



*Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas*

*Desarrollo de funcionalidades que faciliten al docente su
preparación y el control del aprendizaje de los estudiantes en
la Plataforma Educativa ZERA.*

Autores

Irina Ivis Santiesteban Pérez

Miguel Medina Ramirez

Tutor

Ing. Jorge Luis Piña González

Cotutores

Ing. Mairelis Gari Maribona

Ing. Yoennis Garrido Vargas

La Habana, Cuba. 2010-2011

“Año 53 de la Revolución.”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo “Desarrollo de funcionalidades que faciliten al docente su preparación y el control del aprendizaje de los estudiantes en la Plataforma Educativa ZERA” y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autor: Irina Ivis Santiesteban Pérez

Autor: Miguel Medina Ramirez

Tutor: Ing. Jorge Luis Piña González

Cotutor: Ing. Mairelis Gari Maribona

Cotutor: Ing. Yoennis Garrido Vargas

AGRADECIMIENTOS

De Irina:

A mi papá Calero y mi mamá por darme la familia que tengo, apoyarme en todo momento, por guiarme con paciencia y dedicación, por ser mis ejemplos a seguir, gracias a ambos por confiar en mí, mis agradecimientos más especiales son para ustedes dos, sin ustedes, este momento no hubiese sido posible.

A toda mi familia, por ser parte importante en mi vida y sentirse siempre orgullosos de mí. A aputiti mi abuelito que lo quiero mucho, a mi tía Silvia, a mi tío Ovidio. A mi prima tata, a mi prima Thalía y su mamá Dinia. A mi papá, mi madrastra y mis hermanas, que nos llevamos muy bien y nos queremos mucho.

A Miguel, mi novio, mi amigo, mi sostén, primero como mi amigo incondicional y ahora como mi novio ejemplar, apoyándome en cada momento y dándome todo su amor, gracias nené por quererme como soy.

A mis tutores Piña y Yoennis por ayudarme y guiarme durante toda la investigación. A mi tutora Mairelis que me apoyó cuando entré al proyecto y durante el tiempo que he estado en él. A mi tribunal por su ayuda, consejos y seguimientos a lo largo de la investigación. Al líder del proyecto Abel, a Oreste y Leonardo por apoyar y contribuir con el desarrollo de esta investigación.

A mis amigos y amigas, nuevos y viejos, los de todas las etapas de mi vida, a todos les agradezco su apoyo, su cariño, su respeto, sepan que estarán siempre conmigo.

A mis profesores porque de ellos aprendí todo lo que se.

A la Revolución y en especial a Fidel por darme la oportunidad de estudiar en la UCI.

En fin doy las gracias a todas las personas que estuvieron a mi lado a lo largo de este sueño, hoy hecho realidad.

De Miguel:

A la Revolución Cubana por ofrecerme la gran oportunidad de formarme como profesional en el mejor Sistema Educativo del mundo.

A mis padres por todo lo que significan para mí y por su apoyo incondicional a lo largo de todos estos años.

A mi novia Irina, por quererme mucho, por su apoyo, paciencia y confianza en mí en los momentos decisivos, parte de este triunfo se lo debo.

A los tutores Piña, Yoennis y Mairelis por su gran ayuda durante la investigación, así como al equipo de desarrollo del proyecto Alfaomega (Orestes, Leonardo, Yoandy, Yerandy, Ernesto, Roberto, Yaismel, Arcel y Abel).

A todos los que de una forma u otra me han ayudado a llegar hasta aquí, por su confianza, su apoyo y su amistad. Ellos saben quienes son...

DEDICATORIAS

A mi familia en general por lo que significan para mí y por darme todo su apoyo en cada momento.

A mi mamá Ivis que es la persona más importante en mi vida y una de las que más quiero.

Muy especialmente se la dedico a mi papá Calero, por ser tan bueno conmigo, por estar siempre ahí cuando más los necesito, por darme tanto amor y cariño, por lograr hacer de mí una persona de bien y una profesional, por eso ni dándole mi vida entera lo recompensaré jamás.

Muy especialmente también a mi abuelita Odilia, que aunque no estuvo presente en esta etapa de mi vida, yo la quiero, la extraño y siempre la tengo presente.

A todos ustedes va dedicado este trabajo. Gracias.

Irina Ivis Santiesteban Pérez

A la memoria de mis abuelos por el cariño, el amor, la educación que me dieron y por ser siempre un ejemplo para mí, que aunque no me acompañan en esta importante etapa de mi vida, siempre los tengo presente...

A mis padres, a los que considero los mejores padres del mundo. Ellos han sido el impulso para llegar hasta aquí y así se sientan orgullosos de su hijo, quién siempre estará en deuda con ellos por todo lo que representan en su vida...

A toda mi familia por apoyarme en todo momento y por confiar siempre en mí.

Miguel Medina Ramirez

RESUMEN

En la actualidad aunque existen disímiles plataformas para la gestión del aprendizaje, muchas organizaciones o instituciones continúan apoyando el desarrollo de éstas, debido a que las existentes no siempre cumplen con las condiciones requeridas por los profesores para ejercer sus responsabilidades. La Plataforma Educativa ZERA, actualmente en desarrollo es un fiel ejemplo de lo anteriormente mencionado y por la necesidad que ésta presenta, de que los profesores cuenten con un espacio donde puedan obtener documentación para prepararse, seguir el aprendizaje de los estudiantes y aportar sus conocimientos, se realiza el presente trabajo. El cual está encaminado a desarrollar diferentes funcionalidades que estarán presentes en los hiperentornos que contiene la plataforma y que propiciarán una ayuda a los encargados de ejercer la profesión de educar, en cuanto a la preparación que día a día puedan ir obteniendo, el control de los estudiantes que tienen bajo su responsabilidad y la contribución que puedan ofrecer a los contenido de los hiperentornos. Para poder cumplir con los objetivos planteados en la investigación se realizó un análisis profundo de las tecnologías y herramientas más apropiadas para el desarrollo, seleccionando las más adecuadas, así como también se llevaron a cabo todos los pasos necesarios que propone la metodología empleada (RUP) para el desarrollo de las funcionalidades propuestas. Obteniendo como resultado final una solución que cumple con todo lo definido durante la investigación y que contiene la calidad requerida.

Palabras claves: docente, hiperentorno de aprendizaje, learning management systems (LMS) y plataforma educativa.

ÍNDICE

| | |
|---|----|
| INTRODUCCIÓN | 1 |
| CAPÍTULO I Fundamentación Teórica | 6 |
| 1.1 Introducción | 6 |
| 1.2 Estudio de sistemas orientados a apoyar el proceso de enseñanza - aprendizaje..... | 6 |
| 1.2.1 Software Educativo | 6 |
| 1.2.2 Plataformas de Aprendizaje..... | 8 |
| 1.2.2.1 Plataformas de aprendizaje internacionales..... | 8 |
| 1.2.2.2 Plataforma de aprendizaje nacional..... | 11 |
| 1.2.3 Gestión Académica | 12 |
| 1.2.4 Conclusiones parciales | 13 |
| 1.3 Metodologías y estándares para el desarrollo del software | 13 |
| 1.3.1 Rational Unified Process (en lo adelante RUP) | 14 |
| 1.3.2 Extreme Programing (en lo adelante XP)..... | 15 |
| 1.3.3 SCRUM | 16 |
| 1.3.4 Lenguaje de Modelado Unificado (en lo adelante UML)..... | 16 |
| 1.4 Herramientas y lenguajes de desarrollo | 17 |
| 1.4.1 Herramientas CASE de modelado de software | 17 |
| 1.4.2 Lenguajes de programación | 18 |
| 1.4.2.1 Lenguajes del lado del cliente..... | 19 |
| 1.4.2.2 Lenguajes del lado del servidor | 20 |
| 1.4.3 Framework para el desarrollo | 21 |
| 1.4.3.1 Capa de presentación | 21 |
| 1.4.3.2 Capa de lógica del negocio..... | 22 |
| 1.4.3.3 Capa de acceso a datos | 23 |
| 1.4.4 Sistema gestor de base de datos | 23 |
| 1.4.5 Entorno de desarrollo integrado (en lo adelante IDE)..... | 25 |
| 1.4.6 Servidor web | 26 |

| | | |
|---|---|----|
| 1.5 | Conclusiones | 27 |
| CAPÍTULO II Características de las Funcionalidades | | 28 |
| 2.1 | Introducción | 28 |
| 2.2 | Modelo de dominio | 28 |
| 2.2.1 | Conceptos del dominio | 28 |
| 2.2.2 | Diagrama del modelo de dominio | 29 |
| 2.3 | Descripción de las funcionalidades propuestas | 30 |
| 2.4 | Requisitos | 31 |
| 2.4.1 | Requisitos funcionales | 31 |
| 2.4.2 | Requisitos no funcionales | 34 |
| 2.5 | Patrones de caso de uso | 36 |
| 2.6 | Modelo de casos de uso del sistema | 36 |
| 2.6.1 | Descripción de los actores del sistema | 36 |
| 2.6.2 | Diagrama de actores del sistema | 37 |
| 2.6.3 | Diagrama de casos de uso del sistema | 37 |
| 2.6.4 | Descripción de los casos de uso del sistema | 38 |
| 2.7 | Conclusiones | 42 |
| CAPÍTULO III Análisis y Diseño de las Funcionalidades | | 43 |
| 3.1 | Introducción | 43 |
| 3.2 | Modelo de análisis | 43 |
| 3.1.1 | Diagramas de clases del análisis | 43 |
| 3.3 | Patrón arquitectónico Modelo Vista Controlador (en adelante MVC) en Symfony | 44 |
| 3.4 | Aplicación de los patrones de diseño en Symfony | 45 |
| 3.5 | Modelo de diseño | 47 |
| 3.5.1 | Diagrama de clases del diseño | 47 |
| 3.6 | Diagrama de despliegue | 50 |
| 3.7 | Diseño de la base de datos | 50 |
| 3.7.1 | Descripción de las tablas | 52 |
| 3.8 | Conclusiones | 53 |

| | |
|---|----|
| CAPÍTULO IV Implementación y Pruebas de las Funcionalidades | 54 |
| 4.1 Introducción | 54 |
| 4.2 Modelo de implementación | 54 |
| 4.2.1 Diagrama de componentes | 54 |
| 4.3 Pruebas de software | 56 |
| 4.3.1 Niveles de prueba..... | 56 |
| 4.3.2 Métodos de prueba | 57 |
| 4.3.3 Diseño de casos de prueba. | 58 |
| 4.3.4 Resultados obtenidos..... | 63 |
| 4.4 Conclusiones..... | 65 |
| CONCLUSIONES | 66 |
| RECOMENDACIONES..... | 67 |
| REFERENCIAS BIBLIOGRÁFICAS | 68 |
| BIBLIOGRAFÍA | 70 |

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (en lo adelante TIC) están siendo insertadas en todas las áreas de la sociedad, provocando importantes cambios en ellas. La educación es una de estas áreas, donde las posibilidades que estas tecnologías proporcionan, favorecen la introducción de aspectos innovadores en el proceso de enseñanza - aprendizaje, principalmente en la forma en que los profesores y aprendices se relacionan con el conocimiento y en la que las personas involucradas en el proceso educativo interactúan.

Una de las herramientas empleadas en dicho proceso y que constituye un complemento del uso de las TIC son los Sistemas de Gestión de Aprendizaje o en inglés Learning Management System (en lo adelante LMS) que no son más que plataformas educativas (herramientas computacionales, generalmente basadas en la web) que se basan en los principios del aprendizaje colaborativo. Los LMS se emplean para administrar, distribuir y controlar las actividades de formación presencial y no presencial o e-Learning, de una institución u organización. Sus principales funciones son: gestionar usuarios, recursos, materiales y actividades de formación, administrar el acceso, controlar y dar seguimiento al proceso de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión, videoconferencias, entre otros.

Actualmente existen diversos LMS que dan soporte al proceso de enseñanza - aprendizaje, brindándole una serie de facilidades. Cuba ha desarrollado algunos de estos LMS, debido a que cuenta con instituciones dedicadas a ello, entre las que se pueden mencionar está la Universidad de las Ciencias Informáticas (en lo adelante UCI), centro que se ha destacado en el desarrollo de herramientas educativas para fortalecer el proceso de enseñanza - aprendizaje. Ésta y más específicamente el Centro de Tecnologías para la Formación (en lo adelante FORTES), se encuentra desarrollando una Plataforma Educativa que recibe el nombre de ZERA, ésta tiene sus orígenes en la concepción pedagógica denominada "Hiperentornos de Aprendizaje"(sistema informático basado en tecnología hipermedia que contiene una mezcla o elementos representativos de diversas tipologías de software educativo), concepción propuesta y desarrollada por especialistas y profesores del Ministerio de Educación de Cuba (MINED). ZERA es una plataforma para bachillerato que permite la gestión del aprendizaje y es capaz de adaptarse a los procesos del negocio de cualquier institución educacional, donde además de realizar las

acciones de administración, contiene diversos hiperentornos como Redox, Sustancia y Campo, los cuales están compuestos varios módulos básicos: Contenidos, Prácticas, Biblioteca, Tareas, Docente, y para algunas materias, Apoyo.

La Plataforma Educativa que se está desarrollando tiene como objetivo servir de apoyo al proceso de enseñanza - aprendizaje, así como al profesor, él cual tendrá a su disposición un grupo de opciones para configurar el aprendizaje de sus estudiantes, aumentar su preparación profesional y realizar su contribución al contenido presente en los hiperentornos de la plataforma, todas estas acciones deben encontrarse en el módulo Docente, donde se agruparán las funcionalidades que le permitirán al mismo realizar sus responsabilidades como educador y evaluador.

Por lo anteriormente mencionado es necesario que el profesor tenga acceso a través de este módulo al programa de estudio de su materia y del período escolar que está trabajando de acuerdo al sistema educativo de su institución, así como a otros programas de estudio de su misma materia pero en diferentes períodos escolares, a las semejanzas que existe entre el contenido mostrado en la plataforma con el programa de estudio, a documentación con las propuestas didácticas para el manejo del contenido de la plataforma en clase, propuestas de evaluación para las actividades de la plataforma y del aula, a un manual de usuario docente para aprender a trabajar con los hiperentornos de aprendizaje y a información relevante para su preparación. La plataforma tiene entre sus principales características la posibilidad de adaptar, configurar y personalizar el sistema curricular para una escuela o institución específica atendiendo a la diversidad de estos en un mismo sistema educativo, es por ello que es necesario que el profesor tenga un espacio donde pueda en caso de que la institución lo permita, adaptar los períodos evaluativos según su forma de trabajar o criterio. También es preciso brindarle un espacio que funcione como un registro de asistencia y otro de calificación, que posibilite que estas acciones que el profesor realiza manualmente, se puedan automatizar, evitando la pérdida o el deterioro del lugar donde actualmente se encuentran. Dentro de las funciones llevadas a cabo por un docente está el seguimiento a sus estudiantes a través de la revisión de trabajo orientados por él, esta tarea es de vital importancia tanto para estudiantes como para profesores por lo que se hace necesario contar con un vía que le permita al profesor llevar a cabo esta revisión. Además, se debe contar con un lugar donde el profesor pueda registrar sus trabajos, sus materiales, sus informes, sus recomendaciones sobre algún contenido, esto no solo para su consulta sino también para la de otros profesores y estudiantes, además puede registrar su

avance programático en la asignatura, así como mostrar a sus estudiantes artículos que le hayan parecido interesantes, relacionados con la asignatura que este imparte.

A partir de la **situación problémica** planteada, se puede definir como **problema de investigación** la siguiente interrogante: ¿Cómo facilitarle al docente su preparación, el control del aprendizaje de sus estudiantes y su contribución a los contenidos presentes en los hiperentornos de aprendizaje de la Plataforma Educativa ZERA?

Este problema se enmarca en el **objeto de estudio**: desarrollo de plataformas educativas para la gestión del aprendizaje.

El **objetivo general** de la investigación es: desarrollar funcionalidades que le permitan al docente prepararse, supervisar el aprendizaje de sus estudiantes y realizar aportes al contenido presente en los hiperentornos de aprendizaje, siendo los **objetivos específicos** los siguientes:

- ✓ Realizar el estado del arte de la investigación.
- ✓ Realizar el análisis y diseño de las funcionalidades.
- ✓ Implementar las funcionalidades.
- ✓ Realizar las pruebas funcionales a las soluciones dadas.

El **campo de acción** de la investigación es: desarrollo de funcionalidades utilizadas por los docentes en plataformas educativas.

Se plantea como **idea a defender** que si se desarrollan funcionalidades para los docentes, en los hiperentornos de aprendizaje de la Plataforma Educativa ZERA, éste podrá prepararse, supervisar el aprendizaje de sus estudiantes y realizar aportes al contenido, que a su vez serán consultados por sus estudiantes o por otros docentes.

Para llevar a cabo la investigación, se emplearon los métodos teóricos y empíricos siguientes:

- ✓ **Métodos teóricos**: Se utilizó el método **Analítico-Sintético** para analizar la situación y la problemática existente, la documentación, las características de las metodologías, herramientas y tecnologías, para luego llegar a la conclusión de cuáles son las más adecuadas para el desarrollo. El método **Histórico-Lógico** para conocer el fenómeno hasta la actualidad, la evolución histórica de las

soluciones similares y las tendencias actuales, estudiar las acciones que puede realizar un docente a través de plataformas educativas, para prepararse, supervisar el aprendizaje de sus estudiantes y realizar aportes al contenido.

- ✓ **Métodos empíricos:** Se utilizó el método **Observación** para estudiar más de cerca el objeto de la investigación ya que mediante este se puede observar el comportamiento de los sistemas existentes y extraer información de ellos.

Para darle cumplimiento a los objetivos específicos se realizarán las siguientes **tareas de investigación:**

- ✓ Analizar la problemática existente.
- ✓ Evaluar las soluciones existentes.
- ✓ Realizar un estudio acerca de las metodologías de desarrollo de software y las herramientas de modelado.
- ✓ Realizar un estudio acerca de las tecnologías y herramientas de desarrollo.
- ✓ Estudiar la propuesta de arquitectura de la Plataforma Educativa ZERA.
- ✓ Identificar los casos de uso del sistema.
- ✓ Describir casos de uso identificados.
- ✓ Confeccionar el modelo de casos de uso.
- ✓ Identificar las clases del análisis y del diseño.
- ✓ Modelar las clases identificadas.
- ✓ Realizar el modelo de datos.
- ✓ Modelar el diagrama de componentes del sistema.
- ✓ Modelar el diagrama de despliegue del sistema.
- ✓ Implementar los casos de uso.
- ✓ Diseñar los casos de pruebas.
- ✓ Realizar las pruebas funcionales a las soluciones dadas.
- ✓ Documentar el trabajo realizado.

Como resultado del presente trabajo de diploma, se pretende obtener:

- ✓ Documentación relacionada con el desarrollo de las funcionalidades.
- ✓ Diagramas de casos de uso, de análisis, de diseño y de implementación.
- ✓ Descripciones de los casos de uso y los diseños de casos de prueba.

- ✓ Funcionalidades para que el docente pueda prepararse y con ello alcanzar un mejor desempeño, supervisar el aprendizaje de sus estudiantes y realizar aportes al contenido presente en los hiperentornos de aprendizaje, que pueden ser consultados por sus estudiantes, así como por otros docentes.

Estructura Capítular

Capítulo 1. Fundamentación Teórica.

En este capítulo se realizará un estudio de las plataformas de aprendizaje, colecciones educativas y otros sistemas relacionados con el tema y se estudiará acerca de las metodologías, herramientas y tecnologías, seleccionando las que se ajusten a lo requerido.

Capítulo 2. Características de las funcionalidades.

Este capítulo contendrá el modelo de dominio, los requerimientos funcionales y no funcionales que darán solución al problema planteado, la identificación de los casos de uso y los actores del sistema, brindando una descripción de estos, y por último la obtención del diagrama de casos de uso del sistema.

Capítulo 3. Análisis y Diseño de las funcionalidades.

En este capítulo se realizará el análisis y diseño, que se compone por los diagramas de clases del análisis y del diseño, el diagrama de despliegue, el modelo de base de datos, así como el estudio de los patrones de arquitectura y diseño, empleados.

Capítulo 4. Implementación y prueba de las funcionalidades.

En este capítulo se describirá como está implementado el sistema, a través de los diagramas de componentes, y por último se mostrarán los resultados de las pruebas aplicadas y un análisis de éstos.

CAPÍTULO I

Fundamentación Teórica

1.1 Introducción

El estudio de determinadas plataformas de aprendizaje y colecciones educativas es muy importante en la presente investigación, para conocer qué acciones se pueden realizar en ellas, que les permitan a los profesores ejercer su función como educador. Pero no solo este estudio conlleva a una buena investigación sino que también es preciso conocer las tecnologías, metodologías de desarrollo de software, así como las herramientas de desarrollo y modelado que se emplean para implementar dichas plataformas de aprendizaje.

1.2 Estudio de sistemas orientados a apoyar el proceso de enseñanza - aprendizaje

El estudio que se realiza a continuación está enfocado a investigar sobre que funcionalidades y acciones realizan estos sistemas, que posibilitan al profesor prepararse, controlar a los estudiantes y contribuir al contenido presente en ellos.

1.2.1 Software Educativo

Un software educativo puede ser considerado como el conjunto de recursos informáticos diseñados con la intención de ser utilizados en el contexto del proceso de enseñanza - aprendizaje. En Cuba se han desarrollado varios de ellos, entre los que se pueden mencionar: las colecciones Multisaber, Navegante y Futuro, las cuales serán estudiadas a continuación.

Colección Multisaber

Colección desarrollada para fomentar el proceso de enseñanza - aprendizaje en las escuelas cubanas, en especial, en el nivel de enseñanza primario. Es importante mencionar que existe una versión web desarrollada por la UCI. Esta colección está integrada por diversos software educativos, donde cada uno de ellos, se divide en varios módulos, entre los que se encuentra el módulo Maestro, que contiene:

- ✓ Documentos metodológicos, acerca del contenido que abordan y su tratamiento metodológico.
- ✓ Recomendaciones generales y específicas para su inserción en la práctica educativa.

- ✓ Materiales de consulta y artículos científicos que abordan la enseñanza de los contenidos tratados en los productos.

Colección Navegante

Colección que sirve de apoyo al proceso de enseñanza – aprendizaje, dirigida a la secundaria básica, y compuesta por hiperentornos de aprendizaje. Actualmente se encuentra en desarrollo la versión web de los 10 software educativos de la colección. Estos están divididos en módulos, entre los que se encuentra el módulo Maestro, que contiene:

- ✓ Una sección de recomendaciones metodológicas, donde se muestran los temas: tratamiento del contenido y uso del software.
- ✓ Artículos que pueden ser consultados en las secciones recomendaciones metodológicas y artículos de interés, así como pueden ser guardados en la ruta especificada por el profesor.
- ✓ Una opción que permite al profesor seleccionar las categorías de revisión que utilizará en el software para evaluar a los estudiantes, además puede adicionar, modificar o eliminar las categorías de revisión.
- ✓ Una sección de noticias que le permite al profesor adicionar noticias para su sección de clases, así como modificarlas o eliminarlas.
- ✓ Una opción consultar ejercicios resueltos, la cual le permite al profesor consultar las respuestas de los ejercicios en cada uno de los temas disponibles.

Colección Futuro

Colección basada en tecnología hipertexto y enfocada a fomentar el proceso de enseñanza - aprendizaje de los estudiantes de pre-universitario. Está formada por varios software educativos, los cuales abarcan contenidos relacionados con las diferentes asignaturas impartidas en este nivel, cada software se divide en módulos, entre los que se encuentra el módulo Profesor (Ver figura 1), el cual solo aparece activado para éste y presenta varias funcionalidades, las cuales son:

- ✓ Programa de la asignatura: Brinda acceso a los programas de la asignatura en los diferentes grados escolares.
- ✓ Orientaciones metodológicas: Brinda respuestas a un conjunto de interrogantes de carácter metodológico con respecto al uso del software que son preliminares y de obligatorio conocimiento.

- ✓ Artículos: Esta sección presenta artículos de actualización y valor metodológico que están asociados con el contenido del programa.
- ✓ Configuración: Permite configurar algunos comportamientos del programa de estudios como son: cerrar la matrícula, dar la posibilidad de acceso o no a los módulos Temas y Biblioteca para que sean consultados desde los cuestionarios.
- ✓ Visor de ejercicios: Propicia que el profesor tenga un espacio de visualización de ejercicios pudiendo acceder a sus respuestas.
- ✓ Editor de noticias: Posibilita que el docente emita información a través de la red a los estudiantes en dependencia del grupo y el grado.



Figura 1. Módulo Profesor de la Colección Futuro.

1.2.2 Plataformas de Aprendizaje

En la actualidad existen numerosas plataformas de aprendizaje, entre las que se pueden citar en el ámbito internacional están Claroline, Moodle y en el ámbito nacional Sadhea, las cuales serán estudiadas a continuación.

1.2.2.1 Plataformas de aprendizaje internacionales

Claroline

Es una plataforma de aprendizaje y trabajo virtual (e-Learning y e-Working) de código abierto y software libre (open source), que permite a los formadores construir eficaces cursos online y gestionar las actividades de aprendizaje y colaboración en la web. (1) Ésta cuenta con diversas funciones realizadas por un docente como:

- ✓ Publicar documentos en cualquier formato (Word, PDF, HTML, video, SXW): Esta sección proporciona una herramienta para organizar archivos/documentos que se deseen poner a disposición de los alumnos. De algún modo es una especie de FTP simple, donde sólo los profesores pueden subir archivos. En ella se puede borrar, mover los archivos de un directorio a otro, renombrarlos, añadir una descripción a cada uno o hacer invisibles algunos de ellos para sus alumnos. También se puede enviar varios archivos de una sola vez, pero primero se debe comprimir en el ordenador los que se deseen subir. Ver la siguiente figura para un mejor entendimiento.

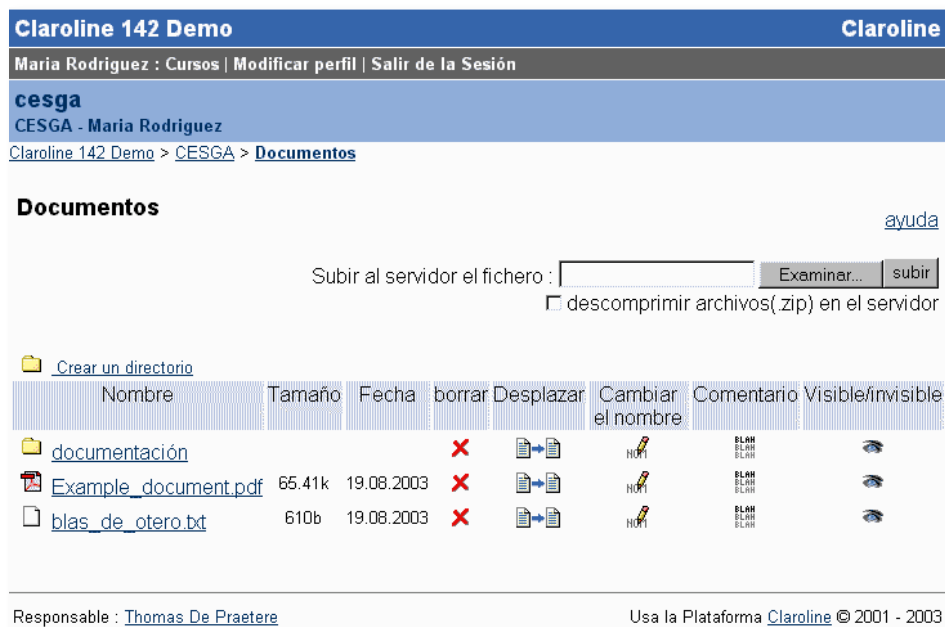


Figura 2. Publicar documentos.

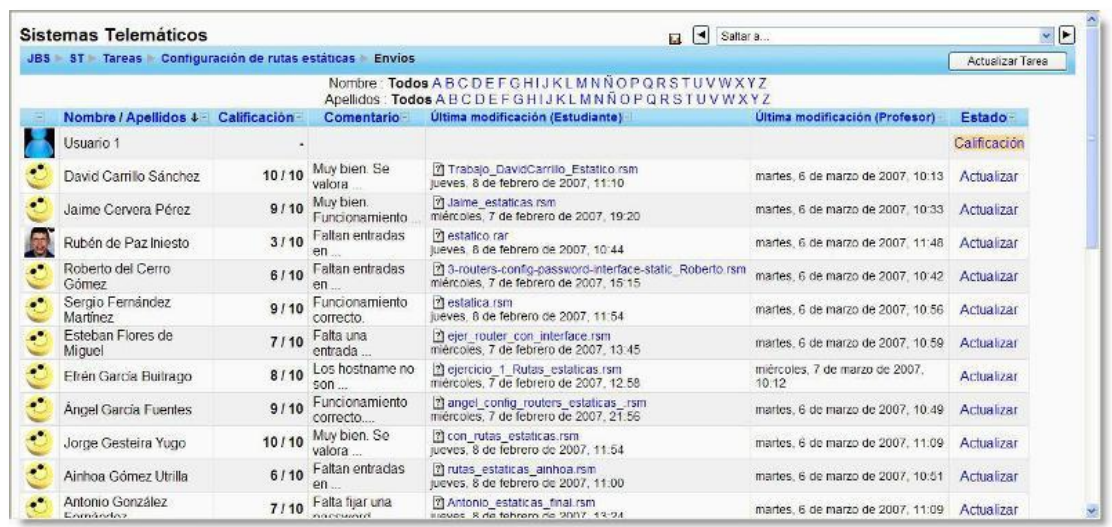
- ✓ Anuncios: Esta sección permite tener un tablón de anuncios importantes (recordatorios, avisos, etc). En él, sólo los profesores podrán publicar. Además pueden modificar y eliminar un anuncio publicado.
- ✓ Enlaces: Esta sección permite tener una colección de enlaces de interés para los alumnos.

- ✓ Añadir enlaces externos y páginas HTML a la página principal del curso: En esta sección se puede hacer un enlace a una página HTML que suba el profesor desde la página de inicio del curso, mediante la opción “Añadir un enlace a la web en la página principal”.

Moodle

Es un Sistema de Gestión de Cursos de Código Abierto (Open Source Course Management System, CMS), conocido también como Sistema de Gestión del Aprendizaje o como Entorno de Aprendizaje Virtual (Virtual Learning Environment, VLE).(2) Éste contiene diversas funcionalidades que puede realizar un docente, como:

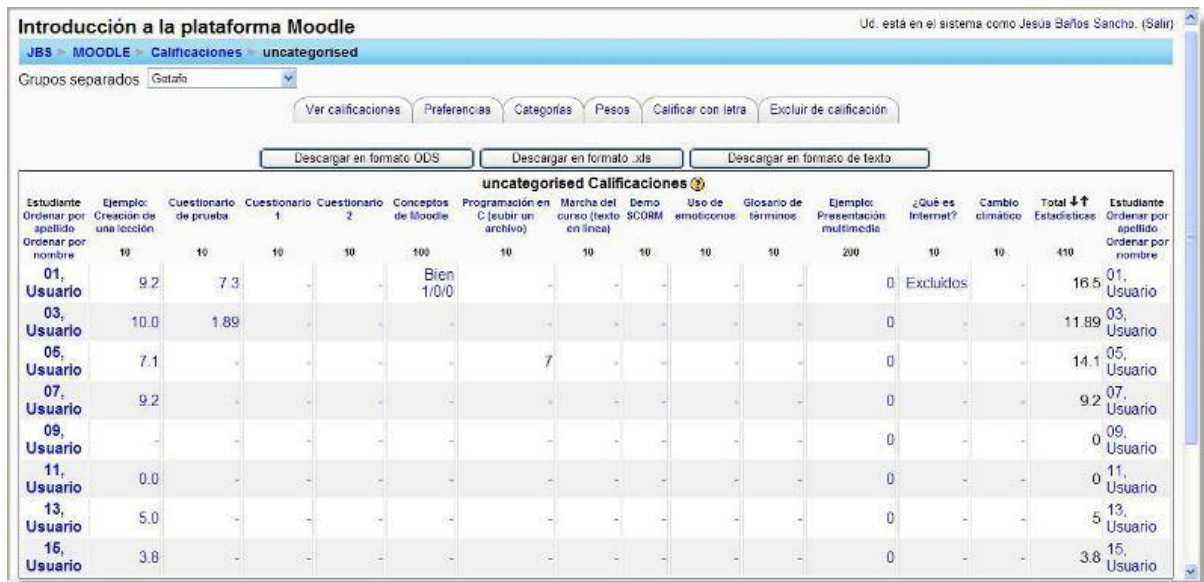
- ✓ Gestionar archivos: Cada curso virtual dispone de un espacio privado, sólo accesible desde ese curso. El profesorado puede estructurar ese espacio creando subdirectorios y subiendo los archivos que necesite para que sean accesibles a través de la web. El gestor presenta una lista de archivos y directorios, y además se puede subir archivo, crear una carpeta, entre otras funcionalidades.
- ✓ Revisar tareas: Esta opción permite revisar los trabajos de los alumnos. Primeramente se muestra un listado de estudiantes, el nombre del archivo que subió cada uno, en qué fecha lo hizo y el enlace Calificación, que es donde se evalúa la tarea y se introduce la valoración del trabajo y algunos comentarios. Una vez evaluada, el enlace Calificación pasa a llamarse Actualizar. La figura 3 muestra, en formato tabla, lo anteriormente mencionado, y además la fecha de corrección por parte del profesor/a, la calificación otorgada y los comentarios del profesorado respecto a la tarea corregida.



| Nombre / Apellidos | Calificación | Comentario | Última modificación (Estudiante) | Última modificación (Profesor) | Estado |
|---------------------------|--------------|----------------------------|--|--------------------------------------|--------------|
| Usuario 1 | - | - | - | - | Calificación |
| David Carrillo Sánchez | 10 / 10 | Muy bien. Se valora ... | Trabajo_DavidCarrillo_Estatico.rsm jueves, 8 de febrero de 2007, 11:10 | martes, 6 de marzo de 2007, 10:13 | Actualizar |
| Jaime Corvera Pérez | 9 / 10 | Muy bien. Funcionamiento | Jaime_estaticas.rsm miércoles, 7 de febrero de 2007, 19:20 | martes, 6 de marzo de 2007, 10:33 | Actualizar |
| Rubén de Paz Iniesto | 3 / 10 | Faltan entradas en ... | estatico.rar jueves, 8 de febrero de 2007, 10:44 | martes, 6 de marzo de 2007, 11:46 | Actualizar |
| Roberto del Cerro Gómez | 6 / 10 | Faltan entradas en ... | 5-routers-config-password-interface-static_Roberto.rsm miércoles, 7 de febrero de 2007, 15:15 | martes, 6 de marzo de 2007, 10:42 | Actualizar |
| Sergio Fernández Martínez | 9 / 10 | Funcionamiento correcto. | estatica.rsm jueves, 8 de febrero de 2007, 11:54 | martes, 6 de marzo de 2007, 10:56 | Actualizar |
| Esteban Flores de Miguel | 7 / 10 | Falta una entrada ... | ejer_router_con_interface.rsm miércoles, 7 de febrero de 2007, 13:45 | martes, 6 de marzo de 2007, 10:59 | Actualizar |
| Eldrén García Buitrago | 8 / 10 | Los hostname no son ... | ejercicio_1_Rutas_estaticas.rsm miércoles, 7 de febrero de 2007, 12:58 | miércoles, 7 de marzo de 2007, 10:12 | Actualizar |
| Ángel García Fuentes | 9 / 10 | Funcionamiento correcto... | angel_config_routers_estaticas_rsm miércoles, 7 de febrero de 2007, 21:56 | martes, 6 de marzo de 2007, 10:49 | Actualizar |
| Jorge Gesteira Yugo | 10 / 10 | Muy bien. Se valora ... | con_rutas_estaticas.rsm jueves, 8 de febrero de 2007, 11:54 | martes, 6 de marzo de 2007, 11:09 | Actualizar |
| Ainhoa Gómez Utrilla | 6 / 10 | Faltan entradas en ... | rutas_estaticas_ainhoa.rsm jueves, 8 de febrero de 2007, 11:00 | martes, 6 de marzo de 2007, 10:01 | Actualizar |
| Antonio González | 7 / 10 | Falta fijar una ... | Antonio_estaticas_final.rsm jueves, 8 de febrero de 2007, 13:24 | martes, 6 de marzo de 2007, 11:09 | Actualizar |

Figura 3. Revisión de tareas enviadas.

- ✓ Ver libro de calificaciones: Recoge las notas obtenidas por todos los estudiantes en cada una de las actividades del curso. Es una tabla donde cada fila corresponde a un estudiante y cada columna a una actividad, salvo la última, que representa la calificación total. Permite además descargar las calificaciones en diferentes formatos, como Microsoft Excel para realizar estudios y representaciones gráficas más completas. Para un mejor entendimiento ver la siguiente figura.



The screenshot shows the Moodle interface for 'uncategorised Calificaciones'. It includes a navigation menu, a dropdown for 'Grupos separados' (Gataña), and buttons for 'Ver calificaciones', 'Preferencias', 'Categorías', 'Pesos', 'Calificar con letra', and 'Excluir de calificación'. Below these are download buttons for 'ODS', 'xls', and 'text'. The main table is titled 'uncategorised Calificaciones' and contains the following data:

| Estudiante | Ejemplo: Creación de una lección | Cuestionario de prueba | Cuestionario 1 | Cuestionario 2 | Conceptos de Moodle | Programación en C (subir un archivo) | Marcha del curso (texto en línea) | Demo SCORM | Uso de emoticonos | Glosario de términos | Ejemplo: Presentación multimedia | ¿Qué es Internet? | Cambio climático | Total Estadísticas | Estudiante |
|-------------|----------------------------------|------------------------|----------------|----------------|---------------------|--------------------------------------|-----------------------------------|------------|-------------------|----------------------|----------------------------------|-------------------|------------------|--------------------|-------------|
| 01, Usuario | 9.2 | 7.3 | - | - | 100 | Bien 1/0/0 | - | - | - | - | 200 | 0 | Excluidos | 16.5 | 01, Usuario |
| 03, Usuario | 10.0 | 1.89 | - | - | - | - | - | - | - | - | - | 0 | - | 11.89 | 03, Usuario |
| 05, Usuario | 7.1 | - | - | - | - | - | 7 | - | - | - | - | 0 | - | 14.1 | 05, Usuario |
| 07, Usuario | 9.2 | - | - | - | - | - | - | - | - | - | - | 0 | - | 9.2 | 07, Usuario |
| 09, Usuario | - | - | - | - | - | - | - | - | - | - | - | 0 | - | 0 | 09, Usuario |
| 11, Usuario | 0.0 | - | - | - | - | - | - | - | - | - | - | 0 | - | 0 | 11, Usuario |
| 13, Usuario | 5.0 | - | - | - | - | - | - | - | - | - | - | 0 | - | 5 | 13, Usuario |
| 15, Usuario | 3.8 | - | - | - | - | - | - | - | - | - | - | 0 | - | 3.8 | 15, Usuario |

Figura 4. Libro de calificaciones.

- ✓ Agregar recursos: Estos recursos los trae predeterminado el sistema y el docente lo que hace es seleccionar y configurar el recurso con el que quiera trabajar. Tener en cuenta que para añadir un recurso debe estar activado el modo de edición.

1.2.2.2 Plataforma de aprendizaje nacional

Sadhea

Es un Sistema de Autor para el desarrollo de Hiperentornos de Aprendizajes en la web, una solución 100% cubana para la producción de software educativo (SE) en forma de hiperentornos de enseñanza - aprendizaje (HEA). Contiene un módulo Profesor que le brinda al maestro, además de las facilidades de administración propias de sistemas, otros servicios que forman parte de la bibliografía puesta a disposición del maestro para su enriquecimiento intelectual y su crecimiento profesional. A continuación se realiza una breve explicación de cada uno de estos servicios:

- ✓ El programa de la asignatura: En él se explica el contenido a tratar dentro del currículo de los grados o el grado para el cual está desarrollado el HEA.
- ✓ Información complementaria: Acceso a tópicos relacionados con el empleo de las TIC en el proceso de enseñanza aprendizaje en general y en particular sobre el empleo en la asignatura específica que se trata en el software.
- ✓ Orientaciones metodológicas: Además de traer las indicaciones de cómo dar salida y tratamiento al contenido de la asignatura, sugiere algunas formas de cómo tratar el contenido con el empleo del software como medio para facilitar el aprendizaje.
- ✓ Artículos de interés: Materiales didácticos, científicos, metodológicos y de otra índole que a consideración del desarrollador pueden ser utilizados para un mejor aprovechamiento de las potencialidades de esta herramienta por parte de los usuarios.(3)

1.2.3 Gestión Académica

Akadosmos

Sistema automatizado para la gestión académica desarrollado por profesores provenientes de la CUJAE, la dirección de Informatización de la UCI y sus estudiantes. Éste cuenta con varias opciones que lo hacen muy útil, una de ellas es el Registro (Ver figura 6), donde se recogen las evaluaciones y las asistencias de grupos de estudiantes. Entre las funcionalidades que se realizan en él se encuentran:

- ✓ Ver registro asistencias: En esta sección se muestra el nombre de la asignatura, el grupo que la está recibiendo y la cantidad de horas clases que se han dado, así como el listado de estudiantes, las fechas en las que se pasó la asistencia, con la posibilidad de modificarlas o eliminarlas y el por ciento de presente, justificado e injustificado de cada uno.
- ✓ Pasar asistencia: En esta opción aparece un campo fecha que por defecto tiene la fecha actual y brinda la posibilidad de cambiarla, además contiene el listado de estudiantes, la foto, el tipo de asistencia (de tipo seleccionable) donde por defecto aparecen todos con el estado presente y permite introducir un argumento por cada uno.
- ✓ Ver registro evaluaciones: En esta sección se muestra el nombre de la asignatura, el grupo que la está recibiendo y el listado de estudiantes con las diferentes evaluaciones realizadas.
- ✓ Registrar evaluación: En esta opción se selecciona el tipo de evaluación (que puede ser frecuente, final o parcial) y la evaluación (que está en dependencia del tipo de evaluación que se realice),

posteriormente aparece un listado de estudiantes y un campo de selección que permite darle la evaluación que va a recibir cada uno.

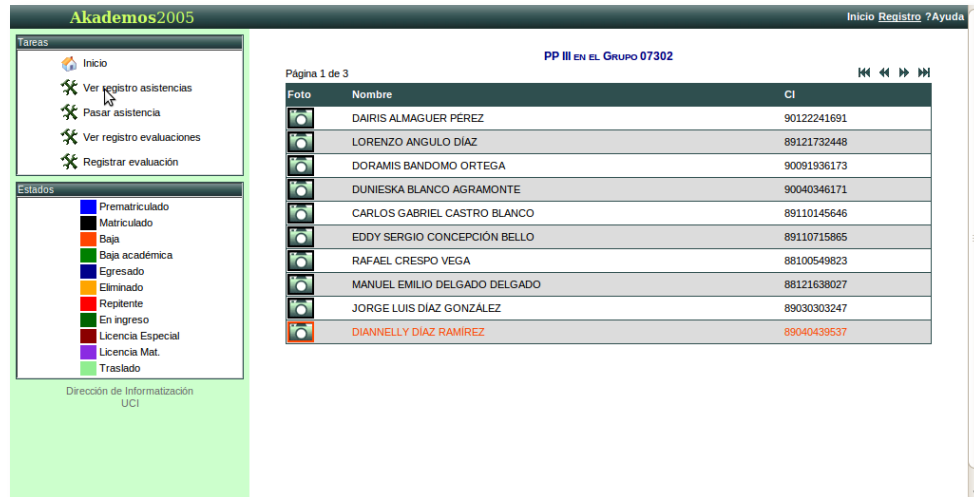


Figura 5. Sección Registro de Akademos para profesores.

1.2.4 Conclusiones parciales

Después de haber realizado un profundo estudio se pudo concluir que estos sistemas permitieron obtener una visión de las principales funcionalidades que se desarrollaran durante la investigación, sirviendo como referencia a la implementación posterior. Las colecciones Multisaber, Navegante y Futuro servirán de apoyo para el desarrollo de las acciones vinculadas con la preparación del profesor, debido a que su estudio estuvo enmarcado fundamentalmente en los materiales y la documentación que éstos deben consultar. Por otro lado las plataformas estudiadas presentaron las siguientes funciones que también servirán de apoyo, como: publicar documentos, anuncios, enlaces, revisar tareas, visualizar resultados de las tareas, adicionar recursos, visualizar el programa de la asignatura, información complementaria, orientaciones metodológicas y artículos de interés. Además se hizo necesario estudiar Akademos debido a que las plataformas no presentaban acciones enfocadas a los procesos de pasar la asistencia y registrar las calificaciones de los estudiantes.

1.3 Metodologías y estándares para el desarrollo del software

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. (4) Ver la siguiente figura.



Figura 6. Proceso de desarrollo de software.

Este proceso es complejo, pues comúnmente los productos que solicitan desarrollar las empresas son complicados y abstractos, lo cual hace difícil el trabajo que realizan los analistas y desarrolladores para definir las funcionalidades del sistema e implementarlas en el tiempo acordado. Debido a esta razón se hace necesario contar con una guía, un estándar, que indique qué pasos seguir para lograr un producto con calidad y que funcione eficientemente. A esta guía se le conoce como metodología de desarrollo de software.

En la actualidad existen una gran variedad de metodologías orientadas al desarrollo de software, entre las más usadas a nivel mundial se pueden citar Rational Unified Process (RUP), Extreme Programming (XP) y SCRUM, las cuales serán estudiadas a continuación.

1.3.1 Rational Unified Process (en lo adelante RUP)

Rational Unified Process (RUP) es un proceso de desarrollo de software, considerado en la actualidad como una de las metodologías más utilizadas junto con el Lenguaje Unificado de Modelado (en lo adelante UML) para el desarrollo de proyectos de software. RUP captura las mejores prácticas del conocimiento de líderes en ingeniería de software y proporciona a los equipos de desarrollo guías, estándares y recomendaciones para la construcción de un software de alta calidad.

Entre las principales características de RUP se encuentran:

- ✓ Dirigido por Casos de Uso: Los casos de usos describen las funcionalidades que presenta un sistema determinado, los mismos se obtienen a partir del modelado del negocio y la captura de requisitos. Estos guían el diseño, la implementación y las pruebas de todo proceso de desarrollo del software. (4)
- ✓ Centrado en la Arquitectura: La arquitectura muestra la visión común del sistema completo donde tanto los desarrolladores como los usuarios están de acuerdo. La arquitectura nos da una clara perspectiva del sistema, necesaria para controlar el desarrollo, describe los elementos del modelo que son más importantes, los cimientos del sistema, que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. (4)
- ✓ Iterativo e Incremental: RUP se divide en 4 fases y cada una de ellas se desarrolla en iteraciones, cada iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla unos más que

otros, esto hace posible que se pueda dividir en pequeños proyectos perfeccionando su comprensión y desarrollo.

Las 4 fases presentes en RUP son: inicio, elaboración, construcción y transición. Además, se divide en 9 flujos de trabajo, donde los 6 primeros son flujos de trabajo básicos y los otros 3 son de soporte, en la figura 8 se muestran estos flujos, las fases y como se comportan éstos en cada una de las fases.

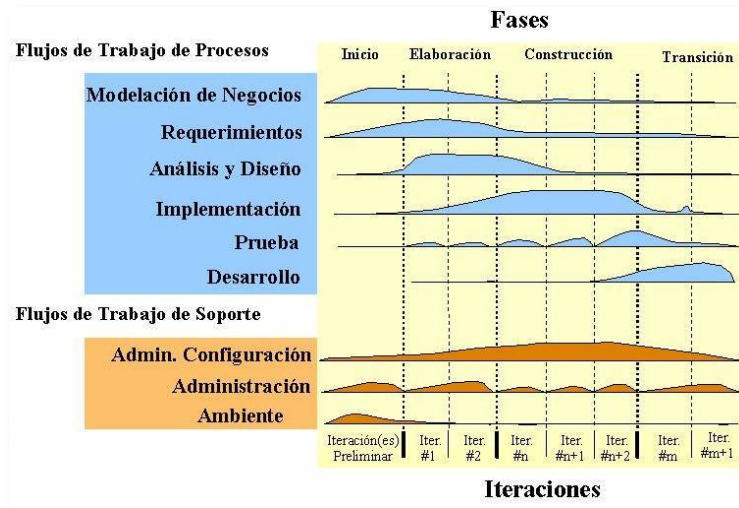


Figura 7. RUP en dos dimensiones.

1.3.2 Extreme Programing (en lo adelante XP)

La Extreme Programing, en español programación extrema, es la metodología más destacada en los procesos ágiles de desarrollo de software. XP se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo con lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.(5)

XP se centra en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promueve el trabajo en equipo y se basa en la retroalimentación entre el cliente y el equipo de desarrollo. Básicamente consiste en trabajar estrechamente con el cliente, haciendo pequeñas iteraciones (mini-entregas) cada dos semanas, donde no existe más documentación que el código en sí; cada versión

contiene las modificaciones necesarias según el cliente vaya retroalimentando el sistema, por eso es necesaria la disponibilidad del cliente durante todo el desarrollo.(6)

1.3.3 SCRUM

SCRUM es una metodología ágil de desarrollo de software, que plantea que los ciclos de desarrollo de los proyectos sean lo más rápidos posible. Esta metodología se ajusta a los cambios que puedan presentarse y mantiene una estrecha relación con el cliente. En ella se realiza un desarrollo incremental del producto con entregas frecuentes al cliente, garantizando así, que éste se encuentre al tanto del progreso del sistema.

SCRUM es muy fácil de aprender, ya que requiere de muy poco esfuerzo para utilizarlo, pero no se tiene una perspectiva global del proyecto más allá de una lista de tareas. Además, no genera toda la documentación que se obtiene con otras metodologías por lo que no es muy conveniente utilizarlo solo.

Selección de la metodología a utilizar

Con el estudio realizado sobre las metodologías de desarrollo y atendiendo a las características y a la complejidad de las funcionalidades, se llegó a la conclusión de que se debe usar es RUP, debido a que tanto XP como SCRUM están destinadas a proyectos de corto plazo y se basan en iteraciones pequeñas, en cambio RUP está pensado para proyectos y equipos grandes, en cuanto a tamaño y duración, además de que no requiere la presencia permanente del cliente, característica presente en las funcionalidades a desarrollar. También RUP genera una gran cantidad de documentación detallada durante todo el proceso de desarrollo, la cual es necesaria tanto para todo el personal del proyecto como para el cliente. Conjuntamente RUP es una metodología robusta y bien definida que posibilita la programación orientada a objetos y permite llevar a cabo un proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades.

1.3.4 Lenguaje de Modelado Unificado (en lo adelante UML 2.0)

El Lenguaje de Modelado Unificado, por sus siglas en inglés UML (Unified Modeling Language) se utiliza para especificar, visualizar, construir y documentar artefactos de un software. Este lenguaje permite la programación orientada a objetos, lo que garantiza la reutilización de código, la herencia, el encapsulamiento, entre otras ventajas.

UML está compuesto por diferentes elementos gráficos que se combinan para formar diagramas. Estos representan diversas vistas de un sistema, a las cuales se les conoce como modelo. Un modelo UML describe lo que debe hacer un sistema, pero no dice cómo implementarlo. UML proporciona a los desarrolladores un vocabulario que incluye tres categorías: elementos, relaciones y diagramas. (4) Éstas se pueden observar claramente en la siguiente figura:

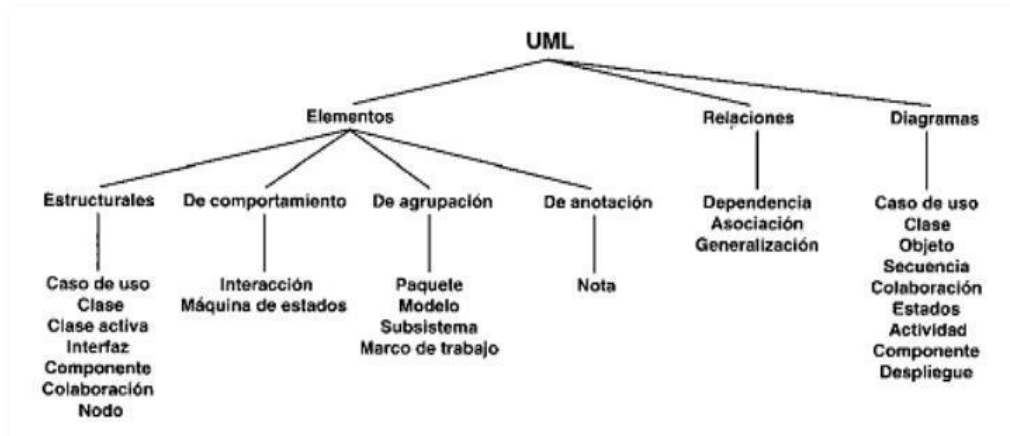


Figura 8. El vocabulario de UML

1.4 Herramientas y lenguajes de desarrollo

1.4.1 Herramientas CASE de modelado de software

Las herramientas CASE de modelado ofrecen muchos beneficios para todos los involucrados en un proyecto, además permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraerse del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar.

Existen varias herramientas de modelado para Linux, entre las que se pueden citar Visual Paradigm y Argo UML, las cuales serán estudiadas a continuación.

Visual Paradigm 6.4

Visual Paradigm es una herramienta para el modelado de software, que ayuda a que el modelado sea más fácil y rápido de realizar, ya que permite diseñar todos los diagramas necesarios y contiene una buena cantidad de productos o módulos para facilitar el trabajo durante la confección de un software, así como garantizar la calidad del producto final.

Desde Visual Paradigm es posible generar código para plataformas como .NET, Java y PHP, así como obtener diagramas a partir del código, esto reduce la posibilidad de cometer errores y le ahorra tiempo a los desarrolladores. Se integra con numerosos ambientes de desarrollo integrados, lo que permite pasar del código al modelado y viceversa. Soporta UML 2.1 y permite documentar todo el trabajo y las especificaciones de los casos de uso sin necesidad de utilizar herramientas externas como editores de texto, plantillas, entre otros. Brinda la posibilidad de obtener una base de datos relacional y el código necesario para acceder a esta a partir de un diagrama entidad relación, además de conectarse fácilmente a varios servidores de base de datos.

ArgoUML 0.32.2

ArgoUML es una herramienta desarrollada en Java que permite crear modelos UML compatibles con los estándares de la versión 1.4 de este lenguaje. Incluye una interfaz muy intuitiva, estable y de sencillo manejo. Los tipos de diagramas que se pueden crear son: diagramas de clases, de estados, de actividad, de casos de uso, de colaboración, de despliegue y de secuencia. (7)

Otras características presentes en ArgoUML son:

- ✓ Compatibilidad con el estándar XML (Metadata Interchange): es posible utilizar los modelos creados con UML en otros programas.
- ✓ Generación de código para los lenguajes Java, C++, C, PHP4 y PHP5.
- ✓ Framework modular para ingeniería inversa.
- ✓ Críticas de diseño o sugerencias realizadas por ArgoUML para mejorarlo. (7)

Selección de la herramienta a utilizar

Dadas las características que se exponen anteriormente de ambas herramientas de modelado y atendiendo a las necesidades del proyecto se seleccionó Visual Paradigm, debido a que soporta la versión de UML 2.0 y la generación de código es más óptima que la de ArgoUML.

1.4.2 Lenguajes de programación

Teniendo en cuenta las características de la Plataforma Educativa Zera (que es un sistema Web), se ha decidido adoptar una arquitectura cliente - servidor.

Una arquitectura cliente - servidor se caracteriza por: el cliente y el servidor pueden actuar como una sola entidad y también como entidades separadas, realizando actividades o tareas independientes. El cliente

no necesita conocer la lógica del servidor, sólo su interfaz externa. Este no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo. Los cambios en el servidor implican pocos o ningún cambio en el cliente. Las funciones de cliente y servidor pueden estar en plataformas separadas o en la misma plataforma. (8)

Actualmente existen un gran número de lenguajes de programación que permiten desarrollar una aplicación con las características planteadas anteriormente, es decir, lenguajes basados en la tecnología web; los cuales pueden dividirse en dos grandes grupos: lenguajes del lado del cliente y lenguajes del lado del servidor.

1.4.2.1 Lenguajes del lado del cliente

Entre los lenguajes del lado del cliente que se estudiarán a continuación se encuentran XHTML, CSS, JavaScript y Ajax.

eXtensible HyperText Markup Language (en lo adelante XHTML 1.0)

XHTML, acrónimo inglés de eXtensible HyperText Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para las hojas de estilo y JavaScript su aspecto y diseño en distintos medios (ordenadores, PDAs, teléfonos móviles, impresoras).(9)

Cascading Style Sheets (en lo adelante CSS 2)

CSS (siglas de Cascading Style Sheets - Hojas de Estilo en Cascada) permite la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento, con las hojas de estilo, se incluyen elementos tales como los colores, fondos, márgenes, bordes, tipos de letras, se modifica la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y el formato de sus documentos. El lenguaje CSS se basa en una serie de reglas que rigen el estilo de los elementos en los documentos estructurados, y que

forman la sintaxis de las hojas de estilo. Cada regla consiste en un selector y una declaración, esta última va entre corchetes y consiste en una propiedad o atributo, y en un valor separado por dos puntos.(10)

JavaScript

Se trata de un lenguaje de programación del lado del cliente, pues es el navegador el que soporta la carga de procesamiento. Se emplea principalmente para crear páginas web dinámicas y gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único con que cuenta este lenguaje es el propio navegador.

Técnicamente, 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.(11)

Asynchronous JavaScript And XML (Ajax)

Ajax (Asynchronous JavaScript And XML - JavaScript Asíncrono y XML). Es la unión de varias tecnologías de desarrollo web (como HTML o XHTML, CSS, Javascript, DOM (Document Object Model), XML, XSLT, y el objeto XMLHttpRequest) que buscan crear aplicaciones interactivas. Cuando se combinan estas tecnologías en el modelo Ajax, las aplicaciones funcionan mucho más rápido, ya que las interfaces de usuario se pueden actualizar por partes sin tener que actualizar toda la página completa. La importancia de esta técnica está en que al utilizarla, ésta se ejecuta en el lado del cliente, pero por debajo mantiene una comunicación asíncrona con el servidor. Con esta asincronía utilizando XML, cuando se desee hacer algún cambio en el desarrollo web, no será necesario que la página se recargue nuevamente.

1.4.2.2 Lenguajes del lado del servidor

Hypertext Pre-Processor (en lo adelante PHP 5.3.2)

PHP (Hypertext Pre-Processor), es un lenguaje interpretado de alto nivel embebido en páginas HTML. Su interpretación y ejecución ocurren en el servidor web, en el cual se encuentra almacenado el script, el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual

procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente. (12)

La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. PHP es la opción natural para los programadores en máquinas con Linux que ejecutan servidores web con Apache. Además brinda soporte a una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras y se integra con varias librerías externas, lo que extiende aún más su capacidad.(13)

Selección del lenguaje a utilizar

Todos estos lenguajes estudiados han sido seleccionados para la implementación de las funcionalidades debido a que cada uno realiza una función específica, por ejemplo XHTML se escogió como estándar para las páginas web ya que no sólo realiza las mismas funciones de HTML, sino que cumple con las especificaciones más estrictas de XML; JAVASCRIPT para agregarle dinamismo a la página y CSS para controlar la apariencia de éstas. Además se escogió el lenguaje PHP pues permite una rápida generación dinámica de páginas y se empleará la tecnología AJAX que combina los lenguajes anteriormente mencionados, permitiendo establecer una comunicación asíncrona con el servidor, lo cual tienes sus ventajas, pues permite actualizar parte de la página sin tener que actualizarla completa.

1.4.3 Framework para el desarrollo

El término framework, se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que pueden añadirse las últimas piezas para construir una aplicación concreta. (14)

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

1.4.3.1 Capa de presentación

jQuery 1.5.2

Es un framework para el lenguaje Javascript, que nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con

jQuery se puede obtener ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, acceder al documento HTML (Document Object Model) de una forma más sencilla, modificar la apariencia, el contenido, así como manejar los eventos que tienen lugar en la página, además de que cuenta con un no amplio pero si potente sistema de plugins que permite extender la capacidad de este framework.

Todas estas ventajas con jQuery se obtienen de manera gratuita, ya que el framework tiene licencia para uso en cualquier tipo de plataforma, personal o comercial.(15)

1.4.3.2 Capa de lógica del negocio

Symfony 1.4.11

Es un completo framework diseñado para optimizar 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 compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.(16) Symfony se encarga de todos los aspectos comunes y aburridos de las aplicaciones web, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto.

Características:

- ✓ Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
- ✓ Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server)
- ✓ Compatible solamente con PHP 5, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP.
- ✓ Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- ✓ Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- ✓ Flexible hasta cualquier límite y extensible mediante un completo mecanismo de plugins. (16)

1.4.3.3 Capa de acceso a datos

Doctrine 1.2

Es un Mapeador de Objetos Relacionales (ORM) para PHP5 que se encuentra en la parte superior de una capa de abstracción de base de datos de gran alcance. Una de sus principales características es la posibilidad de escribir consultas de base de datos en un lenguaje orientado a objetos de propiedad SQL llamada Doctrine Query Lenguaje, inspirado en Hibernate HQL. Esto proporciona a los desarrolladores una poderosa alternativa a SQL que mantiene la flexibilidad sin necesidad de duplicar código innecesario.(17)

1.4.4 Sistema gestor de base 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 (Data Definition Lenguaje (DDL)), de un lenguaje de manipulación de datos (Data Manipulation Lenguaje (DML)) y de un lenguaje de consulta (Structured Query Language (SQL)). Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

Algunos ejemplos de SGBD son Oracle, PostgreSQL, MySQL, los cuales serán estudiados a continuación.

PostgreSQL 8.4.4

PostgreSQL es un sistema de gestión de bases de datos objeto - relacional (ORDBMS). Este proyecto lleva más de una década de desarrollo, siendo hoy en día el sistema libre más avanzado con diferencia, soportando la gran mayoría de las transacciones SQL y control concurrente, teniendo a su disposición varios lenguajes como por ejemplo C, C++, Java, Python, PHP y muchos más. Cuenta además con herramientas gráficas de administración y diseño de Bases de Datos tales como PgAdmin y phpPgAdmin, estas herramientas permiten que se haga sencilla la administración de las Bases de Datos. (18)

A continuación se enumeran las principales características de este gestor de bases de datos:

- ✓ Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de bits, etc. También permite la creación de tipos propios.

- ✓ 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. (18)

MySQL 5.5.9

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Este gestor de bases de datos es uno de los más usados en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación se debe a que existen infinidad de librerías y otras herramientas que permiten su uso a través de un gran número de lenguajes de programación, además de su fácil instalación y configuración. (18)

Oracle 11.2.0.1

Oracle es básicamente una herramienta cliente/servidor para la gestión de bases de datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales. En el desarrollo de páginas web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, PostgreSQL, MySQL, SQL Server, etc. (19)

Selección del sistema gestor de base de datos a utilizar.

A partir del estudio realizado sobre los gestores de base de datos posibles a emplear para el desarrollo de la solución, se descartó la posibilidad de escoger Oracle como SGBD debido a su elevado precio.

Por su parte, PostgreSQL intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle y bajo la licencia BSD. PostgreSQL ofrece una garantía de integridad en los datos mucho más fuerte que MySQL. Aunque sea más lento respondiendo a una única consulta, PostgreSQL presenta una mejor escalabilidad y rendimiento bajo grandes cargas de trabajo. Tiene mejor soporte para triggers y procedimientos en el servidor. Soporta un subconjunto de SQL92 mayor que el que soporta MySQL, además, tiene ciertas características orientadas a objetos. También es importante decir que MySQL solo se puede utilizar para software que se distribuyen con licencias compatibles con GPL,

mientras que PostgreSQL está bajo licencia BSD, la cual concede ciertas libertades adicionales, entre ellas hacer software propietario; siendo esta una de las razones fundamentales por las que se escogió a PostgreSQL como SGBD.

1.4.5 Entorno de desarrollo integrado (en lo adelante IDE)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por si solos o pueden ser parte de aplicaciones existentes.

Entre los IDEs más utilizados en la actualidad para el desarrollo con php están NetBeans IDE, Zend Studio, PHP Designer, los cuales serán estudiados a continuación.

NetBeans IDE 6.9

El IDE NetBeans es un reconocido entorno de desarrollo integrado de código abierto disponible para Windows, Mac, Linux y Solaris, que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript, Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++. El proyecto de NetBeans está apoyado por una comunidad de desarrolladores dinámica y que ofrece documentación y recursos de formación exhaustivos, así como una amplia selección de complementos de terceros. (20)

Zend Studio 8.0

Son muchos los desarrolladores que trabajan con Zend Studio, es posiblemente uno de los mejores IDE del momento. Se trata de un programa de la casa Zend, uno de los mayores impulsores de PHP, orientada a desarrollar aplicaciones web, en PHP. Zend Studio es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código.(21)

PHP Designer 6.0.2

Php Designer, es un completo entorno de desarrollo y programación especialmente diseñado para desarrolladores de PHP, aunque también permite trabajar con comodidad en otros lenguajes de programación como HTML, XHTML, CSS y SQL. Ofrece toda una serie de asistentes y diálogos integrados que dan facilidades en todo momento, además de acceso directo a librerías de código o scripts

de uso habitual, todo ello en una interfaz de diseño sencilla y elegante que se puede personalizar con 18 temas distintos. Cuenta con cliente de FTP y navegador de ficheros integrados, utilidades de corrección y autocompletado, búsqueda integrada en Google y soporte para proyectos.(22)

Selección del IDE a utilizar

Teniendo en cuenta el estudio realizado acerca de los entornos de desarrollo integrado, así como las características de la solución que se desea implementar, se escogió el NetBeans como IDE a utilizar, ya que presenta las características esenciales de acuerdo a las soluciones que se desean implementar. Este IDE soporta los lenguajes de la web, tanto los del lado del cliente (HTML, CSS, JavaScript) como los del servidor (en este caso PHP), y se integra fácilmente con los frameworks escogidos para el desarrollo (Symfony, jQuery y Doctrine). También cuenta con un sistema de proyectos basado en control de versiones y refactorización, además de ser gratuito y sin prohibiciones de uso.

1.4.6 Servidor web

Software que maneja la entrega de los componentes de las páginas web como respuesta a peticiones de los navegadores de los clientes.

A continuación se realizó un estudio de los servidores web más usados y adecuados para el manejo las peticiones de los navegadores, Lighttpd, que es un servidor asincrónico que se ejecuta como un proceso único con un solo hilo y el servidor Apache que es sin duda, el más conocido basado en procesos.

Lighttpd 1.4

Lighttpd también conocido como Lighty, es considerado una alternativa para el servidor de páginas web Apache. Está diseñado para ser seguro, rápido, compatible con los estándares y flexible a la vez ya que está optimizado para entornos en los cuales la velocidad es crítica. Su huella de memoria es muy pequeña (en comparación a otros servidores web), una ligera carga en el CPU y su enfoque en velocidad hacen de lighttpd perfecto para servidores con demasiada carga. (23)

Apache 2.2

Apache es el servidor web más popular y utilizado a nivel mundial, multiplataforma, extensible y de código abierto, puede soportar un sin número de sistemas operativos. Sus últimas versiones le permiten ejecutarse en modo híbrido multiproceso - multihebra, lo cual mejora la escalabilidad para la mayoría de

las configuraciones, posee la infraestructura necesaria para servir distintos protocolos, es muy rápido y estable en sistemas que no son tipo Unix, tales como BeOS, OS/2 y Windows. (24)

Selección del servidor Web a utilizar

Se utilizará como Servidor Web Apache debido a las características y ventajas que presenta: licencia de software libre y de código abierto, presenta una gran seguridad, lo que lo hace especial respecto a otros servidores como Lighttpd que tienen una mayor probabilidad de contener vulnerabilidades. Además, posee una gran comunidad de usuarios, lo que supone gran cantidad de documentación disponible, que implica la solución a errores tanto del propio servidor como a errores de configuración.

1.5 Conclusiones

El estudio realizado permitió identificar varias de las funcionalidades que se tendrán en cuenta para la posterior implementación de la propuesta de solución. Para el desarrollo de éstas se determinó usar, debido a las características anteriormente mencionadas y explicadas en la selección de cada una de las herramientas, PHP 5 como lenguaje de programación, los frameworks Symfony 1.4.11, JQuery 1.5.2 y Doctrine 1.2, Netbeans 6.9 como IDE, Apache 2.2 como servidor web, la metodología RUP y el lenguaje de modelado UML 2.0, la herramientas CASE Visual Paradigm 6.4 para la modelación de los diagramas necesarios en la investigación y el gestor de base de datos PostgreSQL 8.4.4.

CAPÍTULO II

Características de las Funcionalidades

2.1 Introducción

Los flujos de trabajo Modelado del Negocio y Requerimientos son muy importantes en cualquier proceso de desarrollo de software. Pero no siempre se puede realizar un completo levantamiento del negocio y es aquí donde surge el concepto de modelo de dominio, que puede utilizarse cuando los procesos no están bien claros, ni bien definidos. En cambio, en la captura de requisitos siempre se deben identificar los requisitos funcionales y no funcionales, que permitirán tener con claridad las capacidades y cualidades que todo producto debe tener y con ellos surgen los casos de usos, los actores y el diagrama de casos de uso del sistema, los cuales constituyen artefactos fundamentales para continuar con el desarrollo posterior.

2.2 Modelo de dominio

Debido a que los procesos fundamentales que ocurren en el negocio no se encuentran bien definidos, se decidió realizar un modelo de dominio con el objetivo de comprender los conceptos relacionados con el ambiente en que se desarrollarán las funcionalidades, es decir los conceptos que utilizan los usuarios, con los que trabajan y con los que deberán trabajar las funcionalidades.

2.2.1 Conceptos del dominio

Para realizar el modelo de dominio es necesario identificar los conceptos asociados con el contexto, los cuales se muestran a continuación:

- ✓ Plataforma Educativa ZERA: Software educativo destinado a apoyar el proceso de enseñanza - aprendizaje.
- ✓ Hiperentornos de Aprendizaje: Combinación de diferentes tipologías de software, sustentadas en una tecnología hipermedia.
- ✓ Módulos: Conjunto de elementos fundamentales que componen el producto.
- ✓ Contenidos: Módulo que tiene como principal objetivo permitir la navegación por cada uno de los contenidos de las materias a través del índice de contenido.
- ✓ Tareas: Módulo que tiene como principal objetivo agrupar las tareas que puede realizar un estudiante.

- ✓ Docente: Módulo que tiene como principal objetivo agrupar las acciones específicas que se le asignan al docente dentro de los hiperentornos.
- ✓ Grupo: Conjunto de estudiantes, asociados a un profesor y a un programa de estudio.
- ✓ Estudiante: Persona que cursa estudios para obtener el grado de bachillerato.
- ✓ Rol Docente: Persona responsable de educar, guiar y supervisar a los estudiantes en el proceso de enseñanza - aprendizaje.
- ✓ Rol Director: Persona encargada de dirigir una escuela, ya sea porque es el dueño o porque se le asignó la tarea.
- ✓ Programa de Estudio: Es la adecuación del macro índice al plan de estudios del sistema educativo y de la institución.
- ✓ Evidencia: Es un fichero que realiza el estudiante en respuesta a una tarea asignada por el profesor.
- ✓ Índice: Árbol de contenido donde están asociados los capítulos temas y subtemas.
- ✓ Materiales: Son cualquier elemento que contenga una información determinada.
- ✓ Artículo de Interés: Son documentos que poseen alguna información de interés para mostrar.
- ✓ Período Evaluativo: Es un espacio de tiempo establecido para evaluar una parte de una asignatura.
- ✓ Avance Programático: Contiene un resumen de todas las secciones de clases que debe dar un docente.
- ✓ Registro de Evaluaciones: Contiene las evaluaciones de un grupo de estudiantes.
- ✓ Registro de Asistencia: Contiene las asistencias de un grupo de estudiantes.
- ✓ Anecdotario: Contiene los apuntes generados por el docente.
- ✓ Manual de Usuario: Documento que constituye una ayuda de cómo trabajar en los hiperentornos.
- ✓ Equivalencias: Relación que existe entre el contenido mostrado en la plataforma con el programa de estudio.
- ✓ Recomendaciones de Uso: Orientaciones sobre cómo utilizar determinado recurso.
- ✓ Sugerencias de Evaluación: Orientaciones respecto a la forma en que se puede planear y realizar las evaluaciones.

2.2.2 Diagrama del modelo de dominio

El diagrama del modelo de dominio realizado a continuación muestra las relaciones que existen entre los conceptos identificados anteriormente, los cuales están representados por clases.

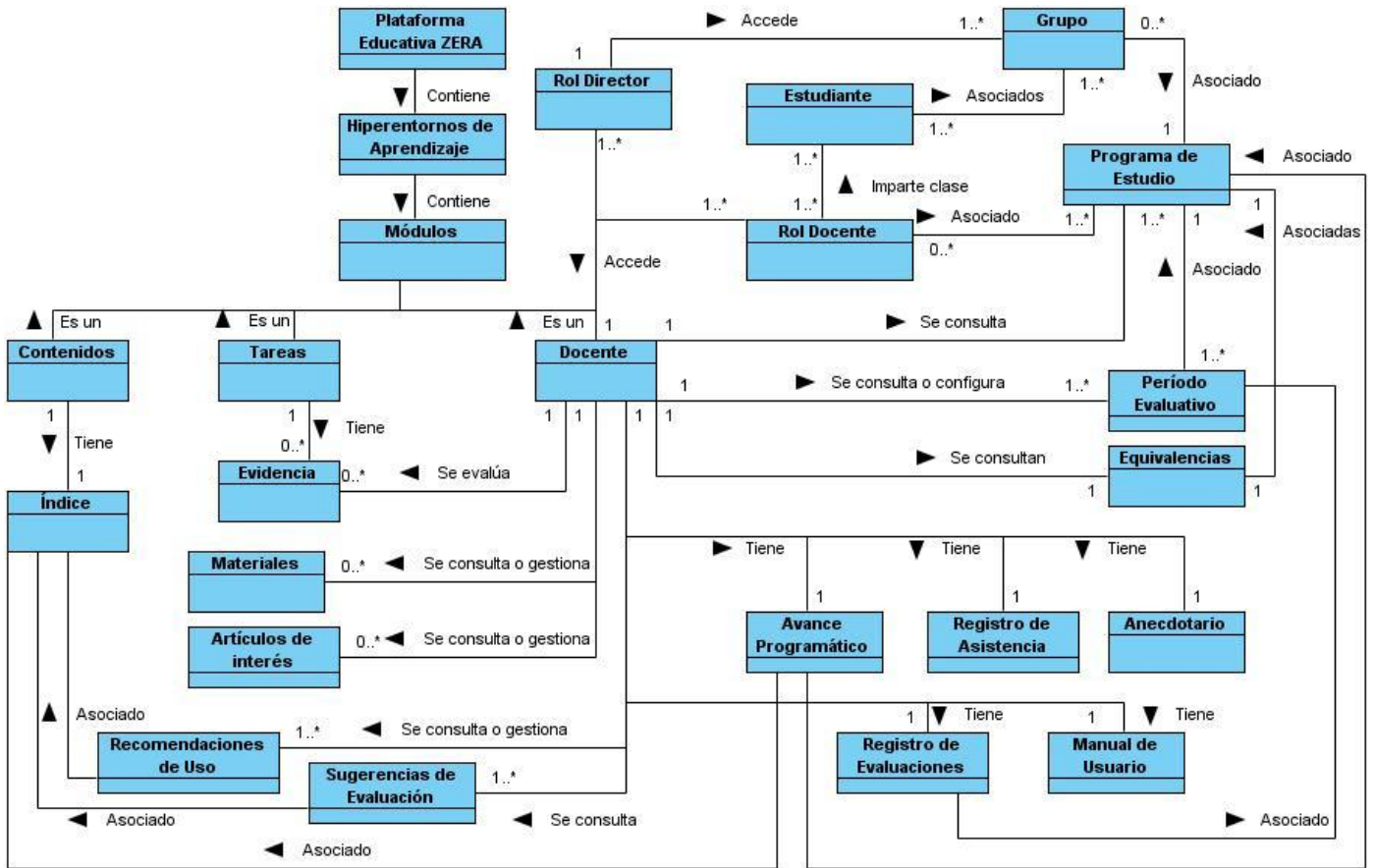


Figura 9. Diagrama del modelo de dominio.

2.3 Descripción de las funcionalidades propuestas

Las funcionalidades propuestas se localizarán en el módulo Docente, módulo encargado de agrupar la información y los contenidos específicos para los profesores, el acceso al mismo se realiza al ingresar a los hipertextos con el rol docente o director, y se tiene acceso a el módulo a través del menú principal. Éste se divide en 6 secciones, donde solamente en 4 de ellas se encontrarán las funcionalidades propuestas, las cuales se mostrarán a continuación:

- ✓ Programa: En esta sección se agruparán los elementos referentes a los programas y planes de estudio. Entre las funcionalidades previstas se encuentran la consulta y visualización del programa de estudio asociado al docente o director registrado en la plataforma, así como otros programas de estudio asociados a la materia. También se visualizarán las equivalencias que existen entre el contenido mostrado en la plataforma con el programa de estudio y además el docente o director

podrán registrar su avance en la asignatura, que no es más que ir realizando un resumen de las secciones de clases impartidas por él.

- ✓ **Orientación Metodológica:** En esta sección se agruparán las propuestas metodológicas de uso de la aplicación, entre las que se encuentran: recomendaciones de uso, sugerencias de evaluación, manual de usuario docente y materiales adicionales. Donde las funcionalidades previstas serán la visualización y en algunos casos la realización de consultas de las propuestas metodológicas anteriormente mencionadas y tendrán acceso a esta sección el docente o el director.
- ✓ **Administración del aprendizaje:** En esta sección se agruparán las herramientas que el docente requiere para evaluar y controlar a sus estudiantes, como son: registro de evaluación y asistencia, consulta de los períodos evaluativos y de la evaluación de las evidencias subidas por estudiantes. Entre las funcionalidades previstas se encuentran registrar la asistencia y las evaluaciones, así como modificarlas y visualizar el registro de ambas. También se podrán evaluar las evidencias de un estudiante y además se puede modificar, si la escuela lo permite, los períodos evaluativos que están asociados al programa de estudio. Solamente tiene acceso a esta sección el docente.
- ✓ **Aportaciones del docente:** En esta sección se agruparán los elementos que el docente o director generará, como son: documentos, imágenes, artículos de interés, recomendaciones de uso, anécdotas, entre otros. Donde las funcionalidades previstas serán la gestión de todos estos elementos anteriormente mencionados, es decir el docente o director podrán incluir, ver, modificar o eliminar cada uno de ellos. Tendrán acceso a esta sección el docente o el director.

2.4 Requisitos

En esta sección se especifican los requisitos funcionales y no funcionales identificados hasta el momento, registrándose las capacidades y las cualidades que el sistema debe poseer.

2.4.1 Requisitos funcionales

Los requisitos funcionales identificados captan las capacidades o condiciones que el sistema debe cumplir, es decir estos definen qué es lo que el sistema debe ser capaz de hacer, permitiendo identificar las funcionalidades requeridas.

A continuación se listan los requisitos funcionales identificados:

RF1: Consultar Programas de Estudio

RF1.1: Mostrar el Programa de Estudio asociado al usuario.

RF1.2: Filtrar campos para realizar la consulta.

RF1.3: Buscar Programas de Estudio dado criterios.

RF1.4: Mostrar un listado de Programas de Estudio según los criterios especificados.

RF1.5: Ver datos de un Programas de Estudio.

RF2: Mostrar Equivalencias

RF2.1: Mostrar las equivalencias que existen entre el contenido presente en el software y el Programa de Estudio.

RF3: Gestionar Avance

RF3.1: Incluir un Avance.

RF3.2: Ver datos del Avance.

RF3.3: Modificar datos del Avance.

RF3.4: Eliminar Avance.

RF4: Listar Avances

RF4.1: Mostrar un listado de Avances ordenados por la fecha de creación.

RF5: Consultar Recomendaciones de Uso

RF5.1: Listar las Recomendaciones de Uso asociadas a un capítulo, tema, subtema o recurso interactivo.

RF5.2: Filtrar campos para realizar una consulta.

RF5.3: Buscar las recomendaciones de uso coincidentes con uno o más criterios.

RF6: Mostrar Sugerencias de Evaluación

RF6.1: Mostrar las Sugerencias de Evaluación asociadas a un capítulo, tema, subtema o al Programa de Estudio en general.

RF7: Mostrar Manual de Usuario Docente

RF8: Consultar Materiales Adicionales

RF8.1: Mostrar un listado de Materiales Adicionales.

RF8.2: Filtrar campos para realizar la consulta.

RF8.3: Buscar Materiales Adicionales dado criterios.

RF8.4: Listar los Materiales Adicionales según los criterios especificados.

RF8.5: Ver el material.

RF9: Listar Períodos Evaluativos

RF9.1: Listar Períodos Evaluativos.

RF10: Evaluar Evidencia

RF10.1: Filtrar campos para realizar la consulta.

RF10.2: Buscar Evidencias dado criterios.

RF10.3: Listar Evidencias según los criterios especificados o el estudiante seleccionado.

RF10.4: Ver datos de una Evidencia.

RF10.5: Evaluar una evidencia de un estudiante determinado.

RF11: Gestionar Asistencia

RF11.1: Registrar la Asistencia de los estudiantes.

RF11.2: Ver Registro de Asistencias.

RF11.3: Modificar datos de la Asistencia.

RF12: Gestionar Evaluación

RF12.1: Registrar una Evaluación.

RF12.2: Ver Registro de Evaluaciones.

RF12.3: Modificar datos de una Evaluación.

RF12.4: Ver datos de una Evaluación.

RF13: Consultar Materiales del Docente

RF13.1: Filtrar campos para realizar la consulta.

RF13.2: Buscar Materiales del Docente dado criterios.

RF13.3: Listar Materiales del Docente según los criterios especificados.

RF14: Gestionar Artículo de Interés

RF14.1: Incluir un Artículo de Interés.

RF14.2: Ver los datos de un Artículo de Interés.

RF14.3: Modificar los datos de un Artículo de Interés.

RF14.4: Eliminar un Artículo de Interés.

RF15: Consultar Artículos de Interés

RF15.1: Filtrar campos para realizar la consulta.

RF15.2: Buscar Artículos de Interés dado criterios.

RF15.3: Listar Artículos de Interés según los criterios especificados.

RF16: Gestionar Recomendación de Uso

RF16.1: Incluir Recomendación de Uso.

RF16.2: Ver datos de las Recomendaciones de Uso.

RF16.3: Modificar Recomendación de Uso.

RF16.4: Eliminar Recomendación de Uso.

RF17: Gestionar Anécdota

RF17.1: Incluir Anécdota.

RF17.2: Ver datos de la Anécdota.

RF17.3: Modificar datos de la Anécdota.

RF17.4: Eliminar Anécdota.

RF18: Consultar Anécdotas

RF18.1: Filtrar campos para realizar la consulta.

RF18.2: Buscar Anécdotas dado uno o varios criterios.

RF18.3: Listar Anécdotas según los criterios especificados.

2.4.2 Requisitos no funcionales

Los requisitos no funcionales identificados especifican las propiedades o cualidades que el producto debe tener. Son características que posibilitan que el producto sea atractivo, usable, rápido, confiable, etc.

A continuación se listan los requisitos no funcionales identificados:

De seguridad

- ✓ La plataforma y sus subsistemas deberán cubrir los siguientes elementos de seguridad:
 - Base de datos de usuarios y calificaciones: ofuscada.
 - Base de datos de elementos multimedia y contenidos: encriptados.
 - Comunicación web de transferencia de datos: encriptados.
 - Acceso mono-usuario a la plataforma, estableciendo sesiones de trabajo.
 - Negar el acceso de los usuarios a acciones no permitidas, haciendo uso de credenciales.

De usabilidad

- ✓ Los elementos gráficos como los íconos deberán contar con un tooltip o mensaje flotante que señalen el tipo de recurso al que se refiere.

De eficiencia

- ✓ Cuando un usuario, sin importar su rol, permanece inactivo durante 20 minutos, se cierra la sesión automáticamente.

De soporte

- ✓ Ser generado en tecnología Web para ser accesible a través de Internet.
- ✓ Compatibilidad con los navegadores: Explorer 7.0 o superior, Mozilla Firefox, Opera, Chrome.
- ✓ Siempre que se requiere de un plugin o instalación adicional, la plataforma avisará del requerimiento, dando la información necesaria para obtener dicho elemento requerido (Adobe Acrobat Reader).

Restricciones de diseño e implementación

- ✓ Las tecnologías y lenguajes utilizados para la integración de las funcionalidades implementadas en la máquina del servidor son los siguientes: Symfony 1.4.11, Postgres SQL 8.4.4, PHP 5.3.2, JQuery 1.5.2, Ubuntu 10.04 LTS
- ✓ Los datos se encontrarán respaldados por el proceso de réplica a nivel central.

De hardware

- ✓ Servidores centrales:
 - Dos o más servidores web, con procesador Intel Xeon de 4 núcleos. La cantidad de núcleos depende de la cantidad de usuarios conectados. (Ejemplo: 80,000 usuarios – 7 núcleos) y RAM de 8 GB o más, con posibilidades de expansión en caso de ser necesario.
 - Dos servidores de Bases de Datos, con procesador Intel Xeon de 4 núcleos. La cantidad de núcleos depende de la cantidad de usuarios conectados. (Ejemplo: 80,000 usuarios – 7 núcleos) y RAM 8 GB o más, con posibilidades de expansión en caso de ser necesario.
 - Un servidor de almacenamiento con 300 GB, inicialmente para el almacenamiento de todas las medias.
- ✓ Servidores locales:
 - En las escuelas en dependencia de la cantidad de usuarios pudiese elegirse usar un solo servidor para la web, Base de Datos y Almacenamiento, o de forma separada en dos: 1 para la web y otro para la BD. Pueden ser servidores normales, o máquinas mejoradas en caso de escuelas pequeñas de menos de 200 usuarios con: 1 Procesador CoreDuo, 2GB de RAM, 120 GB de HDD mínimo, aunque puede ser configurable de acuerdo a la cantidad de contenidos que se vayan contratando y banda ancha.

- Equipos de usuario final. Los usuarios finales deberán contar como mínimo con: Procesador Pentium II o superior, 1GB de RAM, 20GB de HDD, bocinas, si no cuentan con servidor local: conexión de banda ancha 256 bps como mínimo y si cuenta con servidor local: acceso a red interna.

2.5 Patrones de caso de uso

En esta sección se analizan los patrones a utilizar en la realización de los casos de usos. Los cuales permiten con más precisión reflejar los requisitos reales.

Los patrones de caso de uso a utilizar son los siguientes:

- ✓ **Concordancia (Commonality):** Se encarga de extraer una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado, en otras palabras son los casos de usos extendidos o incluidos, un ejemplo lo constituye el caso de uso Gestionar Recomendación de Uso.
- ✓ **CRUD (Creating, Reading, Updating, Deleting):** Se basa en la fusión de casos de uso simples para formar una unidad conceptual, en otras palabras son los casos de uso representados por la palabra “Gestionar”, un ejemplo lo constituye el caso de uso Gestionar Asistencia.
- ✓ **Múltiples actores:** Dentro de este patrón se encuentra el de roles comunes, en éste puede suceder que los dos actores jueguen el mismo rol sobre el caso de uso. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso. Este patrón se puede observar en la figura 10.

2.6 Modelo de casos de uso del sistema

En esta sección se identifican los actores del sistema, se realiza el diagrama de casos de uso y las descripciones de los casos de usos detectados.

2.6.1 Descripción de los actores del sistema

Los actores identificados son personas que van a inicializar e interactuar con los casos de uso del sistema.

| Actor | Descripción |
|---------|---|
| Docente | Actor que realiza las acciones que le permiten supervisar el aprendizaje de los estudiantes |

| | |
|-----------------|---|
| | que pertenecen al o los grupos que el mismo está asociado, además de las acciones que realiza el actor Usuario Docente. |
| Director | Actor que realiza la misma función que el Usuario Docente, pero además puede realizar cualquier otra acción del hiperentorno, debido a que este actor es la persona encargada de dirigir la escuela y por lo tanto tiene acceso a todo el hiperentorno. |
| Usuario Docente | Actor que representa la generalización de los actores Docente y Director. Es el encargado de consultar la información necesaria para prepararse. Puede además subir información o documentación al hiperentorno, tanto para ser consultada por los estudiantes como por el mismo. |

Tabla 1. Descripción de los actores del sistema.

2.6.2 Diagrama de actores del sistema

A continuación se muestra el diagrama de actores, en el que se representa una relación de generalización – especialización, donde al actor Usuario Docente constituye la generalización de los actores Director y Docente, los cuales son la especialización.



Figura 10. Diagrama de actores del sistema.

2.6.3 Diagrama de casos de uso del sistema

El diagrama de casos de uso del sistema muestra las relaciones que existen entre los casos de uso y los actores, así como las relaciones entre casos de usos.

En el diagrama que se muestra a continuación se presentan los 18 casos de uso identificados, basados en los requisitos funcionales, además de otros casos de usos relacionados con éstos. También se visualiza la existencia de 3 actores donde se evidencia una relación de generalización – especialización, el Usuario Docente representa la generalización de los actores Docente y Director, siendo el encargado de inicializar los siguientes casos de uso: Listar y Gestionar Avance, Mostrar Equivalencias, Consultar y Gestionar Anécdotas, Consultar Materiales del Docente, Mostrar Manual de Usuario Docente, Consultar Materiales

Adicionales, Consultar Programas de Estudio, Mostrar Sugerencias de Evaluación, Consultar y Gestionar Artículos de Interés y Consultar y Gestionar Recomendaciones de Uso; el actor Docente que además inicializa los siguientes casos de uso: Gestionar Asistencia, Gestionar Evaluación, Evaluar Evidencia y Listar Períodos Evaluativos; y el director que por ser la máxima autoridad en una escuela tiene el derecho de realizar determinadas acciones. En el diagrama también está presente la relación de extensión entre casos de uso, situación en la que el caso de uso de extensión no es necesario que ocurra y cuando lo hace ofrece un valor extra al objetivo original del caso de uso base, un ejemplo lo constituyen los casos de uso Gestionar Anécdota, Gestionar Artículo de Interés, Gestionar Recomendación de Uso y Gestionar Avance.

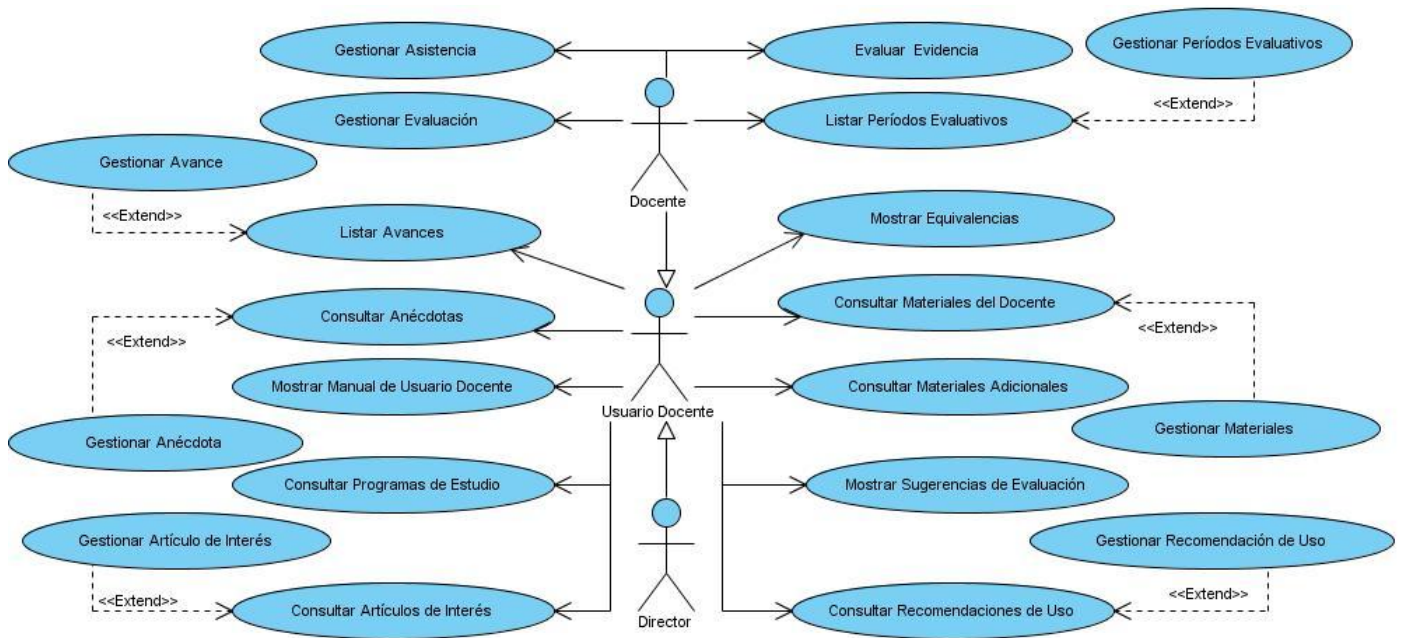


Figura 11. Diagrama de casos de uso del sistema.

2.6.4 Descripción de los casos de uso del sistema

Los casos de uso identificados describen la secuencia de interacciones que se producen entre un actor y el sistema. Estableciendo un acuerdo entre clientes y desarrolladores sobre las condiciones y cualidades que deben cumplir las funcionalidades.

A continuación se presenta la descripción del caso de uso Gestionar Asistencia. Para el estudio de las demás descripción remitirse al Anexo 1: Descripciones de los casos de usos.

CU Gestionar Asistencia

Objetivo

Registrar o modificar la asistencia de los estudiantes y ver el Registro de Asistencia.

Actores

Docente (Inicia): Registra o modifica la asistencia y ve el Registro de Asistencia.

Resumen

El caso de uso se inicia cuando el actor selecciona la opción Asistencia. El actor puede registrar o modificar la Asistencia. En caso que seleccione la opción de registrar la Asistencia, el sistema brinda la posibilidad de insertar los datos que se necesiten. Si el actor selecciona la opción de modificar la Asistencia, el sistema mostrará los datos que pueden ser editables dentro de ésta, y una vez realizados los cambios guardará las modificaciones. En caso de seleccionar la opción de ver el Registro de Asistencia, el sistema mostrará el listado de estudiantes con todos los días que se pasó la asistencia. El caso de uso termina.

Complejidad

Media.

Nivel

Usuario.

Precondiciones

Debe haberse generado el escritorio de trabajo del usuario autenticado.

Para modificar una Asistencia, debe estar seleccionada.

Debe poseer el usuario el rol docente.

Poscondiciones

Se registró o modificó la Asistencia por el actor, así como se vio el Registro de Asistencia.

Flujo de eventos

Flujo básico

| Acciones del actor | Respuesta del sistema |
|--|---|
| 1. El caso de uso se inicia cuando el actor selecciona la opción Asistencia. | |
| | 2. Muestra el listado de los estudiantes del grupo por el cual entró el profesor y permite seleccionar: |

| | |
|--|--|
| | <ul style="list-style-type: none"> • Fecha (aparece por defecto la fecha actual). • Asistencia por cada estudiante, siendo los estados: presente, justificado o injustificado (aparece por defecto presente). • Grupo (Aparece por defecto el grupo por el que entró el profesor, pero puede seleccionar otros grupos a los cuales el profesor imparte clase). <p>Brinda la posibilidad de introducir:</p> <ul style="list-style-type: none"> • Observaciones por cada estudiante. <p>Y permite:</p> <ul style="list-style-type: none"> • Guardar la asistencia de los estudiantes. • Ver Registro de Asistencia. Ver Sección 1: “Ver datos del Registro de Asistencia”. |
| 3. | Selecciona la fecha, la asistencia, el grupo o las observaciones de cada estudiante en caso de que sea necesario. |
| 4. | Selecciona la opción de guardar la asistencia de los estudiantes. |
| | 5. Valida los datos. |
| | 6. Registra la Asistencia en el Registro de Asistencia. |
| | 7. Muestra un mensaje de confirmación. |
| | 8. Muestra el Registro de Asistencia. Ver Sección 1: “Ver datos del Registro de Asistencia”. |
| | 9. El caso de uso termina. |
| Flujo alternativo | |
| 3.a El actor selecciona la opción de ver el Registro de Asistencia. | |
| Acciones del actor | Respuesta del sistema |
| | 3.a.1 Brinda la posibilidad de ver el Registro de Asistencia. Ver Sección 1: “Ver Registro de Asistencia”. |
| | 3.a.2 El caso de uso termina. |
| 3.b El actor selecciona la opción de ver el Registro de Asistencia y no hay asistencias registradas. | |
| Acciones del actor | Respuesta del sistema |

| | |
|--|---|
| | 3.b.1 Muestra un mensaje de información. |
| | 3.b.2 El caso de uso termina. |
| Sección 1: “Ver Registro de Asistencia” | |
| Flujo básico | |
| Acciones del actor | Respuesta del sistema |
| 1. Selecciona la opción de ver el Registro de Asistencia. | 2. Muestra el listado de estudiantes del grupo y el número, así como la asistencia de cada uno de ellos por día y el por ciento de presente, justificado e injustificado de cada uno. Y permite: <ul style="list-style-type: none"> • Pasar la asistencia. • Modificar datos de la Asistencia. Ver Sección 2: “Modificar datos de la Asistencia”. • Exportar a Excel. |
| 3. Selecciona la opción de pasar la Asistencia | 4. Regresa al flujo básico del caso de uso. |
| | 5. El caso de uso termina. |
| Flujo alterno | |
| 3.a El actor selecciona la opción de Modificar los datos de la Asistencia. | |
| Acciones del actor | Respuesta del sistema |
| | 3.a.1. Brinda la posibilidad de modificar los datos de la Asistencia. Ver Sección 2: “Modificar datos de la Asistencia”. |
| | 3.a.2. El caso de uso termina. |
| 3.b El actor selecciona la opción exportar a Excel. | |
| Acciones del actor | Respuesta del sistema |
| | 3.b.1 Exporta a Excel |
| | 3.b.2. El caso de uso termina. |
| Sección 2: “Modificar datos de la Asistencia” | |

| Flujo básico | |
|--|--|
| Acciones del actor | Respuesta del sistema |
| 1. Selecciona la opción de modificar la Asistencia. | 2. Muestra los datos de la Asistencia y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes. Permite además: <ul style="list-style-type: none"> • Actualizar los datos. • Cancelar la operación en cualquier momento. |
| 3. Modifica los datos que necesite y selecciona la opción de actualizar los datos. | 4. Valida los datos. |
| | 5. Actualiza los datos del Registro de Asistencia. |
| | 7. Muestra un mensaje de confirmación. |
| | 8. Muestra los datos del Registro de Asistencia. Ver Sección 1: “Ver Registro de Asistencia”. |
| | 9. El caso de uso termina. |
| Flujo alterno | |
| *.a El actor selecciona la opción de cancelar. | |
| Acciones del actor | Respuesta del sistema |
| | *.a.1 Elimina los datos creados. |
| | *.a.2 Muestra un mensaje de información. |
| | *.a.3 Regresa a la vista anterior. |
| | *.a.4 El caso de uso termina. |

Tabla 2. Descripción del caso de uso Gestionar Asistencia.

2.7 Conclusiones

Los artefactos generados servirán como punto de partida para el análisis y diseño posterior. Donde el modelo de dominio obtenido ayudó a comprender el entorno en que se relacionan las funcionalidades y los requisitos funcionales y no funcionales permitieron definir las capacidades y las cualidades que deben tener las funcionalidades. También se pudo a través de un diagrama de casos de usos del sistema obtener las relaciones que se establecen entre actores y casos de uso, además de obtener las descripciones de los casos de usos.

CAPÍTULO III

Análisis y Diseño de las Funcionalidades

3.1 Introducción

El tercer flujo de trabajo que propone la metodología RUP es el de Análisis y Diseño. En él se generan varios artefactos como el Modelo de Análisis (donde se identifican las clases del análisis y se realizan los diagramas de clases) y el Modelo de Diseño (encargado de realizar los diagramas de clases con estereotipos web). Otro aspecto que se debe definir es el patrón arquitectónico y los patrones de diseño a utilizar, así como el modelo de entidad relación de la base de datos y el diagrama de despliegue.

3.2 Modelo de análisis

En esta sección se modelan las clases del análisis, éstas proporcionan una vista interna del sistema, estructurado por clases estereotipadas (ver figura 12), las cuales son mencionadas a continuación:

- ✓ Entidad: Modela la información que posee larga vida y que es a menudo persistente, en otras palabras representa la información reflejada en el caso de uso.
- ✓ Interfaz: Es la encargada de modelar la interacción entre el sistema y sus actores.
- ✓ Control: Coordina la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.



Figura 12. Clases del análisis

3.1.1 Diagramas de clases del análisis

Los diagramas de clases del análisis representan la relación entre las clases que intervienen en los casos de uso. Son utilizados para comprender de forma general la estructura del sistema y sirve como entrada para la etapa de diseño.

A continuación se presenta el diagrama de clases del análisis del caso de uso Gestionar Asistencia. Para el estudio de las demás diagramas remitirse al Anexo 2: Diagramas de clases del análisis.

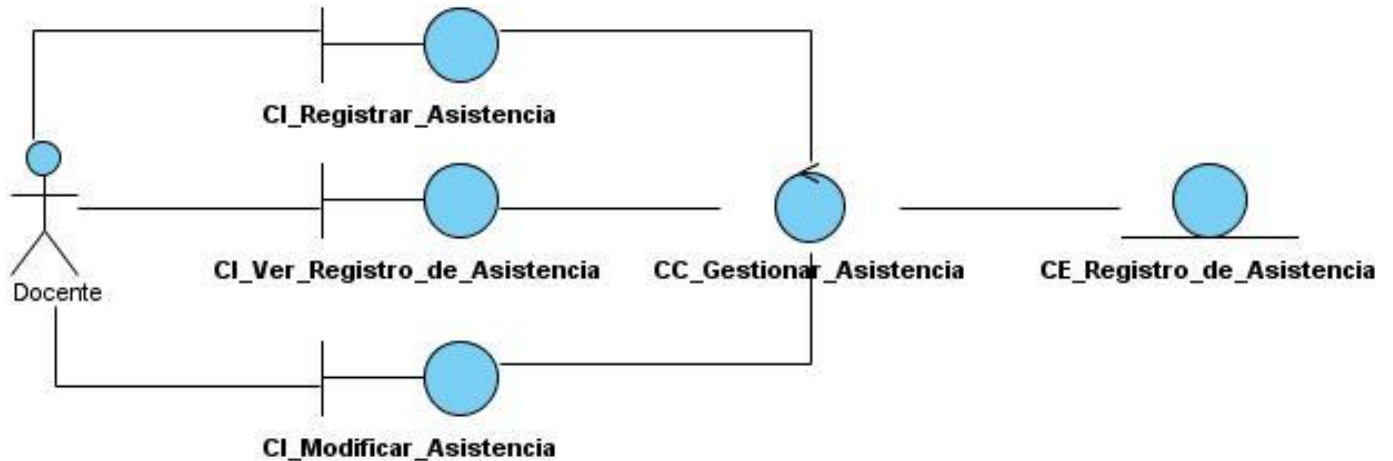


Figura 13. Diagrama de clases del análisis del caso de uso Gestionar Asistencia.

3.3 Patrón arquitectónico Modelo Vista Controlador (en adelante MVC) en Symfony

Para el desarrollo de las funcionalidades se escogió el framework symfony, framework que aunque utiliza MVC, tiene su propia forma de trabajo en este punto, con variantes del mismo como la capa de abstracción de base de datos, el controlador frontal y las acciones. Este patrón separa el código en tres capas:

- ✓ La capa del modelo representa la información con la que trabaja la aplicación, es decir la lógica de negocio (la base de datos pertenece a esta capa). Ésta se puede dividir en la capa de acceso a los datos y en la capa de abstracción de la base de datos, utiliza la librería doctrine que se encarga de la generación automática de las clases que pertenecen a esta capa, en dependencia de la estructura de la base de datos. Symfony guarda todas las clases y archivos relacionados con el modelo en el directorio lib/model/doctrine, ejemplos de estas clases son TbAssistanceTable.class.php y TbAssistance.class.php.
- ✓ La vista es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa), es decir transforma el modelo en una página web. En Symfony la capa de la vista está formada principalmente por plantillas (un ejemplo es recordAssistanceSuccess.php) en PHP y el layout (layout_sin_menu.php) que contiene la información visual común a todas las páginas. Estas plantillas se guardan en varios directorios llamados templates/ repartidos por todo el proyecto.

- ✓ El controlador es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. En Symfony todas las peticiones se canalizan a través de los controladores frontales (`index.php` y `frontend_dev.php`), en este caso los controladores son `bachelor.php` y `bachelor_dev.php`. Estos controladores frontales realmente delegan todo el trabajo en las acciones. Éstas son métodos con el nombre `executeNombreAccion` (ejemplo `executeRecordAssistance.php`) de una clase llamada `nombreModuloActions` (en este caso `eTeachingActions`) que hereda de la clase `sfActions` y se encuentran agrupadas por módulos. La clase que representa las acciones de un módulo se encuentra en el archivo `actions.class.php`, en el directorio `actions/` de `sTeaching`.

En general en symfony el código de la presentación se guarda en la vista, el código de manipulación de datos en el modelo y la lógica de procesamiento de las peticiones constituye el controlador, esto resulta bastante útil además de restrictivo ya que obliga a dividir y organizar el código de acuerdo a lo establecido por el patrón MVC.

3.4 Aplicación de los patrones de diseño en Symfony

Una de las principales ventajas de los framework de desarrollo es que están basados en patrones de diseño, característica que hace posible la gran usabilidad que tienen, casi siempre independientes del tipo de aplicación web que se desea desarrollar. El uso del framework Symfony, evidencia el uso de varios patrones de diseño que este trae incluidos por defecto en su arquitectura, además de estar concebidos de tal manera que obliga al programador a aplicarlos como por ejemplo los patrones GRASP.

Patrones GRASP (General Responsibility Assignment Software Patterns)

Experto

En la arquitectura de Symfony, específicamente en el modelo, existen dos tipos de clases fundamentales:

- ✓ Las clases encargadas de la abstracción de datos, es decir las encargadas de hacer las consultas a la base de datos utilizando Doctrine, ya que tienen los atributos necesarios para efectuar dicha función, por tanto tienen la responsabilidad de realizar directamente las acciones con la base de datos y aquí es donde se aplica el patrón Experto.
- ✓ Las clases de acceso a datos son las responsables de interactuar con las clases de abstracción de datos, y devolver los objetos que necesitan los controladores en su forma original, es decir ellas no

conocen los atributos para interactuar con la base de datos, lo que hacen es que le avisan a las clases de abstracción lo que tienen que hacer, ya que éstas si tienen los datos necesarios para ejecutar esta acción.

Creador

En Symfony hay clases que aplican este patrón, un ejemplo de esto es en las clases “action” cuando se desea usar una instancia de una clase del modelo, por ejemplo se tiene en la clase “eTeachingAction” del módulo eTeaching, la acción AjaxLoadStudyProgramById, toma el identificador de un programa de estudio y desea ver sus datos, para esto necesita hacer uso de una instancia de ese programa de estudio. En este caso el “eTeachingAction” usa al objeto StudyProgramTable, por tanto puede crear instancias de él evidenciándose este patrón.

Bajo acoplamiento

Este patrón se evidencia dentro del framework Symfony en la capa modelo ya que las clases de acceso a los datos tienen bastante independencia de las clases de abstracción de datos. Hay poca dependencia entre esas clases lo que permite una mayor reutilización. Ejemplos de estas clases son TbAssistanceTable.class.php y TbAssistance.class.php.

Alta cohesión

Una de las características principales del framework Symfony es la organización del trabajo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo, la clase “eTeachingAction” del módulo eTeaching contiene varias funcionalidades en las que cada una posee un propósito único, no desempeñado por el resto de los elementos, siendo estas funcionalidades las encargadas de controlar las acciones de las plantillas. Esto hace posible que el software sea flexible a cambios sustanciales con efecto mínimo, garantizando la alta cohesión.

Controlador

Un ejemplo del patrón controlador es fácil de encontrar, por ejemplo se puede ver desde la clase sfFrontController, sfFrontWebController, sfContext, los “actions”, el index.php del ambiente, etc. La arquitectura del framework (MVC) nos ayuda desde el principio, hay una capa específicamente para los controladores, que son el núcleo del mismo. Symfony aplica el patrón “Front Controller” (Controlador frontal) y por tanto tiene una estructura bien organizada de controladores, que parte desde el “index.php”

del ambiente y terminan en los “actions”. Aquí cada clase en esta capa tiene su responsabilidad y es única, hay controladores que se encargan de la seguridad del sistema trabajando con ficheros YML, otros sólo velan por identificar mediante unos datos (peticiones web) las clases (actions) que deben realizar determinadas tareas.

3.5 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en las funcionalidades que se desean implementar.

3.5.1 Diagrama de clases del diseño

Los diagramas de clases del diseño son una representación más concreta y detallada que los diagramas de clases del análisis, aunque también representan la parte estática del sistema conteniendo las clases y sus relaciones. Son empleados para representar las relaciones que se establecen entre las clases.

A continuación se describen de manera general el significado de los principales elementos presentes en los diagramas de clases del diseño para lograr una mejor comprensión de éstos. Es importante mencionar que los paquetes modelo, controlador y vista se usan para distribuir las clases según el patrón Modelo-Vista-Controlador que propone Symfony. Los diagramas se modelaron siguiendo este significado, las diferencias que existen entre ellos se deben a las características independientes de cada caso de uso.

- ✓ **Controlador:** Una parte importante de su trabajo es común a todos los controladores de la aplicación. Entre las tareas comunes se encuentran el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Por este motivo, el controlador se ha dividido en un controlador frontal (en este caso `bachelor.php`), que se encarga de realizar las tareas comunes y el actions (`eTeachingActions.php`), que incluyen el código específico del controlador de cada página.
- ✓ **Vista:** Las páginas web suelen contener elementos que se muestran de forma idéntica a lo largo de toda la aplicación: cabeceras de la página, el layout genérico, el pie de página y la navegación global. En la mayoría de las veces sólo cambia el interior de la página. Por este motivo, la vista se separa en un layout (representado en este caso por el `layout_sin_menu.php`) y en una plantilla (el nombre de las

plantillas está compuesto por el nombre de la acción que la origina seguido por sufijo que puede ser SUCCESS, ejemplo recordAssistanceSuccess.php). El layout será global en toda la aplicación. La plantilla sólo se encarga de visualizar las variables definidas en el paquete del controlador.

- ✓ Modelo: Se encuentran las clases (en este caso TbAssistanceTable y TbAssistance) encargadas de la abstracción y acceso a datos, las cuales son generadas por el ORM (en inglés, Object Relational Model) Doctrine, de forma tal que no se accederá directamente a la Base de Datos y controlando la información que guardará el sistema.
- ✓ Componente Symfony: Los componentes de Symfony se encapsulan en un subsistema que contiene todos los elementos del framework que actúan o intervienen en la realización de los casos de uso, (dígase validators, helpers, templates, forms, filters, partials, etc), pero al ser propios de Symfony, la manera en que trabajan es transparente al programador.
- ✓ Doctrine ORM: Su función es gestionar el acceso a la base de datos. Implica que el acceso y la modificación de los datos almacenados en la base de datos se realicen mediante objetos, nunca de forma explícita, permitiendo un alto nivel de abstracción y fácil portabilidad.

A continuación se presenta el diagrama de clases del diseño correspondiente al caso de uso Gestionar Asistencia. Para el estudio de los demás diagramas de clase del diseño remitirse al Anexo 3: Diagramas de clases del diseño.

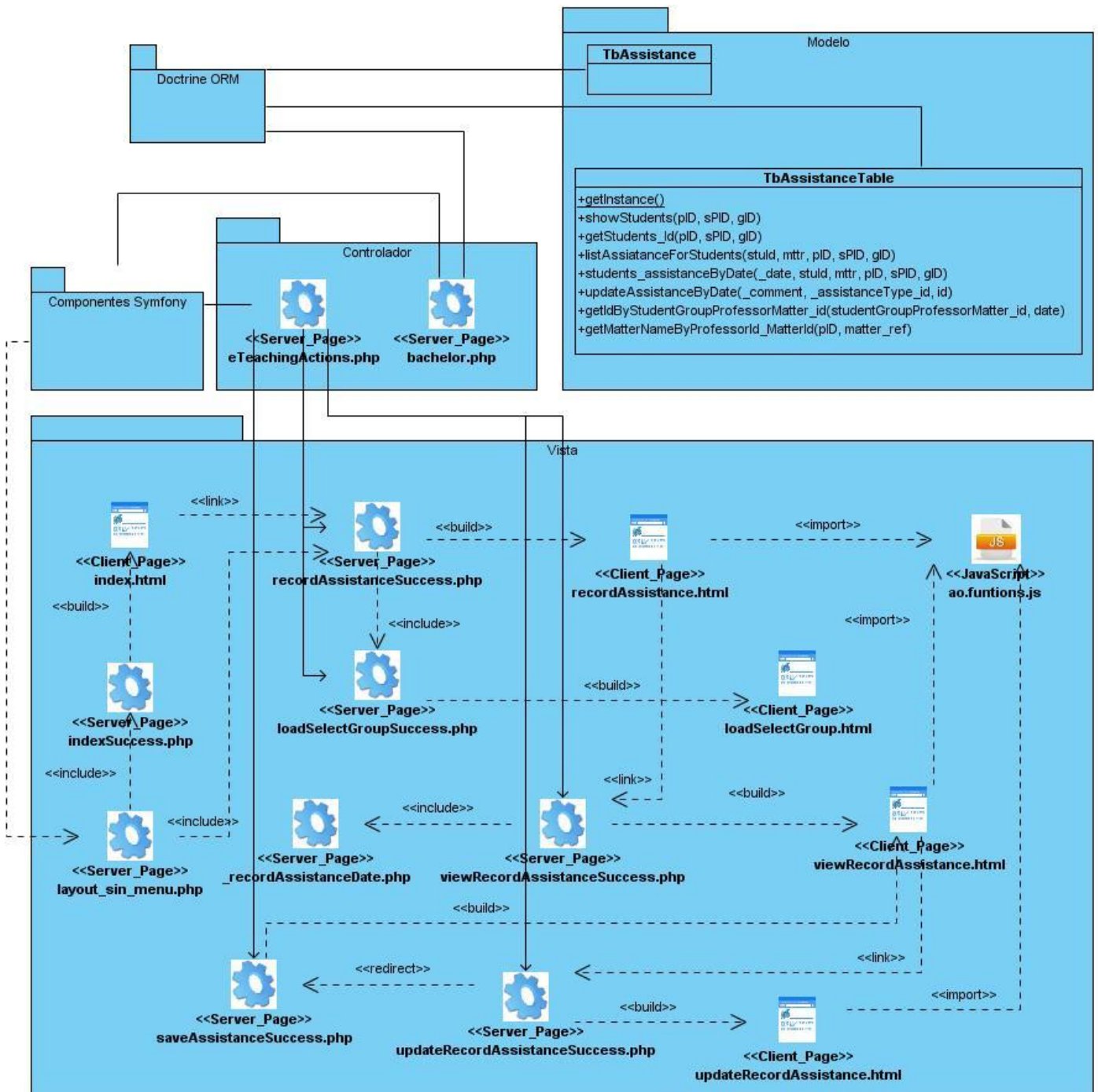


Figura 14. Diagrama de clases del diseño del caso de uso Gestionar Asistencia.

Los diagramas de secuencia del diseño por casos de usos realizados pueden encontrarse en el Anexo 4: Diagramas de secuencia del diseño.

3.6 Diagrama de despliegue

El diagrama de despliegue que se muestra a continuación representa la distribución física del sistema a través de nodos. Está compuesto por una PC Cliente que deberá tener instalado un navegador Web, donde la comunicación entre ella y los servidores se llevará a cabo a través del protocolo HTTP, estos servidores se comunicarán vía TCP-IP, en uno será instalado el servidor Web de la escuela y el otro para el servidor Web de Alfaomega, también contiene dos servidores de base de datos uno para la escuela y otro para Alfaomega, así como otros dos servidores de almacenamiento (destinado a las medias), donde igualmente existirá una para la escuela y otro para Alfaomega.

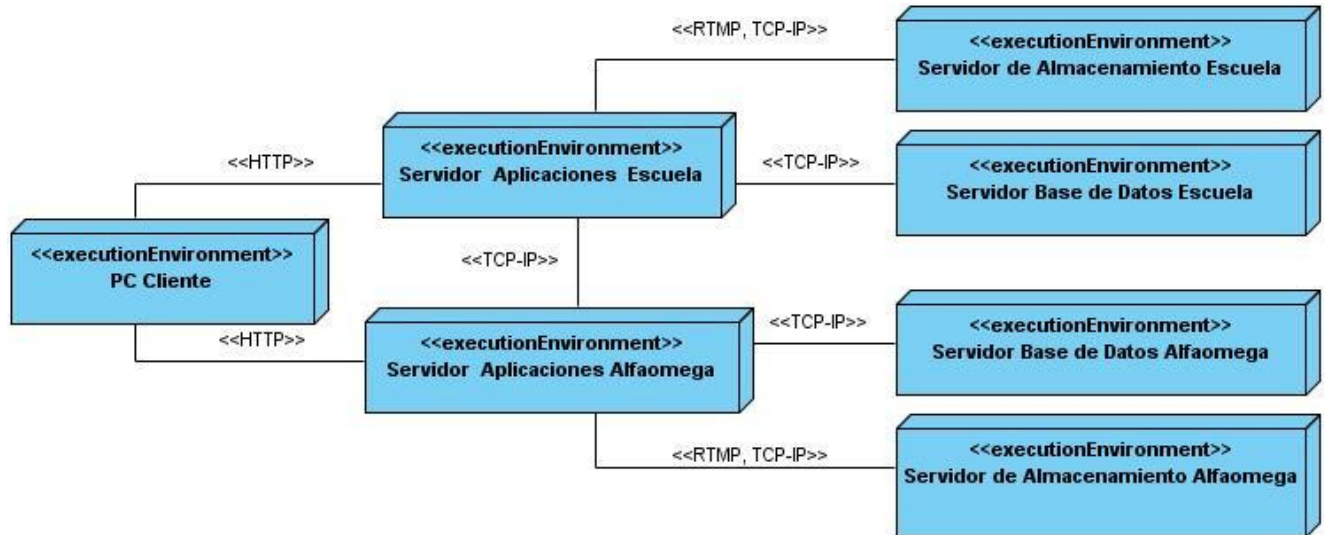


Figura 15. Diagrama de despliegue.

3.7 Diseño de la base de datos

En esta sección se presenta el modelo de datos que estará compuesto por las entidades que pasarán a ser las tablas de la base de datos, para ser utilizadas por las funcionalidades a desarrollar, éste se encuentra simplificado, es decir las tablas sólo contienen los identificadores, para una mejor visualización, además se describen las tablas más importantes que la conforman.

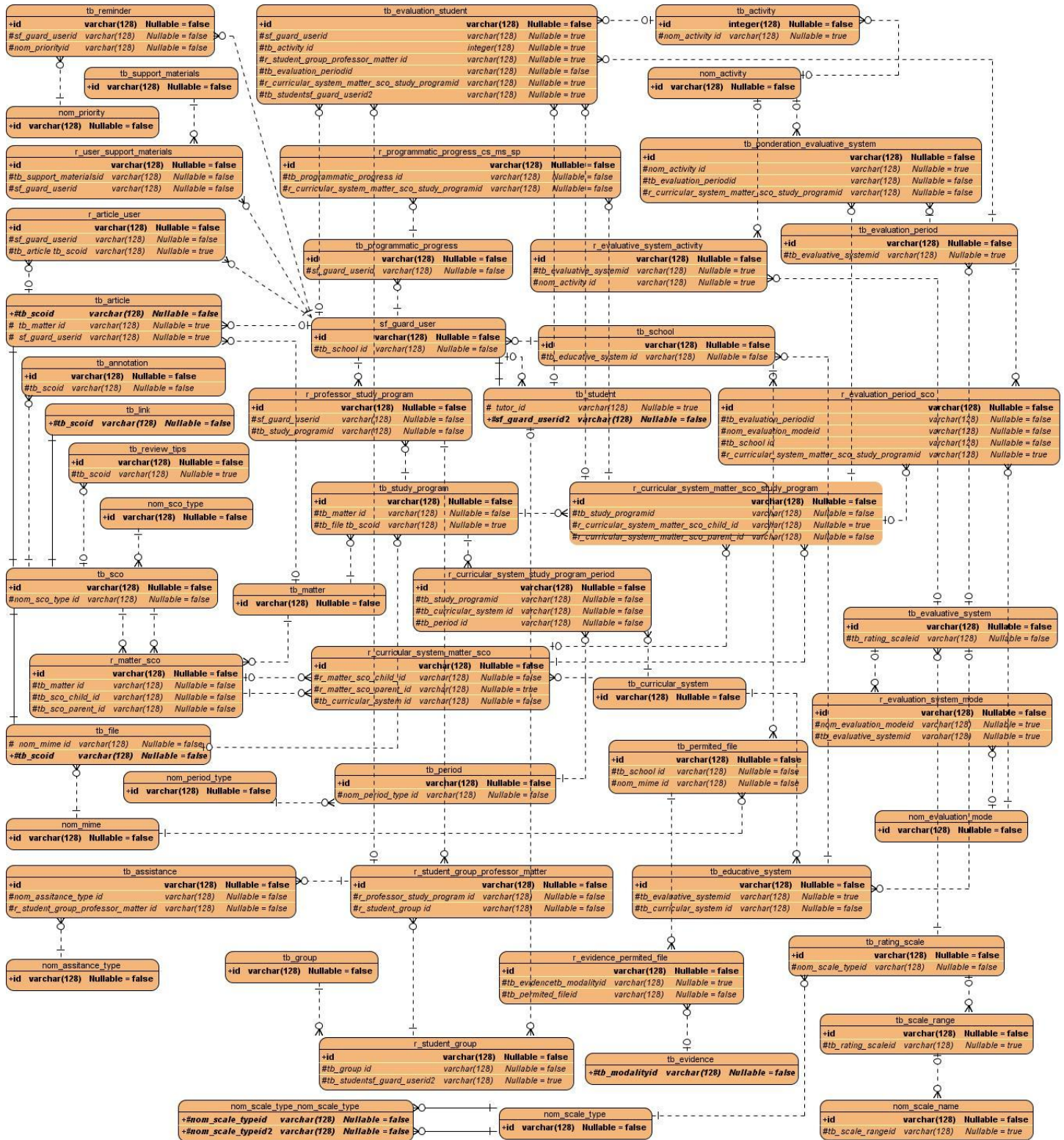


Figura 16. Modelo de datos

3.7.1 Descripción de las tablas

A continuación se describen las tablas de la base de datos relacionadas con el caso de uso Gestionar Asistencia, las demás pueden encontrarse en el Anexo 5: Descripciones de las tablas de la base de datos.

| tb_assistance | | |
|--|--------------|--|
| Descripción: Esta tabla almacena los datos de la asistencia de los estudiantes. | | |
| Atributo | Tipo | Descripción |
| id | Varchar(128) | Es el identificador de cada registro de la tabla. |
| comment | Varchar(250) | Contiene un comentario sobre el estado de la asistencia del estudiante. |
| date | Date | Es la fecha en que se pasó la asistencia. |
| student_group_professor_matter_id | Varchar(128) | Es el identificador de la tabla que guarda relación entre el estudiante, el grupo, el profesor y la materia. |
| assistance_type_id | Varchar(128) | Es el identificador del estado de la asistencia. |

Tabla 3. Descripción de la tabla tb_assistance

| nom_assistance_type | | |
|---|--------------|---|
| Descripción: Esta tabla describe los estados de la asistencia. | | |
| Atributo | Tipo | Descripción |
| id | Varchar(128) | Es el identificador de cada registro de la tabla. |
| name | Varchar(128) | Nombre de un estado. |
| type | Varchar(128) | Representa una abreviatura del estado. |
| description | Text | Describe el significado del estado. |

Tabla 4. Descripción de la tabla nom_assistance_type

| r_student_group_professor_matter | | |
|---|--------------|---|
| Descripción: Esta tabla representa la relación que se establece entre estudiante, grupo, profesor y materia. | | |
| Atributo | Tipo | Descripción |
| id | Varchar(128) | Es el identificador de cada registro de la tabla. |

| | | |
|----------------------------|--------------|--|
| professor_study_program_id | Varchar(128) | Representa el identificador de la relación entre un profesor y un programa de estudio. |
| student_group_id | Varchar(128) | Representa el identificador de la relación entre un estudiante y un grupo. |

Tabla 5. Descripción de la tabla r_student_group_professor_matter

3.8 Conclusiones

Los diagramas de clases del análisis y del diseño que se obtuvieron, fueron enfocados al cumplimiento de los requisitos funcionales y no funcionales. Éstos lograron que se obtuvieran de una forma más concreta y detallada las relaciones que se establecen entre clases. Además, el patrón arquitectónico y los patrones de diseño que se emplean en el desarrollo de las funcionalidades quedaron bien definidos y argumentados. También se realizó el modelo de base datos donde quedaron plasmadas todas las relaciones de las tablas necesarias que forman parte de la investigación y el diagrama de despliegue que permitió obtener de manera detallada los nodos físicos y las asociaciones de comunicación que existen entre ellos.

CAPÍTULO IV

Implementación y Pruebas de las Funcionalidades

4.1 Introducción

Los artefactos generados en el diseño son la entrada fundamental al flujo de trabajo de Implementación, el cual tiene como objetivo, mostrar cómo se implementan las funcionalidades en términos de componentes y cómo éstos se organizan. Posterior a este flujo se encuentra el de Pruebas, donde se realizan las pruebas necesarias a las funcionalidades, y con ellas se verifica que éstas estén correctamente desarrolladas.

4.2 Modelo de implementación

En esta sección se presenta el modelo de implementación, el cual describe cómo los elementos del modelo de diseño se implementan en términos de componentes, ficheros de código, etc. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado, además de cómo dependen los componentes unos de otros. Como parte del modelo de implementación se obtienen los diagramas de componentes que a continuación se presentan.

4.2.1 Diagrama de componentes

Los diagramas de componentes que se muestra a continuación describen los elementos físicos de las funcionalidades y sus relaciones, en otras palabras representan cómo éstos se dividen en componentes y las dependencias existentes entre ellos. Los componentes físicos incluyen archivos, cabeceras, librerías compartidas, módulos, ejecutables o paquetes. Pueden ser usados para modelar y documentar cualquier arquitectura de sistema. Debido a que estos son muy parecidos a los diagramas de casos de usos son utilizados para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. Los diagramas de componentes realizados, muestran cómo están distribuidos según el patrón arquitectónico Modelo-Vista-Controlador que utiliza Symfony.

A continuación se presenta el diagrama de componentes del caso de uso Gestionar Asistencia y la descripción de los principales elementos que lo conforman. Para el estudio de los demás diagramas puede remitirse al Anexo 6: Diagramas de componentes.

- ✓ Paquete Vista: Paquete que agrupa todos los componentes relacionados con la vista del caso de uso, tales como: las plantillas y el layout.
- ✓ Paquete Controlador: Paquete que agrupa todos los componentes actions.class.php donde se gestionan las acciones para la plantilla.
- ✓ Paquete Modelo: Se especifican las clases generadas por Doctrine que es el ORM que utiliza Symfony el cual proporciona persistencia para los objetos y un lenguaje de consultas orientado a objetos.
- ✓ Componente Base de Datos: Encapsula todos los datos del sistema.
- ✓ Componente lib: Almacena las clases y librerías externas, se suele guardar todo el código común a todas las aplicaciones del proyecto. El subdirectorío model/ guarda el modelo de objetos del proyecto.
- ✓ Componente Jquery: Almacena los archivos con código javascript que aportan dinamismo necesario a las páginas web.
- ✓ Componente css: Contiene los archivos que garantizan la correcta presentación de la información en las páginas web.
- ✓ Componente config: Almacena los archivos de configuración que intervienen en la ejecución de los casos de usos.
- ✓ Componente Symfony: subsistema que contiene todos los elementos del framework que actúan o intervienen en la realización de los casos de usos, (dígase helpers, templates, forms, partials, etc).

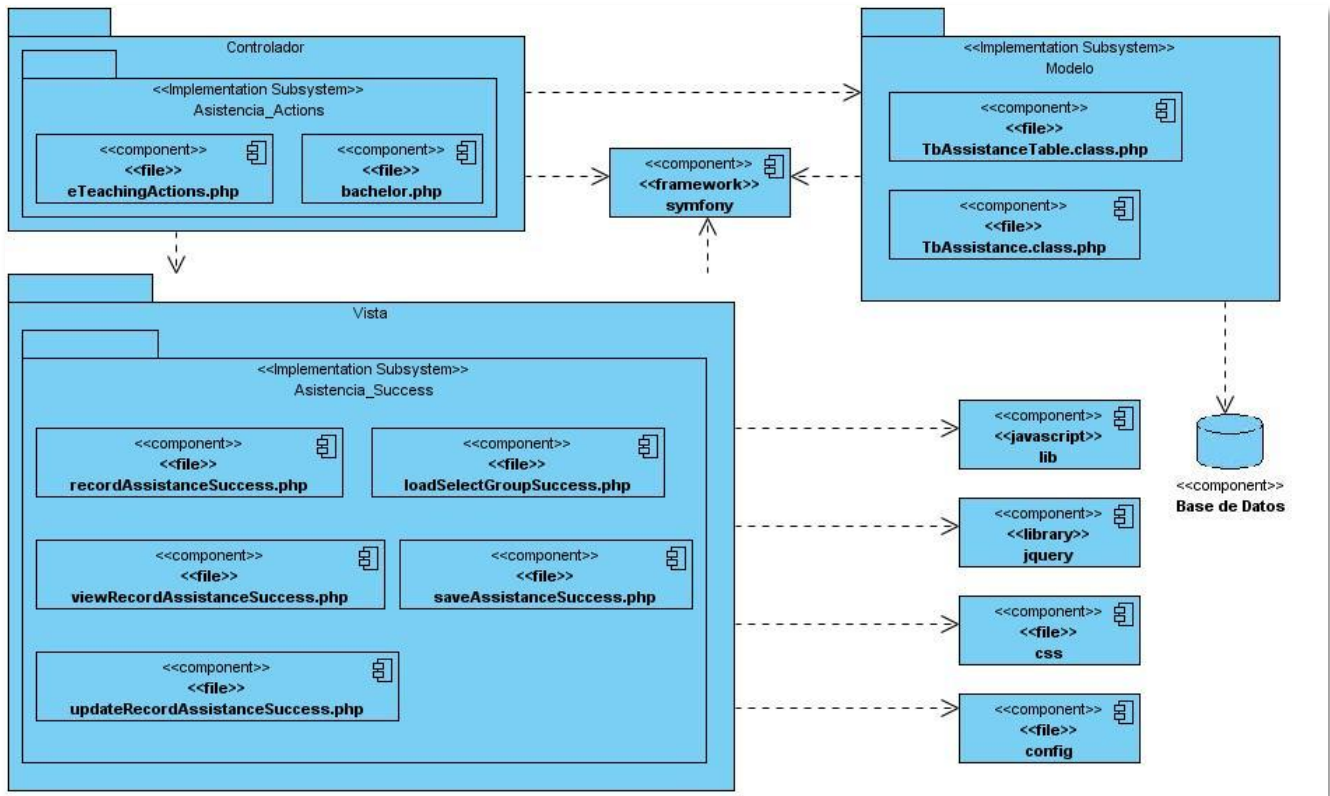


Figura 17. Diagrama de componente del caso de uso Gestionar Asistencia

4.3 Pruebas de software

En esta sección se presentan las pruebas realizadas a las funcionalidades. Éstas se utilizaron para identificar y corregir los fallos de implementación, y de esta forma verificar el perfecto funcionamiento de lo desarrollado.

4.3.1 Niveles de prueba

El proceso de prueba es llevado a cabo en varios niveles, cada uno es realizado en un determinado momento del ciclo de desarrollo de las funcionalidades. A continuación se muestran los niveles de prueba que se emplearon:

- ✓ Prueba de unidad: Es la prueba enfocada a los elementos testeables más pequeño del software. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. La prueba de unidad siempre está orientada a caja blanca (tipo de prueba que comprueba los caminos lógicos del software

proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones y/o bucles.). Antes de iniciar cualquier otra prueba es preciso probar el flujo de datos de la interfaz del componente. Si los datos no entran correctamente, todas las demás pruebas no tienen sentido. El diseño de casos de prueba de una unidad comienza una vez que se ha desarrollado, revisado y verificado en su sintaxis el código a nivel fuente. (25)

- ✓ Prueba de integración: Es ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente cuando son combinados para ejecutar un caso de uso. Se prueba un paquete o un conjunto de paquetes del modelo de implementación. Es el proceso de combinar y probar múltiples componentes juntos. El objetivo es tomar los componentes probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. (25)
- ✓ Prueba de Sistema: Son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados. En un ciclo iterativo estas pruebas ocurren más temprano, tan pronto como subconjuntos de comportamiento de caso de uso son implementados. (25)

4.3.2 Métodos de prueba

En esta sección se describen las pruebas de caja negra y blanca, haciendo más hincapié en las de caja negra, donde con la utilización de casos de pruebas se pudo comprobar la validez en las respuestas de las funcionalidades ante las acciones del usuario y la calidad de las salidas en dependencia de las entradas:

- ✓ Las pruebas de caja negra, también denominadas pruebas funcionales o de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. (26) Éstas se refieren a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de pruebas pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. (25)
- ✓ La prueba de la caja blanca del software denominada a veces prueba de caja de cristal comprueba los caminos lógicos del software proponiendo casos de pruebas que se ejerciten conjuntos específicos de

condiciones y/o bucles. Puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado. (25)

4.3.3 Diseño de casos de prueba.

Los siguientes casos de prueba especifican una forma de probar el sistema, incluyendo los datos de entrada y resultados esperados. Éstos se derivan de los caso de uso y su objetivo fundamental es encontrar la mayor cantidad de defectos en las funcionalidades implementadas.

A continuación se presenta el diseño del caso de prueba perteneciente al caso de uso Gestionar Asistencia, los demás pueden encontrarse en el Anexo 7: Diseños de los casos de pruebas.

| CP Gestionar Asistencia | | | | | | | |
|---|---------------------------------|-------|------------|-------|---------------|--|---|
| Descripción general | | | | | | | |
| El caso de uso se inicia cuando el actor selecciona la opción Asistencia. El actor puede registrar o modificar la Asistencia. En caso que seleccione la opción de registrar la Asistencia, el sistema brinda la posibilidad de insertar los datos que se necesiten. Si el actor selecciona la opción de modificar la Asistencia, el sistema mostrará los datos que pueden ser editables dentro de ésta, y una vez realizados los cambios, guardará las modificaciones. En caso de seleccionar la opción de ver el Registro de Asistencia, el sistema mostrará el listado de estudiantes con todos los días que se pasó la asistencia. El caso de uso termina. | | | | | | | |
| Condiciones de ejecución | | | | | | | |
| Debe haberse generado el escritorio de trabajo del usuario autenticado. | | | | | | | |
| Para modificar una Asistencia, debe estar seleccionada. | | | | | | | |
| Debe poseer el usuario el rol docente. | | | | | | | |
| SC1 Gestionar Asistencia | | | | | | | |
| ID del escenario | Escenario | Fecha | Asistencia | Grupo | Observaciones | Respuesta del sistema | Flujo central |
| EC 1 | Selecciona la opción Asistencia | | | | | Muestra el listado de los estudiantes del grupo por el cual entró el profesor y permite seleccionar: | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ sección Administración del |

| | | | | | | | |
|------|--|---|---|---|---|---|---|
| | a. | | | | | <p>-Fecha (aparece por defecto la fecha actual).</p> <p>-Asistencia por cada estudiante, siendo los estados: presente, justificado o injustificado (aparece por defecto presente)</p> <p>-Grupo.(Aparece por defecto el grupo por el que entro el profesor, pero puede seleccionar otros grupos a los cuales el profesor imparte clase)</p> <p>Brinda la posibilidad de introducir:</p> <p>-Observaciones por cada estudiante</p> <p>Y permite:</p> <p>-Guardar la asistencia de los estudiantes.</p> <p>-Ver Registro de Asistencia. SC2 "Ver Registro de Asistencia".</p> | aprendizaje/ Opción Asistencia |
| EC 2 | Selecciona la fecha, la asistencia, y el grupo o las observaciones de cada estudiante en caso de que sea necesario y la opción guardar | V | V | V | V | <p>Valida los datos.</p> <p>Registra la Asistencia en el Registro de Asistencia.</p> <p>Muestra un mensaje de confirmación.</p> <p>Muestra el Registro de Asistencia. SC2 "Ver Registro de Asistencia".</p> | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ sección Administración del aprendizaje/ Opción Asistencia/ Botón Guardar |

| EC3 | Selección a la opción de ver el Registro de Asistencia. | | | | | Brinda la posibilidad de ver el Registro de Asistencia. SC2 "Ver Registro de Asistencia". | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ sección Administración del aprendizaje/ Opción Asistencia/ícono registro de asistencia |
|---------------------------------------|--|-------|------------|-------|---------------|--|--|
| EC4 | Selección a la opción de ver el Registro de Asistencia y no hay asistencias registradas. | | | | | Muestra un mensaje de información. | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ sección Administración del aprendizaje/ Opción Asistencia/ ícono registro de asistencia |
| SC2 Ver Registro de Asistencia | | | | | | | |
| ID del escenario | Escenario | Fecha | Asistencia | Grupo | Observaciones | Respuesta del sistema | Flujo central |
| EC 1 | Selección a la opción de ver el Registro de Asistencia. | | | | | Muestra el listado de estudiantes del grupo y el número, así como la asistencia de cada uno de ellos por día y el por ciento de presente, justificado e injustificado de cada uno. Y permite: -Pasar la asistencia. -Modificar datos de la Asistencia. SC3 "Modificar datos de la Asistencia". -Exportar a Excel. | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ sección Administración del aprendizaje/ ícono registro de asistencia |

| EC 2 | Selección a la opción de pasar la Asistencia | | | | | Regresa al flujo básico del caso de uso. | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ sección Administración del aprendizaje/ícono registro de asistencia/ botón registrar asistencia |
|---|---|-------|------------|-------|---------------|--|---|
| EC3 | Selección a la opción de Modificar los datos de la Asistencia . | | | | | Brinda la posibilidad de modificar los datos de la Asistencia. SC3 “Modificar datos de la Asistencia”. | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ sección Administración del aprendizaje/ícono registro de asistencia /selecciona una fecha de la tabla |
| EC4 | Selección a la opción exportar a Excel. | | | | | Exporta a Excel. | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ sección Administración del aprendizaje/ícono registro de asistencia/ícono exportar a excel |
| SC3 Modificar datos de la Asistencia | | | | | | | |
| ID del escenario | Escenario | Fecha | Asistencia | Grupo | Observaciones | Respuesta del sistema | Flujo central |
| EC 1 | Selecciona la opción de modificar la Asistencia. | | | | | Muestra los datos de la Asistencia y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes. Permite además -Actualizar los datos. -Cancelar la operación en | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ Sección Administración del aprendizaje/ícono registro de asistencia /selecciona una fecha de la tabla |

| | | | | | | | |
|------|---|----|----|----|----|--|---|
| | | | | | | cualquier momento. | |
| EC 2 | Modifica los datos que necesite y selecciona la opción de actualizar los datos. | V | V | V | NA | Valida los datos. | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ sección Administración del aprendizaje/ícono registro de asistencia /selecciona una fecha de la tabla/ botón actualizar |
| | | V | NA | NA | NA | Actualiza los datos del Registro de Asistencia. | |
| | | NA | V | NA | NA | Muestra un mensaje de confirmación. | |
| | | NA | NA | V | NA | Muestra los datos del Registro de Asistencia. SC2 "Ver Registro de Asistencia". | |
| EC3 | Selecciona la opción de cancelar. | | | | | Elimina los datos. Muestra un mensaje de información. Regresa a la vista anterior. | Sección Mis Cursos/ Seleccionar un grupo/ Pestaña Docente/ sección Administración del aprendizaje/ícono registro de asistencia /selecciona una fecha de la tabla/ botón cancelar |

Descripción de las variables

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|----|-----------------|--------------------|------------|---|
| 1 | Asistencia | Campo de selección | No | Representa el estado del estudiante en la clase, es un campo de selección y por defecto siempre aparece presente, siendo los otros estados ausente con justificación y ausente sin justificación. |
| 2 | Grupo | Campo de selección | No | Campo de selección que le permite al usuario seleccionar el grupo al cual este imparte clase, por defecto aparece el grupo por el cual entró el profesor. |
| 3 | Fecha | Campo de selección | No | Campo de selección que le permite al usuario seleccionar la fecha en la que sea pasar la asistencia, por defecto aparece la actual. |

| | | | | |
|---|-------------|----------------|----|---|
| 4 | Observación | Campo de texto | Sí | Representa una nota que desee escribir el profesor sobre el estado del estudiante, es un campo editable, no es obligatorio. |
|---|-------------|----------------|----|---|

Tabla 6. Diseño del caso de prueba Gestionar Asistencia.

4.3.4 Resultados obtenidos

Las pruebas unitarias se realizaron en la medida que se fueron desarrollando las funcionalidades para ir comprobando su correcta implementación. En estas pruebas se utilizó el método de caja blanca y las realizó el programador gracias a las ventajas de compilación que brinda el IDE Netbeans 6.9. Éstas no se planificaron, ni se registraron sus resultados ya que fueron haciéndose durante el desarrollo de la solución.

Las pruebas de integración se llevaron a cabo durante la implementación de las funcionalidades, donde para evaluar la solución desarrollada se concibieron varias iteraciones, en las cuales se probaron las funcionalidades propuestas con el resto de las acciones que se realizan en los hiperentornos, de forma íntegra. Estas pruebas validaron la integración con diferentes plugins que forman parte de la plataforma educativa Zera, por ejemplo: sfAORecomendationsPlugin, sfAOFiltersPlugin, sfiResourcePlugin y sfEmbedFilePlugin, donde se encontraron fallos en las respuestas cuando la operación depende de los servicios prestados por éstos. Además, los errores más comunes detectados en el resto de las pruebas de integración guardan relación con la comunicación a través de las interfaces, acceso incoherente a estructuras de datos globales, tiempos de respuestas, entre otros. Por tanto las pruebas de integración verificaron que las funcionalidades y demás elementos del hiperentorno operen correctamente cuando son combinados e interactúen de igual forma a través de sus interfaces, cubriendo las funcionalidades establecidas y ajustándose a los requisitos especificados.

Las principales pruebas realizadas fueron las que se muestran a continuación, pertenecientes al nivel de sistema y utilizando el método de caja negra:

- ✓ Pruebas Internas: Fueron realizadas por el equipo de calidad interna del proyecto (en este caso las analistas), con el objetivo de que las funcionalidades realizadas contaran con la menor cantidad de errores posibles.

- ✓ Pruebas Cruzadas: Fueron realizadas por el equipo de desarrollo (analistas de diferentes módulos), con el objetivo de encontrar la mayor cantidad de errores en cuanto a validaciones, pautas de diseño, formato de los campos, entre otras.
- ✓ Pruebas de Liberación: Fueron realizadas por un tercero, en este caso CALISOFT (Dirección de Calidad de Software de la UCI que garantiza el crecimiento continuo de una producción de software con calidad en la organización), con el objetivo de revisar exhaustivamente las funcionalidades y declararlas en estado óptimo para su entrega al cliente.

A continuación se presentan los resultados arrojado por las pruebas de liberación, para consultar los resultados de los demás tipos de pruebas realizadas ver Anexo 8: Resultados de las pruebas realizadas.

| Casos de pruebas | No conformidades | | | | |
|-----------------------------------|------------------|-------|------|------------|-------|
| | Alta | Media | Baja | No Procede | Total |
| Consultar Programas de Estudio | - | - | - | - | 0 |
| Mostrar Equivalencias | - | - | 1 | - | 1 |
| Listar Avances | - | - | 1 | - | 1 |
| Gestionar Avance | - | - | 3 | 1 | 4 |
| Consultar Recomendaciones de Uso | - | - | - | - | 0 |
| Mostrar Sugerencias de Evaluación | - | - | - | - | 0 |
| Mostrar Manual de Usuario Docente | - | - | - | - | 0 |
| Consultar Materiales Adicionales | - | - | - | - | 0 |
| Listar Períodos Evaluativos | - | - | - | - | 0 |
| Evaluar Evidencia | - | 2 | 1 | - | 3 |
| Gestionar Asistencia | - | 1 | 2 | - | 3 |
| Gestionar Evaluación | - | 1 | 2 | 2 | 5 |
| Consultar Materiales del Docente | - | - | 1 | - | 1 |
| Gestionar Anécdota | - | - | 4 | 1 | 5 |
| Consultar Anécdotas | - | - | 2 | - | 2 |
| Gestionar Artículo de Interés | - | - | 4 | 2 | 6 |
| Consultar Artículos de Interés | - | 1 | 2 | - | 3 |
| Gestionar Recomendación de Uso | - | - | - | - | 0 |

Tabla 7. Pruebas de liberación realizadas por CALISOF.

Las pruebas de liberación realizadas arrojaron un total de 34 no conformidades, de ellas 6 no procedieron, 5 fueron de complejidad media y 23 de complejidad baja. Los principales errores detectados fueron de internacionalización y presentación de la información, todos ellos se corrigieron en la medida que se iban encontrando. Por otro lado es importante señalar que teniendo en cuenta la cantidad de casos de uso y la cantidad de no conformidades obtenidas, los resultados de estas pruebas fueron buenos, teniendo presente también que en parte éstos se debieron a las exhaustivas pruebas cruzadas e internas realizadas con anterioridad.

4.4 Conclusiones

Esta etapa del desarrollo se caracterizó por obtener resultados visibles tanto para los que se benefician con la solución brindada como para el equipo de trabajo, ya que quedan completamente implementadas las funcionalidades que se definieron al inicio de la investigación.

Los diagramas de componentes obtenidos permitieron conformar lo que se conoce como modelo de implementación, el cual describió como los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan. Además las pruebas realizadas validaron que las funcionalidades desarrolladas satisfacen los requisitos especificados y además arrojaron que aunque se detectaron varias no conformidades, éstas ya están completamente arregladas y las funcionalidades cumplen con la calidad requerida.

CONCLUSIONES

Con la culminación de la investigación realizada se puede afirmar que se cumplieron los objetivos trazados, obteniéndose como principal resultado todas las funcionalidades propuestas, así como los artefactos y la documentación necesaria que guiaron el proceso de desarrollo de éstas; además se pudo arribar a las siguientes conclusiones:

- ✓ El estudio realizado sobre los sistemas orientados a apoyar el proceso de enseñanza – aprendizaje, sirvió de base para la posterior implementación de las funcionalidades, permitiendo con esto seleccionar los mejores elementos presentes en estos sistemas y aplicarlos a las desarrolladas.
- ✓ La solución desarrollada permitirá a los docentes prepararse, supervisar el aprendizaje de los estudiantes y realizar aportes al contenido presente en los hiperentornos y con ello facilitar sus responsabilidades como educador.
- ✓ Mediante la puesta en práctica de las pruebas aplicadas a las funcionalidades desarrolladas, se pudo detectar y corregir los fallos de implementación, logrando con esto validar el perfecto funcionamiento de la solución propuesta.

RECOMENDACIONES

Teniendo en cuenta los resultados obtenidos en esta investigación y basado en la experiencia adquirida, se recomienda:

- ✓ Seguir trabajando en el diseño de las interfaces de las funcionalidades implementadas para lograr un mayor equilibrio y presentación de la información.
- ✓ Redefinir el modelo de datos que permita registrar la evaluación de todos los contenidos (capítulos, temas y/o subtemas) que contempla una prueba.

REFERENCIAS BIBLIOGRÁFICAS

1. Claroline.net. [En línea] 2008. [Citado el: 5 de Enero de 2011.] <http://www.claroline.net/>.
2. **Moodle.** Moodle. *Moodle*. [En línea] [Citado el: 5 de Enero de 2011.] <http://moodle.org/about/>.
3. **Sadhea.** Sadhea. *Sadhea*. [En línea] 2009. [Citado el: 5 de Enero de 2011.] <http://www.sadhea.rimed.cu/>.
4. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. España : s.n., 1999.
5. Dos Ideas. [En línea] [Citado el: 6 de Enero de 2011.] http://www.dosideas.com/wiki/Extreme_Programming.
6. eleZeta . [En línea] [Citado el: 6 de Enero de 2011.] <http://elezeta.net/2004/08/27/extreme-programming-xp/>.
7. UPTODOWN. [En línea] [Citado el: 12 de 2 de 2011.] <http://argouml.uptodown.com/>.
8. Kioskea.net. *Kioskea.net*. [En línea] [Citado el: 2011 de enero de 23.] <http://es.kioskea.net/contents/cs/csintro.php3>.
9. webtaller.com. [En línea] 2007. [Citado el: 8 de Enero de 2011.] <http://www.webtaller.com/construccion/lenguajes/html/lecciones/que-es-xhtml.php>.
10. masadelante.com. *Definición de CSS - ¿Qué son las hojas de estilo o cascading style sheets?* [En línea] 2005. [Citado el: 9 de Enero de 2011.] <http://www.masadelante.com/>.
11. **Pérez, Javier Eguíluz.** *Introducción a Javascript*. 2008.
12. Programación en castellano. [En línea] 2008. [Citado el: 10 de Enero de 2011.] <http://www.programacion.net/php>.
13. maestros del web. [En línea] 2005. [Citado el: 10 de Enero de 2011.] <http://www.maestrosdelweb.com/>.
14. **Puebla, Ing. Yoan Arlet Carrascoso y Gómez, Ing. Enrique Chaviano.** GestioPolis. *Propuesta de arquitectura orientada a servicios para el módulo de inventario del ERP cubano* . [En línea] 2009. [Citado el: 11 de Enero de 2011.] <http://www.gestiopolis.com/administracion-estrategia/erp-arquitectura-orientada-a-servicios.htm>.
15. **Álvarez, Miguel Ángel.** DesarrolloWeb.com. [En línea] 2009. [Citado el: 10 de Enero de 2011.]
16. **Potencier, Fabien.** symfony.es. [En línea] 2010. [Citado el: 10 de Enero de 2011.] <http://www.symfony.es/>.

17. web.Ontuts. [En línea] [Citado el: 12 de 2 de 2011.] <http://web.ontuts.com/tutoriales/introduccion-a-object-relational-mapping-orm/>.
18. **Pecos, Daniel.** PostgreSQL vs. MySQL. [En línea] 2006. [Citado el: 11 de Enero de 2011.] http://danielpecos.com/docs/mysql_postgres/x15.html.
19. desarrolloweb.com. *Qué es Oracle.* [En línea] 2006. [Citado el: 12 de Enero de 2011.] <http://www.desarrolloweb.com/articulos/840.php>.
20. NetBeans. [En línea] [Citado el: 12 de Enero de 2011.] http://netbeans.org/community/releases/68/index_es.html.
21. buenmaster.com. [En línea] 2007. <http://www.buenmaster.com/?a=1203>.
22. tufunción. *Los mejores IDEs para Php.* [En línea] 2007. <http://www.tufuncion.com/ide-php>.
23. LIGHTTPD fly light. *LIGHTTPD fly light.* [En línea] [Citado el: 30 de enero de 2011.] <http://www.lighttpd.net/>.
24. Versión 2.0 del Servidor HTTP Apache. *Versión 2.0 del Servidor HTTP Apache.* [En línea] [Citado el: 30 de enero de 2011.] http://httpd.apache.org/docs/2.0/new_features_2_0.html.
25. **Ingenieria, Departamento de.** *Conferencia 7 de Ingeniería de Software II, Curso 2009-2010.* 2009-2010.
26. **Pressman.** *Ingeniería de Software. Un enfoque práctico.* Mc Graw Hill. 2005.

BIBLIOGRAFÍA

Caroline. *Manual Caroline.*

Collado, M. *Pruebas de software. Técnicas de prueba del software. Estrategias de prueba del software.*

Educativo, Departamento Nacional de Software. *Manual de usuario de la Colección Futuro.*

Entorno Virtual de Aprendizaje. 2009. *Conferencia 1. Disciplina de Análisis y Diseño.* Entorno Virtual de Aprendizaje.

Fabien Potencier, François Zaninotto. 2008. *Symfony la guía definitiva.* s.l. Apress : s.n., 2008. ISBN-13.

Felipe, Ana Teresa González de. *Guía de apoyo para el uso de Moodle.*

Gutmans, Andi. 2005. *PHP 5 Power Programming.* [En línea] 2005. <http://bibliodoc.uci.cu/pdf/reg04073.pdf>.

Ingeniería, Departamento de. 2009-2010. *Conferencia 7 de Ingeniería de Software II, Curso 2009-2010.* 2009-2010.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 1999. *El Proceso Unificado de Desarrollo de Software.* España: s.n., 1999. [En línea] 2000. <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

Jacobson, Ivar, Rumbaugh, James y Booch, Grady. 2007. *El lenguaje Unificado de Modelado. Manual de Referencia.* Madrid : Addison Wesley, 2007. Segunda Edición.

Lenguajes de Programación, 2009. *Lenguajes de Programación.* www.lenguajes-de-programacion.com. [Online][Cited: 22 de enero de 2011] <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.

NetBeans - The Only IDE You Need. www.netbeans.org. [Online] [Cited: 25 de enero de 2011.] <http://www.netbeans.org/features>.

Pérez, Javier Eguiluz. 2008. *Introducción a Ajax.* [En línea] 2008. www.librosweb.es.

Pressman, Roger S. 2005. *Ingeniería del Software. Un enfoque práctico.* Mc Graw-Hill, 200.

SANCHO, JESÚS BAÑOS. Octubre de 2007. *LA PLATAFORMA EDUCATIVA MOODLE. MANUAL DE CONSULTA PARA EL PROFESORADO.* Madrid : s.n., Octubre de 2007.

Visual Paradigm Organización. 2009. *Sitio Web oficial Visual-Paradigm.* Sitio Web oficial Visual-Paradigm. [En línea] 2009. [Citado el: 22 de enero de 2011.] <http://www.visual-paradigm.com/product/vpuml/>.

Web oficial Visual-Paradigm. Sitio Web oficial Visual-Paradigm. Sitio Web oficial Visual-Paradigm. [Online] [Cited: 25 de enero de 2011.] <http://www.visual-paradigm.com/product/vpuml/>.

1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México : Prentice Hall, 1999

2003. *UML y Patrones, Segunda Edición.* s.l. : PEARSON, 2003.

2008. *Introducción a CSS.* [En línea] 2008. www.librosweb.es.

2008. *Introducción a JavaScript.* [En línea] 2008. www.librosweb.es.

2008. *Introducción a XHTML.* [En línea] 2008. www.librosweb.es.