



FACULTAD 4

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

Desarrollo del módulo para la gestión de notificaciones del Portal Educativo

Autores: Kirenia Gross Laugart

Liliana Martínez Sanabria

Tutor: Ing. Javier Soler Martín

La Habana 2013

Año 55 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaramos que somos los autores de este trabajo y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2013.

AUTOR

Kirenia Gross Laugart

AUTOR

Liliana Martínez Sanabria

TUTOR

Ing. Javier Soler Martín

RESUMEN

Los sistemas de notificaciones tienen como objetivo fundamental mantener informados en todo momento a los clientes de un sitio determinado en relación a las actividades que realizan; además, ayudan en la generación y envío de información a través de una confirmación, resultando ser tareas sencillas de realizar, debido a las herramientas que desarrollan este tipo de servicios. En el entorno virtual del Sistema de Apoyo a la Municipalización en el Ministerio del Poder Popular para la Educación Universitaria en Venezuela, la gama de actores requieren un sistema de alertas o notificaciones configurables y adaptables a su entorno de vida particular. La propuesta de una solución informática que contribuyera a mejorar el proceso de notificaciones dentro del Ministerio del Poder Popular para la Educación Universitaria en Venezuela, constituyó el objetivo general de este trabajo. Considerándose las tecnologías y herramientas establecidas dentro de la universidad para la elaboración de la misma, además de analizarse las características que poseen estos sistemas existentes a nivel mundial.

Palabras Claves: alertas, notificaciones, sistema.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	6
1.1 Introducción	6
1.2 Portales Educativos	6
1.2.1 Características de un portal educativo	7
1.2.2 Ventajas de los portales educativos	7
1.3 Plataformas e-learning	8
1.3.1 Sistemas de gestión del aprendizaje	8
1.3.2 Sistemas de gestión de contenidos educativos	9
1.3.3 Repositorio de objetos de aprendizaje	9
1.4 Interoperabilidad	9
1.5 Servicio de envío de correo electrónico	10
1.5.1 Simple Mail Transfer Protocol (SMTP)	10
1.5.2 Post Office Protocol (POP)	11
1.5.3 Internet Message Access Protocol (IMAP)	11
1.6 Bibliotecas para el manejo de correos	12
1.6.1 Phpmailer	12
1.6.2 Swift Mailer	12
1.7 Sistemas de notificaciones	13
1.8 Canales de redifusión	14
1.8.1 RSS	14
1.8.2 ATOM	15
1.9 Sistemas de recomendación	15
1.9.1 Clasificación de los sistemas de recomendación	16
1.9.1.1 Recomendaciones basadas en contenido	16
1.9.1.2 Sistemas basados en filtrado colaborativo	17
1.9.1.3 Sistemas de recomendación híbridos	18
1.10 Métodos, lenguajes y herramientas de desarrollo.	19
1.10.1 Metodología de desarrollo. RUP	19
1.10.2 Lenguaje de modelado	21
1.10.3 Entorno integrado de desarrollo. Netbeans	21
1.10.4 Herramientas case. Visual Paradigm For UML 8.0	22
1.10.5 Lenguajes de programación	23

1.10.5.1 JavaScript (JSCRIPT)	23
1.10.5.2 Procesador de hipertexto (PHP)	24
1.10.5.3 XML	24
1.10.6 Sistemas de gestión de contenidos. Drupal	24
1.10.7 Desarrollo de módulos en Drupal	27
1.10.8 Sistema gestor de base de datos. Postgresql	28
1.10.9 Servidor web. Apache	29
1.11 Conclusiones del capítulo	30
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	31
2.1 Introducción	31
2.2 Descripción de los procesos de negocio	31
2.3 Modelo de dominio	32
2.4 Propuesta del sistema	32
2.5 Requisitos	36
2.5.1 Requisitos funcionales	36
2.5.2 Requisitos no funcionales	38
2.5.2.1 Usabilidad	38
2.5.2.2 Eficiencia	40
2.5.2.3 Restricciones de diseño	40
2.6 Modelo de casos de uso	41
2.6.1 Actores del sistema	41
2.6.2 Patrones de casos de uso	41
2.6.3 Diagrama de casos de uso de sistema	42
2.6.4 Descripciones de casos de uso	43
2.7 Conclusiones del capítulo	47
CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA	48
3.1 Introducción	48
3.2 Modelo de análisis	48
3.2.1 Diagrama de clase del análisis	48
3.2.2 Diagramas de colaboración	49
3.3 Modelo de diseño	50
3.3.1 Aplicación de patrones de diseño	51
3.3.2 Diagramas de clases de diseño	51
3.3.3 Modelo de datos	52

3.4	Conclusiones del capítulo	54
CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBAS		55
4.1	Introducción	55
4.2	Estándares de codificación	55
4.2.1	Etiquetas de apertura y cierre de php	55
4.2.2	Operadores	56
4.2.3	Uso de comillas	56
4.2.4	Uso de punto y coma (;) en código php	56
4.2.5	Funciones	56
4.2.6	Arrays	57
4.2.7	Nombres de archivos	57
4.3	Modelo de implementación	57
4.3.1	Diagrama de componentes	57
4.3.2	Diagrama de despliegue	60
4.4	Pruebas	61
4.5	Pruebas de caja negra	62
4.6	Resultados de las pruebas	63
4.7	Conclusiones del capítulo	64
CONCLUSIONES		66
RECOMENDACIONES		67
BIBLIOGRAFÍA		68
ANEXOS		71

ÍNDICE DE FIGURAS

Figura 1 Módulos en Drupal tomado del libro Pro Drupal 7 Development	27
Figura 2 Modelo de dominio	32
Figura 3 Diagrama de actividades del sistema de recomendación	36
Figura 4 Diagrama de casos de uso	43
Figura 5 DCD Gestionar suscripción a contenido	52
Figura 6 Modelo de datos	53
Figura 7 Diagrama de despliegue	61
Figura 8 Cantidad de no conformidades detectadas en cada iteración	63

INTRODUCCIÓN

El desarrollo de las Tecnologías de la Información y las Telecomunicaciones (TIC), ha contribuido a mejorar todas las esferas de la sociedad. Los sistemas educativos de todo el mundo se enfrentan al reto de utilizar las TIC con el objetivo de proveer a los estudiantes, herramientas y conocimientos necesarios en el siglo XXI. Las mismas brindan mejoras en el proceso de enseñanza-aprendizaje, obligando al escolar a tomar decisiones, escoger, seleccionar y filtrar información. Entre ellas están las posibilidades de interacción que ofrecen, la búsqueda y replanteamiento continuo de contenidos y procedimientos, así como el aumento de la implicación del alumnado en sus tareas y desarrollo de iniciativa de los mismos. (1)

Un auténtico ejemplo del desarrollo de las TIC en la educación es el surgimiento de una novedosa y avanzada forma de aprendizaje en línea, que permite la informatización de los procesos de formación, capacitación e integración de diferentes aplicaciones. Fruto de esta informatización surge el aprendizaje electrónico o e-learning.

Como alternativa al e-learning y la enseñanza tradicional aparece el b-learning que es la abreviatura de *Blended Learning*, expresión del inglés que, en términos de enseñanza virtual, se traduce como "Formación Combinada" o "Enseñanza Mixta". Se trata de una modalidad semipresencial de estudios, que incluye tanto formación no presencial (cursos online (en línea), conocidos genéricamente como e-learning) como formación presencial.

Hoy día el b-learning ha tenido mayor aceptación que el e-learning. Algunos de los beneficios que presenta el mismo es que se puede enriquecer de los abundantes materiales que existen en la red. No obstante el b-learning no consiste en colocar más materiales, sino utilizar de forma apropiada los ya existentes. (2)

Desde los inicios del aprendizaje electrónico, la interoperabilidad entre las aplicaciones ha sido un factor clave, lo que permite intercambiar información con diferentes funcionalidades y objetivos. Dentro de un entorno e-learning o b-learning existen diversos tipos de herramientas, como los Sistemas de Gestión del Aprendizaje (LMS) por sus siglas en inglés *Learning Management System*, Sistemas de Gestión de

Contenido Educativo (LCMS), los Repositorios Digitales (RD), entre otros. Estas aplicaciones pueden trabajar independientes, pero sólo cuando entre ellas existe un flujo de información, en otras palabras cuando son interoperables entre sí, es que alcanzan su verdadero potencial formativo.

Los portales educativos tienen una enorme capacidad para integrar las tecnologías de la información y las comunicaciones en el ámbito educativo. Los mismos jugando el rol de los LCMS, ponen a disposición de maestros y estudiantes recursos digitales desarrollados con intencionalidad educativa que permiten un tratamiento de las informaciones no viables en otros soportes y propician el espacio para la creación de comunidades virtuales académicas.

En la actualidad la expansión del Internet ha permitido la publicación de la información en portales web, que en ocasiones llegan a convertirse en redes sociales, estas permiten a los usuarios interactuar, intercambiar y publicar datos concernientes a ellos, así como, hechos y eventos que se realizan.

A raíz del flujo de información que presentan los portales educativos se hace necesario el uso de los sistemas de notificaciones, los cuales brindan la posibilidad de coordinar la información de manera rápida y segura entre diferentes áreas, personas u organismos que interactúen con ellos. Los sistemas de notificaciones tienen como objetivo fundamental mantener informados en todo momento a los clientes de un sitio determinado en relación a las actividades; además, ayudan en la generación y envío de información a través de una confirmación, resultando ser tareas sencillas de realizar, debido a las herramientas que desarrollan este tipo de servicios.

En la Universidad de las Ciencias Informáticas (UCI) con el objetivo de organizar la labor productiva fueron creados varios centros de desarrollo, entre los que se encuentra el Centro de Tecnologías para la Formación (FORTES). Este tiene como misión desarrollar tecnologías que permitan ofrecer servicios y productos, para brindar soluciones de formación aplicando las TIC, a todo tipo de instituciones con diferentes modelos de formación y condiciones tecnológicas.

Uno de los proyectos desarrollados en el centro FORTES es el “Portal Educativo”, proyecto realizado para el Ministerio del Poder Popular de la República Bolivariana de Venezuela orientado a la Educación Universitaria.

Dicho ministerio para mejorar el trabajo educativo orientado a la Educación Universitaria Venezolana, hace uso de los espacios virtuales, proveyendo a la comunidad universitaria de un espacio donde el intercambio y el trabajo colaborativo entre estudiantes, profesores e investigadores es cada vez más fácil y frecuente. Esto, apoyándose en las metodologías y tecnologías pedagógico-informáticas, que permitan gestionar una red de Comunidades Virtuales. Dicha comunidad está organizada por categorías, y permiten mediante un muro, compartir recursos educativos, hacer debates, realizar sondeos o encuestas y hacer promociones.

En el entorno virtual del Sistema de Apoyo a la Municipalización en el Ministerio del Poder Popular para la Educación Universitaria en Venezuela (SIAMU) los profesores que integran los equipos de trabajo no poseen un espacio para el desarrollo de los procesos que sustentan sus Comunidades Virtuales. Sólo existen incipientes herramientas brindadas por el Entorno Virtual de Apoyo (Moodle) donde se simulan estos espacios virtuales con la carencia de instrumentos capaces de engranar los espacios de trabajo colaborativo necesarios por las comunidades. Los procesos que tienen vida en el modelo de Portal Educativo, específicamente los de trabajo en grupo por parte de las Comunidades Virtuales, generan grandes flujos de información y otra gran parte es modificada. La gama de actores que participarán en estos espacios de trabajo requieren un sistema de alertas o notificaciones configurables y adaptables a su entorno de vida particular. Esta necesidad data del actual e inadecuado uso de Moodle, como base informática para sustentar las Comunidades Virtuales existentes.

El diseño del Portal Educativo cuenta con siete módulos que son: Gestión de usuarios, Comunidades Virtuales, Avisos, Eventos y boletines, Interoperabilidad, Mensajería y Notificaciones y Búsqueda, los que permiten la interacción del usuario con el sistema. Los mismos se desarrollan con las ideas más avanzadas de la programación web. Es una herramienta de libre acceso, cuenta con el respaldo financiero y político para su desarrollo y se hace sobre la base de las metodologías pedagógicas del ministerio de educación universitaria de Venezuela.

Dada la situación anteriormente descrita, se tiene cómo **problema a resolver**: ¿Cómo facilitar la comunicación a los usuarios que interactúan con un Sistema de Gestión de Contenido Educativo?

Teniendo como **objetivo general**: Desarrollar un sistema de notificaciones que facilite la comunicación a los usuarios que interactúan con un Sistema de Gestión de Contenido Educativo.

Se define como **objeto de estudio**: Las herramientas de trabajo colaborativo en los Sistemas de Gestión de Contenido Educativo.

Enmarcándose en el **campo de acción**: Herramientas de comunicación y gestión de suscripciones, en espacios de trabajo colaborativo en los Sistemas de Gestión de Contenido Educativo.

Como **Hipótesis** de la presente investigación se plantea: Si se desarrolla un sistema de notificaciones para un Sistema de Gestión de Contenido Educativo entonces se facilitará la comunicación a los usuarios que interactúan en el mismo.

Como **objetivos específicos** se definen los siguientes:

1. Elaborar el marco teórico-conceptual de la investigación a partir de una búsqueda y revisión bibliográfica.
2. Realizar el análisis y diseño del módulo Notificaciones.
3. Implementar el módulo Notificaciones.
4. Realizar pruebas internas para garantizar la funcionalidad del módulo.

Las **tareas de investigación** definidas para dar cumplimiento a los objetivos específicos fueron las siguientes:

1. Selección de las herramientas y tecnologías a usar en el ciclo de desarrollo e investigación del módulo Notificaciones.
2. Identificación de los principales componentes y características del módulo Notificaciones.
3. Análisis y diseño del módulo Notificaciones.
4. Implementación del módulo Notificaciones.
5. Realización de pruebas internas para garantizar la funcionalidad del sistema.

Para el desarrollo de la presente investigación se emplearon los siguientes métodos científico-teóricos:

- Histórico-lógico: Utilizado al tener en cuenta la caracterización de la evolución histórica de los sistemas de recomendación.
- Analítico-sintético: Se evidencia cuando se realiza un análisis de toda la teoría y documentación, que permiten la extracción de los elementos fundamentales relacionados con las herramientas que contribuyen al aprendizaje interactivo.

La presente investigación está organizada en cuatro capítulos:

Capítulo 1: Fundamentación teórica. Se hace un análisis de los principales conceptos relacionados con el objeto de estudio. Además, se realiza un análisis del estado del arte, de las herramientas, las distintas técnicas, tecnologías y metodologías utilizadas.

Capítulo 2: Concepción y características del sistema. Se describe la propuesta de solución del sistema y el modelo de dominio, donde se capturan los objetos importantes del entorno en el cual será empleado el sistema. Se identifican además los requerimientos funcionales y los no funcionales, se elabora el modelo de casos de uso y las descripciones de cada uno de ellos.

Capítulo 3: Análisis y diseño. Se describe la solución que se propone a partir de los diagramas de clases del análisis asociado a las funcionalidades del sistema y se representan los diagramas de clases del diseño que reflejan una vista interna del sistema.

Capítulo 4: Implementación y pruebas. Se realiza el diagrama de componentes y de despliegue, obteniéndose una descripción de la implementación del sistema. También se describen los casos de pruebas por los cuales los desarrolladores verifican el producto, además se selecciona el método de prueba y la técnica de prueba que se va a utilizar.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se ofrece el marco teórico en el que se desarrolla la presente investigación, creando un sólido basamento que será el preámbulo de los próximos capítulos. Para ello, se realiza un estudio de la situación actual en la que se encuentran los portales educativos, lo que sirve como entrada para el análisis del módulo que se desea implementar. También se realiza un estudio del ambiente de desarrollo seleccionado y establecido por la dirección del proyecto, y se especifican las principales características de la metodología, lenguajes y herramientas establecidas que se utilizan en la implementación de la solución.

1.2 Portales Educativos

Un portal educativo se define como un sitio Web en el que se incluyen tanto contenidos como servicios y que se brindan a la comunidad educativa en su conjunto (docentes, alumnos/as, familias, etc.). La mayoría de los portales educativos se caracterizan por su desinterés económico siendo patrocinados por instituciones, ministerios y empresas del sector educativo. La naturaleza “En línea (Online)” de estos almacenes de información, permite al usuario acceder a sus contenidos de forma ágil, sencilla e instantánea lo que sumado a la gran calidad del producto final, descubren que este tipo de estructuras de información constituyan una de las mejores soluciones en la construcción del conocimiento.

El origen de los portales educativos es una especificación de los portales de carácter generalista. El porqué de su advenimiento podría entenderse a partir de la desalentadora e incesante búsqueda de información específica en Internet y la consecuente necesidad del usuario de encontrar mayor precisión y mayor diversidad sobre un tema concreto. El que en un sitio único se pudiera encontrar el mayor número de contenidos y servicios relacionados con el trabajo, ocio o cualquier interés personal, provocó el espaldarazo definitivo en la aparición de los portales, y en nuestro caso, los de contenidos educativos. Prácticamente desde la implantación de Internet y su uso generalizado entorno al año 1995, comenzaron a aparecer los primeros portales. (3)

1.2.1 Características de un portal educativo

Entre las características que definen a un portal educativo están las de crear una fidelización con el usuario de tal manera que a través de los diferentes recursos y servicios que se le ofrecen: informativos, recursos didácticos, formativos para el profesorado, canales de comunicación, asesoramiento, instrumentos para la comunicación, etc., se puedan establecer Comunidades Virtuales que vayan agrandando y potenciando las posibilidades del medio gracias a las continuas evaluaciones que sobre aspectos técnicos, estéticos, pedagógicos y funcionales se realicen. (4)

De manera general un portal educativo simplifica la planificación del docente, facilita la selección y ubicación de contenidos y familiariza al estudiante con las herramientas y el manejo de la información en los nuevos medios.

1.2.2 Ventajas de los portales educativos

Las ventajas que proporcionan los portales educativos a sus destinatarios (profesores, estudiantes, entre otros) se derivan de los servicios que ofrecen. Entre las que se destacan:

- Proporcionar información de todo tipo a profesores, estudiantes y padres, así como instrumentos para realizar búsquedas de información.
- Proporcionar recursos didácticos de todo tipo, gratuitos y utilizables directamente desde el lugar donde se encuentre publicado el portal (materiales didácticos en línea).
- Contribuir a la formación del profesorado, mediante informaciones diversas y cursos de actualización de conocimientos.
- Asesorar a los profesores. En algunos casos los portales tienen espacios de asesoramiento sobre diversos temas: didáctica, informática, leyes, etc.
- Abrir canales de comunicación (foros, chat) entre profesores, estudiantes, instituciones y empresas de todo el mundo. A través de ellos se comparten ideas y materiales, se debaten temas, se consultan dudas.
- Proporcionar instrumentos para la comunicación: correo electrónico, chat, espacios para alojar páginas Web.

- Los estudiantes pueden obtener asistencia personalizada, ya que a diferencia de la educación tradicional, en un portal pueden o tienen herramientas para la comunicación asincrónica o sincrónica por los profesores. (5)

1.3 Plataformas e-learning

Las plataformas e-learning, son un espacio virtual de aprendizaje orientada a facilitar la experiencia de capacitación a distancia, tanto para empresas como para instituciones educativas.

Este sistema permite la creación de "aulas virtuales"; en ellas se produce la interacción entre tutores y alumnos, y entre los mismos alumnos; como también la realización de evaluaciones, el intercambio de archivos, la participación en foros, chats, y una amplia gama de herramientas adicionales. (6)

1.3.1 Sistemas de gestión del aprendizaje

Los sistemas de gestión del aprendizaje (LMS) por sus siglas en inglés, es el elemento central de la comunicación de un entorno e-learning. En su mayoría son un sistema basado en la web que permite el acceso a contenidos, la gestión de los recursos y la comunicación entre todos los actores implicados en el proceso (alumnos, profesores, administradores, entre otros).

Los LMS facilitan el aprendizaje distribuido y colaborativo a partir de actividades y contenidos pre elaborados, de forma síncrona o asíncronos, utilizando los servicios de comunicación de Internet como el correo, los foros, las videoconferencias y el chat. (7)

Manejan todos los procesos que rodean el aprendizaje, pero resultan insuficientes a la hora de la creación, reutilización, administración o mantenimiento de los contenidos educativos, haciéndose necesaria la integración con un sistema especializado para la gestión de contenidos educativos o LCMS. Aunque los LMS pueden almacenar estos contenidos, la tendencia es a que exista repositorio encargado de esta función, para ser accedidos y reutilizados por la plataforma LMS.

1.3.2 Sistemas de gestión de contenidos educativos

Los sistemas de gestión de contenidos educativos o LCMS tienen su origen en los sistemas de gestión de contenidos (CMS) y cuyo objetivo es simplificar la creación y la administración de los contenidos en línea.

Los LCMS son sistemas multiusuario que están enfocados a aprobar, publicar, administrar y almacenar recursos educativos, a diferentes niveles, permitiendo de esta manera reestructurar la información y los objetivos de los contenidos de manera dinámica, para crear y modificar los objetos de aprendizaje que atiendan a necesidades y estilos de aprendizaje específicos.

1.3.3 Repositorio de objetos de aprendizaje

Los repositorios de objetos de aprendizaje (ROA) son sistemas complejos, especializados en el almacenamiento de objetos de aprendizaje (OA). Accesibles y operables por usuarios, con funciones encaminadas a la clasificación, localización, reutilización, recuperación y mantenimiento de los OA, que permiten compartir estos con otros sistemas de los entornos e- learning. (8)

Para acceder a los recursos de un ROA, varias instituciones se han encargado de estandarizar mecanismos de acceso que hacen posible a sistemas, herramientas y usuarios reutilizar un recurso sin necesidad de emplear tiempo en crearlo.

1.4 Interoperabilidad

La interoperabilidad es un término definido por innumerables autores, organizaciones e instituciones en todo el mundo. Ha sido adoptado en el argot técnico de la Informática y las Telecomunicaciones en la comunidad hispana internacional, independientemente que la palabra no aparece definida hasta la fecha en el Diccionario de la Real Academia Española en su vigésima segunda edición.

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE 11) en el Glosario Estándar de Terminología de Ingeniería de Software (IEEE, 1990) define interoperabilidad como:

“(...) la habilidad de dos o más sistemas o componentes de intercambiar información y usar la información intercambiada (...)”

Al mismo tiempo que establece una relación semántica entre interoperabilidad y compatibilidad.

En tanto el Comité Técnico TC211 de la Organización Internacional de Estandarización (ISO 12) en la norma ISO/DIS 19101 (ISO, 2001) brinda la siguiente definición de interoperabilidad:

“(...) la capacidad de un sistema o componente de un sistema para proporcionar intercambio de información y el procesamiento cooperativo entre aplicaciones (...)”.

Además agrega que esta se refiere, entre otras cuestiones, a la búsqueda de información y herramientas de procesamiento en el momento que sea necesario e independientemente de la ubicación física de estas.

A partir del análisis de las definiciones en la bibliografía consultada y teniendo en cuenta el contexto en que se desarrolla la investigación, se define la interoperabilidad *como la capacidad de los sistemas que soportan los procesos de negocio, de colaborar en el cumplimiento de objetivos comunes a partir del intercambio de datos, servicios y conocimientos empleando mecanismos y estándares tecnológicos que garanticen conectar los sistemas y estructurar los datos en un formato común.*

1.5 Servicio de envío de correo electrónico

El correo electrónico, también conocido como *e-mail*, permite a los usuarios enviar y recibir mensajes de forma rápida mediante sistemas de comunicación electrónicos. Este medio permite enviar no solo textos, sino todo tipo de información digital. Este servicio está basado en protocolos de redes que posibilitan el envío y recepción de los mensajes. A continuación se reflejan algunas de las principales características que poseen estos protocolos.

1.5.1 Simple Mail Transfer Protocol (SMTP)

Los servidores SMTP (Protocolo simple de transferencia de correo) administran el envío de los mensajes de correo electrónico a Internet. El servidor SMTP administra el correo electrónico saliente y se utiliza en combinación con un servidor POP3 o IMAP de correo electrónico entrante. (9)

Características:

- Confiable
- Eficiente
- El servidor SMTP de origen se comunica directamente con el servidor SMTP de destino a través del Protocolo de control de transporte/Protocolo de Internet (TCP/IP).
- Dispone de mecanismos para transferir mensajes a través de uno o varios servidores SMTP.

1.5.2 Post Office Protocol (POP)

Los servidores POP3 (Protocolo de oficina de correos 3) retienen los mensajes de correo electrónico entrantes hasta que el usuario compruebe su correo y entonces los transfieren al equipo. POP3 es el tipo de cuenta más habitual para el correo electrónico personal. Normalmente, los mensajes se eliminan del servidor después de comprobar el correo. (10)

Características.

- Permite a los usuarios con conexiones intermitentes o muy lentas, descargar su correo electrónico mientras tiene conexión y revisarlo posteriormente incluso estando desconectados.
- Utiliza un mecanismo de firmado sin cifrado.
- Protección contra los accesos ilegales al buzón de correo de los usuarios.

1.5.3 Internet Message Access Protocol (IMAP)

Los servidores IMAP (Protocolo de acceso a mensajes de Internet) permiten trabajar con los mensajes de correo electrónico sin necesidad de descargarlos antes al equipo. Puede obtener una vista previa, eliminar y organizar los mensajes directamente en el servidor de correo electrónico, donde se guardan copias de los mismos hasta que el usuario los elimina. IMAP es más frecuente en las cuentas de correo electrónico de empresas. (10)

Características.

- Permite ver únicamente los encabezados del mensaje antes de decidir si abrirlo o eliminarlo.
- El servidor retiene el correo hasta que se solicite su eliminación.

- Puede consultarse el mismo correo desde diferentes computadoras ya que solo se lee lo que hay en el servidor.
- Permite operaciones avanzadas como creación de carpetas y buzones en el servidor.

1.6 Bibliotecas para el manejo de correos

Para el envío de los correos electrónicos además de los protocolos se utilizan bibliotecas que facilitan el tratamiento de este tipo de notificaciones en las diferentes aplicaciones Web. Entre las más utilizadas podemos encontrar Phpmailer¹, Java Mail² y Swift Mailer³.

1.6.1 Phpmailer

Es una manera eficaz de enviar mensajes de correo electrónico dentro de PHP. Es una clase PHP que proporciona un conjunto de funciones que facilita el envío de correo, añadiendo facilidad en el envío de correos con adjuntos, en formato HTML y con diferentes codificaciones, soporte para imágenes, encabezados personalizados y además funciona con múltiples servidores de correo. El mismo tiene un conjunto de métodos que ayudan al envío de correos. Es utilizado por muchos frameworks populares de desarrollo de PHP y es uno de los más populares de generación del correo electrónico y el envío de las bibliotecas en el mundo. (11)

1.6.2 Swift Mailer

Es una potente librería que permite el envío de *e-mails* y entre las características que ofrece está la de contar con una cabecera de control RFC 2822, internacionalización, balance de carga, soporte para SSL y TLS (para servidores GMail), imágenes embebidas, MIME 1.0 (adjuntos) . (12)

¹ <http://phpmailer.worxware.com>

² <http://javamail.kenai.com>

³ <http://swiftmailer.org/>

Esta es la librería que se utiliza para la implementación del sistema propuesto debido a todas las características mencionadas anteriormente y por estar integrada en el framework de desarrollo Symfony, el cual se emplea para el desarrollo del mismo.

1.7 Sistemas de notificaciones

Según la Real Academia Española, “*las **notificaciones** son acciones o efectos de notificar que constituye comunicar formalmente una resolución o dar una noticia con propósito cierto (13)*”.

Los sistemas de notificaciones son soluciones que permiten el envío de mensajes o notificaciones a partir de suscripciones realizadas sobre algún evento específico.

La red social **Facebook** contiene el módulo *Notifications* que permite tener las notificaciones del mismo directamente al escritorio del usuario. Actualmente en caso de que no se tenga abierta la cuenta de facebook, ni el correo para que notifique que se tiene una nueva notificación, no constituye un problema, gracias a este sistema.

Este componente indica cuando se tienen nuevos mensajes, eventos e invitaciones. Además permite actualizar el estado desde la aplicación. Con este programa ya no se tendrá problema para enterarse al momento de las notificaciones que tengan los perfiles en Facebook. (14)

Facebook utiliza las notificaciones del escritorio para avisar de cualquier cambio o actividad en tu cuenta.

Actualmente permite notificar los siguientes eventos:

- Algún amigo del usuario cambia su estado o su perfil.
- Envían al usuario una petición de amistad, evento o grupo.
- Alguien escribe en el muro de alguno de los amigos del usuario.
- Una foto es etiquetada con el nombre de uno de los amigos del usuario.

Entorno Virtual de Aprendizaje

El Entorno Virtual de Aprendizaje (EVA) creado con el LMS Moodle y utilizado en la Universidad de Ciencias Informáticas (UCI) es una plataforma que permite a estudiantes desarrollar actividades

académicas y facilitar su estudio cotidiano. Existe una tendencia mundial a usar Moodle ⁴ para construir Entornos Virtuales de Aprendizaje. Este permite a los profesores crear actividades para cada uno de los cursos, estas en el caso de ser interactivas poseen mecanismos de notificaciones para actualizar a los profesores y estudiantes del avance de las mismas sin necesidad de que estén conectados al sitio.

RHODA

El repositorio de objetos de aprendizaje RHODA, desarrollando en el centro FORTES, tiene como objetivo primordial apoyar los procesos de enseñanza y aprendizaje, tanto en la universidad como fuera de la misma. Para la mejora de estos procesos es necesario mantener a los usuarios actualizados sin necesidad de visitar la plataforma, debido a que las informaciones relacionadas con los objetos de aprendizaje son actualizadas constantemente, para lograr esto se hizo uso de las notificaciones tanto por correo electrónico como la sindicación de contenidos mediante RSS. Algunos ejemplos de envío de notificaciones es cuando se informa sobre comentarios a objetos de aprendizaje; se publica y se deniega comentario; se informa cuando sea aceptado un objeto de aprendizaje, entre otros.

1.8 Canales de redifusión

La redifusión de contenidos es una forma que tienen los sitios web de distribuir contenidos a los que los usuarios acceden frecuentemente. El contenido se distribuye a través de canales y los usuarios pueden leer esos canales con un software denominado programa Agregador (también llamado Lector de Canales o Lector de Noticias). (15)

1.8.1 RSS

RSS es el acrónimo de “Really Simple Syndication” y de “Rich Site Summary”. RSS es un formato basado en XML para la distribución de contenidos. En los RSS los autores pueden publicar titulares con pequeñas descripciones de la información específica que se desea distribuir. Aunque actualmente la mayoría de los canales RSS contienen titulares de noticias o información de última hora los usos de los RSS son más amplios. (16)

⁴ <http://moodle.org>

1.8.2 ATOM

El formato ATOM fue desarrollado como una alternativa a RSS por la incompatibilidad entre algunas versiones del protocolo RSS.

Gracias a su mejora frente a RSS puede integrar más información y gracias al popular sistema de publicación de *blogs* de Google (Blogger) se ha extendido rápidamente. Es también un sublenguaje de XML, pero no se trata de ninguna versión de RSS. Sin embargo, es un formato similar creado con el mismo objetivo que es el syndicar contenidos. (17)

1.9 Sistemas de recomendación

En un comienzo los sistemas de recomendación eran conocidos tan sólo como filtros colaborativos y los primeros trabajos datan de principios de los años 90. El término fue acuñado en 1992 para un sistema de filtrado de correo electrónico no automatizado. En 1994 se desarrolló el primer “*workshop*” en Berkeley donde se vio la utilidad en diversas áreas de los primeros algoritmos simples de este tipo. También se identificaron algunas cuestiones importantes para el desarrollo de estos algoritmos: escalabilidad, viabilidad económica, puntuaciones implícitas y explícitas. Uno de los grupos de investigación pioneros en el desarrollo del filtrado colaborativo fue el proyecto GroupLens de la universidad de Minnesota que aún permanece muy activo y que ha proporcionado una gran parte de la base algorítmica de muchos sistemas de recomendación. Fueron los primeros en introducir el filtro colaborativo automático usando un algoritmo de búsqueda de vecinos para proporcionar predicciones en los grupos de noticias de USENET. De este grupo de investigación partió también la iniciativa empresarial NetPerceptions, despejando gran parte de las dudas acerca de la viabilidad económica de estos proyectos. En la actualidad es un campo que se encuentra muy activo y genera un gran número de publicaciones y congresos todos los años. Y es que el filtrado colaborativo es un aspecto de gran importancia dentro de las redes sociales y la pequeña revolución que ha supuesto la llamada “Web 2.0”. Algunos ejemplos actuales de uso son (18):

- Recomendaciones en tiendas on-line. Partiendo de un producto se recomiendan otros productos que han interesado a los usuarios que compraron dicho producto. La web pionera en este tipo de recomendaciones fue Amazon.com.
- Filtrado de noticias. Se construye un perfil que almacena las noticias que un usuario consulta.

- Búsqueda de personas afines en comunidades. En webs como meneame.net se tienen en cuenta las noticias que cada usuario ha votado para generar una lista de vecinos con similares intereses.

Los sistemas de recomendación (SR) tienen como finalidad, sugerir nuevos ítems o elementos a un usuario, utilizando nuevas técnicas de descubrimiento de conocimiento. Los SR se basan en el historial de selecciones anteriores o en selecciones de otros usuarios con similar historial de ratings o valoraciones sobre los mismos ítems o similares. (19)

El termino "ítem" se utiliza para denominar el objeto que el sistema debe recomendar, por ejemplo, canciones, películas o noticias. Por lo general un sistema de recomendaciones se especializa en un ítem específico, por lo cual elementos como el diseño, la interfaz gráfica y el algoritmo usado para generar las recomendaciones, son personalizados para obtener sugerencias efectivas para dicho ítem.

1.9.1 Clasificación de los sistemas de recomendación

1.9.1.1 Recomendaciones basadas en contenido

El elemento clave de este tipo de recomendador es determinar cuán relacionado está un objeto con un determinado usuario, para lo cual utiliza la medida de similitud. De esta manera puede encontrar objetos que se adapten lo mejor posible a las preferencias de los usuarios. Para calcular esta relación el sistema se basa exclusivamente en la descripción del elemento y en la información recogida en el perfil del usuario que almacena los intereses de este.

Dicho perfil puede ser rellenado de manera explícita, preguntando directamente al usuario por sus preferencias o su criterio acerca de un elemento determinado. La información puede ser recogida además de manera indirecta, considerando elementos que revelen las preferencias del usuario, entre estos elementos se pueden encontrar: sitios visitados, tipos de archivos descargados, tiempo de permanencia en un sitio, categoría de noticias leídas y cualquier otra información que resulte útil.

Este tipo de sistemas puede ser ampliamente utilizado. Su uso puede extenderse a sitios noticiosos, librerías, descargas de música y video, restaurantes y ventas en general. Los elementos comunes para todos estos sistemas resultan: la necesidad de contar con un perfil de usuario que almacene sus

preferencias, una descripción del elemento que posiblemente sea recomendado y un sistema que determine lo cerca que está dicho elemento del usuario, con el fin de poder recomendárselo.

Una ventaja importante de este tipo de sistema recomendador es que se pueden realizar recomendaciones sin la necesidad de contar con un historial previo, por lo cual se pueden efectuar predicciones independientemente del historial del usuario. Como principal desventaja se encuentra el hecho de que las recomendaciones en su mayoría tienen un alto grado de similitud, ya que se basan en los mismos datos.

1.9.1.2 Sistemas basados en filtrado colaborativo

El filtrado colaborativo tiene como base el agrupamiento de personas que muestren intereses similares. Para realizar estas asociaciones el usuario debe expresar sus preferencias a través de la evaluación de elementos que se le presentan, sirviendo esto para crear un perfil aproximado de este (20). Una vez que se recopilan los datos sobre las preferencias del usuario y se conforma su perfil, el sistema busca entre los restantes usuarios los que posean características similares, formando así grupos de personas con intereses comunes. Estos grupos de usuarios pueden denominarse "usuarios más cercanos". De esta manera, para crear una recomendación efectiva, el sistema puede obtener la lista de los elementos mejor evaluados por el grupo de usuarios y que aún no han sido evaluados por el nuevo individuo. (20)

Una de las condiciones necesarias para que este tipo de recomendador funcione de manera efectiva es la necesidad de que la cantidad de personas que utilizan el sistema sea suficiente para que cualquier nuevo individuo pueda encontrar grupos de usuarios con características similares. Otros elementos importantes que debe poseer el sistema son: contar con una forma eficaz de representar los intereses de los usuarios, de modo que la comparación entre estos no resulte tan compleja y disponer de algoritmos que sean capaces de asociar de manera correcta a las personas dentro del sistema.

En este tipo de sistema coinciden en una misma persona los roles de constructor y consumidor de recomendaciones. Esto se debe a que la representación de un perfil será más exacta en tanto más recomendaciones o evaluaciones realice dicho usuario, por lo tanto se incrementa también la exactitud de las recomendaciones que el sistema puede brindar al usuario.

Entre los Sistemas de Filtrado Colaborativo se encuentran los Basados en Usuarios que cuentan dentro de su flujo básico con tres pasos que suelen ser similares independientemente de su programación:

1. Determinar las preferencias del usuario.
2. Comparar estas preferencias con las de otros usuarios para determinar aquellos con los que tiene gustos similares, a este paso se les conoce como “Determinar Vecindario”, siendo estas personas con gustos comunes los posibles vecinos.
3. Recomendar los elementos que fueron del gusto de los usuarios similares y que no han sido visitados por el usuario en cuestión.

Por tanto una característica que suelen tener los sistemas de recomendación colaborativos, es el uso de pesos para definir qué tan bueno es un elemento dado, siendo esta una deficiencia para el desarrollo de este módulo, al no poderse determinar pesos para las distintas suscripciones hechas por los usuarios, por lo que fue necesario crear nuevas fórmulas para la aplicación del algoritmo basado en los Sistemas de Filtrado Colaborativo.

1.9.1.3 Sistemas de recomendación híbridos

Los enfoques básicos de la recomendación, anteriormente expuestos, han sido utilizados de manera exitosa en disímiles campos y escenarios. Pese a su difusión, de forma general cada uno de ellos presenta varias desventajas relacionadas principalmente con la necesidad de un gran número de usuarios adheridos al sistema para un correcto funcionamiento del mismo y con la tendencia a una sobre-especialización que va aumentando a medida que aumenta la utilización de la aplicación por parte del usuario.

Con vista a mitigar estas desventajas, una opción muy utilizada es la de combinar la recomendación basada en contenido con el filtrado colaborativo, conformando un sistema de recomendación con características híbridas. A través de estos, la construcción de preferencias se basa tanto en los intereses del usuario, como en los intereses de los usuarios similares, coincidiendo así con dos de las técnicas específicas antes expuestas. (20)

1.10 Métodos, lenguajes y herramientas de desarrollo

Para la construcción del módulo Notificaciones se emplea el ambiente de desarrollo utilizado en el proyecto Portal Educativo, software al que se integrará la solución propuesta. A continuación se mencionan las características más relevantes del ambiente de desarrollo empleado, enfatizando las que tienen mayor impacto en el desarrollo de la solución.

1.10.1 Metodología de desarrollo. RUP

Se define metodología como: “el conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.” (21)

“El proceso de desarrollo de software es aquel en que las necesidades del usuario son traducidas en requisitos de software, los mismos son transformados en diseño y el diseño implementado en código; el código es probado, documentado y certificado para su uso operativo. Concretamente ‘define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo’”. (22)

Debido a lineamientos de la directiva del centro de desarrollo en el cual se enmarca el proyecto donde se desarrolla el Portal educativo y a exigencias del propio cliente se utilizará la metodología del Proceso Unificado de Desarrollo (RUP).

El Proceso Unificado del Software, RUP (Rational Unified Process), es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software. (...) utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del Proceso Unificado. (23)

Los autores del Proceso Unificado de Desarrollo destacan que los verdaderos aspectos definatorios del Proceso Unificado se resumen en tres frases clave: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. Esto es lo que hace único al Proceso Unificado. (23)

Dirigido por casos de uso

Un caso de uso (CU) es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. (...) Dirigido por casos de

uso quiere decir que el proceso de desarrollo sigue un hilo avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan, y los casos de uso finales son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba. Aunque es cierto que los casos de uso guían el proceso, no se desarrollan aisladamente. Se desarrollan a la vez que la arquitectura del sistema. Es decir, los casos de uso guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso. (23)

Centrado en la arquitectura:

El papel de la arquitectura software es parecido al papel que juega la arquitectura en la construcción de edificios. (...) Análogamente, la arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción. El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema. (...) Sin embargo, también se ve influida por muchos otros factores, como la plataforma en la que tiene que funcionar el software (arquitectura hardware, sistema operativo, sistema de gestión de base de datos, protocolos para comunicaciones en red), (...) consideraciones de implantación, sistemas heredados, y requisitos no funcionales (por ejemplo, rendimiento, fiabilidad). Podemos pensar que la arquitectura de un sistema es la visión común en la que todos los empleados (desarrolladores y otros usuarios) deben estar de acuerdo, o como poco, deben aceptar. La arquitectura nos da una clara perspectiva del sistema completo, necesaria para controlar el desarrollo. (24)

Iterativo e incremental:

Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar controladas; esto es, deben seleccionarse y ejecutarse de una forma planificada. Es por esto por lo que son mini-proyectos. (23)

Estos conceptos los de desarrollo dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental son de igual importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. La

eliminación de una de las tres ideas reduciría drásticamente el valor del Proceso Unificado. Es como un taburete de tres patas. Sin una de ellas, el taburete se cae. (23)

1.10.2 Lenguaje de modelado

UML (Unified Modeling Language, traducido al idioma español como Lenguaje Unificado de Modelado) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. (24)

UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (24)

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. (24)

En el desarrollo del módulo se utilizará la versión 2 de UML.

1.10.3 Entorno integrado de desarrollo. Netbeans

Un IDE (Integrated Development Environment, traducido al español Ambiente Integrado de Desarrollo) es un entorno de programación, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica que además pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. (25)

Netbeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, aunque permite programar en distintos lenguajes. Ofrece un excelente

entorno para programar en PHP. Tiene un excelente balance entre una interfaz con múltiples opciones y un aceptable completamiento de código. (26)

NetBeans en su versión 7.x se caracteriza por: (26)

- Creación de proyectos PHP: NetBeans provee a los desarrolladores de una estructura para los proyectos que se puedan crear junto a este IDE. Propone un esqueleto para organizar el código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS. Además NetBeans posee un sistema para examinar todos los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación.
- Integración con Drupal: Permite realizar aplicaciones utilizando frameworks, haciendo el trabajo más ágil.
- Editor de código fuente: Mejora en su editor, sobre todo en el editor de PHP, es mucho más rápido y a la vez robusto, contiene más ayuda en línea, reconocimiento de sintaxis y todo lo que provee la última versión de PHP, la 5.3.
- Depuración de PHP: NetBeans integra muy bien la utilización Xdebug, permitiendo inspeccionar y examinar cada variable local, establecer puntos de interrupción y evaluar el código en nuestra lógica.
- Integración con sistemas de control de versiones: Esta es una de las condiciones necesarias para los proyectos y es la posibilidad de contar con la integración de sistemas de control de versiones, tales como SVN, CVS, Mercurial y Git. Desde el editor es posible realizar la administración de estos sistemas versionados, sus commit, branch, importar, exportar, revert, clonar, entre otras funciones.

1.10.4 Herramientas case. Visual Paradigm For UML 8.0

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del

código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. (27)

Visual Paradigm para el Lenguaje Unificado de Modelado (VP-UML) está diseñado para una amplia gama de usuarios, tales como: ingenieros de software, analistas de sistemas, analistas de negocio y arquitectos de sistemas, o para cualquiera que esté interesado en la construcción de forma fiable y a gran escala de sistemas que utilizan un enfoque orientado a objetos. (28)

Visual Paradigm es una herramienta CASE profesional que de igual forma, soporta el ciclo de vida completo del desarrollo de software (...). Ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Utiliza UML como lenguaje de modelado y se integra con herramientas Java tales como: Netbeans, Eclipse, JBuilder, (...) entre otras. Es una herramienta que genera la documentación del proyecto automáticamente en varios formatos como Web o .pdf, soporta la realización de ingeniería tanto directa como inversa y permite control de versiones. Presenta una interfaz de uso intuitiva y con muchas facilidades a la hora de modelar los diagramas (...). (29)

1.10.5 Lenguajes de programación

En una arquitectura cliente-servidor los lenguajes de programación pueden dividirse en dos grupos, los lenguajes del lado del Servidor y los lenguajes del lado del Cliente. En el lado del cliente existe tecnología como JavaScript utilizado para ofrecer dinamismo a la aplicación y como técnica de desarrollo para la web. Del otro lado la principal alternativa es el procesador de hipertexto (PHP).

1.10.5.1 JavaScript (JSCRIPT)

JavaScript es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML. La aparición del JavaScript produjo una importante revolución, ya que dio al usuario la posibilidad de crear aplicaciones en línea o sea modificar páginas web en tiempo real. (30)

1.10.5.2 Procesador de hipertexto (PHP)

Como lenguaje de desarrollo se propone PHP ya que este lenguaje es multiplataforma, está orientado al desarrollo de aplicaciones web manteniendo acceso a información almacenada en una base de datos, el código fuente escrito en PHP es transparente al navegador y al usuario haciendo la programación segura y confiable. Presenta capacidad de conexión con la mayoría de los motores de base de datos destacándose principalmente en MySQL y PostgreSQL. (31)

Posee una amplia documentación en su página oficial, donde todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

1.10.5.3 XML

Lenguaje de Etiquetado Extensible XML es uno de los lenguajes más utilizados para el intercambio de datos sobre la web. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información. (32)

XML se compone por tres partes fundamentales como son:

- El esquema (Doctype y XML Schema)
- Consulta (XPath y XQuery)
- Transformación (XSLT)

1.10.6 Sistemas de gestión de contenidos. Drupal

Los Sistemas de Gestión de Contenidos (CMS) logran resolver infinidad de cuestiones relativas a la administración y publicación de información online. Excepto en casos muy particulares donde se requiere

un programa web a medida (ya sea para gestión específica como para cubrir funcionalidades y servicios que surgen en demanda), los CMS son la solución ideal para cualquier tipo de sitios web.

Debido a lineamientos de la directiva del centro de desarrollo en el cual se enmarca el proyecto donde se desarrolla el Portal Educativo y a exigencias del propio cliente se utilizará como CMS Drupal en su versión 7.x.

Drupal es una solución de portal dinámico profesional concebida para el desarrollo de sitios Web corporativos e intranets. Permite lograr una sencilla administración de contenidos online. (33)

Características

Es un paquete de software que permite publicar, manejar y organizar fácilmente una gran variedad de contenido. Drupal es utilizado en todo el mundo para distintos tipos de sitios a saber:

- Portales Web de comunidades
- Sitios de discusión
- Sitios Web corporativos
- Intranets
- Blogs y sitios personales
- Sitios de e-commerce
- Directorios online

Drupal posee una gran cantidad de extensiones que le permiten cubrir funcionalidades como por ejemplo:

- Sistema de manejo de contenido.
- Blogs
- Ambientes colaborativos de generación de contenidos

- Foros
- Boletines web
- Galerías de fotos
- Carga/descarga de ficheros

Es una solución Open-Source distribuida bajo la licencia GPL ("General Public License") y es mantenido y desarrollado por una gran cantidad de usuarios y desarrolladores en todo el mundo.

Por su diseño modular, puede ser fácilmente ampliado y adaptado a las necesidades de diversas empresas y proyectos de cualquier envergadura, ya que este producto ha sido concebido desde el comienzo para proveer una escalabilidad inmejorable comparado con otras soluciones similares disponibles en el mercado.

Su terminología difiere de la de otros gestores de contenido. La siguiente sección explora las partes más importantes de un sistema Drupal y explica los términos más relevantes.

Un nodo es el bloque básico con el que construimos una página Web con Drupal: este puede ser un artículo, una imagen o una entrada de un foro. Los siguientes tipos de nodo se habilitan por defecto:

- Page: para páginas estáticas
- Story: para artículos

Se pueden añadir más tipos de nodos instalando y habilitando módulos. Por ejemplo, el módulo fórum define un nuevo tipo de contenido llamado fórum topic. El módulo blog ofrece el nodo personal blog entry.

Un bloque es un cuadro con contenido que aparece en alguna región de una página Web. Drupal nos ofrece varios bloques pre configurados, como UserLogin, Who's online o RecentComments.

Adicionalmente, cualquier nuevo módulo que se instale puede proporcionar bloques, nuevos tipos de contenido, entre otros elementos, que habilitará en su caso el administrador.

1.10.7 Desarrollo de módulos en Drupal

Uno de los puntos fuertes de Drupal es la facilidad con la que puede extenderse o modificarse el funcionamiento del mismo a través de cómo se puede apreciar en la figura 2.

Un módulo es una colección de funciones que se enlazan dentro de Drupal, ofreciendo funcionalidades adicionales a la instalación Drupal. Drupal contiene un conjunto de módulos llamados “Módulos del núcleo” que vienen en la distribución de la instalación del CMS. Existe la posibilidad de descargar módulos desarrollados por terceros lo que se conoce como “Módulos contribuidos” y por último las funcionalidades que no sean cubiertas por estos dos tipos de módulos se desarrollan en “módulos específicos”.

El área de descarga del portal de Drupal en su portal en Internet tiene una enorme selección de módulos contribuidos que cubren una amplia gama de nuevas características, todos ellos creados por la comunidad de desarrolladores y guardados en repositorios GIT (Drupal.org), donde continúan desarrollándose. Por supuesto, dichos módulos están bajo licencia GPL.

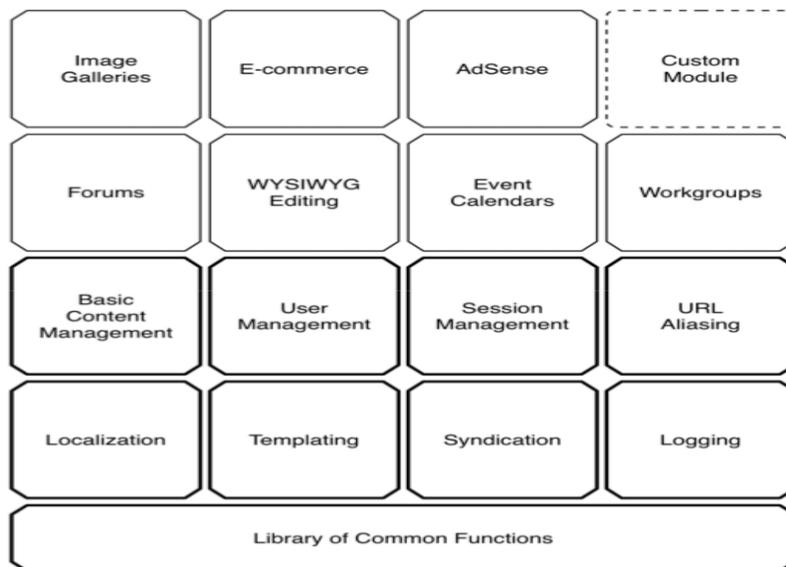


Figura 1 Módulos en Drupal tomado del libro Pro Drupal 7 Development

1.10.8 Sistema gestor de base de datos. Postgresql

Pro Drupal 7 Development

Un SGBD (Sistema Gestor de Bases de Datos) o DBMS (Data Base Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (34)

Debido a lineamientos de la directiva del centro de desarrollo en el cual se enmarca el proyecto donde se desarrolla el Portal educativo y a exigencias del propio cliente se utilizará como sistema gestor de base de datos PostgreSQL.

PostgreSQL es un sistema de gestión de base de datos relacional compatible con ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad); orientado a objetos. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación en cuanto a fiabilidad, corrección e integridad de datos. Se ejecuta en los principales sistemas operativos, incluyendo Linux, Unix y Windows. Tiene soporte completo para llaves foráneas, uniones, vistas, disparadores y procedimientos almacenados. Incluye la mayor parte de los tipos de datos: integer, boolean, char, varchar, date, interval y timestamp. También es compatible con el almacenamiento de objetos binarios grandes como imágenes, sonidos o vídeo. Tiene interfaces nativas de programación en C/C ++, Java, .Net, Perl, Python, Ruby, entre otros y una documentación amplia.

Ventajas de PostgreSQL: (35)

- Gran escalabilidad, se ajusta al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta y algunos casos llegando a soportar el triple de carga que soporta MySQL.
- Soporta transacciones y llaves foráneas con comprobaciones de integridad referencial.

- Tiene mejor soporte para triggers y almacenamiento de procedimientos en la propia base de datos, comparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.
- Implementa subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en los que MySQL no podría.

1.10.9 Servidor web. Apache

Un servidor web es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente, generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. (36)

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache es usado primariamente para enviar páginas web estáticas y dinámicas en la World Wide Web. La licencia de software bajo la cual el software de la fundación Apache es distribuido es una parte distintiva de la historia de Apache HTTP Server y de la comunidad de código abierto. La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original. (37)

Ventajas de Apache: (37)

- **Modular:** La arquitectura del servidor Apache consta de diversos módulos que aportan muchas de las funcionalidades que podrían considerarse básicas para un servidor web.
- **Código abierto:** La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original.
- **Multiplataforma:** Es usado en varias plataformas, tanto, libres como propietarias.
- **Extensible:** Se pueden añadir módulos y escribir sus propios códigos adaptándolos a las funciones del servidor Apache, de forma tal, que se ajusten a las necesidades de los desarrolladores.

- Popular: Es fácil conseguir ayuda y soporte. Apache tiene amplia aceptación en la red desde 1996 y es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en el 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo.

1.11 Conclusiones del capítulo

El análisis de conceptos asociados al objeto de estudio y de aplicaciones similares a la que se desarrolla en la investigación, arrojó como resultado que es necesario facilitar la comunicación en el proyecto Portal Educativo. Luego de realizar un estudio del arte de los sistemas de recomendación, teniendo en cuenta la forma en que procesan la información se concluye que de las clasificaciones se utilizará un sistema basado en filtrado colaborativo. Estas mejoras deben estar sustentadas en el uso de tecnologías libres y para ello se hizo un estudio detallado de las principales tendencias en lenguajes, metodologías y demás herramientas para el desarrollo de aplicaciones web con soporte de datos. De ellos fueron seleccionados los que estaban en concordancia con la arquitectura previamente definida en el proyecto al que pertenece la solución.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

Para lograr una mejor comprensión del sistema es necesaria una buena descripción de los procesos de negocio del mismo, también es muy importante una correcta definición de requisitos funcionales y no funcionales, así como la descripción de los actores involucrados. También es necesaria la generación de diferentes artefactos como el modelo de dominio, el diagrama de casos de uso del sistema y una breve descripción de cada caso de uso identificado.

2.2 Descripción de los procesos de negocio

Para lograr una mayor comprensión del diagrama de modelo de dominio que se expondrá a continuación, es necesario realizar previamente la definición de los conceptos que involucra dicho modelo.

Usuario: Personas con permisos de acceso que se registran en los portales.

Notificaciones: Confirmaciones recibidas por los usuarios pertenecientes al portal, de las acciones desarrolladas en relación a ellos.

Portal: Sitios existentes encargados de brindar información a la comunidad universitaria, tanto docentes, productivas, culturales, deportivas e investigativas.

Mensajes: Representa todos los mensajes que son creados para enviar a los usuarios como los enviados por los usuarios.

Suscripción: Objeto que representa a los suscriptores con sus servicios.

Recomendaciones: Sugerencias de contenidos que recibe el usuario de acuerdo a su perfil.

2.3 Modelo de dominio

El modelo de dominio ayuda a comprender los conceptos relacionados a los procesos que se desean desarrollar. Una vez analizado el problema e identificados los principales conceptos presentes en el mismo, se hace posible la elaboración del modelo de dominio, donde se presenta en un marco conceptual, la relación existente entre las definiciones.

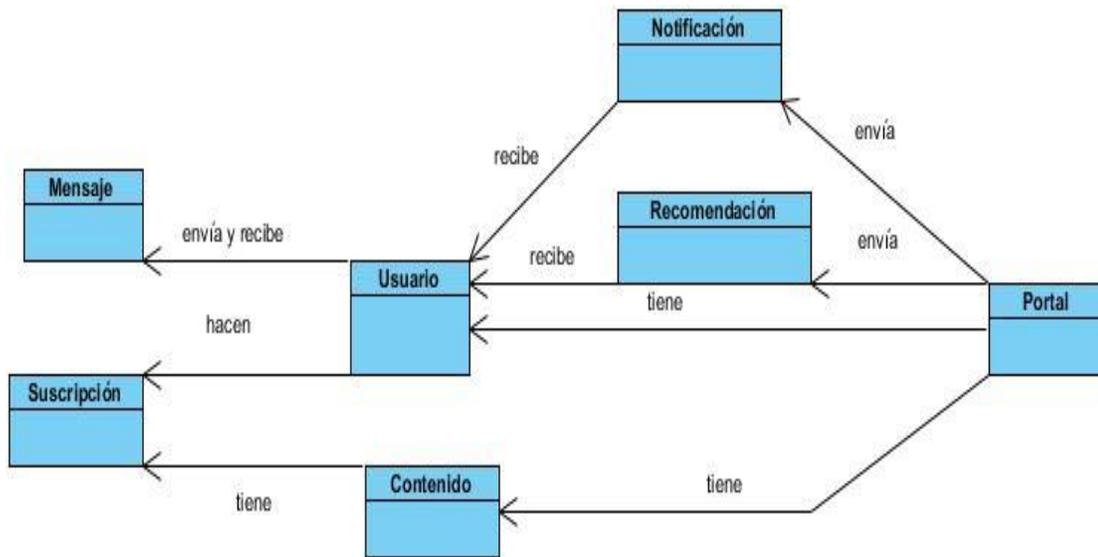


Figura 2 Modelo de dominio

2.4 Propuesta del sistema

La propuesta del sistema de notificaciones dentro de la comunidad universitaria tiene como objetivo agilizar el proceso de información que se lleva a cabo en la comunidad y llevar el mismo a todos los miembros de esta. Con este sistema se podrá enviar una notificación a todos los usuarios autenticados y podrán ver las notificaciones que le hayan sido enviadas.

El módulo Notificaciones está compuesto por submódulos, entre ellos se encuentra el que se encarga de la gestión de suscripciones, a las cuales responden las notificaciones, con el fin de mejorar el proceso de escoger a qué contenido suscribirse, se creó un recomendador de suscripciones.

El recomendador cuenta con dos posibles estados para la recomendación, el primero es cuando el usuario no se ha suscrito a ningún contenido y el segundo se llevará a cabo una vez que la situación expuesta en el caso uno cambie, desechándose el perfil para la determinación de la vecindad y tomándose para ello las suscripciones comunes entre el usuario autenticado y el resto, dicho sistema recomendará únicamente a aquellos usuarios que se hayan registrado.

Características del recomendador

El recomendador de suscripciones cuenta, entre sus principales características con la no existencia de pesos o votos por parte de los usuarios, esto se debe a que no se implementó una manera de determinar cuándo una suscripción es mejor que otra para un usuario, por lo cual, los algoritmos desarrollados hasta el momento para la recomendación, pierden validez para este nuevo sistema de recomendación de suscripciones, por lo que surge la necesidad de crear nuevas fórmulas matemáticas para resolver el problema.

Si el usuario autenticado se encuentra en el caso uno, es necesario determinar las posibles recomendaciones a partir de su perfil, el cual presenta una serie de atributos que posibilitarán llevar a cabo la recomendación. Sin embargo, este perfil no tiene carácter obligatorio, por lo que, si el usuario decide no llenarlo, entonces serán recomendados los ítems a los que más se han suscrito los usuarios.

Si el usuario autenticado se encuentra en el caso dos, pero no se puede determinar una vecindad debido a la no existencia de otros usuarios con gustos similares, o existe esa vecindad, pero no hay posibles ítems a recomendar (esto sucede si los vecinos no se han suscrito a contenido distinto al usuario autenticado), el sistema retomará para la recomendación, el caso uno.

Caso uno (Suponiendo que se ha llenado el perfil)

El perfil de usuarios cuenta con una serie de atributos a los que se les definió previamente un valor constante entre uno y cinco, el cual determinará la importancia del mismo para todos los usuarios. Una vez que el usuario ha llenado el perfil, ya sea parcial o totalmente, se determinará la vecindad mediante el listado de usuarios con perfiles semejantes, utilizando la siguiente fórmula:

$$P_v = \frac{\sum_{a=1}^n P_{v,a}}{n}$$

Donde:

- P_v = Peso del vecino (v).
- n = Cantidad de atributos comunes entre el vecino (v) y el usuario autenticado.
- $P_{v,a}$ = Peso del atributo a que pertenece el perfil del vecino (v) y que es común con el del usuario.

Luego de determinado dicho vecindario, se listan los nodos a los que se suscribieron los vecinos y no así el usuario autenticado, para determinar de ellos cuales son los mejores a recomendar a partir de la fórmula:

$$P_{i,sub} = \frac{\sum_{v=0}^n P_{v,i,sub}}{V_{SI}}$$

Donde:

- $P_{i,sub}$ = Peso de las suscripciones del ítem (i).
- V_{SI} = cantidad de vecinos suscritos al ítem (i).
- $P_{v,i,sub}$ = Peso del vecino (v), suscrito al ítem (i).
- n = cantidad de vecinos suscritos al ítem (i).

En el caso de las des-suscripciones el peso del ítem está dado por la siguiente formula:

$$P_{a,i,des} = \frac{\sum_{v=0}^n P_{v,i,des}}{V_{desI}}$$

Donde:

- $P_{i,des}$ = Peso de las des suscripciones del ítem (i).
- V_{desI} = cantidad de vecinos des suscritos al ítem (i).
- $P_{v,i,des}$ = Peso del vecino (v), des suscrito al ítem (i).
- n = cantidad de vecinos des suscritos al ítem (i).

Finalmente, el peso del ítem (i) va a estar dado por la siguiente fórmula:

$$P_i = P_{i,sub} - P_{i,des}$$

Donde:

- P_i = Peso del ítem (i).
- $P_{i_{sub}}$ = Peso de las suscripciones del ítem (i).
- $P_{i_{des}}$ = Peso de las des suscripciones del ítem (i).

Este algoritmo se basa en la forma de “aprender” de la Familia de Predictores “Slope – One”, específicamente el Bi-Polar, el cual plantea una forma de aprendizaje para el recomendador basada tanto en lo que los usuarios determinan bueno como lo que no.

Caso dos (Cuando se suscribe a los contenidos)

Una vez que el usuario ha comenzado a suscribirse a los distintos contenidos, se determinará una nueva vecindad, la cual partirá de los usuarios con gustos similares a los que este demuestre en su navegación, dicha vecindad se determinará con la fórmula:

$$P_v = Sub_{l,v} + Des_{l,v}$$

Donde:

- P_v = Peso del vecino (v).
- Des_l = Cantidad de usuarios des suscritos del usuario autenticado (l).
- Sub_l = Cantidad de usuarios suscritos del usuario autenticado (l).

Luego de creado el vecindario se procede a determinar las posibles suscripciones a ser recomendadas de la misma manera que en el caso uno. Teniendo en cuenta que si no se puede determinar dicha vecindad o que la misma no permite tener recomendaciones, se desecha y se continúa recomendando mediante los perfiles, hasta que se pueda crear una vecindad capaz de generar recomendaciones.

Para un mejor entendimiento del funcionamiento del sistema de recomendación se muestra la siguiente figura:

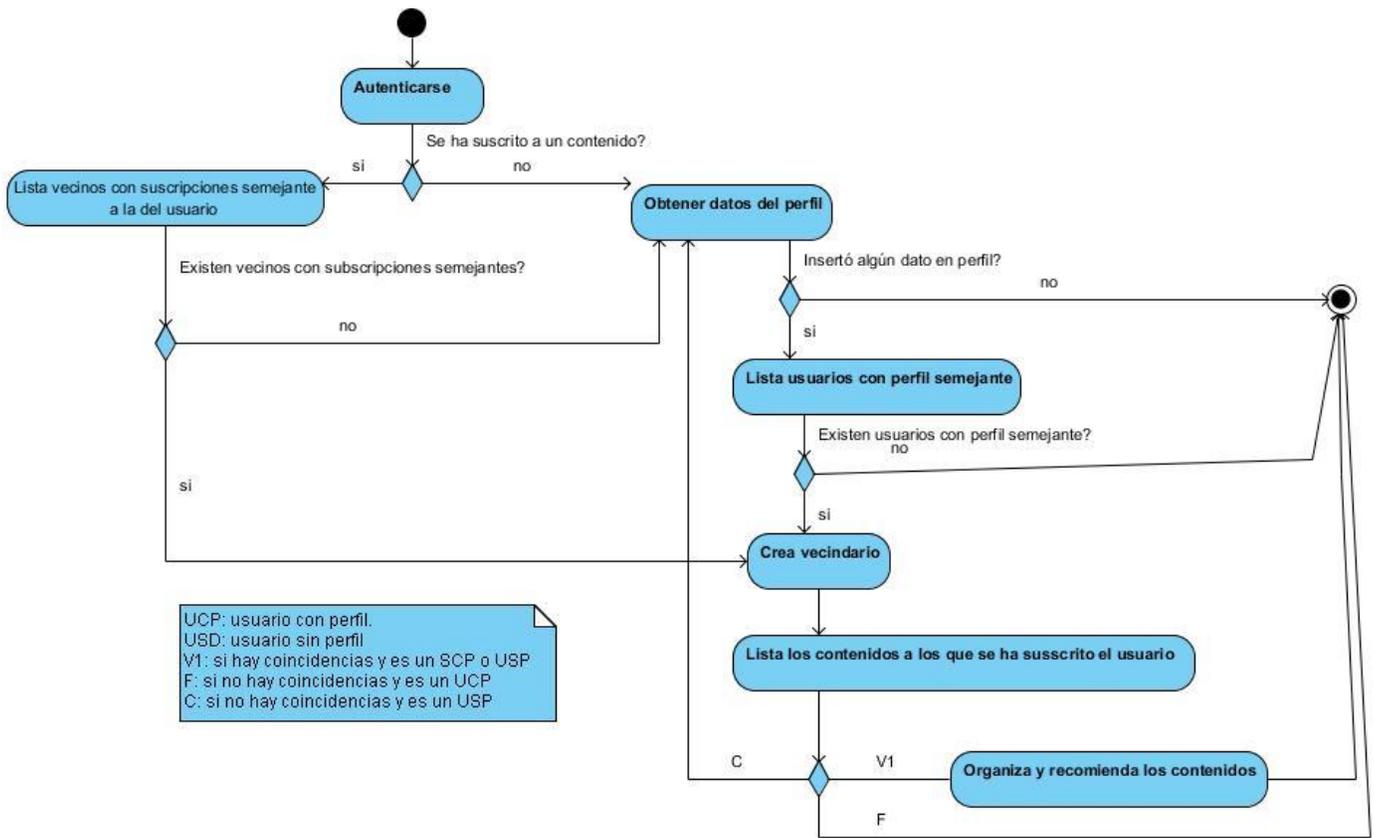


Figura 3 Diagrama de actividades del sistema de recomendación

2.5 Requisitos

Los requisitos tienen como principal objetivo guiar el desarrollo hacia un sistema correcto. Este propósito es conseguido por medio de los requisitos funcionales y los no funcionales que presente el software a desarrollarse. (38)

2.5.1 Requisitos funcionales

Los requisitos funcionales son las condiciones o capacidades que el sistema debe cumplir lo suficientemente bien para poder llegar a un acuerdo con los clientes. A continuación se muestran los requisitos funcionales del sistema de notificaciones.

RF 1: Gestionar mensajes.

RF 1.1: Visualizar mensaje.

RF 1.2 Crear mensaje.

RF: 1.3 Eliminar mensaje.

RF 2: Filtrar mensajes.

RF 3: Bloquear/desbloquear usuario en la mensajería.

RF 4: Configurar plantillas de mensaje.

RF 5: Recomendación de ítems de suscripción.

RF 6: Gestionar suscripciones a contenidos.

R6.1- Adicionar suscripción a contenidos.

R6.2- Modificar suscripción a contenidos.

R6.3- Eliminar suscripción a contenidos.

RF 7: Gestionar suscripciones a autor.

R7.1- Adicionar suscripción a autor.

R7.2- Modificar suscripción a autor.

R7.3- Eliminar suscripción a autor.

RF 8: Gestionar suscripciones a categoría.

R8.1- Adicionar suscripción a categoría.

R8.2- Modificar suscripción a categoría.

R8.3- Eliminar suscripción a categoría.

RF 9: Responder mensaje.

RF 10: Filtrar Lista de suscripciones.

RF 11: Procesar suscripciones.

RF12: Listar suscripciones.

2.5.2 Requisitos no funcionales

Los requisitos no funcionales son las cualidades o propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, soporte, entre otras.

2.5.2.1 Usabilidad

Comprensión del usuario final.

El sistema debe garantizar una ayuda legible y comprensible, capaz de enseñar y guiar el uso del software, sobre todo para estudiantes universitarios, profesores e investigadores, que en esencia constituyen el usuario final. El uso de prácticas de diseño de interfaces similares a las populares redes sociales o profesionales es una fortaleza en este sentido, ya que son comúnmente conocidos por el usuario final, dadas sus características. Por tanto se define que el usuario alcance a entender y hacer uso correcto del software en un periodo entre 15 y 30 días.

Accesibilidad.

La aplicación debe ser accesible, garantizando que personas con algunas discapacidades visuales y/o auditivas puedan hacer uso de la misma. Para lograr este propósito se tendrán en cuenta algunas de las Directrices de Accesibilidad para el Contenido Web 2.0 establecidas por Iniciativa para la Accesibilidad Web (WAI por sus siglas en inglés). A continuación se describen cada una de ellas:

- Los elementos que conforman una interfaz de usuario, llámese imágenes, botones, campos de entrada de datos, hipervínculos, deben tener un nombre sugerente, que logre que el usuario perciba la acción, además de mostrar una breve descripción de su función o razón de ser, en la propia interfaz o con solo pasarle el cursor por encima.

- En el caso de los formularios deben especificar las reglas de sus campos, como pueden ser: los obligatorios, aquellos que solo aceptan tipos de datos específicos o longitudes de cadenas limitadas, así como garantizar mediante interfaces amigables que no se introduzcan datos innecesariamente en campos que dependen de otros.
- Los formularios que sean enviados con datos erróneos deben especificar claramente cuál fue el error y mantener el resto de los datos.
- Asegurar que toda la información que utilice el color como elemento informativo pueda ser entendida por las personas o dispositivos que no pueden distinguir los colores, es decir, el color no puede ser la única información que brinde un contenido.
- La página debe poder leerse completamente cuando no se utilicen hojas de estilos.
- Ningún elemento debe parpadear en la pantalla, para que personas con neuropatías puedan hacer uso del sistema.
- Utilizar cabeceras de fila y de columna para alcanzar un mejor entendimiento.
- En el caso de contenidos de audio, garantizar una alternativa textual para que personas con discapacidad auditiva puedan hacer uso de la herramienta.
- No debe ser necesario el desplazamiento de izquierda a derecha para poder visualizar toda la página.
- El sistema debe permitir al usuario modificar el tamaño de la letra, así como el contraste de la página para que personas con alguna discapacidad visual puedan hacer un mejor uso del sistema.
- Dada la naturaleza de la aplicación, se debe garantizar que los contenidos tengan validez gramatical y correcta ortografía.
- Impedir la apertura repentina de nuevas ventanas o no abusar de esta función.
- El software debe contar con un mapa de sitio que modele con precisión la navegación del sistema.
- Especificar la expansión de cada abreviatura o acrónimo que aparezca en el contenido del sitio.
- El sistema debe contar con un estilo de presentación coherente para cada página.

2.5.2.2 Eficiencia

Para un software es muy importante la eficiencia, un sistema rápido y preciso puede ser la clave del éxito. Teniendo en cuenta la finalidad de este sistema, debe ser capaz de responder a las peticiones en un periodo entre 1 y 5 segundos, aunque podría variar dependiendo de otros factores ajenos, como conectividad del cliente o talla del contenido que se utilice, para ello se recomienda no subir recursos con más de 10 MB. Además el trabajo en comunidades, tiene como característica la participación simultánea de muchas personas, por tanto va a necesitar que el sistema sea capaz de soportar eficientemente una alta concurrencia de usuarios. Otro aspecto importante a tener en cuenta es la transferencia de datos y la capacidad de almacenamiento, ya que en el mismo se van a alojar numerosos contenidos y recursos digitales.

2.5.2.3 Restricciones de diseño

Lenguaje de programación.

El lenguaje de programación utilizado está condicionado por la herramienta Drupal v7, que obliga a hacer uso de PHP en su versión 5.2.x o superior. Otro lenguaje usado, aunque para ser interpretado por el cliente es Javascript.

Componentes y librerías.

El componente fundamental que motoriza la aplicación es Drupal 7, que además de imponer el lenguaje de programación, también propone una arquitectura y restricciones de diseño, que el desarrollador no puede ignorar (para más información visitar <http://drupal.org>). Deben utilizarse algunas librerías, como JQuery⁵, que utiliza Javascripts y hace más amigables y usables las interfaces de usuario. Otro componente relevante en el sistema es Apache Solr⁶ que utilizando Lucene⁷, hace que la recuperación de la información sea rápida y eficiente.

⁵ <http://jquery.com>

⁶ <http://lucene.apache.org/solr>

⁷ <http://lucene.apache.org>

2.6 Modelo de casos de uso

El modelo de casos de usos permite que los desarrolladores y clientes lleguen a un acuerdo acerca de los requisitos de la aplicación. Es un modelo del sistema que contiene actores, casos de uso y sus relaciones.

2.6.1 Actores del sistema

Un actor es alguien o algo, externo al sistema, que interactúa con este y se beneficia de esa interacción. Puede ser una persona, un dispositivo de hardware u otro sistema.

Tabla1: Actores del sistema

Actores del sistema	Descripciones
Usuario	Persona que interactúa con la aplicación con el objetivo de ver todos los contenidos educativos expuestos.
Administrador	Persona encargada de administrar los datos de los usuarios que se registran en el sistema de notificación y de las acciones a realizar por dicho sistema.

2.6.2 Patrones de casos de uso

La experiencia en la utilización de casos de uso ha evolucionado en un conjunto de patrones que permiten con más precisión reflejar los requisitos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento. Dado un contexto y un problema a resolver, estas técnicas han mostrado ser la solución adoptada en la comunidad del desarrollo de software. A continuación se explican los patrones presentes en el diagrama de casos de uso del sistema.

Concordancia

Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado.

Reusabilidad: Consta de tres casos de uso. El primero llamado subsecuencia común, modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo. Los otros casos de uso modelan el uso del sistema que comparte la subsecuencia común de acciones. De manera que deben existir al menos dos de ellos.

Adición

La subsecuencia común extiende los casos de uso compartiendo la subsecuencia de acciones.

Este patrón se usa cuando otros casos de uso se encuentran propiamente completos, o sea que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema.

CRUD (Crear, Leer, Modificar, Eliminar)

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

Parcial: Este patrón alternativo modela uno de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja.

Múltiples actores

Roles comunes: Está presente cuando los dos actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso.

2.6.3 Diagrama de casos de uso de sistema

Un caso de uso es un fragmento de la funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más clara, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores. El diagrama de casos de uso del sistema es el modelo que contiene actores, casos de uso y las relaciones entre estos.

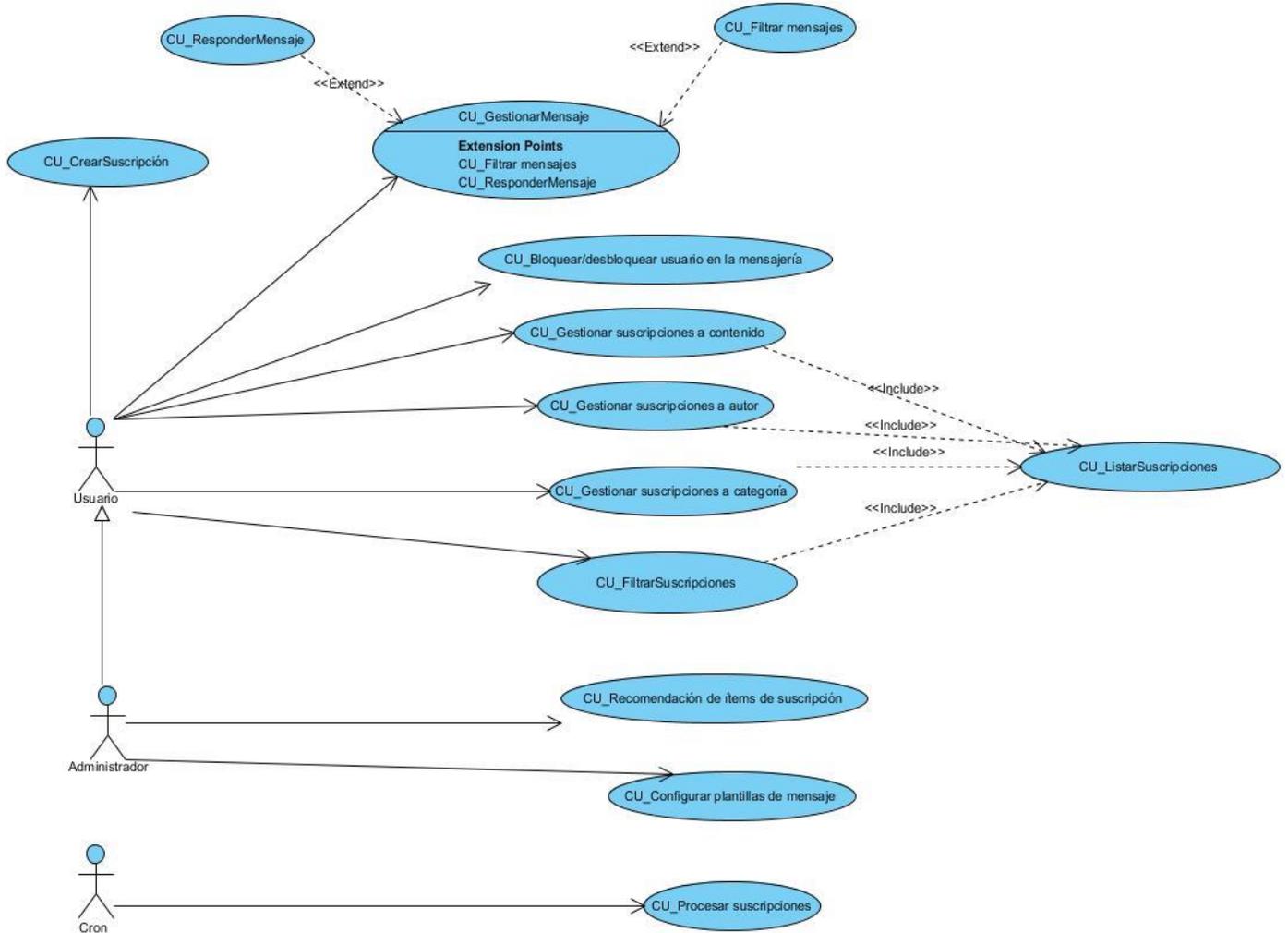


Figura 4 Diagrama de casos de uso

2.6.4 Descripciones de casos de uso

A continuación se describe el caso de uso que servirá de guía para entender el funcionamiento del mismo, las restantes descripciones se pueden consultar en Anexo 1 Descripción de los casos de uso.

Tabla 2. Descripción del CU Gestionar suscripción a contenidos.

Objetivo	Ver, editar y eliminar una suscripción a contenidos.
Actores	Usuario.
Resumen	El caso de uso inicia cuando el usuario decide ver, editar y eliminar una suscripción de contenidos, finalizando así el caso de uso.
Complejidad	Media.
Prioridad	Crítico.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado.
Postcondiciones	Se vio, editó y eliminó una suscripción de contenidos.

Flujo de eventos

Flujo básico Gestionar suscripción a contenidos

	Actor	Sistema
	1. Selecciona la opción "Suscripciones.	2. Muestra una interfaz con un listado de las suscripciones y por cada una las opciones de: <ul style="list-style-type: none"> • Ver Suscripciones (Ver Sección 1). • Editar Suscripciones (Ver Sección 2). • Eliminar Suscripciones (Ver Sección 3).
		3. Termina el caso de uso.

Sección 1 Ver suscripciones de contenidos

Flujo básico Ver suscripciones de contenidos

Actor	Sistema
1. Selecciona una suscripción de contenidos.	2. Muestra el contenido de la suscripción.
	3. Termina el caso de uso.

Sección 2 Modificar suscripciones de contenidos

Flujo básico Modificar Suscripciones de contenidos

Actor	Sistema
1. Selecciona la opción: "Editar".	2. Muestra una interfaz con una lista despegable para seleccionar el tipo de contenido y permite: <ul style="list-style-type: none">• Salvar suscripción.• Eliminar <i>"Ver sección 3 Eliminar suscripciones de contenidos"</i>• Cancelar.
3. Selecciona el nuevo tipo de contenido y la opción Salvar suscripción.	4. Guarda los cambios realizados.
	5. Termina el caso de uso

Flujos alternos

2. a Cancelar la operación

Actor	Sistema
1. Selecciona la opción Cancelar.	2. Cancela la operación.
	3. Termina el caso de uso.

Sección 3 Eliminar suscripciones de contenidos

Flujo básico Eliminar suscripciones de contenidos

Actor	Sistema
1. Selecciona la opción: "Eliminar".	2. Muestra una interfaz para eliminar la suscripción y permite: <ul style="list-style-type: none">• Eliminar suscripción.• Cancelar.
3. Selecciona la opción Eliminar.	4. Elimina la suscripción.
	5. Termina el caso de uso

Flujos alternos

3. a Cancelar la operación

Actor	Sistema
1. Selecciona la opción Cancelar.	2. Cancela la operación.
	3. Termina el caso de uso.

Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos funcionales	no	Rendimiento, Usabilidad, Apariencia o Interfaz externa.
Asuntos pendientes		

2.7 Conclusiones del capítulo

La realización del modelo de dominio y la descripción de los conceptos asociados al mismo fueron actividades que contribuyeron a una correcta captura de requisitos funcionales y no funcionales. Además se describió de manera detallada el módulo Notificaciones así como las nuevas fórmulas para el sistema recomendador basado en filtrado colaborativo. Como parte de la descripción del sistema también se identificaron y modelaron los casos de uso que guiarán el proceso de desarrollo del software; los cuales se describieron detalladamente de acuerdo a lo que plantea la metodología en uso.

CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

El propósito del análisis es transformar los requerimientos del sistema en un formato que se corresponda con el área de conocimiento de los diseñadores del software. De una forma u otra, se encarga de que los requerimientos funcionales del sistema sean manipulables (39), ignorando con propósitos de simplicidad gran parte de los no funcionales, así como las restricciones del ambiente de implementación. Este constituye una abstracción del modelo de diseño.

3.2 Modelo de análisis

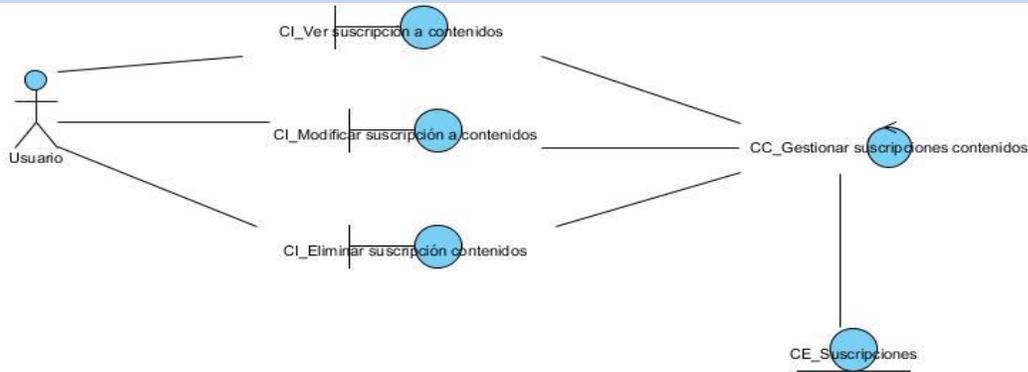
El modelo de análisis contiene el resultado del análisis de los casos de uso, instanciados en el artefacto diagrama de clases de análisis. Este diagrama está conformado por las clases interfaz, clases control y clases entidad. A continuación se presentan algunos de los diagramas de clases de análisis asociados a los casos de uso del módulo en desarrollo.

3.2.1 Diagrama de clase del análisis

En la elaboración del modelo del análisis se deben identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el diagrama de clases del análisis (DCA), un ejemplo de este se muestra en la Figura 3 correspondiente al casos de usos descrito anteriormente y los restantes se pueden observar en Anexo 2 Diagrama de clase del análisis.

Tabla 3 DCA Gestionar suscripciones de contenido

Gestionar suscripciones de contenido

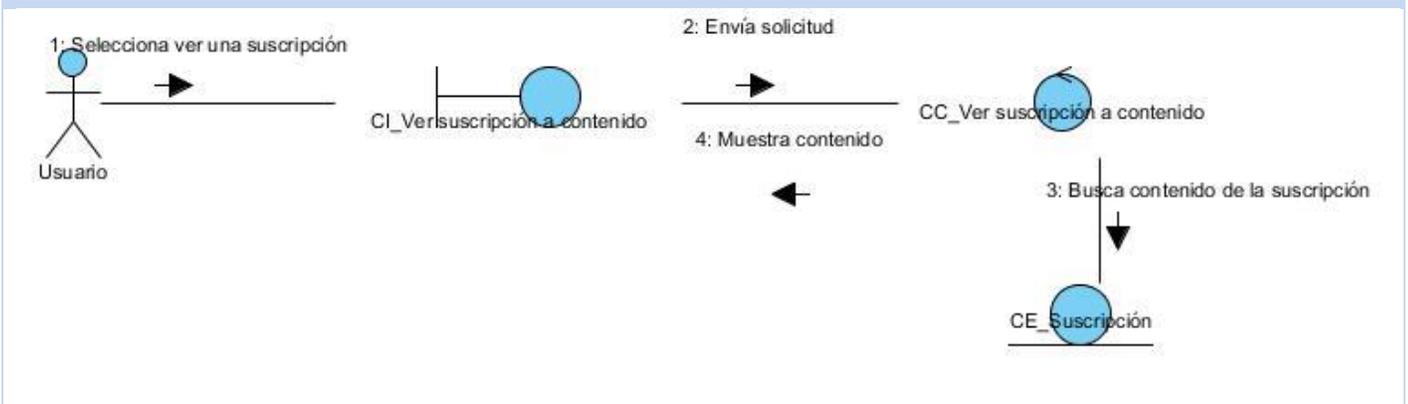


3.2.2 Diagramas de colaboración

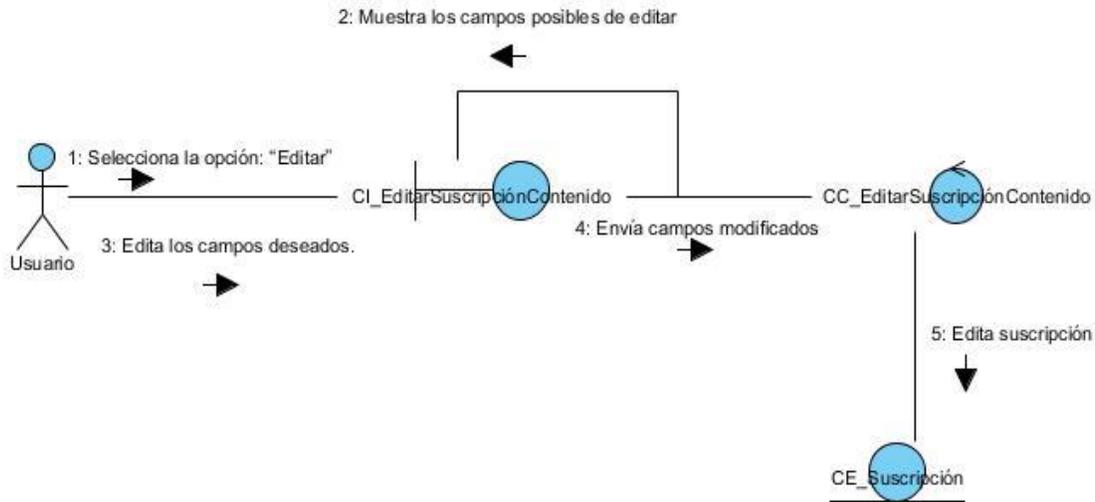
Un diagrama de colaboración (DC) es un diagrama de interacción que modela los objetos y enlaces implicados en la implementación. Se utilizan para ilustrar la realización de un CU. Este muestra las interacciones organizadas alrededor de instancias y los enlaces entre ellas. Se presentan los diagramas de colaboración correspondientes a los casos de usos descritos anteriormente y los restantes se pueden observar en Anexo 3 Diagramas de colaboración.

Tabla 4 Diagrama de colaboración Gestionar suscripciones de a contenido

Ver suscripciones a contenido



Editar suscripciones a contenido



Eliminar suscripciones a contenido



3.3 Modelo de diseño

El propósito principal del diseño es conjugar el resultado del análisis con las restricciones impuestas por los requisitos no funcionales, el ambiente de implementación, los requisitos de rendimiento, entre otros (39), dando como resultado un modelo preciso del producto final con un cubrimiento total de los

requerimientos. A diferencia del análisis, que en muchos proyectos tiene carácter opcional, el diseño es considerado una disciplina clave dentro de la ingeniería de software, siendo el centro de atención al final de la fase de elaboración e inicio de la fase de construcción, y enfocándose en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema a desarrollar.

El artefacto más importante dentro de la disciplina de diseño lo constituye el modelo de diseño, que no es más que un modelo de objetos que describe la realización de los casos de uso, y sirve como una abstracción del modelo de implementación y del código fuente. Más adelante se muestran varios artefactos de la aplicación en desarrollo intrínsecamente asociados a este modelo de diseño, como son diagramas de interacción y los diagramas de clases de diseño.

Además, se muestra el modelo de datos considerado también como parte de esta disciplina.

3.3.1 Aplicación de patrones de diseño

Un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en diversas circunstancias (40), es decir, es un par problema-solución al que se le asigna un nombre y puede ser aplicado a varias situaciones, teniendo en cuenta las características del nuevo contexto en que se va a usar.

En la implementación del módulo Notificaciones están reflejados los siguientes patrones de diseño: Bridge, Decorator, Chain of Responsibility, Command, Singleton que garantizan que el código sea más legible.

3.3.2 Diagramas de clases de diseño

Un diagrama de clases del diseño representa las clases del sistema y sus relaciones. Este contiene una clase servidora que construye la clase cliente que está compuesta por un formulario que realiza envíos a la servidora; la cual accede a la controladora y esta a su vez a las entidades. A continuación se presentan los diagramas de clases del diseño correspondiente a los casos de usos descritos anteriormente y los restantes se pueden observar en [Anexo 4 Diagrama de clase del diseño](#)

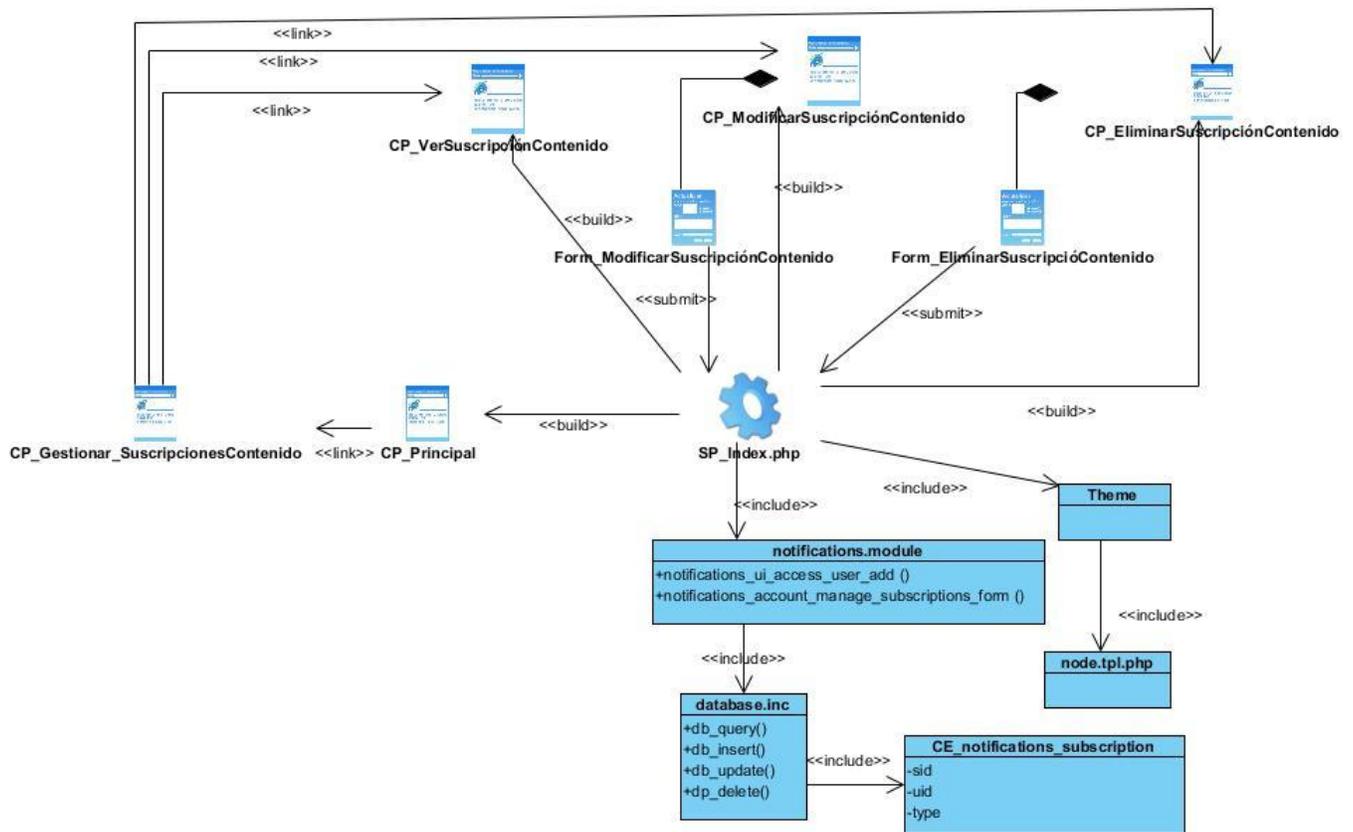


Figura 5 DCD Gestionar suscripción a contenido

3.3.3 Modelo de datos

Un modelo es una representación de cualquier aspecto o tema extraído del mundo real. En un enfoque más relacionado al manejo de datos, se puede decir que un modelo de datos es un lenguaje utilizado para la descripción de una base de datos; que permite representar los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí.

Los modelos de datos presentan dos sublenguajes:

- Lenguaje de Definición de Datos o DDL (Data Definition Language): Describe de una forma abstracta las estructuras de datos y las restricciones de integridad.

- Lenguaje de Manipulación de Datos o DML (Data Manipulation Language): Describe las operaciones de manipulación de los datos. Además la parte del DML enfocada a la recuperación de datos, se le suele conocer como Lenguaje de Consulta o QL (Query Language).

Se clasifican según su nivel de abstracción en tres modelos:

- Modelo de datos conceptuales: Describen las estructuras de datos y restricciones de integridad. Se utilizan durante la etapa de análisis de un problema dado y están orientados a representar los elementos que intervienen y sus relaciones.
- Modelo de datos lógicos: Se centran en las operaciones y se implementan en algún manejador de base de datos.
- Modelos de datos físicos: Son estructuras de datos a bajo nivel implementadas dentro del propio manejador.

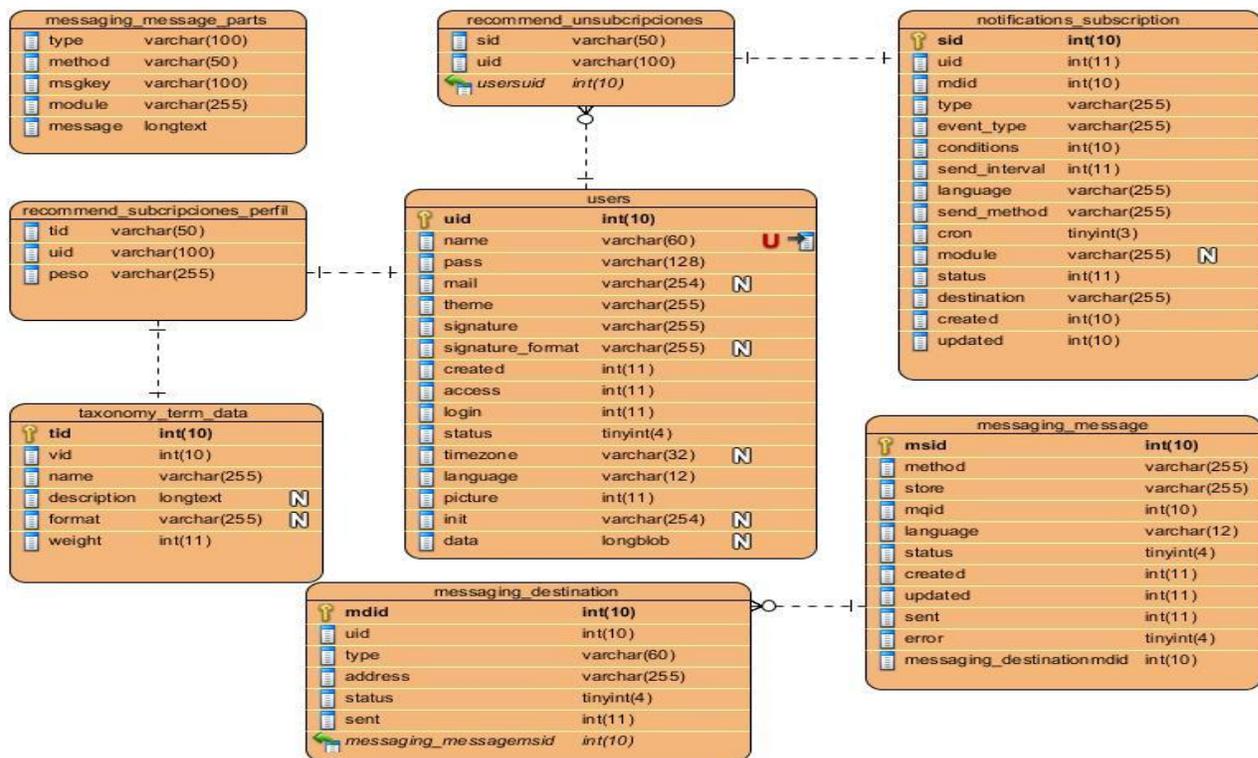


Figura 6 Modelo de datos

3.4 Conclusiones del capítulo

A lo largo del capítulo fueron descritos todos los aspectos que tuvieron un impacto significativo en el diseño del software en cuestión. Es importante destacar que el CMS Drupal cuenta con una serie de patrones de diseño que, en su conjunto, ponen de manifiesto el uso de buenas prácticas en la codificación del sistema. Además fueron representados en sus respectivos diagramas el modelo de análisis y diseño para uno de los casos de uso de la aplicación y el de base de datos, artefactos que son la entrada al flujo de trabajo implementación.

CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBAS

4.1 Introducción

En el presente capítulo se describen las características del funcionamiento del módulo Notificaciones, además se muestra cómo va a ser desarrollada la aplicación, la cual se inicia con los artefactos derivados del análisis y diseño y continúa con la construcción del sistema en términos de componentes, obteniéndose los diagramas de despliegue y de componentes; también se presenta la fase de pruebas del producto final, en la cual se diseñan las pruebas de caja negra a través de la confección de los casos de prueba para garantizar la calidad del producto.

4.2 Estándares de codificación

El proceso de codificación incluye el conjunto de reglas o normas usadas para escribir código, lo cual aporta eficiencia al proceso de desarrollo, logrando que los programas sean más comprensibles y de mayor calidad.

Para el desarrollo del módulo se tuvieron en cuenta algunos de los estándares de codificación de Drupal definidos en drupal.org para garantizar la compatibilidad con el *framework*.

4.2.1 Etiquetas de apertura y cierre de php

Cuando estemos escribiendo en PHP, siempre se deben utilizar las etiquetas `<?php y ?>`, y en ningún caso la versión corta `<? y ?>`.

En general se omite la etiqueta de cierre de PHP (`?>`) al final de los archivos `.module` y `.inc`. Esta convención evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre `?>`), que serían identificados como salida HTML.

Por tanto, la etiqueta de cierre final del archivo (`?>`) es opcional en Drupal.

No hay que confundir esto con el uso normal del lenguaje PHP en archivos que también contienen HTML (como por ejemplo los archivos de plantilla .tpl.php), donde cada fragmento de PHP debe llevar sus correspondientes etiquetas de apertura y cierre, para diferenciarlo del código HTML.

4.2.2 Operadores

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, `$numero = 3`, en lugar de `$numero=3`. Esto se aplica a operadores como `+`, `-`, `*`, `/`, `=`, `==`, `!=`, `>`, `<`, `.` concatenación de cadenas), `.=`, `+=`, `-=`, etc.

Los operadores unarios como `++`, `--` no deben tener separación. Por ejemplo, `$numero++`.

4.2.3 Uso de comillas

Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres.

Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, `"<h1>${title}</h1>"`. También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple.

4.2.4 Uso de punto y coma (;) en código php

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo `<?php print $title ?>`. En Drupal es siempre obligatorio: `<?php print $title;?>`.

4.2.5 Funciones

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo. Además, se debe incluir siempre como prefijo el nombre del módulo, tema, etc., para evitar así duplicidad de funciones.

En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior.

4.2.6 Arrays

Los valores dentro de un array (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados.

Cuando la línea de declaración del array supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

4.2.7 Nombres de archivos

Los nombres de archivos deben escribirse siempre en minúsculas. La única excepción son los archivos de documentación, que tendrás extensión .txt y el nombre es mayúsculas. Por ejemplo, README.txt, INSTALL.txt.

4.3 Modelo de implementación

La organización de los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en los lenguajes de programación utilizados es descrita en el modelo de implementación. Este modelo además describe la forma en que los elementos obtenidos del modelo de diseño como las clases) se implementan en términos de componentes y las dependencias que presentan dichos componentes entre sí. El modelo de implementación es la entrada principal para las pruebas de software que siguen a la etapa de implementación (23).

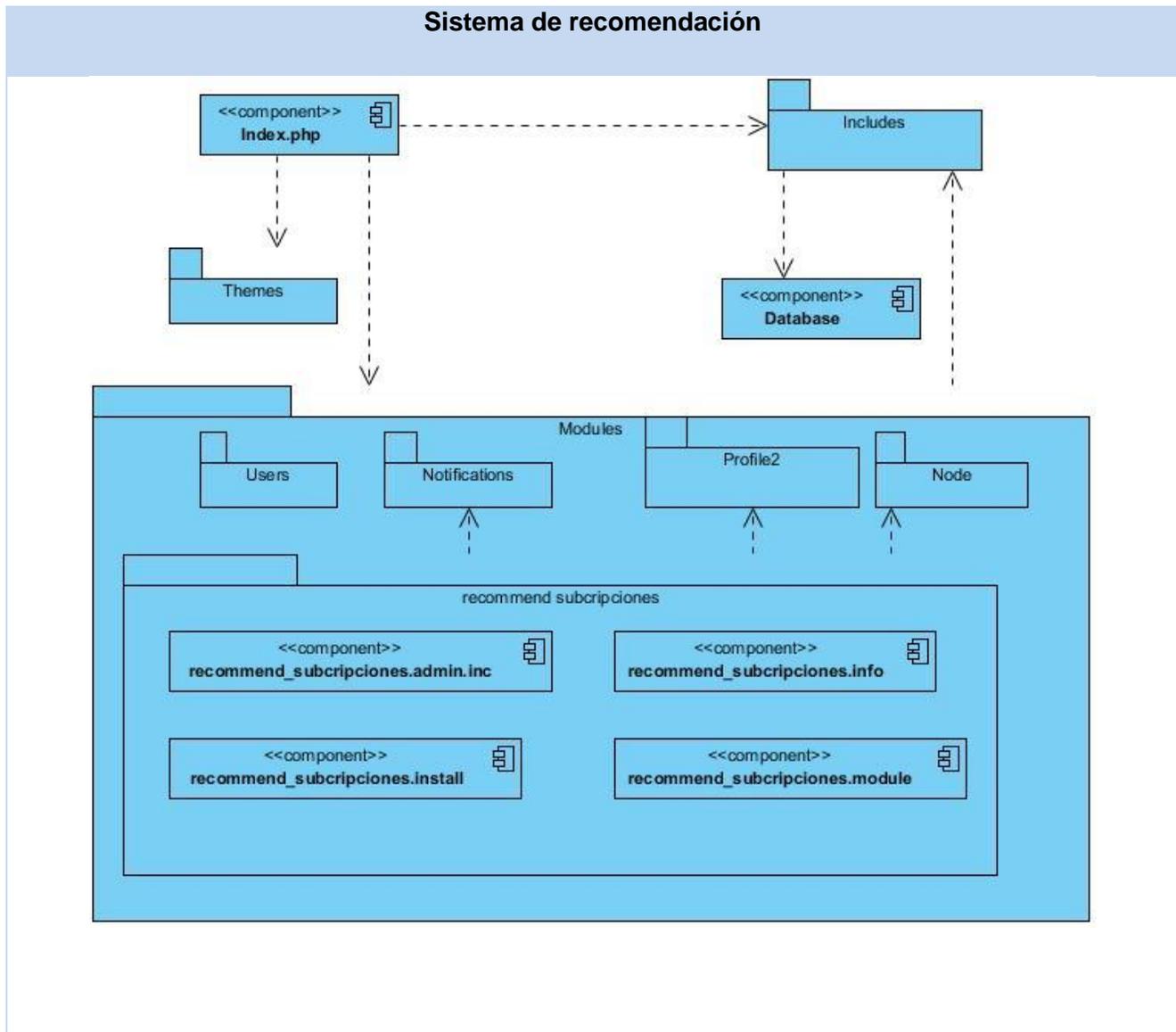
4.3.1 Diagrama de componentes

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño (23). El diagrama de componentes se utiliza para modelar la vista estática de un

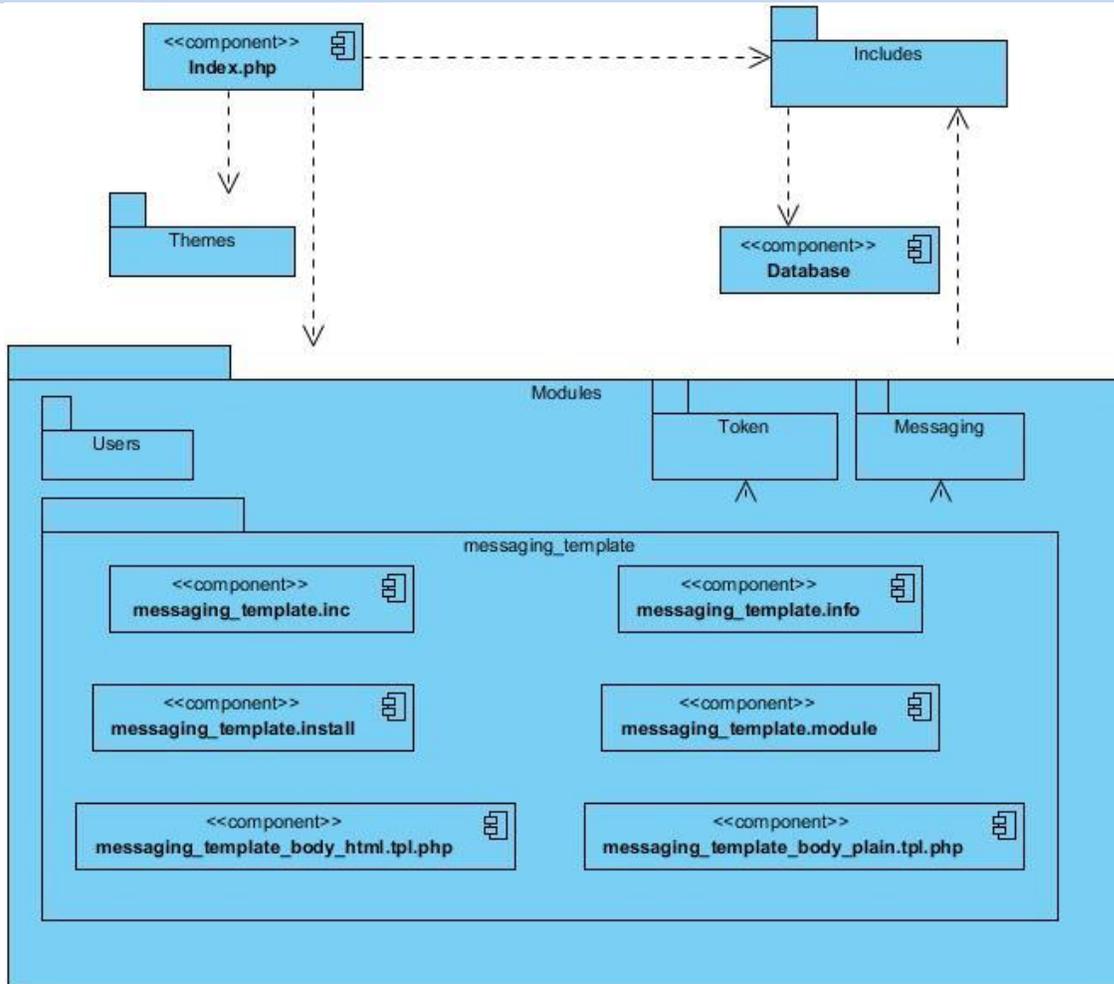
sistema. Se encarga de mostrar los componentes de software y las relaciones lógicas entre ellos en un sistema, además de estructurar el modelo de implementación en términos de subsistemas de implementación.

A continuación se muestra el diagrama de componentes para la implementación del software:

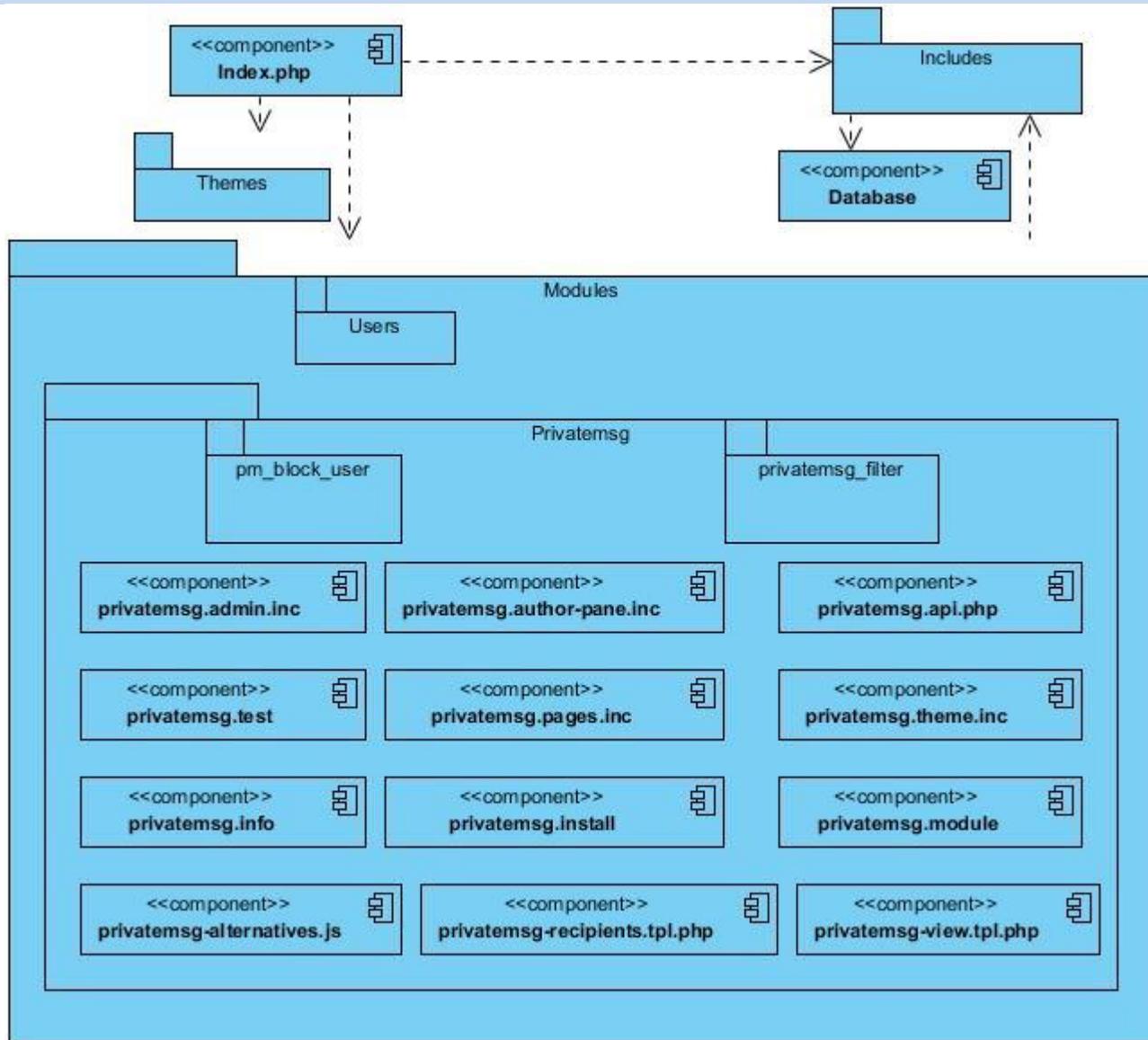
Tabla 5 Diagramas de componentes



Plantillas de mensaje



Mensajería interna



4.3.2 Diagrama de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos;

donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar (38).

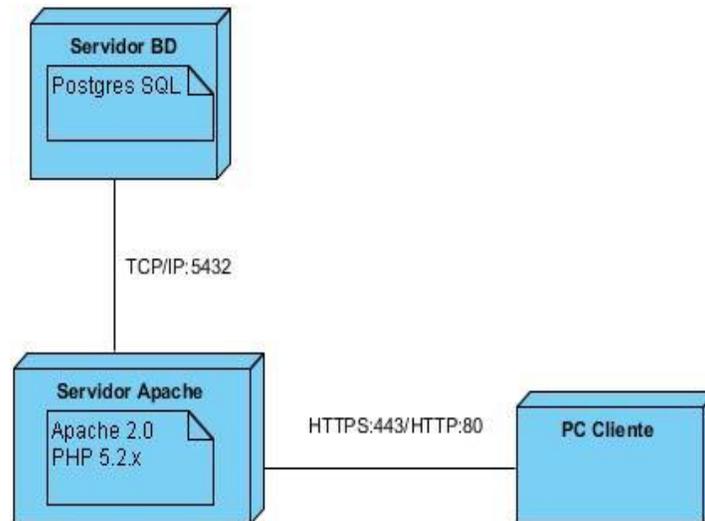


Figura 7 Diagrama de despliegue

4.4 Pruebas

Una vez generado el código fuente, el software como producto puede tener defectos o fallos, por lo que es necesario probarlo para descubrir y corregir la mayor cantidad de errores posibles, para así garantizar un producto con calidad.

Existen diferentes niveles de prueba de software, entre las que se encuentran:

- **Pruebas unitarias:** La prueba de unidad se centra en el módulo, usando la descripción del diseño detallado como guía, se prueban los caminos de control importantes con el fin de descubrir errores dentro del ámbito del módulo.
- **Pruebas de requerimientos:** Un asunto de gran importancia es asegurar la corrección, coherencia y exactitud de los requisitos. Se revisará el documento de especificación de requerimientos, con la lista de chequeo general del documento y la lista de chequeo de requerimientos.

- Pruebas de integración: Las pruebas de integración se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto.
- Pruebas de aceptación: Las pruebas de aceptación, son las que se hacen con los clientes y define su aceptación del sistema. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario.
- Pruebas de Sistema: En el diseño de sistemas, estos no se toman como sistemas completos ni se prueban como sistemas únicos, razón por la cual deben hacerse pruebas parciales y del sistema en su totalidad.
- Pruebas Funcionales: Se asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

Para probar la solución se escogen las pruebas funcionales, ya que este tipo de pruebas están basadas en técnicas de caja negra, la cual verifica la aplicación (y sus procesos internos) mediante la interacción con la aplicación y analizar la salida. (41)

4.5 Pruebas de Caja Negra

“Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.” (42)

La prueba de caja negra intenta encontrar errores de las siguientes categorías (42):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

4.6 Resultados de las pruebas

Para evaluar el módulo desarrollado se realizaron 3 iteraciones de pruebas en las cuales se probó el módulo a un alto grado de detalle. Se tuvieron en cuenta los métodos y técnicas de pruebas expuestas anteriormente en cada una de las iteraciones, arrojando resultados visibles, los cuales demuestran la calidad del módulo. A continuación se muestra la cantidad de no conformidades identificadas en las iteraciones realizadas:

Tabla 6 Cantidad de no conformidades por iteraciones

Clasificación NC	1ra Iteración	2da Iteración	3ra Iteración
Alta	3	2	0
Media	2	1	0
Baja	2	1	1

De manera más detallada se muestra en un gráfico de cómo se comporta la cantidad de no conformidades encontradas en cada una de las iteraciones realizadas:

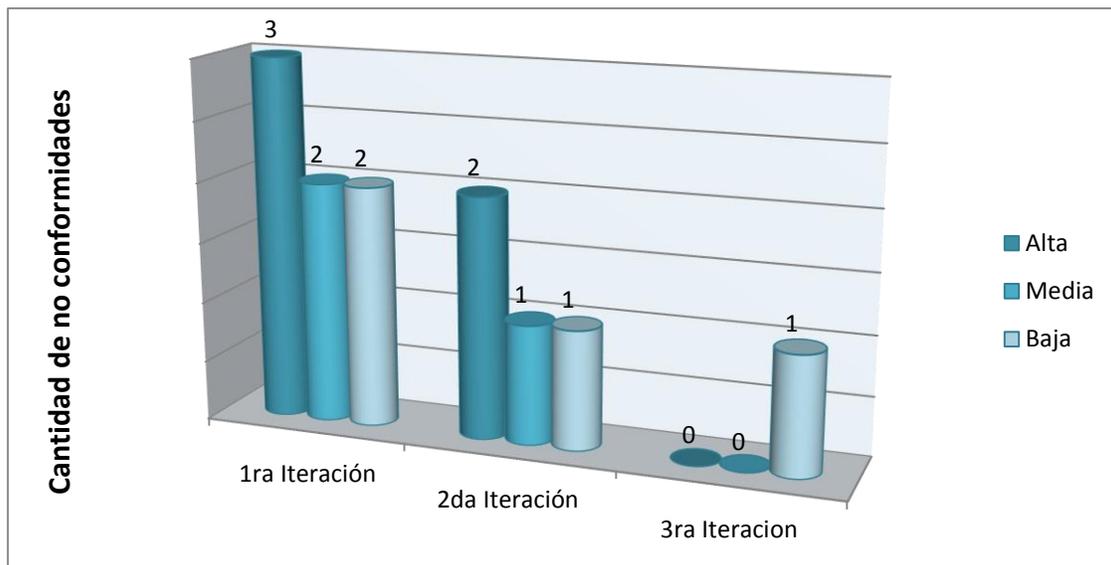


Figura 8 Cantidad de no conformidades detectadas en cada iteración

Las pruebas se realizaron de forma iterativa e incremental, comprobando en cada iteración que hubiesen sido corregidos los errores detectados en la iteración anterior, lo que contribuyó a mejorar la calidad y funcionalidad del módulo.

Después de terminada la implementación del módulo el mismo se le integró a las comunidades virtuales y fueron seleccionados 20 usuarios para que se conectaran a la aplicación durante tres días. El primer día sin el módulo y los dos últimos días con el módulo incluido.

A continuación se muestra una tabla con el comportamiento de las variables durante los tres días.

Tabla 7 Comportamiento de las variables durante de los tres días

Variables/Días	1	2	3
Tiempo de respuesta promedio	30 min	20 min	15 min
Cantidad de mensajes publicados en el sitio	5	10	20
La publicación de tipos de contenidos	3	15	20
Cantidad de mensajes enviados a otros usuarios	10	20	30
Cantidad de suscripciones promedio	30	60	70

Como se puede apreciar en la tabla anterior después de los tres días de probado el módulo se logró la comunicación entre los usuarios de Sistema de Gestión de Contenido Educativo por lo que la hipótesis planteada es válida.

4.7 Conclusiones del capítulo

El desarrollo del modelo de despliegue y los diagramas de componentes permitieron modelar la estructura física de los nodos que componen el sistema, y la relación entre cada uno de los elementos que intervienen en el mismo, respectivamente. La realización de las pruebas y el análisis de sus resultados permitieron determinar y erradicar las deficiencias encontradas, garantizando la calidad del sistema. Por lo que se concluye que la incorporación del módulo Notificaciones a la gestión de comunidades virtuales aumenta la comunicación entre los usuarios de la comunidad universitaria.

CONCLUSIONES

La concepción y desarrollo del módulo Notificaciones para el Portal Educativo facilitó arribar a las siguientes conclusiones:

- El estudio de los sistemas de notificaciones y los sistemas de recomendación permitieron identificar las herramientas, tecnologías, lenguajes y metodologías para guiar el proceso de desarrollo y discriminar los elementos que no se ajustaban al desarrollo de la investigación.
- Mediante el análisis y diseño fue posible obtener una mayor comprensión del sistema y se definieron los principios y modelos para realizar la implementación de la solución obtenida.
- El desarrollo del módulo Notificaciones facilita la comunicación de los usuarios con el entorno virtual de SIAMU.
- La realización de las pruebas y el análisis de sus resultados permitieron determinar y erradicar las deficiencias encontradas, garantizando la calidad del sistema.

RECOMENDACIONES

A partir del trabajo realizado se recomienda al proyecto Portal Educativo:

- Aplicar técnicas de minería de datos al historial de suscripciones de los usuarios.
- Realizar la integración de otros medios de comunicación personal como es el caso de la telefonía celular en el módulo de suscripciones.
- Mejorar el recomendador a partir de su transformación de sistema colaborativo a híbrido con el fin de solucionar los problemas de los sistemas colaborativos.

BIBLIOGRAFÍA

1. **Bautista, Juan.** El blog de JUANBAUTISTA. *El blog de JUANBAUTISTA*. [En línea] 20 de noviembre de 2007. <http://comunidadesvirtuales.obolog.com/tic-conceptualizacion-caraterizacion-tecnologias-informacion-40188> caracterizacion-tecnologias-informacion-40188..
2. *Blended Learning. Conceptos básicos.* **Bartolomé, Antonio.** 23, Barcelona : Píxel-Bit, 2004.
3. **BARRIO, F. G.** *Los Portales Educativos como fuentes de Recursos Materiales.* 2006.
4. **ASCARZA, A. B.** Técnicas e indicadores para la evaluación de portales educativos en Internet. [En línea] 2006. http://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/administracion/N14_2005/a09.pdf.
5. **GRAELLS, P. M.** Diseño de páginas Web educativas. [En línea] 2000. <http://dewey.uab.es/pmarques/diswguio.htm>.
6. **e-abclearning.** e-abclearning. *e-abclearning*. [En línea] 2011. <http://www.e-abclearning.com/queesunaplataformadeelearning>.
7. **López Guzmán, Clara.** *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning.* Salamanca : s.n., 2005.
8. **GONZÁLEZ, L. R.** *Análisis y Diseño de la versión 3.0 de RHODA.* La Habana, Cuba : s.n., 2011. pág. 80p.
9. **Programas, Portal.** Portal Programas. [En línea] <http://gratis.portalprogramas.com/MDaemon.html>.
10. **Descargarte.** Descargarte. [En línea] <http://www.descargarte.net/2008/11/mailenable-servidor-de-correo-para.html>.
11. **Tom Klingenberg.** PHPmailer. [En línea] 2008. <http://phpmailer.worxware.com/index.php?pg=tutorial>.
12. **sentidoweb.** [En línea] 2008. <http://sentidoweb.com/2008/07/30/swift-mailer-libreria-php-para-envio-de-emails.php> php-para-envio-de-emails.php.
13. **RAE.** [En línea] lema.rae.es/drae/srv/search?key=notificación.
14. **WEBADICTOS.** Notificaciones Facebook en tu escritorio. [En línea] <http://www.webadictos.com.mx/2010/01/13/notificaciones-facebook-en-tu-escritorio>.
15. **Dioses, Sin.** Sin Dioses. [En línea] <http://www.sindioses.org/redifusion.html>.
16. **RSS.** RSS Specifications. *RSS Specifications*. [En línea] <http://www.rss-specifications.com/rss-specifications.htm>.

17. **WWW, Sindicación de contenidos en la.** [En línea] 20 de noviembre de 2011.
http://cv.uoc.edu/~cv081_79_007_01_w01/garcia_sanchez/PEC_2/formatos.html.
18. **Nieto, Sergio Manuel Galán.** Filtrado Colaborativo y Sistemas de Recomendación.
19. **Manuel Fernando Caro Piñeres, Jaime Hernández, Jovani Alberto Jiménez Builes.** DISEÑO DE UN SISTEMA DE RECOMENDACIÓN EN REPOSITORIOS DE UN OBJETO DE APRENDIZAJE. *DISEÑO DE UN SISTEMA DE RECOMENDACIÓN EN REPOSITORIOS DE UN OBJETO DE APRENDIZAJE.* Bogotá : s.n., 2011. ISSN 0124-8170.
20. **Toledo, Raciél Yera.** Concepción y desarrollo de un sistema de recomendación para jurados online de programación. La Habana : s.n., 2010.
21. **ESPAÑOLA, REAL ACADEMIA.** DICCIONARIO DE LA LENGUA ESPAÑOLA - Vigésima segunda edición. [En línea] http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=informaci%C3%B3n..
22. **Angelfire.** ¿Qué es la ingeniería de Software? [En línea] 16 de 12 de 2011.
www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html..
23. **Jacobson, Ivar, Booch, Grady and Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software* . s.l. : Pearson Educacion , 2000.
24. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado. Manual de referencias.* California : s.n., 1998.
25. **Maldonado, Daniel M.** El CoDiGo K. *El CoDiGo K.* [En línea] 2007 de septiembre de 2007.
<http://elcodigok.blogspot.com/2007/09/que-son-los-ide-de-programacin.html>.
26. **Maldonado, Daniel M.** El CoDiGo K. *El CoDiGo K.* [En línea] 2010 de septiembre de 2010.
<http://www.elcodigok.com.ar/2010/09/7-caracteristicas-de-netbeans-6-9-1-integrado-a-php-a-php/>.
27. **ITESCAM.** Instituto Tecnológico Superior de Calkiní en el Estado de Campeche. [En línea] 2012.
www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r19670.DOC .
28. **Limited, Visual Paradigm International.** *Visual Paradigm for UML 8.0 Released.* 2010.
29. **Manager, Free Download.** Free Download Manager. . [En línea] .
30. **Uazuay.** uazuay. *uazuay.* [En línea] 2011.
http://www.uazuay.edu.ec/estudios/sistemas/lenguaje_iii/MAnnualJavaScript/introduccion.htm.
31. **PHP.** PHP. [En línea] 2013. www.php.net.

32. **W3C**. Guía Breve de Tecnologías XML. [En línea] 2011.
<http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>.
33. **Tomlinson, John K. VanDyk Todd**. *Pro Drupal 7 Deveploment Drupal 7 Development*. 2010.
34. **Integrator, CAVSI Computer Audio Video Systems**. CAVSI Computer Audio Video Systems Integrator. [En línea] 8 de enero de 2012. [Citado el: 8 de enero de 2012.]
<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-base-de-datos-o-sgbd>.
35. **Aguilar, Vicente y Suau, Pablo**. bisente. [En línea] [Citado el: 8 de enero de 2013.]
<http://www.bisente.com/documentos/mysql-postgres.html>.
36. **disalzar**. Tecnoweb2.com. *Tecnoweb2.com*. [En línea] 5 de septiembre de 2011.
<http://tecnoweb2.com/que-es-servidor-web>.
37. **Apache**. Apache. [En línea] [Citado el: 15 de enero de 2013.]
http://httpd.apache.org/docs/2.0/es/new_features_2_0.html.
38. **Wesley, Addison**. *El proceso unificado de desarrollo de software*. 1999.
39. **Dominio, Modelo de**. [En línea]
http://iie.fing.edu.uy/ense/asign/desasoft/practico/hoja8/ejemplos_clase2.pdf.
40. **Larman, Graig**. *UML y patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall Hispanoamericana, S.A. pág. 167.
41. **Torres, Paticio Letelier**. <http://es.scribd.com/doc/19317161/57/Diagrama-de-Componentes>. [En línea] Universidad Politécnica de Valencia.
42. **Pressman, Roger S**. *Ingeniería de Software un enfoque práctico*. Quinta Edición.
43. **Lavin Cuello, Alejandro y Reyes Pupo, Hector Luis**. Desarrollo de un módulo para gestionar los cuestionarios interactivos en la coelccion El Navegante en su version multiplataforma cuestionarios interactivos en la Colección El Navegante en su versión multiplataforma. La Habana : s.n., 2010.
44. **Alonso, Randy Falcón**. Cubadebate. [En línea] 9 de Mayo de 2009. <http://cubadebate.cu>.
45. **BARRIO, F. G**. *Los Portales Educativos como fuentes de Recursos Materiales*. 2006.
46. **Peña., Juan Carlos Sepúlveda**. *Plataforma de Teleformación aprendIST*.
47. **Sampier, Roberto Hernández**. *Metodología de la investigación*. La Habana : Felix Varela, 2008.

ANEXOS

Anexo 1 Descripción de los casos de uso

Tabla 8 CU Gestionar mensaje

Objetivo	Crear, eliminar y visualizar un mensaje.	
Actores	Usuario.	
Resumen	El caso de uso inicia cuando el usuario accede a la mensajería privada. El sistema permite crear, eliminar y visualizar un mensaje, finalizando así el caso de uso.	
Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>El usuario debe tener permiso de uso de la mensajería privada.</p>	
Postcondiciones	Se creó, eliminó y visualizó un mensaje.	
Flujo de eventos		
Flujo básico Revisar bandeja de entrada		
	Actor	Sistema
	1. Selecciona la opción Mensajes.	2. Muestra la Bandeja de entrada seleccionada de forma predeterminada, donde se listan los mensajes recibidos (50 por páginas) y de cada uno: <ul style="list-style-type: none"> • Asunto • Participantes

		<ul style="list-style-type: none"> • Fecha de la última respuesta <p>Y permite:</p> <ul style="list-style-type: none"> • Crear nuevo mensaje. <i>Ver sección 1: "Crear nuevo mensaje"</i> • Visualizar mensaje. <i>Ver sección 2: "Visualizar mensaje"</i> • Eliminar mensaje. <i>Ver sección 3: "Eliminar mensaje"</i>. • Ver mensajes enviados. <i>Ver sección 4 "Ver mensajes enviados"</i> • Ver todos los mensajes. <i>Ver sección 5 "Ver todos los mensajes"</i> • Bloquear usuario. <i>Ver CU "Bloquear usuario"</i>. • Filtrar mensaje. <i>Ver CU "Filtrar mensaje"</i>
		3. Termina el caso de uso.
Sección 1 Crear nuevo mensaje		
Flujo básico Crear nuevo mensaje		
	Actor	Sistema
	1. Selecciona la opción Escribir nuevo mensaje.	2. Muestra la interfaz para crear un nuevo mensaje que permite introducir los siguientes campos: <ul style="list-style-type: none"> 6. Para 7. Asunto 8. Cuerpo del mensaje <p>Y realizar las acciones:</p> <ul style="list-style-type: none"> 9. Enviar mensaje. 10. Cancelar.
	3. Introduce los campos y selecciona la opción Enviar mensaje.	4. Valida los datos, crea el mensaje y lo envía.

		5. Termina el caso de uso.
Flujos alternos		
3. a Existen datos incorrectos (dirección de correo electrónico)		
	Actor	Sistema
		1. Muestra el mensaje de información: “La dirección de correo es incorrecta” Y lo señala en color rojo.
		2. Regresa al paso 3 del flujo básico.
3. b Cancelar la operación		
	Actor	Sistema
	1. Selecciona la opción Cancelar.	2. Cancela la operación.
		3. Termina el caso de uso.
3. c Existen campos obligatorios vacíos		
	Actor	Sistema
		1. Si el usuario deja el campo del destinatario muestra el mensaje de error: “El campo Para es obligatorio.” Y lo señala en color rojo. 2. Si el usuario deja el campo del Asunto muestra el mensaje de error: “El campo Asunto es obligatorio.” Y lo señala en color rojo. 3. Si el usuario deja el campo del Mensaje muestra el mensaje de error: “El campo Mensaje es obligatorio.” Y lo señala en color rojo.

		4. Regresa al paso 3 del flujo básico.
Sección 2 Visualizar mensaje		
Flujo básico Visualizar mensaje		
	Actor	Sistema
	1. Selecciona el mensaje que desea visualizar realizando un clic encima.	2. Muestra el contenido del mensaje y permite: <ul style="list-style-type: none"> • Responder el mensaje. Ver CU “Responder mensaje” • Eliminar mensaje. Ver sección 3: “Eliminar mensaje”.
		3. Termina el caso de uso.
Sección 3 Eliminar mensaje		
Flujo básico Eliminar mensaje		
	Actor	Sistema
	1. Selecciona el o los mensajes que desea eliminar y luego la opción Eliminar.	2. Elimina el mensaje.
		3. Termina el caso de uso.
Sección 4 Ver mensajes enviados		
Flujo básico Ver mensajes enviados		
	Actor	Sistema
	1. Selecciona la opción Mensajes enviados.	2. Muestra un listado de los mensajes enviados (50 por páginas) y de cada uno: <ul style="list-style-type: none"> • Asunto • Participantes • Fecha del envío

		<p>Y permite:</p> <ul style="list-style-type: none"> • Crear nuevo mensaje. <i>Ver sección 1: "Crear nuevo mensaje"</i> • Visualizar mensaje. <i>Ver sección 2: "Visualizar mensaje"</i> • Eliminar mensaje. <i>Ver sección 3: "Eliminar mensaje"</i>. • Ver todos los mensajes. <i>Ver sección 5 "Ver todos los mensajes"</i> • Bloquear usuario. <i>Ver CU "Bloquear usuario"</i>. • Filtrar mensaje. <i>Ver CU "Filtrar mensaje"</i>
		3. Termina el caso de uso.

Sección 5 Ver todos los mensajes

Flujo básico Ver todos los mensajes

	Actor	Sistema
	1. Selecciona la opción Todos los mensajes.	<p>2. Muestra un listado de todos los mensajes recibidos y enviados (50 por páginas) y de cada uno:</p> <ul style="list-style-type: none"> • Asunto • Participantes • Fecha del mensaje <p>Y permite:</p> <ul style="list-style-type: none"> • Crear nuevo mensaje. <i>Ver sección 1: "Crear nuevo mensaje"</i> • Visualizar mensaje. <i>Ver sección 2: "Visualizar mensaje"</i> • Eliminar mensaje. <i>Ver sección 3: "Eliminar mensaje"</i>. • Ver los mensajes enviados. <i>Ver sección 5 "Ver los mensajes enviados"</i> • Bloquear usuarios. <i>Ver CU "Bloquear usuario"</i>.

		<ul style="list-style-type: none"> Filtrar mensaje. Ver CU "Filtrar mensaje"
		3. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	Filtrar mensaje: Paso 1 del Flujo básico.
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	
Asuntos pendientes		

Tabla 9 CU Responder mensaje

Objetivo	Permite responder el mensaje al remitente.	
Actores	Usuario.	
Resumen	El caso de uso inicia cuando el usuario accede a la mensajería privada. El sistema permite responder el mensaje al remitente.	
Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones	El usuario debe estar registrado y debe tener permiso de uso de mensajería privada.	
Postcondiciones	El usuario podrá responder los mensajes.	
Flujo de eventos		
Flujo básico Visualizar mensaje		
	Actor	Sistema

	1. Selecciona la opción “Responder mensaje”.	2. Muestra área para responder y permite: <ul style="list-style-type: none"> • Enviar mensaje. • Limpiar
	3. Responde el mensaje y da la opción “Enviar”.	4. Envía el mensaje.
		5. Termina el caso de uso.

Flujos alternos

1. a Limpiar

	Actor	Sistema
	1. Selecciona la opción Limpiar.	2. Borra el texto del mensaje.
		3. Termina el caso de uso.

Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	
Asuntos pendientes		

Tabla 10 CU Bloquear/desbloquear usuario en la mensajería

Objetivo	Permite bloquear y desbloquear usuarios para que no puedan enviar mensajes ni recibir comentarios.
Actores	Usuario.
Resumen	El caso de uso inicia cuando el usuario accede a la mensajería privada. El sistema permite responder

	bloquear/desbloquear a los usuarios con los que no quiere interactuar.	
Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones	El usuario debe estar registrado y debe tener permiso de uso de mensajería privada.	
Postcondiciones	El usuario bloqueará o desbloqueará a otros usuarios.	
Flujo de eventos		
Flujo básico Bloquear/desbloquear usuario en la mensajería		
	Actor	Sistema
	1. Selecciona la opción "Bloquear".	2. Muestra una interfaz para introducir el nombre del usuario a bloquear y una lista con los usuarios bloqueados. Permite: <ul style="list-style-type: none"> • Bloquear usuario Ver Sección 1: "Bloquear usuario". • Desbloquear usuario Ver Sección 2: "Desbloquear usuario".
		3. Termina el caso de uso.
Sección 1: Bloquear usuario		
Flujo básico Bloquear usuario		
	Actor	Sistema
	1. Introduce el nombre del usuario que desea bloquear y selecciona la opción bloquear.	2. Valida los datos y bloquea el usuario.
		3. El caso de uso termina.

Flujos alternos		
1. a Existen datos incorrectos (usuario)		
	Actor	Sistema
		1. Muestra el mensaje: "El usuario no existe".
		2. Vuelve al paso 1 del flujo básico.
Sección 2: "Desbloquear usuario"		
Flujo básico Desbloquear usuario		
	Actor	Sistema
	1. Selecciona la opción desbloquear del usuario que desea bloquear.	2. Desbloquea el usuario bloqueado.
		3. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	
Asuntos pendientes		

Tabla 11 CU Filtrar Mensajes

Objetivo	Mostrar solo los mensajes que coincidan parcialmente con el nombre o la descripción especificada.
Actores	Usuario (Inicia): Filtra la lista de mensajes.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción "Filtrar", luego de haber entrado la opción

	para filtrado, se muestra un listado con todos los mensajes que coinciden parcialmente con el nombre especificado.	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. El usuario debe tener permiso de uso de mensajería privada.	
Postcondiciones	Se filtró la lista de suscripciones.	
Flujo de eventos		
Flujo básico Filtrar mensajes		
	Actor	Sistema
	1. Selecciona la opción "Filtrar".	2. Muestra una interfaz para introducir los datos de filtrado por el asunto, participante o por etiqueta si está etiquetado y permite: <ul style="list-style-type: none"> • Filtrar. • Guardar filtro.
	3. Introduce los datos para el filtrado y selecciona la opción Filtrar.	4. Filtra los datos y muestra un listado con los mensajes correspondientes a los datos entrados.
		5. Termina el caso de uso.
Flujos alternos		
1. a No hay mensajes que coincidan con esos datos		

	Actor	Sistema
		1. Muestra el mensaje "No hay mensajes".
		2. Regresa al paso 3 del flujo básico.
1. b Guardar filtro		
	Actor	Sistema
	1. Selecciona la opción guardar filtro.	2. Guarda el filtro.
		3. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	
Asuntos pendientes		

Tabla 12 CU Configurar plantillas de mensaje

Objetivo	Configurar las plantillas para los distintos tipos de mensajes.
Actores	Administrador
Resumen	El caso de uso inicia cuando el administrador decide configurar las plantillas de los mensajes para que sean

	enviados con la estructura que el defina.	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones	Debe haberse generado el escritorio de trabajo del administrador autenticado.	
Postcondiciones	El usuario configurará las plantillas de mensajes.	
Flujo de eventos		
Flujo básico Configurar plantillas de mensaje		
	Actor	Sistema
	1. Selecciona la opción "Plantillas de mensaje".	2. Muestra una interfaz todas las plantillas de mensaje que pueden ser configuradas.
	3. Selecciona la plantilla deseada.	4. Muestra una interfaz para introducir los campos: <ul style="list-style-type: none"> • Asunto • Encabezado • Contenido • Pie de página
	5. Introduce los campos deseados y selecciona la opción Guardar.	6. Guarda la configuración.
		7. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	

Asuntos pendientes

Tabla 13 CU Crear suscripciones

Objetivo	Crear una suscripción.	
Actores	Usuario.	
Resumen	El caso de uso inicia cuando el usuario decide crear una suscripción, finalizando así el caso de uso.	
Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado.	
Postcondiciones	Se creó una suscripción.	
Flujo de eventos		
Flujo básico Crear suscripciones a contenidos		
	Actor	Sistema
	1. Selecciona la opción: “Crear suscripción”.	2. Muestra una interfaz con los diferentes tipos de suscripciones.
	3. Selecciona el tipo de suscripción.	4. Si selecciona tipo de suscripción a autor muestra una interfaz para introducir el autor y permite: <ul style="list-style-type: none">• Crear suscripción.• Cancelar.

		<p>Si selecciona tipo de suscripción a contenido muestra una interfaz para seleccionar el tipo de contenido y permite:</p> <ul style="list-style-type: none"> • Crear suscripción. • Cancelar <p>Si selecciona tipo de suscripción a categoría muestra una interfaz para introducir la categoría y permite:</p> <ul style="list-style-type: none"> • Crear suscripción. • Cancelar.
	5. Introduce los datos y selecciona la opción Crear suscripción.	6. Crea la suscripción.
		7. Termina el caso de uso.
Flujos alternos		
1. a Campos vacíos		
	Actor	Sistema
		<p>1. Si seleccionó suscripción de tipo autor y dejó el campo vacío:</p> <ul style="list-style-type: none"> • Muestra un mensaje: " El campo Usuario es obligatorio". • Si seleccionó suscripción a contenido y no selecciona un tipo de contenido muestra un mensaje: " El campo Tipo de contenido es obligatorio". • Si seleccionó suscripción a categoría y no introduce la categoría muestra un mensaje: "El campo Categoría es obligatorio".
		2. Termina el caso de uso.

Flujos alternos		
1. b Cancelar la operación		
	Actor	Sistema
	1. Selecciona la opción Cancelar.	2. Cancela la operación.
		3. Termina el caso de uso.

Tabla 14 CU Gestionar suscripciones a autor

Objetivo	Ver, editar y eliminar una suscripción de autor.
Actores	Usuario.
Resumen	El caso de uso inicia cuando el usuario decide ver, editar y eliminar una suscripción a autor, finalizando así el caso de uso.
Complejidad	Media.
Prioridad	Crítico.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado.
Postcondiciones	Se vio, editó y eliminó una suscripción a autor.
Flujo de eventos	

Flujo básico Gestionar suscripción a autor

	Actor	Sistema
	1. Selecciona la opción "Suscripciones".	2. Muestra una interfaz con un listado de las suscripciones y por cada una las opciones de: <ul style="list-style-type: none">• Ver Suscripciones (Ver Sección 1).• Editar Suscripciones (Ver Sección 2).• Eliminar Suscripciones (Ver Sección 3).
		3. Termina el caso de uso.

Sección 1 Ver suscripciones a autor

Flujo básico Ver suscripciones a autor

	Actor	Sistema
	1. Selecciona una suscripción de autor.	2. Muestra el contenido de la suscripción.
		3. Termina el caso de uso.

Sección 2 Modificar suscripciones a autor

Flujo básico Modificar suscripciones a autor

	Actor	Sistema
	1. Selecciona la opción: "Editar"	2. Muestra una interfaz con el campo para introducir el nombre del autor y permite: <ul style="list-style-type: none">• Salvar suscripción.• Eliminar. <i>"Ver sección 3 Eliminar suscripciones a autor"</i>

		<ul style="list-style-type: none"> • Cancelar.
	3. Introduce el nombre del autor y selecciona la opción Salvar suscripción.	4. Guarda los cambios realizados.
		5. Termina el caso de uso

Flujos alternos

2. a Cancelar la operación

	Actor	Sistema
	1. Selecciona la opción Cancelar.	2. Cancela la operación.
		3. Termina el caso de uso.

Sección 3 Eliminar suscripciones a autor

Flujo básico Eliminar suscripciones a autor

	Actor	Sistema
	1. Selecciona la opción: "Eliminar".	2. Muestra una interfaz para eliminar la suscripción y permite: <ul style="list-style-type: none"> • Eliminar suscripción. • Cancelar.
	3. Selecciona la opción Eliminar.	4. Elimina la suscripción.
		5. Termina el caso de uso

Flujos alternos

3.a Cancelar la operación

	Actor	Sistema
	1. Selecciona la opción Cancelar.	2. Cancela la operación.
		3. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	
Asuntos pendientes		

Tabla 15 CU Gestionar suscripciones a categoría

Objetivo	Ver, editar y eliminar una suscripción a categoría.
Actores	Usuario.
Resumen	El caso de uso inicia cuando el usuario decide ver, editar y eliminar una suscripción a categoría, finalizando así el caso de uso.
Complejidad	Media.
Prioridad	Crítico.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado.

Postcondiciones	Se vio, editó y eliminó una suscripción a categoría.	
Flujo de eventos		
Flujo básico Gestionar suscripción a categoría		
	Actor	Sistema
	1. Selecciona la opción "Suscripciones".	2. Muestra una interfaz con un listado de las suscripciones y por cada una las opciones de: <ul style="list-style-type: none"> • Ver Suscripciones (Ver Sección 1). • Editar Suscripciones (Ver Sección 2). • Eliminar Suscripciones (Ver Sección 3).
		3. Termina el caso de uso.
Sección 1 Ver suscripciones a categoría		
Flujo básico Ver suscripciones a categoría		
	Actor	Sistema
	Selecciona una suscripción a categoría.	Muestra el contenido de la suscripción.
		Termina el caso de uso.
Sección 2 Modificar suscripciones a categoría		
Flujo básico Modificar suscripciones a categoría		
	Actor	Sistema

	1. Selecciona la opción: “Editar suscripciones a categoría”.	2. Muestra una interfaz con el campo para introducir la categoría y permite: <ul style="list-style-type: none"> • Salvar suscripción. • Eliminar. “Ver sección 3 Eliminar suscripciones a autor” • Cancelar.
	3. Introduce la categoría y selecciona la opción Salvar suscripción.	4. Guarda los cambios realizados.
		5. Termina el caso de uso

Flujos alternos

2. a Cancelar la operación

	Actor	Sistema
	1. Selecciona la opción Cancelar.	2. Cancela la operación.
		3. Termina el caso de uso.

Sección 3 Eliminar suscripciones a categoría

Flujo básico Eliminar suscripciones a categoría

	Actor	Sistema
	1. Selecciona la opción: “Eliminar suscripciones a categoría”.	2. Muestra una interfaz para eliminar la suscripción y permite: <ul style="list-style-type: none"> • Eliminar suscripción. • Cancelar.
	3. Selecciona la opción Eliminar suscripción.	4. Elimina la suscripción.

		5. Termina el caso de uso
Flujos alternos		
3. a Cancelar la operación		
	Actor	Sistema
	1. Selecciona la opción Cancelar.	2. Cancela la operación.
		3. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	
Asuntos pendientes		

Tabla 16 CU Filtrar lista de suscripciones

Objetivo	Mostrar solo aquellas suscripciones que coincidan parcialmente con el nombre o la descripción especificada.
Actores	Usuario (Inicia): Filtra la lista de suscripciones.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción "Filtrar", luego de haber entrado la opción para filtrado, se muestra un listado con todas las suscripciones que coinciden parcialmente con el nombre o la descripción especificada.

Complejidad	Media	
Prioridad	Crítico	
Precondiciones	Se han listado las suscripciones creadas en el sistema.	
Postcondiciones	Se filtró la lista de suscripciones.	
Flujo de eventos		
Flujo básico Filtrar lista de suscripciones		
	Actor	Sistema
	1. Selecciona la opción Filtrar.	2. Muestra una interfaz para seleccionar el estado y el tipo de suscripción por el que se desea filtrar. Permite: <ul style="list-style-type: none"> • Filtrar.
	3. Selecciona el estado y el tipo de suscripción y la opción filtrar.	4. Filtra las suscripciones y muestra las suscripciones correspondientes con los datos. Permite: <ul style="list-style-type: none"> • Reiniciar.
		5. Termina el caso de uso.
Flujos alternos		
1. a No hay suscripciones que coincidan con esos datos		
	Actor	Sistema

		1. Muestra el mensaje: “No existe ninguna suscripción que coincida con los datos”.
		2. Termina el caso de uso.
1.b Reiniciar la operación		
	Actor	Sistema
	1. Selecciona la opción Reiniciar.	2. Muestra la lista de suscripciones.
		3. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	
Asuntos pendientes		

Tabla 17 CU Listar suscripciones

Objetivo	Ver las suscripciones existentes.
Actores	Usuario (Inicia): Lista las suscripciones existentes.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción “Suscripciones” y se muestra un listado con todas las suscripciones existentes.

Complejidad	Media	
Prioridad	Crítico	
Precondiciones	Ha sido creada alguna suscripción.	
Postcondiciones	Se mostraron las suscripciones.	
Flujo de eventos		
Flujo básico Listar suscripciones		
	Actor	Sistema
	1. Selecciona la opción "Suscripciones".	2. Muestra un listado con todas las suscripciones. Por cada una se muestra el id, tipo, estado de la descripción, intervalo, fecha de creado y operaciones
		3. Termina el caso de uso.
Flujos alternos		
1. No hay suscripciones disponibles		
	Actor	Sistema
		1. Muestra el mensaje: "No hay ninguna suscripción disponible".
		2. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU	

	Extendidos	
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	
Asuntos pendientes		

Tabla 18 CU Recomendar ítems de suscripción

Objetivo	Recomendar de ítems de suscripción.	
Actores	Usuario (Inicia).	
Resumen	El caso de uso se inicia cuando el actor accede a la página principal.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Ha sido creada alguna suscripción y el usuario este autenticado.	
Postcondiciones		
Flujo de eventos		
Flujo básico Listar suscripciones		
	Actor	Sistema
1	1. Accede la página principal.	2. Muestra una interfaz con un bloque de suscripciones recomendadas.
		3. Termina el caso de uso.

Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	
Asuntos pendientes		

Tabla 19 CU Procesar suscripciones

Objetivo	Procesar suscripciones.
Actores	Sistema.
Resumen	El caso de uso se inicia cuando el sistema invoca el caso de uso procesar suscripciones.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Hayan sido modificadas las suscripciones.
Postcondiciones	
Flujo de eventos	
Flujo básico Procesar suscripciones	

	Actor	Sistema
1. 1	1. Invoca procesar suscripciones.	2. Envía correo a los usuarios que tengan suscripciones y que las mismas hayan sido modificadas.
		3. Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	Rendimiento, Usabilidad, Apariencia o Interfaz externa.	
Asuntos pendientes		

Anexo 2 Diagrama de clase del análisis

Tabla 20 DCA Gestionar mensaje

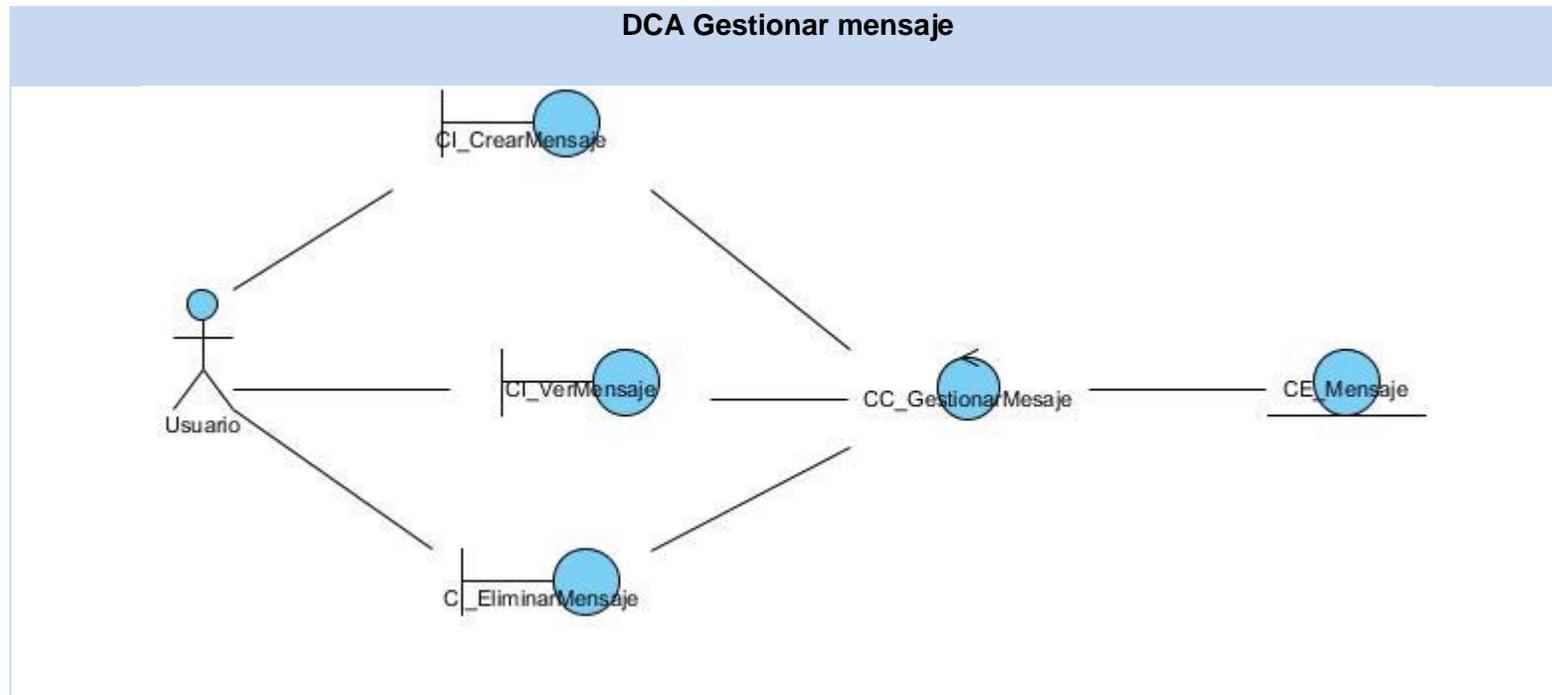


Tabla 21 DCA Responder mensaje

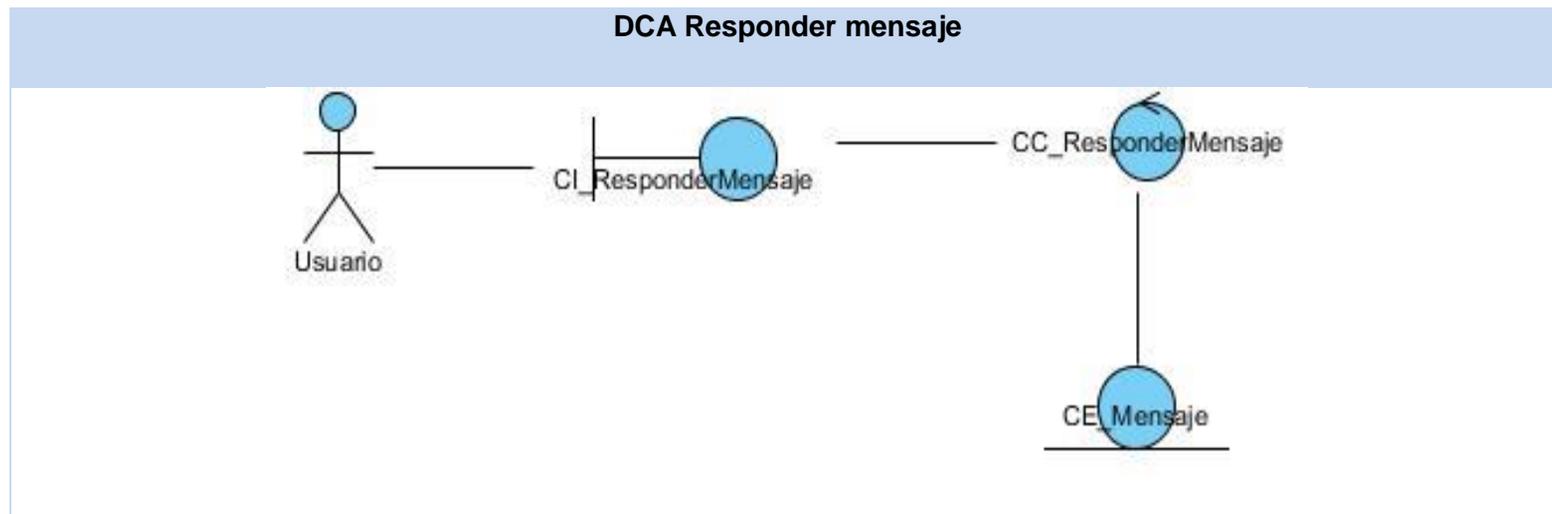


Tabla 22 DCA Responder mensaje

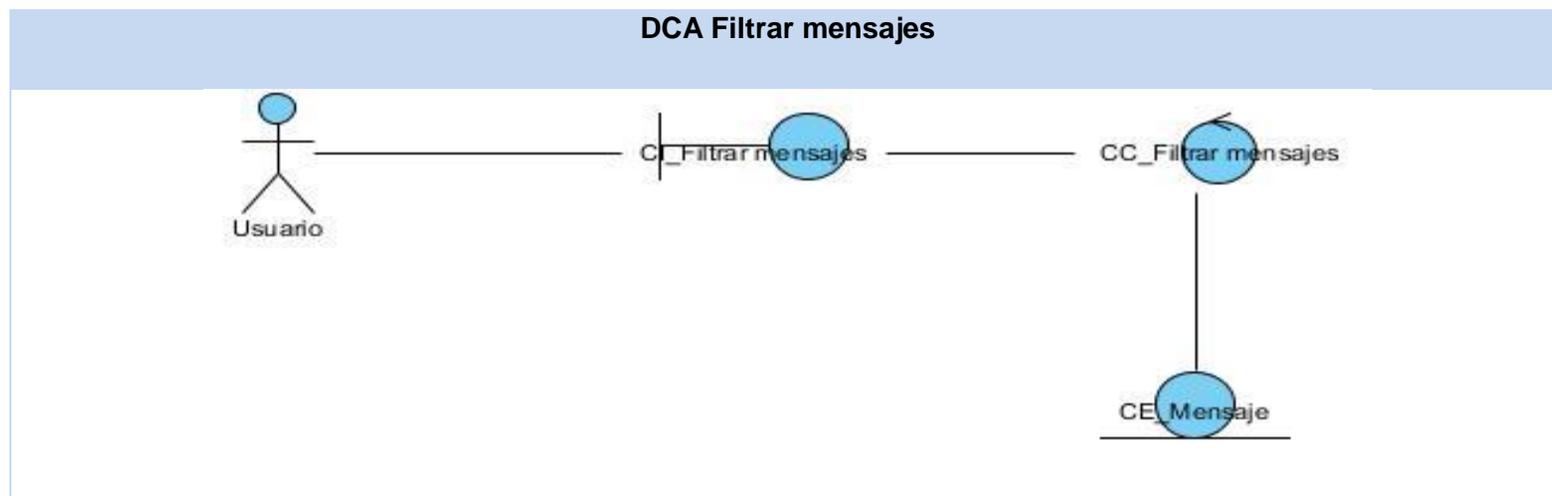


Tabla 23 DCA Bloquear/desbloquear usuario en la mensajería

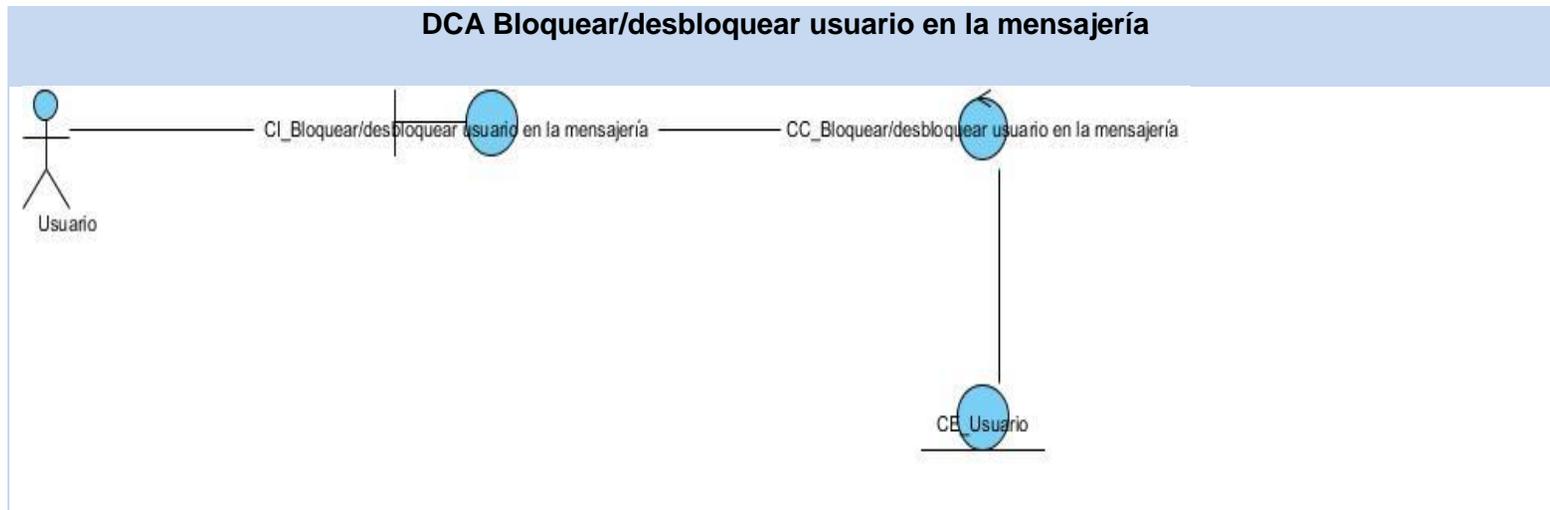


Tabla 24 DCA Configurar plantillas de mensaje

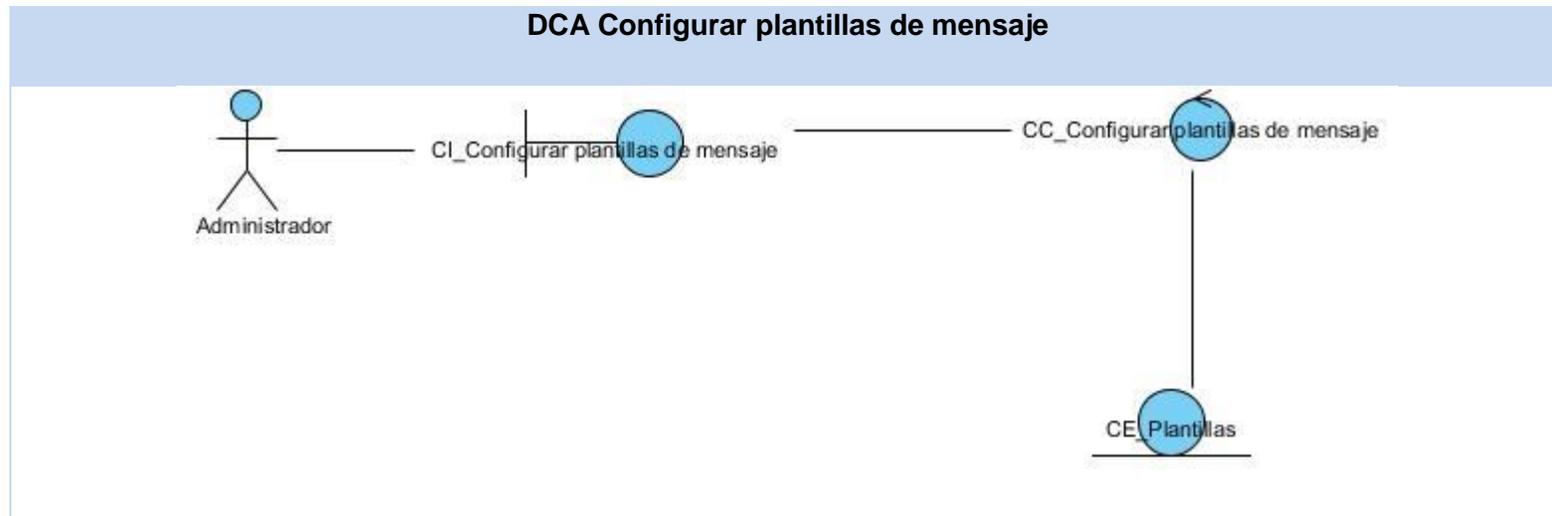


Tabla 25 DCA Crear suscripciones

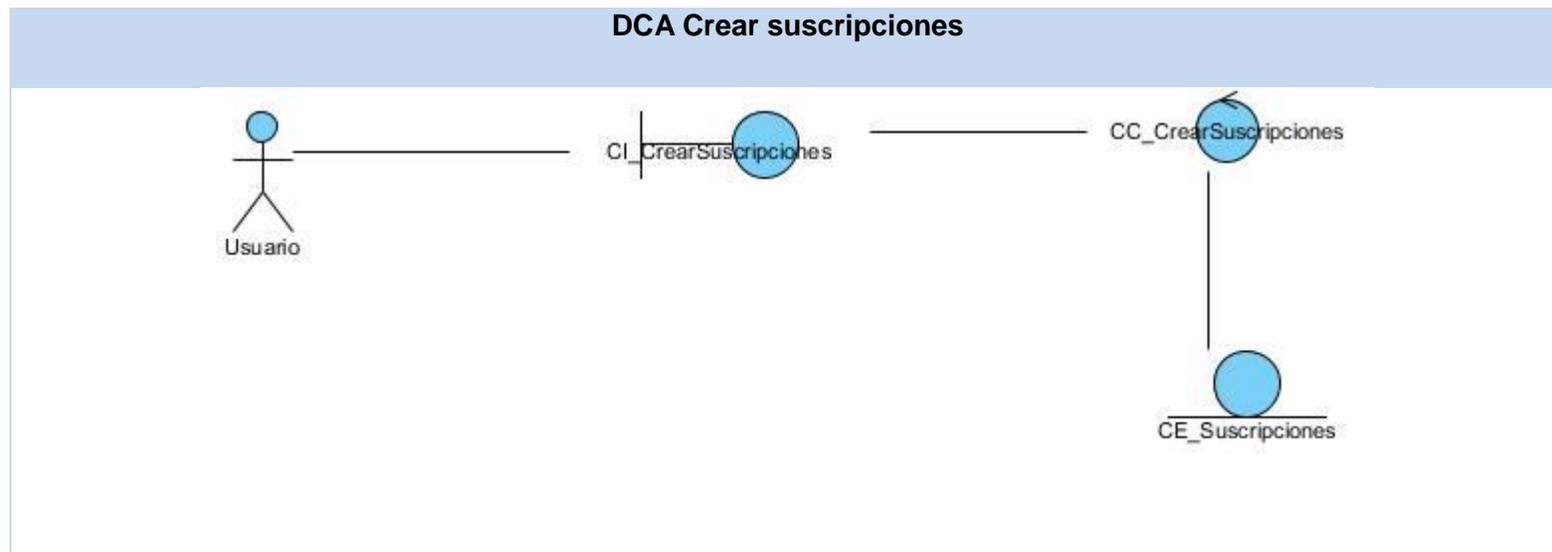


Tabla 26 DCA Gestionar suscripciones a autor

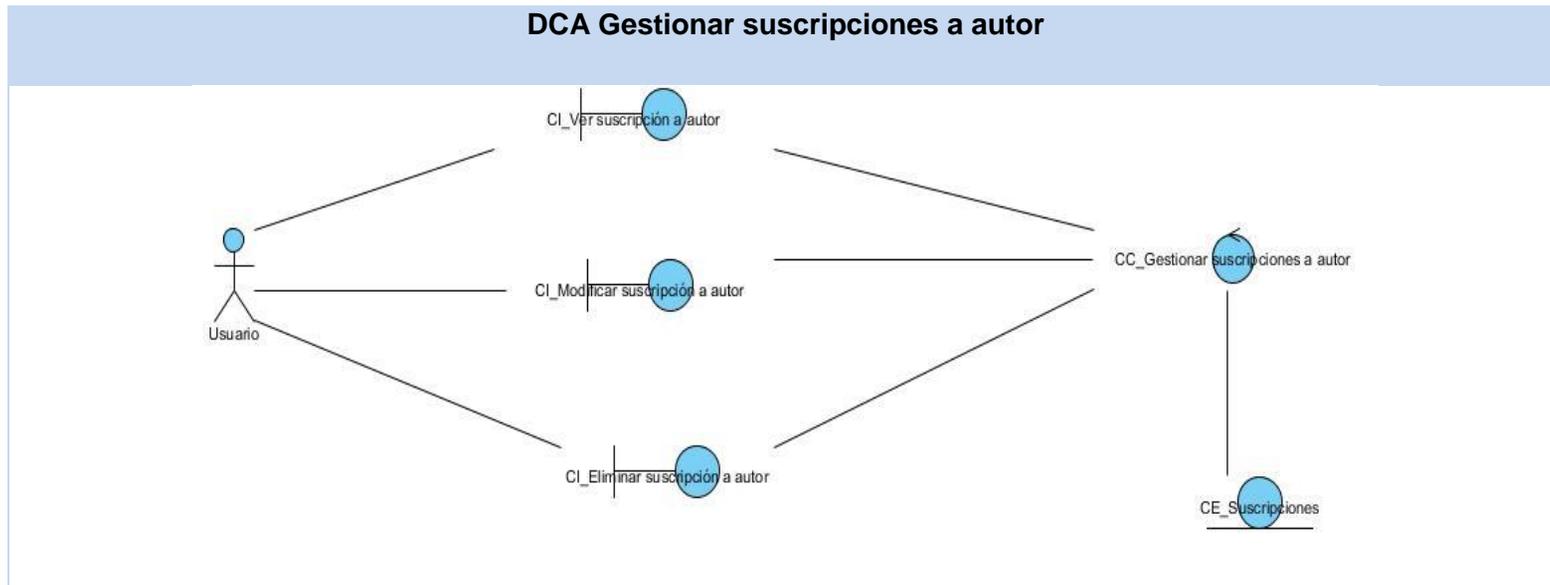


Tabla 27 DCA Gestionar suscripciones a categoría

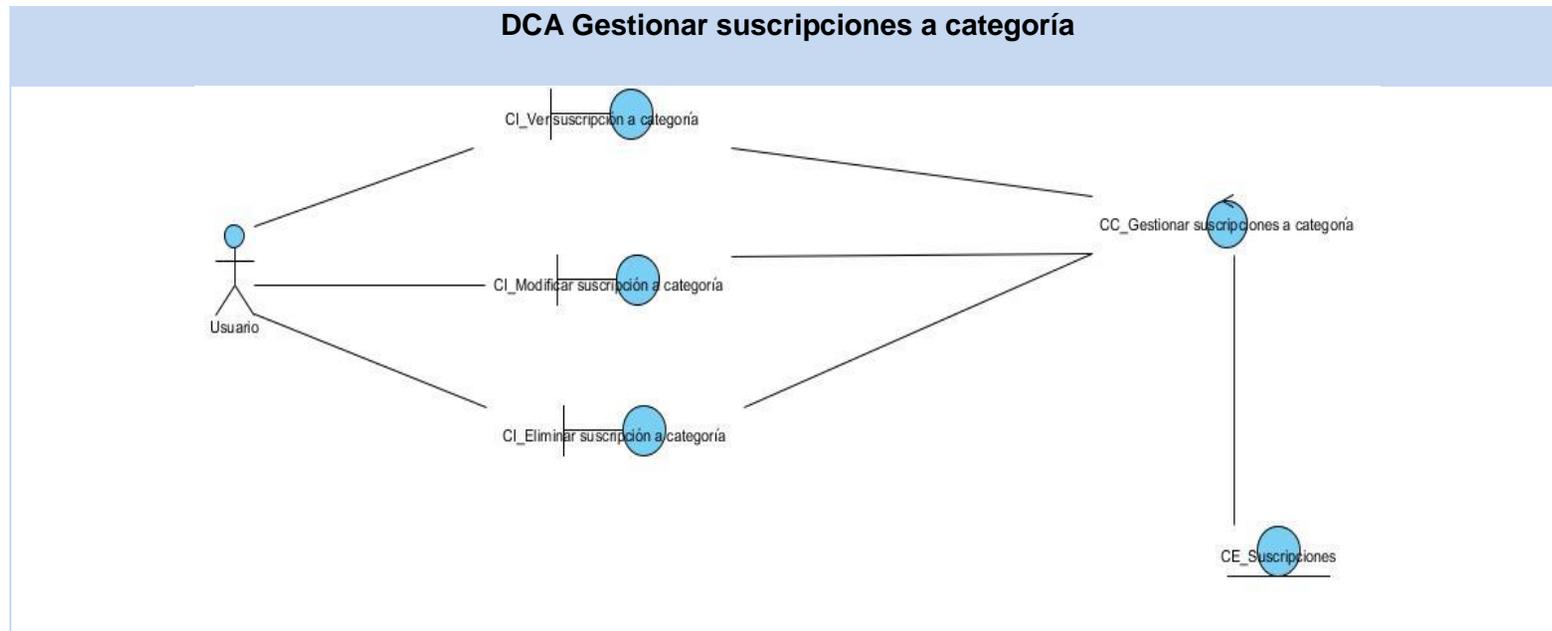


Tabla 28 DCA Filtrar lista de suscripciones



Tabla 29 DCA Listar suscripciones

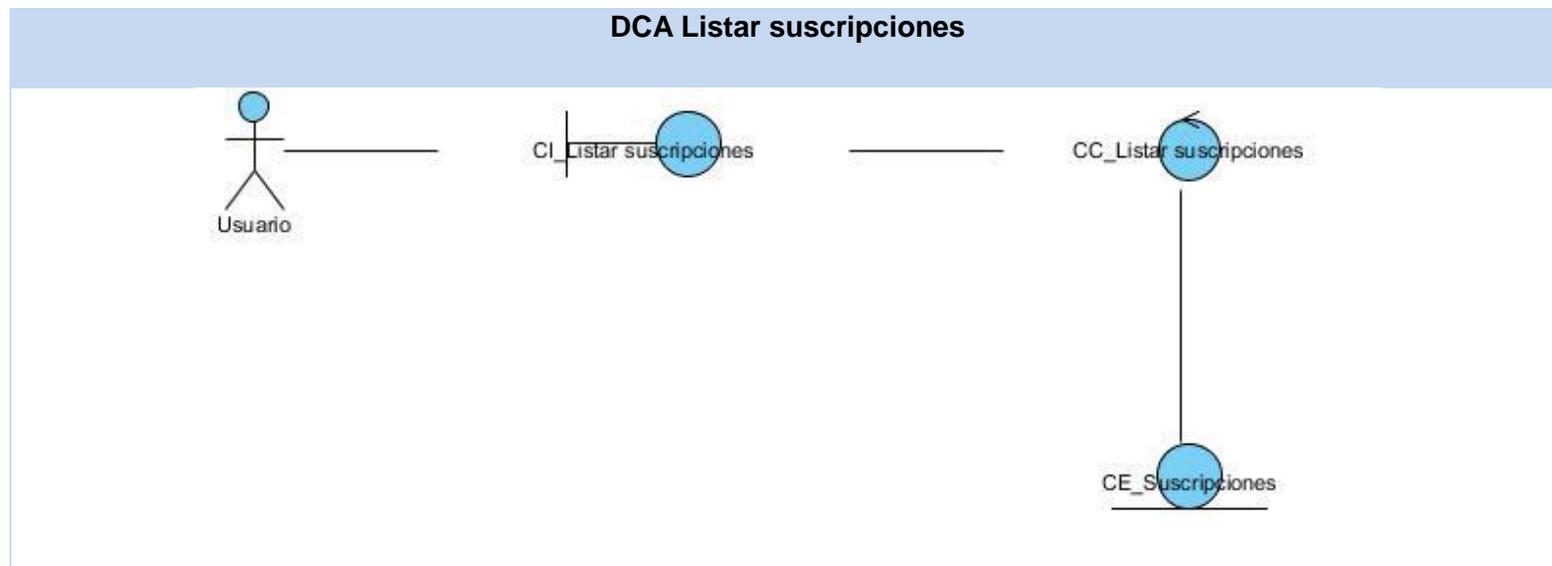
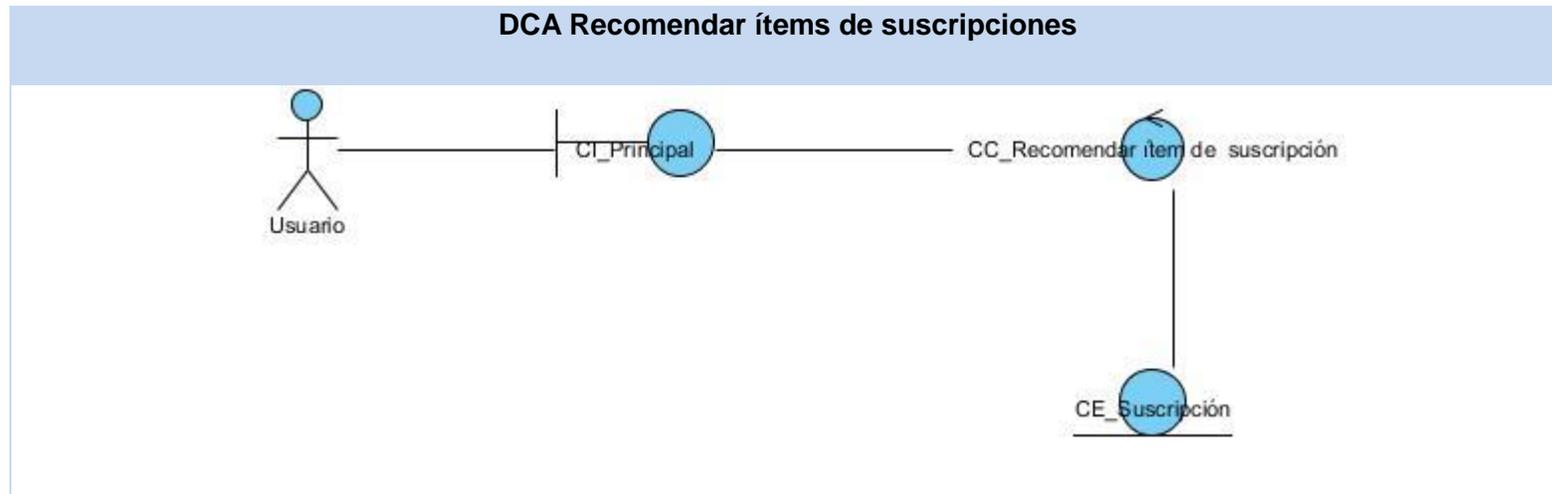
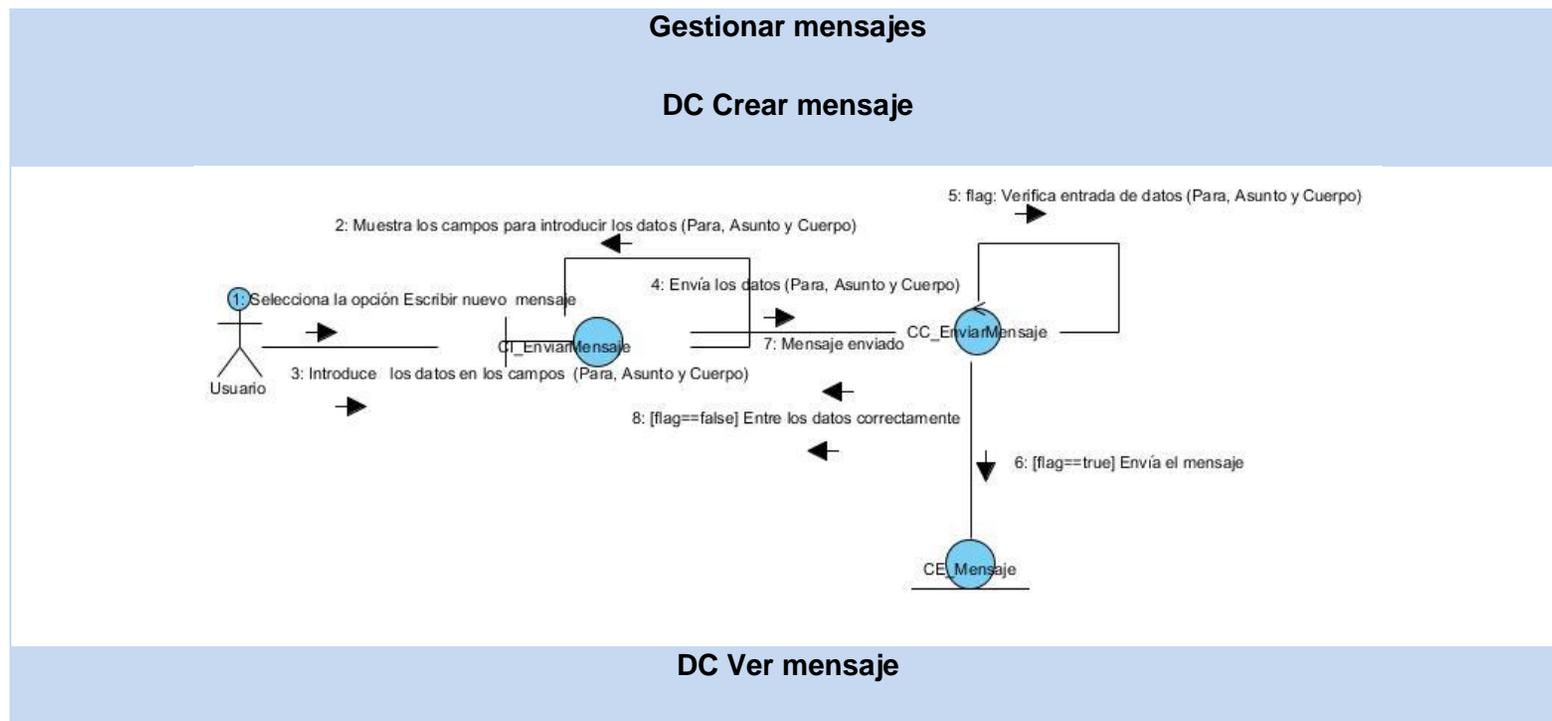


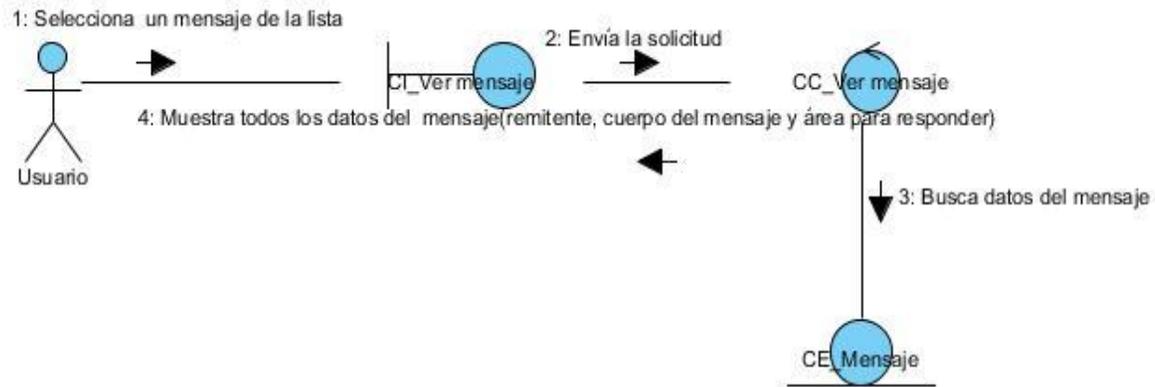
Tabla 30 DCA Recomendar ítems de suscripciones



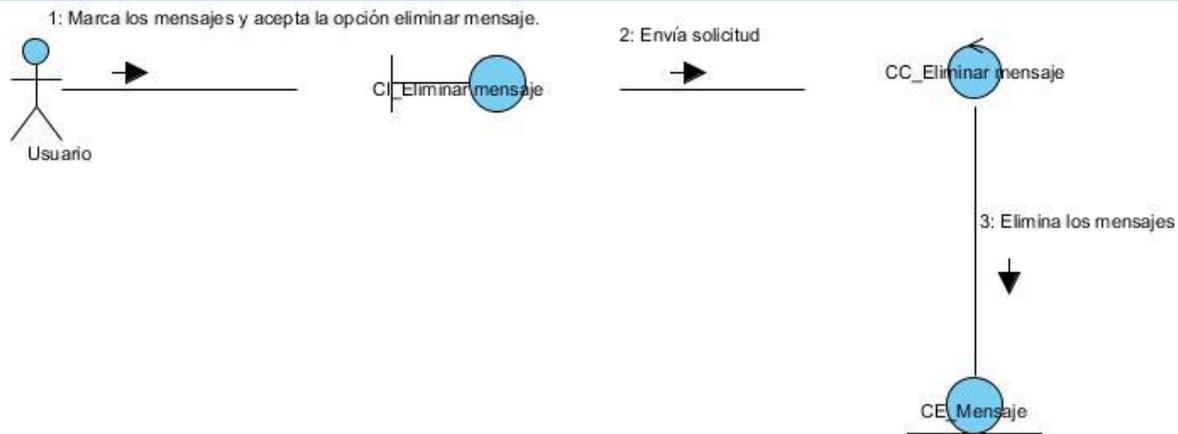
Anexo 3 Diagrama de colaboración

Tabla 31 DC Gestionar mensajes

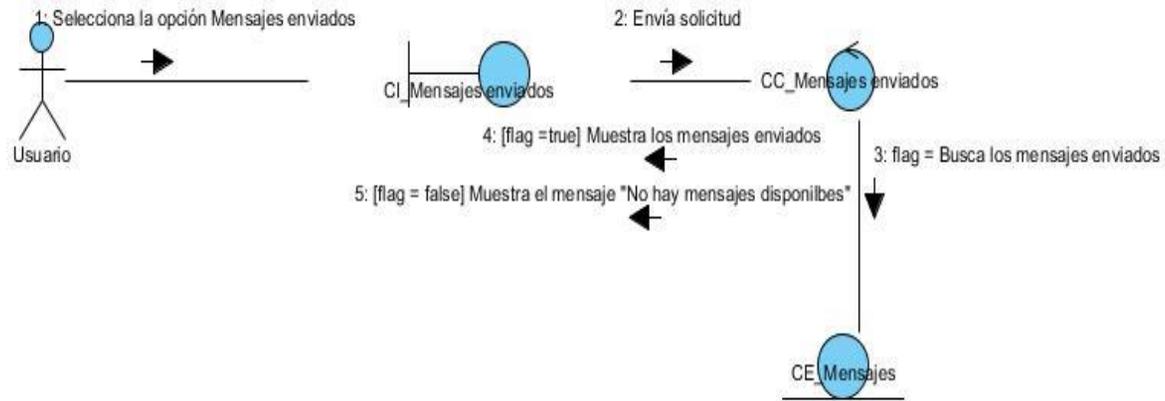




DC Eliminar mensaje



DC Mensajes enviados



DC Todos los mensajes

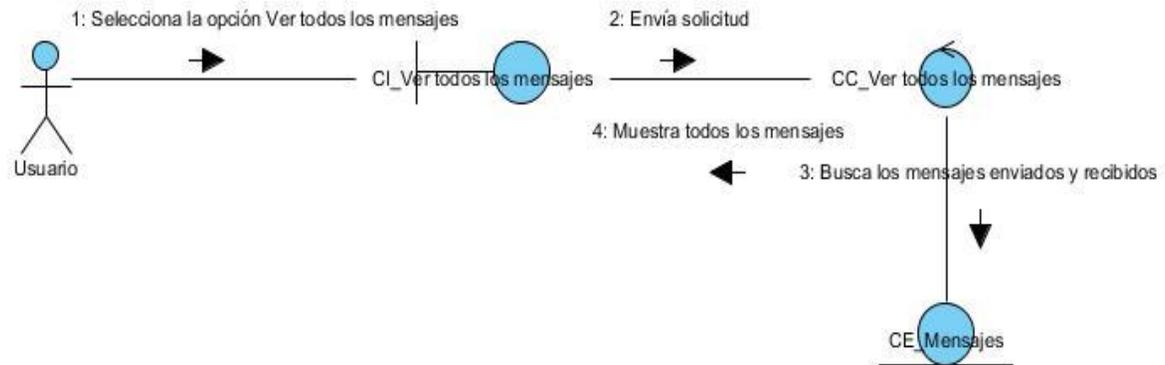


Tabla 32 DC Responder mensajes

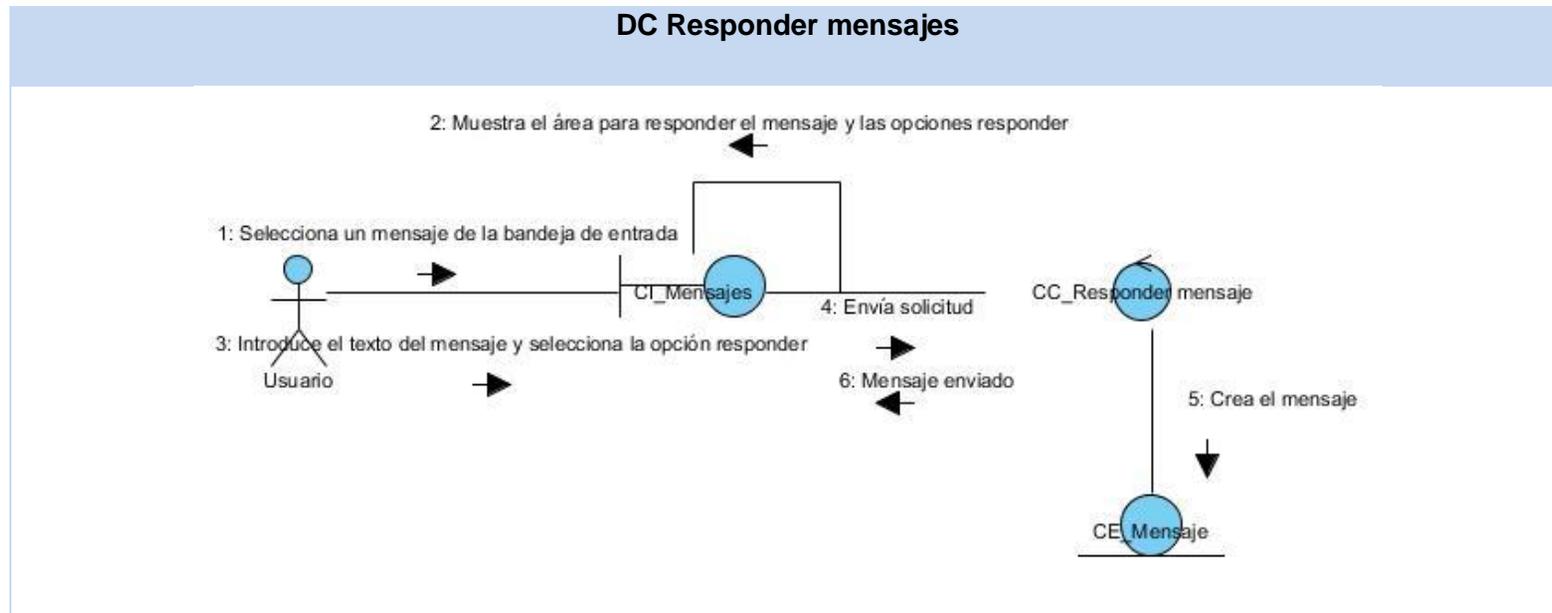
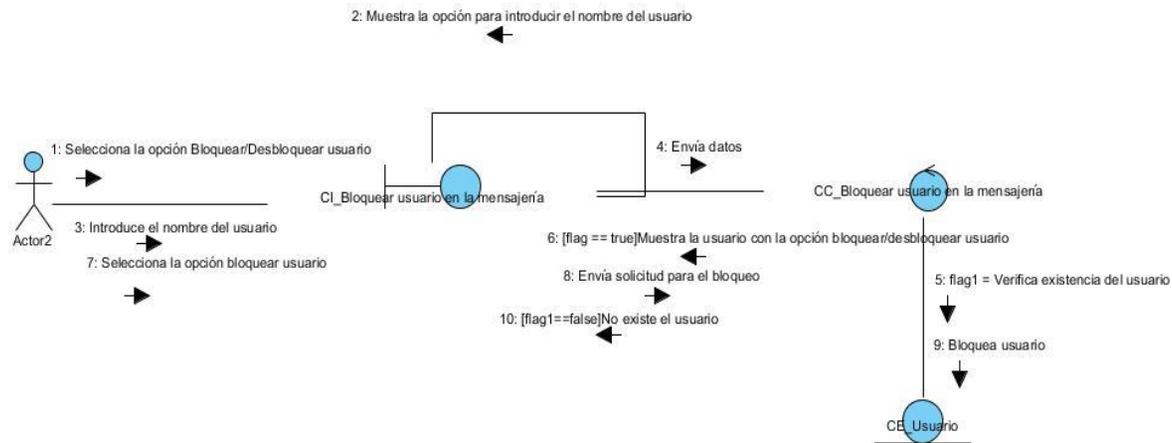


Tabla 33 DC Bloquear/desbloquear usuario en la mensajería

DC Bloquear/desbloquear usuario en la mensajería

DC Bloquear usuario en la mensajería



DC Desbloquear usuario en la mensajería

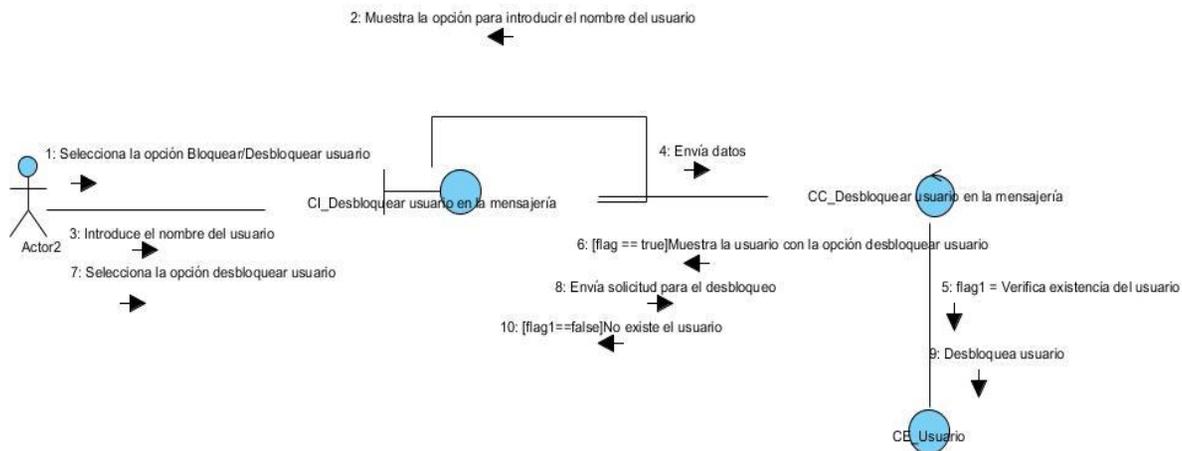


Tabla 34 DC Filtrar mensajes

DC Filtrar mensajes

1: Selecciona la opción Filtrar Mensajes e introduce el asunto, el usuario



Tabla 35 DC Configurar plantillas de mensaje

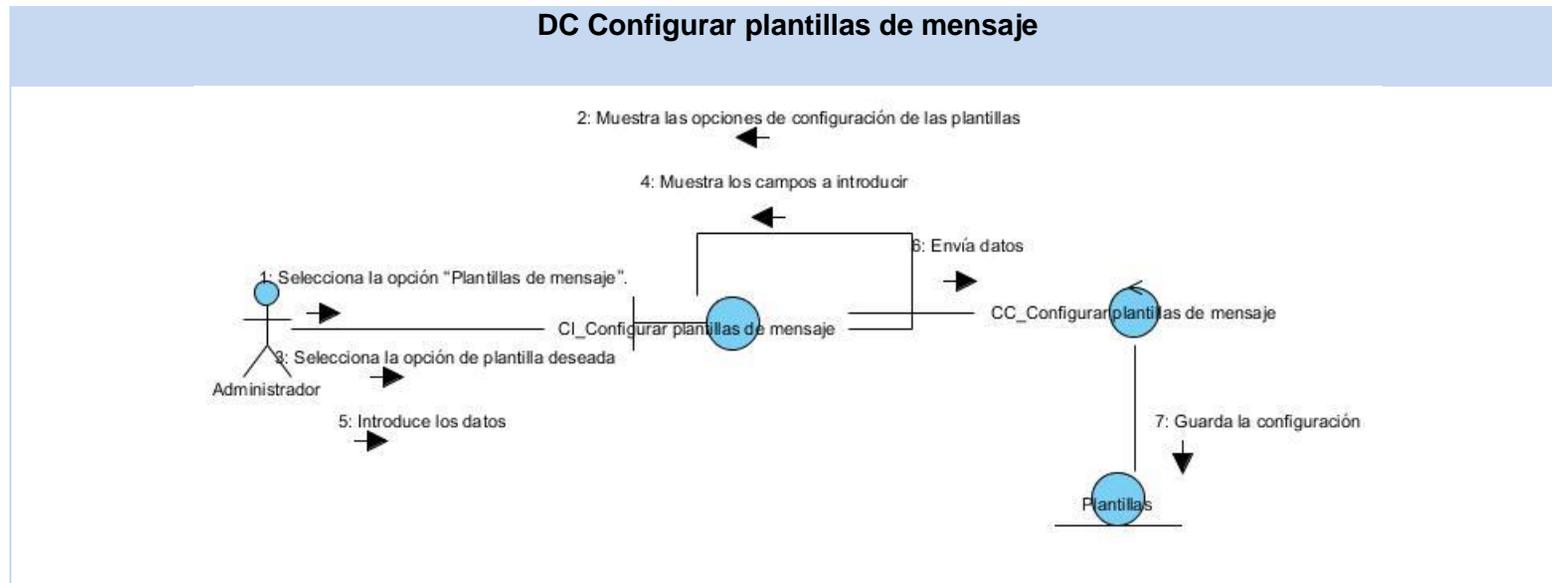


Tabla 36 DC Crear suscripciones

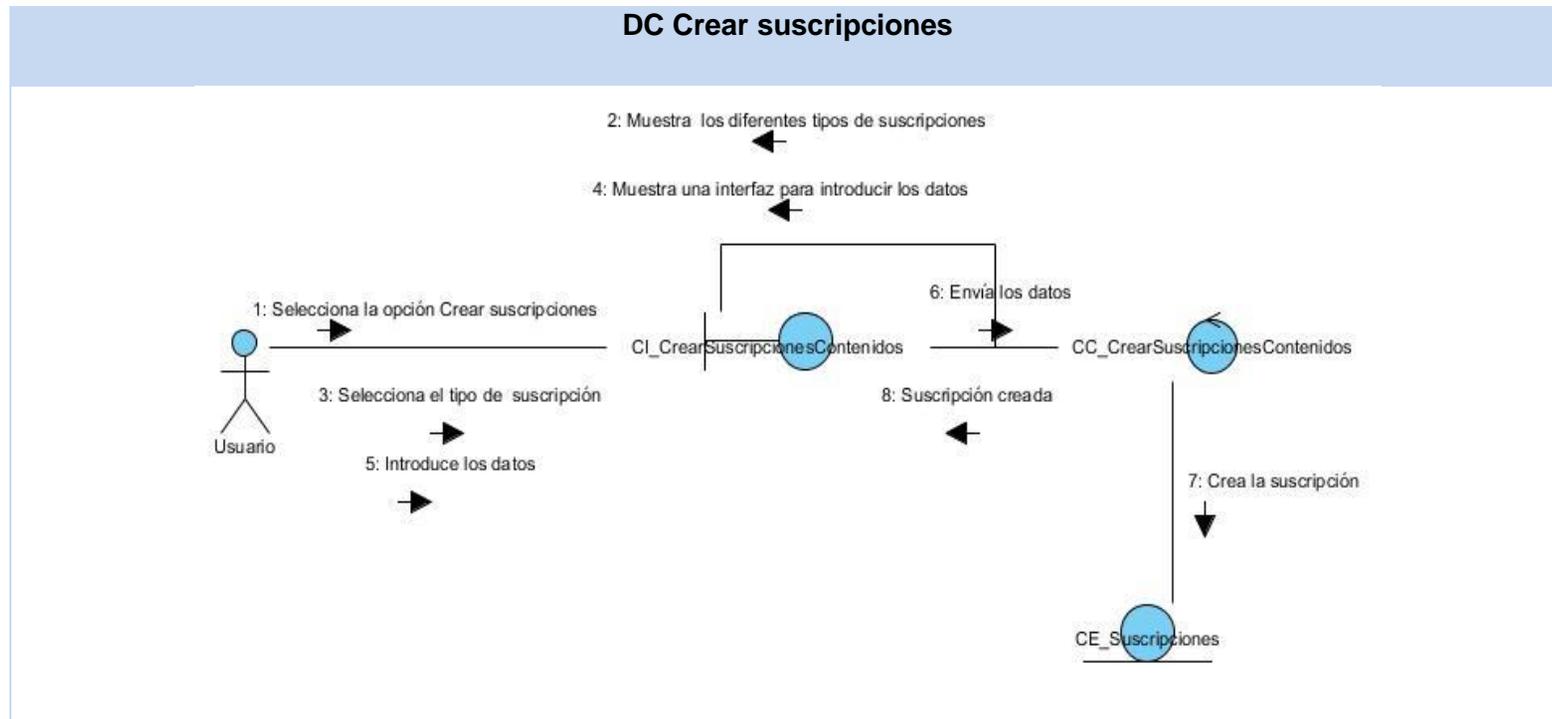
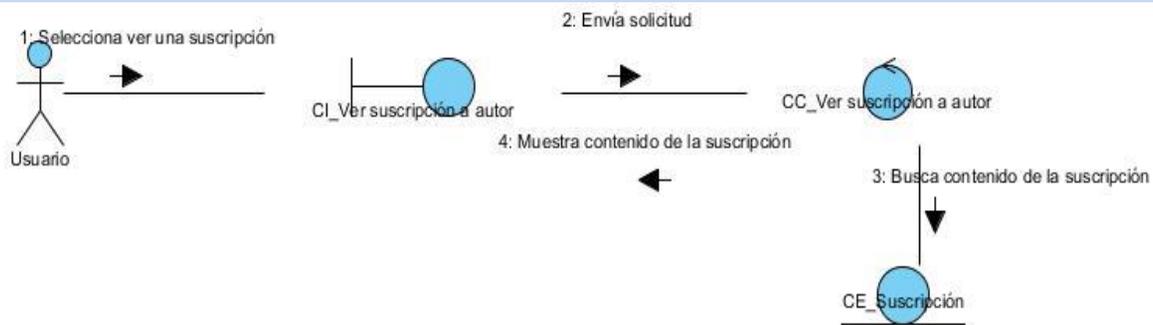


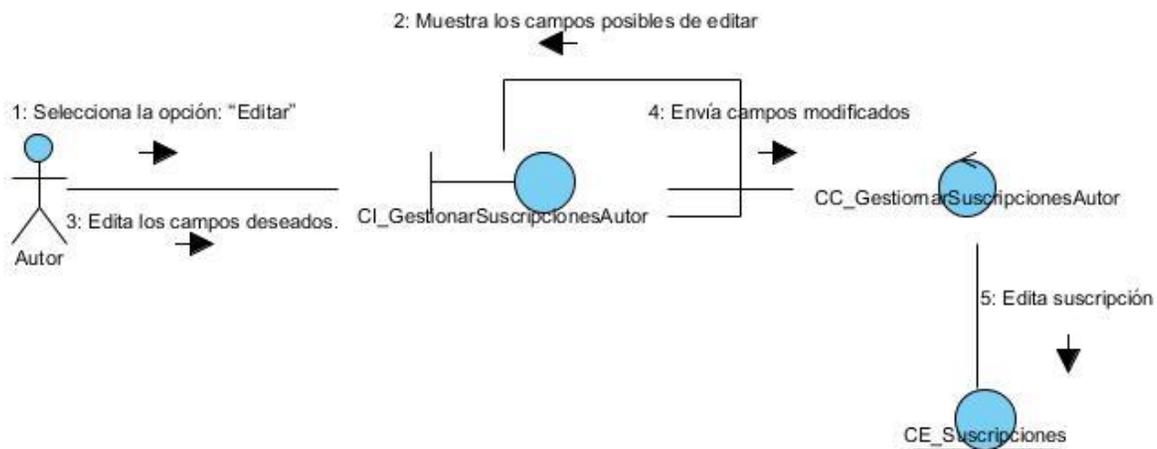
Tabla 37 DC Gestionar suscripciones a autor

DC Gestionar suscripciones a autor

DC Ver suscripción



DC Modificar suscripciones



DC Eliminar descripciones

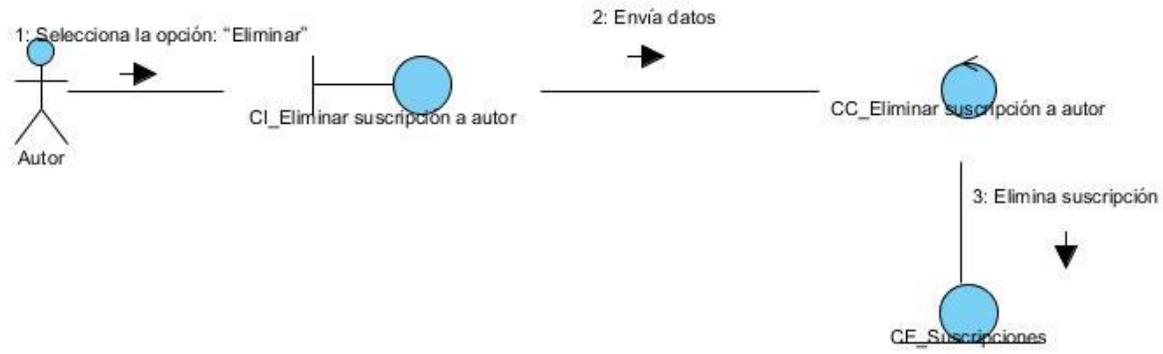
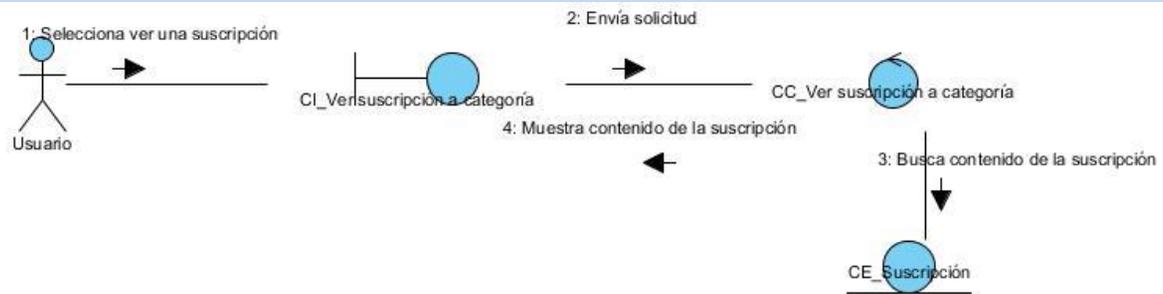


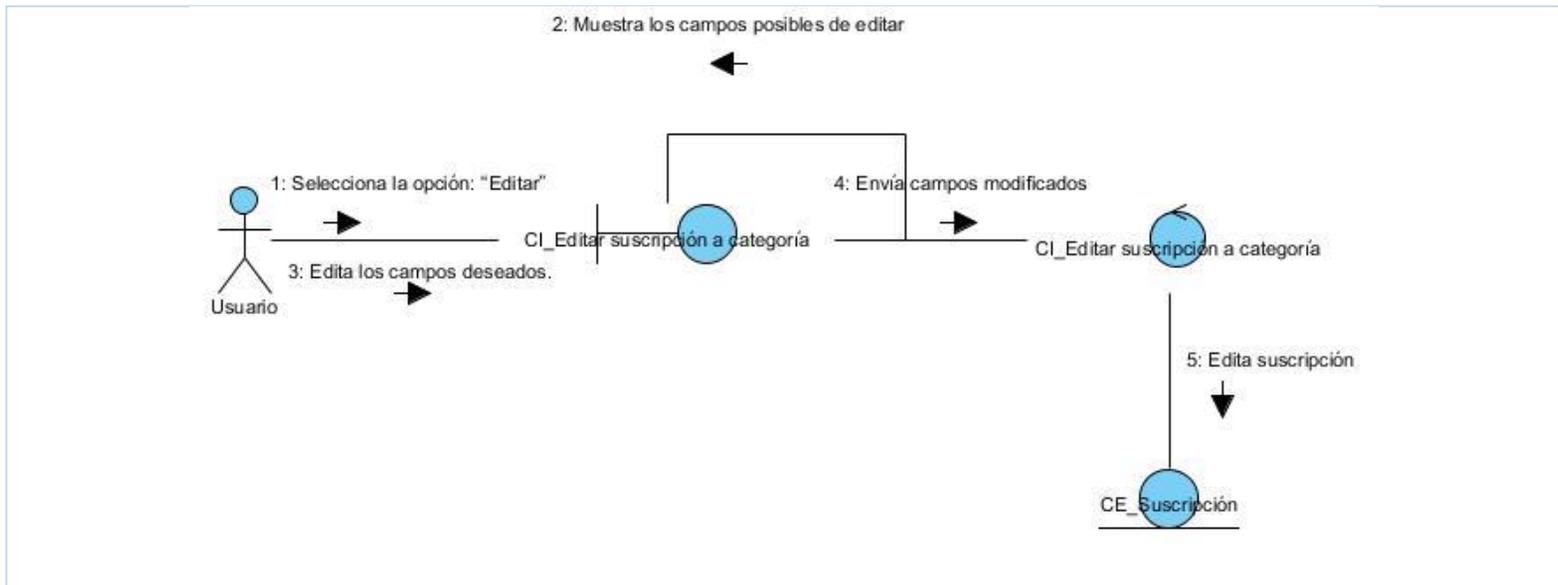
Tabla 38 DC Gestionar suscripciones a categoría

DC Gestionar suscripciones a categoría

DC Ver suscripción



DC Modificar suscripciones



DC Eliminar descripciones

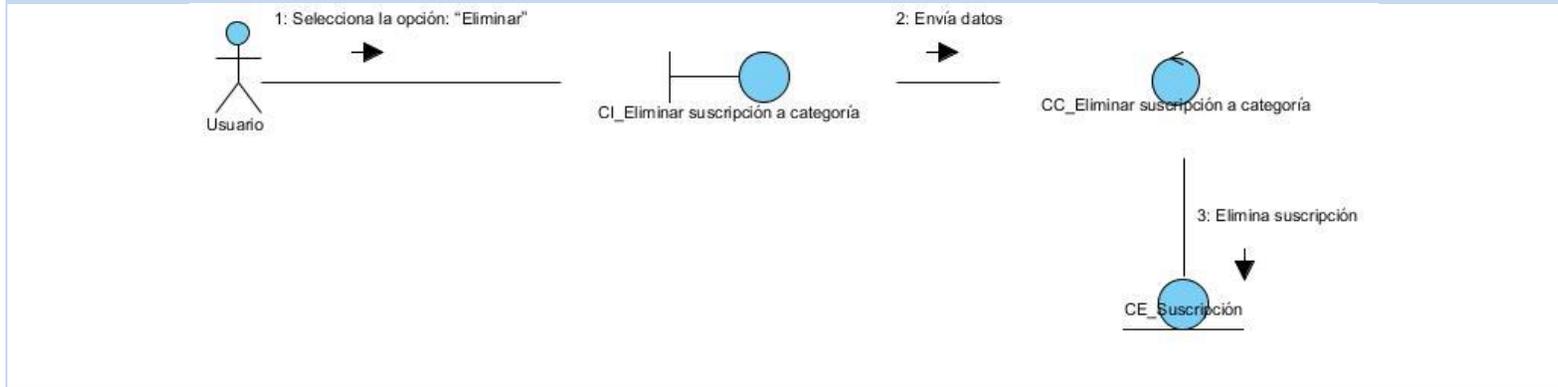


Tabla 39 DC Filtrar lista de suscripciones



Tabla 40 DC Listar suscripciones

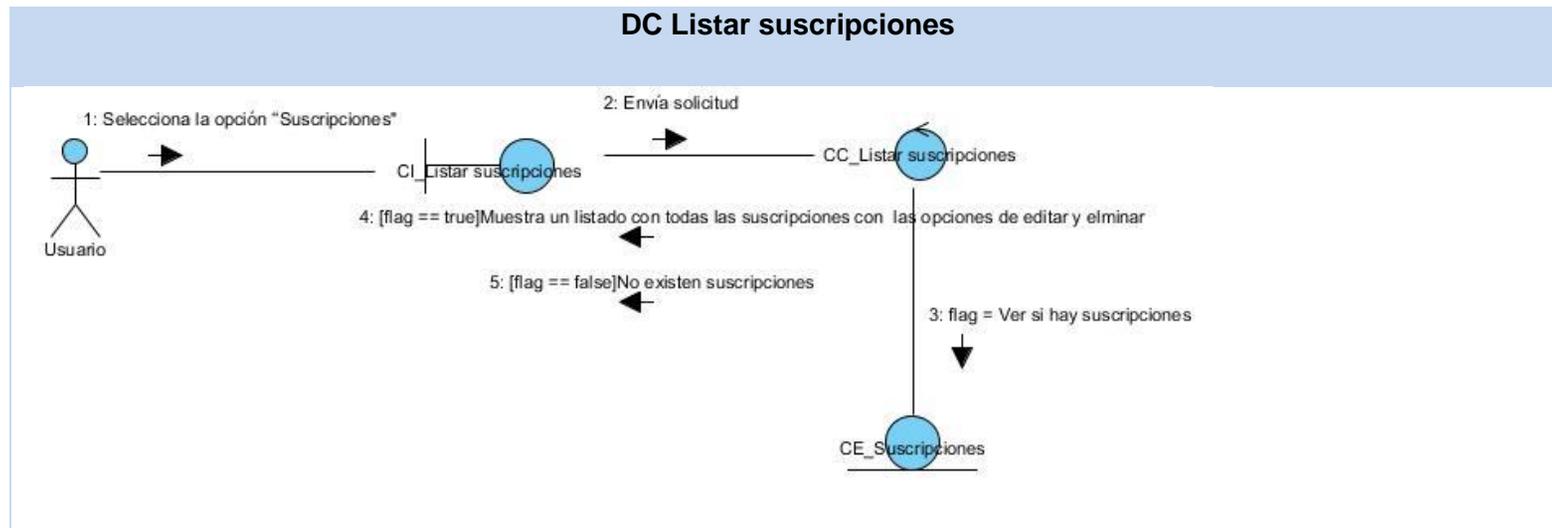
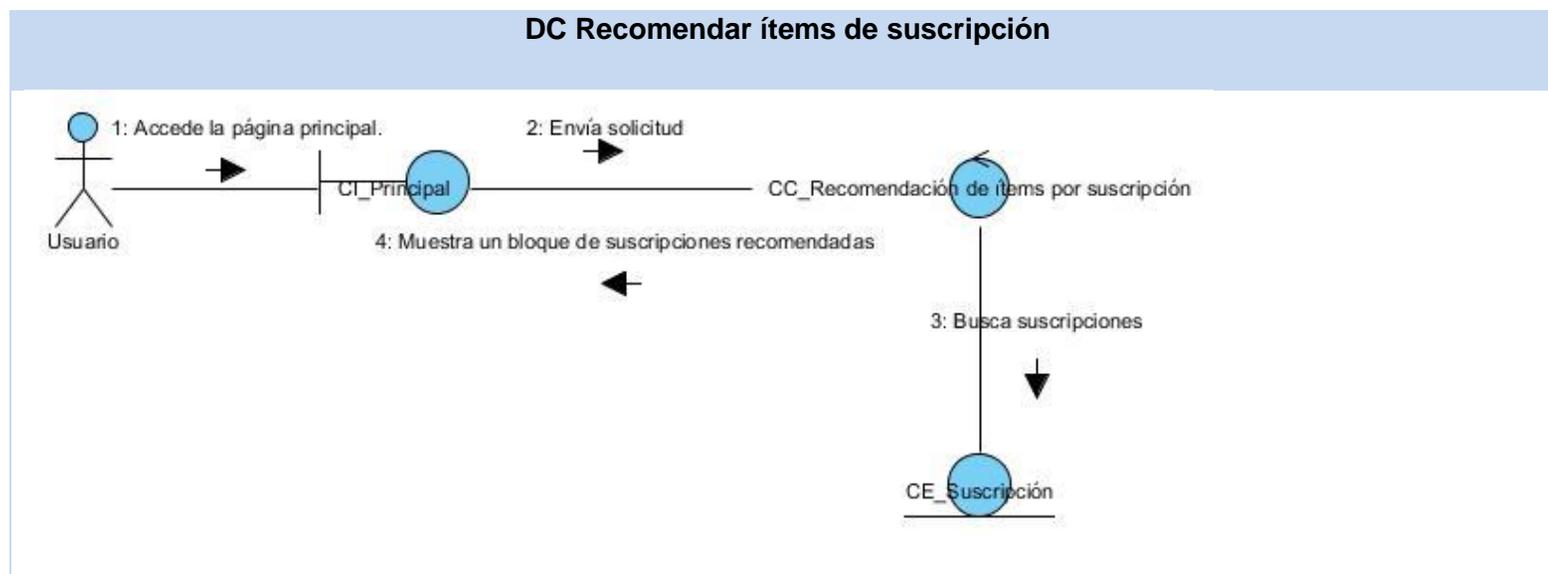


Tabla 41 DC Recomendar ítems de suscripción



Anexo 4 Diagrama de clases del diseño

Tabla 42 DCD Configurar plantillas de mensaje

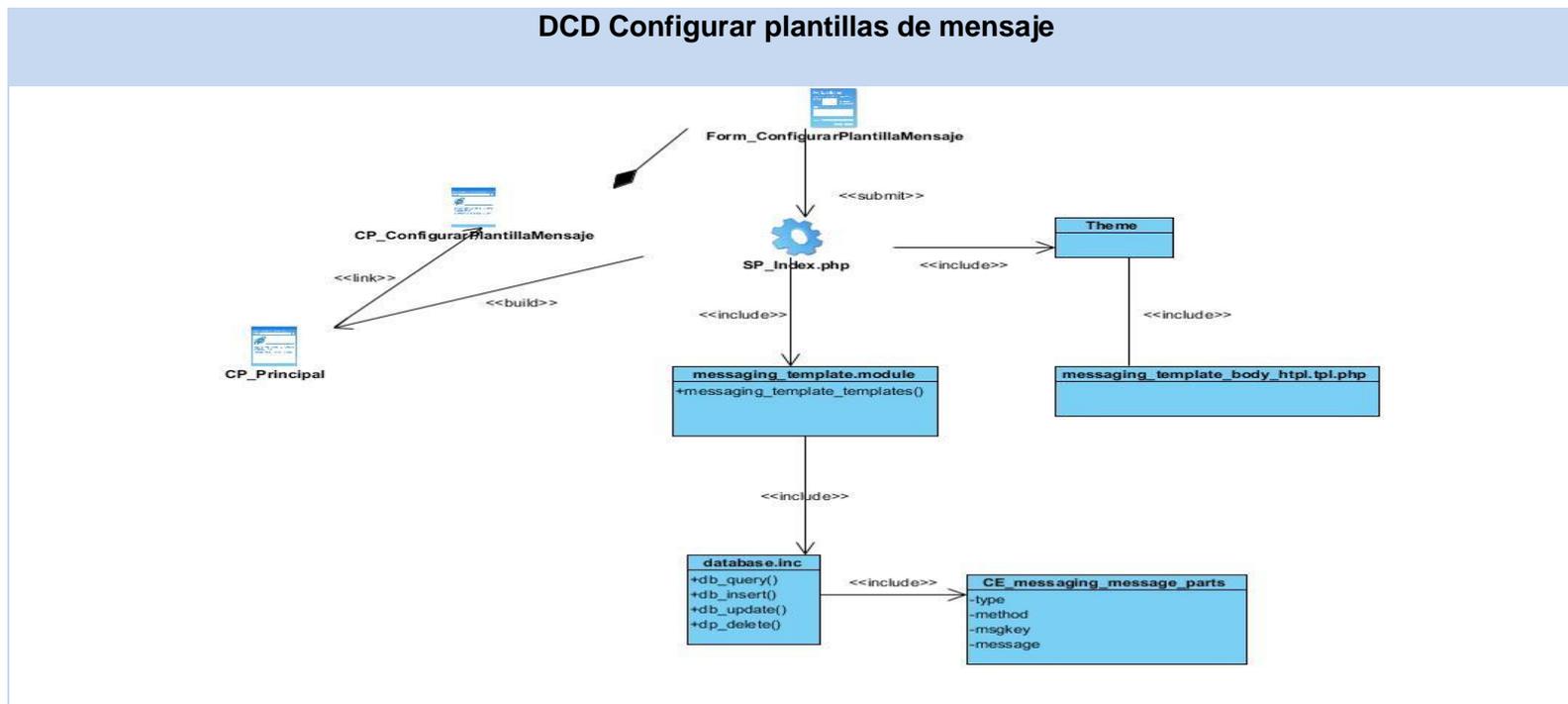


Tabla 43 DCD Crear suscripciones

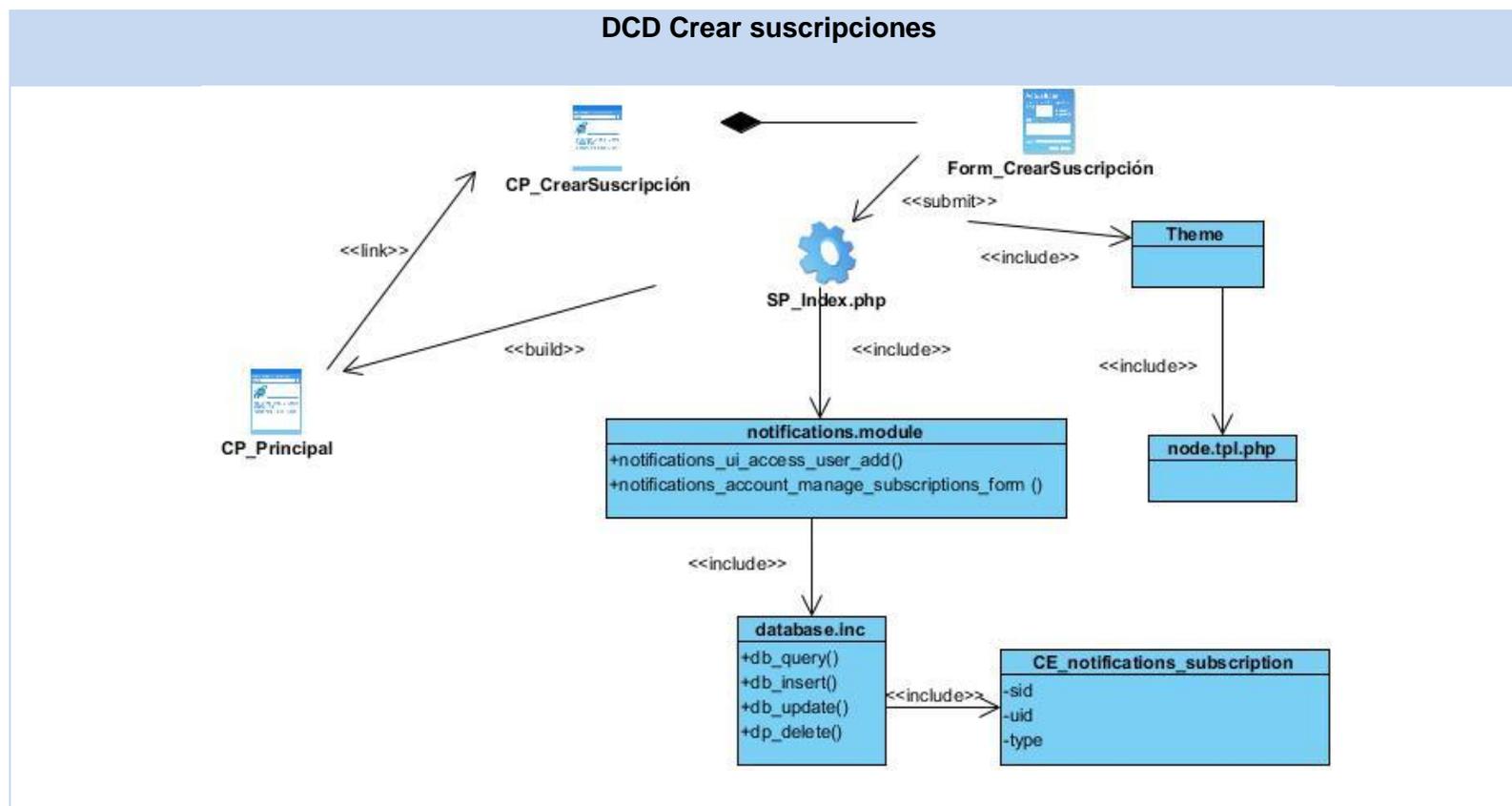


Tabla 44 DCD Gestionar suscripciones a autor

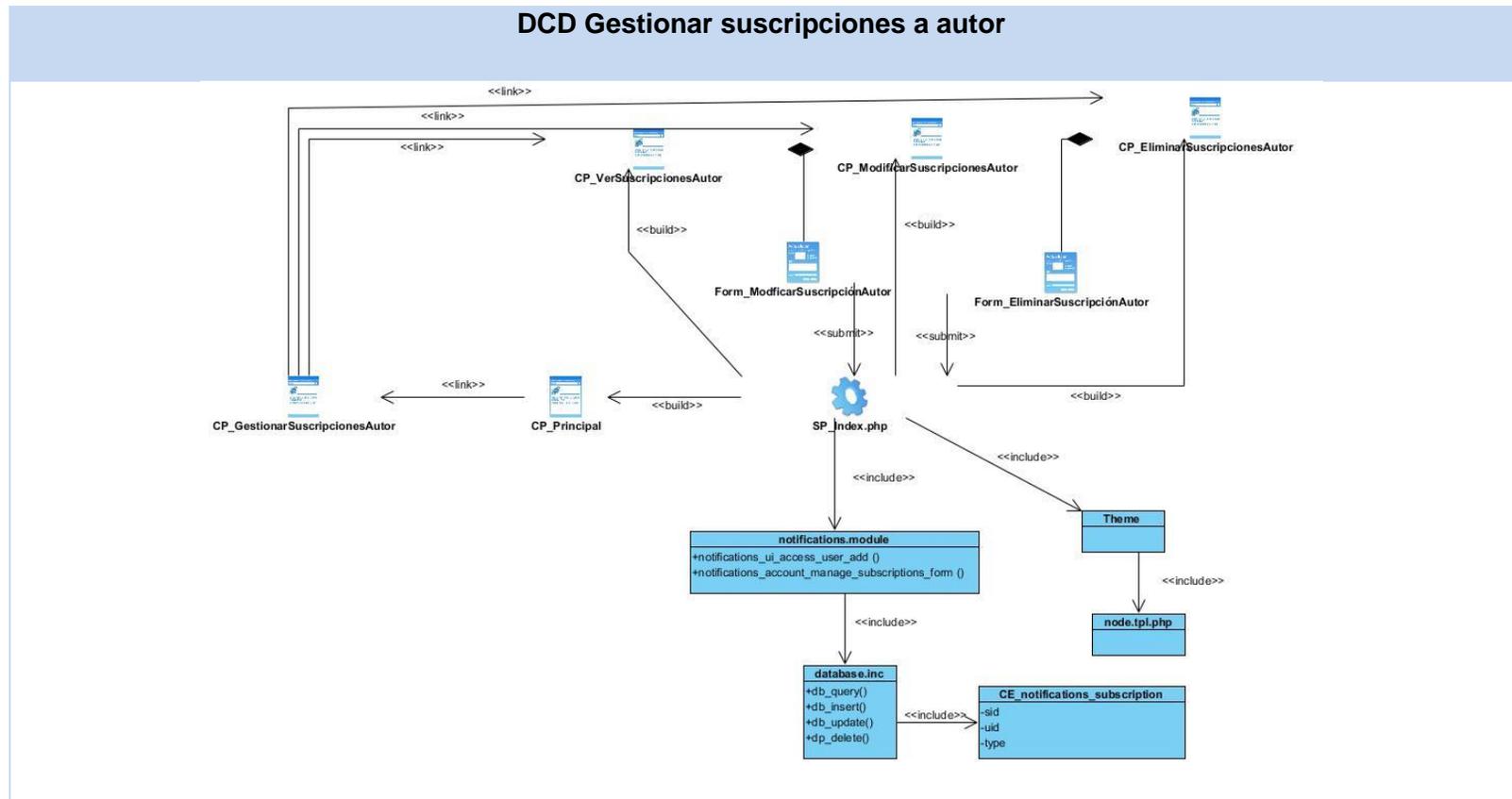


Tabla 45 DCD Gestionar suscripciones a categoría

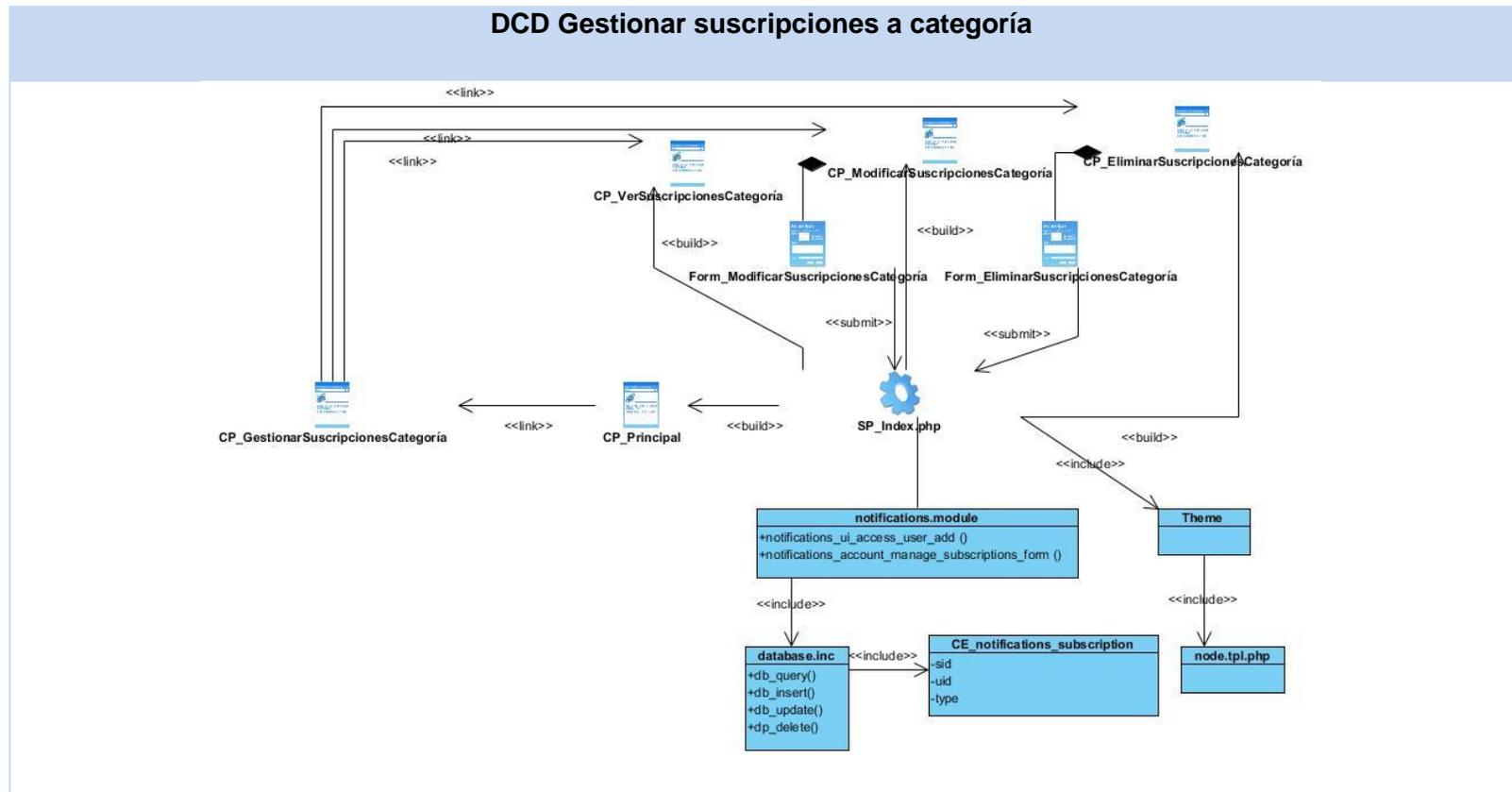


Tabla 46 DCD Filtrar lista de suscripciones

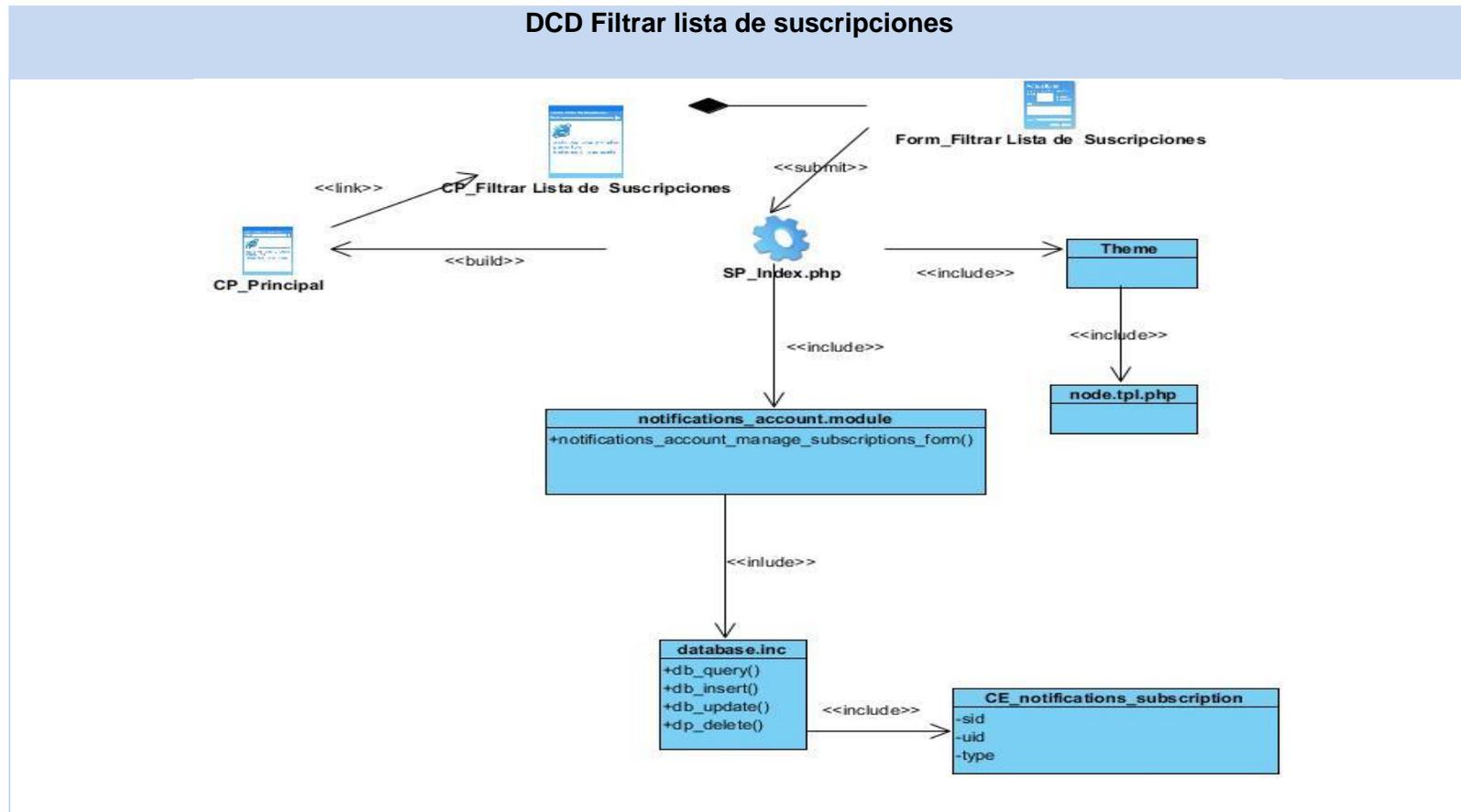


Tabla 47 DCD Listar suscripciones

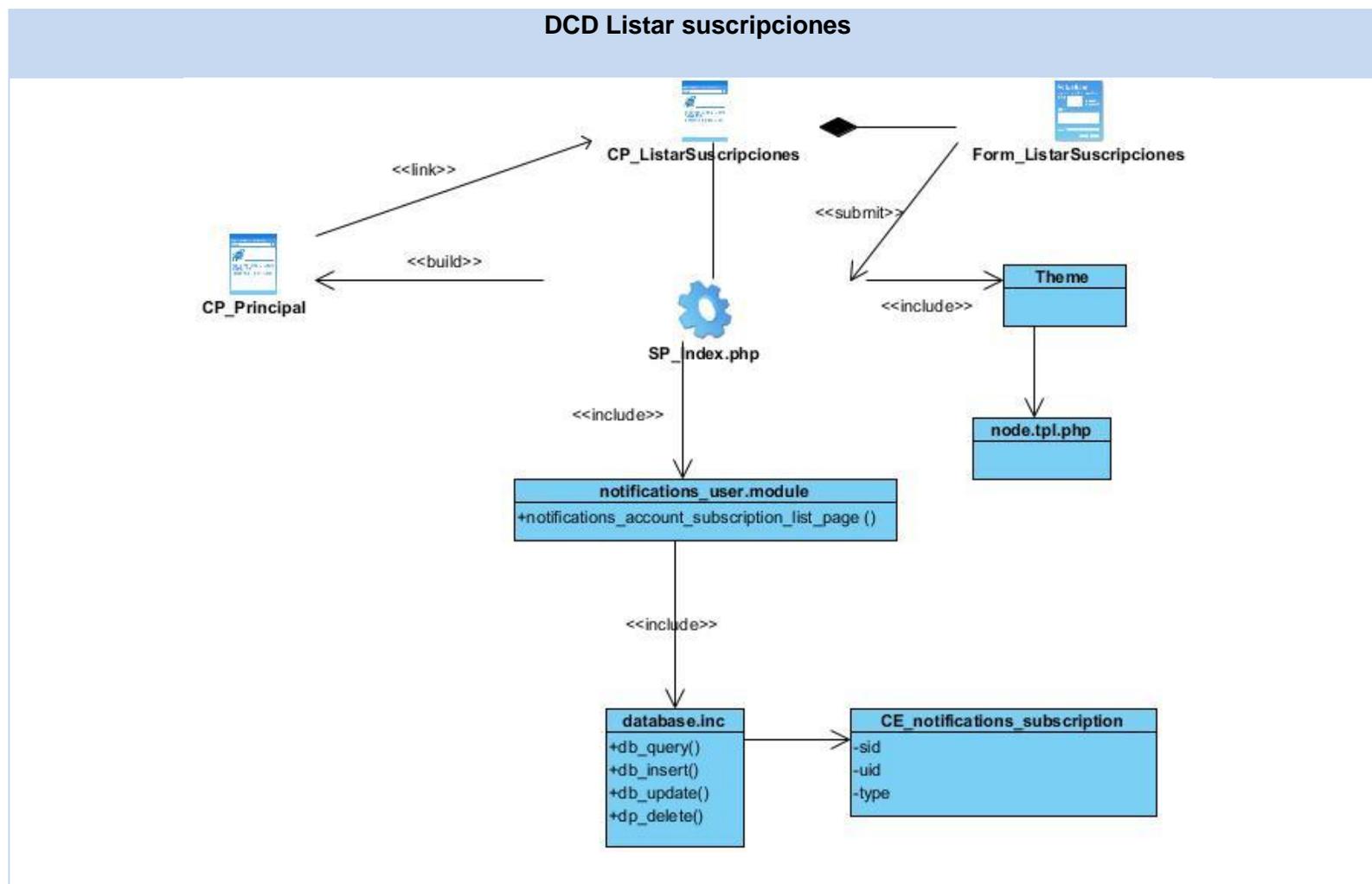
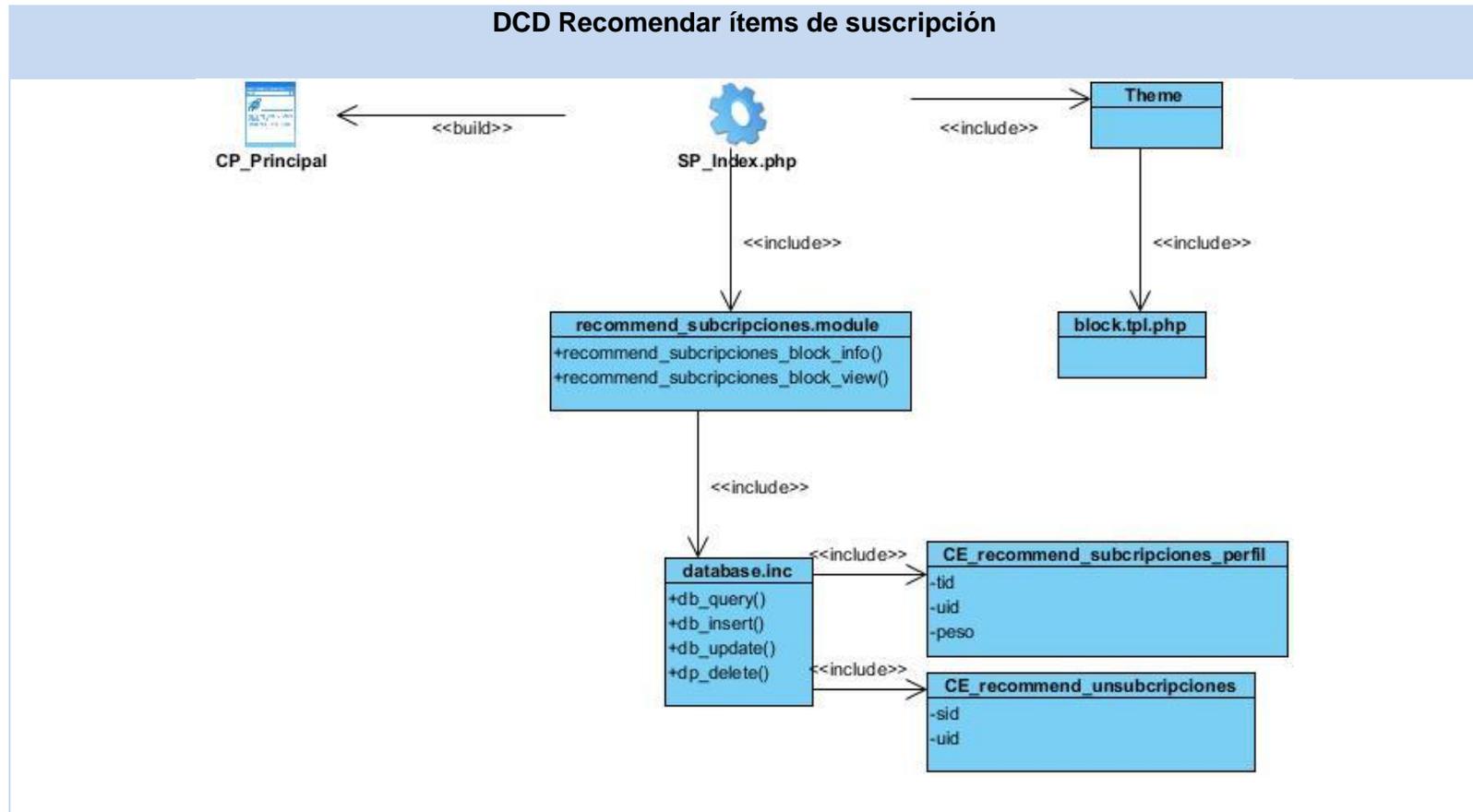


Tabla 48 DCD Recomendar ítems de suscripción



Anexo 5 Diseño de casos de prueba

Tabla 49 CP Gestionar mensaje.

SC "Gestionar mensajes"

Escenario	Descripción	Para	Asunto	Cuerpo	Respuesta del sistema	Flujo central
EC 1.1 Selecciona la opción Mensajes.	El actor selecciona la opción Mensajes..	NA	NA	NA	El sistema muestra un listado con los mensajes existentes y permite, crear, ver o eliminar.	Escritorio de trabajo/ Mensajes
		NA	NA	NA		
		NA	NA	NA		
		NA	NA	NA		

SC 1: Crear nuevo mensaje

Escenario	Descripción	Para	Asunto	Cuerpo	Respuesta del sistema	Flujo central
EC 1.1 Selecciona Escribir nuevo mensaje	El usuario selecciona la opción de Escribir nuevo mensaje	NA	NA	NA	Muestra un formulario para introducir los datos del nuevo mensaje: Para, Asunto, Cuerpo Permite: • Cancelar. • Enviar mensaje.	Escritorio de trabajo/Mensajes/Escribir nuevo mensaje.
		NA	NA	NA		
		NA	NA	NA		
		NA	NA	NA		
EC 1.2	El actor envía el	V	V	V	Valida los datos, crea	Escritorio de

<i>Selecciona Enviar mensaje</i>	<i>mensaje</i>				<i>el mensaje y lo envía.</i>	<i>trabajo/Mensajes/Escribir nuevo mensaje/ Enviar mensaje.</i>
<i>EC 1.3 Datos Incorrectos</i>	<i>Los datos entrados por el usuario son incorrectos.</i>	<i>V</i>	<i>I</i>	<i>I</i>	<i>Muestra un mensaje de información y señala los campos en rojo</i>	<i>Escritorio de trabajo/ Mensajes /Escribir nuevo mensaje/ Enviar mensaje.</i>
<i>EC 1.4 Datos obligatorios</i>	<i>Existen datos obligatorios.</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>Muestra un mensaje de información y señala los campos en rojo</i>	<i>Escritorio de trabajo/ Mensajes/ Escribir nuevo mensaje/ Enviar mensaje.</i>
		<i>V</i>	<i>I</i>	<i>V</i>		
		<i>V</i>	<i>V</i>	<i>I</i>		
		<i>I</i>	<i>V</i>	<i>V</i>		
		<i>I</i>	<i>V</i>	<i>I</i>		
		<i>I</i>	<i>I</i>	<i>V</i>		
		<i>v</i>	<i>I</i>	<i>I</i>		
		<i>V</i>	<i>I</i>	<i>I</i>		
		<i>I</i>	<i>I</i>	<i>I</i>		
<i>EC 1.5 Selecciona Cancelar</i>	<i>El usuario cancela el nuevo mensaje</i>	<i>V</i>	<i>I</i>	<i>V</i>	<i>Elimina los datos creados.</i>	<i>Escritorio de trabajo/ Mensajes/ Escribir nuevo mensaje/ Cancelar</i>
		<i>V</i>	<i>V</i>	<i>I</i>		
		<i>I</i>	<i>V</i>	<i>V</i>		
		<i>I</i>	<i>V</i>	<i>I</i>		
		<i>I</i>	<i>I</i>	<i>V</i>		
		<i>v</i>	<i>I</i>	<i>I</i>		
		<i>V</i>	<i>I</i>	<i>I</i>		
		<i>I</i>	<i>I</i>	<i>I</i>		

SC 2: Ver mensaje

Escenario	Descripción	Para	Asunto	Destinatario	Respuesta del sistema	Flujo central
EC 2.1 Ver mensaje	El actor selecciona un mensaje haciendo clic sobre él.				Muestra el contenido del mensaje y permite: Responder el mensaje. Ver CU "Responder mensaje"	Escritorio de trabajo/ Mensajes.
		NA	NA	NA		

SC 3: Eliminar

Escenario	Descripción	Para	Asunto	Destinatario	Respuesta del sistema	Flujo central
EC 3.1 Selecciona Eliminar.	El actor selecciona la opción de eliminar un mensaje.	NA	NA	NA	Muestra la opción de eliminar el mensaje seleccionado.	Escritorio de trabajo/ Mensajes/ Eliminar.

SC 3: Ver todos los mensajes

Escenario	Descripción	Para	Asunto	Destinatario	Respuesta del sistema	Flujo central
EC 3.1 Selecciona Ver todos los	El actor selecciona la opción de ver todos los mensajes.	NA	NA	NA	Muestra un listado de todos los mensajes recibidos y enviados y	Escritorio de trabajo/ Mensajes/ Ver todos los mensajes.

mensajes.					de cada uno muestra : <ul style="list-style-type: none"> • Asunto • Participantes • Fecha del mensaje
-----------	--	--	--	--	---

Tabla 50 CP Configurar plantillas de mensaje

SC "Configurar plantillas de mensaje"

Escenario	Descripción	Asunto	Encabezado	Contenido	Pie	Respuesta del sistema	Flujo central
EC 1.1	El actor selecciona la opción Plantillas de mensaje.	NA	NA	NA	NA	Muestra una interfaz todas las plantillas de mensaje que pueden ser configuradas.	Escritorio de trabajo/Plantillas de mensaje
EC 1.2	El actor selecciona la plantilla deseada.	NA	NA	NA	NA	Muestra una interfaz para introducir los datos de la plantilla y permite: Guardar.	Escritorio de trabajo/Plantillas de mensaje/Guardar
		NA	V	NA	NA		
		V	V	V	V		
		I	I	I	I		
		NA	NA	NA	NA		
		V	V	V	NA		
		I	I	NA	NA		
		I	NA	NA	NA		
		V	V	I	I		
		V	V	V	I		
		I	I	I	NA		