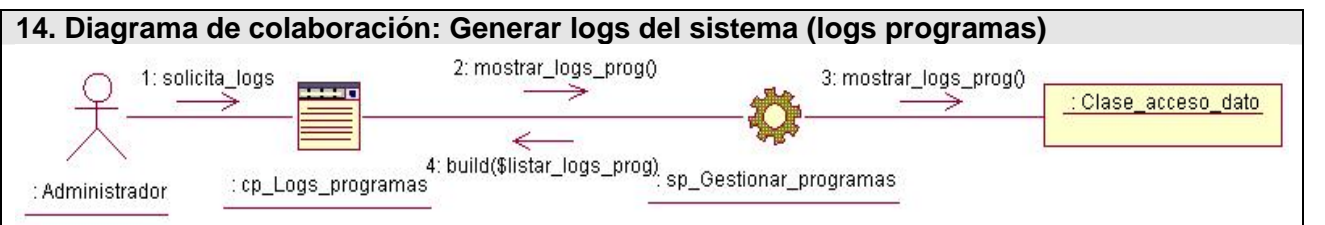


Figura 27: Diagrama de colaboración del escenario generar logs de opiniones.

Anexo 4: Diagrama de clases del diseño

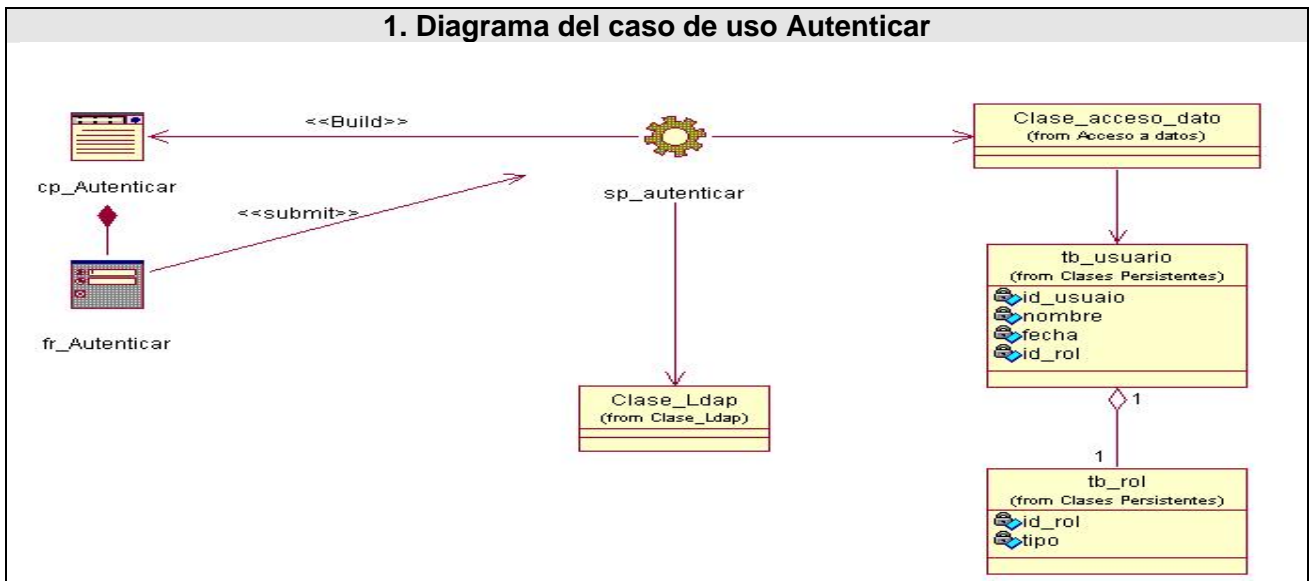


Figura 28: Diagrama de clases del diseño del caso de uso autenticar.

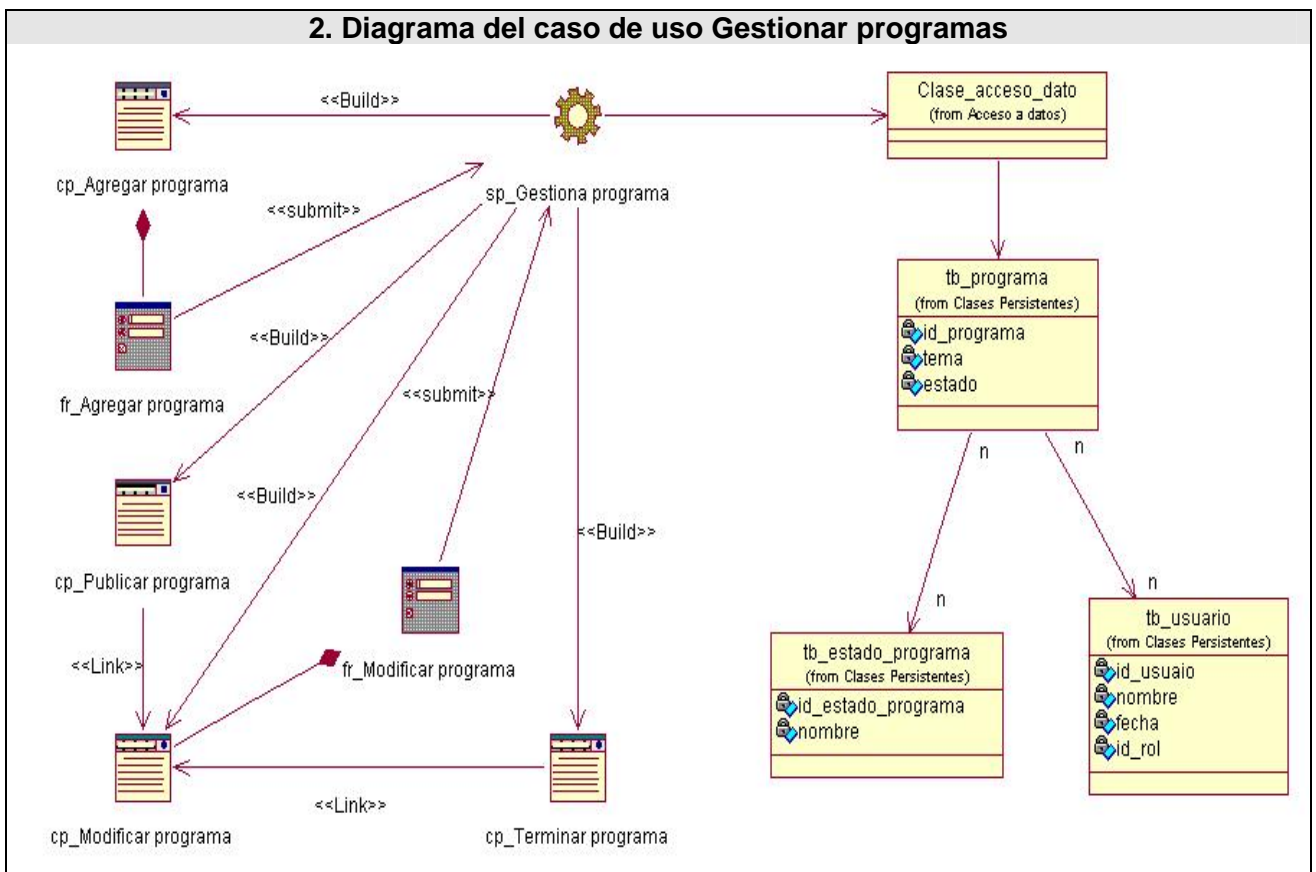


Figura 29: Diagrama de clases del diseño del caso de uso gestionar programas.

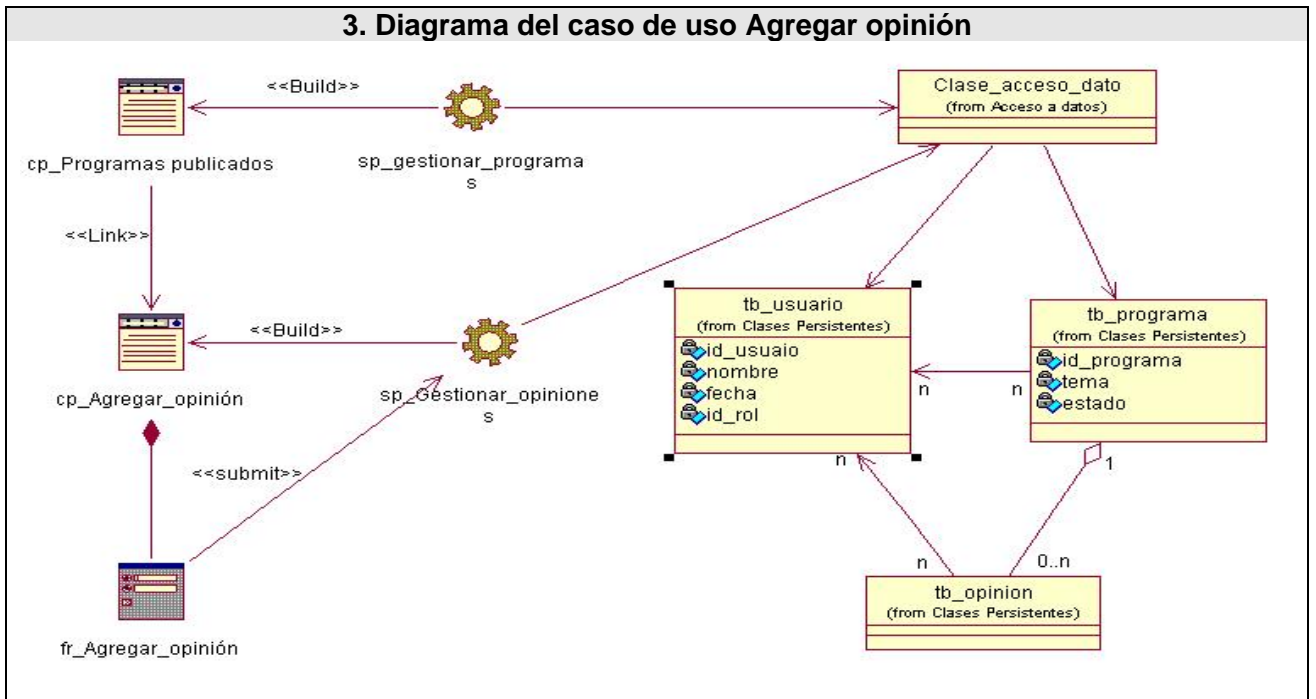


Figura 30: Diagrama de clases del diseño del caso de uso agregar opinión.

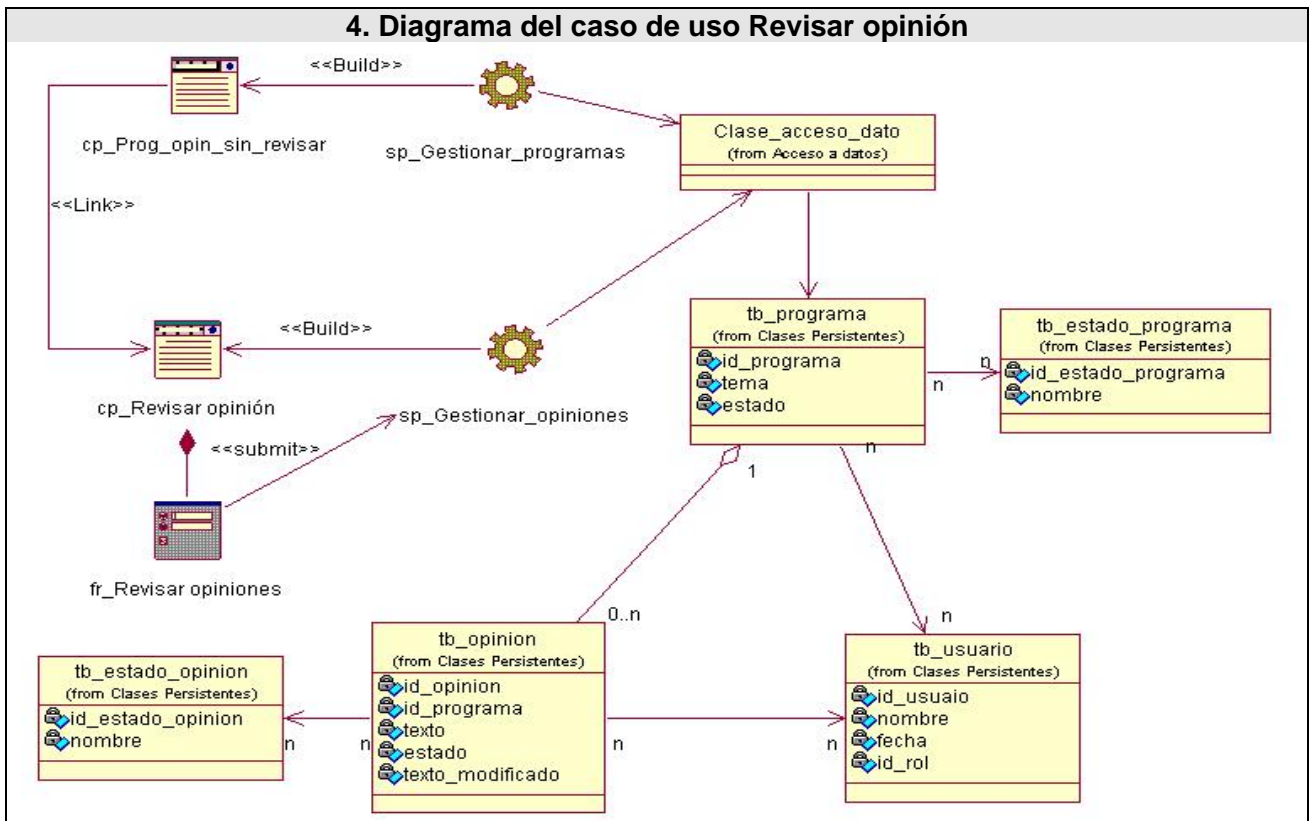


Figura 31: Diagrama de clases del diseño del caso de uso revisar opinión.

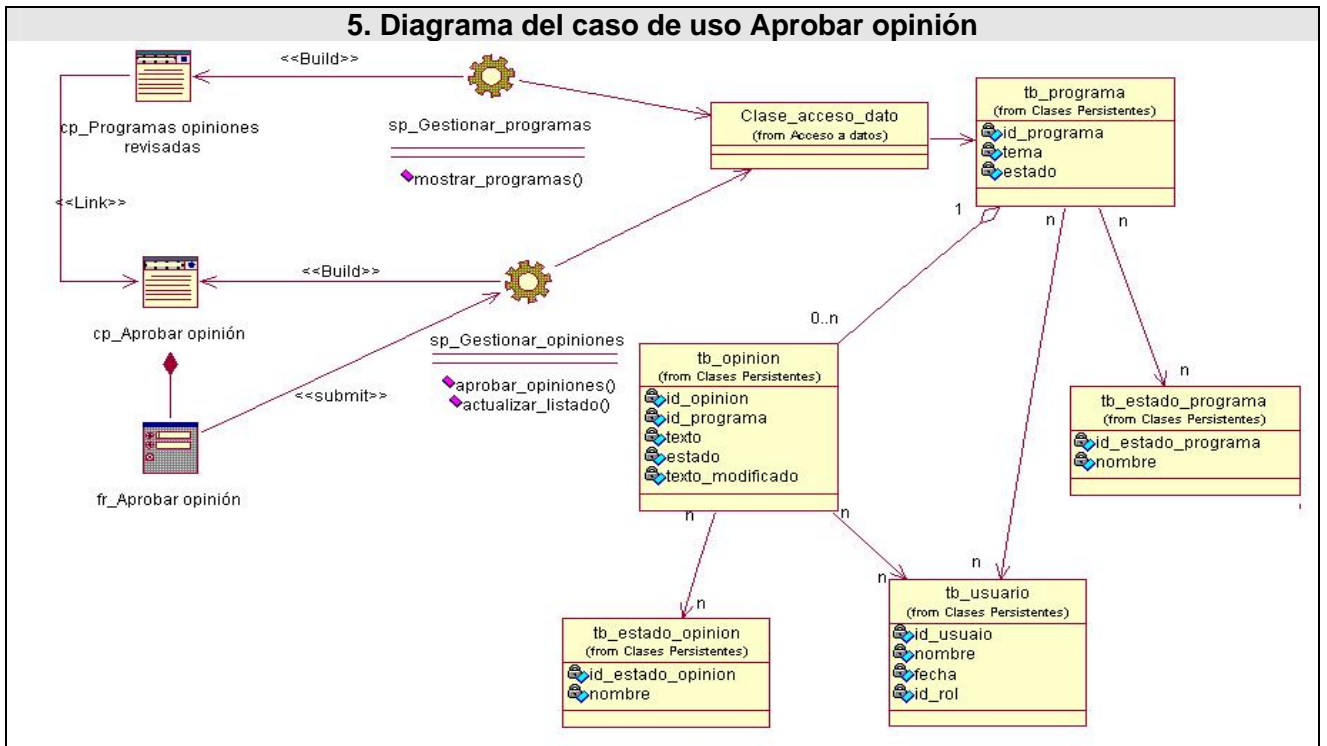


Figura 32: Diagrama de clases del diseño del caso de uso aprobar opinión.

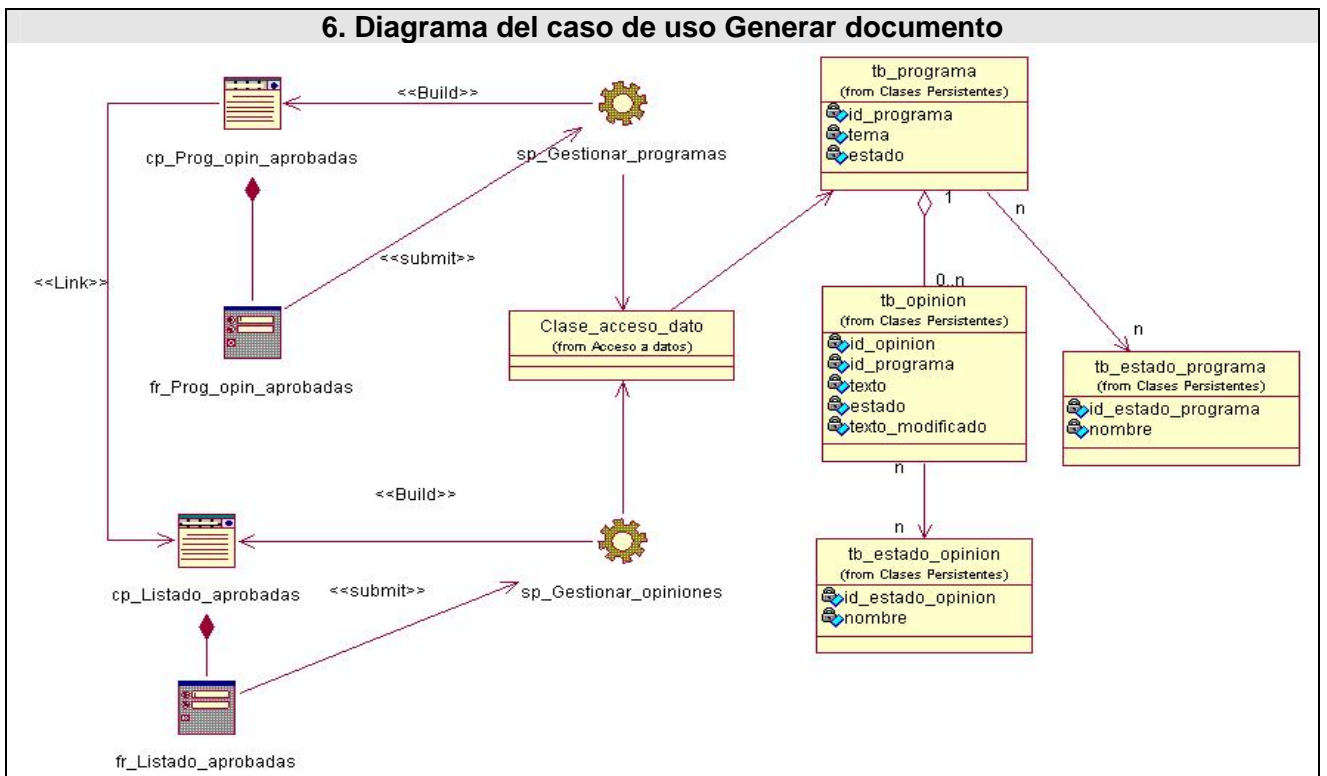


Figura 33: Diagrama de clases del diseño del caso de uso generar documento.

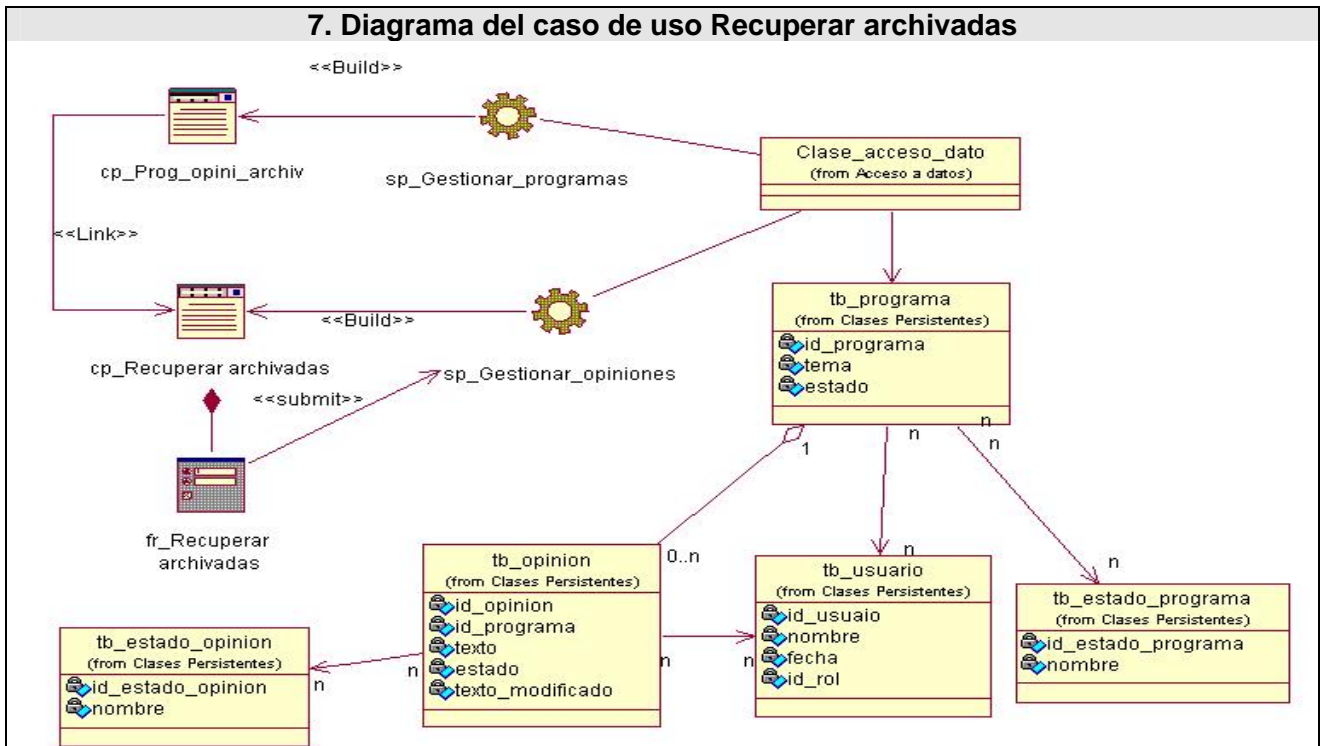


Figura 34: Diagrama de clases del diseño del caso de uso recuperar archivadas.

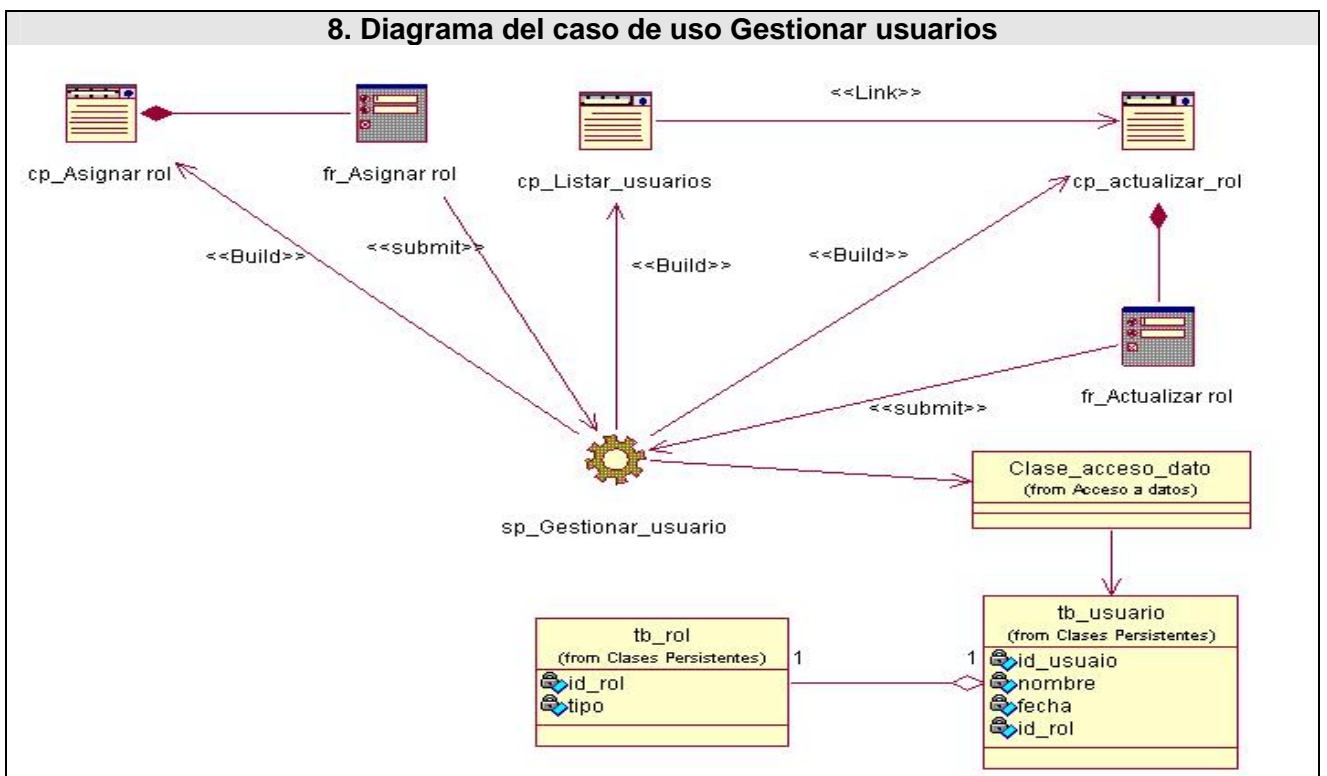


Figura 35: Diagrama de clases del diseño del caso de uso gestionar usuarios.

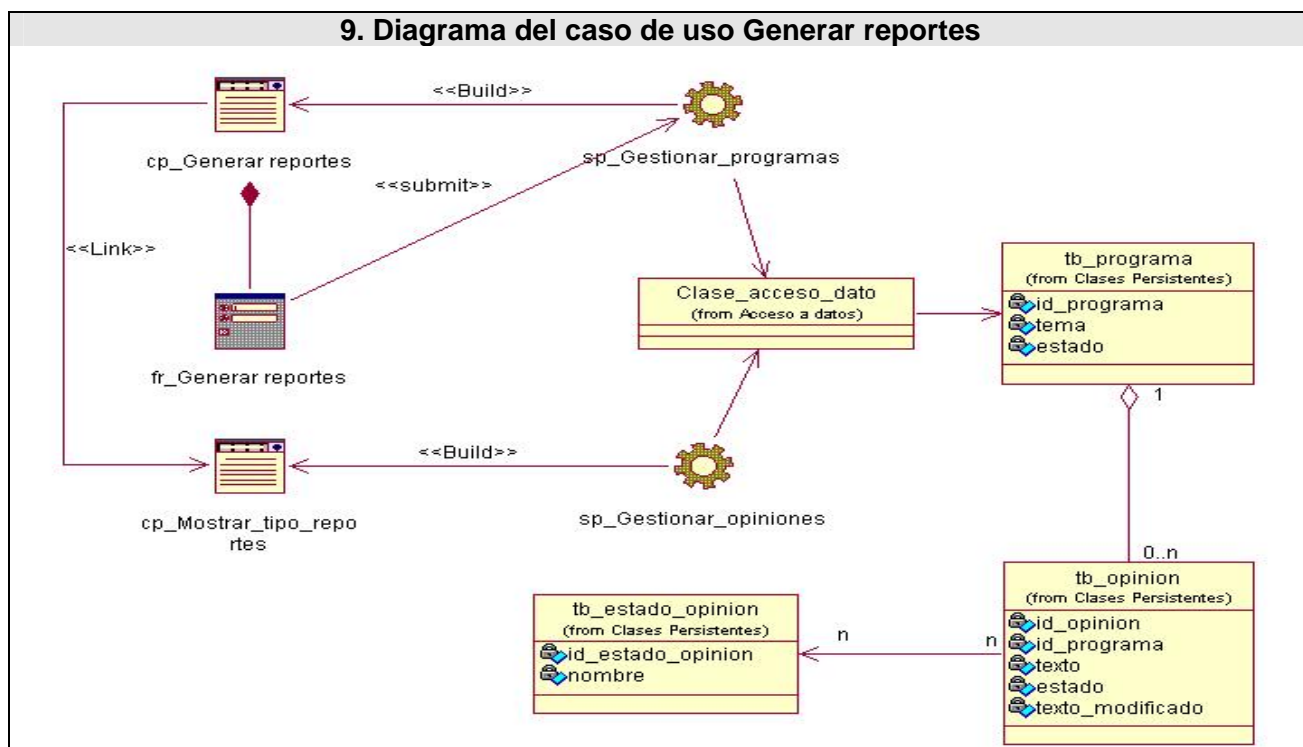


Figura 36: Diagrama de colaboración del escenario generar reportes.

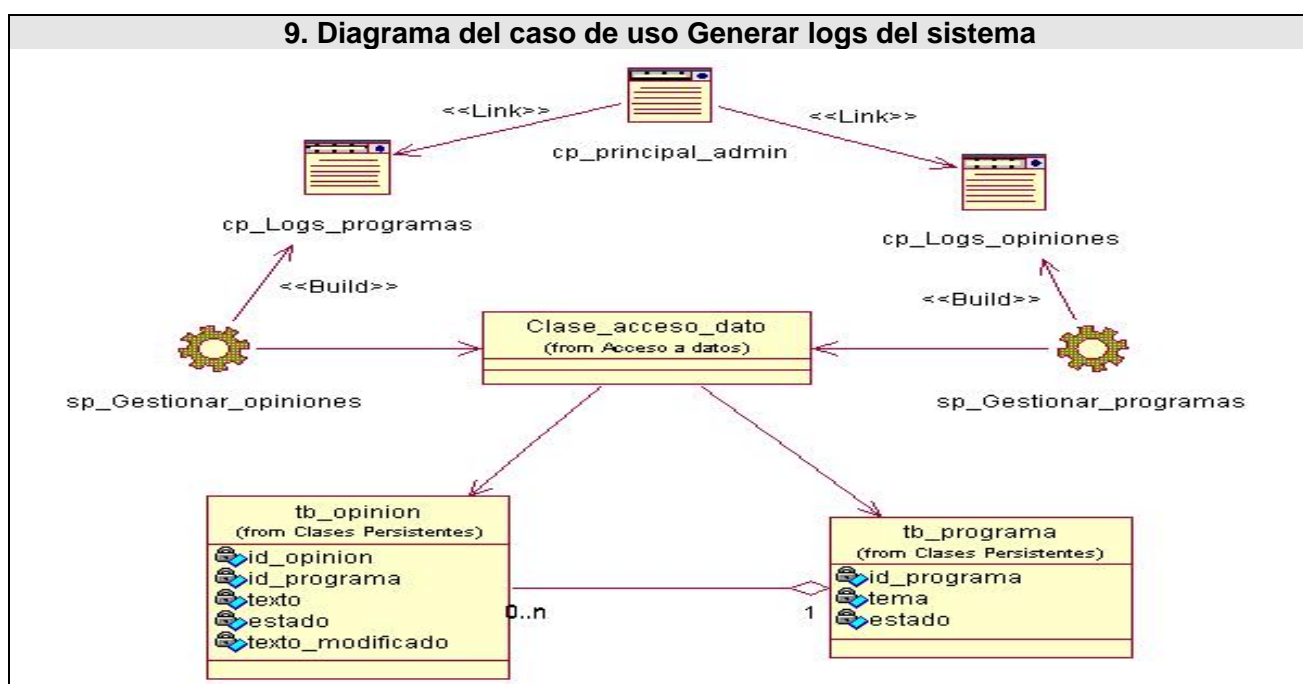


Figura 37: Diagrama de colaboración del escenario generar logs del sistema.

Anexo 5: Descripción de las clases del diseño

Clase Interfaz (Client page)

Tabla 22: CI_Autenticar

Nombre: Autenticar	
Tipo de clase: interfaz	
Para cada responsabilidad	
Nombre:	Login()
Descripción:	Verifica el usuario y la contraseña contra el directorio ldap.
Nombre:	Acceso_usuario()
Descripción:	Según los roles con que cuente el usuario da los permisos de acceso al sistema.

Tabla 23: CI_Agregar programa

Nombre: Agregar programa	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Agregar_programa (tema: Varchar)
Descripción:	Inserta un nuevo programa en la tabla.

Tabla 24: CI_Publicar programa

Nombre: Publicar programa	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Publicar_programa (id_programa: Serial, tema: Varchar)
Descripción:	Permite publicar el nombre del programa.

Tabla 25: CI_Modificar programa

Nombre: Modificar programa	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Modificar_programa ()
Descripción:	Modifica el texto del programa.

Tabla 26: CI_Terminar programa

Nombre: Terminar programa	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Terminar_programa (id_programa: Serial)
Descripción:	Cambia el estado del programa.

Tabla 27: CI_Insertar opinión

Nombre: Insertar opinión	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Insertar_opinión ()
Descripción:	Permite agregar nuevas opiniones.

Tabla 28: CI_Revisar opinión

Nombre: Revisar opinión	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Revisar_opinion()
Descripción:	Permite revisar las opiniones por programa.

Tabla 29: CI_Aprobar opinión

Nombre: Aprobar opinión	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Aprobar_opinion()
Descripción:	Permite aprobar las opiniones por programa.

Tabla 30: CI_Recuperar archivadas

Nombre: Recuperar archivadas	
Tipo de clase: interfaz	
Para cada responsabilidad:	

Nombre:	Recuperar archivadas()
Descripción:	Actualiza el estado de las opiniones archivadas.

Tabla 31: CI_Generar documento

Nombre: Generar documento	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Generar_documento()
Descripción:	Genera documento pdf.

Tabla 32: CI_Asignar rol

Nombre: Asignar rol	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Asignar_rol()
Descripción:	Asigna rol a un usuario.

Tabla 33: CI_Actualizar rol

Nombre: Quitar rol	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Actualizar_rol()
Descripción:	Actualiza el tipo de rol al usuario.

Tabla 34: CI_Generar Reportes

Nombre: Generar Reportes	
Tipo de clase: interfaz	
Para cada responsabilidad:	
Nombre:	Generar_reporte()
Descripción:	Genera reportes estadísticos por programa.

Clases controladoras (Server page)

Tabla 35: CC_Autenticar

Nombre: Autenticar	
Tipo de clase: controladora	
Atributo	Tipo
Nombre	varchar
Id_rol	integer
Para cada responsabilidad:	
Nombre:	Login(nombre:varchar, contraseña:varchar)
Descripción:	Verifica el usuario y la contraseña contra el directorio ldap.
Nombre:	Acceso_usuario(id_rol:integer)
Descripción:	Según los roles con que cuente el usuario da los permisos de acceso al sistema.
Nombre:	Insertar_usuario(nombre: Varchar, fecha: Varchar, id_rol: Integer, area: Varchar, tipo: Varchar)
Descripción:	Inserta un nuevo usuario.

Tabla 36: CC_Agregar programa

Nombre: Agregar programa	
Tipo de clase: controladora	
Atributo	Tipo
tema	Varchar
Para cada responsabilidad:	
Nombre:	Agregar_programa (tema: Varchar)
Descripción:	Inserta un nuevo programa en la tabla.

Tabla 37: CC_Publicar programa

Nombre: Publicar programa	
Tipo de clase: controladora	
Atributo	Tipo
Id_programa	Serial
Para cada responsabilidad:	
Nombre:	Publicar_programa (id_programa: Serial, tema: Varchar)

Descripción:	Permite publicar el nombre del programa.
---------------------	--

Tabla 38: CC_Modificar programa

Nombre: Modificar programa	
Tipo de clase: controladora	
Atributo	Tipo
Id_programa	Integer
tema	Varchar
Para cada responsabilidad:	
Nombre:	Modificar_programa (id_programa:Integer, tema: Varchar)
Descripción:	Modifica el texto del progama.

Tabla 39: CC_Terminar programa

Nombre: Terminar programa	
Tipo de clase: controladora	
Atributo	Tipo
Id_programa	Serial
Para cada responsabilidad:	
Nombre:	Terminar_programa (id_programa: Serial)
Descripción:	Cambia el estado del programa.

Tabla 40: CC_Insertar opinión

Nombre: Insertar opinión	
Tipo de clase: controladora	
Atributo	Tipo
id_programa	Integer
texto	Varchar
estado	Integer
Para cada responsabilidad:	
Nombre:	Insertar_opinión (id_programa: Integer, texto: Varchar)
Descripción:	Permite agregar nuevas opiniones.

Tabla 41: CC_Revisar opiniones

Nombre: Revisar opiniones	
Tipo de clase: controladora	
Atributo	Tipo
Id_opinión	Integer
texto	Varchar
Para cada responsabilidad:	
Nombre:	Revisar_opiniones (Id_opinión: Integer, texto: Varchar)
Descripción:	Permite revisar las opiniones por programa.

Tabla 42: CC_Aprobar opinión

Nombre: Aprobar opinión	
Tipo de clase: controladora	
Atributo	Tipo
Id_opinión	Integer
Texto_modificado	Varchar
Para cada responsabilidad:	
Nombre:	Aprobar_opinion(Id_opinión: Integer, texto_modificado: Varchar)
Descripción:	Permite aprobar las opiniones por programa.

Tabla 43: CC_Recuperar archivadas

Nombre: Recuperar archivadas	
Tipo de clase: controladora	
Atributo	Tipo
texto	Varchar
Para cada responsabilidad:	
Nombre:	Recuperar archivadas(id_opinion: Integer, estado: Varchar)
Descripción:	Actualiza el estado de las opiniones archivadas.

Tabla 44: CC_Generar documento

Nombre: Generar documento	
Tipo de clase: controladora	
Atributo	Tipo

id_opinion	Integer
estado	Varchar
Para cada responsabilidad:	
Nombre:	Generar_documento(texto: Varchar)
Descripción:	Genera documento pdf.

Tabla 45: CC_Asignar rol

Nombre: Asignar rol	
Tipo de clase: controladora	
Atributo	Tipo
nombre	Varchar
Para cada responsabilidad:	
Nombre:	Asignar_rol(nombre: Varchar)
Descripción:	Asigna rol a un usuario.

Tabla 46: CC_Quitar rol

Nombre: Quitar rol	
Tipo de clase: controladora	
Atributo	Tipo
Id_rol	Integer
Para cada responsabilidad:	
Nombre:	Quitar rol(id_rol: Integer)
Descripción:	Actualiza el tipo de rol al usuario.

Tabla 47: CC_Generar Reportes

Nombre: Generar Reportes	
Tipo de clase: controladora	
Atributo	Tipo
id_program	Integer
Para cada responsabilidad:	
Nombre:	Generar_reporte(id_program:Integer)
Descripción:	Genera reportes estadísticos por programa.

Clases entidad

Tabla 48: CE_tb_usuario

Nombre: tb_usuario	
Tipo de clase: entidad	
Atributo	Tipo
id_usuario	Serial
nombre	Varchar
fecha	Varchar
id_rol	Integer
area	Varchar
tipo	Varchar
Para cada responsabilidad:	
Nombre:	Acceso_usuario(id_rol: Integer)
Descripción:	Permite el acceso del usuario al sistema.
Nombre:	Asignar_rol(nombre: Varchar)
Descripción:	Asigna los permisos a un usuario.
Nombre:	Quitar_rol(id_usuario: Serial, nombre: Varchar)
Descripción:	Quita los permisos a un usuario.
Nombre:	Insertar_usuario(nombre: Varchar, fecha: Varchar, id_rol: Integer, area: Varchar, tipo: Varchar)
Descripción:	Inserta un nuevo usuario.

Tabla 49: CE_tb_programa

Nombre: tb_programa	
Tipo de clase: entidad	
Atributo	Tipo
id_programa	Serial
tema	Varchar
estado	Varchar
Para cada responsabilidad:	
Nombre:	Publicar_programa (id_programa: Serial, tema: Varchar)

Descripción:	Permite publicar el nombre del programa.
Nombre:	Agregar_programa (tema: Varchar)
Descripción:	Inserta un nuevo programa en la tabla.
Nombre:	Modificar_programa (id_programa: Serial, tema: Varchar)
Descripción:	Modifica el texto del programa.
Nombre:	Terminar_programa (id_programa: Serial)
Descripción:	Cambia el estado del programa.

Tabla 50: CE_tb_estado_programa

Nombre: tb_estado_programa	
Tipo de clase: entidad	
Atributo	Tipo
id_estado_programa	Serial
nombre	Varchar
Para cada responsabilidad:	
Nombre:	Agregar (nombre: Varchar)
Descripción:	Adiciona un nuevo estado.
Nombre:	Eliminar (id_estado_programa: Serial)
Descripción:	Elimina el estado del programa.

Tabla 51: CE_estado_programa

Nombre: estado_programa	
Tipo de clase: entidad	
Atributo	Tipo
id_estado_programa	Integer
id_programa	Integer
fecha	Varchar
Para cada responsabilidad:	
Nombre:	Agregar_estado (id_estado_programa: Integer, id_programa: Integer)
Descripción:	Adiciona un nuevo estado de programa.

Tabla 52: CE_tb_opinión

Nombre: tb_opinión	
Tipo de clase: entidad	
Atributo	Tipo
id_opinión	Serial
id_programa	Integer
texto	Varchar
estado	Varchar
texto_modificado	Varchar
Para cada responsabilidad:	
Nombre:	Insertar_opinión (id_programa: Integer, texto: Varchar)
Descripción:	Agregar una nueva opinión.
Nombre:	Buscar_opinión_estado (id_opinión: Serial, estado: Varchar)
Descripción:	Listar opiniones por estado.
Nombre:	Recuperar_archivadas(id_opinión: Serial, estado: Varchar)
Descripción:	Actualiza el estado de las opiniones archivadas.
Nombre:	Generar_documento(id_opinión: Serial, texto: Varchar)
Descripción:	Genera documento pdf.

Tabla 53: CE_tb_estado_opinión

Nombre: tb_estado_opinión	
Tipo de clase: entidad	
Atributo	Tipo
id_estado_opinión	Serial
nombre	Varchar
Para cada responsabilidad:	
Nombre:	Agregar (nombre: Varchar)
Descripción:	Adiciona un nuevo estado.
Nombre:	Eliminar (id_estado_opinión: Serial)
Descripción:	Elimina el estado de la opinión.

Tabla 54: CE_estado_opinión

Nombre: estado_opinión

Tipo de clase: entidad	
Atributo	Tipo
id_opinión	Integer
id_estado_opinión	Integer
fecha	Varchar
Para cada responsabilidad:	
Nombre:	Agregar_estado (id_estado_opinión: Integer, id_opinión: Integer)
Descripción:	Adiciona un nuevo estado de opinión.

Tabla 55: CE_tb_rol

Nombre: tb_rol	
Tipo de clase: entidad	
Atributo	Tipo
id_rol	Serial
tipo	Varchar
Para cada responsabilidad:	
Nombre:	Agregar_rol (tipo: Varchar)
Descripción:	Adiciona un nuevo estado de opinión.

Tabla 56: CE_acción_programa

Nombre: acción_programa	
Tipo de clase: entidad	
Atributo	Tipo
id_usuario	Integer
id_programa	Integer
Acción	Varchar
fecha	Varchar
Ip_pc	Varchar
Para cada responsabilidad:	
Nombre:	Agregar_accion_programa (id_usuario: Integer, id_programa: Integer)
Descripción:	Adiciona una nueva acción realizada sobre un programa.

Tabla 57: CE_ acción_opinión

Nombre: acción_opinión	
Tipo de clase: entidad	
Atributo	Tipo
id_opinión	Integer
id_usuario	Integer
Acción	Varchar
fecha	Varchar
Ip_pc	Varchar
Para cada responsabilidad:	
Nombre:	Agregar_accion_opinion (id_opinión: Integer, id_usuario: Integer)
Descripción:	Adiciona una nueva acción realizada sobre una opinión.

Anexo 6: Descripción de las tablas de la Base de datos.

Tabla 58: tb_programa

Nombre: tb_programa		
Descripción: registra los datos referentes a los programas, que sean insertados en el sistema.		
Atributo	Tipo	Descripción
id_programa	Serial	Un identificador autonumérico de los programas.
tema	Varchar	El nombre del programa.
estado	Varchar	Me permite saber si el programa está sin publicar, publicado o terminado.

Tabla 59: tb_estado_programa

Nombre: tb_estado_programa		
Descripción: registra los diferentes tipos de estados predefinidos.		
Atributo	Tipo	Descripción
id_estado_programa	Serial	Un identificador autonumérico del estado de un programa.
nombre	Varchar	Los nombres de los estados: sin publicar, publicado o terminado.

Tabla 60: estado_programa

Nombre: estado_programa		
Descripción: registra los diferentes estados por los que pasan los programas y la fecha.		
Atributo	Tipo	Descripción
id_estado_programa	Integer	Un identificador entero del estado de un programa.
id_programa	Integer	Un identificador entero de los programas.
fecha	Varchar	Almacena la fecha y la hora en que ocurre cada estado del programa.

Tabla 61: tb_opinión

Nombre: tb_opinión		
Descripción: registra todos los datos referentes a las opiniones emitidas.		

Atributo	Tipo	Descripción
id_opinión	Serial	Un identificador autonumérico de la opinión.
id_programa	Integer	Un identificador entero de los programas.
texto	Varchar	El texto de la opinión.
estado	Varchar	Pueden ser los estados: sin editar, editada, revisada y archivada.
texto_modificado	Varchar	Actualiza el texto de la opinión después de ser editado.

Tabla 62: estado_opinión

Nombre: estado_opinión		
Descripción: registra los diferentes estados por los que pasan las opiniones y la fecha.		
Atributo	Tipo	Descripción
id_opinión	Integer	Un identificador entero de la opinión.
id_estado_opinión	Integer	Un identificador entero del estado de la opinión.
fecha	Varchar	Almacena la fecha y la hora en que ocurre cada estado de la opinión.

Tabla 63: tb_estado_opinión

Nombre: tb_estado_opinión		
Descripción: registra los diferentes tipos de estados predefinidos.		
Atributo	Tipo	Descripción
id_estado_opinión	Serial	Un identificador autonumérico del estado de la opinión.
nombre	Varchar	Los nombres de los estados: sin editar, editada, revisada y archivada.

Tabla 64: tb_usuario

Nombre: tb_usuario		
Descripción: registra los datos de los usuarios.		
Atributo	Tipo	Descripción
id_usuario	Serial	Un identificador autonumérico del usuario.
nombre	Varchar	Es el usuario del dominio UCI.

fecha	Varchar	Almacena la fecha y la hora.
id_rol	Integer	Es el identificador entero de cada rol.

Tabla 65: tb_rol

Nombre: tb_rol		
Descripción: registra los diferentes tipos de roles predefinidos.		
Atributo	Tipo	Descripción
id_rol	Serial	Un identificador autonumérico del rol.
tipo	Varchar	Recoge la clasificación del rol.

Tabla 66: acción_programa

Nombre: acción_programa		
Descripción: registra las diferentes acciones que son realizadas por usuarios sobre los programas.		
Atributo	Tipo	Descripción
id_usuario	Integer	Un identificador entero del usuario.
id_programa	Integer	Un identificador entero del programa.
Acción	Varchar	Recoge la clasificación del acceso.
fecha	Varchar	Recoge la fecha y la hora en que se realiza la acción.
Ip_pc	Varchar	Recoge el IP de la PC desde donde se realiza la acción.

Tabla 67: acción_opinión

Nombre: acción_opinión		
Descripción: registra las diferentes acciones que son realizadas por usuarios sobre las opiniones.		
Atributo	Tipo	Descripción
id_opinión	Integer	Un identificador entero de la opinión.
id_usuario	Integer	Un identificador entero del usuario.
Acción	Varchar	Recoge la clasificación del acceso.
fecha	Varchar	Recoge la fecha y la hora de la acción.
Ip_pc	Varchar	Recoge el IP de la PC desde donde se realiza la acción.

Glosario:

ASP: acrónimo inglés de Active Server Pages. Lenguaje de scripts servidor, creado por Microsoft para páginas web.

Artefactos: información que se utiliza o produce mediante un proceso de software.

CASE: acrónimo inglés de Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador.

Comandante: aquel militar de más alto cargo.

Componentes software: procesos y objetos que se ejecutan en ellos.

Consejo de Estado: es el supremo órgano consultivo del Gobierno.

CU: abreviatura de Caso de Uso.

DBMS: acrónimo inglés de Data Base Management System, Sistema Administrador de Bases de Datos.

FEU: Federación Estudiantil Universitaria.

FTP: acrónimo inglés de File Transfer Protocol, es un protocolo de transferencia de ficheros.

Hosting: conocido como alojamiento, hospedaje web o web site hosting. Consiste en alojar, servir, y mantener archivos para uno o más sitios web.

IBM: acrónimo inglés de International Business Machines, empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática.

IMAP: acrónimo inglés de Internet Message Access Protocol. Es un protocolo de red de acceso a mensajes electrónicos.

Internet: es un método de interconexión de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP y garantiza que redes físicas heterogéneas funcionen como una red única.

IP: acrónimo inglés de Internet Protocol. El Protocolo de Internet es un protocolo orientado de datos, usado tanto por el origen como por el destino para la comunicación de estos a través de una red (Internet) de paquetes conmutados.

Java: Lenguaje de programación orientado a objetos.

LDAP: acrónimo inglés de Lightweight Directory Access Protocol. Consiste en base de datos accesible mediante la red donde una organización guarda información sobre sus usuarios y los privilegios que poseen.

Log: es un archivo creado por el servidor donde se registran las acciones que los usuarios generan en la web.

Máxima Dirección del país: el término se utiliza cuando se refiere al Comandante en Jefe Fidel Castro y el consejo de estado

MVCC: acrónimo inglés de Multi-Version Concurrency Control. Acceso concurrente multiversión.

OLTP: procesamiento en línea de transacciones.

Open source: Código abierto.

PC: acrónimo inglés de Personal Computer, término genérico para referirse a las microcomputadoras.

PCC: Partido Comunista de Cuba.

Perl: Lenguaje de Programación Práctico para la Extracción e Informe.

Rational: Herramientas de desarrollo de software.

RF: Requerimientos funcionales.

SGBD: Sistema de gestor de Base de Datos.

SGO: Sistema de Gestión de Opiniones.

SMTP: acrónimo inglés de Simple Mail Transfer Protocol. Protocolo simple de transferencia de correo electrónico.

SNMP: acrónimo inglés de Simple Network Management Protocol. Protocolo Simple de administración de red.

SO: Sistema Operativo.

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

Symbian: sistema operativo que fue producto de la alianza de varias empresas de telefonía celular.

UCI: Universidad de las Ciencias Informáticas.

UJC: Unión de Jóvenes Comunista.

UML: Lenguaje Unificado de Modelación.

URL: significa Uniform Resource Locator o localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

WDDX: Intercambio Web de Datos Distribuidos.

XML: acrónimo inglés de eXtensible Markup Language, lenguaje de marcado extensible. Lenguaje utilizado en la Web que complementa y mejora al HTML.