



Universidad de las Ciencias Informáticas
Facultad 4

Implementación y prueba del Módulo Comunicaciones de la Plataforma Educativa ZERA

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

Autor:

Blas Antonio Tejera Mendoza

Tutor:

Ing. Daniel Rodríguez Soberats

Co-tutor:

Ing. Renier Portelles Cobas

La Habana, Junio 2011
"Año 53 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

Blas Antonio Tejera Mendoza

Firma del Autor

Ing. Daniel Rodríguez Soberats

Firma del Tutor

Ing. Renier Portelles Cobas

Firma del Co-tutor



“Es una locura seguir haciendo siempre lo mismo y esperar resultados diferentes”

Albert Einstein.

DATOS DE CONTACTO

Tutor:

Ing. Daniel Rodríguez Soberats: Profesor graduado en el año 2010 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI). Pertenece al Centro de Tecnologías para la Formación (FORTES). Forma parte del proyecto Alfaomega de la Facultad 4.

Correo electrónico: dsobrats@uci.cu

Co-tutor:

Ing. Renier Portelles Cobas: Profesor de las asignaturas de Arquitectura de Computadoras y Teleinformática II. Facultad 4, Universidad de las Ciencias Informáticas (UCI). Ingeniero Automático, Universidad de Oriente, 2003. Se desempeñó de profesor de Máquinas Computadoras I y II en el primer año de trabajo en la universidad, en ese mismo año participó por 4 meses en el proyecto Identidad de gran importancia para la universidad y el país. En el curso 2004 fue nombrado Vicedecano de Producción e Investigación de la Facultad 8 el cual desempeño por 3 años. En el 2008 fue seleccionado como profesor para cumplir misión en Venezuela frente a los estudiantes de CDI y en el 2009 fue seleccionado jefe de la misión CDI por la parte de la UCI en Venezuela, al frente de todos los profesores y estudiantes, actualmente se mantiene de profesor en la Facultad 4.

Correo electrónico: renierpc@uci.cu

DEDICATORIA

A mi mamá, por haberme dedicado toda su vida.

A mi padrastro, por estar con nosotros en los momentos más difíciles.

A todos los que me aprecian y se preocupan por mí.

AGRADECIMIENTOS

A mi madre por apoyarme en mis decisiones y por la confianza que ha depositado en mí.

A mi padrastro por haberme acogido como su hijo y brindarme su apoyo en todo momento.

A mi hermano Yosbel y a Naryana por toda la ayuda brindada.

A todos mis familiares por su apoyo durante todo este tiempo.

A los que conocí en la universidad y consideré amigos, porque aprendí a confiar en ellos y de los que recibí apoyo y cariño: Roberto, Deymis, Anita, Julio, Alejandro, Mailín, Ifrain, Mónica, Yoyi, Pachi, Raidel, Liset, Leo y demás amigos, que siempre estuvieron ahí y me apoyaron cada vez que los necesite.

A mis amigos de toda la vida: Reinier, Janier, Alberto y Ramón Luis.

A mis vecinos y amigos por quererme tanto y preocuparse por mí.

A todos los profesores que me ayudaron en mi preparación profesional.

A mis tutores Renier Portelles y Daniel Rodríguez por guiarme durante el desarrollo de la tesis.

A los compañeros del proyecto, en especial a Álvarez por todo el apoyo en el desarrollo de la tesis.

A la Revolución cubana por darme la oportunidad de hacerme un profesional.

Gracias a todos por la comprensión y la dedicación que me han brindado...se les quiere a todos...!!!!

RESUMEN

Con el desarrollo de la informática y las comunicaciones se ha desarrollado la educación de forma vertiginosa, pues mediante las plataformas para la gestión de aprendizaje se logra realizar acciones formativas sobre un gran número de estudiantes sin restricciones geográficas. En la Universidad de las Ciencias Informáticas se está desarrollando unas de estas plataformas y toma como nombre ZERA.

En el presente trabajo se desarrolla un módulo dedicado a las comunicaciones de la plataforma ZERA, el cual se encarga de gestionar mensajes, foros, temas, comentarios y además es quien permite notificar muchas de las tareas que realizan los usuarios sobre dicha plataforma.

Para llegar a la solución se realiza un amplio análisis de herramientas y tecnologías Web más utilizadas por los desarrolladores de software y se seleccionan las pertenecientes al software libre. Partiendo de los requisitos y el diseño definido por los analistas de sistema se hace una breve descripción de las funcionalidades del módulo Comunicaciones, se establecen estándares de codificación para lograr una un código legible y de fácil mantenimiento; y para lograr una programación bien estructurada y guiada por buenas prácticas se aplican un conjunto de patrones arquitectónicos y de diseño. Finalmente, se realizan un conjunto de pruebas obteniendo una satisfactoria validación de la solución.

Palabras claves: comunicaciones, módulo, plataforma, ZERA.

ÍNDICE

Introducción	1
Capítulo 1: Fundamentación Teórica	5
1.1 Conceptos asociados al dominio del problema	5
1.1.1 Hipertexto	5
1.1.2 Hipermedia	5
1.1.3 Hiperentorno de aprendizaje	5
1.1.4 Plataforma Educativa	5
1.1.5 LMS (Learning Management Systems)	6
1.2 Análisis de soluciones similares existentes.....	6
1.2.1 Sistemas a nivel internacional.....	6
1.2.2 Sistemas a nivel nacional.....	7
1.3 Selección de herramientas, tecnologías y lenguajes para el desarrollo	8
1.3.1 Sistemas de Foros	8
1.3.2 Clientes de Correo Electrónico.....	9
1.3.3 Lenguaje de programación.....	10
1.3.3.1 Lenguajes y tecnologías del lado del cliente	11
1.3.3.2 Lenguajes y tecnologías del lado del servidor	12
1.3.4 Frameworks	12
1.3.5 Servidor Web	14
1.3.6 Gestor de Bases de Datos	15
1.3.7 Metodología de desarrollo de software y estándares	16
1.3.7.1 Metodologías Tradicionales o Pesadas.....	17
1.3.7.2 Metodologías Ágiles.....	19
1.3.8 Herramientas CASE.....	21
1.3.9 Herramientas de desarrollo	21
1.3.10 Selección de las tecnologías y herramientas.....	23
1.4 Conclusiones parciales.....	25
Capítulo 2: Implementación del sistema.....	26

2.1 Descripción de los submódulos	26
2.1.1 Submódulo Foro	26
2.1.2 Submódulo Mensajería Interna	26
2.2 Requisitos del software.....	27
2.2.1 Requisitos Funcionales	27
2.2.2 Requisitos no funcionales	28
2.3 Estándares de Codificación	30
2.3.1 Identificadores.....	30
2.3.2 Líneas y espacios en blanco	31
2.3.3 Comentarios.....	32
2.4 Patrones utilizados en la implementación	32
2.4.1 Patrones arquitectónicos.....	32
2.4.2 Patrones de diseño	33
2.4.2.1 Patrones GRASP	33
2.4.2.2 Patrones GoF.....	34
2.5 Descripción de las principales clases y funcionalidades empleadas en el desarrollo del sistema	34
2.5.1 Clases del Modelo	35
2.5.2 Clases Controladoras.....	37
2.5.2.1 Submódulo Foro	37
2.5.2.2 Submódulo Mensajería Interna	38
2.6 Modelo de Implementación.....	39
2.6.1 Diagrama de Despliegue.....	39
2.6.2 Diagrama de Componentes	40
2.6.2.1 Submódulo Foro	40
2.6.2.2 Submódulo Mensajería Interna	42
2.7 Conclusiones parciales.....	43
Capítulo 3: Validación de la solución propuesta.....	44
3.1 Pruebas del software	44
3.1.1 Niveles de prueba	44

3.1.2 Métodos de prueba	45
3.1.3 Diseño de Casos de Prueba	45
Submódulo Foro	46
3.2 Resultado obtenido.....	59
3.3 Conclusiones parciales.....	60
Conclusiones Generales	61
Recomendaciones	62
Referencias Bibliográficas.....	63
Bibliografía.....	66

TABLAS

Tabla 1. Clase del modelo	36
Tabla 11 CP Ver foro	48
Tabla 12 Crear Tema de foro	50
Tabla 13 SC Ver datos del tema	54
Tabla 14 SC Modificar datos del tema	57
Tabla 15 SC Eliminar Tema	57
Tabla 16 Descripción de las variables.....	58
Tabla 17 SC Listar Tema del Foro	59
Tabla 18 Resultado de las NC	60

ILUSTRACIONES

Ilustración 1 Diagrama de despliegue	40
Ilustración 2 Diagrama de componentes foro	41
Ilustración 3 Diagrama de componentes mensajería.....	42

INTRODUCCIÓN

A partir de la Era Digital se han logrado grandes avances tecnológicos y científicos que han permitido mejoras en procesos que se realizan cotidianamente, de estos adelantos nació la Informática que se ha desarrollado de manera vertiginosa y es aplicable a casi todas las complejas tareas que realiza la sociedad en estos días.

La informática como ciencia engloba muchas ramas como “la Informática Educativa, que no es más que una disciplina, producto de la sinergia entre la Ciencia de la Educación y la Ciencia de la Informática; utilizando una computadora en contextos de aprendizaje de las distintas áreas curriculares; haciendo ver al alumno que a través de este recurso tecnológico puede afianzar y ampliar sus conocimientos”. (1)

Hoy día existen herramientas y aplicaciones enfocadas a la educación tales como: multimedias, bibliotecas virtuales, laboratorios virtuales, libros electrónicos, etc.; que mediante Internet o una red entre ordenadores logran que se desarrollen acciones formativas y un satisfactorio proceso enseñanza-aprendizaje. Este proceso sin limitaciones de horarios caracterizado por una separación física entre estudiantes y profesores con un apoyo continuo de tutores especializados toma el nombre de e-Learning o Teleformación. (2)

Existen herramientas encargadas de gestionar y controlar las actividades no presenciales de una institución con el nombre de Sistema de Gestión del Aprendizaje (LMS, Learning Management System) y en Internet se pueden encontrar muchas que gozan de gran reconocimiento, algunas de ellas son Claroline, Moodle basadas en software libre y Blackboards basada en software propietario.

En la informática es muy común el uso del mismo software (ejemplo: la mayoría de los usuarios a nivel mundial utilizan como sistema operativo Microsoft Windows y Microsoft Office como herramienta ofimática), en caso de los LMS se tiende a utilizar Moodle y BlackBoard que son los que más se ajustan de forma general a las especificaciones de los usuarios, pero muchas veces para utilizar y/o comercializar dichas herramientas es necesario contar con un permiso especial que provee el productor del software y no todas estas herramientas están basadas en software libre, por tanto se dificulta su distribución en

países con pocos recursos; por tal motivo algunos de estos países apoyándose en las facilidades que provee el software libre se empeñan en desarrollar herramientas educativas con las principales funcionalidades que garantizan el proceso de enseñanza-aprendizaje y permitan adaptarse a necesidades específicas de usuarios y/o clientes.

Actualmente, Cuba cuenta con un personal altamente calificado en materia de educación, informática y comunicaciones, dándose a la tarea de realizar herramientas educativas basadas en software libre con el objetivo de promover la educación y lograr beneficios económicos; evidenciándose el trabajo realizado con la publicación de la Colección Futuro¹ y Multisaber, patrocinadas por el Ministerio de Educación en Cuba (MINED), que cumplen con los requerimientos de los usuarios tanto dentro como fuera del país.

La problemática

En la Universidad de las Ciencias Informáticas (UCI), modelo de universidad productiva que agrupa más de doce mil profesionales, técnicos y estudiantes; existe un centro llamado Centro de Tecnologías para la Formación (FORTES), que su razón de ser es desarrollar tecnologías que permitan ofrecer servicios y productos para la implementación de soluciones de formación aplicando las Tecnologías de la Información y las Comunicaciones.

En este centro se está desarrollando la Plataforma Educativa ZERA que tiene como misión fundamental ser una potente plataforma capaz de adaptarse a los procesos del negocio de una institución o escuela, la misma se inició a petición del Grupo Editorial Mexicano ALFAOMEGA, motivados principalmente por la Colección Futuro. En esta plataforma se integran las principales especificaciones y estándares educativos utilizados a nivel mundial en plataformas educativas, y permitiendo la gestión y visualización de hiperentornos de aprendizaje.

La plataforma ZERA está dividida en varios módulos o subsistemas que están estrechamente interrelacionados y se hace necesario un espacio para atender las comunicaciones. Por tanto, para lograr que el proceso de enseñanza-aprendizaje mediante la Plataforma Educativa ZERA sea una actividad de éxito, es necesario mejorar la comunicación entre los usuarios, el intercambio directo de información y notificar las acciones que se realizan en la plataforma. Por lo que el **problema a resolver** quedaría

¹ Hiperentornos de aprendizaje desarrollados por el MINED para estudiantes de la enseñanza media.

resumido en la siguiente interrogante: ¿Cómo gestionar la comunicación entre los usuarios en la Plataforma Educativa ZERA?

Basados en el problema anterior se identificó como **objeto de estudio** los procesos para la gestión de foros y mensajes en Plataformas Educativas; enmarcando el **campo de acción** en la gestión de foros y mensajes en la Plataforma Educativa ZERA.

El **objetivo general** que persigue este trabajo, consiste en: realizar la Implementación y Prueba del Módulo Comunicaciones de la Plataforma Educativa ZERA.

Del objetivo general se derivan los siguientes **objetivos específicos**:

- ✓ Realizar un estudio sobre las herramientas que posibiliten la implementación del módulo Comunicaciones de la Plataforma Educativa ZERA.
- ✓ Implementar el módulo Comunicaciones de la Plataforma Educativa ZERA.
- ✓ Realizar prueba del módulo Comunicaciones de la Plataforma Educativa ZERA.

La **idea a defender** que persigue este trabajo es que con la realización de la implementación y prueba del módulo Comunicaciones de la Plataforma Educativa ZERA, se logrará la interacción entre los usuarios, logrando un acercamiento virtual entre docentes y estudiantes; y de esta manera se fortalece el proceso enseñanza-aprendizaje a través de la Plataforma ZERA.

Con el propósito de dar la validez a lo anteriormente planteado, se establecen las siguientes **tareas de investigación**:

- ✓ Estudio e investigación de las herramientas, tecnologías y tendencias actuales relacionadas con la interacción de los usuarios en plataformas educativas.
- ✓ Identificación de las herramientas que se utilizarán en el desarrollo de la presente investigación.
- ✓ Implementación del módulo Comunicaciones.
- ✓ Realización de pruebas al módulo Comunicaciones.

Los métodos de investigación que soportan la siguiente investigación es una combinación dialéctica entre los **métodos teóricos y empíricos**. Los empleados son los que se muestran a continuación:

Métodos Teóricos

- ✓ **Histórico – Lógico:** mediante el empleo de este método se realizó un estudio de las Plataformas Educativas para la gestión del aprendizaje.
- ✓ **Analítico – Sintético:** sirve de base para la revisión de bibliografías y materiales relacionados con el objeto de estudio, permitiendo llegar a una solución al problema.
- ✓ **Modelación:** se utiliza principalmente en la realización de los diagramas que permiten una mejor comprensión de la solución.

Métodos Empíricos

- ✓ **Observación:** mediante la utilización del presente método se logra conocer la esencia del problema planteado, pues analizando desde varios puntos de vista la propuesta de solución y otras soluciones existentes, se permite identificar que está hecho y que falta por hacer.

Estructura capitular

Capítulo 1: Fundamentación Teórica

En este capítulo se presentan algunos de los principales conceptos utilizados en la investigación, se realiza un estudio de herramientas, tecnologías, lenguajes y se eligen aquellas que más se adapten a las necesidades que exige la realización de la aplicación.

Capítulo 2: Implementación del módulo Comunicaciones

La razón de ser de este capítulo, es representar en el diagrama de componentes la forma que está estructurado el sistema, reflejando la transformación de los elementos del modelo de diseño en términos de componentes, ficheros que contienen código fuente, ejecutables, así como dependencias entre ellos. Además es presentado el diagrama de despliegue.

Capítulo 3: Pruebas del módulo Comunicaciones

Este capítulo está dedicado a las pruebas, mediante estas se lograrán validar las funcionalidades realizadas en la implementación del módulo, con el objetivo de verificar la propuesta de solución.

Capítulo 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se tratarán algunos de los principales conceptos asociados al dominio del problema, se estudiarán los procesos similares existentes y además se llevará a cabo un estudio sobre las tecnologías y herramientas que se utilizarán para el desarrollo del software.

1.1 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA

1.1.1 HIPERTEXTO

Es una tecnología que organiza bloques de información de distintos contenidos, conectados a través de una serie de enlaces. El hipertexto ha sido definido como enfoque para manejar y organizar información, en el cual los datos se almacenan en una red de nodos conectados por enlaces. Los nodos contienen textos y si contienen además gráficos, imágenes, audio, animaciones y videos, se les da el nombre de hipermedia, es decir una generalización de hipertexto. (3)

1.1.2 HIPERMEDIA

Es un sistema que provee al usuario una forma libre y única de acceder y explorar la información realizando saltos entre un documento y otro. Los usuarios no están obligados a seguir una secuencia establecida, sino que se pueden mover a través de la información, siguiendo sus intereses en búsqueda de un término o concepto. Los sistemas hipermedia han evolucionado hasta disponer de las capacidades necesarias para incluir en los sistemas hipertextuales documentos e información con presentación gráfica y/o sonora. (3)

1.1.3 HIPERENTORNO DE APRENDIZAJE

Es un sistema informático apoyado en tecnología hipermedia, que contiene una mezcla o elementos representativos de diversas tipologías de software educativo. (4)

1.1.4 PLATAFORMA EDUCATIVA

La implementación de una plataforma educativa para la administración de cursos permite la autonomía de producción y publicación en la red de recursos y contenidos por parte de los participantes. El docente, teniendo en cuenta los objetivos educativos, autónomamente y con la posibilidad de editar en cualquier

momento pone a disposición de sus estudiantes mediante alguno de los recursos ofrecidos por la plataforma: el programa del curso, contenidos o unidades temáticas (expuestos en textos, hipertextos, presentaciones, animaciones, videos...), actividades, bibliografía y evaluación. De manera análoga, con otro nivel de autonomía, el estudiante puede acceder a los contenidos y al desarrollo de las actividades propuestas. (5)

1.1.5 LMS (LEARNING MANAGEMENT SYSTEMS)

Es un Sistema de Administración de Aprendizaje, ampliamente conocido como Plataforma de Aprendizaje, instalado en un servidor que se emplea para administrar, distribuir y controlar las actividades de formación presencial o E-Learning de una institución u organización.

Los sistemas LMS o plataformas de E-Learning generalmente incorporan las siguientes funciones:

- ✓ Configurar Cursos.
- ✓ Matricular Alumnos.
- ✓ Registrar Profesores.
- ✓ Asignar cursos a un alumno.
- ✓ Realizar Evaluaciones.
- ✓ Informes de Progreso y Calificaciones.
- ✓ Publicación de Contenidos y Actividades. (6)

1.2 ANÁLISIS DE SOLUCIONES SIMILARES EXISTENTES

En la actualidad, a nivel mundial existen una gran variedad de plataformas educativas destacándose las que siguen a continuación:

1.2.1 SISTEMAS A NIVEL INTERNACIONAL

MOODLE

La palabra Moodle (Entorno de Aprendizaje Dinámico Modular Orientado a Objetos) es una herramienta que permite crear espacios virtuales de trabajo, formados por recursos de información (en formato textual o tabular, fotografías o diagramas, audio o vídeo, páginas Web, entre muchos otros) así como recursos de formación tipo tareas enviadas por la Web, exámenes, encuestas, foros, entre otros. Facilita los

mecanismos mediante los cuales el material de aprendizaje y las actividades de evaluación son realizadas por el estudiante, pero también donde los tutores o profesores pueden introducirse en el diseño y la forma de llevar el conocimiento hasta sus alumnos. Entre los recursos con los que cuenta la plataforma Moodle para garantizar la comunicación dentro de la misma se encuentra: (7)

- ✓ **Foro:** es aquí donde se desarrollan la mayor parte de los debates. Pueden estar estructurados de diferentes maneras, e incluso permitir evaluar la participación. Posibilitan 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.

CLAROLINE

Claroline es una plataforma de aprendizaje y trabajo virtual. Publicada bajo una licencia Open Source, permite a los formadores construir eficaces cursos en línea y gestionar las actividades de aprendizaje y colaboración en la Web. Está organizada alrededor del concepto de espacios relacionados con un curso y actividad pedagógica. Es adaptable a varios contextos de formación, Claroline no es usada únicamente por escuelas y universidades, también en centros de formación, asociaciones y compañías. La plataforma es adaptable y ofrece un entorno de trabajo flexible y ajustable. Cada espacio provee una lista de herramientas que permite crear contenidos de aprendizaje y gestión de actividades de formación. Esta plataforma posibilita: administrar una agenda con tareas y fechas límite, publicar anuncios (también por e-mail), administrar foros tanto públicos como privados. Entre las actividades que se pueden realizar en esta plataforma que hacen posible la comunicación entre usuarios está:

- ✓ **Hablar:** Foro, espacio público para hablar/debatir dividida en distintos temas (conversación asíncrona). Herramienta en línea para el debate (conversación síncrona). (7)

1.2.2 SISTEMAS A NIVEL NACIONAL

APRENDIST

AprenDIST es una plataforma de educación a distancia. Es un sistema diseñado para un alumnado con determinadas características, además de ser una plataforma flexible y ajustable a toda una variedad de cursos.

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 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. (8)

Al realizar el análisis de las soluciones anteriores se llega a la conclusión de que estas herramientas están pensadas para estudiantes universitarios y no para alumnos de la enseñanza media superior que es una de las metas que se quiere lograr con la Plataforma Educativa ZERA. Por otra parte, con respecto a las comunicaciones, ninguna cumple con todas las herramientas necesarias para lograr una adecuada interacción entre los usuarios, pero si sirven como guía y además permiten obtener una visión más amplia de las funcionalidades que se desean realizar en el módulo Comunicaciones de ZERA.

1.3 SELECCIÓN DE HERRAMIENTAS, TECNOLOGÍAS Y LENGUAJES PARA EL DESARROLLO

1.3.1 SISTEMAS DE FOROS

PUNBB

PunBB es un sistema de foros basado en PHP eficiente y muy personalizable. Proporciona un foro más rápido, de menor tamaño y con menos consumo de recursos, razón por la cual no ofrece tantas opciones como otros sistemas de foros similares, pero cumple con sus funciones a la perfección. Ofrece todas la

funciones de los foros: registro de usuarios con su propio perfil personal, buscador interno, uso de avatares y emoticonos, inserción de enlaces y fotos, etc. Cuenta con un diseño muy simple que se pueden personalizar con skins², genera un código fuente fácil de entender y es muy sencillo de administrar y moderar. (9)

PHPBB

PhpBB es un conjunto gratuito de paquetes de código basados en el popular lenguaje de programación web PHP, cuya intención es la de proporcionar fácilmente, y con amplia posibilidad de personalización, un sistema de foros. Funciona sobre bases de datos basadas en el lenguaje SQL como MySQL, PostgreSQL, Microsoft SQL Server, otras como Microsoft Access y, con una modificación, también sobre Oracle, donde almacena la información para poder recuperarla en cada petición. Está formado por la propia comunidad y sus aportaciones, con la intención de promover el software libre. (10)

1.3.2 CLIENTES DE CORREO ELECTRÓNICO

ZIMBRA

Zimbra es un sistema cliente/servidor para e-mail, contactos y calendarios. Está escrito en Java, y trabaja con MySQL, Postfix (con SpamAssassin y ClamAV) y un servidor de aplicaciones Tomcat.

Se trata de una solución con una interfaz muy trabajada y personalizable que hace uso intensivo de AJAX. Entre sus características se encuentran:

- ✓ El aspecto y la interfaz tienen aspecto de una aplicación de escritorio.
- ✓ Compatible con todos los navegadores.
- ✓ Organización de cuentas de correo.
- ✓ Búsquedas Rápidas (incluyendo mensajes y anexos).
- ✓ Gestión de búsquedas a través de carpetas.
- ✓ Filtros de búsquedas.
- ✓ Calendario corporativo: Organización y grupos y control de horas ocupadas en las agendas.
- ✓ Creación/edición contextual de contactos en el correo.

² Carátulas que permiten a una aplicación cambiar la apariencia.

MOZILLA THUNDERBIRD

Mozilla Thunderbird es un cliente de correo y lector de noticias gratuito, de código abierto y multiplataforma para la mayoría de sistemas operativos modernos incluyendo, pero no limitado a, Windows, GNU/Linux y Macintosh. Está basado en la base de código Mozilla. Es un cliente robusto y fácil de usar.

Características principales:

- ✓ Proporciona las herramientas más eficaces para la detección del correo basura. Puede hacer que el correo basura se borre automáticamente o puede ponerlo en la carpeta que desee.
- ✓ Forma de acceder al correo mediante la nueva vista de tres columnas de Mozilla Thunderbird. Barra de herramientas personalizable, permite cambiar el aspecto mediante el uso de temas y vistas de correo para ordenar rápidamente los mensajes.
- ✓ Proporciona características de seguridad necesarias en gobiernos y empresas como S/MIME, firma digital, permite cifrar mensajes, soporte de certificados y dispositivos de seguridad.
- ✓ Tiene soporte IMAP/POP, correo HTML, etiquetas, búsqueda rápida, libreta de direcciones inteligente, acuses de recibo, filtrado avanzado de mensajes, completado de direcciones LDAP, herramientas de importación, potencia en búsquedas y, además, la capacidad de manejar múltiples cuentas de correo electrónico y noticias.
- ✓ Permite añadir características adicionales según las necesite mediante el uso de extensiones. Las extensiones son una herramienta poderosa que le ayudan a conseguir que su cliente de correo se adapte a todas sus necesidades.

1.3.3 LENGUAJE DE PROGRAMACIÓN

Un lenguaje es un sistema de comunicación que tiene forma, contenido y uso. La programación en informática, es el proceso de escritura del código fuente de un software. De esta forma, la programación le señala al programa informático qué tiene que hacer y cómo realizarlo.

1.3.3.1 LENGUAJES Y TECNOLOGÍAS DEL LADO DEL CLIENTE

HTML

HTML es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto, así como complementar el texto con objetos tales como imágenes. Puede además incluir un script, el cual puede afectar el comportamiento de navegadores web.

(11)

CSS

Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación, es imprescindible para crear páginas web complejas. (12)

JAVASCRIPT

Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (13)

XML

Son las siglas de **Extensible Markup Language**, una especificación/lenguaje de programación desarrollada por el W3C. XML es una versión de SGML (Lenguaje Generalizado Estándar para el Formato de Documentos), diseñado especialmente para los documentos de la web. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones. (14)

JQUERY

jQuery es una biblioteca o framework de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Cuenta con las siguientes características.

- ✓ Selección de elementos DOM.
- ✓ Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.

- ✓ Eventos.
- ✓ Manipulación de la hoja de estilos CSS.
- ✓ Efectos y animaciones.
- ✓ Animaciones personalizadas.
- ✓ AJAX.
- ✓ Soporta extensiones.
- ✓ Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones como trim () (elimina los espacios en blanco del principio y final de una cadena de caracteres), etc.
- ✓ Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 9+ y Google Chrome 1+.

1.3.3.2 LENGUAJES Y TECNOLOGÍAS DEL LADO DEL SERVIDOR

PHP

El acrónimo recursivo, sin embargo, actualmente está vinculado a PHP Hypertext Pre-Processor. El lenguaje es administrado por The PHP Group³ y es utilizado para la creación de páginas web dinámicas. Se considera la licencia PHP como parte del software libre. Suele utilizarse en interpretación del lado del servidor aunque también puede usarse desde una interfaz de línea de comandos. (15)

ASP

Microsoft Active Server Page (ASP) es una tecnología de script que corre del lado de servidor y puede ser usado para crear aplicaciones Web dinámicas e interactivas. Una página ASP es una página HTML que contienen scripts que corren del lado del servidor que son procesados por un servidor Web antes de ser utilizado por el navegador. (16)

1.3.4 FRAMEWORKS

Es una estructura de archivos y utilidades que aceleran la programación de una aplicación informática, proveyendo una metodología de trabajo que sistematiza y facilita la generación de formularios, funciones

³ Grupo de desarrolladores que mantienen a diario las actualizaciones disponibles para el lenguaje PHP.

y módulos de uso común, permitiendo al desarrollador dedicar su atención hacia los aspectos específicos de cada aplicación. (17)

SYMFONY

Symfony es un completo framework diseñado para optimizar, gracias a sus características en el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

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. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. A continuación se muestran algunas de sus características:

- ✓ 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.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador sólo debe configurar aquello que no es convencional.
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la web
- ✓ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✓ 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. (18)

ZEND FRAMEWORK

Es un framework para el desarrollo de aplicaciones Web y servicios Web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además es Open Source y trabaja con PHP 5.

Características fundamentales.

- ✓ Trabaja con el patrón de diseño Modelo Vista Controlador (MVC, Model View Controller)
- ✓ 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 desde la base de datos, sin tener que escribir ninguna consulta SQL.
- ✓ Una solución para el acceso a bases de datos que balancea el ORM con eficiencia y simplicidad.
- ✓ Completa documentación y pruebas de alta calidad.
- ✓ Soporte avanzado para i18n (internacionalización).
- ✓ Robustas clases para autenticación y filtrado de entrada.
- ✓ Clientes para servicios web, incluidos Google Data APIs y Strikelron. (19)

1.3.5 SERVIDOR WEB

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. (20)

APACHE

El Servidor Apache HTTP es un servidor Web de tecnología Open Source sólido y para uso comercial desarrollado por la Apache Software Foundation.

Principales funcionalidades.

- ✓ Los módulos Apache API utilizan un nuevo conjunto de interfaces de programación de aplicaciones (APIs).
- ✓ Filtrado — Los módulos pueden actuar como filtros de contenido.

- ✓ Soporte a IPv6.
- ✓ Directrices simplificadas — Se han eliminado una serie de directrices complicadas y otras se han simplificado.
- ✓ Respuestas a errores en diversos idiomas — Cuando usan documentos Server Side Include (SSI), las páginas de errores personalizables se pueden entregar en diversos idiomas. (21)

INTERNET INFORMATION SERVER (IIS)

El servidor web de Microsoft, el IIS (Internet Information Server), es el motor que ofrece esta compañía a modo profesional, con él es posible programar en ASP (Active Server Pages, Páginas de Servidor Activo) las cuales vienen a ser algo similares al PHP, este servidor posee componentes programables desde ASP accediendo a cada uno de sus módulos para una función específica. (22)

1.3.6 GESTOR DE BASES DE DATOS

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

MYSQL

MySQL es un gestor de base de datos sencillo de usar e increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es libre para aplicaciones no comerciales.

Las características principales de MySQL son:

- ✓ Es un sistema de base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.

- ✓ Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet. (23)

POSTGRESQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) Open Source. Es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Principales características:

- ✓ Implementación del estándar SQL92/SQL99.
- ✓ Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
- ✓ Incorpora una estructura de datos array.
- ✓ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. (24)

1.3.7 METODOLOGÍA DE DESARROLLO DE SOFTWARE Y ESTÁNDARES

Las metodologías de desarrollo de software son un conjunto de métodos, procedimientos, técnicas que guían paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. (25)

En dependencia de las necesidades para desarrollar el software se pueden utilizar varios enfoques teniendo en cuenta la disponibilidad de los recursos y el tiempo de ejecución.

1.3.7.1 METODOLOGÍAS TRADICIONALES O PESADAS

Imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. (26)

RUP. RATIONAL UNIFIED PROCESS

Es un proceso para el desarrollo de un producto software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto.

Presenta tres características esenciales:

- ✓ **Dirigido por los Casos de Uso:** Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.
- ✓ **Centrado en la arquitectura:** relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden.
- ✓ **Iterativo e incremental:** Este proceso consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

Organiza el ciclo de vida de un producto en 4 fases:

- ✓ **Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- ✓ **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- ✓ **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios release⁴ del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.
- ✓ **Transición:** El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

En cada una de estas fases se trabaja iterativamente desarrollando flujos de trabajo, uno más que otros en dependencia de la fase, dichos flujos están agrupados en:

Disciplinas de Ingeniería:

- ✓ **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- ✓ **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ✓ **Análisis y Diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- ✓ **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y las ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ✓ **Prueba:** Busca los defectos a lo largo del ciclo de vida.

⁴ Versión entregable

- ✓ **Despliegue:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

Disciplinas de Soporte:

- ✓ **Gestión de Configuración y Cambios:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ✓ **Gestión de Proyectos:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- ✓ **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización. (27)

1.3.7.2 METODOLOGÍAS ÁGILES

Las metodologías ágiles de desarrollo de software promueven un conjunto de prácticas que permiten enfrentar de mejor manera la incertidumbre y adecuarse rápidamente a un mundo cambiante. El dinamismo propio de la industria hace que esta metodología sea una alternativa interesante para los desarrolladores. En su mayoría, todo lo referente a desarrollo ágil se encuentra muy ligado al contacto permanente con el cliente, al feedback⁵ o al contacto con miembros del equipo de desarrollo. (28)

EXTREME PROGRAMMING (XP)

Extreme Programming (XP) es una disciplina para el desarrollo de software basada en los valores de simpleza, comunicación, respeto, feedback y coraje. Reúne al equipo completo junto a prácticas simples, con el feedback suficiente para permitirle al equipo ver en dónde está y ajustar las prácticas a su situación única. (29)

Sustentados en estos 5 valores los proyectos XP deben tener muy presentes 24 prácticas valiosas, que se complementan unas con otras y que ofrecen una base sólida para un óptimo desempeño, alta productividad e inestimables beneficios. Entre ellas se destacan:

⁵ Retroalimentación

- ✓ Pruebas antes de programar.
- ✓ Diseño incremental.
- ✓ Ciclo semanal.
- ✓ Integración continua.
- ✓ Programación en parejas.
- ✓ Implicación real del cliente.

El ciclo de vida ideal consta de 6 fases:

- ✓ **Exploración:** En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.
- ✓ **Planificación de Entregas:** En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.
- ✓ **Iteraciones:** Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas.
- ✓ **Producción:** La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.
- ✓ **Mantenimiento:** Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente.
- ✓ **Muerte:** Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. (30)

1.3.8 HERRAMIENTAS CASE

Se pueden definir a las herramientas CASE (Computer-Aided Software Engineering) 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. Como es sabido, los estados en el Ciclo de Vida de desarrollo de un Software son: Investigación Preliminar, Análisis, Diseño, Implementación e Instalación.

VISUAL PARADIGM

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, obtener código y documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Se encuentra soportada en todos los sistemas operativos aunque precisa de licencia la universidad tiene acceso a la misma. (31)

RATIONAL ROSE

Rational Rose es una aplicación que fue diseñada para el desarrollo de arquitectura de base del software. Está basada en modelos que se integran con las bases de datos y los IDE de las principales plataformas del sector. Esta aplicación es un entorno de modelado que permite generar código desde los modelos Ada, ANSI C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Por lo que al igual que todos los productos de Rational Rose, brinda un lenguaje de modelado común que acelera la creación del software. (32)

1.3.9 HERRAMIENTAS DE DESARROLLO

NETBEANS IDE (ENTORNO DE DESARROLLO INTEGRADO)

El IDE NetBeans es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y

móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++.

Principales características:

- ✓ Lenguajes Web: HTML, CSS, JavaScript
 - Reestructuración y búsqueda de usos para CSS y lenguajes parecidos a HTML.
 - Autocompletado y enlace para atributos id y class.
 - Reestructuración de estilos en línea de CSS.
- ✓ PHP
 - Compatibilidad con PHP Zend Framework.
 - Compatibilidad con PHP Symfony.
- ✓ General
 - Corrector ortográfico en el Editor.
 - Brinda reportes sobre fallos.
 - Compatibilidad para diversos servidores de trabajo en equipo, basados en Kenai (por ejemplo: kenai.com y netbeans.org). (33)

ECLIPSE

Eclipse es una plataforma de software de código abierto independiente y que es y ha sido muy utilizada para desarrollar entornos de desarrollo (IDE), pero del mismo modo se puede usar para otros tipos de aplicaciones cliente.

Características fundamentales:

- ✓ Tiene una gran comunidad de usuarios extendiendo constantemente las aplicaciones.
- ✓ Editor sensible al contexto, el cual provee de resaltamiento y autocompletado de código.
- ✓ Permite inspeccionar el uso de las vistas del contorno del fichero y del proyecto, así como la nueva vista PHP Explorer.
- ✓ Soporte para el debug incremental del código de PHP.
- ✓ Extensos frameworks y APIs que permiten a los desarrolladores e ISVs (vendedores de software independientes) fácilmente extender PDT para crear nuevas e interesantes herramientas orientadas al desarrollo de PHP. (34)

1.3.10 SELECCIÓN DE LAS TECNOLOGÍAS Y HERRAMIENTAS

Para resolver el problema se decide utilizar la tecnología Web por permitir a los usuarios el acceso con el único requerimiento de un navegador Web instalado, por la manera en que se centraliza la información y no necesita ser actualizado constantemente. Las aplicaciones Webs pueden ser accedidas por una mayor cantidad de usuarios y brindan seguridad para acceder a los datos.

Por tanto, para llevar a cabo satisfactoriamente el proceso de selección de las herramientas y tecnologías se priorizaron herramientas que cuentan con gran auge a nivel mundial en el desarrollo de software, que contribuirán al desarrollo de una aplicación Web y que además están basadas sobre código abierto o perteneciente al software libre. Basado en los requisitos anteriores se obtiene lo siguiente:

Para el desarrollo de la aplicación se decide utilizar el lenguaje de programación PHP, pues se caracteriza por ser flexible, posee mucha documentación, está bajo los términos del software libre y cuenta con una gran comunidad de desarrollo en la universidad.

Acompañado de PHP se utilizará HTML para estructurar las páginas, para lograr un formato y apariencia en cada una de estas se establecerán hojas de estilos (CSS), en caso de ser necesario se añadirán comportamientos a estas páginas mediante JavaScript y para agregar un poco de interacción con el usuario se utilizará la librería JQuery 1.4.

Con PHP como punto de partida, se define seleccionar Symfony como Framework para el desarrollo de la aplicación pues es de fácil aprendizaje, genera gran cantidad de código automáticamente, agrega un alto nivel de seguridad a las aplicaciones, posee buena cantidad de información y lo más importante es libre para uso comercial.

Como servidor Web, Apache es el mejor de los candidatos ya que es uno de los más utilizados por ser una herramienta totalmente libre, además es compatible con la mayoría de los gestores de bases de datos y lenguajes del lado del servidor.

El Gestor de Bases de Datos que se escoge es PostgreSQL, pues junto a Apache y PHP forma una potente plataforma totalmente libre, robusta, multiplataforma⁶ y compatible con muchas herramientas de desarrollo.

De los sistemas de discusión (foros) no se utilizará ninguno porque a pesar de que se han ganado gran reputación a nivel internacional, no son de fácil integración con el framework seleccionado y además no se adaptan a los requerimientos de la plataforma.

Con el estudio de los clientes de correo se identificaron las principales funcionalidades que estos realizan y a partir de ello se realizará una aplicación que permita gestionar los mensajes y se integre totalmente con la Plataforma Educativa ZERA.

RUP es la escogida como metodología para el desarrollo, pues los clientes potenciales del software en cuestión no están todo el tiempo junto al equipo de trabajo y RUP a pesar de ello, se encarga de mantener al equipo enfocado a producir software de alta calidad y en el tiempo requerido.

Como herramienta CASE se decide trabajar con Visual Paradigm que utiliza UML como lenguaje de modelado muy utilizado por estos días en la comunidad de desarrollo de software, es una herramienta multiplataforma, con muchas posibilidades de uso y posee una amplia documentación.

Por la gran integración que tiene con varias tecnologías, se adopta NetBeans como entorno de trabajo ya que es totalmente libre, es multiplataforma, facilita el trabajo en equipo y consta con módulos para el trabajo con las versiones de Symfony.

Con lo antes expuesto, el perfil tecnológico queda resumido de la siguiente manera:

- ✓ **Lenguaje de Programación:** PHP 5.3
- ✓ **Framework:** Symfony 1.4.12
- ✓ **Servidor Web:** Apache 2.0
- ✓ **Gestor de Bases de Datos:** PostgreSQL 8.4
- ✓ **Metodología de desarrollo:** RUP (RATIONAL UNIFIED PROCESS)
- ✓ **Herramienta CASE:** Visual Paradigm.

⁶ Funciona en diversos Sistemas Operativos

- ✓ **Entorno de trabajo:** NetBeans IDE 6.9.

1.4 CONCLUSIONES PARCIALES

Con la realización de este capítulo se sentaron las bases sobre las cuales se realizará la solución del problema de investigación. Se profundizó en el estudio de algunas definiciones que ayudarán a un mejor entendimiento de este trabajo. Además se realizó un análisis de las herramientas y tecnologías que dominan el mercado del software, permitiendo la realización de una propuesta para llegar a la solución de la problemática.

Capítulo 2: IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se presenta una descripción de las principales funcionalidades de los subsistemas pertenecientes al módulo Comunicaciones partiendo del diseño propuesto por el analista de sistema. Se realizan los diagramas de componentes correspondientes al módulo Comunicaciones, con el objetivo de lograr una mejor comprensión de la estructura de los subsistemas Foro y Mensajería Interna. Para identificar el hardware utilizado en la implementación se establece el diagrama de despliegue. Se analiza el estilo de programación empleado en el desarrollo del módulo, así como los estándares de codificación.

2.1 DESCRIPCIÓN DE LOS SUBMÓDULOS

El módulo Comunicaciones consta con un submódulo dedicado a la gestión de foros y otro para la gestión de los mensajes entre usuarios en la Plataforma. Para lograr interactuar de manera directa con el módulo Comunicaciones se hace necesario que los usuarios se registren previamente en el sistema. En lo que sigue se da una breve explicación del funcionamiento de módulo Comunicaciones.

2.1.1 SUBMÓDULO FORO

En el submódulo Foro además de gestionar foros, temas y comentarios, se realizan un conjunto de tareas que pueden ser de forma automática o mediante acciones del usuario sobre la aplicación. A continuación se describirán brevemente algunas de estas tareas y/o acciones sobre el foro.

Al crearse un grupo se le asocia un profesor, un programa de estudio y una cantidad de estudiantes, a partir de la creación del grupo se crea un foro automáticamente para ese grupo. En caso de eliminarse el grupo de la escuela, automáticamente se borra el foro asociando al grupo. Para el acceso al foro cada rol en la aplicación y de acuerdo a los permisos que posea, podrá realizar acciones sobre los foros, temas y comentarios realizados. El submódulo foro permitirá a los usuarios adjuntar ficheros en sus comentarios, tendrá opciones para el cierre y bloqueo de temas y/o foros.

2.1.2 SUBMÓDULO MENSAJERÍA INTERNA

El submódulo Mensajería Interna permite la creación, la eliminación, el envío y recibo de mensajes. Además brinda las opciones de guardar y editar borradores, en este submódulo se gestionan listas de distribución con utilidad en el envío de mensajes de forma masiva. Se puede decir que la Mensajería

tiene una estrecha relación con la mayoría de los subsistemas de la Plataforma, pues al realizarse una acción en cualquiera de estos subsistemas se notifica al usuario o usuarios implicados mediante el uso de mensajes.

2.2 REQUISITOS DEL SOFTWARE.

Los requerimientos del software son las características y cualidades que el sistema debe tener. Estos se dividen en dos grupos, los funcionales y los no funcionales (35). A continuación se muestran los requisitos obtenidos por los analistas de sistema.

2.2.1 REQUISITOS FUNCIONALES

Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

- ✓ **RF 1: Gestionar Foro.**
 - RF 1.1: Crear Foro.
 - RF 1.2: Eliminar Foro.
 - RF 1.3: Ver Foro.
- ✓ **RF 2: Bloquear Foro.**
- ✓ **RF 3: Gestionar Temas de Foro.**
 - RF 3.1: Incluir un nuevo Tema.
 - RF 3.2: Ver los datos de un Tema.
 - RF 3.3: Modificar los datos de un Tema.
 - RF 3.4: Eliminar un Tema.
- ✓ **RF 4: Listar Temas de Foro.**
- ✓ **RF 5: Bloquear Tema.**
- ✓ **RF 6: Cerrar Tema.**
- ✓ **RF 7: Gestionar Comentarios del Foro.**
 - RF 7.1: Incluir nuevo Comentario.
 - RF 7.2: Ver Comentario.
 - RF 7.3: Eliminar Comentario.
 - RF 7.4: Modificar Comentario.

- ✓ **RF 8: Ver Comentarios por usuario.**
- ✓ **RF 9: Adjuntar Ficheros al Foro.**
- ✓ **RF 10: Ver Ficheros adjuntos.**
- ✓ **RF 11: Gestionar Correspondencia Interna.**
 - RF 11.1: Incluir Correspondencia Interna.
 - RF 11.2: Modificar los datos de una Correspondencia Interna.
 - RF 11.3: Eliminar Correspondencia Interna.
- ✓ **RF 12: Ver contenido de la Carpeta Bandeja de Entrada.**
- ✓ **RF 13: Ver contenido de la Carpeta Borradores.**
- ✓ **RF 14: Ver contenido de la Carpeta Papelera.**
- ✓ **RF 15: Ver contenido de la Carpeta Enviados.**
- ✓ **RF 16: Crear Lista de Contactos.**
- ✓ **RF 17: Ver Lista de Contactos.**

2.2.2 REQUISITOS NO FUNCIONALES

Los requisitos no funcionales se definen como las cualidades o propiedades que el software debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. (35)

Requisitos de seguridad

- ✓ 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 plataforma.
- ✓ 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 1.4.12.
- ✓ RNF 12: Como IDE se empleará NetBeans 6.9.
- ✓ RNF 13: Como servidor Web se explotará Apache 2.2.16.
- ✓ RNF 14: El Sistema Gestor de Bases de Datos deberá ser PostgreSQL 8.4.
- ✓ RNF 15: Sistemas Operativos Windows y Linux.

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 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 a medida que el usuario interactúa con la plataforma.
- ✓ 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.

2.3 ESTÁNDARES DE CODIFICACIÓN

Es muy necesario establecer estándares de codificación para todos los programadores. Estos estándares consisten en estilos de codificación a la hora de escribir líneas de código. Los aspectos fundamentales para los que generalmente se establecen estándares son los que siguen:

- ✓ Identificadores.
- ✓ Líneas y espacios en blanco.
- ✓ Comentarios.

En cada grupo de desarrollo se definen los principales aspectos a estandarizar y cuales se aplicarán a cada uno de ellos. El cumplimiento de estándares hace que todo el código sea lo más parecido a una firma, es decir se puede deducir a partir del código su desarrollador, por tanto, con un grupo de desarrolladores con un estándar bien definido se logra un código legible, parcialmente parecido y con facilidades para el mantenimiento. A continuación se presentan los estándares a utilizar en la implementación de este trabajo. Cabe destacar que estos estándares se definen teniendo en cuenta el estilo del programador, las características propias del lenguaje de programación, los recursos que utilizarán y el tipo de programa que se desea implementar.

2.3.1 IDENTIFICADORES

Se considera como identificador a los nombres de las variables, funciones y/o constantes definidas por el programador. Dichos identificadores deberán seguir las siguientes normas, además de las definidas por el propio lenguaje.

- ✓ **Funciones y métodos:** Se deben nombrar usando el formato lowerCamelCase (notación camello).
- ✓ **Clases:** Los nombres de las clases personalizadas comienzan con sf seguidos del formato UpperCamelCase. Las clases del modelo siguen el formato UpperCamelCase y las controladoras el formato lowerCamelCase.
- ✓ **Otros:** Los nombres deben ser cortos y descriptivos, facilitando el entendimiento del código generado. Se deben utilizar comentarios en todos los casos que sean normados y en caso de que el programador lo considere necesario.

```
/* Nombre de Clases*/  
class TbForum extends BaseTbForum {  
    /* Nombre de Funciones*/  
    public function getCantidadComentarioTable(){  
        /* Nombre de Variables*/  
        $cantidadVisitas = 0;  
    }  
}
```

2.3.2 LÍNEAS Y ESPACIOS EN BLANCO

Se deben utilizar dos (2) espacios en blanco en vez de utilizar el TABULADOR (8 espacios).

- ✓ **Longitud de la Línea:** Evitar dejar líneas con más de 80 caracteres.
- ✓ **Líneas plegadas:** Cuando una expresión excede de los 80 caracteres se hace necesario dividirla en más de una línea, siguiendo las siguientes precisiones:
 - Dividir después de una coma.
 - Dividir después de un operador.
 - Alinear la nueva línea al inicio de la expresión en el mismo nivel que la línea anterior.
 - Si las reglas anteriores hacen que el código sea confuso, o son confusas de codificar simplemente utilice 8 espacios como alineación.

2.3.3 COMENTARIOS

Además de los comentarios que genera Symfony, se establecen 4 estilos de comentarios: bloque, línea simple, seguimiento y fin de línea.

- ✓ **Bloque:** Se deben utilizar cuando se desea describir ficheros, métodos, estructuras, etc.
- ✓ **Línea:** Se deben utilizar cuando se desea poner solamente en una línea un comentario.
- ✓ **Seguimiento:** Se deben utilizar cuando en una línea existe código y seguido en la misma línea se desea establecer un comentario.
- ✓ **Fin de línea:** Se utiliza cuando se quieren comentar líneas completas.

```
/* Comentarios
 * Este es de Bloque
 * @author btejera@ */
public function getCantidadComentarioTable() {
    /* Este comentario es de línea */
    $query = Doctrine_Query::create()
    //         ->from('TbMessageTopic mt') Fin de línea
        ->where('mt.forum_id = ?', $this->getId())
    return count($query->execute()); //Se seguimiento
}
```

2.4 PATRONES UTILIZADOS EN LA IMPLEMENTACIÓN

Existen buenas prácticas en el diseño arquitectural, especialmente en cuanto a la arquitectura lógica a gran escala, y estas prácticas se han escrito en forma de patrones. Muchos de ellos ayudan en la implementación de aspectos particulares del diseño en un específico lenguaje de programación. A continuación se muestran los principales patrones que guían y hacen posible estructurar los elementos de Symfony, entre estos patrones se pueden encontrar los de diseño y los arquitectónicos.

2.4.1 PATRONES ARQUITECTÓNICOS

Los patrones arquitectónicos están relacionados con la interacción de objetos dentro o entre niveles arquitectónicos. Proporcionan solución a problemas arquitectónicos, adaptabilidad a requerimientos

cambiantes, rendimiento, modularidad, acoplamiento. Symfony implementa en su estructura la combinación de dos patrones arquitectónicos, a continuación se describen sus principales características.

- ✓ **Patrón MVC:** Este patrón sugiere dividir el código en varios componentes. El código de la presentación se guarda en la vista, el código de manipulación de datos se guarda en el modelo y la lógica de procesamiento de las peticiones constituye el controlador. (36)
- ✓ **Patrón Front Controller (Controlador Frontal):** Este patrón establece un controlador que se encarga de gestionar todas las peticiones web de la aplicación, este controlador frontal recibe todas las peticiones entrantes de los clientes, remitiendo a su vez cada petición al gestor de peticiones adecuado, que se encargará de gestionar la construcción de una respuesta adecuada al cliente.

2.4.2 PATRONES DE DISEÑO

Los patrones de diseño fueron creados con el propósito de eliminar problemas con la claridad de diseño, multiplicación de clases, adaptabilidad de requerimientos cambiantes, proponiendo una solución al comportamiento de factoría y Clase-Responsabilidad-Contrato (CRC). Dentro de estos patrones se pueden encontrar dos grupos: los GRASP y los GoF, seguidamente se definen los que guían la implementación de la solución.

2.4.2.1 PATRONES GRASP

Los Patrones GRASP representan los principios básicos de la asignación de responsabilidades a objetos. GRASP es el acrónimo para General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades).

- ✓ **Experto:** Es uno de los más utilizados con Symfony, con la inclusión de la librería Doctrine para mapear la Base de Datos. Symfony utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades, las clases de abstracción de datos que poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan.

- ✓ **Alta Cohesión:** Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Este patrón se evidencia en las clases Actions de los submódulos, las cuales están constituidas por varias funcionalidades que están estrechamente relacionadas, además estas clases también son las responsables de definir las acciones para las plantillas y colaboran con otras clases para realizar diferentes operaciones.
- ✓ **Controlador:** Todas las peticiones Web son manipuladas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en las clases sfFrontController, sfWebFrontController, sfContext, los “actions” y el index.php del ambiente.

2.4.2.2 PATRONES GOF

Los patrones GoF (Gang of Four, en español Pandilla de los Cuatro) se clasifican en tres categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

Creacionales:

- ✓ **Singleton:** Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En las acciones se usan los métodos ->getRequest (), ->getUser (), esto se debe a que, en la acción, el método getContext (), guarda una referencia a todos los objetos del núcleo de Symfony, estos métodos pueden ser accedidos desde la vista y desde el controlador, solo varía la forma de llamarlos.

Estructurales:

- ✓ **Decorator:** Añade funcionalidad a una clase, dinámicamente. En cada archivo layout.php se define el código HTML común de cada vista, evitando que este sea repetido en cada página.

2.5 DESCRIPCIÓN DE LAS PRINCIPALES CLASES Y FUNCIONALIDADES EMPLEADAS EN EL DESARROLLO DEL SISTEMA

Una clase por lo general, es un modelo dentro de un programa que encapsula el estado y el comportamiento de lo que representa. Encapsula el estado a través de marcadores llamados atributos y comportamiento a través de secciones de código reutilizable llamados métodos. Para llegar a la solución

del sistema se utilizaron un conjunto de clases y métodos, los principales y/o de mayor complejidad se presentan en lo que sigue.

2.5.1 CLASES DEL MODELO

Estas clases son generadas automáticamente por Symfony en conjunto con el ORM Doctrine y sus atributos son los identificadores de las columnas de las tablas en la base de datos y entre sus principales métodos se pueden encontrar los que permiten modificar cada uno de estos atributos (ej. `setAtributo($valor)`) y los que devuelven los valores de cada uno de los atributos (ej. `getAtributo()`), además se pueden encontrar otros métodos auxiliares que posibilitan el manejo de los datos. Cada una de estas clases permite definir nuevas funcionalidades así como modificar las existentes. Las principales clases utilizadas en la solución son:

- ✓ **Submódulo Foro:**
 - TbForum
 - TbTopic
 - TbMessageTopic
 - TbAttachment
- ✓ **Submódulo Mensajería Interna:**
 - TbMessage
- ✓ **General**
 - SfGuardUser
 - TbGroup
 - TbStudyProgram
 - TbSchool
 - TbStudent

Por su gran similitud en sus estructuras se citará un solo ejemplo, aunque muchos de estos métodos se definen en sus clases padres, por tanto pueden utilizarse cualquiera de sus hijas.

Clase	Propiedades	Métodos 2
TbAttachment	id attachment message_topic_id	setId(\$value) setAttachment(\$value) setMessageTopicId(\$value) getId() getAttachment() getMessageTopicId()

TABLA 1. CLASE DEL MODELO

En el modelo existen otras clases que sirven de apoyo a las clases mencionadas anteriormente, pues son las que deben interactuar con los datos mediante el ORM Doctrine y al final del nombre se le agrega la cadena “Table”. Los métodos de estas clases generalmente devuelven arreglos de objetos Doctrine y se utilizan en las clases principales del modelo. Para comprender mejor su funcionamiento se cita el siguiente ejemplo:

```

/*En la clase TbForumTable.php*/
public function getCantidadComentarioTable() {
    $query = Doctrine_Query::create()
        ->from('TbMessageTopic mt')
        ->where('mt.forum_id = ?', $this)
    return count($query->execute());
}

```

Con este método se obtienen los datos, en este caso la cantidad de comentarios de un foro y para poder llamarlo desde una instancia de la clase se debe definir un método dentro su clase principal en el modelo.

```

/* En la clase TbForum.php*/
public function getCantidadComentario() {
    $cantidad = Doctrine::getTable('TbForum')
        ->getCantidadComentarioTable();
    return $cantidad;
}

```

Ahora a partir de cualquier instancia de la clase TbForum se puede obtener la cantidad de comentarios. Las demás clases del modelo se rigen por la misma estructura.

2.5.2 CLASES CONTROLADORAS

Estas clases tienen como misión fundamental comunicarse con el modelo para obtener la información solicitada por el usuario y pasarla a la vista para visualizarla. En la solución propuesta se utilizan estas clases siguiendo la arquitectura que propone Symfony, seguidamente se explicarán las mismas y sus principales funcionalidades.

2.5.2.1 SUBMÓDULO FORO

forumActions: Clase controladora que se encarga de realizar todas las acciones sobre los foros, temas y comentarios. Para realizar estas acciones la clase cuenta con las siguientes funcionalidades:

executeCrearTema(sfWebRequest \$request): Función utilizada para crear un tema a partir de la creación de un foro, se valida que el tema creado pertenece al mismo programa de estudio que el foro, al crearse dicho tema se crea un comentario que sirve como guía para la discusión.

executeEditTopic(sfWebRequest \$request): Partiendo de la selección del tema a editar por parte del usuario, la función busca si ese tema existe en la base de datos, si existe manda la información a la vista correspondiente (editTopicSuccess.php), el usuario llena un formulario y se envía a esta funcionalidad, se valida si los datos son correctos y se actualiza los campos en la base de datos.

executeCrearMensaje(sfWebRequest \$request): Esta funcionalidad permite al usuario a partir de una lista de comentarios crear uno nuevo, adjuntarle un fichero y validarlo.

executeForo(sfWebRequest \$request): Partiendo del usuario conectado esta función busca el foro al que tiene acceso mediante el grupo y programa de estudio, también busca la lista de temas que pertenece al foro previamente seleccionado y manda esta lista por secciones es decir paginada.

executeTema(sfWebRequest \$request): Esta función depende del tema seleccionado para poder buscar los datos de los comentarios que pertenecen a dicho tema y enviarlos a la vista divididos en secciones.

executeForoMensajesUsuario(sfWebRequest \$request): Esta función está enfocada específicamente a los moderadores del foro, y permite a partir del identificador de un participante del foro ver todos los comentarios que ha realizado.

2.5.2.2 SUBMÓDULO MENSAJERÍA INTERNA

messagingActions: Clase controladora que se encarga de realizar todas las acciones sobre los mensajes, notificaciones y listas de distribución. Para realizar estas acciones la clase cuenta con las siguientes funcionalidades:

executeNotificacion(sfWebRequest \$request): Esta función se encarga de buscar los mensajes que tiene sin leer del usuario conectado, estas peticiones se realizan en un pequeño intervalo de tiempo para mantener actualizado los datos en la vista.

executeIndex(sfWebRequest \$request): A partir del identificador del usuario conectado la función busca los mensajes relacionados con ese usuario para poderlos ubicar en la vista con las clasificaciones correspondientes (ej. Bandeja de Entrada, Borrador, Enviados, etc.)

executeShow(sfWebRequest \$request): Partiendo del identificador de un mensaje, esta función lo busca en la base de datos y para mostrarlo en la vista con sus diferentes opciones (ej. Eliminar, Reenviar, Guardar).

executeTrash(sfWebRequest \$request): Esta función permite buscar todos los correos borrados por el usuario.

executeSend(sfWebRequest \$request): Función que se encarga de guardar el mensaje en la base de datos y establecer la relación con sus respectivos usuarios.

executeDraft(sfWebRequest \$request): Esta función permite buscar todos los correos guardados por el usuario. Es decir son todos los mensajes de tipo borrador.

executeResponse(sfWebRequest \$request): Función que permite responder al remitente los mensajes que ha enviado.

executeSearch(sfWebRequest \$request): Función que permite el autocompletado de los usuarios o listas de distribución para el envío de mensajes.

executeAccion(sfWebRequest \$request): Función que permite el cambio de estado de los mensajes y asociarle acciones que puede realizar en cada uno de los estados.

executeCreateDistributionList(sfWebRequest \$request): Partiendo de una lista de usuarios esta función permite asociarlos en un identificador.

2.6 MODELO DE IMPLEMENTACIÓN

El modelo de implementación denota la programación de la aplicación en términos de componentes y subsistemas de implementación. Describe la organización de los datos, archivos, ejecutables, código fuente y directorios de acuerdo a los mecanismos de estructuración disponibles en el entorno de implementación. Fundamentalmente, se describen las relaciones que existen entre los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. En este artefacto se describe cómo se implementan los componentes que lo conforman, acoplándolos en subsistemas organizados en capas y jerarquías, señalando las dependencias entre estos.

2.6.1 DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue permite modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

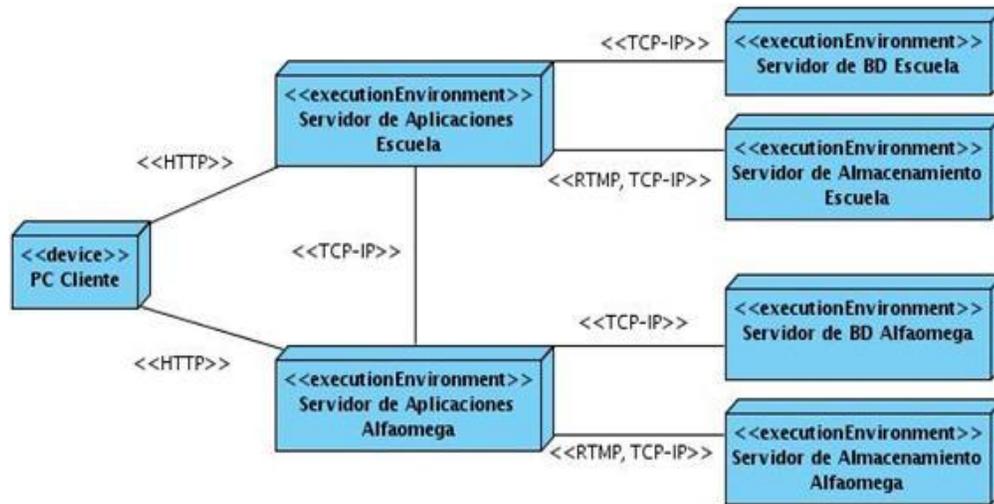


ILUSTRACIÓN 1 DIAGRAMA DE DESPLIEGUE

2.6.2 DIAGRAMA DE COMPONENTES

El diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, módulos, ejecutables, o paquetes.

2.6.2.1 SUBMÓDULO FORO

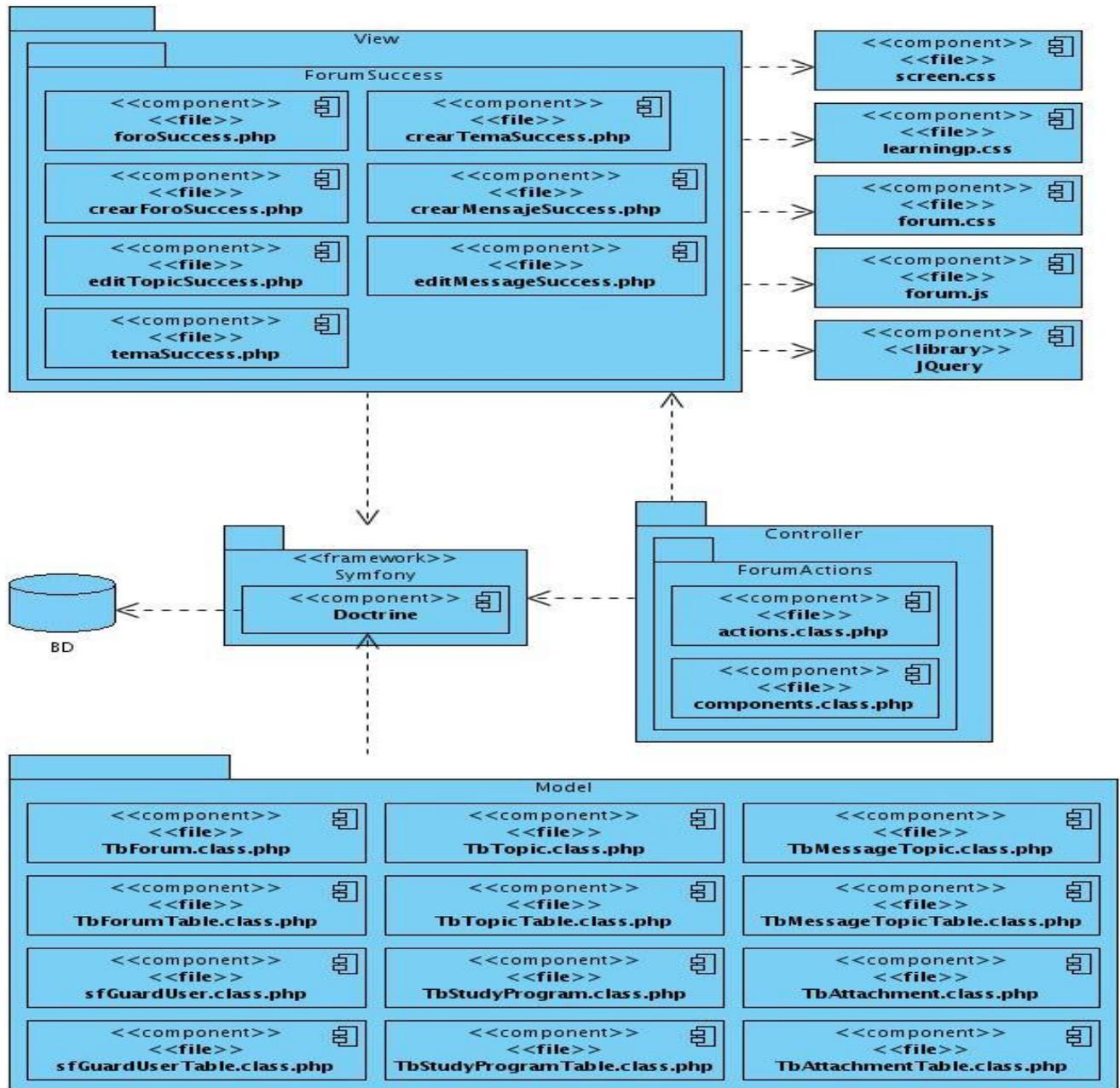


ILUSTRACIÓN 2 DIAGRAMA DE COMPONENTES FORO

2.6.2.2 SUBMÓDULO MENSAJERÍA INTERNA

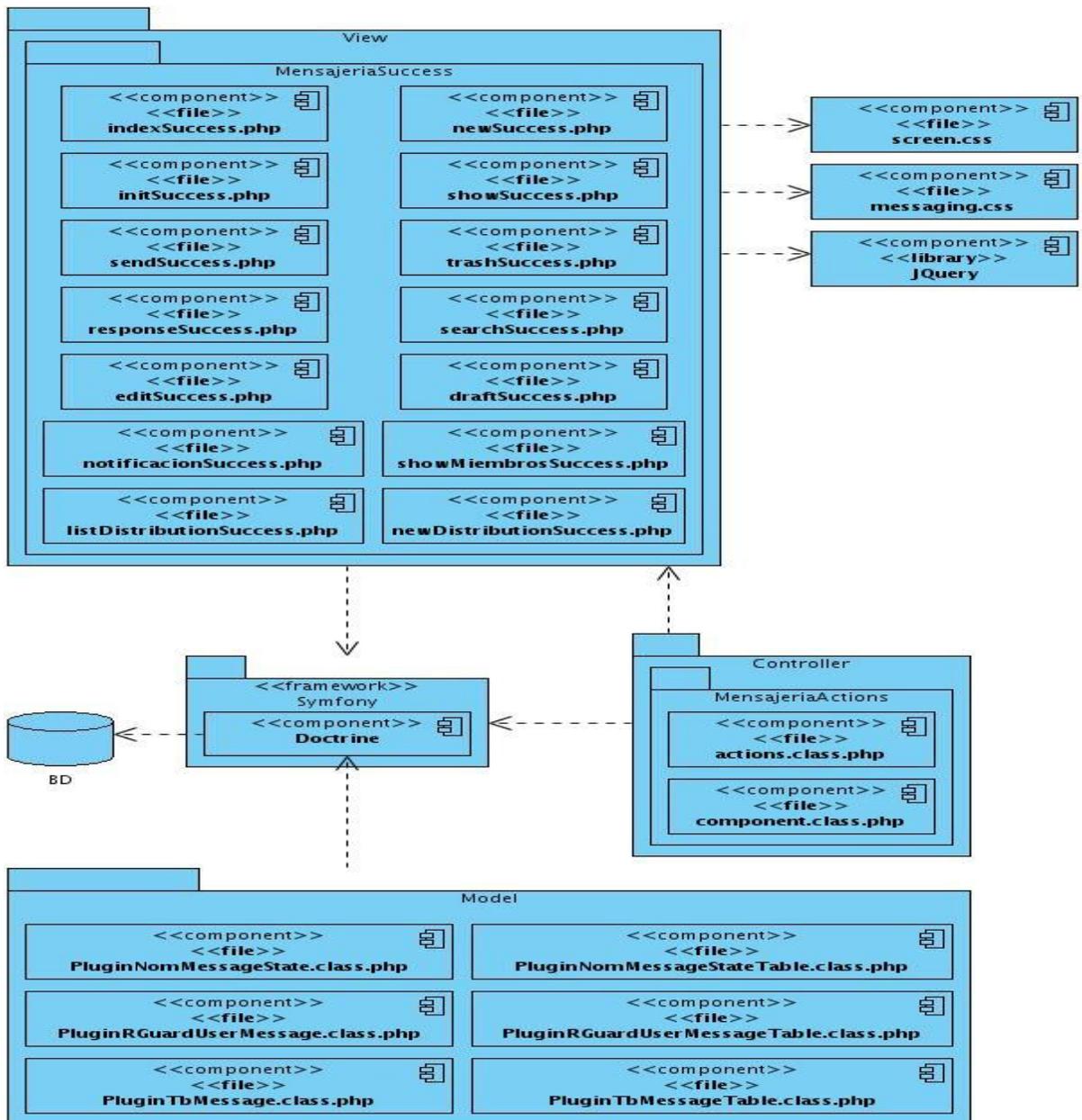


ILUSTRACIÓN 3 DIAGRAMA DE COMPONENTES MENSAJERÍA

2.7 CONCLUSIONES PARCIALES

Partiendo de los artefactos generados por los analistas de sistema, en este capítulo se hizo una breve descripción del funcionamiento del módulo Comunicaciones, se definieron pautas que permitieron la implementación de un sistema funcional y por último se plasmaron los elementos utilizados en diagramas de componentes y de despliegue.

Capítulo 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

La realización de pruebas es un elemento crítico para garantizar la calidad del software. Es una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos específicos. En este capítulo se hace un estudio sobre las pruebas existentes y se hará uso de la más adecuada a la solución propuesta.

3.1 PRUEBAS DEL SOFTWARE

Las Pruebas son aplicadas en las diferentes fases del desarrollo de software, con el objetivo de ir limando los posibles errores a medida que el desarrollo avanza.

3.1.1 NIVELES DE PRUEBA

Existen pruebas que varían en dependencia del alcance y complejidad, por tal motivo se definen en varios niveles. A continuación se muestran los niveles de prueba a utilizar.

- ✓ **Pruebas de desarrolladores:** Pruebas diseñadas e implementadas por el equipo de desarrollo. Tradicionalmente estas pruebas han sido consideradas solo para la prueba de unidad, aunque en la actualidad en algunos casos se pueden ejecutar pruebas de integración. Se recomienda que estas pruebas cubran más que las pruebas de unidad.
- ✓ **Pruebas independientes:** Pruebas diseñadas e implementadas por alguien independiente del grupo de desarrolladores. El objetivo de estas pruebas es proporcionar una perspectiva diferente y en un ambiente más rico que el de los desarrolladores.
- ✓ **Pruebas unitarias:** Se encargan de probar una clase en concreto, testeando cada uno de sus métodos y tratan de validar si dados unos parámetros de entrada, la salida es la esperada.
- ✓ **Pruebas funcionales:** Como su propio nombre indica, prueban una funcionalidad completa, donde pueden estar implicadas una o varias clases, la propia interfaz de usuario.
- ✓ **Prueba de aceptación:** Son las pruebas que se realizan antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por los usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

- ✓ **Pruebas de integración:** Conjunto de pruebas unitarias, funcionales y/o de aceptación que se realizan para probar el software. Incluye también comprobar que lo programado por los diferentes desarrollados no entra en conflicto y que funcionará en un entorno real.

3.1.2 MÉTODOS DE PRUEBA

Actualmente existen métodos que posibilitan la validación de software y los mismos tienen diferentes enfoques, los cuales se dividen en dos grupos: las pruebas de caja blanca y las pruebas de caja negra.

Las pruebas de Caja Blanca se nombran de esta forma porque revisan la parte interna del software, específicamente sobre el código fuente. Se basan en el examen minucioso de los detalles procedimentales. Se comprueban los caminos lógicos del sistema generando casos de prueba que ejerciten las estructuras condicionales y los bucles. Existen varios métodos que analizan diferentes partes del software y se complementan entre sí para garantizar la calidad del sistema.

Las Pruebas de Caja Negra se basan en los requerimientos funcionales del sistema y se llevan a cabo desde el exterior de la aplicación. Estas pruebas permiten medir el grado de cumplimiento de los requerimientos solicitados por el cliente y se aplican sobre la interfaz de la aplicación observando las respuestas del sistema antes determinadas acciones. Estas pruebas se realizan mediante casos de prueba los cuales tienen como objetivo principal demostrar que las funcionalidades de los submódulos son correctas.

Para validar la solución propuesta solamente se utilizarán las pruebas de caja negra, pues con las pruebas de caja blanca se hace muy engorroso abarcar todo el código fuente en un corto intervalo de tiempo y además su aplicación se torna compleja si los algoritmos son de complejidad media o superior.

3.1.3 DISEÑO DE CASOS DE PRUEBA

A continuación se muestran algunos de los casos de prueba utilizados para validar la solución propuesta, los restantes están disponibles en la documentación generada en el desarrollo de la plataforma ZERA.

SUBMÓDULO FORO

Caso de Prueba: Ver Foro.

Descripción general: El caso de uso se inicia cuando el Usuario de la Escuela decide realizar la acción de ver un foro, activar o desactivar un foro, el sistema muestra los datos del Foro y brinda la posibilidad de activarlo o desactivarlo según sea el caso. Terminando el caso de uso.

Condiciones de ejecución

- ✓ Debe haberse generado el escritorio de trabajo del usuario autenticado.
- ✓ Debe estar creado el Foro.
- ✓ Para Ver los datos de un Foro, debe estar seleccionado previamente por el actor.
- ✓ Para activar o desactivar un foro el usuario debe tener los permisos necesarios para hacerlo.

ID del escenario	Escenario	Respuesta del sistema	Flujo central
EC 1	Ver Foro	<p>Según el usuario autenticado el sistema muestra para:</p> <p><i>Rol estudiante:</i></p> <ul style="list-style-type: none"> • Se muestran los programas de estudio a los cuales pertenece. <p><i>Rol docente:</i></p> <ul style="list-style-type: none"> • Se muestran sus programas de estudios y los grupos asociados a cada uno. <p><i>Rol director:</i></p> <ul style="list-style-type: none"> • Se muestran todos los programas de estudios y todos los foros de cada programa de estudio. <p><i>Rol administrador local:</i></p> <ul style="list-style-type: none"> • Se muestran todos los programas de estudios y todos los foros de cada programa de estudio. <p><i>Rol tutor:</i></p>	<p>Selecciona un programa de estudio</p> <p>Selecciona un Foro para ver sus datos</p>

		<ul style="list-style-type: none"> • Se muestra el nombre de los estudiantes asociados a él y los foros a los que tiene acceso cada estudiante. 	
	El actor selecciona un programa de estudio.	Muestra los datos del Foro correspondiente a ese programa de estudio. Y permite: <ul style="list-style-type: none"> • Crear un nuevo tema. • Modificar un tema. • Eliminar un tema. • Activar o desactivar el foro. • Salir de la vista actual. 	
	Selecciona la opción de salir de la vista actual.	Muestra la vista anterior. El caso de uso termina.	
EC2	El actor selecciona un grupo dentro de un programa de estudio	Muestra los datos del Foro correspondiente al grupo seleccionado. El caso de uso termina	Selecciona un grupo dentro de un programa de estudio.
EC3	El actor selecciona un Foro en un programa de estudio	Muestra los datos del Foro. El caso de uso termina	Selecciona un Foro en un programa de estudio
EC4	El actor selecciona un Foro de uno de los estudiantes.	Muestra los datos del Foro correspondiente al estudiante seleccionado. El caso de uso termina	Selecciona un Foro de uno de los estudiantes.
EC5	El actor selecciona la opción de eliminar un tema.	El sistema elimina el tema. <u>Ver CU Gestionar Temas de Foro.</u>	Selecciona un Foro en un programa de estudio, selecciona la opción de eliminar un tema.
EC6	El actor	El sistema permite crear un tema. <u>Ver CU Gestionar</u>	Selecciona un

	<i>selecciona la opción de crear un tema.</i>	<u><i>Temas de Foro.</i></u>	<i>Foro en un programa de estudio, selecciona la opción de crear un tema.</i>
<i>EC7</i>	<i>El actor selecciona la opción de modificar un tema.</i>	<i>El sistema permite modificar el tema. <u>Ver CU Gestionar Temas de Foro.</u></i>	<i>Selecciona un Foro en un programa de estudio, selecciona la opción de modificar un tema.</i>
<i>EC8</i>	<i>El actor selecciona la opción de activar o desactivar el foro.</i>	<i>El sistema permite activar o desactivar el foro actual.</i>	<i>Selecciona un Foro en un programa de estudio, selecciona la opción de activar o desactivar el foro.</i>

TABLA 2 CP VER FORO

Caso de Prueba: Gestionar temas del foro.

Descripción general: El caso de uso se inicia cuando el usuario de la escuela decide realizar acciones sobre un tema del foro, estas acciones pueden ser: ver, eliminar o modificar 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 permitiendo guardar los cambios realizados. Para eliminar, el actor debe seleccionar el tema que desea eliminar y aceptar el mensaje de confirmación. El sistema permite ver una vista previa del tema y termina el caso de uso.

Condiciones de ejecución:

- ✓ Debe haberse generado el Escritorio de Trabajo del usuario autenticado.
- ✓ Para Modificar los datos de un Tema del Foro, debe estar seleccionado previamente por el actor.
- ✓ Para Ver los datos de un Tema del Foro, debe estar seleccionado previamente por el actor.
- ✓ Para eliminar un tema del Foro, debe estar seleccionado previamente y el actor debe tener el

permiso de eliminar el elemento.

SC Crear Tema del Foro

ID del escenario	Escenario	Título	Contenido	Activar / Desactivar	Respuesta del sistema	Flujo central
EC 1	Crear Tema de Foro				<p>Brinda la posibilidad de realizar las acciones:</p> <ul style="list-style-type: none"> • Ver los datos del Tema. <u>Ver Sección: "Ver datos del Tema"</u>. <p>Según el rol del usuario autenticado permitirá además:</p> <p>Rol Docente:</p> <ul style="list-style-type: none"> • Incluir un nuevo Tema al Foro. • Modificar los datos del Tema. Ver Sección: "Modificar datos del Tema". • Eliminar Tema del Foro. Ver Sección 3: "Eliminar Tema del Foro". • Establece los parámetros de participación: temporalidad, si hay restricción de veces en participación. <p>Rol Estudiante:</p> <ul style="list-style-type: none"> • Incluir un nuevo Tema al Foro. <p>Rol Director:</p> <ul style="list-style-type: none"> • Incluir un nuevo Tema al Foro. 	<p><i>Selecciona un programa de estudio</i> <i>Selecciona un Foro para ver sus datos</i></p>
EC 2	Crear Tema de Foro				<p>Brinda la posibilidad de introducir los datos siguientes:</p> <ul style="list-style-type: none"> • Título. • Contenido • Cantidad de comentarios por 	<p><i>Selecciona un programa de estudio</i> <i>Selecciona un Foro para ver sus datos</i></p>

					estudiantes. Permite: <ul style="list-style-type: none"> • Opcionalmente Activar / Desactivar. • Crear un nuevo Tema al Foro. • Cancelar la operación en cualquier momento. 	<i>Selecciona la opción de incluir un nuevo Tema al Foro.</i>
	<i>Introduce los datos del Tema Selecciona la opción crear el Tema</i>	V	V	N/A	<i>Valida los datos. Crea el Tema. Muestra un mensaje de información.</i>	
EC 3	<i>Existen datos incompletos</i>	I	V		<i>Muestra encima del campo vacío un mensaje de información</i>	
		V	I		<i>Muestra un indicador sobre los campos vacíos</i>	
		I	I			
EC 4	<i>Existen datos incorrectos</i>	I	V		<i>Muestra encima del campo incorrecto un mensaje de información</i>	
		V	I		<i>Muestra un indicador sobre los campos incorrectos</i>	
		I	I			
EC 5	<i>El actor selecciona la opción de Cancelar</i>				<i>Elimina los datos creados Muestra un mensaje de información Muestra el escritorio de trabajo</i>	
EC 6	<i>El Tema ya se encuentra creado en el Foro</i>				<i>Muestra un mensaje de error.</i>	

TABLA 3 CREAR TEMA DE FORO

SC Ver datos del tema

EC 1	<i>Ver Datos del Tema</i>				El sistema muestra los datos del tema y según el rol del usuario autenticado permitirá además: <i>Rol Docente:</i>	<i>Selecciona un Programa de estudio Selecciona un Foro para ver sus</i>
------	---------------------------	--	--	--	---	--

				<ul style="list-style-type: none"> • Cerrar un Tema de Foro (los comentarios quedan visibles, pero los usuarios no pueden realizar comentarios). • Consultar comentarios. • Emitir comentarios. • Responder comentarios. • Eliminar un comentario (en caso de ser necesario puede eliminar un comentario realizado por un estudiante, con opción a justificar la eliminación o no. Si la justifica esta información llegará al estudiante por mensajería interna). • Editar un comentario. • Bloquear o desbloquear el tema de foro (Los usuarios invitados al tema, no pueden acceder al mismo. Se mantendrá oculto hasta que sea desbloqueado). • Siempre se genera traza de los eventos. <p>Rol Director:</p> <ul style="list-style-type: none"> • Emite comentarios en los foros. • Consulta comentarios emitidos por los alumnos. • Siempre se genera traza de los eventos. • Puede cerrar un tema creado por él, por los estudiantes o docentes. • Eliminar un comentario (en caso de ser necesario puede eliminar un comentario realizado por un estudiante o un docente, con opción a justificar la eliminación o no. Si la justifica esta información llegará al estudiante o docente por mensajería 	<p><i>datos</i> <i>Selecciona un Tema del Foro para ver sus datos.</i></p>
--	--	--	--	--	---

					<p>interna).</p> <p>Rol Estudiante:</p> <ul style="list-style-type: none"> • Consulta los comentarios públicos realizados por el docente y otros estudiantes. • Emite comentarios en los foros asignados para él de acuerdo a los parámetros establecidos en cada foro. • Siempre se genera traza de los eventos. • Edita los comentarios realizados por él. • Crear nuevos temas de foro. • Cerrar temas de foro creados por él. <p>Rol Tutor:</p> <ul style="list-style-type: none"> • Consulta los comentarios de los foros de discusión asignados al estudiante con quien está relacionado. • No genera comentarios. <p>Rol Administrador Local:</p> <ul style="list-style-type: none"> • Consulta foros y sus comentarios, no emite comentarios ni tiene permiso de edición en comentarios. 	
EC 2	<i>Selecciona la opción de salir de la vista actual</i>				<i>Muestra la vista anterior</i>	<i>Selecciona un programa de estudio</i> <i>Selecciona un Foro para ver sus datos</i> <i>Selecciona un Tema del Foro para ver sus datos.</i> <i>Selecciona la opción de salir de la vista actual</i>

EC 3	El actor selecciona la opción de Editar un comentario				Brinda la posibilidad de editar el comentario El caso de uso termina.	Selecciona un programa de estudio Selecciona un Foro para ver sus datos Selecciona un Tema del Foro para ver sus datos. Selecciona la opción de editar el comentario.
EC 4	El actor selecciona la opción de Cerrar un Tema de Foro				Los comentarios quedan visibles, pero los usuarios no pueden realizar comentarios	Selecciona un programa de estudio Selecciona un Foro para ver sus datos Selecciona un Tema del Foro para ver sus datos. Selecciona la opción de cerrar el tema.
EC 5	Emitir comentarios				Permite introducir los datos del comentario Muestra la interfaz anterior	Selecciona un programa de estudio Selecciona un Foro para ver sus datos Selecciona un Tema del Foro para ver sus datos. Selecciona la opción de emitir un comentario.
EC 6	Responder comentarios				Permite introducir los datos para responder el comentario Muestra la interfaz anterior	Selecciona un programa de estudio Selecciona un

						<p>Foro para ver sus datos Selecciona un Tema del Foro para ver sus datos. Selecciona la opción de responder el comentario.</p>
EC 7	Eliminar comentario				<p>Permite eliminar el comentario del estudiante Permite justificar la eliminación del comentario De ser justificada, envía la justificación por mensajería interna al estudiante Muestra la interfaz anterior</p>	<p>Selecciona un programa de estudio Selecciona un Foro para ver sus datos Selecciona un Tema del Foro para ver sus datos. Selecciona la opción de eliminar el comentario.</p>
EC 8	Bloquear o desbloquear el tema de foro				<p>Permite bloquear o desbloquear el tema de foro (Los usuarios invitados al tema, no pueden acceder al mismo. Se mantendrá oculto hasta que sea desbloqueado).</p>	<p>Selecciona un programa de estudio Selecciona un Foro para ver sus datos Selecciona un Tema del Foro para ver sus datos. Selecciona la opción de bloquear o desbloquear el tema de foro.</p>

TABLA 4 SC VER DATOS DEL TEMA

SC Modificar datos del tema

EC 1	Modificar Datos del Tema				<p>Muestra el Foro seleccionado con sus temas asociados, permitiendo modificar los valores de los mismos.</p> <p>Permite además:</p> <ul style="list-style-type: none"> • Guardar los datos. • Cancelar la operación en cualquier momento. 	<p>Selecciona un programa de estudio</p> <p>Selecciona un Foro para ver sus datos</p> <p>Selecciona un Tema del Foro para ver sus datos.</p> <p>Selecciona la opción de modificar los datos de un Tema.</p> <p>Modifica los datos que necesite y selecciona la opción de guardar los datos.</p>
	Modifica los datos que necesite y selecciona la opción de guardar los datos.	V	V	N/A	<p>Valida los datos.</p> <p>Actualiza los datos del Tema.</p> <p>Muestra un mensaje de información.</p> <p>Muestra los datos del Tema</p>	
EC 2	Existen datos incompletos	I	V		<p>Muestra encima del campo vacío un mensaje de información</p> <p>Muestra un indicador sobre los campos vacíos</p>	<p>Selecciona un programa de estudio</p> <p>Selecciona un Foro para ver sus datos</p> <p>Selecciona un Tema del Foro para ver sus datos.</p> <p>Selecciona la opción de modificar los datos de un Tema.</p> <p>Modifica los datos que necesite y selecciona la opción de guardar los datos.</p>
		V	I			
		I	I			
EC 3	Existen	I	V		Muestra encima del	Selecciona un

	<i>datos incorrectos</i>	<i>V</i>	<i>I</i>		<i>campo incorrecto un mensaje de información</i> <i>Muestra un indicador sobre los campos incorrectos</i>	<i>programa de estudio</i> <i>Selecciona un Foro para ver sus datos</i> <i>Selecciona un Tema del Foro para ver sus datos.</i> <i>Selecciona la opción de modificar los datos de un Tema.</i> <i>Modifica los datos que necesite y selecciona la opción de guardar los datos.</i>
<i>EC 4</i>	<i>El actor selecciona la opción de Cancelar</i>				<i>Elimina los datos creados</i> <i>Muestra un mensaje de información</i> <i>Muestra el escritorio de trabajo</i>	<i>Selecciona un programa de estudio</i> <i>Selecciona un Foro para ver sus datos</i> <i>Selecciona un Tema del Foro para ver sus datos.</i> <i>Selecciona la opción de modificar los datos de un Tema.</i> <i>Selecciona la opción de Cancelar</i>
<i>EC 5</i>	<i>El Tema ya se encuentra creado</i>				<i>Muestra un mensaje de error.</i>	<i>Selecciona un programa de estudio</i> <i>Selecciona un Foro para ver sus datos</i> <i>Selecciona un Tema del Foro para ver sus datos.</i> <i>Selecciona la opción de</i>

						<p><i>modificar los datos de un Tema. Modifica los datos que necesite y selecciona la opción de guardar los datos.</i></p>
--	--	--	--	--	--	--

TABLA 5 SC MODIFICAR DATOS DEL TEMA

SC Eliminar Tema

EC 1	<i>Eliminar Tema</i>				<p><i>Muestra un mensaje de confirmación y permite:</i></p> <ul style="list-style-type: none"> • <i>Aceptar y eliminar el Tema</i> • <i>Cancelar la operación</i> 	<p><i>Selecciona un programa de estudio Selecciona un Foro para ver sus datos Selecciona un Tema del Foro para ver sus datos. Selecciona la opción de eliminar Tema.</i></p>
	<i>Selecciona la opción de aceptar y eliminar el elemento mostrado.</i>				<p><i>Elimina el Tema Muestra un mensaje de información Regresa a la vista anterior actualizando los datos</i></p>	
EC 2	<i>El actor selecciona la opción de Cancelar</i>				<i>Regresa a la vista anterior</i>	<p><i>Selecciona un programa de estudio Selecciona un Foro para ver sus datos Selecciona un Tema del Foro para ver sus datos. Selecciona la opción de eliminar Tema. Selecciona la opción de Cancelar</i></p>

TABLA 6 SC ELIMINAR TEMA

Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Título	campo de texto	No	Cadena de caracteres. Admite una longitud de 100 caracteres y no permite números. Ejemplo: Ecuaciones trigonométricas.
2	Contenido	campo de texto	No	Cadena de caracteres que solo acepta letras. Admite una longitud de 100 caracteres
3	Activar / Desactivar	campo de selección	Si	Cadena de caracteres de tipo seleccionable

TABLA 7 DESCRIPCIÓN DE LAS VARIABLES

Caso de Prueba: Listar Tema del Foro.

Descripción general: El caso de uso se inicia cuando el Usuario de la Escuela desea consultar los datos de un Tema. El sistema brinda un listado de los temas existentes. El actor selecciona un Tema. El sistema muestra los datos de este Tema seleccionado. El actor consulta los datos y el caso de uso termina.

Condiciones de ejecución:

- Debe haberse generado el escritorio de trabajo del usuario autenticado.

SC Listar Tema del Foro

ID del escenario	Escenario				Respuesta del sistema	Flujo central
EC 1	Listar Temas de Foro				Muestra el listado de los Temas existentes con un resumen de sus	Selecciona un programa

					<p>datos:</p> <ul style="list-style-type: none"> • Nombre. • Cantidad de Respuestas. • Cantidad de Visitas. • Últimos mensajes. <p>Permite:</p> <ul style="list-style-type: none"> ✓ Salir de la vista actual. 	<p><i>de estudio</i> <i>Selecciona un Foro</i> <i>Selecciona la opción de listar Temas de Foro</i> <i>Selecciona el Tema que desea consultar</i></p>
	<i>El actor selecciona el tema que desea consultar.</i>				<p>Muestra los datos del Tema.</p> <p>Permite:</p> <ul style="list-style-type: none"> ✓ Salir de la vista actual 	
EC 2	<i>El actor selecciona la opción de Cancelar</i>				<p>Regresa a la vista anterior</p>	<p><i>Selecciona un programa de estudio</i> <i>Selecciona un Foro</i> <i>Selecciona la opción de listar Temas de Foro</i> <i>Selecciona el Tema que desea consultar</i> <i>Selecciona la opción de Cancelar</i></p>
EC 3	<i>No existen Temas de Foro creados</i>				<p>Muestra un mensaje de información</p>	<p><i>Selecciona un programa de estudio</i> <i>Selecciona un Foro</i> <i>Selecciona la opción de listar Temas de Foro</i></p>

TABLA 8 SC LISTAR TEMA DEL FORO

3.2 RESULTADO OBTENIDO

Al módulo desarrollado se le han ejecutado dos iteraciones de pruebas de caja negra, guiadas por los Casos de Prueba (CP) anteriormente descritos. Las pruebas realizadas arrojaron las siguientes No Conformidades (NC).

Tipo de No Conformidad	Iteración 1	Iteración 2
Error Técnico		
Correspondencia con otro artefacto	2	1
Ortografía	4	1
Redacción		
Validación	3	1
Opciones que no funcionan	2	1
Error de Interfaz	1	
Error Idioma	5	2
Funcionalidad	6	4
Excepciones		
Seguridad	2	
Diseño CP	2	1
Total	27	11

TABLA 9 RESULTADO DE LAS NC

Partiendo de las No Conformidades arrojadas, se corrigieron los problemas identificados. Se lograron validar todas las funcionalidades implementadas trayendo consigo el aumento de la calidad del módulo Comunicaciones de la plataforma ZERA.

3.3 CONCLUSIONES PARCIALES

En este capítulo se realizó un estudio de las pruebas existentes y se definió validar la solución mediante casos de prueba utilizando el método de caja negra. Estos casos de pruebas permitieron probar todas las funcionalidades del módulo Comunicaciones demostrando que la solución es totalmente funcional y que responde a los requisitos definidos por los analistas de sistema.

CONCLUSIONES GENERALES

Una vez culminado el desarrollo de la investigación es posible arribar a las siguientes conclusiones:

- ✓ Se seleccionaron las herramientas y tecnologías adecuadas para la implementación del módulo Comunicaciones de la plataforma ZERA.
- ✓ Se desarrolló el módulo Comunicaciones de la plataforma ZERA, permitiendo a los usuarios el envío/recepción de mensajes y el intercambio directo de información mediante el uso de los foros de discusión.
- ✓ Para certificar la veracidad de los algoritmos empleados en la implementación de la solución se desarrollaron Casos de Prueba que aseguran la validación de los mismos.

RECOMENDACIONES

Tomando como base la investigación realizada y la experiencia acumulada durante la implementación del módulo Comunicaciones de la plataforma ZERA, se proponen las siguientes recomendaciones:

- ✓ Dar continuidad al desarrollo del módulo, adicionándole las siguientes funcionalidades:
 - En el foro, agregar una funcionalidad que permita al profesor evaluar los comentarios realizados por los estudiantes.
 - En la mensajería, posibilitar adjuntar ficheros y establecer un límite del tamaño en los buzones.
- ✓ Mejorar la interfaz de los submódulos implementados, teniendo en cuenta la apariencia de herramientas similares.
- ✓ Seguir optimizando el sistema para mejorar el rendimiento de la aplicación.

REFERENCIAS BIBLIOGRÁFICAS

1. **tareas., Enciclopedia de.** INFORMÁTICA EDUCATIVA. [En línea] 12 de Enero de 2011. <http://www.encyclopediadetareas.net/2010/05/informatica-educativa.html>.
2. **Ortega, Jesús Fernández.** ¿Qué es la Teleformación? [En línea] 24 de Octubre de 2004. http://www.adrformacion.com/articulos/formacion/_que_es_la_teleformacion_y_que_ventajas_aporta_/articulo28.html.
3. **Vignatti, Magdalena.** Hipertexto e hipermedia: POTENCIALIDAD EXPRESIVA Y DIVERSIDAD. [En línea] 5 de Mayo de 2009. <http://vignattimagdalena.blogspot.com/2009/05/hipertexto-e-hipermedia-potencialidad.html>.
4. **Odalys Rabelo Vázquez, Ayli Milián Pérez y Dinorah Romo Abascal, Bibiana Guerra Duarte.** Hiperentornos de aprendizaje. [En línea] 13 de Enero de 2011. http://educarte.rimed.cu/module/profesor/visor_articulos.php?ver=infocomp&num=1&cat=0&Tip=sub&idMod=16.
5. **B, Mónica María Agudelo.** Plataforma Educativa. [En línea] 2008. <http://aprendeonline.udea.edu.co/banco/html/plataformaseducativas/>.
6. **Vásquez, Heydi.** LMS. [En línea] <http://www.slideshare.net/heydivas/tipos-de-lms>.
7. **University, The California.** Plataforma de Teleformación aprenDIST. [En línea] 2009.
8. **INSTITUTO SUPERIOR POLTÉCNICO “JOSE A. ECHEVERRÍA”, CUJAE.** Plataforma de Teleformación aprenDIST. [En línea] Septiembre de 2005.
9. **Santos, Elena.** PunBB. [En línea] 2007. <http://punbb.softonic.com/>.
10. **phpBB-Es.COM.** phpBB-Es.COM. [En línea] 2007. <http://www.phpbb-es.com/>.
11. **Alvarez, Miguel Angel.** Qué es HTML. [En línea] 12 de Enero de 2009. <http://www.desarrolloweb.com/articulos/que-es-html.html>.
12. **Pérez, Javier Eguíluz.** Introducción a CSS | LibrosWeb.es. [En línea] 19 de mayo de 2009. <http://www.librosweb.es/css/capitulo1.html>.
13. —. Introducción a JavaScript | LibrosWeb.es. [En línea] 4 de Mayo de 2009. <http://www.librosweb.es/javascript/capitulo1.html>.

14. **www.masadelante.com.** ¿Qué es XML? - Definición de XML. [En línea] Enero de 2011. <http://www.masadelante.com/faqs/xml>.
15. **definicion.de.** Definición de PHP. [En línea] 2009. <http://definicion.de/php/>.
16. **help.websiteos.com.** Definición de ASP. [En línea] 2010. http://help.websiteos.com/websiteos_sp/websiteos.html#definicion_de_as1.htm.
17. **www.ecured.cu.** Definición de Framework. [En línea] 2007. <http://www.ecured.cu/index.php/Symfony>.
18. **symfony-project.org.** Symfony. [En línea] 2007. <http://www.ecured.cu/index.php/Symfony>.
19. **Leopoldo, Carlos.** Zend Framework. [En línea] 27 de Noviembre de 2007. <http://techtastico.com/post/zend-framework-una-introduccion>.
20. **misrespuestas.com.** Qué es un servidor web . [En línea] 2005. <http://www.misrespuestas.com/que-es-un-servidor-web.html>.
21. **web.mit.edu.** Red Hat Enterprise Linux 4: Manual de referencia. «Servidor Apache HTTP». [En línea] 11 de Enero de 2011. <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-httpd.html>.
22. **SALVADOR, JUAN RODRIGUEZ.** IIS. [En línea] 2011. <http://www.webtaller.com/construccion/lenguajes/php/lessons/phpiss/phpiss.php>.
23. **MySQL-Hispano.org.** MySQL. [En línea] <http://www.webestilo.com/mysql/intro.phtml>.
24. **Martínez, Daniel Pecos.** Postgres. [En línea] 2011. http://danielpecos.com/docs/mysql_postgres/x15.html.
25. **Murcia, Universidad de.** Metodología de desarrollo de software. [En línea] Diciembre de 2007. <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.
26. **www.um.es.** Metodologías Pesadas. [En línea] 2007. <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.
27. **GALLEGO, JUAN PABLO GOMEZ.** RUP. [En línea] 16 de Nov de 2007. <http://www.scribd.com/doc/297224/RUP>.
28. **teki.** Metodologías ágiles. [En línea] 3 de Nov de 2009. <http://teki.cl/2009/11/metodologias-agiles-de-desarrollo-de-software/>.
29. **Seta, Leonardo De.** XP. [En línea] 5 de Feb de 2010. <http://www.dosideas.com/noticias/metodologias/822-una-introduccion-a-extreme-programming.html>.

30. **ness.sourceforge.net.** XP Fases. [En línea] 13 de Ene de 2011. <http://ness.sourceforge.net/proyecto/html/ch05s02.html>.
31. **Ltd., Visual Paradigm International.** Visual Paradigm. [En línea] 5 de Mar de 2007. [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(PE\)_%5Bcuenta_de_Windows_14730_p](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(PE)_%5Bcuenta_de_Windows_14730_p).
32. **Sánchez, Eduardo.** Rational Rose. [En línea] 22 de Mar de 2010. <http://www.amenito.com/rational-rose-enterprise-edition-suite-2003-perfecto-programa-para-analistas-que-usan-el-proceso-unificado-de-desarrollo-con-herramientas-uml>.
33. **Corporation, Oracle.** NetBeans. [En línea] 2010. http://netbeans.org/community/releases/69/index_es.html.
34. **tufuncion.com.** Los mejores IDEs para Php | TuFuncion. [En línea] 9 de Dic de 2007. <http://www.tufuncion.com/ide-php>.
35. **Potencier, Fabien y Zaninotto, François.** *Symfony 1.1, la guía definitiva*. s.l. : Apress , 2009. ISBN-13: 978-1590597866.

BIBLIOGRAFÍA

1. **IBM Corporation.** Ayuda Rational Unified Process, 2007. Tipos de prueba. [En línea] 2006. Ayuda.del.RUP.2007.Español/core.base_rup/guidances/concepts/types_of_test_8AB94831.html.
2. **Oré B., Ing. Alexander.** CalidadySoftware.com. FUNCTIONAL TESTING - PRUEBAS FUNCIONALES. [En línea] 2009. http://www.calidadyssoftware.com/testing/pruebas_funcionales.php
3. **Mendoza S, María A.** Informatizate. [En línea] 2004.
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
4. **Achour, Mehdi, Betz, Friedhelm y Dovgal, Antony.** Grupo de documentación de PHP. PHP. [En línea] 2009. <http://www.php.net/manual/es/index.php>.
5. **Eguíluz Pérez, Javier.** Introducción a CSS. libros Web. [En línea] 2009. www.librosweb.es Revista Electrónica Tropical. PostgreSQL: Una solución para el almacenamiento y el análisis de los datos espaciales. [En Línea] Marzo 6, 2008 http://www.iit.jalisco.gob.mx/RET/E6/RET6_postg.pdf
6. **Wells, Don.** Extreme Programming: A gentle introduction. [En Línea] Febrero 17, 2006. www.extremeprogramming.org