



Universidad de las Ciencias Informáticas

Facultad 4

Análisis y diseño del módulo Comunicaciones de la plataforma ZERA

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

Autor:

Jandy Rosada Valladares

Tutora:

Ing. Yuneikys Recio Miranda

Cotutora:

Ing. Karenia Donatien Goliath

Junio, 2011

La Habana, Cuba

Declaración de Autoría

Yo, Jandy Rosada Valladares, declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste, firmo la presente declaración de autoría en La Habana a los días ____ del mes de _____ del año _____.

Firma del Autor
Jandy Rosada Valladares

Firma del Tutor
Yuneikys Recio Miranda

Resumen

Existen actualmente diversos sistemas y plataformas para la gestión del aprendizaje virtual. En la Universidad de las Ciencias Informáticas se desarrolla una Plataforma para la Gestión del Aprendizaje. En el presente trabajo se realizó el Análisis y Diseño del módulo de comunicaciones para dicha plataforma. Inicialmente se documentaron los resultados del estudio realizado con respecto a las soluciones similares existentes y se describieron las tecnologías y herramientas que son factibles para el desarrollo, seleccionando las que se ajustan a los requerimientos de la aplicación. Luego se describieron las características del sistema que se propone, a través de los artefactos propuestos por la metodología de desarrollo RUP. Finalmente se obtuvo un completo análisis y diseño de las funcionalidades necesarias para el sistema, lo cual permitirá su posterior implementación.

Palabras claves: comunicaciones, módulo, plataforma, ZERA.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	5
1.1. Introducción	5
1.2. Conceptos asociados al dominio del problema	5
1.2.1. Plataformas de teleformación	5
1.2.2. Mensajería interna.....	6
1.2.3. Avisos y anuncios.....	6
1.2.4. Agenda.....	7
1.2.5. Foro de debate	7
1.3. Análisis de soluciones similares existentes	7
1.3.1. Sistemas a nivel internacional	8
1.3.2. Sistemas a nivel nacional	12
1.4. Tendencias y tecnologías actuales	14
1.4.1. Metodologías de desarrollo	14
1.4.2. Lenguajes de desarrollo y de modelado	17
1.4.3. Herramientas CASE	26
1.5. Selección de la metodología y tecnologías para el desarrollo	27
1.6. Conclusiones	28
Capítulo 2: Características del Sistema	29
2.1. Introducción	29
2.2. Descripción de los procesos de negocio	29
2.2.1. Glosario de términos del modelo de dominio	30
2.3. Descripción del sistema propuesto.....	30
2.4. Especificación de requisitos	31
2.4.1. Requisitos funcionales.....	31
2.4.2. Requisitos no funcionales.....	33

2.5. Definición de los Casos de Uso del Sistema	35
2.6. Patrones utilizados en la creación de los CU	35
2.7. Descripción de los actores del sistema	36
2.8. Diagrama de casos de uso del sistema.....	37
2.9. Descripción de los casos de uso del sistema	38
2.10. Conclusiones	42
Capítulo 3: Análisis y Diseño	43
3.1. Introducción	43
3.2. Modelo de Análisis	43
3.2.1. Diagramas de clases del Análisis	44
3.3. Modelo del Diseño	46
3.3.1. Patrones utilizados	46
3.3.2. Descripción de las clases del Diseño.....	48
3.3.3. Diagramas de clases del Diseño	58
3.3.4. Diagramas de secuencia del Diseño	63
3.3.5. Diseño de la Base de Datos	65
3.4. Conclusiones	67
Conclusiones	68
Recomendaciones	69
Glosario de Términos.....	72
Referencias Bibliográficas.....	71

Índice de Figuras

Figura 1.1: Calendario de Moodle	9
Figura 1.2: Foros de Debate de Moodle	9
Figura 1.3: Foros de Debate de Claroline	10
Figura 1.4: Agenda de Claroline.....	11
Figura 1.5: Logo de UML	24
Figura 2.1: Diagrama de Modelo del Dominio	29
Figura 2.2: Diagrama de CUS	38
Figura 3.1: DCA CU Gestionar Anuncio.	44
Figura 3.2: DCA CU Gestionar Eventos de la Agenda.	44
Figura 3.3: DCA CU Gestionar Correspondencia Interna.	45
Figura 3.4: DCA CU Gestionar Temas de Foro.....	46
Figura 3.5: Patrón Modelo – Vista – Controlador	47
Figura 3.6: DCD CU Gestionar Anuncio.....	59
Figura 3.7: DCD CU Gestionar Temas de Foro.....	60
Figura 3.8: DCD CU Gestionar Eventos de la Agenda.	61
Figura 3.9: DCD CU Gestionar Mensajería Interna.	62
Figura 3.10: DSD CU Gestionar Eventos de la Agenda – Incluir Evento.	63
Figura 3.11: DSD CU Gestionar Eventos de la Agenda – Modificar Evento.	63
Figura 3.12: DSD CU Gestionar Eventos de la Agenda – Eliminar Evento.....	64
Figura 3.13: DSD CU Gestionar Eventos de la Agenda – Ver Evento.....	64
Figura 3.14: Diagrama de Modelo de Datos de Avisos	65
Figura 3.15: Diagrama de Modelo de Datos de Foro.....	65
Figura 3.16: Diagrama de Modelo de Datos de Mensajería	66
Figura 3.18: Diagrama de Modelo de Datos de Anuncios	67

Índice de Tablas

Tabla 2.1: Descripción de los actores del sistema.....	37
Tabla 2.2: Descripción a nivel resumen del CU Gestionar Temas de Foro.....	39
Tabla 2.3: Descripción a nivel resumen del CU Gestionar Correspondencia Interna.....	40
Tabla 2.4: Descripción a nivel resumen del CU Gestionar Eventos de la Agenda.....	41
Tabla 2.5: Descripción a nivel resumen del CU Gestionar Anuncios.....	42
Tabla 3.1: Clases del Análisis.....	43
Tabla 3.2: Descripción de la clase tb_notice	48
Tabla 3.3: Descripción de la clase r_notice_user	49
Tabla 3.4: Descripción de la clase sf_guard_user	50
Tabla 3.5: Descripción de la clase SP_noticesActions	50
Tabla 3.6: Descripción de la clase SP_AdvertisementActions	50
Tabla 3.7: Descripción de la clase tb_advertisement	51
Tabla 3.8: Descripción de la clase tb_forum	52
Tabla 3.9: Descripción de la clase tb_topic	53
Tabla 3.10: Descripción de la clase tb_message_topic	54
Tabla 3.11: Descripción de la clase tb_study_program	54
Tabla 3.12: Descripción de la clase tb_group	54
Tabla 3.13: Descripción de la clase SP_ForumActions	55
Tabla 3.14: Descripción de la clase sf_calendar	56
Tabla 3.15: Descripción de la clase SP_sfCalendarActions	56
Tabla 3.16: Descripción de la clase SP_MessagingActions	57
Tabla 3.17: Descripción de la clase tb_message	58
Tabla 3.18: Descripción de la clase r_guard_user_message	58

Introducción

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) juegan un papel fundamental en el desarrollo de todos los sectores de la sociedad, desde la comunicación hasta la gestión de negocios. Principalmente, en la educación las TIC causaron un gran impacto ofreciendo nuevos y ventajosos ambientes para el aprendizaje. Esto favoreció el surgimiento de un nuevo modelo de enseñanza, mediante el cual es posible aprender sin necesidad de estar el profesor y los estudiantes en el mismo lugar y al mismo tiempo. Esta nueva modalidad educativa se conoce como: educación a distancia.

El surgimiento de la educación a distancia dio paso al aprendizaje electrónico o *e-Learning*, que permite transmitir el conocimiento por medios electrónicos. De esta manera se logra optimizar el proceso de enseñanza y aprendizaje, aprovechando las ventajas de la educación a distancia.

La aplicación de las TIC a los procesos de enseñanza, así como los cambios en los modelos pedagógicos, se ha visto plasmado en los entornos virtuales de enseñanza y aprendizaje (EVEA). Los EVEA se apoyan en sistemas informáticos que incluyen herramientas adaptadas a las necesidades de la institución para la que se desarrollan o adaptan. Estos sistemas reciben el nombre de plataformas y actualmente algunas de ellas están estandarizadas (aunque permiten la adaptación a situaciones concretas), mientras que otras son completamente personalizadas.

En Cuba desde hace varios años diversas instituciones educativas comenzaron a utilizar estas plataformas web, como apoyo a la educación tradicional y para impartir algunos cursos a distancia. También se desarrollaron herramientas para el uso del aprendizaje electrónico en los procesos docentes educativos.

Un ejemplo claro de la aplicación de estas tecnologías en nuestro país es en la Universidad de las Ciencias Informáticas (UCI), la misma cuenta con un gran número de proyectos productivos, entre estos podemos encontrar el proyecto Alfaomega, el cual desarrolla la plataforma para la gestión del aprendizaje ZERA. Esta plataforma integra las principales especificaciones y estándares educativos, desarrollados y utilizados a nivel mundial en plataformas de aprendizaje colaborativo. Permite además la gestión de los hiperentornos de aprendizaje y su visualización.

La plataforma ZERA está dividida en varios subsistemas, los cuales se encuentran estrechamente interrelacionados. Alfaomega Grupo Editor S.A necesita un espacio en la plataforma ZERA dirigido a atender las comunicaciones en la misma, es necesario que exista en la plataforma una comunicación directa entre los docentes y los estudiantes; para que los estudiantes puedan aclarar sus dudas con el profesor mediante el sistema y poder además debatir las mismas. Se necesita que mediante diversos medios se puedan dar a conocer diferentes informaciones a los docentes y estudiantes. Medios como los anuncios, que son mensajes cuya finalidad es informar de algún suceso o actividad de interés para usuarios de la plataforma, se encuentren o no autenticados en esta, y los avisos o noticias que se publican con el fin de difundir diferentes informaciones de manera más personalizada y solo a usuarios autenticados.

Es necesario el uso de una mensajería interna, un sistema de envío de mensajes o correos dentro de la plataforma por parte de los usuarios de la misma, ya que esto tributa a una mejor comunicación. Además de un calendario para controlar las tareas y disponer de la calendarización de eventos. Un calendario que permita observar de forma rápida cuales son las actividades de los próximos días. Con el fin de crear un espacio de debate y discusión de las diferentes temáticas establecidas por los docentes, es indispensable la creación de un foro, pues este permitirá intercambiar ideas, llevar a cabo una participación activa de los estudiantes y docentes, lo cual produce en la mayoría de los casos un aprendizaje activo. El sistema deberá contar con notificaciones, que son el espacio conceptual de recordatorios de tareas y eventos, estas se hacen llegar por mensajería y/o agenda. A partir de estas necesidades se precisa la creación de un módulo de comunicaciones para la plataforma ZERA.

Partiendo del análisis de las necesidades planteadas se define como **problema a resolver**: *¿Cómo garantizar la comunicación de todos los usuarios dentro de la plataforma ZERA?* El **objetivo general** de la investigación es: *Realizar el análisis y diseño del módulo de Comunicaciones de la plataforma de gestión de aprendizaje ZERA, que permita la interacción de los usuarios en la misma.* Del cual se desglosan los siguientes **objetivos específicos**:

- ✓ Realizar un estudio del estado del arte de plataformas educativas.

- ✓ Investigar las tendencias y tecnologías actuales para el desarrollo de plataformas educativas.
- ✓ Realizar el análisis y diseño de una propuesta de solución que permita la interacción entre los usuarios de la plataforma ZERA.

El **objeto de estudio** es, las Comunicaciones en las plataformas educativas para la gestión del aprendizaje y el **campo de acción**, las Comunicaciones en la plataforma para la gestión del aprendizaje ZERA.

Los métodos de investigación que soportan el desarrollo del presente trabajo son la combinación dialéctica de los métodos Teóricos y Empíricos. Entre los métodos empleados se encuentran:

Métodos Teóricos

Histórico-Lógico: posibilita el estudio de la evolución y desarrollo de las plataformas educativas para la gestión del aprendizaje.

Modelación: se utiliza en la confección de los diagramas que permitirán representar la propuesta de solución.

Analítico-Sintético: permite analizar las bibliografías y materiales relacionados con las Comunicaciones en las plataformas educativas para la gestión del aprendizaje.

Métodos Empíricos

Observación: para conocer la esencia del problema planteado, lo cual aporta un conocimiento más detallado de lo que se quiere, lo que hace falta hacer y cómo hay que hacerlo.

Tareas a cumplir en el desarrollo del trabajo:

- ✓ Realización del análisis bibliográfico para conformar la base teórico metodológico de la investigación.
- ✓ Investigación de las tendencias y las tecnologías actuales relacionadas con la interacción de los usuarios en plataformas de aprendizaje.
- ✓ Investigación del estado del arte de soluciones similares a la plataforma para la gestión del aprendizaje ZERA.
- ✓ Elaboración de un Plan de Captura de Requisitos.

- ✓ Descripción de las funcionalidades del sistema.
- ✓ Realización del análisis y diseño de una propuesta de solución.

El presente documento está compuesto por 3 capítulos que incluyen todo lo relacionado con el trabajo investigativo, así como el análisis y diseño del módulo de Comunicaciones de la plataforma de gestión de aprendizaje ZERA.

Capítulo 1: Fundamentación Teórica. Se hace referencia a los elementos teóricos que constituyen la base de la investigación realizada. Se describen los conceptos esenciales para comprender el problema, así como las tecnologías y herramientas a utilizar en el análisis y diseño del módulo de Comunicaciones de la plataforma ZERA.

Capítulo 2: Características del sistema. Se especifican los requisitos funcionales y no funcionales necesarios para el desarrollo del sistema, así como las descripciones de los casos de uso del mismo.

Capítulo 3: Análisis y diseño del sistema. Se describe la solución que se propone a partir de diferentes modelos y diagramas.

Capítulo 1

Fundamentación Teórica

1.1. Introducción

Para lograr el desarrollo de una buena aplicación es necesario realizar un estudio de los diferentes sistemas similares al que se desea crear. Además investigar acerca de las herramientas y tecnologías necesarias para su desarrollo.

1.2. Conceptos asociados al dominio del problema

La revolución dada por el uso de Internet y los servicios que se brindan, han posibilitado diferentes formas de trabajo en el área del aprendizaje electrónico. En la actualidad los espacios de formación empleados por instituciones educativas incluyen plataformas de teleformación.

1.2.1. Plataformas de teleformación

Las plataformas de teleformación tienen varias denominaciones, pueden ser Entornos Virtuales de Aprendizaje (EVA), Ambientes Virtuales de Aprendizaje (AVA), Sistemas Telemáticos de Teleformación o Plataformas de Gestión de Aprendizaje.

En un EVA se combinan varias herramientas virtuales con la finalidad de dar soporte a profesores y estudiantes, para poder optimizar las distintas fases del proceso de enseñanza y aprendizaje. Estas herramientas posibilitan: comunicación sincrónica y asincrónica, gestión de materiales de aprendizaje (para generación de contenidos y actividades), gestión de personas participantes (gestión administrativa y sistemas de seguimiento, así como evaluación de proceso de los alumnos). (1)

Un AVA es una propuesta metodológica para aplicar los modelos educativos innovadores, en los cuales deben fomentar ambientes de aprendizaje interactivos, sincrónicos y asincrónicos, donde el docente se encuentra comprometido con el aprendizaje de sus alumnos y cumple un papel como asesor y facilitador, los estudiantes se convierten en actores de cambio con habilidades y modos de trabajo innovadores, en los cuales utilizan tecnologías de vanguardia, materiales didácticos, recursos de información y contenidos digitales. (2)

En “Sistemas de gestión del aprendizaje – Plataformas de teleformación” el autor plantea que una plataforma de teleformación, es una herramienta informática y telemática organizada en función de objetivos formativos de forma integral (es decir que se puedan conseguir exclusivamente dentro de ella) y de principios de intervención psicopedagógica y organizativos. (3)

En general se puede decir que una plataforma de teleformación no es más que una aplicación informática, la cual es utilizada para garantizar el aprendizaje y el intercambio de conocimiento entre los estudiantes y docentes de una institución educacional. Estas plataformas sirven de apoyo a un proceso docente, sea este completamente a distancia, presencial, o que combine ambas modalidades. Un papel importante dentro de estas plataformas lo constituye la comunicación entre los miembros que participan en el proceso de enseñanza y aprendizaje.

Esta comunicación es posible a través de herramientas síncronas (comunicación en tiempo real) y asíncronas (comunicación en espacio y tiempo diferente). Entre ellas podemos encontrar: Mensajería interna, avisos, anuncios, agenda y foro.

1.2.2. Mensajería interna

La mensajería interna hace posible la comunicación privada, también en caso necesario el envío de mensajes a grupos de alumnos para poder mantener una comunicación fluida y rápida. Además permite administrar los mensajes que son recibidos o enviados por alumnos y profesores.

1.2.3. Avisos y anuncios

Los avisos y anuncios se refieren a aquellas noticias que están destinadas a los docentes y estudiantes, pueden referirse a cualquier situación que se necesite comunicar, con la intención de aclarar, informar y enseñar respecto a cuestiones de interés para la institución. Además permiten a los docentes publicar anuncios dentro de su curso, para que estos aparezcan en un lapso de tiempo determinado. Es posible presentar anuncios a usuarios específicos, es decir a usuarios con un mismo rol o a todos los usuarios de la plataforma. Mientras que los avisos están destinados a dar a conocer informaciones para todo aquel que visite la plataforma, sea usuario o no de la misma.

1.2.4. Agenda

La agenda provee un calendario interactivo para controlar las actividades del curso y de la institución, de igual forma los usuarios de la plataforma pueden controlar sus tareas y disponer de la calendarización de eventos, posee la ventaja de encontrar un plan de actividades actualizado y preciso de forma sencilla en todo momento, además conocer de forma rápida cuales son las actividades de los próximos días.

1.2.5. Foro de debate

El foro de debate permite la comunicación asincrónica y plantear diferentes temas de debate que fomenten la comunicación y reflexión de los alumnos. Desde un principio el docente implicado en el funcionamiento del curso debe controlar las participaciones y el uso que se haga de esta herramienta. Es muy común que los foros tengan una serie de normas que sus miembros deben seguir para integrar el grupo, las que tienen como finalidad mantener un clima ameno entre sus integrantes para poder intercambiar opiniones e información de la mejor manera posible.

Con los foros se habilita un espacio virtual para el debate y discusión entre los miembros del grupo. Un docente puede proponer un tema, que desarrollará respuestas y opiniones por parte de sus grupos. Los foros tienen gran importancia ya que se pueden utilizar como espacio para introducir un tema, buscando la participación de los alumnos, partiendo de los conocimientos previos que estos posean. Añadiendo la posibilidad de ofrecer una presentación de lo que se verá, saber sus expectativas sobre el tema, el nivel de interés y motivación. También el foro constituye el medio ideal para la consulta, planteamiento o resolución de dudas, cuyo carácter abierto no solo ofrece la posibilidad de que la duda sea resuelta, de recibir una retroalimentación por parte del moderador o docente, sino que también puede recibirla de los compañeros o participantes del foro.

1.3. Análisis de soluciones similares existentes

Actualmente existe a nivel mundial una amplia gama de entornos virtuales de aprendizaje, destacándose entre ellos los descritos a continuación.

1.3.1. Sistemas a nivel internacional

Los procesos de enseñanza y aprendizaje han cambiado con el uso de las TIC, se han desarrollado diferentes entornos virtuales de aprendizaje, los cuales cuentan con herramientas aplicadas a las necesidades de la institución para la que se implementan o adaptan. Entre ellos podemos encontrar:

Moodle

La plataforma Moodle, proporciona los elementos que podrán utilizar los docentes para crear y diseñar diversas formas de transmitir el conocimiento a sus estudiantes, además permitirá al alumnado consultar el material de aprendizaje y realizar las actividades evaluativas. Moodle es una herramienta para crear espacios virtuales de aprendizaje, constituidos por recursos de información, como son los textos, las imágenes, los archivos de audio y video, las páginas web, entre otros. Los entornos virtuales creados con esta herramienta poseen además recursos de formación, como los exámenes, las encuestas y los foros. Entre los recursos con los que cuenta esta plataforma para garantizar la comunicación dentro de la misma se encuentran:

- ✓ **Chat:** es el espacio en el cual los participantes discuten en tiempo real a través de la plataforma temas de interés. Permite una interacción fluida mediante texto síncrono y soporta direcciones URL, emoticonos, integración de HTML, imágenes y otros. Todas las sesiones quedan registradas para verlas posteriormente, y pueden ponerse a disposición de los estudiantes. Pueden programarse sesiones periódicas que aparecerán en el calendario.
- ✓ **Calendario:** por medio del calendario, se podrá ver de una forma rápida cuales son las actividades de los próximos días.
Está dirigido solo para alumnos y no para el grupo o curso. Luego cada estudiante podrá mantener su propia agenda de actividades. (4)



Figura 1.1: Calendario de Moodle

- ✓ **Foro:** es aquí donde se desarrolla la mayor parte de los debates. Pueden estar estructurados de diferentes maneras, e incluso permite evaluar la participación. Posibilita ver los mensajes de varias maneras incluyendo imágenes adjuntas. Al suscribirse a un foro, los participantes recibirán copias de cada mensaje en su buzón de correo electrónico. Los estudiantes pueden ser inscritos a un foro por un profesor.

Añadir un nuevo tema de debate				
Debate	Empezado por	Grupo	Respuestas	
problema con una imagen	Manuel Suárez Flores		1	Juan Mar Sat, 21 de f
! Último empujón !	Francisco Fernández Delgado		0	Francisco Sat, 21 de f
Sobre las Tareas	José Rubio Rodríguez		2	Jo Mon, 16 de f
Restricciones en el acceso a zonas.	Juan Manuel Aljjo Lebrón		3	Juan Mar Fri, 13 de f
Cuestionario y subir un archivo de imagen al texto del curso.	Alicia Herranz Sánchez		3	Francisco Thu, 12 de f
Tarea en grupo (trabajo final)	María Victoria Vázquez Romero		1	Tue, 10 de f
INTRODUCCIÓN DE LAS PREGUNTAS DEL CUESTIONARIO	Alicia Herranz Sánchez		0	Alici Sat, 7 de f
Estoy harta de este editor!!!	maria.jesús.hernández.bisquet		1	Francisco Fri, 6 de f
solucionado bloque II	REYES TEJERA ALGARIN		1	Francisco Thu, 5 de f
Trabajo en grupo	Juan Manuel Aljjo Lebrón		1	Juan Mar Thu, 5 de f

Figura 1.2: Foros de Debate de Moodle

Moodle cuenta con distintos tipos de foros:

- ✓ Foros de debate sencillo: simplemente un intercambio de ideas sobre un solo tema, todo en una página, respondiendo a un único planteamiento inicial. Útil para debates cortos y muy concretos.
- ✓ Foro normal, para uso general: foro abierto donde cualquiera puede empezar un nuevo tema de debate cuando desee. Este es el foro más adecuado para uso general.

- ✓ Foro un debate por persona: cada persona puede plantear un nuevo tema de debate. Esta modalidad es muy útil cuando se desea que cada alumno o miembro del grupo comience una discusión y que todos los demás le respondan.

La característica más importante que posee esta plataforma es que mediante sus distintos tipos de foros logra una comunicación directa entre profesores y estudiantes. Posibilita ver los mensajes de varias maneras incluyendo imágenes adjuntas y permite además evaluar participación.

Claroline

Claroline es una plataforma de aprendizaje y trabajo virtual creada en el año 2001 en la Universidad católica de Louvain en Bélgica (UCL) (5). Cada espacio de dicha plataforma proporciona diversas herramientas que permiten crear contenidos de aprendizaje y gestión de actividades de formación. Esta plataforma posibilita: administrar una agenda con tareas y fechas límite, publicar anuncios que pueden ser enviados por correo electrónico, así como administrar foros públicos y privados, entre otros.

- ✓ **Hablar:** chat y foro, espacio público para hablar/debatir dividida en distintos temas (conversación asíncrona). Herramienta en línea para el debate (conversación síncrona).

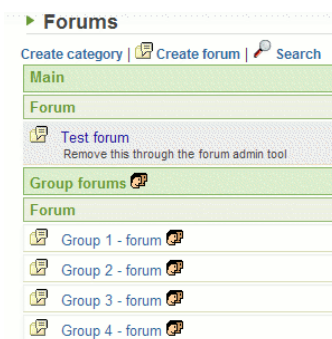


Figura 1.3: Foros de Debate de Claroline

- ✓ **Organizar:** agenda y anuncios, permite añadir actividades en el calendario del curso, mostrar el calendario completo visualizando las actividades de todos los cursos, adjuntar a una actividad un enlace a otras herramientas del curso o a un

recurso existente. Escribir un anuncio que pueda ser visualizado en la página de inicio del curso o mandarlo por correo electrónico a un usuario o grupo de usuarios.

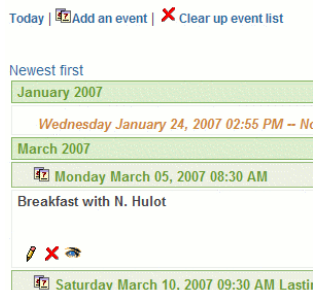


Figura 1.4: Agenda de Claroline

Entre las funcionalidades de esta plataforma se considera como la más importante la posibilidad de mostrar los anuncios de manera que puedan estar accesibles para todos los usuarios.

Dokeos

Dokeos es un entorno de aprendizaje electrónico, es una aplicación de administración de contenidos de cursos y también una herramienta de colaboración, es usada como un sistema de gestión de contenido para la educación. Los principales objetivos de la misma son: ser un sistema flexible y de muy fácil uso mediante una interfaz de usuario sumamente amigable, lograr una herramienta de aprendizaje, especialmente recomendada a usuarios que tengan nociones mínimas de computación cuyo objetivo es la preocupación por el contenido.

Entre las características principales con que cuenta esta plataforma para administrar contenidos se encuentran: distribución de contenidos, calendario, proceso de entrenamiento, chat en texto, audio y video, administración de pruebas y guardado de registros. (6)

Las características de mayor relevancia son:

- ✓ Aspecto: es una plataforma visiblemente agradable.
- ✓ Flexibilidad: se puede acceder a ella desde cualquier lugar con Internet, ya que no hace falta la instalación de la plataforma.

- ✓ Datos: brinda la posibilidad de llevar estadísticas para conocer los artículos o contenidos más vistos.
- ✓ Espacio: ofrece un espacio de alojamiento de 20Mb.
- ✓ Es intuitiva: para aquellos alumnos que jamás usaron una plataforma, Dokeos es fácil y organizada.
- ✓ Seguridad: permite crear copias de respaldo.

Respecto a las herramientas de comunicación ofrecidas por Dokeos se encuentra, un foro principal, aunque cada grupo puede tener a su vez todos los foros que sean necesarios. En cuanto al servicio de correo o mensajería interna, se puede utilizar a través de la sección “usuarios”, aunque si se desea tener en la plataforma un servicio de correo interno es necesario instalar un *plugin* adicional. Dokeos dispone de una sala de chat para discutir entre estudiantes y con el profesor. Además, esta plataforma ofrece la interesante posibilidad de realizar una conferencia, permitiendo la transmisión en directo de la voz del conferencista en fichero mp3, mp4 o mov. La posibilidad de seguir la conferencia con transparencias en *Power Point*, con un fichero pdf u otro soporte informático; y la intervención de los participantes en la conferencia a través del chat. También en esta plataforma cada grupo tiene unas herramientas comunes como: documentos, tareas, anuncios y agenda, en esta última el profesor puede incluir eventos dirigidos a todos los usuarios de curso o solo a un grupo en concreto.

Las características más importantes de esta plataforma son: el servicio de correo o mensajería interna y la agenda, en la cual un aspecto importante es que el profesor pueda incluir eventos dirigidos a los estudiantes de un curso o solo a un grupo en concreto.

1.3.2. Sistemas a nivel nacional

AprenDIST

AprenDIST es una plataforma para la educación a distancia, es flexible y ajustable a gran cantidad de cursos y está diseñada para un alumnado con determinadas características. Su diseño es sencillo e intuitivo por lo que los usuarios de este sistema, pueden llegar a dominarlo con solo conocer un poco de navegación web. Esta plataforma ha comenzado a revolucionar el proceso de educación existente y a medida que aumenta su explotación, los impactos se van extendiendo hacia diversas entidades.

Existen dos modos de emplear el sistema:

Versión Online: a través de este modo se utilizan todas las funciones de comunicación y retroalimentación del proceso educativo a distancia, para el cual el sistema está diseñado. Entre ellas el chat, el correo interno, el foro de discusión, el mural de anuncios, la matrícula en línea (alta baja y modificación) de los diferentes usuarios, el acceso a cada una de las partes del contenido, en caso de que el curso sea guiado, el profesor puede hacer un seguimiento activo de los estudiantes, a partir de sus estadísticas, entre otras. Ambas versiones poseen toda una variedad de ejercicios, algunos los califica la máquina y otros el profesor.

Versión Local: este modo ha sido una nueva solución ante las restricciones de comunicación que han presentado una parte significativa de los estudiantes (como es el caso de los alumnos de la universalización de la enseñanza que no tenían acceso a conexiones físicas). Consiste en la confección de un CD-ROM que contiene el mismo formato de AprenDIST, donde se ofrecen los contenidos, glosarios de términos, presentaciones y materiales complementarios de las asignaturas.

Como es una versión local (no utiliza conexiones), no aparecen las opciones de comunicación o retroalimentación. (7)

Al analizar los elementos que se han brindado anteriormente, se concluye que dichas plataformas no se adaptan al modelo interactivo cubano reflejado en los hiperentornos de aprendizaje. Algunas de ellas están pensadas para estudiantes universitarios y no para alumnos de pre-universitario que es el objetivo que se quiere lograr. No obstante en cuanto a las comunicaciones entre los usuarios de estas plataformas, se obtiene una visión más amplia de las funcionalidades a tener presentes en el módulo de Comunicaciones de la plataforma ZERA. A partir del estudio realizado se tendrán en cuenta funcionalidades como los foros de discusión, presentes en la mayoría de las plataformas estudiadas. La mensajería interna como apoyo al flujo de la información, una agenda para los usuarios de la plataforma como una manera de controlar sus eventos, anuncios y avisos para la divulgación de información de interés.

1.4. Tendencias y tecnologías actuales

Para realizar un correcto análisis y diseño de un sistema es necesario investigar algunas de las diferentes metodologías y herramientas existentes a nivel mundial. Para de esta manera seleccionar las apropiadas.

1.4.1. Metodologías de desarrollo

Todo desarrollo de software es riesgoso y difícil de controlar, si no se utiliza una metodología que guíe el proceso, se obtienen clientes insatisfechos con los resultados que esperan alcanzar del producto final. Además de que no existirá una guía sólida que les facilite el trabajo a los integrantes del equipo de desarrollo.

Programación Extrema (XP): es una metodología ligera o ágil para el desarrollo de software eficiente y altamente efectivo. Básicamente, la programación extrema, busca dos objetivos claramente: hacer un software con calidad y de la forma más rápida posible. De hecho estos son los objetivos fundamentales de cualquier metodología aplicada al desarrollo de software y a cualquier otra área en general. XP centra su atención en la producción de software con una fuerte arquitectura, intentando sacar productos al mercado rápidamente, con gran calidad y motivando al equipo de trabajo para seguir mejorando esta tendencia.

Esta metodología presenta muchos puntos comunes con el desarrollo incremental, comenzando por el hecho de que el software desarrollado con XP se realiza de forma incremental. Se divide en varias fases: (8)

- ✓ Probar: se debe desarrollar la prueba desde el momento que se conocen las historias del usuario.
- ✓ Escuchar: tanto para diseñar, como para desarrollar pruebas, como para desarrollar, se necesita conocer exactamente lo que se quiere, para ello, se debe aprender a escuchar muy bien al cliente, al jefe de proyecto y a todos los implicados en el proyecto.
- ✓ Diseñar: el diseño también debe ser incremental y debe estar incluido en el software, lo cual quiere decir que la estructura de este debe ser clara. Se debe diseñar lo que las necesidades del problema requieren, no lo que se crea que debería ser el diseño.

Tanto el análisis como el diseño son tareas muy importantes en XP, pero se realizan con un enfoque más ligero y transparente. El análisis es una parte fundamental, aquí se intentan recoger todas las necesidades del usuario. A partir de estas necesidades surgen las “historias de usuarios” que serán el paso inicial para comenzar a desarrollar nuestro sistema.

Esta metodología suele estar especialmente orientada a proyectos cortos, aquellos en los cuales los equipos de desarrollo son pequeños, con plazos reducidos, requisitos volátiles, y basados en nuevas tecnologías, y en los cuales el cliente forma parte del equipo de desarrollo.

Scrum: es un proceso en el que se aplican de manera sistemática un conjunto de mejores prácticas para trabajar colaborativamente, obteniendo el mejor resultado posible de un proyecto. Con este se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales. Por lo que se utiliza para resolver situaciones en que no se está entregando al cliente el producto que realmente necesita, cuando las entregas se atrasan demasiado, los costos se elevan muy por encima de la planificación o la calidad no es aceptable, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

Las actividades que se llevan a cabo en Scrum son las siguientes:

- ✓ Planificación de la iteración: en esta actividad se realiza la selección de requisitos donde el cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo selecciona los requisitos más prioritarios que se compromete a completar en la iteración.
- ✓ Ejecución de la iteración: diariamente el equipo realiza una reunión de sincronización, la cual no sobrepasa los 15 minutos. Cada miembro del equipo revisa el trabajo que el resto está realizando (dependencias entre tareas, progreso

hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder realizar las adaptaciones necesarias que permitan cumplir con el compromiso adquirido.

- ✓ Inspección y adaptación: al culminar la iteración se realiza una revisión de la iteración en la cual se tiene en cuenta la demostración y la retrospectiva. En la demostración el quipo presentará al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. (9)

Proceso Unificado de Desarrollo (RUP): es una metodología pensada para la Ingeniería de Software que va más allá del análisis y diseño orientado a objetos (10), para proporcionar un conjunto de técnicas que soportan el ciclo completo de desarrollo de software. Además de que unifica completamente a un equipo de desarrollo y optimiza la productividad de cada uno de los miembros del mismo, ayuda a los líderes de proyecto a incrementar su experiencia en el desarrollo. Los verdaderos aspectos definitorios del proceso unificado se resumen en tres características claves, dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

Iterativo e incremental: donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo. RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente unos más que otros.

Centrado en la arquitectura: la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para la construcción del software.

Dirigido por casos de uso: los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requisitos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

RUP cuenta con varias fases:

- ✓ Inicio: alcanzar un acuerdo entre todos los interesados respecto a los objetivos del ciclo de vida para el proyecto, generar el ámbito de trabajo, el caso de negocio, síntesis de arquitectura posible y el alcance del proyecto. (11)
- ✓ Elaboración: establecimiento de la línea base para la arquitectura del sistema y proporcionar una base estable para el diseño y el esfuerzo de implementación de la siguiente fase, mitigando la mayoría de los riesgos tecnológicos.
- ✓ Construcción: completar el desarrollo del sistema basado en la línea base de la arquitectura. En otras palabras lograr la funcionalidad operativa del software.
- ✓ Transición: garantizar que el software esté listo para entregarlo a los usuarios, y lograr la aprobación cuanto antes para liberar el producto al mercado.

Es considerada una de las metodologías más importantes, debido a que en cada ciclo de iteración se exige el uso de artefactos. Además, se aplica a un proceso más controlado, con numerosas políticas o normas, lo cual proporciona una mayor estabilidad en el proceso de desarrollo del software.

1.4.2. Lenguajes de desarrollo y de modelado

Un lenguaje de programación es una herramienta que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis. Es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo y especificarle, qué tipo de datos actúan y que acciones tomar bajo diferentes circunstancias.

1.4.2.1. Lenguajes del lado del cliente

HTML: Es el Lenguaje para el Formato de Documentos de Hipertexto, HTML por sus siglas del inglés "*HyperText Markup Language*". Es un sistema para definir tipos de documentos estructurados y lenguajes de marcas para representar dichos documentos. HTML no es un lenguaje de programación como C++, sino un sistema de etiquetas que no se compila por lo que si existe algún error este no lo señalará sino que representará en el navegador lo que interprete del código erróneo. Estas etiquetas permiten representar textos con diversos estilos, así como aplicar diferentes efectos a los párrafos que permitan que el entorno del sitio web sea agradable para el usuario. Además se pueden incorporar

a las páginas multimedia como sonidos, imágenes y videos. También con HTML se pueden crear vínculos a otras páginas y pueden incluir los *scripts* que permiten más interactividad en la navegación, como los de JavaScript.

El propio *World Wide Web Consortium (W3C)* define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global" (12). Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica. Es un lenguaje utilizado únicamente para dar estructura a una página web.

XHTML: el lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML. Técnicamente, HTML es descendiente directo del lenguaje SGML (Lenguaje de Marcado Generalizado), mientras que XHTML lo es de XML (que a su vez también es descendiente de SGML). (13)

Las páginas y documentos creados con XHTML son muy similares a las páginas y documentos HTML, este lenguaje va a permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella. Las discusiones sobre si HTML es mejor que XHTML o viceversa son recurrentes en el ámbito de la creación de contenidos web, aunque no existe una conclusión ampliamente aceptada. Actualmente, entre HTML 4.01 y XHTML 1.0, la mayoría de diseñadores escogen XHTML, pero tal vez en un futuro al elegir entre versiones diferentes esto cambie.

CSS: es un lenguaje de hojas de estilo creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML, es decir páginas web. (14) CSS (Hojas de Estilo en Cascada) es la mejor forma de separar los contenidos y su presentación, es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Cuando se crea una página web con HTML/XHTML se enmarcan los contenidos como son: párrafo, titular, texto destacado, tabla, lista de elementos, entre otros. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, posición de cada elemento dentro de la página, entre otros.

Ajax: en realidad el término es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como “JavaScript asíncrono + XML”. En el artículo publicado por Jesse James Garrett en el 2005 define Ajax como: “Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes”. (15)

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y *Document Object Model* (DOM). Las tecnologías que forman Ajax son XHTML y CSS para crear una presentación basada en estándares, DOM para la interacción y manipulación dinámica de la presentación. XML, XSLT y JSON para el intercambio y la manipulación de información, XMLHttpRequest para el intercambio asíncrono de la información y JavaScript para unir todas las demás tecnologías. Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores.

JavaScript: técnicamente, es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. (16) Se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios y es interpretado por la mayoría de estos en sus últimas versiones.

Se utiliza principalmente en un navegador web, permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. Puede incluirse en cualquier documento y es compatible con HTML. Puede ser utilizado por profesionales y por quienes se inician en el

desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

1.4.2.2. Lenguaje del lado del servidor:

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el lado del servidor pueden efectuar consultas a la base de datos y otras tareas para crear la página final que es lo que verá el cliente, la cual contiene solamente código HTML, como resultado de la ejecución de la página PHP.

ASP: es una tecnología desarrollada por Microsoft para el desarrollo de sitios web dinámicos. ASP proviene del inglés “*Active Server Pages*”, fue liberado por Microsoft en 1996. Las páginas web desarrolladas bajo este lenguaje necesitan tener instalado *Internet Information Server* (IIS).

El ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que este se encuentra únicamente en los archivos del servidor que al ser solicitado a través del web, es ejecutado, por lo que los usuario no tienen acceso más que a la página resultante en su navegador. (17)

Ruby: es un lenguaje de programación dinámico y de código abierto enfocado en la simplicidad y productividad, siguiendo los principios de una buena interfaz de usuario. (18) Su elegante sintaxis se siente natural al leerla y fácil al escribirla. Mezcla partes de Perl, Smalltalk, Eiffel, Ada, y Lisp, para formar un lenguaje que incorporara tanto la programación funcional como la programación imperativa. Desde su liberación pública en 1995, Ruby ha atraído a muchos desarrolladores de todo el mundo.

Este lenguaje posee entre sus principales características que reconoce la diferencia entre mayúsculas y minúsculas, soporta múltiples expresiones por líneas, separadas por punto

y coma “;”. Además dispone de manejo de excepciones y puede cargar librerías de extensiones dinámicamente si el sistema operativo lo permite.

Python: es un lenguaje de programación que permite trabajar más rápido e integrar sus sistemas de manera más eficaz, además es software libre, lo que permite utilizarlo incluso en productos comerciales.

Es multiplataforma, se ejecuta en Windows, Linux / Unix, Mac OS X, y ha sido portado a Java y .NET, máquinas virtuales. (19)

Es un lenguaje de scripting independiente de la plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones a servidores de red o incluso, páginas web. Es un lenguaje interpretado, por lo que no se necesita compilar el código fuente para poder ejecutarlo, esto ofrece ventajas de rapidez de desarrollo y el inconveniente de una menor velocidad.

Entre sus ventajas se encuentran:

- ✓ Gran cantidad de funciones y librerías
- ✓ Sencillo y rápido de programar
- ✓ Multiplataforma
- ✓ Licencia de código abierto
- ✓ Orientado a objetos

PHP: es un lenguaje de script interpretado, utilizado para la generación de páginas web dinámicas. PHP es un acrónimo recursivo que significa “*Hypertext Pre-processor*”. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo (20). El objetivo que persigue este lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas web. No es un lenguaje de marcas como podría ser HTML o XML. Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas.

Entre sus ventajas principales se pueden encontrar:

- ✓ El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- ✓ Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.

- ✓ Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- ✓ Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.

1.4.2.3. Framework

El concepto framework es ampliamente empleado en muchos ámbitos del desarrollo de sistemas de software, no solo para aplicaciones web. Se pueden encontrar framework para el desarrollo de aplicaciones médicas, visualización gráfica, desarrollo de juegos y otros. En general el término framework, se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar el código ya existente y promover buenas prácticas de desarrollo con el uso de patrones. Un framework web, por tanto, se puede definir como un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. (21)

Symfony: es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web, simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, es un framework que proporciona una estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, es un framework facilitador de la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. (22)

Entre sus principales características están:

- ✓ Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Zend Framework: es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. Es una implementación que usa código 100% orientado a objetos. La estructura de sus componentes es única; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como uso a voluntad del inglés *use-at-will*. (23)

Entre sus principales características están:

- ✓ Trabaja con MVC (*Model View Controller*, Modelo Vista Controlador).
- ✓ Cuenta con módulos para manejar archivos PDF, canales RSS, Web Services (Amazon, Flickr, Yahoo), etc.
- ✓ El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.
- ✓ Una solución para el acceso a base de datos que balancea el ORM con eficiencia y simplicidad.
- ✓ Completa documentación y pruebas de alta calidad.
- ✓ Soporte avanzado para internacionalización.
- ✓ Clases robustas para autenticación y filtrado de entrada.
- ✓ Soporta múltiples gestores de bases de datos entre ellos MySQL, PostgreSQL, Oracle.

CakePHP: es un framework de código abierto creado con PHP, siguiendo la filosofía de Ruby on Rails, que brinda una estructura sólida que permite el desarrollo rápido de aplicaciones web.

Sus principales características son:

- ✓ Licencia flexible.
- ✓ Caché flexible.
- ✓ Compatible con PHP4 y PHP5.
- ✓ CRUD integrado para la interacción con la base de datos.
- ✓ Soporte de aplicación.
- ✓ Arquitectura Modelo Vista Controlador.
- ✓ Despachador de peticiones, con URLs y rutas personalizadas y limpias.
- ✓ Plantillas rápidas y flexibles (sintaxis de PHP, con ayudantes).
- ✓ Ayudantes para AJAX, JavaScript, formularios HTML y otros.
- ✓ Componentes de Email, Cookie, Seguridad, Sesión y Manejo de solicitudes.
- ✓ Listas de control de acceso flexibles
- ✓ Funciona en cualquier subdirectorío del sitio web, con poca o ninguna configuración de Apache. (24)

1.4.2.4. Lenguaje Unificado de Modelado

Lenguaje Unificado de Modelado (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. UML es utilizado para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten.



Figura 1.5: Logo de UML

El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos y otros, hasta la implementación y configuración con los diagramas de despliegue. Este modelado visual

está diseñado para que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos). (25)

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- ✓ Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ Especificar: UML posibilita especificar cuáles son las características de un sistema antes de su construcción.
- ✓ Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

1.4.2.5. Modelado de interfaces de usuario

Al desarrollar software es muy importante diseñar correctamente la interfaz, no importa si es una página web o una aplicación de escritorio. Durante el análisis y el diseño es altamente recomendable realizar bocetos o esquemas de las pantallas con las que contará la aplicación, así como la forma de interactuar con las mismas. Para ello se cuenta con la ayuda de herramientas para crear bocetos de interfaces de usuario. Algunas de estas son:

Pencil Project: es la alternativa libre (GPL v2) y multiplataforma la cual permite el desarrollo de interfaces de usuario de manera completa. Es una aplicación disponible tanto para Windows como para Linux, esta cuenta con una potencia y flexibilidad considerables, nos permite crear proyectos con varias pantallas, en las cuales es posible añadir cualquier elemento de una interfaz de usuario típica (botones, tablas, cajas de activación, deslizadores y otros). (26)

No solo es útil para aplicaciones de escritorio sino que cuenta con multitud de elementos web. Una de sus características más importantes es la capacidad de alinear objetos para que el conjunto quede ordenado.

Balsamiq Mockups: es un programa de escritorio, programado en Flex y AIR (27), el cual proporciona gran rapidez en la creación de interfaces de usuario. Tiene como desventaja que no es una aplicación libre de costo y además al ser creada en AIR, es instalable en Windows, Linux y Mac OS.

Su interfaz es fácil de usar y posee una amplia colección de controles los cuales permiten la creación de una interfaz de usuario de forma sencilla.

1.4.3. Herramientas CASE

CASE (*Computer Aided Software Engineering*), su traducción al español es: Ingeniería de Software Asistida por Computación. (28) Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

Algunas de las herramientas CASE más populares son:

ArgoUML: fue concebida como un entorno y herramienta para usar en el análisis y diseño de sistemas de software orientados a objeto. En este sentido es similar a muchas de las herramientas CASE comerciales que son vendidas como herramientas para modelar sistemas software. ArgoUML tiene un número de distinciones muy importantes tales como:

- ✓ Se enfoca en investigación en psicología cognitiva para proporcionar nuevas características que incrementen la productividad, soportando las necesidades cognitivas de diseñadores y arquitectos de software orientado a objeto.
- ✓ Soporta estándares abiertos como UML, XMI, SVG, OCL y otros.
- ✓ Es una aplicación escrita en Java, debido a esto está disponible en cualquier plataforma soportada por Java.
- ✓ Es un proyecto de código abierto. La disponibilidad del código fuente asegura que una nueva generación de diseñadores de software e investigadores posea un entorno de trabajo probado, desde el que pueden conducir el desarrollo y evolución de tecnologías de herramientas. (29)

Visual Paradigm: es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Soporta todos los diagramas UML y el diagrama Entidad-Relación, ayudando a una rápida construcción de aplicaciones con calidad y a un menor coste (30). Posee una interfaz amigable y se puede modelar en varios idiomas. Es una herramienta colaborativa, pues soporta múltiples usuarios trabajando sobre el mismo proyecto. La documentación del proyecto puede ser generada automáticamente en varios formatos (web o .pdf), y permite control de versiones. Soporta la notación UML 2.1, ingeniería inversa, generador de informes, editor de figuras, integración con MS Visio, *plugin*, integración IDE con Visual Studio, IntelliJ IDEA, Eclipse, NetBeans y otros.

Entre sus ventajas:

- ✓ Apoya todo lo básico en cuanto a artefactos generados en las etapas de definición de requisitos y de especificación de componentes.
- ✓ Tiene disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java.
- ✓ Brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros con aplicaciones como por ejemplo Visio y Rational Rose.
- ✓ Brinda la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código.
- ✓ Brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.

1.5. Selección de la metodología y tecnologías para el desarrollo

Con la investigación realizada para la selección de las herramientas y tecnologías que se usan en el desarrollo del sistema, se consideró como característica fundamental el uso de herramientas de código abierto o perteneciente a la familia del software libre.

Como lenguaje de programación por parte del servidor se selecciona PHP en su versión 5.3.2, debido a la flexibilidad y claridad de su código. Además se ajusta a los requisitos del sistema y posee una gran comunidad de desarrollo tanto en la UCI como en Internet, lo cual garantiza la solución de problemas que puedan surgir. Para la programación por parte del cliente se selecciona JavaScript, ya que es un lenguaje interpretado por la

mayoría de los navegadores en sus últimas versiones. Permite el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. Además puede incluirse en cualquier documento y es compatible con HTML.

Como metodología a utilizar se decide seleccionar RUP, la cual utiliza como lenguaje de modelado UML para la descripción de los procesos. Dentro de sus características fundamentales se puede observar que mantiene al equipo enfocado en producir software de alta calidad en el tiempo requerido, brinda un conjunto de actividades orientadas a construir y documentar los artefactos necesarios para el desarrollo del software. Como framework se selecciona Symfony en su versión 1.4.6 debido a la gran cantidad de código que genera automáticamente y su facilidad de aprendizaje. Además provee mecanismos de defensa contra ataques a la base de datos, sistema y agrega un alto nivel de seguridad a las aplicaciones.

Para la creación de prototipos de interfaces de usuario es seleccionado Pencil Project versión 1.2 ya que es una herramienta completa, de código abierto y fácil de utilizar. Se adopta Visual Paradigm en su versión 6.4 como herramienta CASE ya que soporta la notación UML 2.1, ingeniería inversa, además brinda la posibilidad de generar código a partir de los diagramas y viceversa. Es multiplataforma, fácil de operar y posee una amplia documentación, lo que resulta provechoso para futuras modificaciones del subsistema.

1.6. Conclusiones

El estudio de diferentes plataformas existentes en la actualidad posibilitó la identificación de las principales herramientas de comunicación presentes en las mismas. La investigación realizada posibilitó la selección de la metodología y las herramientas más adecuadas para el desarrollo de 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

Debido a la dificultad que existe para determinar el proceso del negocio de manera que se logre observar claramente el alcance del mismo, las personas que lo inician y quiénes son los beneficiados con cada uno de estos procesos, se propone el desarrollo de un Modelo de Dominio. En el mismo se obtienen los tipos de objetos más importantes en el contexto del negocio.

El Modelo de Dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir. (31)

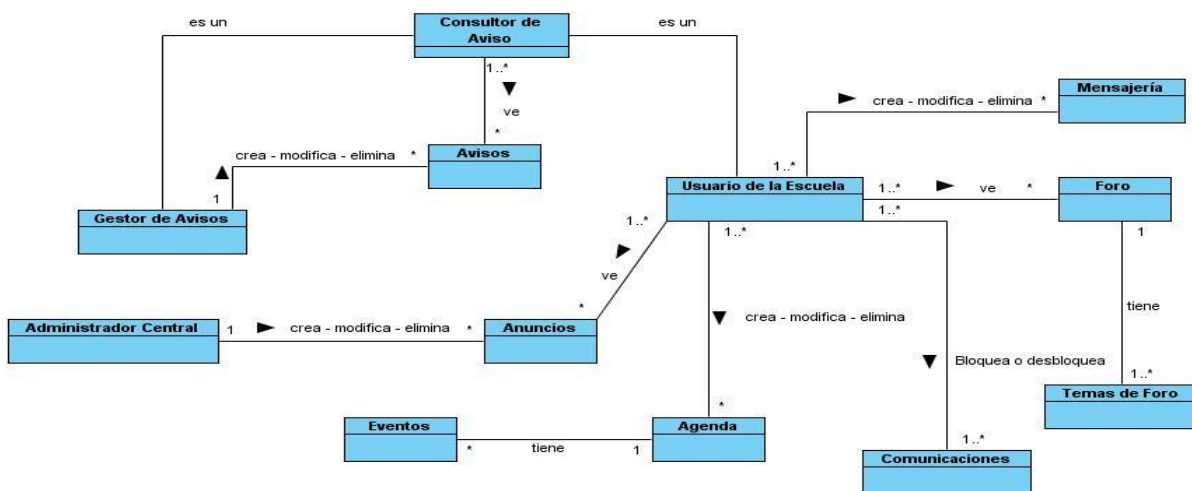


Figura 2.1: Diagrama de Modelo del Dominio

2.2.1. Glosario de términos del modelo de dominio

Usuario de la escuela: hace referencia a la persona que gestiona la mensajería interna, la agenda; y además puede ver anuncios y foros.

Administrador central: persona que crea, modifica o elimina los anuncios.

Gestor de avisos: persona que crea, modifica o elimina los avisos.

Consultor de avisos: persona que puede consultar los avisos existentes.

Mensajería interna: sistema de envío de mensajes o correos dentro de la plataforma.

Foro: espacio para debatir y discutir los distintos temas creados por los docentes.

Temas de Foro: diferentes contenidos a discutir en un foro.

Agenda: herramienta para llevar a cabo la calendarización de eventos y tareas de los usuarios de la plataforma.

Evento: actividades planificadas en una determinada fecha y horario.

Anuncios: espacio para informar de algún suceso o actividad de interés para los usuarios, se encuentren estos autenticados o no en la plataforma.

Avisos: son aquellos mensajes utilizados para dar a conocer variadas informaciones de una forma más dirigida, solo disponible para usuarios autenticados.

2.3. Descripción del sistema propuesto

Para dar solución a los objetivos anteriormente planteados, el módulo que se implementará constará de cinco submódulos (Foro, Mensajería Interna, Agenda, Anuncios y Avisos) y cinco roles (Administrador central, Gestor de avisos, Consultor de avisos, Usuario de la escuela y Sistema).

Para una mayor seguridad los usuarios que accedan al módulo deberán estar previamente autenticados en la plataforma, para de esta manera impedir el acceso no autorizado al módulo.

En el submódulo Foro al ser asociado un programa de estudio a un grupo y a un profesor se creará el foro correspondiente al grupo, de igual manera al dejar de estar asociado el profesor, al eliminarse un grupo o un programa de estudio se eliminará también el foro creado. Además se podrán ver los foros existentes según los permisos de cada usuario,

así como crear o eliminar los temas que se tratarán en los mismos. En el submódulo Mensajería Interna se crean, envían y reciben mensajes, además se pueden crear listas de contactos y ver las creadas hasta el momento. En el submódulo Agenda los profesores cuentan con la posibilidad de crear eventos para todos sus estudiantes o para algunos en específico, además envía una notificación al correo de los implicados en el evento. De manera general cada usuario puede gestionar sus eventos, los cuales estarán conformados por: asunto, descripción, fecha, hora.

El submódulo Anuncios permite la creación, modificación y eliminación de los mismos, estos estarán compuestos por un asunto, una descripción y una imagen. Los anuncios pueden ser consultados por todo el mundo, no es necesario estar autenticado en la plataforma para poder verlos. En el submódulo Avisos se gestionan los avisos de la plataforma, de los mismos se indicarán, el asunto, la descripción y los grupos asociados. Los avisos solo son accesibles para usuarios autenticados y pueden ser creados ser consultados por todos o de manera dirigida.

En resumen el módulo Comunicaciones brinda la posibilidad de gestionar todo lo referente a las comunicaciones en la plataforma, lo cual resultaría complejo de no contarse con este sistema.

2.4. Especificación de requisitos

El módulo de Comunicaciones tendrá que cumplir con una serie de condiciones y capacidades para dar solución a la necesidad de un sistema de intercambio en la plataforma. Se definen dichas condiciones y funcionalidades en los siguientes requisitos clasificados como funcionales y no funcionales.

2.4.1. Requisitos funcionales

Los requisitos funcionales especifican las capacidades o condiciones que el sistema debe ser capaz de cumplir. (32) A continuación se muestra el listado de los requisitos funcionales obtenidos.

RF1: Crear Foro.

RF2: Eliminar Foro.

RF3: Ver Foro.

RF4: Gestionar Temas de Foro

RF4.1: Incluir un nuevo Tema de Foro.

RF4.2: Ver los datos de un Tema de Foro.

RF4.3: Modificar los datos de un Tema de Foro.

RF4.4: Eliminar un Tema de Foro.

RF5: Listar Temas de Foro

RF6: Gestionar Correspondencia Interna

RF6.1: Incluir Correspondencia Interna.

RF6.2: Ver contenido de la Carpeta Bandeja de Entrada.

RF6.3: Ver contenido de la Carpeta Borradores.

RF6.4: Ver contenido de la Carpeta Papelera.

RF6.5: Ver contenido de la Carpeta Enviados.

RF6.6: Modificar los datos de una Correspondencia Interna.

RF6.7: Eliminar Correspondencia Interna.

RF6.8: Crear Lista de Contactos.

RF6.9: Ver Lista de Contactos.

RF7: Gestionar Eventos de la Agenda

RF7.1: Incluir un nuevo Evento.

RF7.2: Ver los datos de un Evento.

RF7.3: Modificar los datos de un Evento.

RF7.4: Eliminar un Evento.

RF8: Gestionar Anuncios

RF8.1: Incluir un nuevo Anuncio.

RF8.2: Modificar los datos de un Anuncio.

RF8.3: Eliminar un Anuncio.

RF9: Ver Anuncio.

RF10: Ver Datos de un Anuncio.

RF11: Ver listado de todos los Anuncios creados.

RF12: Gestionar Avisos

RF12.1: Incluir un nuevo Aviso.

RF12.2: Modificar los datos de un Aviso.

RF12.3: Eliminar un Aviso.

RF13: Ver Aviso.

RF14: Ver Datos de un Aviso.

RF15: Ver listado de todos los Avisos creados.

RF16: Mostrar la Agenda.

RF17: Bloquear o desbloquear las Comunicaciones

RF17.1: Bloquear Mensajería.

RF17.2: Bloquear Foro.

RF17.3: Bloquear Chat.

RF17.4: Desbloquear Mensajería.

RF17.5: Desbloquear Foro.

RF17.6: Desbloquear Chat.

2.4.2. Requisitos no funcionales

Los requisitos no funcionales se definen como las cualidades o propiedades que el software debe tener (33). Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Requisitos de seguridad y confiabilidad:

RNF 1: Seguridad de acceso y administración de usuarios: otorgamiento de privilegios y roles, asignación de perfiles. Los niveles de acceso están determinados por los diferentes roles válidos dentro de la misma.

RNF 2: Se debe tener en cuenta la seguridad de los contenidos negando la posibilidad de copiar estos. Se debe negar la posibilidad de descargar tanto los contenidos como los recursos asociados a estos.

RNF 3: Verificación sobre acciones irreversibles (mensajes de eliminaciones).

RNF 4: Garantizar que la información sea editada únicamente por quien está autorizado y posea permisos para ello.

Requisitos de hardware

Los usuarios finales de la plataforma deberán contar como mínimo con:

RNF 5: Procesador Pentium II o superior.

RNF 6: 1GB de RAM, 20GB de HDD.

RNF 7: Si no cuentan con servidor local: conexión de banda ancha 256 bps como mínimo.

RNF 8: Si cuenta con servidor local: acceso a red interna.

Requisitos de software

RNF 9: Ordenadores con navegador Mozilla Firefox, Internet Explorer 7 o superior, Chrome, Opera o Safari.

RNF 10: Lenguaje de programación: PHP 5.2.4 o superior.

RNF 11: El marco de trabajo base de desarrollo que se utilizará es: Symfony 1.4.6

RNF 12: Como IDE se empleará NetBeans 6.9

RNF 13: Como servidor web se explotará Apache 2.2.16

RNF 14: El SGDB deberá ser PostgreSQL 8.4.

RNF 15: Sistema Operativo Windows y Linux.

RNF 16: *Plugin* Adobe Reader para el navegador web.

Requisitos de apariencia e interfaz externa

RNF 17: El diseño de las interfaces debe ser amigable y sencillo.

RNF 18: Claridad y buena organización de la información. Ejecución de acciones de una manera rápida y minimizar los pasos a dar en cada proceso.

Requisitos de portabilidad

RNF 19: El sistema podrá ser utilizado bajo cualquier Sistema Operativo y bajo los navegadores Mozilla Firefox, Internet Explorer 7 o superior, Chrome, Opera o Safari.

Requisitos de funcionalidad

RNF 20: Reducir al mínimo el tiempo en que carga la plataforma.

Requisitos de usabilidad

RNF 21: El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.

RNF 22: El sistema debe tener acceso al menú general desde cualquiera de sus páginas.

RNF 23: Se deben mostrar las rutas de acceso según la navegación que tenga el usuario.

RNF 24: Las rutas de acceso deben tener vínculos a las secciones que muestran.

Requisitos de rendimiento

RNF 25: Tiempos de respuestas rápidas al igual que la velocidad de procesamiento de la información.

2.5. Definición de los Casos de Uso del Sistema

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores, en respuesta a un evento que inicia un actor principal sobre el propio sistema, para conseguir un objetivo específico. (34)

Es posible afirmar que los casos de usos son los requisitos funcionales del sistema, agrupados según su funcionalidad, independientes o relacionados entre sí. Para la especificación de un caso de uso se parte necesariamente de los requisitos funcionales y se siguen tres pasos fundamentales que dan lugar a la correcta identificación de los mismos:

- ✓ Se define un caso de uso por requisito funcional.
- ✓ Se agrupan los casos de usos fuertemente relacionados.
- ✓ Se dividen los casos de usos sobrecargados o complejos.

2.6. Patrones utilizados en la creación de los CU

Para obtener una mejor estructuración de los casos de uso se hace necesaria la utilización de patrones, estos han demostrado ser de gran utilidad ante diversos problemas brindando las mejores soluciones. Existen diversos tipos de patrones de caso de uso, a continuación se describirán los que se encuentran reflejados en el diagrama de casos de uso del sistema.

- Patrón CRUD, está basado en la unión de casos de uso simples, formando una unidad conceptual. Este patrón está presente de dos maneras:
 - CRUD total (contiene todas las operaciones que pueden ser realizadas sobre una parte de la información, tales como creación, lectura, actualización y eliminación). Este patrón está reflejado en casos de uso como el Gestionar Mensajería Interna.
 - CRUD parcial (no contiene todas las operaciones que pueden ser realizadas sobre una parte de la información, debido a que no eran necesarias o eran muy complejas y no permitían el encapsulamiento). Este patrón está presente en casos de uso como el Gestionar Avisos.
- Múltiples actores, roles comunes (sucede cuando dos actores desempeñan el mismo rol sobre un CU, este rol es representado por otro actor resultante de la generalización de los actores que comparten este rol). Este patrón fue utilizado para unir las acciones comunes del Usuario de la Escuela y el Gestor de Avisos como Consultor de Avisos.

A continuación se listan los casos de uso del sistema:

CU1: Crear Foro

CU2: Eliminar Foro

CU3: Ver Foro

CU4: Gestionar Temas de Foro

CU5: Listar Temas de Foro

CU6: Gestionar Correspondencia Interna

CU7: Gestionar Eventos de la Agenda

CU8: Gestionar Anuncios

CU9: Ver Anuncios

CU10: Gestionar Avisos

CU11: Ver Aviso

CU12: Mostrar Agenda

CU13: Bloquear y desbloquear las comunicaciones

2.7. Descripción de los actores del sistema

Un actor del sistema es un rol que un usuario desempeña con respecto al sistema cuando interactúa con los casos de uso (35). Es importante destacar el uso de la palabra rol, pues

con esto se especifica que un actor no necesariamente representa a una persona en particular, sino más bien las funcionalidades que podrá realizar en el sistema.

Actor	Descripción
Usuario de la escuela	Usuario responsable de gestionar la mensajería interna, la agenda y puede ver avisos, anuncios y foros.
Gestor de avisos	Usuario responsable de crear, modificar o eliminar los avisos.
Consultor de avisos	Usuario con permisos para consultar los avisos creados.
Administrador central	Usuario responsable de crear, modificar o eliminar los anuncios.
Sistema	Responsable de crear y eliminar los foros.

Tabla 2.1: Descripción de los actores del sistema

2.8. Diagrama de casos de uso del sistema

En el diagrama de casos de uso se muestra la relación que existe entre los actores y de los mismos con los casos de uso del sistema. Los diferentes actores son los encargados de inicializar todas las acciones posibles dentro del sistema.

El administrador central es el actor con mayor responsabilidad ya que es el encargado no solo de inicializar el caso de uso gestionar anuncios, sino que puede comportarse como el actor usuario de la escuela, de este modo podrá inicializar casos de uso como: ver foro, gestionar temas de foro, ver anuncios, gestionar eventos, gestionar mensajería interna, bloquear o desbloquear las comunicaciones, mostrar agenda, listar temas de foro.

A partir de la necesidad de que los actores, usuario de la escuela y gestor de avisos estén relacionados con el caso de uso ver avisos, se crea el actor consultor de avisos el cual es una generalización de los anteriores. A su vez el gestor de avisos podrá inicializar el caso de uso gestionar avisos.

Por otra parte el actor sistema es el encargado de crear o eliminar los foros, estos se crean automáticamente cuando es asociado un programa de estudios a un grupo y a un

profesor y se eliminan de manera automática también cuando uno de los componentes de la creación desaparece.

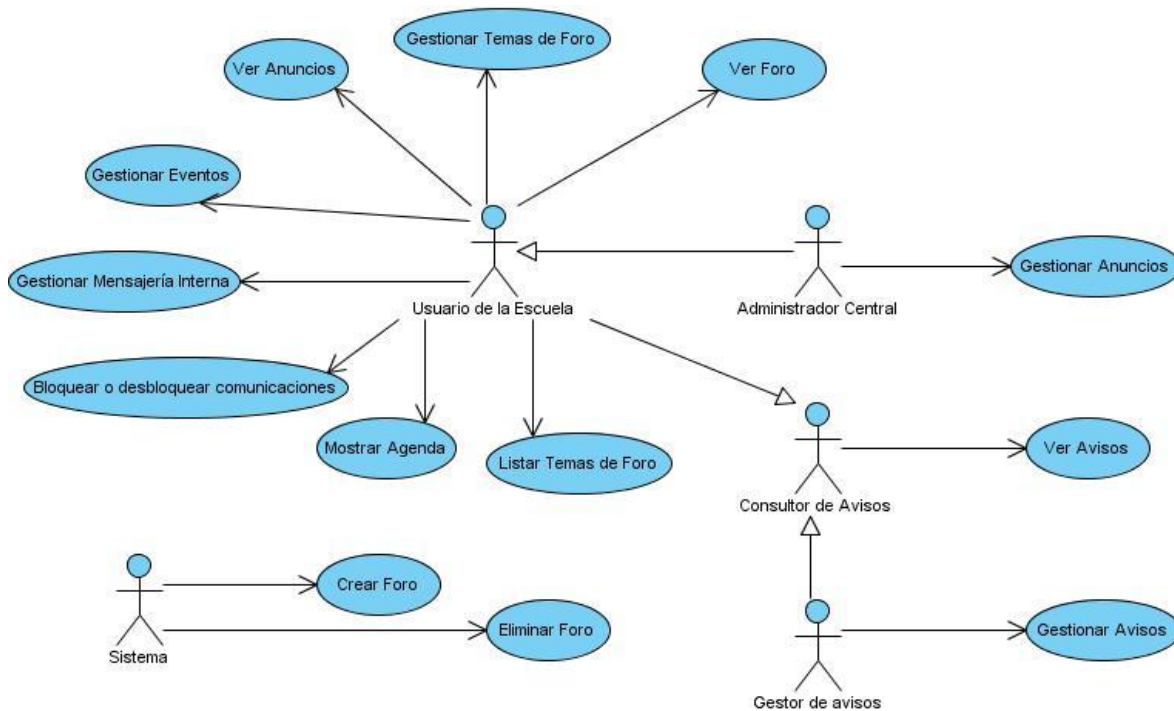


Figura 2.2: Diagrama de CUS

2.9. Descripción de los casos de uso del sistema

Se mostrara la descripción a nivel resumen de algunos casos de uso del sistema, con el objetivo de comprender como se van a desarrollar las funcionalidades expresadas en los requisitos funcionales.

Nombre	Gestionar Temas de Foro
Objetivo	Incluir, ver, modificar o eliminar Temas de Foro.
Actores	Usuario de la Escuela (Inicia): Incluye, ve, modifica o elimina Temas de Foros.
Complejidad	Media.
Nivel	Usuario.

Precondiciones	<p>Debe haberse generado el Escritorio de Trabajo del usuario autenticado.</p> <p>Para Modificar los datos de un Tema del Foro, debe estar seleccionado previamente por el actor.</p> <p>Para Ver los datos de un Tema del Foro, debe estar seleccionado previamente por el actor.</p> <p>Para Eliminar un tema del Foro, debe estar seleccionado previamente y el actor debe tener el permiso de eliminar el elemento.</p>
Poscondiciones	Se incluyó, vio, modificó o eliminó un Tema del Foro por el actor.
Resumen del Caso de Uso	
<p>El caso de uso se inicia cuando el Usuario de la Escuela decide realizar una acción sobre un tema del foro, esta acción puede ser la de ver, modificar o eliminar los datos de un tema de un foro. En caso de que desee modificar los datos de un tema, el actor selecciona el elemento a modificar, el sistema brindará la posibilidad de hacerle modificaciones a los campos existentes permitiéndole guardar los cambios realizados. Para eliminar el actor debe seleccionar el tema que desea eliminar. El sistema permite ver una vista previa del tema. Terminando el caso de uso.</p>	
Requisitos Funcionales	RF2: RF4.1, RF4.2, RF4.3, RF4.4

Tabla 2.2: Descripción a nivel resumen del CU Gestionar Temas de Foro.

Nombre	Gestionar Correspondencia Interna
Objetivo	Redactar, ver, modificar o eliminar correspondencia interna de la Plataforma.
Actores	Usuario de la Escuela (Inicia): Redacta, ve, modifica o elimina la Correspondencia Interna.
Complejidad	Alta.
Nivel	Usuario.
Precondiciones	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>Para redactar una Correspondencia Interna, el actor debe tener el</p>

	<p>permiso de crear el elemento.</p> <p>Para ver una Correspondencia Interna, debe estar seleccionado previamente, y el actor debe tener permiso de acceder al elemento.</p> <p>Para modificar una Correspondencia Interna, debe estar seleccionado previamente y el actor debe ser el responsable temporal del mismo.</p> <p>Para eliminar una Correspondencia Interna, debe estar seleccionado previamente y el actor debe tener el permiso de eliminar el elemento.</p>
Poscondiciones	Se redactó, vio, modificó o eliminó una Correspondencia Interna por el actor.
Resumen del Caso de Uso	
<p>El caso de uso se inicia cuando el Usuario de la Escuela decide realizar una acción sobre la correspondencia interna, esta acción puede ser la de ver los elementos que se encuentran en la bandeja de entrada, borradores, papelera o elementos enviados. En cada una de estas sesiones el sistema permitirá realizar acciones como la de eliminar, modificar una correspondencia que se encuentre en borradores, eliminar una o varias correspondencias, guardarlas en borradores, redactar una nueva. En caso de que desee redactar una nueva correspondencia, el sistema solicitará los datos necesarios, permitiendo enviar la correspondencia, guardarla en borradores o cancelar la operación. Para eliminar una o varias correspondencias el actor debe seleccionar la o las correspondencias que desea eliminar. Terminando el caso de uso.</p>	
Requisitos Funcionales	RF6: RF6.1, RF6.2, RF6.3, RF6.4, RF6.5, RF6.6, RF6.7, RF6.8, RF6.9

Tabla 2.3: Descripción a nivel resumen del CU Gestionar Correspondencia Interna.

Nombre	Gestionar Eventos de la Agenda
Objetivo	Crear, ver, modificar o eliminar un evento en la Agenda.
Actores	Usuario de la Escuela (Inicia): Redacta, ve, modifica o elimina los eventos de la Agenda.
Complejidad	Media.
Nivel	Usuario.

Precondiciones	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>Para crear un evento en la agenda, el actor debe tener el permiso de crear el elemento y debe estar seleccionado el día que se desea crear el evento.</p> <p>Para ver un evento, debe estar seleccionado previamente, y el actor debe tener permiso de acceder al elemento.</p> <p>Para modificar un evento, debe estar seleccionado previamente y el actor debe ser el responsable temporal del mismo.</p> <p>Para eliminar un evento, debe estar seleccionado previamente y el actor debe tener el permiso de eliminar el elemento.</p>
Poscondiciones	Se redactó, vio, modificó o eliminó un evento en la agenda.
Resumen del Caso de Uso	
<p>El caso de uso se inicia cuando el Usuario de la Escuela decide realizar una acción sobre los eventos de la agenda, esta acción puede ser la de ver los eventos que hay creados en la agenda. En cada una de estas sesiones el sistema permitirá realizar acciones como la de eliminar, modificar un evento que se encuentre en la agenda o crear uno nuevo. En caso de que desee crear un nuevo evento, el sistema solicitará los datos necesarios, permitiendo crear el mismo, o cancelar la operación. Para eliminar un evento el actor debe seleccionar el evento que desea eliminar. Terminando el caso de uso.</p>	
Requisitos Funcionales	RF7: RF7.1, RF7.2, RF7.3, RF7.4

Tabla 2.4: Descripción a nivel resumen del CU Gestionar Eventos de la Agenda.

Nombre	Gestionar Anuncios
Objetivo	Incluir, modificar o eliminar anuncios.
Actores	Administrador Central (Inicia): Incluye, modifica o elimina Anuncios.
Complejidad	Media.
Nivel	Usuario.
Precondiciones	<p>Debe haberse generado el Escritorio de Trabajo del usuario autenticado.</p> <p>Para Modificar los datos de un anuncio, debe estar seleccionado</p>

	<p>previamente por el actor.</p> <p>Para Eliminar un anuncio, debe estar seleccionado previamente y el actor debe tener el permiso de eliminar el elemento.</p>
Poscondiciones	Se incluyó, modificó o eliminó un anuncio por el actor.
Resumen del Caso de Uso	
<p>El caso de uso se inicia cuando el Administrador Central decide realizar una acción sobre un anuncio, esta acción puede ser la de incluir, modificar los datos de un anuncio o eliminar el mismo. Si el usuario selecciona la opción de incluir un nuevo anuncio el sistema le brindará la posibilidad de incluir los datos necesarios y le permitirá cancelar la operación en cualquier momento o crear dicho anuncio. En caso de que desee modificar los datos de un anuncio, el actor debe tener previamente seleccionado el elemento a modificar, el sistema brindará la posibilidad de realizarle modificaciones a los campos existentes permitiéndole guardar los cambios realizados. Para eliminar un anuncio el actor debe tener previamente seleccionado el mismo que desea eliminar. Terminando el caso de uso.</p>	
Requisitos Funcionales	RF8: RF8.1, RF8.2, RF8.3

Tabla 2.5: Descripción a nivel resumen del CU Gestionar Anuncios.

Para consultar el resto de las descripciones a nivel resumen de los casos de uso y de esta manera lograr comprender mejor la propuesta de solución; ver anexos del 1 al 8.

2.10. Conclusiones

A partir de la descripción de los procesos del negocio se obtuvo una visión más amplia sobre el entorno al cual se integrará el sistema, además se describió la propuesta de solución del mismo. Esto posibilitó la creación de un conjunto de artefactos que conformarán la base del futuro sistema.

Capítulo 3

Análisis y Diseño

3.1. Introducción

Luego de una correcta captura de requisitos funcionales, no funcionales y la identificación de los diferentes casos de uso, la metodología RUP propone el Análisis y Diseño de la propuesta de solución. En este flujo de trabajo se desarrollan artefactos como los diagramas de clases del análisis, del diseño y los diagramas de secuencia del diseño entre otros, los cuales contribuirán a la futura implementación del sistema.

3.2. Modelo de Análisis

En el análisis se refinan y estructuran los requisitos obtenidos con anterioridad, de manera que sea más fácil su comprensión, su preparación y modificación. En el Modelo de Análisis se identifican clases y relaciones. Tiene como objetivo refinar los casos de uso de forma más detallada. Además, en el mismo están contenidas las clases de análisis, las cuales se centran en los requisitos funcionales. Estas clases se clasifican en tres tipos:

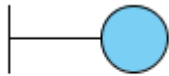


Nombre	Características	Estereotipo
Interfaz	Modela la interacción entre el sistema y sus actores.	 CI_Ver Temas de Foro
Control	Coordina la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.	 CC_Gestionar Temas de Foro
Entidad	Modela información que posee larga vida.	 CE_Temas de Foro

Tabla 3.1: Clases del Análisis.

3.2.1. Diagramas de clases del Análisis

A continuación se mostraran algunos diagramas de clases del análisis y la relación que existe entre los actores y los diferentes tipos de clases del análisis.

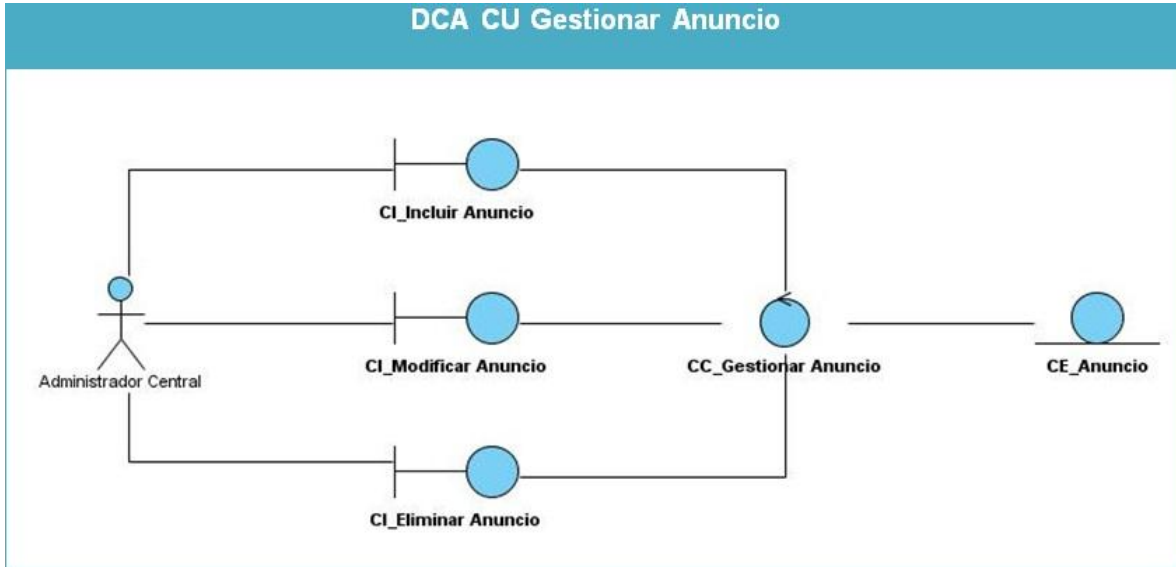


Figura 3.1: DCA CU Gestionar Anuncio.

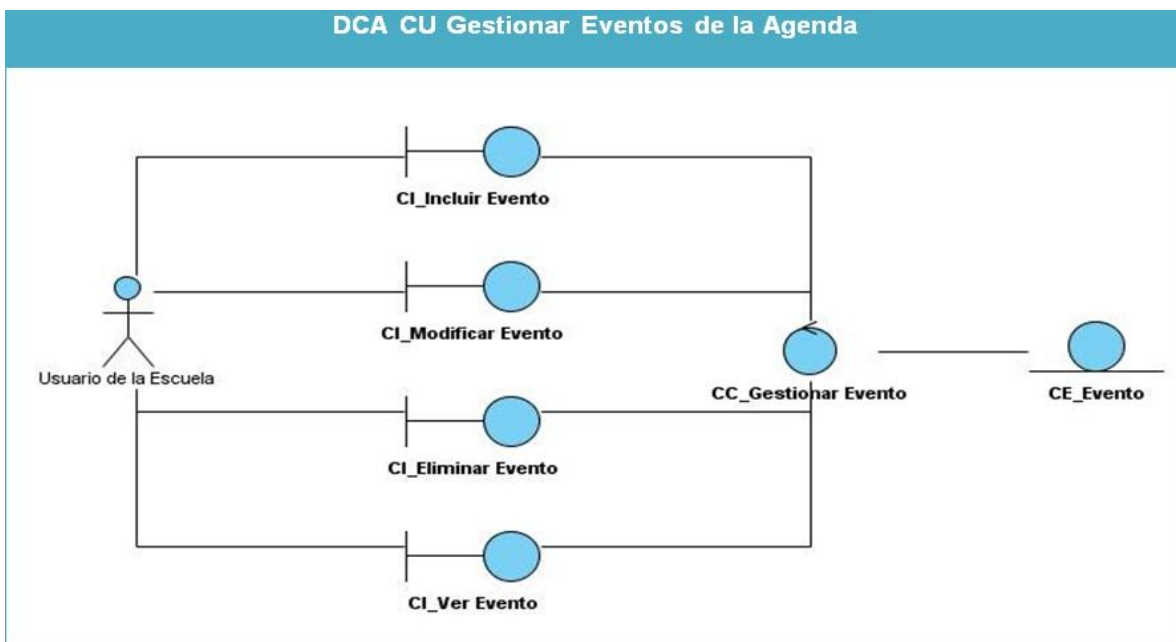


Figura 3.2: DCA CU Gestionar Eventos de la Agenda.

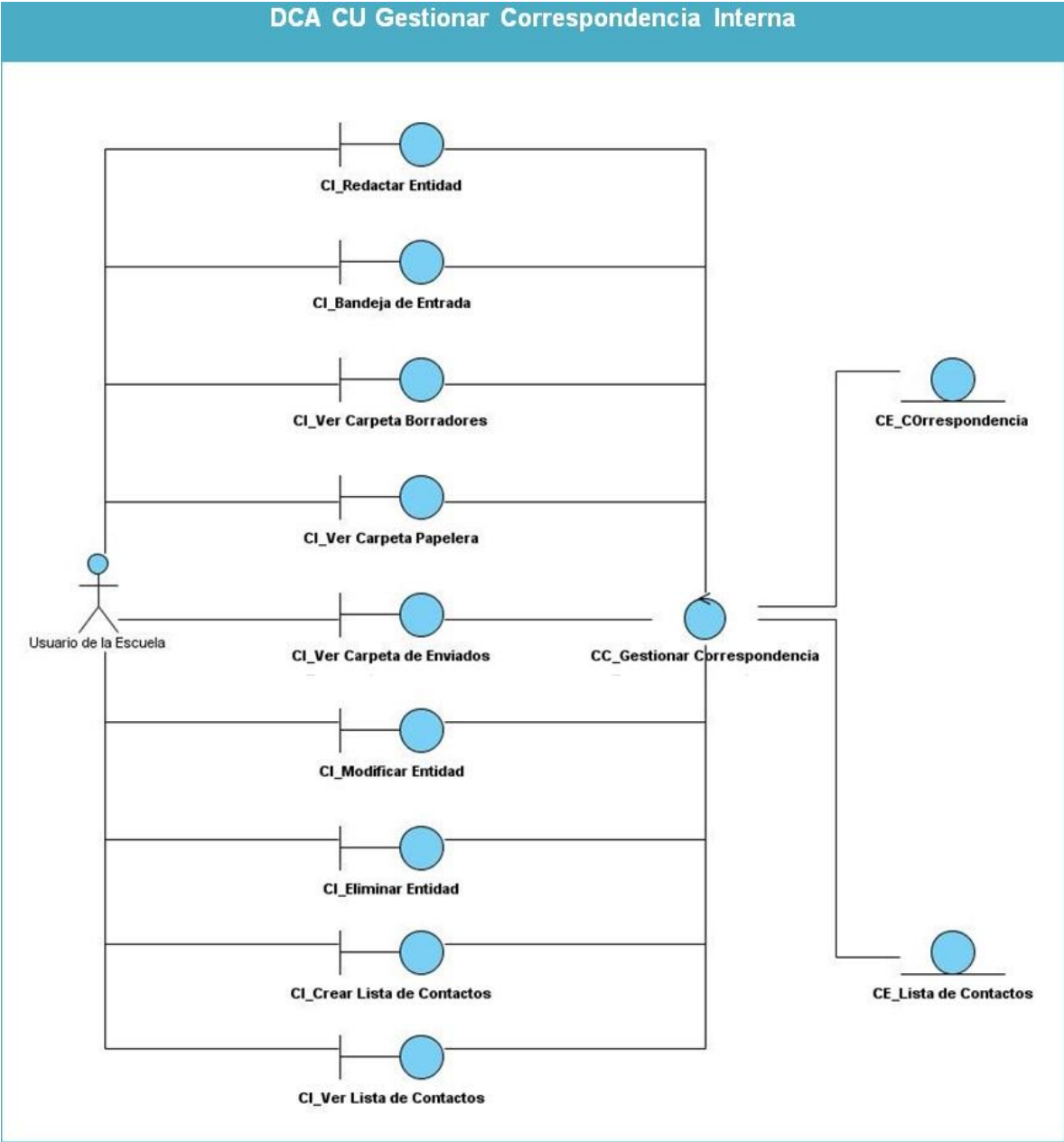


Figura 3.3: DCA CU Gestionar Correspondencia Interna.

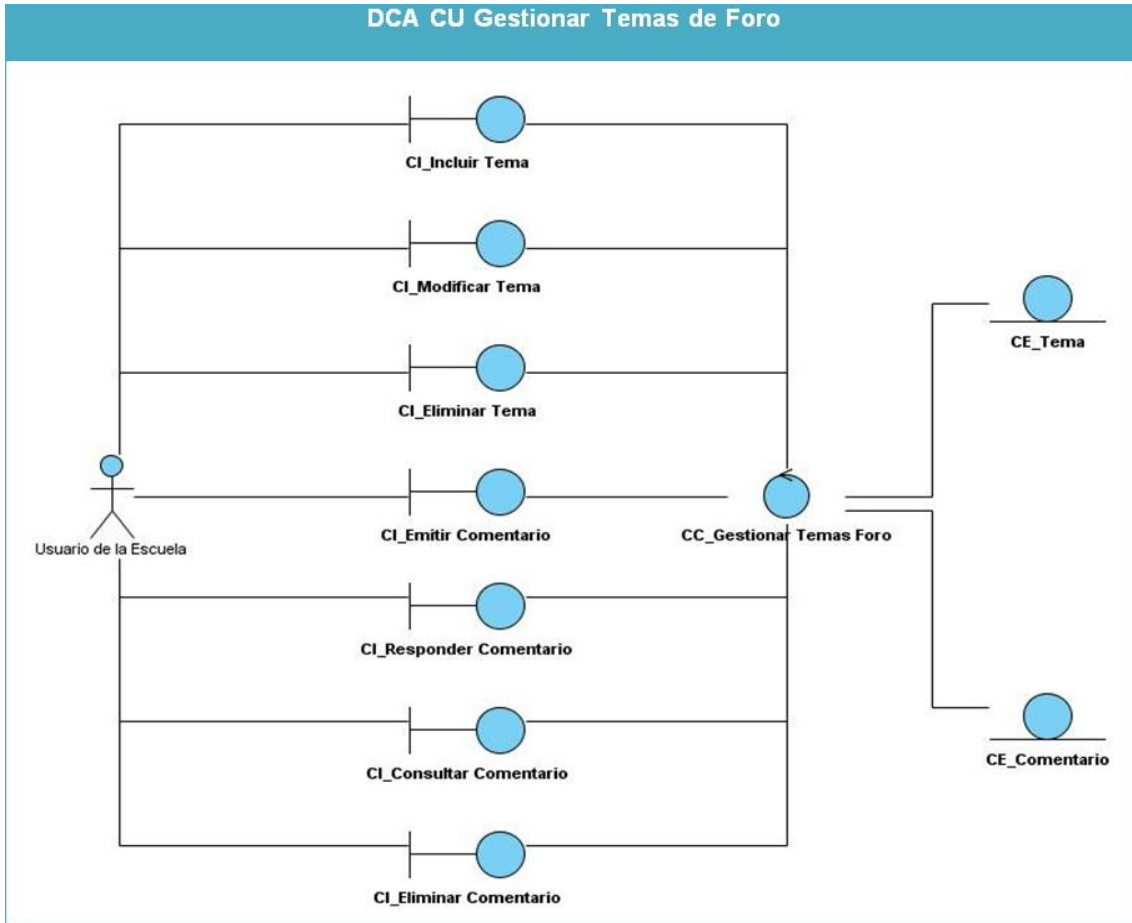


Figura 3.4: DCA CU Gestionar Temas de Foro.

Para consultar el resto de los diagramas de clases del análisis y lograr una mayor comprensión del sistema y de las relaciones existentes entre las diferentes clases que lo componen; ver anexos del 9 al 13.

3.3. Modelo del Diseño

El diseño es un refinamiento del análisis, el cual proporciona una completa idea de lo que es el software, enfocado en el comportamiento del sistema desde el punto de vista de la implementación.

3.3.1. Patrones utilizados

Un patrón es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema general en un contexto particular. Es una manera más práctica de describir ciertos aspectos de la organización de un programa.

El framework Symfony está basado en un patrón clásico del diseño web conocido como arquitectura Modelo – Vista – Controlador (MVC), que está formado por tres niveles:

- ✓ El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- ✓ La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- ✓ El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (36)

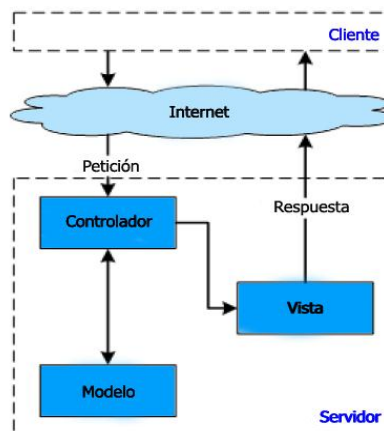


Figura 3.5: Patrón Modelo – Vista – Controlador

Al estar presente este patrón en todos los submódulos permite una mejor organización de los mismos, por ejemplo en el submódulo foro el controlador `forum_actions` posee diferentes funciones que permiten crear, modificar, eliminar o mostrar los temas de foro, entre otras. Para cada una de estas funciones existe una vista asociada, la vista es la encargada de interactuar con el usuario, mostrar los resultados de las acciones del

controlador y de solicitar la información necesaria. Un ejemplo de estas vistas es crear_foro_success.php. Por su parte el modelo es el encargado de recoger los datos, en el caso de tb_forum se encuentran todos los atributos y métodos de la tabla foro, estos métodos son los encargados de realizar las llamadas a las consultas presentes en tb_forum_table.

3.3.2. Descripción de las clases del Diseño

A continuación se describirán las diferentes clases del diseño, las mismas poseerán un identificador delante del nombre seguido de un guion bajo y el resto del nombre. Ejemplo de estas clases es: **SP_AdvertisementActions**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador del aviso
notice_name	varchar(128)	Nombre del aviso
notice_body	varchar(1000)	Contenido del aviso
user_id	varchar(128)	Identificador del usuario
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador del aviso
GetNoticeName()	varchar(128)	Devuelve el nombre del aviso
SetNoticeName(varchar notice_name)	void	Permite modificar el nombre actual del aviso
GetNoticeBody()	varchar(1000)	Devuelve el contenido del aviso
SetNoticeBody(varchar notice_body)	void	Permite modificar el contenido actual del aviso
GetUserId()	varchar(128)	Devuelve el identificador del usuario
SetUserId(varchar user_id)	void	Permite modificar el identificador actual del usuario

Tabla 3.2: Descripción de la clase **tb_notice**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador de la relación
notice_id	varchar(128)	Identificador del aviso

Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador de la relación
GetNoticeId()	varchar(128)	Devuelve el identificador del aviso
SetNoticeId(varchar notice_id)	void	Permite modificar el identificador actual del aviso
GetUserId()	varchar(128)	Devuelve el identificador del usuario
SetUserId(varchar user_id)	void	Permite modificar el identificador actual del usuario

Tabla 3.3: Descripción de la clase `r_notice_user`.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador del usuario
username	varchar(128)	Nombre del usuario
password	varchar(128)	Contraseña
is_active	boolean	Si se encuentra o no activo
last_login	timestamp/date	Ultima fecha en que entró al sistema
first_name	varchar(30)	Primer nombre
second_name	varchar(30)	Segundo nombre
first_last_name	varchar(30)	Primer apellido
second_last_name	varchar(30)	Segundo apellido
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador del usuario
GetUsername()	varchar(128)	Devuelve el nombre del usuario
SetUsername(varchar username)	void	Permite modificar el nombre actual del usuario
GetIsActive()	boolean	Devuelve si el usuario se encuentra activo
SetIsActive (boolean is_active)	void	Permite modificar el estado activo o no del usuario
GetLastLogin()	timestamp/date	Devuelve la última fecha en que el usuario entró al sistema

SetLastLogin(timestamp/date last_login)	void	Permite modificar la última fecha en que el usuario entró al sistema
GetFirstName()	varchar(30)	Devuelve el primer nombre del usuario
SetFirstName(varchar first_name)	void	Permite modificar el primer nombre del usuario
GetSecondName()	varchar(30)	Devuelve el segundo nombre del usuario
SetSecondName(varchar second_name)	void	Permite modificar el segundo nombre del usuario
GetFirstLastName()	varchar(30)	Devuelve el primer apellido del usuario
SetFirstLastName(varchar first_last_name)	void	Permite modificar el primer apellido del usuario
GetSecondLastName()	varchar(30)	Devuelve el segundo apellido del usuario
SetSecondLastName(varchar second_last_name)	void	Permite modificar el segundo apellido del usuario

Tabla 3.4: Descripción de la clase **sf_guard_user**.

Método	Descripción
executeIndex(sfWebRequest \$request)	Gestiona los datos para la interfaz principal.
executeShow(sfWebRequest \$request)	Muestra los datos.
executeNew(sfWebRequest \$request)	Gestiona los datos para Incluir Aviso.
executeEdit(sfWebRequest \$request)	Gestiona los datos para Modificar Aviso.
executeDelete(sfWebRequest \$request)	Gestiona los datos para Eliminar Aviso.

Tabla 3.5: Descripción de la clase **SP_noticesActions**.

Método	Descripción
executeIndex(sfWebRequest \$request)	Gestiona los datos para la interfaz principal.
executeShow(sfWebRequest \$request)	Muestra los datos.
executeNew(sfWebRequest \$request)	Gestiona los datos para Incluir Anuncio.
executeEdit(sfWebRequest \$request)	Gestiona los datos para Modificar Anuncio.
executeDelete(sfWebRequest \$request)	Gestiona los datos para Eliminar Anuncio.

Tabla 3.6: Descripción de la clase **SP_AdvertisementActions**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador del anuncio
body	varchar(250)	Contenido del anuncio
title	varchar(128)	Título del anuncio
media	image/long binary	Imagen correspondiente al anuncio
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador del anuncio
GetBody()	varchar(250)	Devuelve el contenido del anuncio
SetBody(varchar body)	void	Permite modificar el contenido actual del anuncio
GetTitle()	varchar(128)	Devuelve el título del anuncio
SetTitle(varchar title)	void	Permite modificar el título actual del aviso
GetMedia()	image/long binary	Devuelve la imagen correspondiente al anuncio
SetMedia(image/long binary media)	void	Permite modificar la imagen correspondiente al anuncio

Tabla 3.7: Descripción de la clase **tb_advertisement**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador del foro
name	varchar(128)	Nombre del foro
description	varchar(250)	Descripción del foro
active	boolean	Si se encuentra o no activo
quantity_of_topics	numeric(10,0)	Cantidad de temas del foro
quantity_of_messages	numeric(10,0)	Cantidad de mensajes del foro
study_program_id	varchar(128)	Identificador del programa de estudio
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador del foro
GetName()	varchar(128)	Devuelve el nombre del foro
SetName(varchar name)	void	Permite modificar el nombre actual del foro
GetDescription()	varchar(250)	Devuelve la descripción del foro

SetDescription(varchar description)	void	Permite modificar la descripción del foro
GetActive()	boolean	Devuelve si el foro se encuentra activo
SetActive(boolean active)	void	Permite modificar si el foro se encuentra activo
GetQuantityOfTopics()	numeric(10,0)	Devuelve la cantidad de temas del foro
SetQuantityOfTopics(numeric quantity_of_topics)	void	Permite modificar la cantidad de temas del foro
GetQuantityOfMessages()	numeric(10,0)	Devuelve la cantidad de mensajes del foro
SetQuantityOfMessages(numeric quantity_of_messages)	void	Permite modificar la cantidad de mensajes del foro
GetStudyProgramId()	varchar(128)	Devuelve el identificador del programa de estudio
SetStudyProgramId(varchar study_program_id)	void	Permite modificar el identificador del programa de estudio

Tabla 3.8: Descripción de la clase **tb_forum**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador del tema de foro
title	varchar(128)	Título del tema de foro
active	boolean	Si se encuentra o no activo
quantity_of_messages	numeric(10,0)	Cantidad de mensajes del tema
quantity_of_visits	numeric(10,0)	Cantidad de visitas
forum_id	varchar(128)	Identificador del foro
user_id	varchar(128)	Identificador del usuario
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador del tema
GetTitle()	varchar(128)	Devuelve el título del tema
SetTitle(varchar title)	void	Permite modificar el título del tema
GetActive()	boolean	Devuelve si se encuentra o no activo
SetActive(boolean active)	void	Permite modificar si el tema se encuentra activo

GetQuantityOfMessages()	numeric(10,0)	Devuelve la cantidad de mensajes del tema
SetQuantityOfMessages(numeric quantity_of_messages)	void	Permite modificar la cantidad de mensajes del tema
GetQuantityOfVisits()	numeric(10,0)	Devuelve la cantidad de visitas al tema
SetQuantityOfVisits(numeric quantity_of_topics)	void	Permite modificar la cantidad de visitas al tema
GetForumId()	varchar(128)	Devuelve el identificador del foro
SetForumId(varchar forum_id)	void	Permite modificar el identificador del foro
GetUserId()	varchar(128)	Devuelve el identificador del usuario
SetUserId(varchar user_id)	void	Permite modificar el identificador del usuario

Tabla 3.9: Descripción de la clase **tb_topic**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador de un mensaje del tema
title	varchar(128)	Título de un mensaje del tema
body	varchar(250)	Descripción de un mensaje del tema
topic_id	varchar(128)	Identificador del tema
user_id	varchar(128)	Identificador del usuario
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador de un mensaje del tema
GetTitle()	varchar(128)	Devuelve el título de un mensaje del tema
SetTitle(varchar title)	void	Permite modificar el identificador de un mensaje del tema
GetBody()	varchar(250)	Devuelve la descripción de un mensaje del tema
SetBody(varchar body)	void	Permite modificar la descripción de un mensaje del tema
GetTopicId()	varchar(128)	Devuelve el identificador del tema
SetTopicId(varchar topic_id)	void	Permite modificar el identificador del tema

GetUserId()	varchar(128)	Devuelve el identificador del usuario
SetUserId(varchar user_id)	void	Permite modificar el identificador del usuario

Tabla 3.10: Descripción de la clase **tb_message_topic**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador del programa de estudio
matter_id	varchar(128)	Identificador de la materia
name	varchar(250)	Nombre del programa de estudio
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador del programa de estudio
GetMatterId()	varchar(128)	Devuelve el identificador de la materia
SetMatterId(varchar title)	void	Permite modificar el identificador de la materia
GetName()	varchar(250)	Devuelve el nombre del programa de estudio
SetName(varchar name)	void	Permite modificar el nombre del programa de estudio

Tabla 3.11: Descripción de la clase **tb_study_program**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador del grupo
name	varchar(250)	Nombre del grupo
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador del grupo
GetName()	varchar(250)	Devuelve el nombre del grupo
SetName(varchar name)	void	Permite modificar el nombre del grupo

Tabla 3.12: Descripción de la clase **tb_group**.

Método	Descripción
executeCrearTema(sfWebRequest \$request)	Gestiona los datos para Crear Tema.
executeModificarTema(sfWebRequest \$request)	Gestiona los datos para Modificar Tema.
executeCrearMensaje(sfWebRequest \$request)	Gestiona los datos para Crear Mensaje.
executeModificarMensaje(sfWebRequest \$request)	Gestiona los datos para Modificar Mensaje.
executeEliminarTema(sfWebRequest \$request)	Gestiona los datos para Eliminar Tema.
executeEliminarMensaje(sfWebRequest \$request)	Gestiona los datos para Eliminar Mensaje.
executeForo (sfWebRequest \$request)	Muestra los datos del Foro.
executeTema(sfWebRequest \$request)	Muestra los datos del Tema.
executeMensaje (sfWebRequest \$request)	Muestra los datos del Mensaje.

Tabla 3.13: Descripción de la clase **SP_ForumActions**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador del evento de la agenda
subject	varchar(128)	Asunto del evento
location	varchar(128)	Lugar del evento
event_type	varchar(128)	Tipo de evento
reminder	numeric(10,0)	Recordatorio del evento
description	varchar(250)	Descripción del evento
start_time	timestamp/date	Fecha de comienzo del evento
end_time	timestamp/date	Fecha final del evento
user_id	varchar(128)	Identificador del usuario
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador del evento
GetSubject()	varchar(128)	Devuelve el asunto del evento
SetSubject(varchar subject)	void	Permite modificar el asunto del evento
GetLocation()	varchar(128)	Devuelve el lugar del evento
SetLocation(varchar location)	void	Permite modificar el lugar del evento
GetEventType()	varchar(128)	Devuelve el tipo de evento
SetEventType(varchar event_type)	void	Permite modificar el tipo de evento
GetReminder()	numeric(10,0)	Devuelve el recordatorio del evento

SetRemider(numeric reminder)	void	Permite modificar el recordatorio del evento
GetDescription()	varchar(250)	Devuelve la descripción del evento
SetDescription(varchar description)	void	Permite modificar la descripción del evento
GetStartTime()	timestamp/date	Devuelve la fecha de comienzo del evento
SetStartTime(timestamp/date start_time)	void	Permite modificar la fecha de comienzo del evento
GetEndTime()	timestamp/date	Devuelve la fecha final del evento
SetEndTime(timestamp/date end_time)	void	Permite modificar la fecha final del evento
GetUserId()	varchar(128)	Devuelve el identificador del usuario
SetUserId(varchar user_id)	void	Permite modificar el identificador del usuario

Tabla 3.14: Descripción de la clase **sf_calendar**.

Método	Descripción
executeAddDetail(sfWebRequest \$request)	Gestiona los datos para Crear Evento.
executeDelete(sfWebRequest \$request)	Gestiona los datos para Eliminar Evento.
executeEdit(sfWebRequest \$request)	Gestiona los datos para Modificar Evento.
executeIndex(sfWebRequest \$request)	Gestiona los datos de la Agenda.

Tabla 3.15: Descripción de la clase **SP_sfCalendarActions**.

Método	Descripción
executeCreate(sfWebRequest \$request)	Gestiona los datos para Crear Correspondencia.
executeDraft(sfWebRequest \$request)	Gestiona los datos de la carpeta Borradores.
executeListDistributionList(sfWebRequest \$request)	Mostrar las listas de distribución.
executeEdit(sfWebRequest \$request)	Gestiona los datos para Modificar Correspondencia.
executeIndex(sfWebRequest \$request)	Gestiona los datos de la Correspondencia Interna.

executeNewDistributionList(sfWebRequest \$request)	Gestiona los datos para Crear Lista de Distribución.
executeNotification(sfWebRequest \$request)	Gestiona las Notificaciones.
executeTrash(sfWebRequest \$request)	Gestiona los datos de la carpeta Papelera.
executeResponse(sfWebRequest \$request)	Gestiona los datos para Responder Correspondencia.
executeSend(sfWebRequest \$request)	Gestiona los datos para Enviar Correspondencia.
executeShow(sfWebRequest \$request)	Muestra los datos de la Correspondencia.

Tabla 3.16: Descripción de la clase **SP_MessagingActions**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador del mensaje
subject	varchar(128)	Asunto del mensaje
body	varchar(1000)	Contenido del mensaje
send_to	varchar(128)	Destinatario del mensaje
send_date	timestamp/date	Fecha de envío del mensaje
state_id	varchar(128)	Identificador de estado
user_id	varchar(128)	Identificador del usuario
message_type_id	varchar(128)	Identificador de tipo de mensaje
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador del mensaje
GetSubject()	varchar(128)	Devuelve el asunto del mensaje
SetSubject(varchar subject)	void	Permite modificar el asunto del mensaje
GetBody()	varchar(1000)	Devuelve el contenido del mensaje
SetBody(varchar body)	void	Permite modificar el contenido del mensaje
GetSendTo()	varchar(128)	Devuelve el destinatario del mensaje
SetSendTo(varchar send_to)	void	Permite modificar el destinatario del mensaje
GetSendDate()	timestamp/date	Devuelve la fecha de envío del mensaje
SetSendDate(timestamp/date)	void	Permite modificar la fecha de envío del

send_date)		mensaje
GetStateId()	varchar(128)	Devuelve el identificador de estado
SetStateId(varchar state_id)	void	Permite modificar el identificador de estado
GetUserId()	varchar(128)	Devuelve el identificador del usuario
SetUserId(varchar user_id)	void	Permite modificar el identificador del usuario
GetMessageTypeId()	varchar(128)	Devuelve el identificador del tipo de mensaje
SetMessageTypeId(varchar message_type_id)	void	Permite modificar el identificador del tipo de mensaje

Tabla 3.17: Descripción de la clase **tb_message**.

Atributo	Tipo	Descripción
id	varchar(128)	Identificador de la relación
message_id	varchar(128)	Identificador del mensaje
user_id	varchar(128)	Identificador del usuario
Método	Retorno	Descripción
GetId()	varchar(128)	Devuelve el identificador de la relación
GetMessageId()	varchar(128)	Devuelve el del mensaje
SetMessageId(varchar message_id)	void	Permite modificar el identificador del mensaje
GetUserId()	varchar(128)	Devuelve el identificador del usuario
SetUserId(varchar user_id)	void	Permite modificar el identificador del usuario

Tabla 3.18: Descripción de la clase **r_guard_user_message**.

3.3.3. Diagramas de clases del Diseño

A continuación se muestran algunos diagramas de clases del diseño, de cada uno se describieron sus clases en el epígrafe anterior.

En el siguiente Diagrama de Clase del Diseño (DCD) se muestra la relación de la clase controladora SP_AdvertisementActions con las páginas clientes y de cómo estas le envían

la información a través de los formularios, como el FR_CrearAnuncios. También muestra que las páginas controladoras hacen función de intermediarias entre la vista y el modelo para poder acceder a la entidad CE_Anuncio.

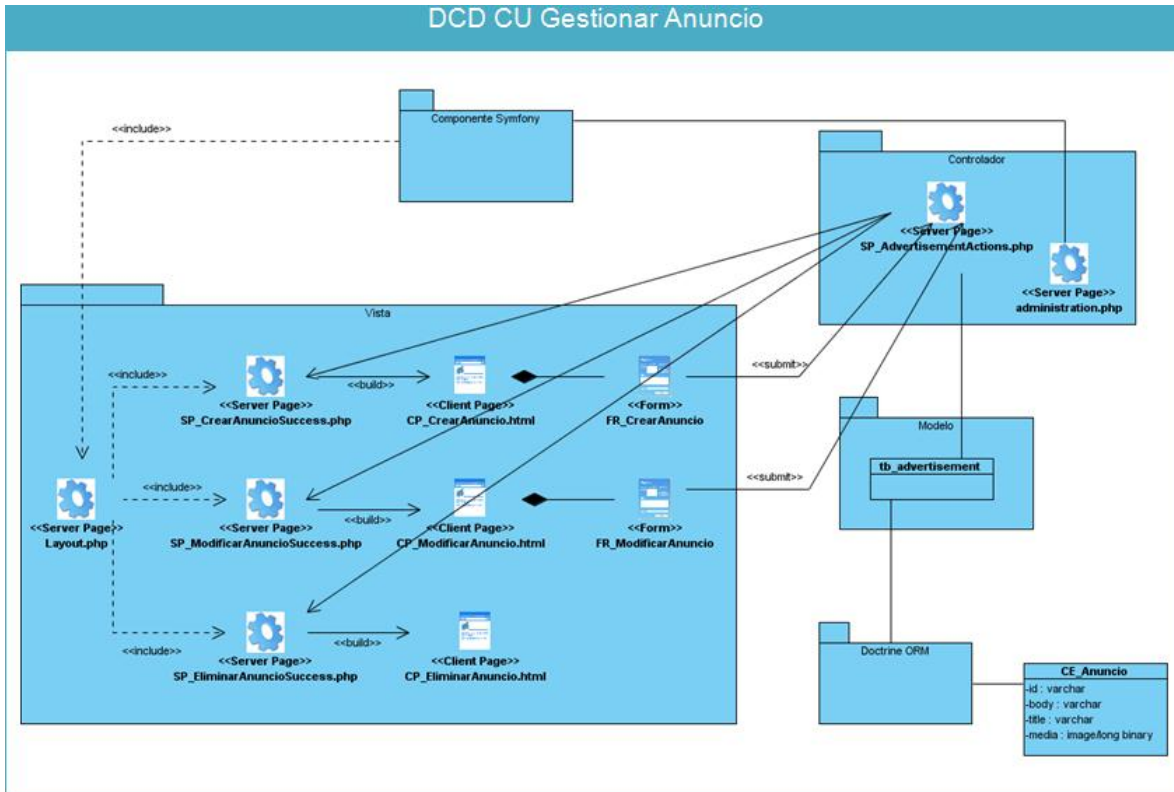


Figura 3.6: DCD CU Gestionar Anuncio.

En el DCD correspondiente a Gestionar Temas de Foro se muestra la relación entre la clase controladora `SP_ForumActions` y las páginas clientes `CP_IncluirTema`, `CP_ModificarTema`, entre otras, las cuales a través de los formularios envían los datos a las paginas servidoras y estas acceden a las tablas de la base de datos para actualizar, consultar o guardar información.

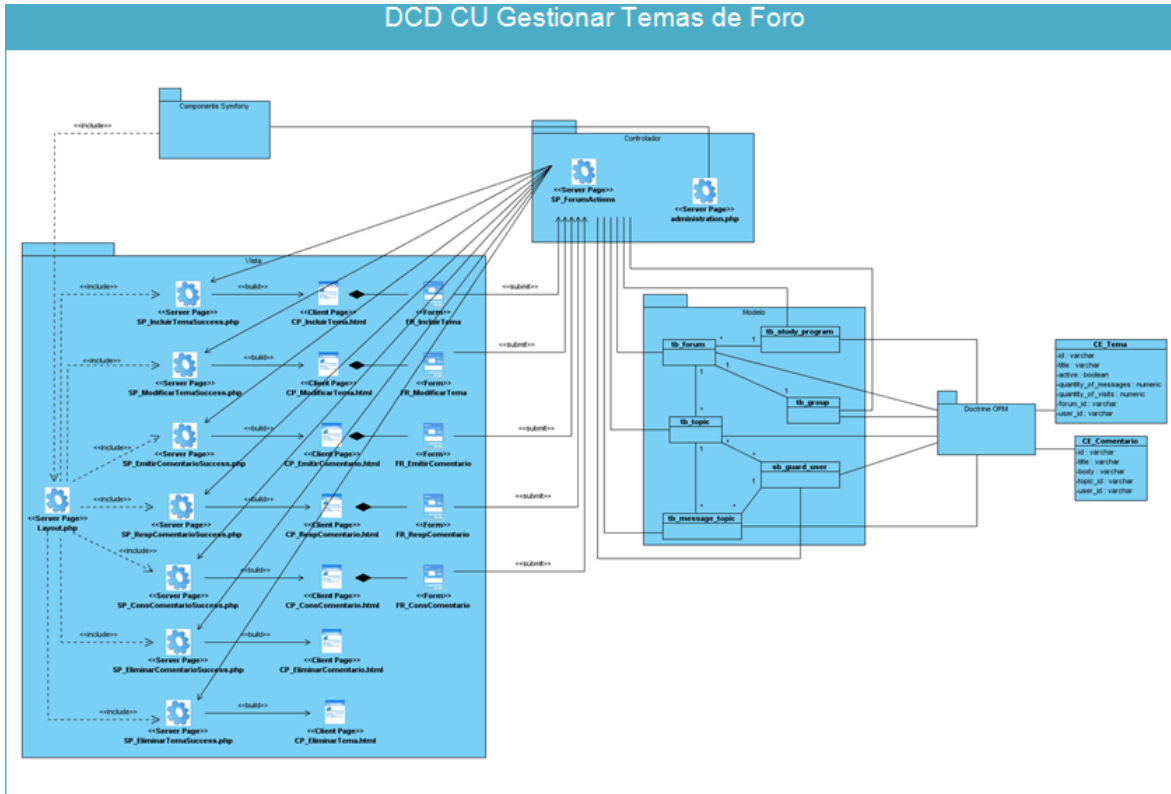


Figura 3.7: DCD CU Gestionar Temas de Foro.

En el DCD siguiente se puede observar la interacción de las clases servidoras del paquete controlador con los componentes Symfony, también con el modelo para acceder a la clase entidad `CE_Evento`. Además la página servidora `SP_sfCalendarActions` se relaciona con la vista, para brindar y obtener información mediante las páginas clientes.

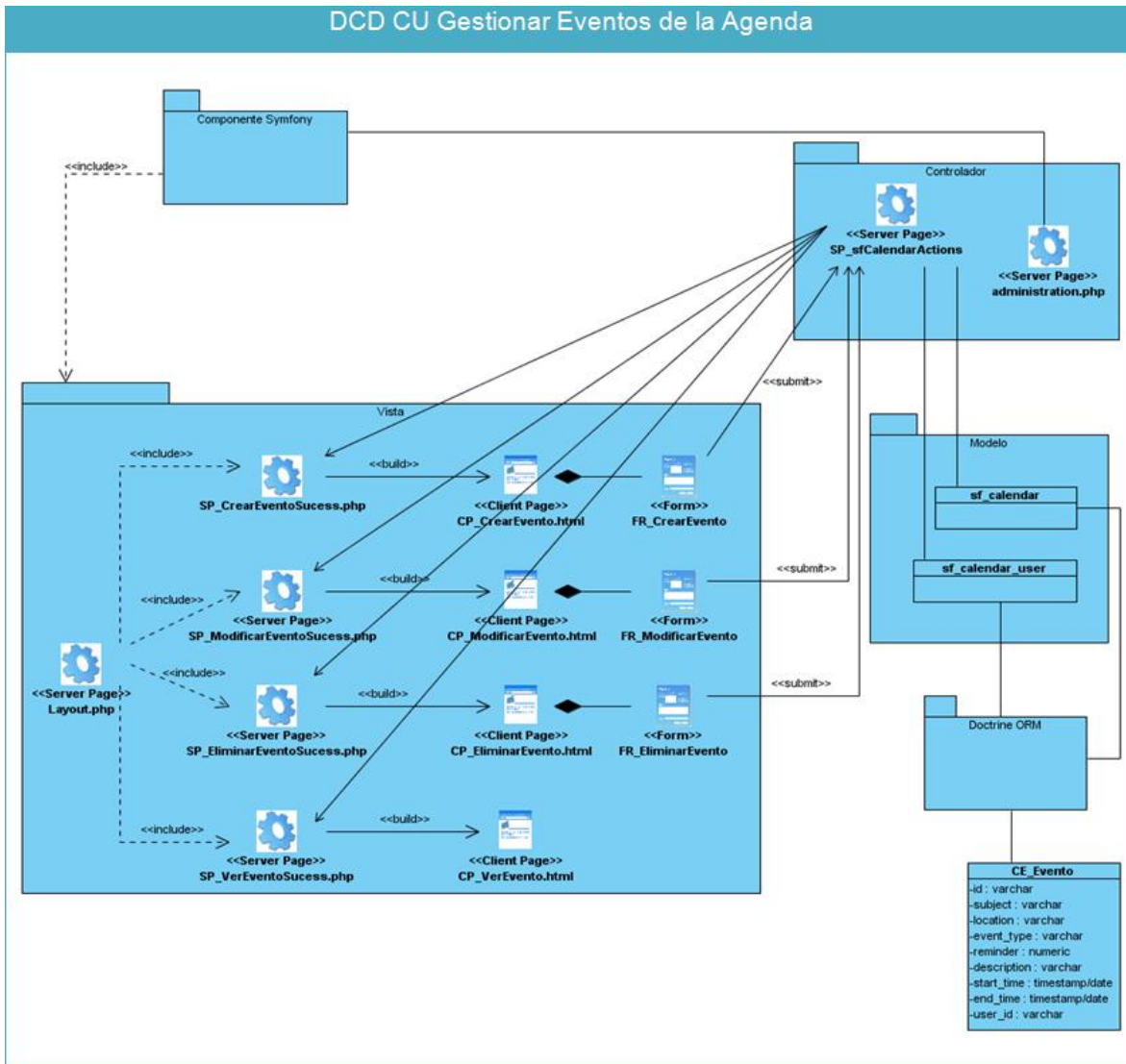


Figura 3.8: DCD CU Gestionar Eventos de la Agenda.

En el DCD Gestionar Mensajería Interna se puede observar cómo se encuentran separados los componentes por paquetes, en la vista, el modelo y el controlador, de esta manera permite a las clases controladoras acceder a las tablas de la base de datos para buscar, actualizar o guardar información en las clases entidad CE_Correspondencia y CE_ListaContacto, a su vez interactuar con el usuario mediante las páginas clientes de la vista como son CP_RedactarEntidad, CP_CrearListCont, entre otras.

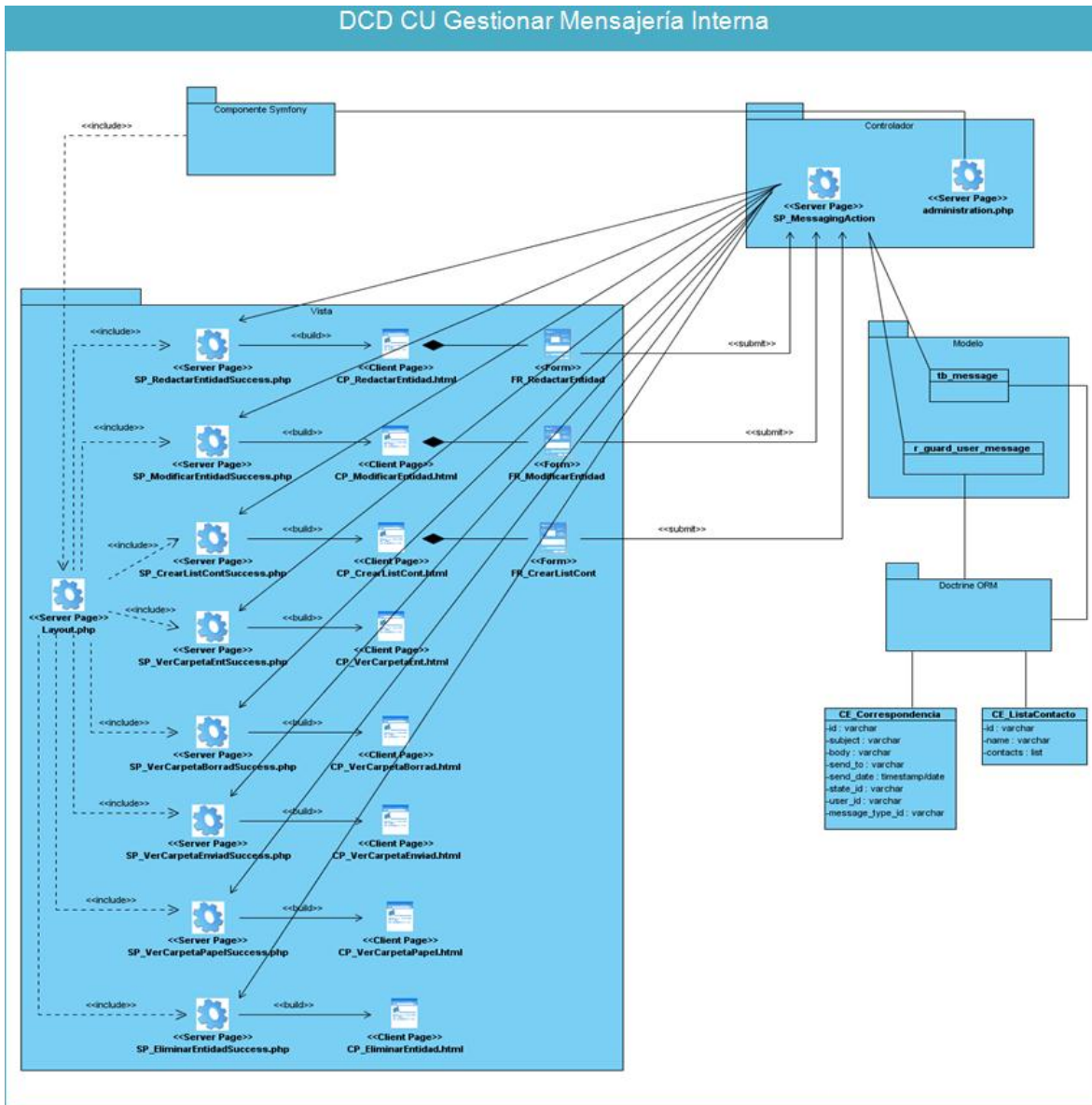


Figura 3.9: DCD CU Gestionar Mensajería Interna.

Para consultar el resto de los diagramas de clases del diseño y lograr una mayor comprensión del diseño del sistema y de las relaciones existentes entre las diferentes clases que lo componen; ver anexos del 14 al 18.

3.3.4. Diagramas de secuencia del Diseño

A continuación se muestran un diagrama de secuencia del diseño, estos diagramas tienen como objetivo describir de una forma más completa la interacción entre el actor y el sistema.

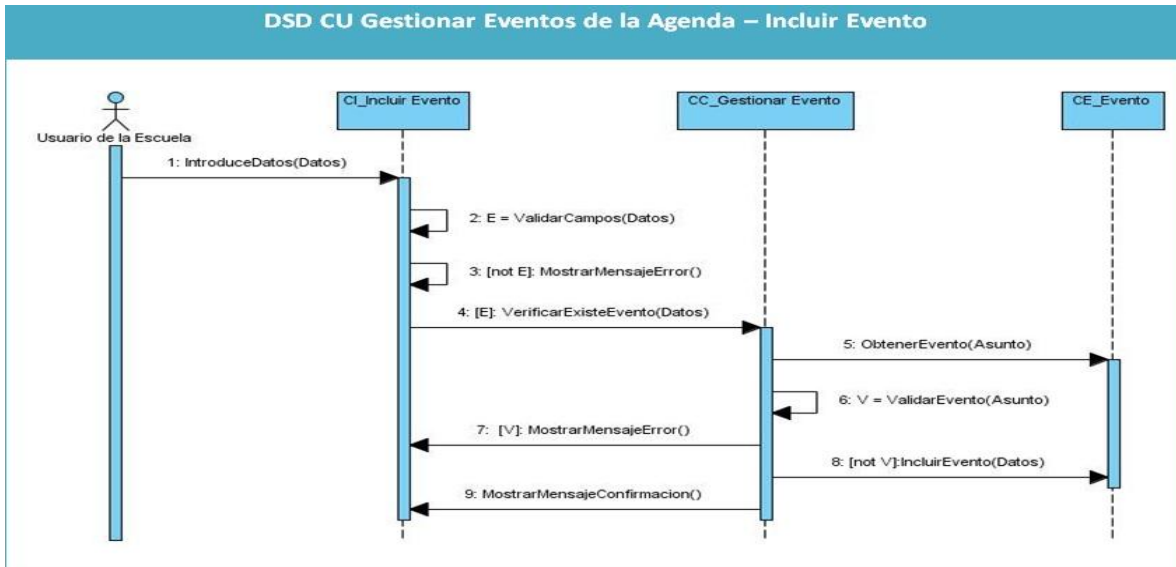


Figura 3.10: DSD CU Gestionar Eventos de la Agenda – Incluir Evento.

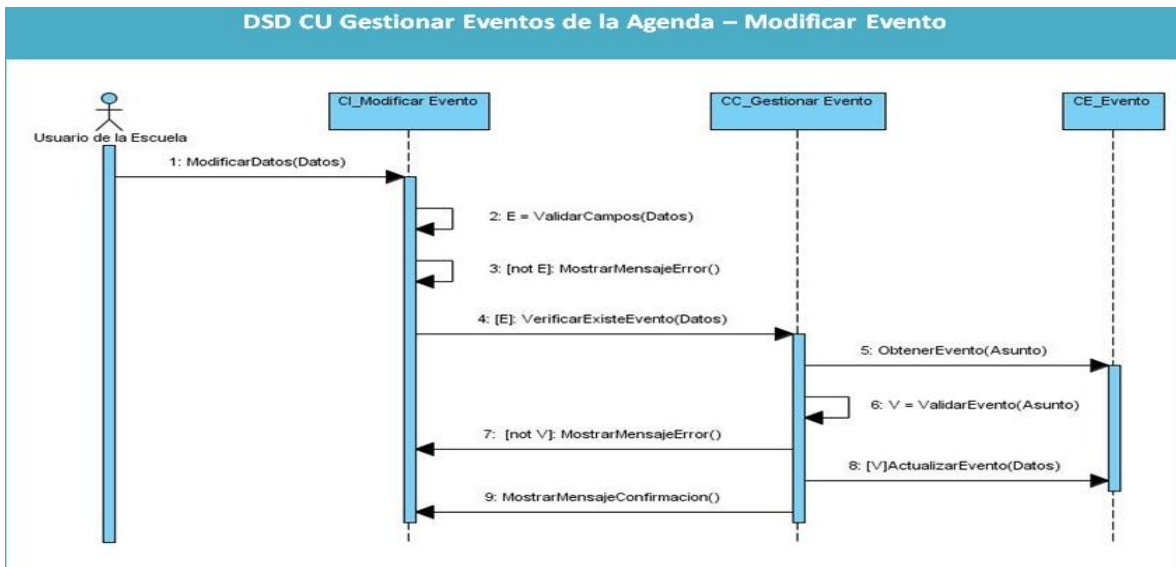


Figura 3.11: DSD CU Gestionar Eventos de la Agenda – Modificar Evento.

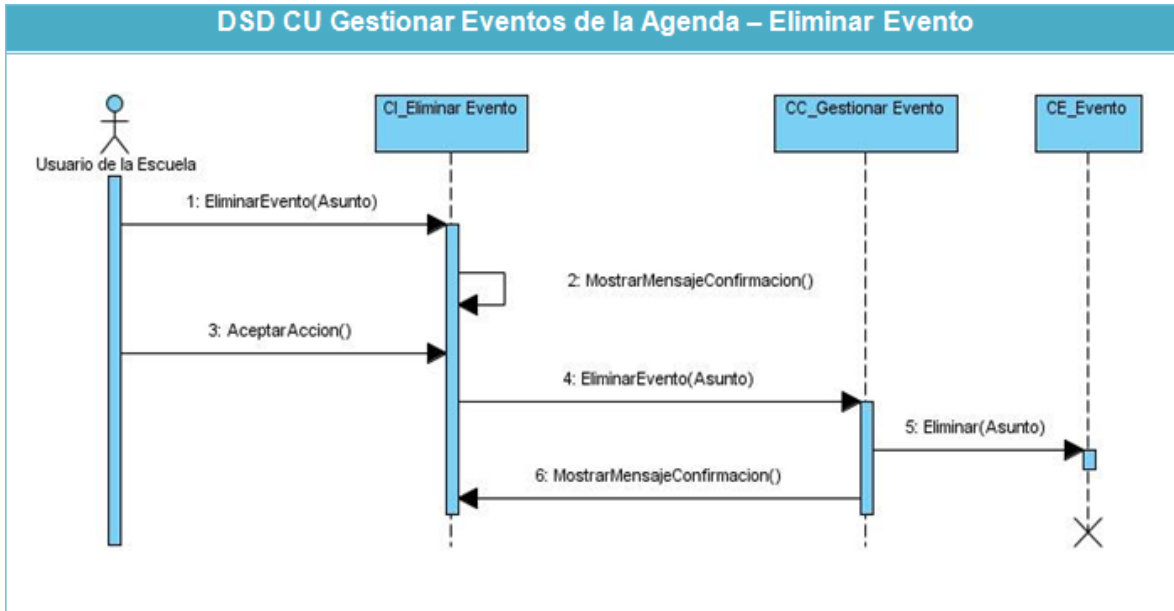


Figura 3.12: DSD CU Gestionar Eventos de la Agenda – Eliminar Evento.

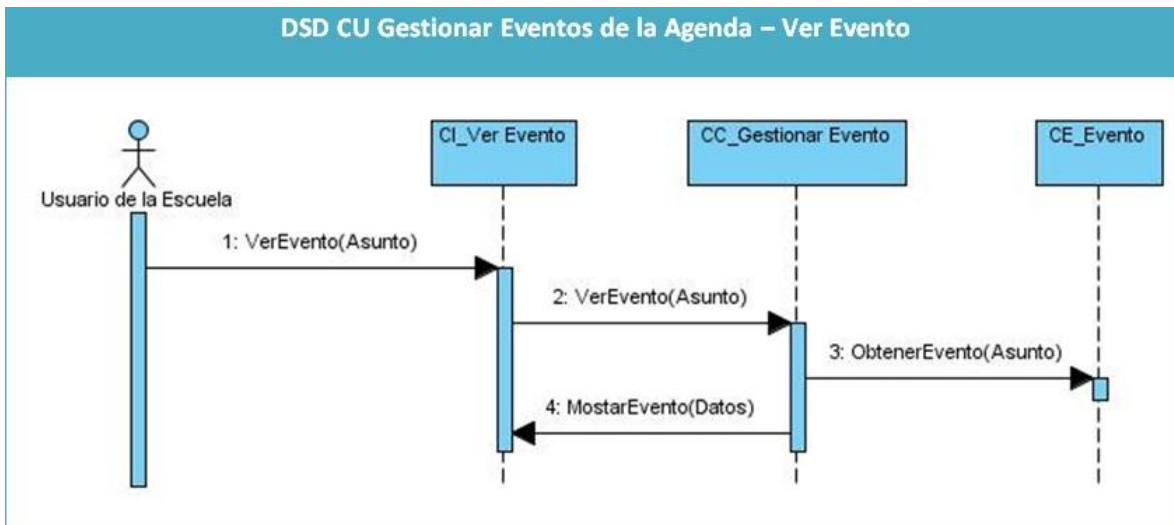


Figura 3.13: DSD CU Gestionar Eventos de la Agenda – Ver Evento.

Para consultar otros diagramas de secuencia del diseño y lograr una comprensión más detallada de la interacción entre usuario y sistema, así como entre los componentes del sistema; ver anexos del 19 al 25.

3.3.5. Diseño de la Base de Datos

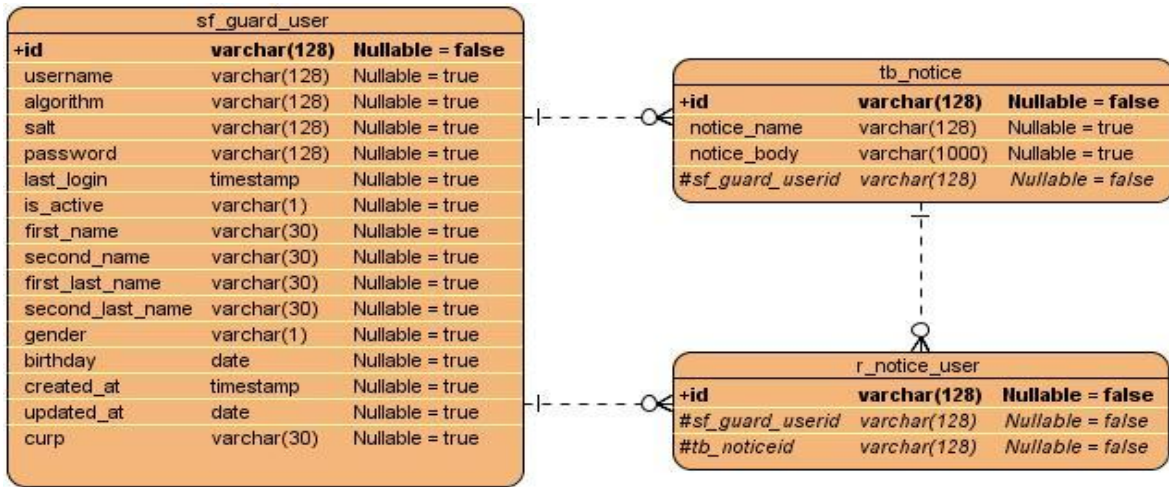


Figura 3.14: Diagrama de Modelo de Datos de Avisos

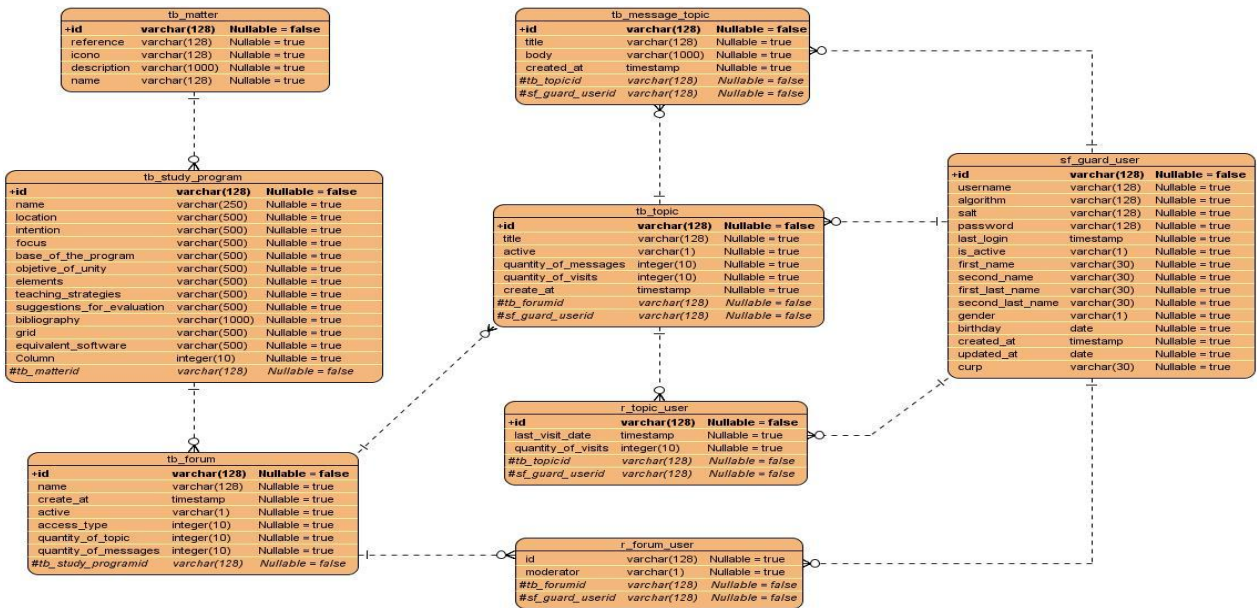


Figura 3.15: Diagrama de Modelo de Datos de Foro

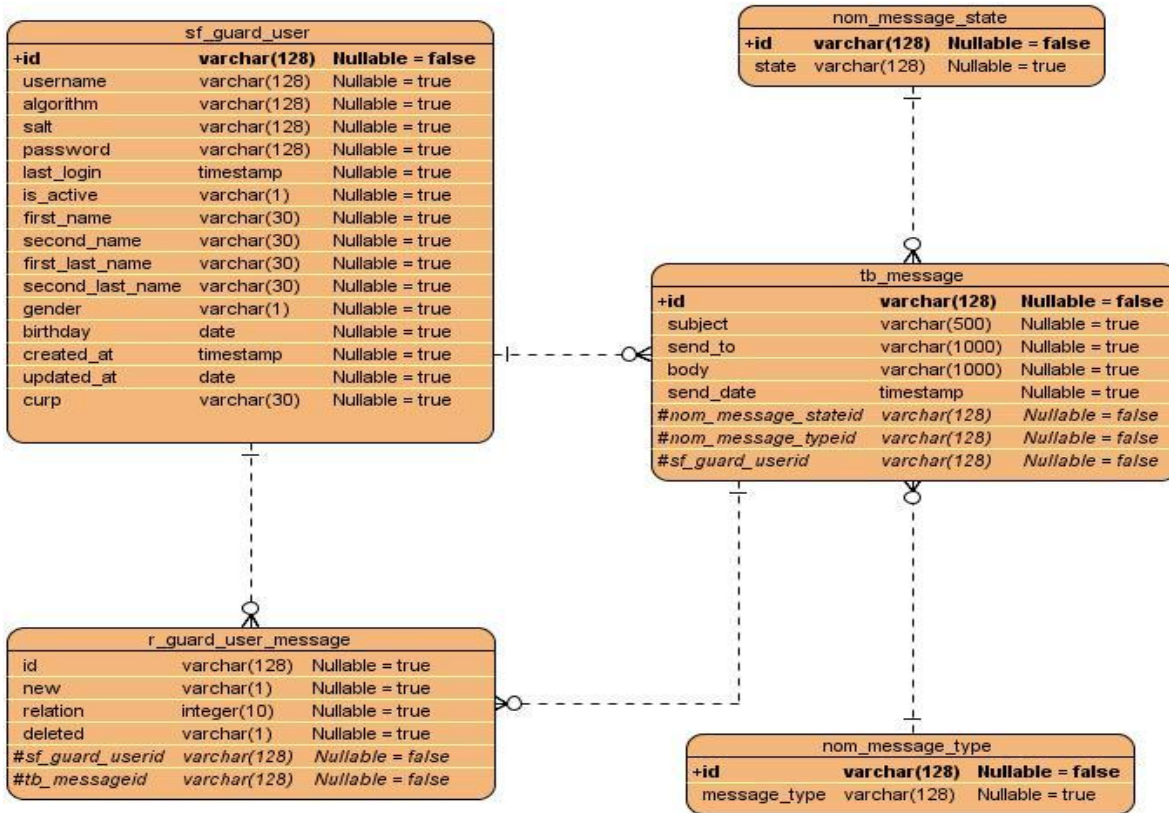


Figura 3.16: Diagrama de Modelo de Datos de Mensajería

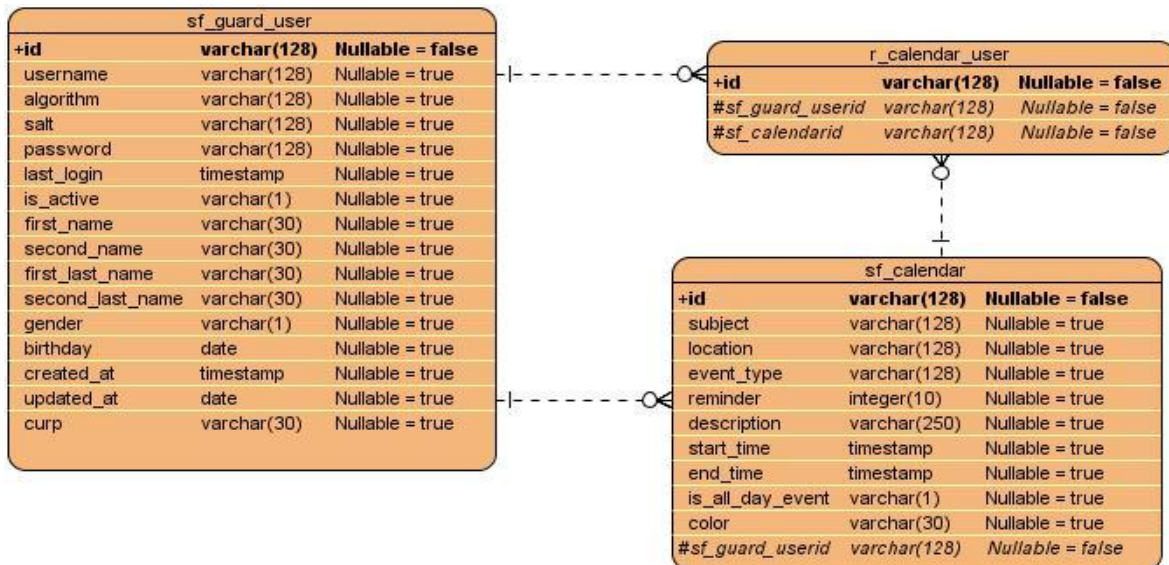


Figura 3.17: Diagrama de Modelo de Datos de Agenda

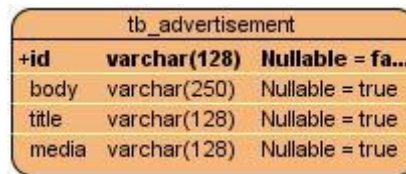


Diagrama de Modelo de Datos de Anuncios (tb_advertisement) que muestra los atributos y sus propiedades de nulosidad.

tb_advertisement		
+id	varchar(128)	Nullable = fa...
body	varchar(250)	Nullable = true
title	varchar(128)	Nullable = true
media	varchar(128)	Nullable = true

Figura 3.18: Diagrama de Modelo de Datos de Anuncios

3.4. Conclusiones

Se generaron los artefactos propuestos por la metodología RUP en el flujo de trabajo Análisis y Diseño, obteniendo la presentación del diseño del módulo Comunicaciones de la Plataforma ZERA. Se crearon además las bases para la implementación de la propuesta.

Conclusiones

El estudio de las diferentes plataformas educativas existentes en el mundo, haciendo énfasis en las comunicaciones de las mismas facilitó la creación de las funcionalidades del Módulo de Comunicaciones de la plataforma ZERA.

La selección de la Metodología RUP utilizada para el desarrollo del sistema posibilitó la creación de artefactos fundamentales que permitieron la creación del módulo con calidad y cumpliendo con los requerimientos del cliente.

Se generaron los artefactos propuestos por la metodología RUP en el flujo de trabajo Análisis y Diseño, obteniendo la presentación del diseño del módulo Comunicaciones de la Plataforma ZERA, permitiendo además la creación de las bases para la implementación de la propuesta.

Recomendaciones

Implementar las funcionalidades correspondientes al Módulo de Comunicaciones de la Plataforma ZERA, partiendo del análisis y diseño propuesto en el trabajo.

Continuar las investigaciones sobre las tendencias en las comunicaciones de las plataformas educativas para actualizar en un futuro las funcionalidades existentes.

Tener en cuenta este trabajo para el futuro desarrollo de sistemas similares.

Glosario de Términos

E-Learning: es la transferencia de habilidades y conocimientos a través de una computadora y una red habilitada. Puede ser autónomo o guiado por un instructor en forma de texto, imágenes, animaciones, video y audio.

Framework: estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Hiperentornos: no son más que una mezcla de diferentes tipos de software educativos sustentados en tecnología hipermedia, los cuales constituyen un medio de enseñanza para el apoyo al proceso de enseñanza y aprendizaje.

Hipermedia: es la organización de información textual, visual y sonora a través de vínculos que crean asociaciones entre información relacionada dentro del sistema.

Plataformas Educativas: espacios para facilitar la interacción entre docentes y estudiantes con fines pedagógicos.

Plugin: es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica.

Referencias Bibliográficas

1. **García Aretio, L.** *De la educación a distancia a la educación virtual*. Editorial Ariel, Barcelona, 2007.
2. **López Rayón Parra, A. E., Escalera Escajeda, S., Ledesma Saucedo, R.** *Ambientes Virtuales de Aprendizaje*. En Presimposio Virtual Somece, México, 2002.
3. **Zapata Ros, M.** *Sistemas de gestión del aprendizaje – Plataformas de teleformación*, Argentina, 2003.
4. **De la Torre, A.** *Introducción a la plataforma Moodle*, 2006.
5. **UCL.** *CLAROLINE.NET*. 2001. <http://www.claroline.net>
6. **educarenergetico.wordpress.com.** *Plataforma Educativa Dokeos*. [En Línea] [Citado el: 19 de enero de 2011]. 3 de agosto 2007.
7. <http://educarenergetico.wordpress.com/2007/08/03/dokeos-una-plataforma-educativa/>
8. **Sepúlveda Peña, J. C.** *Plataforma de Teleformación aprenDIST*. Centro de Estudios de Ingeniería de Sistemas (CEIS), Instituto Superior Politécnico “José A. Echeverría” (CUJAE). 2005.
9. **Cortizo Pérez, J.C., Expósito Gil, D., Ruiz Leyva M.** *eXtreme Programming*, AINetSolutions Technical Report No. 01, 2003.
10. **Albaladejo, X.** *SCRUM*. [En Línea] [Citado el: 16 de enero de 2011]. 2008. <http://www.proyectosagiles.org/que-es-scrum>
11. **Grupo soluciones GSInnova.** *Rational Unified Process*. [En Línea] [Citado el: 16 de enero de 2011]. 2007. <http://www.rational.com.ar/index.html>.
12. **Gómez Gallego, J. P.** *Fundamentos de la Metodología RUP, Rational Unified Process*. Universidad Tecnológica de Pereira, Colombia, 2007. Disponible en: <http://www.scribd.com/doc/297224/RUP>
13. **Martínez Echevarría, A.** *Manual Práctico de HTML*. Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Madrid, España, 1995.

14. **Pérez, J. E.** *Introducción a XHTML*. España, 2009. Disponible en: <http://librosWeb.es/>
15. **Pérez, J. E.** *Introducción a CSS*. España, 2009. Disponible en: <http://librosWeb.es/>
16. **Pérez, J. E.** *Introducción a Ajax*. España, 2009. Disponible en: <http://librosWeb.es/>
17. **Pérez, J. E.** *Introducción a JavaScript*. España, 2009. Disponible en: <http://librosWeb.es/>
18. **Pérez, D.** *Maestros del Web*. [En línea] [Citado el: 18 de enero de 2011]. 2007. <http://www.maestrosdelWeb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-Web/>.
19. **Ruby.** *Ruby A Programmer's Best Friend*. [En línea] [Citado el: 18 de enero de 2011]. <http://www.ruby-lang.org/es/about>.
20. **Python.** *Python*. [En línea] [Citado el: 18 de enero de 2011]. 2009. <http://www.python.org>.
21. **Hinostroza, R. R.** *Características de PHP*. 2005. Disponible en: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>
22. **Gutiérrez, J. J.** *¿Qué es un framework Web?*, 2009. Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
23. **Potencier, F.** *Symfony la guía definitiva*, 2009. Disponible en: <http://librosWeb.es/>
24. **Zend Framework Des.com.** *Introducción a Zend framework*. [En línea] [Citado el: 18 de enero de 2011], 14 de Febrero de 2010. <http://manual.zfdes.com/es/introduction.overview.html>
25. **Cookbook.** *¿Qué es CakePHP y por qué hay que utilizarlo? El Manual*. [En línea] [Citado el: 18 de enero de 2011] <http://book.cakephp.org/es/view/8/What-is-CakePHP-Why-Use-it>
26. **Hernández Orallo, E.** *El Lenguaje Unificado de Modelado (UML)*.2004.
27. **pencil.evolus** *Pencil Project*. [En Línea] [Citado el: 19 de enero de 2011]. 3 de agosto 2007. <http://pencil.evolus.vn/en-US/Home.aspx>

28. **crystalab**. *Prototipos, bocetos y wireframes con Balsamiq Mockups*. [En Línea] [Citado el: 19 de enero de 2011]. 2008. <http://www.crystalab.com/>
29. **Tuesta, Martín**. *El prisma*. [En línea] [Citado el: 18 de enero de 2011]. 2005. <http://www.elprisma.com/apuntes/curso.asp?id=13324>
30. **Ramírez, A., Vanpeperstraete, P., Rueckert, A., Odutola, K., Bennett, J., Tolke, L., van der Wulp, M.** *ArgoUML: Manual de Usuario. Tutorial y descripción de referencia*. 2006.
31. **Visual Paradigm**. *Visual Paradigm*. [En Línea] [Citado el: 19 de enero de 2011]. <http://www.visual-paradigm.com/>
32. **Garcerant, I.** *Tecnología y Synergix. Visión de Synergix de los Sistemas de Información y la Ingeniería del Software. Modelo de dominio*. [En Línea] [Citado el: 10 de febrero de 2011]. 2008. <http://synergix.wordpress.com/author/igarcerant/>
33. **Gomes Gallego, J. P., Galves, J.A.** *Ingeniería de Requerimientos*. Universidad Tecnológica de Pereira. 2006.
34. **Díez González, O.** *Safety y Requisitos No Funcionales*. Universidad Politécnica de Madrid. Tesis de Doctorado. 2006.
35. **Rondón Rodríguez, M.P.** *Tecnología de la Información*. Programa Profesional de Ingeniería Industrial de la Universidad Católica de Sao Pablo. 2011.
36. **Brito Acuña, K.** *Selección de Metodologías de Desarrollo para Aplicaciones Web en la Facultad de Informática de la Universidad de Cienfuegos*. Universidad de Cienfuegos “Carlos Rafael Rodríguez”. 2009.
37. **Potencier, F.** *Symfony la guía definitiva*. 2009.
38. **Koch, N., Escalona, M. J.** *Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo*. Sevilla, España. 2002.