



Universidad de las Ciencias Informáticas
Facultad 4

*Desarrollo del Módulo Maestro de la colección
El Navegante en su versión multiplataforma*



**Trabajo de diploma para optar por el Título de Ingeniero en Ciencias
Informáticas**

Autores:

Yordanka Rieumont Martínez
Carlos Ernesto Díaz Rojas

Tutores:

Ing. Osmany Montes de Oca Rodríguez
Ing. Ángel Alberto Vázquez Sánchez

**La Habana, junio 2011
“Año 53 de la Revolución”**

Declaración de autoría

Declaramos ser los autores del trabajo de diploma titulado: “Desarrollo del módulo Maestro de la colección El Navegante en su versión multiplataforma” y autorizamos a la Universidad de las Ciencias Informáticas para que haga uso de la misma como lo estime pertinente, cediéndole de esta forma los derechos patrimoniales de la misma con carácter exclusivo.

Para que así conste firmamos la misma a los — días del mes de junio del año 2001.

Firma del autor

Yordanka Rieumont Mrínez

Firma del autor

Carlos Ernesto Días Rojas

Firma del tutor

Ing. Osmany Montes de Oca Rodríguez

Firma del tutor

Ing. Angel Alberto Vazquez Sanchez

"Lo importante en la vida no es solo el triunfo, sino la lucha, lo esencial no es haber vencido, sino haber luchado bien"

Pierre de Coubertin

Mis primeros agradecimientos son a nuestra Revolución por haberme dado la oportunidad de estudiar y formarme como una futura profesional eficiente en una universidad de excelencia. A mi mamá por apoyarme siempre, y en muchas ocasiones dejar de pensar en ella para pensar en mí y velar siempre por mi bienestar.

A mis hermanos por todo su cariño y apoyo en toda mi carrera, en especial a mi hermano Jorge Luis por ser mi amigo, mi padre, mi consejero, por enseñarme que a los grandes logros en la vida solo se llega con mucho sacrificio y optimismo.

A Raicel por su paciencia, su comprensión, su apoyo incondicional, y sobre todo por ser mi compañero en este largo camino a veces muy difícil que se llama vida. Te quiero mucho mi amor.

A mis abuelos por su cariño y comprensión.

A mis suegros por haberse comportados como padres conmigo y apoyarme cuando lo he necesitado.

A mis amigos que se han convertido en familia para mí, Lesdie, Lisay, Yofí, Suramis, Víctor, Pipo, Reinier, Yurisbel, Agustín, Addiel, Roberto, Emilio, en fin todos aquellos que me han apoyado y sobre todo, han confiado en mí.

Quiero agradecer además el apoyo incondicional de todos los profesores que jugaron un papel fundamental a lo largo de toda mi carrera, Isabel Lombillo, Rafael, Ismael, Yasirys, Rosalba, Yosetti, Vislet, Osdalme, Ismael, en fin, a todos aquellos que me han brindado su apoyo incondicional cuando lo he necesitado.

A mi compañero de tesis Carlos Ernesto por su preocupación y dedicación al trabajo.

A mis tutores Osmany y Vázquez, por su ayuda y apoyo en todo el desarrollo del presente trabajo de diploma.

De Yordanka

Primeramente quiero agradecer a toda mi familia, en especial a mi papá, a mi mamá y a mi hermana, por haberme apoyado tanto durante toda mi etapa educacional y por haberse sacrificado de todas las maneras posibles para que este día y este triunfo fueran posibles.

A todos los buenos amigos que he encontrado en la universidad y también a los de toda la vida, Yuniel, Leo, Pepe, Yosnay, Mario, Ancel, Alejandro, Guillermo, Joel, Eric, Sol, que me han apoyado y han estado conmigo estos cinco años.

A todos los que han compartido conmigo en el grupo o en el apartamento y han soportado mis majaderías durante la carrera.

A Ela, María Antonia, Aracelis, Jesús, Jesusito, Robertico, Yosvany, Rachel, Marbelis e Ismarys por haberme acogido desde muy pequeño y quererme como si fuera su familia.

A todos los profesores que he tenido en la carrera, en especial a Greysi, Arcel, Yovany, Rogelio, por haber ayudado a formar un profesional preparado para la vida y para el futuro laboral.

Al deporte universitario por haberme ayudado a relajarme en los momentos más tensos y ser fuente de alegrías, tristezas y sobre todo, buenas amistades.

A mi compañera de tesis Yordanka porque sin su preocupación y buen trabajo no hubiera sido posible el éxito. A los profesores del proyecto Multisaber-El Navegante, en especial a mis tutores Osmany y Vázquez por habernos apoyado tanto en el desarrollo del presente trabajo de diploma.

De Carlos Ernesto

Dedico el presente trabajo de diploma a la memoria de la persona más importante en mi vida, mi papá, él fue mi inspiración en toda mi carrera.

De Yordanka

Dedico el presente trabajo de diploma a mi mama, a mi papa y a mi hermana que han sido el motivo por el que me he sacrificado tanto para terminar satisfactoriamente esta hermosa carrera.

De Carlos Ernesto

Resumen

Atendiendo a la necesidad del Ministerio del Poder Popular para la Educación en Venezuela (MPPE) de incrementar la calidad del proceso de enseñanza–aprendizaje de sus estudiantes, en la Universidad de las Ciencias Informáticas (UCI) se desarrolla, a partir de sus especificaciones, una nueva versión multiplataforma de la colección educativa existente en Cuba: “El Navegante”. En el presente trabajo se exponen las diferentes tareas realizadas vinculadas al análisis, diseño e implementación del módulo “Maestro” perteneciente a dicha colección. Este módulo permite la gestión de toda la información relacionada con estudiantes, escuelas, grupos académicos, noticias y recomendaciones metodológicas para fomentar la preparación de los maestros. Para su desarrollo se determinó hacer uso de herramientas de software libre. Se documentaron los resultados arrojados de un estudio realizado con respecto a las aplicaciones similares existentes, luego, se describieron las herramientas y tecnologías definidas y aprobadas por el equipo del proyecto para el desarrollo de la nueva versión multiplataforma de la colección El Navegante. Se exponen además, las características del módulo a través de los artefactos propuestos por la metodología de desarrollo RUP. Con este trabajo se obtuvo una aplicación que posibilita la administración dinámica de la información referida al módulo “Maestro” de la colección El Navegante en su versión multiplataforma.

Palabras claves: multiplataforma, software educativo.

Abstract

Given the need of the Ministry of Popular Power for Education in Venezuela (MPPE) to increase the quality of the teaching-learning process in its students, the University of Informatics Sciences (UCI) is developing, based on its specifications, a new version of the existing educational multi-platform collection in Cuba: "The Navigator". In this document is described the different tasks performed related to the analysis, design and implementation of the module "Teacher" belonging to that collection. This module allows the management of all related information about students, schools, academics, news and methodological recommendations to enhance teacher preparation. For its development is determined to make use of free software tools. All the results obtained in a study done on similar existing applications is documented, described the tools and technologies defined and approved by the project team to develop the new version of the platform. It's explained the characteristics of the module through the devices proposed by development methodology RUP. With this work was obtained an application that enables dynamic management of information referred to the module "Teacher" of the platform "The Navigator".

Keywords: multi-platform, educational software

ÍNDICE DE CONTENIDOS

Introducción.....	1
Capítulo 1. Fundamentación Teórica.....	6
Introducción.....	6
1.1. Descripción del objeto de estudio	6
1.2. Análisis de aplicaciones similares existentes.....	9
1.3. Tendencias y tecnologías actuales.....	12
1.3.1. Metodologías de desarrollo	12
1.3.2. Lenguajes de desarrollo y modelado	15
1.3.3. Herramientas CASE	19
1.3.4. Arquitectura y patrones.....	19
1.3.5. Entorno de desarrollo integrado	20
1.3.6. Sistemas Gestores de Base Datos (SGBD)	20
1.3.7. Servidores Web.....	21
1.4. Propuesta de solución	22
Capítulo 2. Características del Sistema	23
Introducción.....	23
2.1. Descripción del Modelo de Dominio.....	23
2.2. Especificación de requisitos	25
2.2.1. Requisitos funcionales (RF)	25
2.2.2. Requisitos no funcionales.....	26
2.3. Modelo de Casos de Uso del Sistema	28
2.3.1. Actores del sistema.....	28
2.3.2. Diagrama de Casos de Uso del sistema.....	29
2.3.3. Descripción de los casos de uso.....	29
2.4. Conclusiones del capítulo.....	49
Capítulo 3. Análisis y Diseño	50
Introducción.....	50
3.1. Modelo de Análisis.....	50
3.1.1 Clases de análisis, estereotipos y funciones	50

3.1.2. Diagramas de Clases de Análisis (DCA).....	51
3.2. Modelo de diseño.....	53
3.2.1. Terminologías utilizadas en los diagramas de clase de diseño	53
3.2.2. Diagramas de clases del diseño (DCD)	54
3.2.3. Descripción de las Clases del Diseño.....	59
3.3. Diagrama de clases persistentes	62
3.4. Modelo de datos	63
3.5. Patrones de diseño	63
Capítulo 4. Implementación del sistema	65
Introducción.....	65
4.1. Vista de despliegue.....	65
4.1.1. Variante 1 del DD	65
4.1.2. Variante 2 del DD	66
4.2. Vista de implementación.....	66
4.2.1. Diagrama de componentes	66
4.2.2. Componentes del subsistema controlador.....	68
4.2.3. Componentes del subsistema modelo	69
4.2.4. Componentes del subsistema vista.....	69
4.2.5. Componentes del paquete JavaScript	70
4.3. Interfaz de la aplicación	70
4.4. Conclusiones del capítulo.....	76
Conclusiones Generales.....	77
Recomendaciones	78
Referencias Bibliográficas	79

ÍNDICE DE FIGURAS

Figura 2.1. Modelo de Dominio	23
Figura 3.1. DCA_CU Gestionar Escuela	51
Figura 3.3. DCA_CU Gestionar Grupo Académico	52
Figura 3.4. DCA_CU Gestionar Maestro	53
Figura 3.5. DCD_CU Gestionar Escuela	55
Figura 3.6. DCD_CU Gestionar Estudiantes	56
Figura 3.7. DCD_CU Gestionar Grupo Académico	57
Figura 3.8. DCD_CU Gestionar Maestros	58
Figura 3.10. Modelo de datos	63
Figura 4.1. Diagrama de despliegue 1	65
Figura 4.2. Diagrama de despliegue 2	66
Figura 4.3. Diagrama de componentes del módulo Maestro	67
Figura 4.6. Diagrama de componentes del subsistema “vista”	69
Figura 4.7. Diagrama de componentes del paquete JavaScript	70
Figura 4.8. Interfaz para listar los datos de las escuelas	71
Figura 4.9. Interfaz para adicionar una escuela	71
Figura 4.10. Interfaz para modificar los datos de una escuela	72
Figura 4.11. Interfaz para eliminar los datos de una escuela	72
Figura 4.12. Interfaz para listar los datos de los estudiantes	73
Figura 4.13. Interfaz para adicionar un estudiante	73
Figura 4.16. Interfaz para listar los datos de los maestros	75
Figura 4.18. Interfaz para modificar los datos de un maestro	76
Figura 4.19. Interfaz para eliminar los datos de un maestro	76

ÍNDICE DE TABLAS

Tabla 1.1 Funciones que pueden realizar los Materiales Educativos Multimedia (4)	6
Tabla 1.2 Ventajas e inconvenientes de los Materiales Educativos Multimedia (4)	8
Tabla 2.1. Actores del sistema y su descripción.....	28
Tabla 2.2. Descripción del caso de uso: Gestionar Escuela.....	29
Tabla 2.3. Descripción del caso de uso: Gestionar Estudiantes.....	34
Tabla 2.4. Descripción del caso de uso: Gestionar Grupo Académico.....	39
Tabla 2.5. Descripción del caso de uso: Gestionar Maestros	43
Tabla 3.1. Clases de análisis y su descripción.....	50
Tabla 3.3. Descripción de las clases del diseño del caso de uso: Gestionar Escuela.....	59
Tabla 3.4. Descripción de las clases del diseño del caso de uso: Gestionar Estudiantes	59
Tabla 3.6. Descripción de las clases de diseño del caso de uso: Gestionar Maestros	61

Introducción

En la actualidad, el avance en el desarrollo tecnológico acompañado por el impacto ocasionado en la humanidad el uso de las nuevas Tecnologías de la Información y las Comunicaciones (TIC), ha jugado un papel de suma importancia en el desarrollo económico, cultural e intelectual de muchos países.

Cuba, a pesar de ser un país subdesarrollado no queda exenta de esta revolución tecnológica. La aplicación de la informática en diferentes sectores como la educación, ha proporcionado notables mejoras en el sistema educacional. Dentro de ellas se encuentran tres colecciones de software educativo desarrolladas por la dirección del Ministerio de Educación (MINED), y dirigidas a los diferentes niveles de enseñanza; estas son:

- La colección Multisaber, dirigida a estudiantes del nivel primario.
- La colección El navegante, dirigida a estudiantes del nivel secundario.
- La colección Futuro, dirigida a estudiantes del nivel pre-universitario.

Estas colecciones educativas, se componen de varios software o módulos, los cuales se sustentan en la concepción de hiperentornos de aprendizaje, definidos como: “sistema informático basado en tecnología hipermedia que contiene una mezcla o elementos representativos de diversas tipologías de software educativo”. (1)

Nuestro país, se ha convertido en un patrón a seguir para Latinoamérica por su hermandad y colaboración con otros países. Un ejemplo de ello se evidencia en la República Bolivariana de Venezuela, a donde se han extendido la asistencia de diversos programas y técnicas en el proceso docente. En diciembre del año 2009, como resultado de la X Mixta Convención Cuba–Venezuela, surge el proyecto: “Colecciones de Software Educativo Multisaber y El Navegante para planteles de los niveles primaria y secundaria”. Dicho proyecto, perteneciente a la Universidad de las Ciencias Informáticas, se caracteriza por el desarrollo de nuevas versiones de las colecciones educativas ya existentes en Cuba, pero ahora en una versión multiplataforma y haciendo uso de herramientas de software libres para su desarrollo.

Dentro de las nuevas colecciones a desarrollar se encuentra “El Navegante”. En esta nueva versión, actualmente en fase de desarrollo, subsisten deficiencias. En ella no se tiene un control detallado de toda la información referente a los estudiantes y profesores, así como, las escuelas, los grupos académicos, las noticias. Además, no se realizan diferentes actividades deseadas por el “Ministerio del Poder Popular para

la Educación en Venezuela”, tales como: cambiar la contraseña a los estudiantes, eliminar, exportar e importar las trazas de los estudiantes en el software, consultar la trayectoria de los maestros en la aplicación, importar un listado de estudiantes, modificar las categorías de revisión, adicionar, eliminar y modificar los datos de estudiantes, maestros, grupos académicos, escuelas, así como, las noticias que le posibilitan al maestro estar actualizado de los acontecimientos sucedidos tanto en el ámbito nacional, como internacional. Todas estas actividades responden al módulo “Maestro” de la Colección El Navegante en su versión multiplataforma. Las mismas están enfocadas en lograr un proceso de enseñanza–aprendizaje exitoso en los estudiantes venezolanos. A raíz de esta situación se identifica el siguiente:

Problema de investigación:

¿Cómo garantizar el funcionamiento y la gestión de la información referida al módulo “Maestro” de la colección El Navegante en su versión multiplataforma?

Objeto de estudio:

Software educativo para la gestión de información y superación de maestros.

Objetivo general de la investigación:

Lograr un proceso eficiente de gestión de la información relacionada al módulo “Maestro”, así como, su correcto funcionamiento en la colección El Navegante en su versión multiplataforma.

Campo de acción:

Módulo “Maestro” de la colección El Navegante en su versión multiplataforma.

Idea a defender:

Con la implementación del módulo “Maestro” se garantiza un proceso eficiente de gestión de información referida al mismo, así como, su correcto funcionamiento en la Colección El Navegante.

Objetivos específicos de la investigación:

- Realizar un estudio crítico valorativo del estado del arte.
- Realizar el análisis y diseño del módulo “Maestro” de la colección El Navegante en su versión

multiplataforma.

- Implementar e integrar el módulo “Maestro” a la colección El Navegante en su versión multiplataforma.

Para dar cumplimiento a los objetivos planteados, se proponen las siguientes **Tareas de investigación:**

1. Investigación acerca de otras aplicaciones existentes similares a la solución propuesta.
2. Realización del diseño teórico-metodológico de la investigación.
3. Realización del análisis y diseño del módulo “Maestro”.
4. Definición de la estructura de la Base de Datos.
5. Implementación de la administración de la información de los estudiantes y profesores en el módulo “Maestro”.
6. Implementación del proceso de exportación e importación de las trazas generadas de las actividades realizadas por los estudiantes en el software.
7. Implementación de la capa de presentación del módulo “Maestro”.
8. Integración del módulo “Maestro” a la colección El Navegante.
9. Documentación de toda la información generada en el proceso de investigación.

Resultados esperados

Con el desarrollo de este trabajo de diploma, se podrá contar con:

- Un módulo a través del cual, los maestros puedan elevar su preparación y administrar información sobre sus estudiantes, grupos académicos, noticias, escuelas y demás maestros en la colección El Navegante en su versión multiplataforma, en aras de lograr un proceso de enseñanza–aprendizaje con mayor calidad.
- Se logrará un mayor control por parte de los maestros del avance de los estudiantes en su propio proceso de aprendizaje.
- Se contará además, con la documentación detallada de todo el proceso de desarrollo del módulo “Maestro”, de modo que sirva de apoyo a futuras versiones de la colección.

Métodos científicos:

Dentro de los métodos científicos a utilizar, se encuentran:

Teóricos:

- Análisis histórico lógico: se aplicará este método para realizar un estudio del estado del arte de las versiones existentes, analizar la trayectoria de la colección El Navegante en su nueva versión multiplataforma, así como los módulos que la componen, principalmente el módulo “Maestro” en todo su desarrollo.
- Modelación: una vez analizada toda la información resultante de la investigación previamente realizada, este método resulta de gran utilidad para realizar los modelos correspondientes al ciclo de vida del software, proporcionando facilidades a la hora de dar cumplimiento a las tareas de análisis y diseño de todos los procesos involucrados en la aplicación, así como, la implementación del sistema.

Empíricos:

- Observación: mediante el uso del registro visual, se determinará como debe comportarse el objeto de investigación en un ámbito real, es decir, es una forma de obtener información inmediata o directa del objeto que se investiga, en este caso, el módulo “Maestro” de la colección El Navegante en su versión multiplataforma.

Estructura del documento:

- **Capítulo 1: Fundamentación teórica**: en este capítulo se hace referencia a todos los elementos teóricos utilizados en la investigación, se describen las metodologías, tecnologías, lenguajes de programación y herramientas utilizadas en el desarrollo de la solución propuesta, así como, los principales conceptos para una mejor comprensión del problema a resolver.
- **Capítulo 2: Características del sistema**: en este capítulo se describe el modelo de dominio donde se capturan los principales eventos que suceden donde se encuentra el sistema. Se detallan además, los requisitos funcionales y no funcionales del software, así como el diagrama de casos de uso del sistema y sus respectivas descripciones.

- **Capítulo 3: Análisis y diseño del sistema:** en este capítulo se analiza, describe y diseña la solución propuesta a través de los diferentes modelos y diagramas presentes en el flujo de trabajo análisis y diseño.
- **Capítulo 4: Implementación del sistema:** en este capítulo, a través de diferentes diagramas, se expone como está implementado el sistema en términos de componentes.

Capítulo 1. Fundamentación Teórica

Introducción

En el presente capítulo, se realizará una investigación de las diferentes aplicaciones existentes similares a la solución propuesta, se describe toda la tecnología necesaria para el cumplimiento exitoso de los objetivos planteados, donde se abordarán diferentes aspectos, como: características, ventajas y desventajas de los mismos.

1.1. Descripción del objeto de estudio

La llegada de la informática a los centros escolares ya es un hecho. El uso de las TIC en la educación se reproduce a pasos agigantados, gracias, en gran medida, a las disímiles facilidades que las mismas le proporcionan al proceso de enseñanza–aprendizaje. Esto provoca que sea cada vez mayor el proceso de producción y desarrollo de software educativos. Dentro de los más populares se encuentra la:

Multimedia: uso de múltiples modalidades sensoriales combinadas para conseguir un objetivo concreto (3). Este término es utilizado además, para referirse a cualquier objeto o sistema que utilice múltiples medios de expresión, ya sean físicos o digitales, para presentar o comunicar cierta información. Estos medios pueden ser variados, desde texto e imágenes, hasta animación, gráficos, sonido, video. Dentro de los diversos tipos de multimedia se encuentra el siguiente:

Sistemas multimedia educativos: por estos se entiende, a la utilización de aplicaciones multimedia como herramientas para la formación. Estos programas se componen de un conocimiento que se trata de transmitir en las mejores condiciones posibles, de tal forma que los alumnos lo adquieran con el mínimo esfuerzo mental. Las tecnologías multimedia en la formación no son más que un caso particular de la informática educativa, ya que incorporan diversos tipos de información (video, audio, bases de datos, hipertexto, documentos, imágenes, etc.). (3)

Tabla 1.1 Funciones que pueden realizar los Materiales Educativos Multimedia (4)

Función	Características	Programas
Informativa	La mayoría de estos materiales, a través de sus actividades,	Base de datos.

	presentan contenidos que proporcionan información, estructuradora de la realidad a los estudiantes.	Tutoriales. Simuladores.
Instructiva Entendedora	Orientan y regulan el aprendizaje de los estudiantes ya que, explícita o implícitamente, promueven determinadas actuaciones de los mismos encaminadas a este fin. Además, mediante sus códigos simbólicos, estructuración de la información e interactividad, condicionan los procesos de aprendizaje.	Tutoriales.
Motivadora	La interacción con el ordenador suele resultar por sí misma motivadora. Algunos programas incluyen además elementos para captar la atención de los alumnos, mantener su interés hacia los aspectos más importantes.	Todos en general.
Evaluadora	La posibilidad de “feed back” (retroalimentación) inmediato a las respuestas y acciones de los alumnos, hace adecuados a los programas para evaluarlos. Esta evaluación puede ser: <ul style="list-style-type: none"> • Implícita; el estudiante detecta sus errores, se evalúa a partir de las respuestas que le da el ordenador. • Explícita; el programa presenta informes valorando la actuación del alumno. 	Tutoriales con módulos de evaluación.
Explorar Experimentar	Algunos programas ofrecen a los estudiantes entornos donde explorar, experimentar, investigar, buscar determinadas informaciones, cambiar los valores de las variables del sistema, etc.	Base de datos. Simuladores. Constructores.
Expresiva Comunicativa	Los estudiantes se expresan y se comunican con el ordenador y con otros compañeros a través de las actividades de los programas.	Editores de texto Constructores. Editores gráficos.
Metalingüística	Al usar los recursos multimedia, los estudiantes también aprenden los lenguajes propios de la informática.	Todos.

Ventajas e inconvenientes de los Materiales Multimedia Educativos

Dentro de las ventajas e inconvenientes potenciales que poseen los Materiales Multimedia Educativos se encuentran:

Tabla 1.2 Ventajas e inconvenientes de los Materiales Educativos Multimedia (4)

Ventajas	Inconvenientes
<u>Interés, motivación:</u> la motivación (el querer) es uno de los motores del aprendizaje, ya que incita a la actividad y al pensamiento.	<u>Adicción:</u> el multimedia interactivo resulta motivador, pero un exceso de motivación puede provocar adicción. <u>Distracción:</u> los alumnos en ocasiones se dedican a jugar en vez de trabajar.
<u>Interacción. Continua actividad intelectual:</u> los estudiantes están permanentemente activos al interactuar con el ordenador y mantienen un alto grado de implicación en el trabajo.	<u>Ansiedad:</u> la continua interacción ante el ordenador puede provocar ansiedad en los estudiantes.
Los alumnos aprenden con menos tiempo.	Aprendizajes incompletos y superficiales.
Aprendizaje a partir de los errores.	<u>Desarrollo de estrategias de mínimo esfuerzo:</u> los estudiantes pueden centrarse en la tarea que les plante el programa en un sentido demasiado estrecho y buscar estrategias para cumplir con el mínimo esfuerzo mental ignorando las posibilidades de estudio que les ofrece el programa.
<u>Individualización:</u> estos materiales individualizan el trabajo de los estudiantes ya que el ordenador puede adaptarse a sus conocimientos previos y a su ritmo de trabajo.	<u>Aislamiento:</u> los materiales didácticos multimedia permiten al alumno aprender solo, hasta le animan a hacerlo, pero este trabajo individual, en exceso, puede acarrear problemas de sociabilidad.

Los recursos multimedia, proporcionan diversas facilidades que permiten la interoperabilidad, esto posibilita captar toda la atención del estudiante y motivarlo a participar con un mayor interés y libertad en su propio proceso de aprendizaje; siendo esta la mayor ventaja que presentan los recursos multimedia.

1.2. Análisis de aplicaciones similares existentes

En la actualidad, existen diversos sistemas que permiten la gestión de información referida al personal de cierta institución. Dentro de éstas se encuentran:

1. MyClass

MyClass es un avanzado Cuaderno Electrónico, utilizado en las escuelas españolas, que se adapta a cualquier Sistema Educativo y metodología utilizada para impartir una clase. Mediante dicho cuaderno, se puede personalizar y configurar los datos y la información requerida en función de las peculiaridades de las asignaturas impartidas. (5)

Mediante el uso de MyClass, cada inicio de un curso escolar resulta más cómodo, automático y gestionable, ya que, se podrán importar de los e-cuadernos, uno por cada curso, datos tales como sistema de calificación, tipos de incidencias, notas, observaciones, evaluaciones, etc. (5)

Principales beneficios

- Es una herramienta interactiva, fiable, y sencilla compatible con cualquier sistema de trabajo.
- Posee una nueva vía de conexión a las nuevas formas de gestión de datos, incrementando la capacidad profesional del maestro/a y profesor/a.
- Personalizable: presenta amplias posibilidades de configuración y personalización, con plena libertad de movimiento y total eficiencia. (5)

MyClass a pesar de ser una herramienta muy potente, no da respuesta al problema identificado en la nueva versión de la colección El Navegante. La misma se centra solo en tareas necesarias para facilitar el trabajo a realizar de los maestros y profesores en un curso escolar, dígase seguimiento del avance del aprendizaje de los estudiantes a través sus calificaciones, temáticas a impartir en las asignaturas; permite además incrementar la preparación personal de los profesores. La colección El Navegante requiere una aplicación que permita realizar estas actividades, así como, todas las acciones expuestas anteriormente y que esté dirigida a estudiantes de nivel secundario.

2. Herramienta informática para la superación profesional en Farmacología de los profesores de medicina en Cuba.

Esta herramienta constituye un soporte durante la ejecución de las actividades de superación profesional

del profesorado de la carrera de medicina para enfrentar el perfeccionamiento del proceso de enseñanza–aprendizaje de la Farmacología durante la formación del Médico General. Esta aplicación está dirigida por el profesor pero obliga la participación activa de los estudiantes para resolver las tareas indicadas. En esencia, este sistema agrupa sus principales funcionalidades en tres aspectos fundamentales, los cuales se centran en la autopreparación de los estudiantes; la gestión de materiales docentes por parte de los profesores y control administrativo y gestión de datos y privilegios de los profesores que integran el sistema.

Un profesor será el encargado del control administrativo y de la gestión de los datos y privilegios de los demás profesores que integran el sistema. Dicha persona podrá interactuar (insertar, editar o eliminar) con la información de los diferentes profesores, además de gestionar (definir, editar o modificar) los privilegios de estos. Los profesores a su vez pueden realizar la evaluación de las tareas emitidas por el estudiante y posteriormente publicar los resultados de dicha evaluación (6)

El sistema antes expuesto, fue desarrollado mediante el uso de algunas herramientas de software propietario, como, Adobe Photoshop. Dicho sistema no permite realizar todas las funcionalidades requeridas por el cliente (MPPE), dentro de estas se pueden mencionar, cambiarles la contraseña a los estudiantes, exportar, eliminar e importar las trazas de los estudiantes en el software, etc. Por lo antes expuesto, esta aplicación no da respuestas al problema identificado en la nueva versión de la colección El Navegante.

3. Agtica Edusoft (Plataforma para la administración y control de instituciones educativas)

La empresa mexicana Agtica desarrolló una plataforma de software para administrar y gestionar la información administrativa, docente, académica y de recursos humanos en cualquier tipo de centro educativo, sin importar su nivel o complejidad. Este sistema se concentra y muestra la información de cada empleado, profesor o alumno. En este último caso, puede identificar su desempeño escolar con el fin de que el docente no corra el riesgo de, por ejemplo, bajar en sus calificaciones. (7)

Principales características

- Gestión total en los procesos administrativos y docentes, optimizando los tiempos y brinda información oportuna para la toma de decisiones.

- Es funcional para cualquier grado académico, desde nivel preescolar hasta posgrados, pudiéndose aplicar a maestrías y cursos especiales. (8)

La aplicación antes expuesta, no da respuesta al problema identificado en la nueva versión de la colección El Navegante. La misma realiza funciones que no son necesarias en dicha colección, y no realiza otras que sí lo son, por ejemplo, consultar las trayectorias de los maestros en el software, eliminar, exportar, e importar las trazas de los estudiantes en el software etc. Además, es una aplicación dirigida a estudiantes de cualquier nivel educacional, sin embargo, el sistema requerido en la colección El Navegante, debe recoger contenidos adecuados al nivel de enseñanza secundaria básica.

4. Colección Multisaber

Fue desarrollada con el objetivo de fomentar el proceso de enseñanza–aprendizaje en las escuelas cubanas, en especial, en el nivel de enseñanza primaria. La misma contempla una interfaz estandarizada, que proporciona un ambiente de trabajo intuitivo, ameno y dinámico, con un alto nivel de interactividad para acceder a la información existente en la aplicación. (9)

5. Colección El Navegante

El Navegante, al igual que las colecciones anteriores, está basada en tecnología multimedia y tiene como objetivo profundizar el avance en el proceso de enseñanza-aprendizaje en los estudiantes de secundaria básica. Dentro de los productos que integran esta colección, se encuentra, el módulo Maestro el cual permite llevar un control sobre el avance en el desarrollo educativo de los estudiantes. (9)

6. Colección Futuro

Este, es un producto con tecnología multimedia el cual se enfoca en fomentar el proceso de enseñanza-aprendizaje en los estudiantes de pre-universitario. Está formado por varios software educativos, los cuales abarcan contenidos relacionados con las diferentes asignaturas impartidas en este nivel (10).

Las colecciones educativas analizadas anteriormente, Multisaber, El Navegante y Futuro presentan varias limitantes, una de ellas es que fueron creadas por diversos equipos de desarrollo haciendo uso de herramientas de software propietario, tales como: Macromedia Flash , Macromedia Director, Toolbook Instructor y Borland Delphi. En el caso de la colección Multisaber, como parte del proyecto surgido en la X

Mixta Convención anteriormente mencionado, en la UCI se está desarrollando una nueva versión multiplataforma y con un nuevo nombre: “La caja mágica”. Esta colección también consta de un módulo Maestro pero aun así no es de gran utilidad para la colección El Navegante; ya que, está desarrollado con tecnologías que no son muy compatibles con la arquitectura de la colección El Navegante, tales como: un framework de Javascript diferente, un CMS para la gestión de contenidos con una versión un poco antigua, utiliza MySQL como Sistema Gestor de Base de Datos, etc. Dentro de los diferentes módulos que componen las restantes colecciones, se encuentra el módulo Maestro, el cual permite realizar diferentes actividades, tales como, adicionar, eliminar y modificar los datos de los maestros y estudiantes que interactúan con la aplicación, consultar los ejercicios resueltos por los estudiantes, adicionar, eliminar y modificar los datos de las noticias, entre otras. A pesar de estas funcionalidades, dicho módulo no brinda la posibilidad de realizar otras actividades requeridas por el cliente, tales como: cambiar la contraseña de los estudiantes, eliminar, importar y exportar las trazas de los estudiantes en el software, así como consultar la trayectoria de los maestros en el mismo, adicionar, eliminar, modificar los datos de los grupos académicos, las escuelas en las cuales se implantará el software. Por lo antes expuesto, dicho módulo no cumple con los requisitos especificados por el cliente, ya que, el mismo requiere una aplicación web multiplataforma de código abierto y desarrollada mediante el uso de herramientas de software libre.

1.3. Tendencias y tecnologías actuales

Según estudios, sobre los lenguajes de programación, metodologías de desarrollo, así como la arquitectura y en general toda la tecnología y sus respectivos conceptos a utilizar en el desarrollo de la colección El Navegante en su versión multiplataforma realizado por el equipo de desarrollo del proyecto, se determinó utilizar los siguientes:

1.3.1. Metodologías de desarrollo

Una metodología de desarrollo de software, se puede definir como un conjunto de pasos y procedimientos que deben seguirse para desarrollar un software con una organización y calidad sólida.

Una metodología está compuesta por varias interrogantes, tales como:

- ¿Cómo dividir un proyecto en etapas?
- ¿Qué tareas se llevan a cabo en cada etapa?
- ¿Qué restricciones deben aplicarse?

- ¿Qué técnicas y herramientas se utilizan?
- ¿Cómo se gestiona y controla un proyecto?

Para el desarrollo del módulo propuesto, se hace uso de la siguiente metodología de desarrollo:

➤ **Proceso de desarrollo unificado (RUP)**

Un proceso de desarrollado de software, organiza de forma eficiente todo el trabajo a realizar por parte del equipo de desarrollo, es decir, define quién hace, qué, cómo y cuándo. RUP, define cuatro elementos: trabajadores (roles) que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los artefactos (productos), que responden a la pregunta ¿Qué?, y por último los flujos de trabajo de las disciplinas que responden a la pregunta ¿Cuándo? Estos se describen a continuación: (11)

- **Trabajadores (“quién”)**: realizan las actividades y son propietarios de los elementos. Además, definen el comportamiento y las responsabilidades (rol) de un individuo, grupo de estos, sistema automatizado o máquina que trabajan en conjunto como equipo.
- **Actividades (“cómo”)**: tarea o grupo de tareas que tienen un propósito claro; estas son realizadas por un trabajador y manipulan los elementos.
- **Artefactos (“qué”)**: productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser: modelos, elementos dentro del modelo, código fuente y ejecutables.
- **Flujo de actividades (“cuándo”)**: secuencia de actividades realizadas por trabajadores y que producen un resultado de valor observable.(11)

Fases de RUP

- **Conceptualización (concepción o inicio)**: se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración**: se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.
- **Construcción**: se obtiene un producto listo para su utilización, el cual está documentado y tiene un manual de usuario. Se obtiene uno o varios release (iteraciones) del producto que han pasado las pruebas; estos release son puestos a consideración de un subconjunto de usuarios.
- **Transición**: en esta fase, el producto ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Cada una de estas fases, se realizan mediante el ciclo de iteraciones, el cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evolución de las iteraciones precedentes. (11)

Flujos de trabajo de RUP

- **Modelado del negocio:** describe los procesos de negocio, identificando quienes participan y las actividades que requieren automatización.
- **Requerimientos:** define qué es lo que el sistema debe hacer, para lo cual identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** busca los defectos a lo largo del ciclo de vida del software.
- **Instalación:** produce releas (iteración) del producto y realiza actividades tales como: empaque, instalación, asistencia a usuarios, entre otras, para entregar el software a los usuarios finales.

Flujos de trabajo de soporte

- **Administración del proyecto:** involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización, control de versiones, entre otras.
- **Ambiente:** contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como, el procedimiento para implementar el proceso en una organización. (11) [Ver Anexo 1](#)

Características de RUP

- **Dirigido por casos de uso (CU):** en RUP, los CU no son sólo una herramienta para especificar los requisitos del sistema, también guían su diseño, implementación y prueba.
- **Centrado en la arquitectura:** RUP presta especial atención al establecimiento temprano de una

buena arquitectura que se vea fuertemente implicada ante cambios posteriores durante la construcción y el mantenimiento.

- **Iterativo e Incremental:** con el objetivo de lograr un equilibrio correcto entre los CU y la arquitectura, RUP propone un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Este proceso iterativo e incremental, consta de una secuencia de iteraciones, cada una abordando una parte de la funcionalidad total. Pasando por todos los flujos de trabajo relevantes y refinando la arquitectura.

RUP presenta además, diferentes características, tales como:

- Utiliza UML como lenguaje de representación visual.
- Orientado a Objeto
- Preparado para desarrollar grandes y complejos proyectos.
- Unifica los mejores elementos de metodologías anteriores.

Una particularidad de esta metodología es que, en cada iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. (11)

1.3.2. Lenguajes de desarrollo y modelado

Lenguajes de programación

Son herramientas que nos permiten crear programas y software; estos son de gran importancia, ya que, facilitan la tarea de programación al disponer de formas adecuadas que permiten ser leídas y escritas por personas, a su vez, resultan independientes del modo de computador a utilizar. Los lenguajes de programación de una computadora en particular, se conocen como código de máquinas o lenguaje de máquinas.

En el desarrollo del módulo propuesto, se hace uso de los siguientes lenguajes de programación:

Lenguajes del lado del cliente

➤ Hojas de estilo en cascada (CSS, Cascading Style Sheets)

Las hojas de estilo se desarrollan para compensar los defectos de HTML (acrónimo de Hyper-Text Markup Language) con respecto a la presentación y al diseño de las páginas. El principio de las hojas de estilo consiste en hacer uso de un solo documento para almacenar las características de presentación de las

páginas asociadas a grupos de elementos.

Se denomina hojas de estilo en cascada, porque se pueden definir múltiples hojas y los estilos pueden aplicarse a todas las páginas.

CSS presenta tres modos para aplicar reglas de estilos a las páginas web. El primero de ellos consiste en aplicar los estilos en una hoja externa, o sea, el código de los estilos está almacenado en otro archivo. Este código se hace activo en el documento HTML cuando se referencia el CSS usando la etiqueta <link>. Otro modo está basado en incluir el código HTML en la cabecera de la página mediante el uso de la etiqueta <style>. Por último, se puede incluir código CSS también en las propias etiquetas HTML. Incluir los estilos en una hoja externa ofrece facilidad de uso del código cuando se está en presencia de una gran cantidad de documentos HTML y se desea brindar el mismo estilo a todas, por lo que mundialmente es el modo más utilizado para lograr mantener la homogeneidad del código. Por otra parte incluir el código CSS en la cabecera de la página brinda la posibilidad de aplicarle un estilo específico a la misma, principalmente si no se trata de un código extenso. (12)

➤ **JavaScript**

JavaScript, es un lenguaje de programación multiplataforma basado en objetos e interpretado (no se necesita compilar los programas para ejecutarlos). Está implementado como parte del navegador web, el cual permite establecer mejoras en la interfaz de usuario y crear páginas web dinámicas mediante la manipulación sencilla de eventos que brinda (orientado a eventos).

➤ **HTML (Lenguaje de marcado de texto)**

HTML es el lenguaje estándar que se usa para describir la estructura y el contenido de una página web. La sintaxis del mismo es muy sencilla: el código HTML se escribe en forma de etiquetas rodeadas por corchetes regulares; la mayoría de estas etiquetas son semánticas y su interpretación es realizada por el navegador web. HTML puede incluir uno o varios scripts que permiten agregar efectos y darle un comportamiento dinámico a la página, como por ejemplo JavaScript, JQuery, entre otros. HTML también permite incorporar estilos CSS que le dan belleza al documento.

Lenguajes del lado del servidor

➤ **PHP**

PHP (acrónimo de PHP: Hypertext Preprocessor), es un lenguaje de programación interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Una de sus características más potentes, es su soporte a gran cantidad de bases de datos, entre estos pueden mencionarse MySQL, Oracle, PostgreSQL, entre otras. PHP también ofrece la integración con varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa, desde generar documentos en formato pdf, hasta analizar código XML. Como producto del código abierto, PHP goza de la ayuda de un grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen fácilmente. (13)

Lenguaje unificado de modelado

➤ **UML**

El lenguaje unificado de modelado (UML por sus siglas en inglés, Unified Model Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de sistemas que involucran gran cantidad de software. Hoy en día, UML es consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante el mismo, es posible establecer una serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. UML, no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso, es un lenguaje que permite modelar los sistemas con tecnología orientada a objetos. Son varios los beneficios que se obtienen con el uso de UML, por un lado el uso de lenguajes visuales facilitan su asimilación y entendimiento por parte del equipo de desarrollo; el tiempo invertido en el desarrollo de la arquitectura se minimiza; la detección y resolución de errores se agiliza siempre y cuando se haga uso de herramientas adecuadas de diagnóstico y depuración; además, la trazabilidad y documentación del proyecto se realiza de una forma ordenada y guiada por casos de uso. (14)

Framework de desarrollo para el lado del servidor

➤ **Symfony**

Symfony, es uno de los framework de PHP más populares entre los usuarios y las empresas, ya que, permite que los programadores sean mucho más productivos a la vez que desarrollan código de mayor calidad y más fácil de mantener. Symfony es maduro, estable, profesional y está muy bien documentado.

El framework Symfony, es utilizado para construir aplicaciones web con PHP, en otras palabras, Symfony, es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de aplicaciones web. (15) Numerosas son las razones para decidir utilizar Symfony en el desarrollo de una aplicación web debido a las características que presenta, las cuales se detallan a continuación:

- Emplea el tradicional patrón MVC (modelo – vista – controlador) para separar las distintas partes que forman una aplicación Web.
- Su licencia es de tipo software libre.
- Probado: ha sido probado con éxito durante años en varias aplicaciones gigantescas, (Yahoo!, Answer, etc.) así como, en sitios medianos y pequeños.
- Documentado: es el framework mejor documentado, ya que ha publicado 5 libros gratuitos de calidad y siempre actualizados. Además, toda la documentación está traducida al español
- Código: desde su primera versión Symfony ha sido creado para PHP 5, para obtener el máximo rendimiento de PHP y aprovechar todas sus características.
- Calidad: su código fuente incluye más de 9000 pruebas unitarias.(16)

Framework de desarrollo para el lado del cliente

➤ **JQuery**

JQuery es un framework de JavaScript que cuenta con una amplia comunidad en Internet, por lo que cuenta con una gran cantidad de bibliografía y usuarios que la usan y aportan a su desarrollo debido a que es una tecnología Open Source. Es fácil de aprender y de utilizar, es compatible con una gran cantidad de navegadores WEB, como Internet Explorer a partir de la versión 6, Opera a partir de su versión 9, Safari a partir de su versión 2.0.2 y Firefox a partir de su versión 2.0. JQuery también permite acceder de manera fácil a las distintas partes de un documento HTML mediante las funciones que posee para manipular el DOM, por lo que facilita alterar el contenido de una página de manera fácil y eficiente, permite además, gestionar los eventos que tienen lugar en una página de una manera más sencilla, así como, añadir animaciones a las páginas y simplifica el trabajo con AJAX y con CSS ya que posee funciones muy potentes para trabajar con estas tecnologías.

1.3.3. Herramientas CASE

➤ Visual Paradigm

Visual Paradigm para UML, es una herramienta UML profesional que soporta el ciclo de vida completo de desarrollo de un software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue.

La herramienta de modelado UML, resulta de suma importancia para lograr un proceso de construcción más rápido, con mayor calidad y a un menor costo. Permite además, dibujar todos los tipos de diagramas de clases, código inverso, generar códigos de diagramas y generar documentación, siendo estos artefactos muy importantes para un desarrollo de software exitoso. (17)

Características de Visual Paradigm

Dentro de las características más comunes de Visual Paradigm, se encuentran:

- Soporte de UML versión 2,1
- Permite realizar diagramas de proceso de negocio – proceso, decisión, actor del negocio, documentos.
- Permite el modelado colaborativo con CVS y Subversión
- Ingeniería de ida y vuelta.(17)

1.3.4. Arquitectura y patrones

Los patrones de diseño, son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

La arquitectura de software, es el diseño de más alto nivel de la estructura de un sistema. La misma consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software; establece los fundamentos para que analistas, diseñadores, programadores, etc. trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades existentes. (17)

La arquitectura utilizada en el desarrollo de la solución propuesta es:

➤ Modelo Vista Controlador

La arquitectura Modelo Vista Controlador (MVC), está formada por tres niveles:

- **Modelo:** representa la información con la que trabaja la aplicación, es decir, su lógica de

negocio.

- **Vista:** transforma el modelo en una página web que permite al usuario interactuar con ella.
- **Controlador:** se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC, separa la lógica de negocio (el modelo) y la presentación (la vista), por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado para la aplicación. (18)

1.3.5. Entorno de desarrollo integrado

➤ NetBeans

NetBeans se define como una herramienta de código abierto para escribir, compilar, depurar y ejecutar programas; está escrito en Java, pero es compatible con cualquier otro lenguaje de programación. Otro aspecto a destacar, es que es un producto libre y gratuito sin restricciones de uso. Además, NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente. (16)

NetBeans presenta soporte completo para Symfony y PHP (sin necesidad de instalar ningún plugin), ya que ejecuta todas las tareas del framework, lo cual permite crear aplicaciones de forma más sencilla, productiva y eficiente. También brinda soporte para HTML, CSS, JavaScript y JQuery, ya que permite gozar al programador del completamiento de código.

1.3.6. Sistemas Gestores de Base Datos (SGBD)

Un SGBD, es una aplicación que permite a los usuarios definir, crear y mantener la base de datos. Proporciona además, un acceso controlado a la misma. El mismo debe proporcionar los siguientes servicios:

- Creación y definición de la base de datos.
- Manipulación de los datos.
- Acceso controlado a los datos mediante mecanismos de seguridad.

- Mantener la integridad y consistencia de los datos.
- Acceso compartido de respaldo y recuperación. (20)

El SGDB utilizado es:

➤ **PostgreSQL**

PostgreSQL, es un sistema de gestión de base de datos objeto – relacional (ORDBMS), basado en PostgreSQL, versión 4,2. El mismo posee las siguientes características:

- Es de libre uso.
- Alta concurrencia: mediante un sistema denominado MVCC (acceso concurrente multiversión, por sus siglas en inglés), PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
- Amplia variedad de tipos nativos: provee nativamente soporte para: números de precisión arbitraria, textos de largo ilimitado, figuras geométricas, direcciones IP (Ipv4 e Ipv6), bloques de direcciones estilo CIDR, direcciones MAC.
- Claves ajenas, también denominadas Llaves ajenas o Claves foráneas.(21)

1.3.7. Servidores Web

Un servidor web, es un software que provee información a los usuarios mediante un navegador web con el cual se comunica por el protocolo HTTP.

➤ **Apache**

Apache, es un software de código abierto multiplataforma muy potente y el más usado tanto en LINUX como en otros sistemas operativos, por lo que su documentación es amplia y su comunidad de desarrollo también es muy vasta. Al ser su configuración a través de ficheros resulta ser muy sencilla y eficiente. Presenta soporte avanzado para programas CGI (Common Gateway Interface), SSI (Server-Side Includes) y SSL (Secured Socket Layer). Apache trabaja de modo muy eficiente con PHP y otros lenguajes del lado del servidor, además, provee una fácil interacción entre el Sistema Gestor de Bases de Datos PostgreSQL y el lenguaje de programación Web PHP.

1.4. Propuesta de solución

Teniendo en cuenta las necesidades existentes en la colección El Navegante en su versión multiplataforma se propone el desarrollo del módulo “Maestro”, a través del cual los maestros pueden gestionar toda la información referente a los estudiantes, profesores, noticias, escuelas, grupos académicos, categorías de revisión, trazas, artículos de interés y recomendaciones metodológicas necesarias para mejorar su preparación. En el desarrollo del módulo propuesto, se hace uso de las siguientes herramientas: como metodología de desarrollo, RUP, la cual provee los pasos necesarios a seguir para lograr un desarrollo exitoso; los lenguajes del lado del cliente a utilizar son: CSS, HTML, Java Script y el framework de Java Script JQuery para facilitar la visualización y darle una estética y efectos visuales agradables a la aplicación; como lenguajes del servidor se utilizaran PHP y el framework de desarrollo Symfony 1.4 de PHP; el lenguaje de modelado UML, la herramienta CASE Visual Paradigm, el patrón de arquitectura Modelo - Vista - Controlador, el servidor web Apache 2, y como Sistema Gestor de Bases de Datos PostgreSQL 8.4. Se utiliza NetBeans 6.8 en el desarrollo completo del módulo propuesto por su capacidad de soportar todos los lenguajes mencionados, tanto los del cliente como los del servidor. Todas estas tecnologías y lenguajes presentan una gran comunidad en la web, ya que, son herramientas libres; además, fueron estudiadas, analizadas y aprobadas por la dirección del proyecto para utilizarlas en el desarrollo de la colección El Navegante en su versión multiplataforma.

1.5. Conclusiones del capítulo

Una vez culminado el desarrollo del presente capítulo

- A pesar de existir diversas aplicaciones similares a la que se propone como posible solución, ninguna da solución a los problemas identificados; por tal motivo se requiere implementar la aplicación antes propuesta.
- Con los lenguajes de programación, así como las tecnologías de desarrollo descritas se logra dar cumplimiento a los objetivos planteados de forma exitosa.

Capítulo 2. Características del Sistema

Introducción

Con el objetivo de realizar el diseño de un sistema informático eficiente, es necesario dejar claro todos los conceptos y sus relaciones utilizando un lenguaje perceptible por todos los involucrados. En el presente capítulo se describe el Modelo del Dominio y sus principales conceptos. Se exponen además, los requisitos funcionales y no funcionales requeridos en el desarrollo de la aplicación, así como, los actores que interactúan con el sistema, el Diagrama de Caso de Uso del sistema y la descripción de cada uno de ellos.

2.1. Descripción del Modelo de Dominio

El Modelo de Dominio, es un artefacto que ayuda a comprender los conceptos utilizados por los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación propuesta.

Como consecuencia de la dificultad existente para determinar el proceso de negocio debido a fronteras que dificultan ver claramente, quienes son las personas que lo inician, quienes son los beneficiados con cada uno de estos procesos, quienes son las personas que desarrollan las actividades en cada uno de estos procesos, se propone el desarrollo de un Modelo de Dominio, en el cual se encierran los objetos más importantes que existen, o los eventos que suceden en el entorno donde se encontrará el sistema.

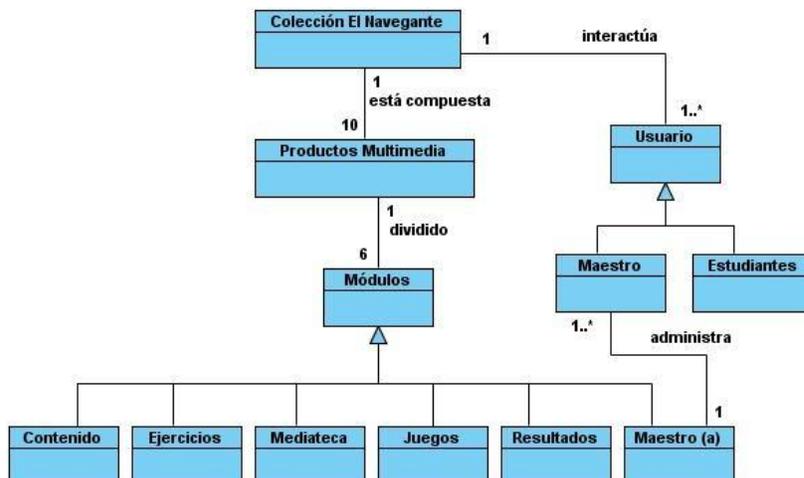


Figura 2.1. Modelo de Dominio

- **Usuario:** persona que interactúa con el software y puede realizar determinadas actividades según el rol al cual pertenezca.
- **Maestro:** persona autorizada que interactúa con la aplicación y puede tener el cargo de control de estudio o administrador del sistema. Es el encargado de educar, guiar y supervisar el proceso de enseñanza–aprendizaje en los estudiantes.
- **Estudiante:** persona que cursa estudios en las instituciones de nivel secundario y tienen permiso para realizar determinadas acciones con el fin de fomentar su aprendizaje.
- **Colección El Navegante:** constituye un conjunto de software educativo orientados a un área de conocimiento común.
- **Productos multimedia:** productos que mezclan diversos tipos de medias. (audio, video, imagen, texto, etc.)
- **Módulos:** conjunto de elementos fundamentales que integran un producto de la colección El Navegante.
- **Contenido:** módulo en el cual se presentan los contenidos o temas de las asignaturas correspondientes al software.
- **Ejercicio:** módulo encargado de controlar el aprendizaje a través de cuestionarios que emiten un criterio evaluativo a cada una de las respuestas.
- **Mediateca:** módulo encargado de gestionar todo lo referente a recursos multimedia mediante el cual los estudiantes pueden interactuar con imágenes, videos, textos, siendo esto muy importante en el proceso de aprendizaje de los mismos.
- **Juegos:** módulo mediante el cual los estudiantes pueden ejercitar su conocimiento y habilidades mientras juegan.
- **Resultados:** módulo encargado de visualizar el itinerario del estudiante durante su interacción con los restantes módulos.
- **Maestro(a):** módulo mediante el cual los profesores pueden gestionar toda la información referente a los estudiantes, profesores, escuelas, noticias, grupos académicos, recomendaciones metodológicas, así como, evaluar y controlar el avance en el desarrollo del aprendizaje del estudiante.

2.2. Especificación de requisitos

2.2.1. Requisitos funcionales (RF)

- RF 1-** Mostrarlas secciones de consulta para el maestro.
 - RF 1.1.** Consultar un artículo.
 - RF 1.2.** Consultar recomendaciones metodológicas.
 - RF 1.3.** Guardar un artículo.
- RF 2-** Consultar ejercicios resueltos.
- RF 3-** Cambiar contraseña a estudiantes.
- RF 4-** Realizar acciones con las trazas (trayectoria) de los estudiantes en el software.
 - RF 4.1.** Eliminar trazas.
 - RF 4.2.** Importar las trazas de los estudiantes.
 - RF 4.3.** Exportar las trazas de los estudiantes.
- RF 5-** Realizar acciones en relación a noticias
 - RF 5.1.** Adicionar noticias.
 - RF 5.2.** Eliminar noticias.
 - RF 5.3.** Modificar noticias.
 - RF 5.4.** Mostrar noticias.
- RF 6-** Realizar acciones en relación a los estudiantes.
 - RF 6.1.** Adicionar estudiantes.
 - RF 6.2.** Modificar los datos de los estudiantes.
 - RF 6.3.** Eliminar estudiantes.
- RF 7-** Listar estudiantes.
- RF 8-** Listar maestros.
- RF 9-** Realizar acciones en relación a los maestros (as).
 - RF 9.1.** Adicionar maestros (as).
 - RF 9.2.** Modificar los datos de los maestros (as).
 - RF 9.3.** Eliminar maestros (as).
 - RF 9.4.** Listar maestros (as).
- RF 10-** Realizar acciones en relación a las escuelas.
 - RF 10.1.** Adicionar escuelas.

RF 10.2. Eliminar escuelas.

RF 10.3. Modificar los datos de las escuelas.

RF 11- Realizar acciones en relación a los grupos académicos.

RF 11.1. Adicionar grupos académicos.

RF 11.2. Eliminar grupos académicos

RF 11.3. Modificar los datos de los grupos académicos.

RF 12- Modificar categorías de revisión.

RF 13- Importar listado de estudiantes.

RF 14- Consultar las trazas (trayectoria) de los maestros en el sistema.

2.2.2. Requisitos no funcionales

➤ **Requisitos de hardware**

- Resolución de la pantalla: 1024 x 600 píxeles.

Uso local

- Procesador Pentium 500 MHz
- 256 MB RAM
- 9 GB de espacio en disco duro.
- Tiempo de respuesta de una página: no mayor a 30 segundos.

Uso en red

- Velocidad de conexión mínima: 10Mbit/s.
- Cantidad máxima de máquinas conectadas: 20.
- 128 MB RAM.
- 9 GB de HDD (capacidad del disco duro).
- Procesador Pentium 500 MHz
- Velocidad de respuesta: 5 minutos.

➤ **Requisitos de software**

- Computadora personal con navegador Mozilla Firefox versión 3.5 o cualquier otra versión compatible con éste.

- Sistema operativo: Canaima versión 2.1 y Ubuntu en su versión 10.04. La aplicación podrá utilizarse en los sistemas operativos Linux, Mac y Windows.

- **Requisitos de restricciones de diseño e implementación**
 - Gestor de base de datos: PostgreSQL.
 - Estilo arquitectónico empleado: Modelo – Vista – Controlador.
 - Lenguajes de programación: JavaScript, PHP 5.2.
 - Framework: Symfony 1.4, JQuery 1.4
 - Apache 2.x.

- **Requisitos de apariencia e interfaz externa**
 - El diseño de las interfaces debe ser amigable y sencilla.
 - El sistema deberá proporcionar una organización clara y correcta de la información, permitiendo así, la interpretación inequívoca de ésta.
 - El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.

- **Requisitos legales**
 - Las imágenes que se utilicen en el producto deben tener permiso legal de sus autores y su aprobación para hacer uso de ellas.
 - La aplicación debe ser liberada bajo alguna de las licencias de software libre.

- **Requisitos de soporte**
 - Se impartirán clases a los profesores venezolanos para explicar el funcionamiento y utilidad del producto.
 - Se realizará transferencia tecnológica de la colección a los clientes.

- **Requisitos de portabilidad**
 - El sistema podrá ser utilizado bajo cualquier sistema operativo.

2.3. Modelo de Casos de Uso del Sistema

2.3.1. Actores del sistema

Tabla 2.1. Actores del sistema y su descripción

Actores	Descripción
Maestro	<p>Representa a una persona autorizada que interactúa con la aplicación y puede realizar determinadas actividades, tales como:</p> <ul style="list-style-type: none"> • Cambiar la contraseña de los estudiantes • Consultar ejercicios resueltos por los. estudiantes. • Consultar artículos de interés, así como recomendaciones metodológicas. • Adicionar, eliminar y modificar una noticia determinada.
Control de estudio	<p>Representa a una persona autorizada que interactúa con la aplicación y puede:</p> <ul style="list-style-type: none"> • Importar un listado de estudiantes. • Consultar un listado de estudiantes determinado. • Adicionar, eliminar o modificar datos, tanto de estudiantes, como grupos académicos y escuelas. • Adicionar, eliminar o modificar los datos de los maestros. • Consultar un listado de maestros.
Administrador	<p>Representa a una persona autorizada que interactúa con la aplicación y puede:</p> <ul style="list-style-type: none"> • Modificar las categorías de revisión. • Consultar la trayectoria de los maestros (as) en el software. • Eliminar, importar y exportar las trazas (trayectoria) de los estudiantes en el software.

2.3.2. Diagrama de Casos de Uso del sistema

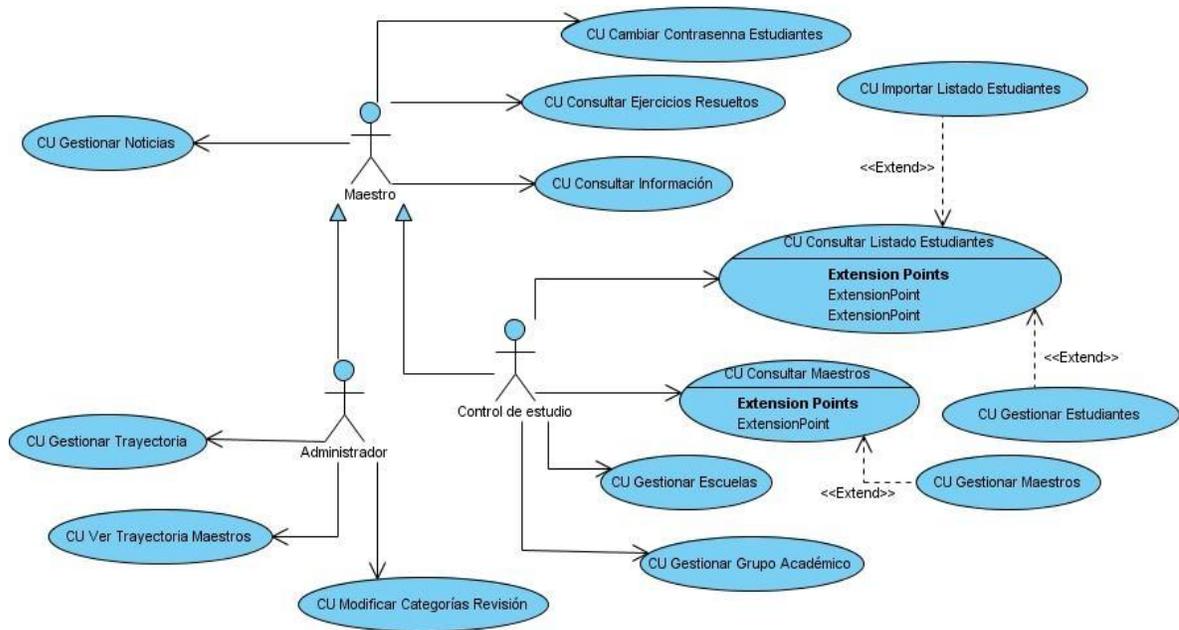


Figura 2.2.Diagrama de Casos de Uso del Sistema

2.3.3. Descripción de los casos de uso

Tabla 2.2. Descripción del caso de uso: Gestionar Escuela

Caso de uso	Gestionar escuelas
Actores	Control de estudio
Resumen	El caso de uso inicia cuando el actor accede a al sistema y selecciona la opción “Gestor de escuelas”. El sistema permite adicionar, modificar o eliminar los datos de las escuelas. Si el actor selecciona la opción de adicionar una escuela, el sistema permite insertar los datos requeridos para realizar esta acción. Si el actor selecciona la opción de modificar los datos de una escuela, el sistema muestra los datos de la escuela seleccionada y permite modificarlos. Si el actor selecciona la opción de eliminar una escuela, el sistema permite seleccionar una escuela para ser eliminada. El caso de uso termina.
Precondiciones	Debe haberse generado el entorno de trabajo del actor autenticado
Referencias	RF 10, RF 10.1, RF 10.2 y RF 10.3

Flujo normal de eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Gestor de escuelas”	
	2. Permite realizar las siguientes actividades: <ul style="list-style-type: none"> • Adicionar una nueva escuela • Modificar los datos de una escuela, ver sección 1: “Modificar escuela” • Eliminar una escuela, ver sección 2: “Eliminar escuela”
3. Selecciona la opción “Adicionar”	
	4. Brinda la posibilidad de insertar los siguientes datos de la escuela: <ul style="list-style-type: none"> • Nombre • Ciudad Seleccionar: <ul style="list-style-type: none"> • Estado Y permite: <ul style="list-style-type: none"> • Cancelar la operación en cualquier momento • Aceptar
5. Introduce los datos: <ul style="list-style-type: none"> • Nombre(s) • Ciudad Y selecciona los datos: <ul style="list-style-type: none"> • Estado 	
6. Selecciona la opción “Aceptar”	
	7. Valida que los datos introducidos sean correctos
	8. Crea una nueva escuela

	9. Muestra un mensaje informando: “Se adicionó correctamente una nueva escuela”
	10. El caso de uso termina
Flujos Alternos	
6.a Selecciona la opción “Cancelar”	
Acciones del actor	Respuesta del sistema
	6. a.1 Elimina los datos creados
	6. a.2 Muestra un mensaje informando: “La acción ha sido cancelada”
	6. a.3 Regresa a la vista anterior
	6. a.4 El caso de uso termina
7.a Existen datos incorrectos	
Acciones del actor	Respuesta del sistema
	7.a.1 Muestra un mensaje informando: “Existen campos escritos incorrectamente, por favor rectifique estos datos”
	7.a.2 Muestra el indicador de campos incorrectos
	7.a.3 Regresa al paso 2 del flujo básico
7. b Existen campos incompletos	
Acciones del actor	Respuesta del sistema
	7. b.1 Muestra un mensaje informando: “Existen campos incompletos, por favor, complete los datos”
	7. b.2 Muestra el indicador de campos incorrectos
	7.b. 3 Regresa al paso 2 del flujo básico
Sección 1: “Modificar escuela”	
Flujo básico	
Acciones del actor	Respuesta del sistema
1. Selecciona una escuela y elige la opción: “Modificar”	

	<p>2. Muestra los datos de la escuela :</p> <ul style="list-style-type: none"> • Nombre • Ciudad <p>Seleccionar:</p> <ul style="list-style-type: none"> • Estado <p>Y permite:</p> <ul style="list-style-type: none"> • Modificar los datos de la escuela • Aceptar • Cancelar la operación
3. Modifica los datos	
4. Selecciona la opción “Aceptar”	
	5. Valida los nuevos datos introducidos
	6. Guarda los datos en el sistema
	7. Muestra un mensaje informando: “Los datos fueron modificados satisfactoriamente”
	8. El caso de uso termina
Flujos alternos	
4.a Selecciona la opción “Cancelar”	
Acciones del actor	Respuesta del sistema
	4. a.1 Elimina los datos creados
	4. a.2 Muestra un mensaje informando: “La acción ha sido cancelada”
	4.a.3 Regresa a la vista anterior
	4.a.4 El caso de uso termina
5.a Existen datos incorrectos	
Acciones del actor	Respuesta del sistema
	5. b.1 Muestra un mensaje de error: “Existen campos escritos incorrectamente, por favor rectifique estos datos”
	5. b.2 Muestra el indicador de campos incorrectos

	5. b.3 Regresa al paso 2 del Flujo Básico
Sección 2: “Eliminar escuela”	
Flujo básico	
Acciones del actor	Respuesta del sistema
1. Selecciona una escuela y elige la opción: “Eliminar”	
	2. Muestra el mensaje de confirmación: “Se eliminarán todos los datos de la escuela seleccionada. ¿Desea continuar?” Permite: <ul style="list-style-type: none"> • Aceptar • Cancelar la operación
3. Selecciona la opción “Aceptar”	
	4. Muestra el mensaje de información: “Se ha eliminado la escuela seleccionada”
	5. Regresa a la vista anterior actualizando los datos
	6. El caso de uso termina
Flujos alternos	
3.a Selecciona la opción “Eliminar” sin haber seleccionado una escuela	
Acciones del actor	Respuesta del sistema
	3. a.1 Muestra un mensaje informando: “Debe seleccionar un elemento para ser eliminado”
	3. a.2 Regresa al paso 2 del Flujo Básico
5. a Selecciona la opción “Cancelar”	
Acciones del actor	Respuesta del sistema
	5. a.1 Elimina los datos creados
	5. a.2 Muestra un mensaje informando: “La acción ha sido cancelada”
	5.a.3 Regresa a la vista anterior

5.a.4 El caso de uso termina

Tabla 2.3. Descripción del caso de uso: Gestionar Estudiantes

Caso de uso	Gestionar estudiantes	
Actores	Control de estudio	
Resumen	El caso de uso inicia cuando el actor accede a al sistema y selecciona la opción “Gestor de estudiantes”. El sistema permite incluir, modificar o eliminar los datos de los estudiantes. Si el actor selecciona la opción de incluir un nuevo estudiante, el sistema permite insertar los datos requeridos para realizar esta acción. Si el actor selecciona la opción de modificar los datos de un estudiante, el sistema muestra los datos del estudiante seleccionado y permite modificarlos. Si el actor selecciona la opción de eliminar un estudiante, el sistema permite eliminar un estudiante previamente seleccionado. El caso de uso termina.	
Precondiciones	Debe haberse generado el entorno de trabajo del usuario autenticado	
Referencias	RF 6, RF 6.1, RF 6.2, RF 6.3, RF 6.4	
Flujo normal de eventos		
Acciones del actor	Respuesta del sistema	
1. El caso de uso inicia cuando el actor selecciona la opción “Gestor de estudiantes”		
	2. Permite realizar las siguientes actividades: <ul style="list-style-type: none"> • incluir un nuevo estudiante • Modificar los datos de un estudiante, ver sección 1: “Modificar” • Eliminar estudiantes, ver sección 2: “Eliminar” Además permite: <ul style="list-style-type: none"> • Importar un listado de estudiantes • Listar los datos de los estudiantes 	

3. Selecciona la opción de incluir un nuevo estudiante	
	<p>4. Brinda la posibilidad de insertar los siguientes datos:</p> <ul style="list-style-type: none"> • Cédula de identidad • Contraseña • Nombre (s) • Apellido (s) • Sección • Año • Institución <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar • Cancelar la operación
	5. Adiciona un nuevo estudiante y valida los datos introducidos
	6. Muestra un mensaje informando: “Se adicionó correctamente un nuevo estudiante”
	7. El caso de uso termina
Flujos Alternos	
3.a Selecciona la opción “Importar listado”	
Acciones del actor	Respuesta del sistema
	3. a.1 Permite importar un listado de estudiantes. Ver <u>CU Importar listado Estudiantes</u>
	3. a.2 Regresa al paso 2 del flujo básico
3.b Selecciona la opción “Listar estudiantes”	
Acciones del actor	Respuesta del sistema
	3. b.1 Permite seleccionar uno o más criterios de búsqueda para listar los estudiantes. Ver <u>CU Consultar</u>

	<u>listado estudiantes</u>
	3. b.2 Regresa al paso 2 del flujo básico
4.a Selecciona la opción “Cancelar”	
Acciones del actor	Respuesta del sistema
	4. a.1 Elimina los datos creados
	4. a.2 Muestra un mensaje informando: “La acción ha sido cancelada”
	4.a.3 Regresa a la vista anterior
	4.a.4 El caso de uso termina
5. a Existen campos incorrectos	
Acciones del actor	Respuesta del sistema
	5. a.1 Muestra un mensaje informando: “Existen campos escritos incorrectamente, por favor, rectifique estos datos”
	5. a. 2 Muestra el indicador de campos incorrectos
	5.a. 3 Regresa al paso 4 del flujo básico
5.b Existen campos incompletos	
Acciones del actor	Respuesta del sistema
	5. b.1 Muestra un mensaje informando: “Existen campos incompletos, por favor, complete los datos”
	5. b. 2 Muestra el indicador de campos incorrectos
	5. b. 3 Regresa al paso 4 del flujo básico
Sección 1: “Modificar”	
Flujo básico	
Acciones del actor	Respuesta del sistema
1. Selecciona un estudiante y elige la opción: “Modificar”	
	2. Muestra los siguientes datos de los estudiantes: <ul style="list-style-type: none"> • Nombre (s)

	<ul style="list-style-type: none"> • Apellido (s) • Institución • Año • Sección <p>Y permite:</p> <ul style="list-style-type: none"> • Modificar sus datos • Cancelar la operación • Aceptar
3. Modifica los datos deseados y selecciona la opción “Aceptar”	
	4. Modifica y valida los datos
	5. Guarda los cambios en el sistema
	6. Muestra un mensaje informando: “Se han actualizado los cambios realizados sobre el estudiante”
	7. El caso de uso termina
Flujos alternos	
3.a Selecciona la opción “Cancelar”	
Acciones del actor	Respuesta del sistema
	3. a.1 Elimina los datos creados
	3. a.2 Muestra un mensaje informando: “La acción ha sido cancelada”
	3.a.3 Regresa a la vista anterior
	3.a.4 El caso de uso termina
4.a Existen datos incorrectos	
Acciones del actor	Respuesta del sistema
	4. a.1 Muestra un mensaje de error: “Existen campos escritos incorrectamente, por favor rectifique estos datos”.
	4.a.2 Muestra el indicador de campos incorrectos

	4. a.3 Regresa al paso 2 del Flujo Básico
Sección 2: “Eliminar”	
Flujo básico	
Acciones del actor	Respuesta del sistema
1. Selecciona un estudiante y elige la opción: “Eliminar”	
	2. Muestra el mensaje de confirmación: “Se eliminarán todos los datos del estudiante seleccionado. ¿Desea continuar?” Permite: <ul style="list-style-type: none"> • Aceptar • Cancelar la operación
3. Selecciona la opción “Aceptar”	
	4. Muestra el mensaje de información: “El estudiante fue eliminado satisfactoriamente”
	5. Regresa a la vista anterior actualizando los datos
	6. El caso de uso termina
Flujos alternos	
3.a Selecciona la opción “Eliminar” sin haber seleccionado previamente un estudiante	
Acciones del actor	Respuesta del sistema
	3. a.1 Muestra un mensaje informando: “Debe seleccionar un elemento para ser eliminado”
	3. a.2 Regresa al paso 2 del Flujo Básico
5. a Selecciona la opción “Cancelar”	
Acciones del actor	Respuesta del sistema
	5. a.1 Elimina los datos creados
	5. a.2 Muestra un mensaje informando: “La acción ha sido cancelada”
	5.a.3 Regresa a la vista anterior

5.a.4 El caso de uso termina

Tabla 2.4. Descripción del caso de uso: Gestionar Grupo Académico

Caso de uso	Gestionar grupo académico	
Actores	Control de estudio	
Resumen	El caso de uso inicia cuando el actor accede a al sistema y selecciona la opción “Gestor de grupo académico”. El sistema permite incluir, modificar o eliminar los datos de un (o varios) grupo académico. Si el actor selecciona la opción de incluir un nuevo grupo, el sistema permite insertar los datos requeridos para realizar esta acción. Si el actor selecciona la opción de modificar los datos de un grupo académico, el sistema muestra los datos del grupo seleccionado y permite modificarlos. Si el actor selecciona la opción de eliminar un grupo académico, el sistema permite eliminar un grupo académico previamente seleccionado. El caso de uso termina.	
Precondiciones	Debe haberse generado el entorno de trabajo del usuario autenticado	
Referencias	RF 11, RF 11.1, RF 11.2, RF 11.3, RF 11.4	
Flujo normal de eventos		
Acciones del actor	Respuesta del sistema	
1. El caso de uso inicia cuando el actor selecciona la opción “Gestor de estudiantes”		
	2. Permite realizar las siguientes actividades: <ul style="list-style-type: none"> • Incluir un nuevo grupo académico • Modificar los datos de un grupo académico, ver sección 1: “Modificar” • Eliminar grupo académico, ver sección 2: “Eliminar” 	
3. Selecciona la opción de incluir un nuevo grupo académico		

	<p>4. Brinda la posibilidad de insertar el siguiente dato:</p> <ul style="list-style-type: none"> • Grupo <p>Seleccionar:</p> <ul style="list-style-type: none"> • Escuela • Grado <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar • Cancelar la operación
<p>5. Introduce el dato:</p> <ul style="list-style-type: none"> • Grupo <p>Y selecciona:</p> <ul style="list-style-type: none"> • Escuela • Grado <p>Selecciona la opción “Aceptar”</p>	
	<p>6. Crea un nuevo grupo académico y valida los datos</p>
	<p>7. Muestra un mensaje informando: “Se adicionó correctamente un nuevo grupo académico”</p>
	<p>8. El caso de uso termina</p>
Flujos Alternos	
5.a Selecciona la opción “Cancelar”	
Acciones del actor	Respuesta del sistema
	5. a.1 Elimina los datos creados
	5. a.2 Muestra un mensaje informando: “La acción ha sido cancelada”
	5.a.3 Regresa a la vista anterior
	5.a.4 El caso de uso termina
6. a Existen campos incorrectos	
Acciones del actor	Respuesta del sistema
	6. a.1 Muestra un mensaje informando: “Existen campos

	escritos incorrectamente, por favor, rectifique estos datos”
	6. a. 2 Muestra el indicador de campos incorrectos
	6.a. 3 Regresa al paso 4 del flujo básico
6. b Existen campos incompletos	
Acciones del actor	Respuesta del sistema
	6. b.1 Muestra un mensaje informando: “Existen campos incompletos, por favor, complete los datos”
	6. b. 2 Muestra el indicador de campos incorrectos
	6. b. 3 Regresa al paso 4 del flujo básico
Sección 1: “Modificar”	
Flujo básico	
Acciones del actor	Respuesta del sistema
1. Selecciona un grupo académico y elige la opción “Modificar”	
	<p>2. Muestra los siguientes datos de los grupos académicos:</p> <ul style="list-style-type: none"> • Grupo <p>Seleccionar:</p> <ul style="list-style-type: none"> • Escuela • Grado <p>Y permite:</p> <ul style="list-style-type: none"> • Modificar los datos de un grupo • Cancelar la operación en cualquier momento
3. Modifica los datos deseados y selecciona la opción “Aceptar”	
	4. Valida y guardo los cambios en el sistema
	5. Muestra un mensaje informando: “Los datos fueron modificados satisfactoriamente”
	6. El caso de uso termina

Flujos alternos	
3.a Selecciona la opción “Cancelar”	
Acciones del actor	Respuesta del sistema
	3. a.1 Elimina los datos creados
	3. a.2 Muestra un mensaje informando: “La acción ha sido cancelada”
	3.a.3 Regresa a la vista anterior
	3.a.4 El caso de uso termina
4.a Existen datos incorrectos	
Acciones del actor	Respuesta del sistema
	4. a.1 Muestra un mensaje informando: “Existen datos escritos incorrectamente. Por favor, rectifique estos datos”
	4.a.2 Muestra el indicador de campos incorrectos
	4. a.3 Regresa al paso 2 del Flujo Básico
Sección 2: “Eliminar”	
Flujo básico	
Acciones del actor	Respuesta del sistema
1. Selecciona un grupo y elige la opción: “Eliminar”	
	2. Muestra el mensaje de confirmación: “Se eliminarán todos los datos del grupo seleccionado. ¿Desea continuar?” Y permite: <ul style="list-style-type: none"> • Aceptar • Cancelar la operación
3. Selecciona la opción “Aceptar”	
	4. Muestra el mensaje de información: “El grupo académico fue eliminado satisfactoriamente”
	5. Regresa a la vista anterior actualizando los datos

	6. El caso de uso termina
Flujos alternos	
3.a Selecciona la opción “Eliminar” sin haber seleccionado previamente un grupo	
Acciones del actor	Respuesta del sistema
	3. a.1 Muestra un mensaje informando: “Debe seleccionar un elemento para ser eliminado”
	3. a.2 Regresa al paso 2 del Flujo Básico.
5. a Selecciona la opción “Cancelar”	
Acciones del actor	Respuesta del sistema
	5. a.1 Elimina los datos creados
	5. a.2 Muestra un mensaje informando: “La acción ha sido cancelada”
	5.a.3 Regresa a la vista anterior
	5.a.4 El caso de uso termina

Tabla 2.5. Descripción del caso de uso: Gestionar Maestros

Caso de uso	Gestionar maestros
Actores	Control de estudio
Resumen	El caso de uso inicia cuando el actor accede a al sistema y selecciona la opción “Gestor de maestros”. El sistema permite incluir, modificar o eliminar los datos de los maestros. Si el actor selecciona la opción de incluir un nuevo maestro, el sistema permite insertar los datos requeridos para realizar esta acción. Si el actor selecciona la opción de modificar los datos de un maestro, el sistema muestra los datos del maestro seleccionado y permite modificarlos. Si el actor selecciona la opción de eliminar un maestro, el sistema permite eliminar un maestro previamente seleccionado. El caso de uso termina.
Precondiciones	Debe haberse generado el entorno de trabajo del usuario autenticado
Referencias	RF 9, RF 9.1, RF 9.2, RF 9.3, RF 9.4
Flujo normal de eventos	

Acciones del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor selecciona la opción “Gestor de maestras y maestros”	
	2. Permite realizar las siguientes actividades: <ul style="list-style-type: none"> • Incluir un nuevo maestro • Modificar los datos de los maestros (as), ver sección 1: “Modificar” • Eliminar maestros, ver sección 2: “Eliminar”
3. Selecciona la opción de incluir un nuevo maestro	
	4. Brinda la posibilidad de insertar los siguientes datos del maestro: <ul style="list-style-type: none"> • Nombre(s) • Apellido(s) • Asignatura • Cédula de identidad • Contraseña • Cargo Seleccionar: <ul style="list-style-type: none"> • Sección • Año • Institución Y permite: <ul style="list-style-type: none"> • Cancelar la operación en cualquier momento • Aceptar (permite guardar los datos)
5. Introduce los datos: <ul style="list-style-type: none"> • Nombre(s) 	

<ul style="list-style-type: none"> • Apellido(s) • Asignatura • Cédula de identidad • Contraseña • Cargo <p>Y selecciona:</p> <ul style="list-style-type: none"> • Sección • Año • Institución <p>Selecciona la opción "Aceptar"</p>	
	6. Valida los datos introducidos por el actor
	7. Crea un nuevo maestro (a)
	8. Muestra un mensaje de información: "Se adicionó correctamente un nuevo maestro (a)"
	9. El caso de uso termina
Flujos Alternos	
3.a Selecciona la opción "Consultar maestros"	
Acciones del actor	Respuesta del sistema
	3. a.1 Permite consultar un listado de maestros. Ver <u>CU Consultar Maestros (as)</u>
	3. a.2. Regresa al paso 2 del flujo básico
5.a Selecciona la opción "Cancelar"	
Acciones del actor	Respuesta del sistema
	5. a.1 Elimina los datos creados
	5. a.2 Muestra un mensaje informando: "La acción ha sido cancelada"
	5.a.3 Regresa a la vista anterior
	5.a.4 El caso de uso termina
6.a Existen datos incorrectos	

Acciones del actor	Respuesta del sistema
	6. a.1 Muestra un mensaje informando: “Existen datos escritos incorrectamente. Por favor, rectifique estos datos”
	6.a.2 Muestra el indicador de campos incorrectos
	6. a.3 Regresa al paso 4 del Flujo Básico
6.b Existen datos incompletos	
Acciones del actor	Respuesta del sistema
	6. b.1 Muestra un mensaje informando: “Existen campos incompletos, por favor, complete los datos”
	6. b. 2 Muestra el indicador de campos incorrectos
	6.b. 3Regresa al paso 4 del flujo básico
Sección 1: “Modificar”	
Flujo básico	
Acciones del actor	Respuesta del sistema
1. Selecciona un maestro (a) y elige la opción “Modificar”	
	<p>2. Muestra los datos de los siguientes datos de los maestros (as):</p> <ul style="list-style-type: none"> • Nombre(s) • Apellido(s) • Asignatura • Cédula de identidad • Contraseña • Cargo <p>Seleccionar:</p> <ul style="list-style-type: none"> • Sección • Año • Institución <p>Y permite:</p>

	<ul style="list-style-type: none"> • Modificar estos datos de maestros. • Cancelar la operación en cualquier momento • Aceptar
3. Modifica los datos deseados y selecciona la opción: "Aceptar"	
	4. Valida y guarda los nuevos datos
	5. Muestra un mensaje informando "Los datos fueron modificados satisfactoriamente."
	6. El caso de uso termina
Flujos alternos	
3.a Selecciona la opción "Cancelar"	
Acciones del actor	Respuesta del sistema
	3. a.1 Elimina los datos creados
	3. a.2 Muestra un mensaje informando: "La acción ha sido cancelada"
	3.a.3 Regresa a la vista anterior
	3.a.4 El caso de uso termina
4.a Existen datos incorrectos	
Acciones del actor	Respuesta del sistema
	4. a.1 Muestra un mensaje informando: "Existen datos escritos incorrectamente. Por favor, rectifique estos datos"
	4.a.2 Muestra el indicador de campos incorrectos
	4. a.3 Regresa al paso 2del Flujo Básico.
Sección 2: "Eliminar"	
Flujo básico	
Acciones del actor	Respuesta del sistema

1. Selecciona un maestro (a) y elige la opción: "Eliminar"	
	2. Muestra un mensaje de confirmación para eliminar el estudiante: "Se eliminaran todos los datos del maestro seleccionado. ¿Desea continuar?"
3. Selecciona la opción "Aceptar"	
	4. Muestra un mensaje informando: "El maestro (s) fue eliminado satisfactoriamente"
	5. Regresa a la vista anterior actualizando los datos
	6. El caso de uso termina
Flujos alternos	
3.a Selecciona la opción "Eliminar" sin haber seleccionado previamente un maestro	
Acciones del actor	Respuesta del sistema
	3. a.1 Muestra un mensaje informando: "Debe seleccionar un elemento para ser eliminado"
	3. a.2 Regresa al paso 2 del Flujo Básico
5. a Selecciona la opción "Cancelar"	
Acciones del actor	Respuesta del sistema
	5. a.1 Elimina los datos creados
	5. a.2 Muestra un mensaje informando: "La acción ha sido cancelada"
	5.a.3 Regresa a la vista anterior
	5.a.4 El caso de uso termina

Las restantes descripciones de los casos de uso se pueden consultar en los anexos, específicamente en el [Anexo 2](#).

2.4. Conclusiones del capítulo

Una vez culminado el desarrollo del presente capítulo

- Se tiene una visión más amplia sobre el entorno al cual se integrará el sistema a implementar
- Se describen, las propiedades, funcionalidades, el comportamiento interno, de la aplicación propuesta, y va a interactuar con los actores.

Capítulo 3. Análisis y Diseño

Introducción

En el presente capítulo se presenta el análisis y diseño del módulo propuesto, a través de los artefactos necesarios para su implementación. Tales artefactos son: Modelo de Análisis, Modelo de Diseño, Modelo de Datos y Diagrama de Clases Persistentes.

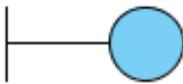
3.1. Modelo de Análisis

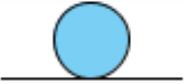
El Modelo de Análisis encierra en las clases y los objetos. Dichas clases se centran en la estructuración de los requisitos funcionales y son evidentes en el dominio del problema, ya que, representan conceptos y relaciones del dominio. Los diagramas de clases de diseño, representan la interacción del actor con el sistema de forma conceptual, es decir, es una representación visual de los objetos, resaltando las relaciones entre ellos, de manera que sea más fácil su comprensión, su modificación, su preparación y su mantenimiento en general.

En el Modelo de Análisis están contenidas las clases de análisis, las cuales se centran en los requisitos funcionales de la aplicación. Las cuales se describen a continuación:

3.1.1 Clases de análisis, estereotipos y funciones

Tabla 3.1. Clases de análisis y su descripción

Clase	Prototipo	Descripción
Clase interfaz (CI)	 CI_Clase_Interfaz	Se encarga del modelado de toda la interacción que pueda existir entre los actores y el sistema. Ejemplos: ventanas, formularios, comunicación con otros sistemas o dispositivos.
Clase controladora (CC)	 CC_Clase_Controladora	Coordina los eventos necesarios para la realización o especificación del caso de uso, es decir, son las que ejecutan el caso de uso. Usualmente son dependientes de la aplicación, además de tener control sobre las acciones a realizar.

Clase entidad (CE)	 CE_Clase_Entidad	Representan la información de larga duración y a menudo persistente que se maneja en el sistema.
--------------------	--	--

La realización del modelo de análisis, trae consigo un mejor entendimiento entre los miembros del equipo de desarrollo. Estudios previos han demostrado que la integración al desarrollo de software del análisis y el diseño como un único flujo de trabajo resulta mucho más efectivo, ya que, este suaviza la transición al diseño, permite tener una visión general del sistema propuesto, apoya el cambio a otras plataformas, así como la aplicación de ingeniería inversa a aplicaciones ya existentes. Los diagramas de clases de análisis se estructuran por cada caso de uso, es decir, por cada caso de uso, se obtendrá un diagrama de clases de análisis (DCA).

3.1.2. Diagramas de Clases de Análisis (DCA)

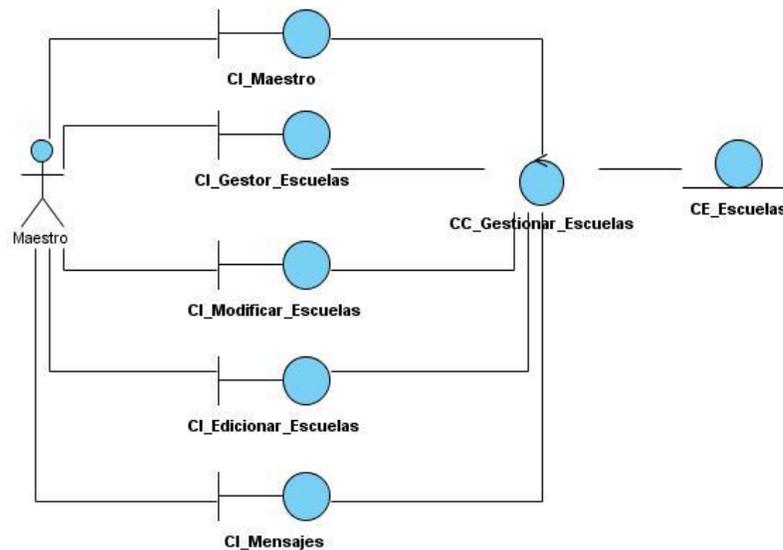


Figura 3.1. DCA_CU Gestionar Escuela

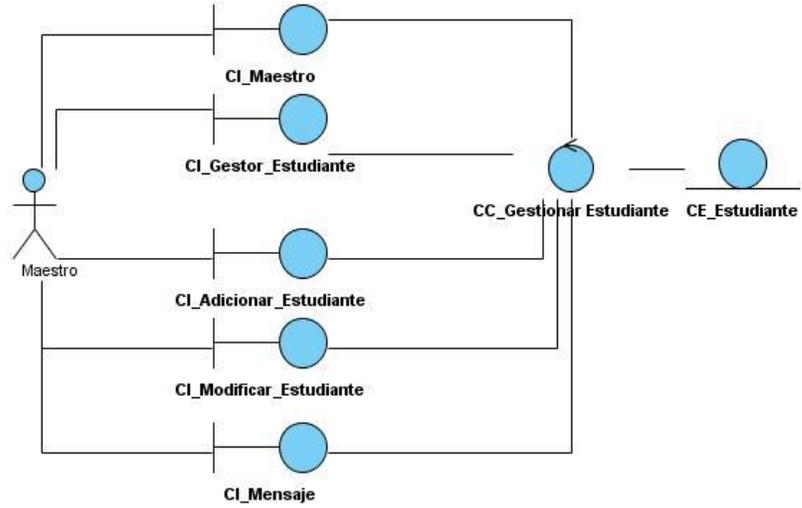


Figura 3.2. DCA_CU Gestionar Estudiante

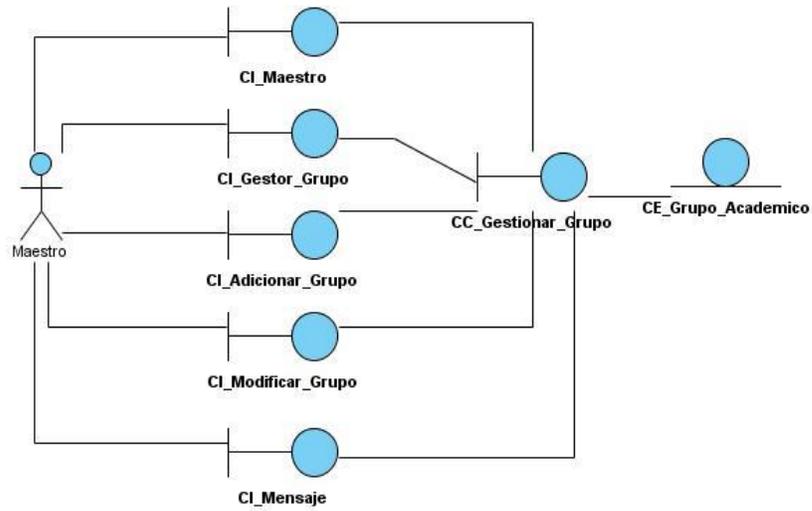


Figura 3.3. DCA_CU Gestionar Grupo Académico

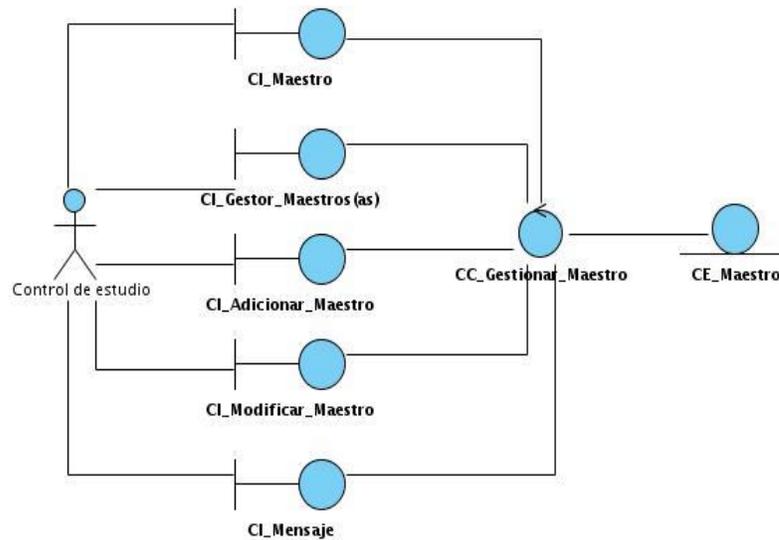


Figura 3.4. DCA_CU Gestionar Maestro

Las restantes descripciones de los casos de uso se pueden consultar en el [Anexos 3](#).

3.2. Modelo de diseño

El diseño proporciona una idea más completa de lo que es el software enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la implementación. El mismo es considerado como un refinamiento del análisis donde se tiene en cuenta los requisitos no funcionales del sistema, es decir, expresa como el sistema cumple sus objetivos. Es un modelo físico, dinámico y no genérico, el cual se utiliza para dar forma al sistema mientras intenta preservar la estructura definida por el modelo de análisis.

3.2.1. Terminologías utilizadas en los diagramas de clase de diseño

Tabla 3.2. Clases de diseño y su descripción

Clases	Estereotipos	Función
Server Page [SP] (página servidora)	 SP_Ejemplo	Representa una clase para el acceso a los datos. Su principal función es la de construir la CP
Client Page [CP] (página cliente)	 CP_Ejemplo	Representa una clase que interactúa con el cliente. Su función es visualizar, interactuar y mostrar todo lo que el usuario necesita.

Form (formulario)		Clase que interactúa directamente con el usuario. Su función es enviar los datos entrados por el usuario a la CP.
-------------------	---	---

Con el objetivo de lograr una mejor comprensión de los Diagramas de Clase del Diseño seguidamente mostrados, es necesario esclarecer los siguientes aspectos:

- Se usan paquetes para agrupar y mostrar las responsabilidades de las clases en el patrón Modelo Vista Controlador.
- El subsistema Doctrine, representa todos los componentes que utiliza el ORM Doctrine.
- El subsistema Componentes Symfony, contiene todos los componentes, dígame núcleo del framework, clases internas, ficheros, que brinda Symfony.
- El Modelo de Datos utilizado en cada clase del diseño es común, ya que poseen entre todos los archivos la misma relación.

3.2.2. Diagramas de clases del diseño (DCD)

A continuación se muestran los diferentes DCD para los respectivos casos de uso.

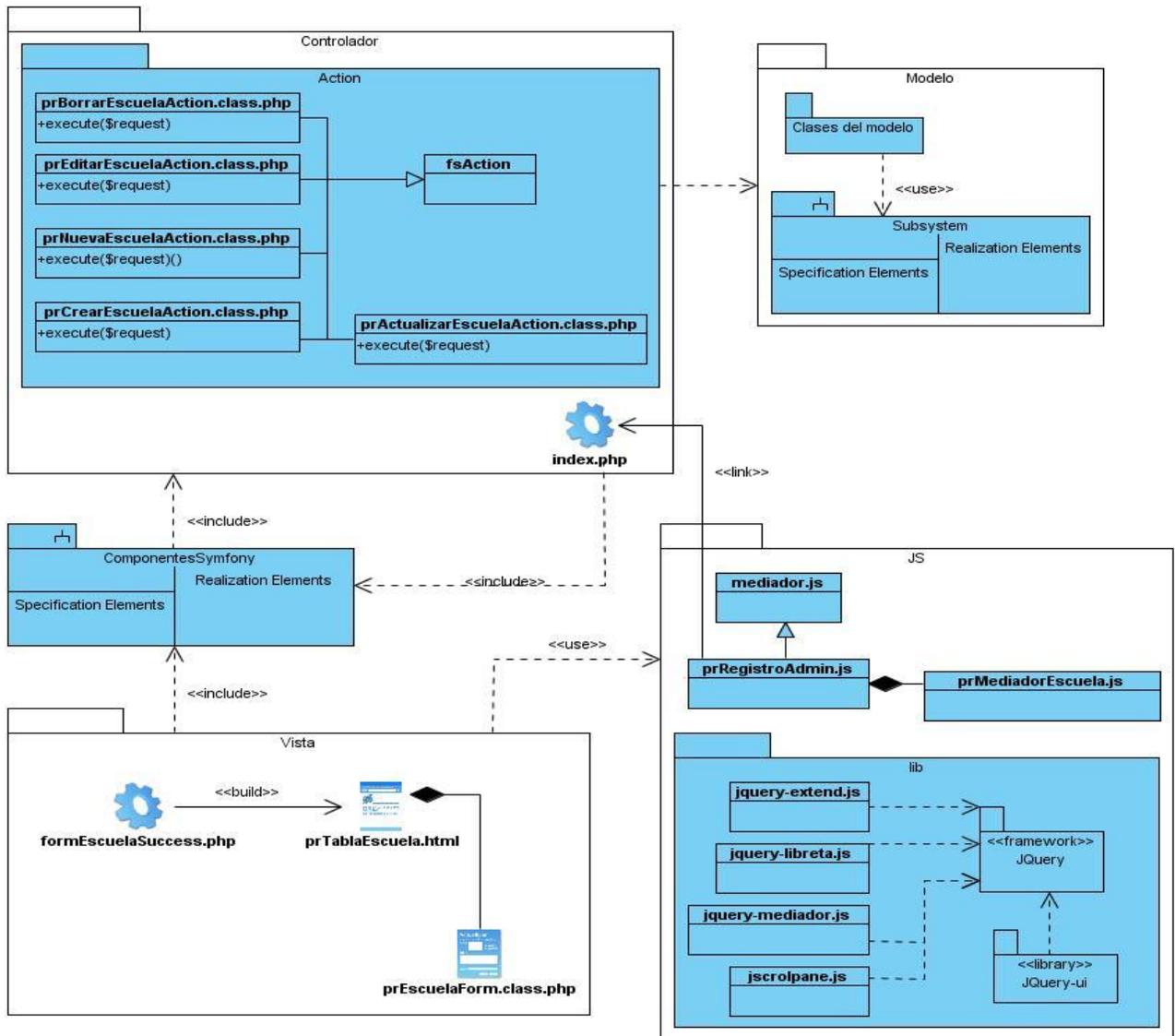


Figura 3.5. DCD_CU Gestionar Escuela

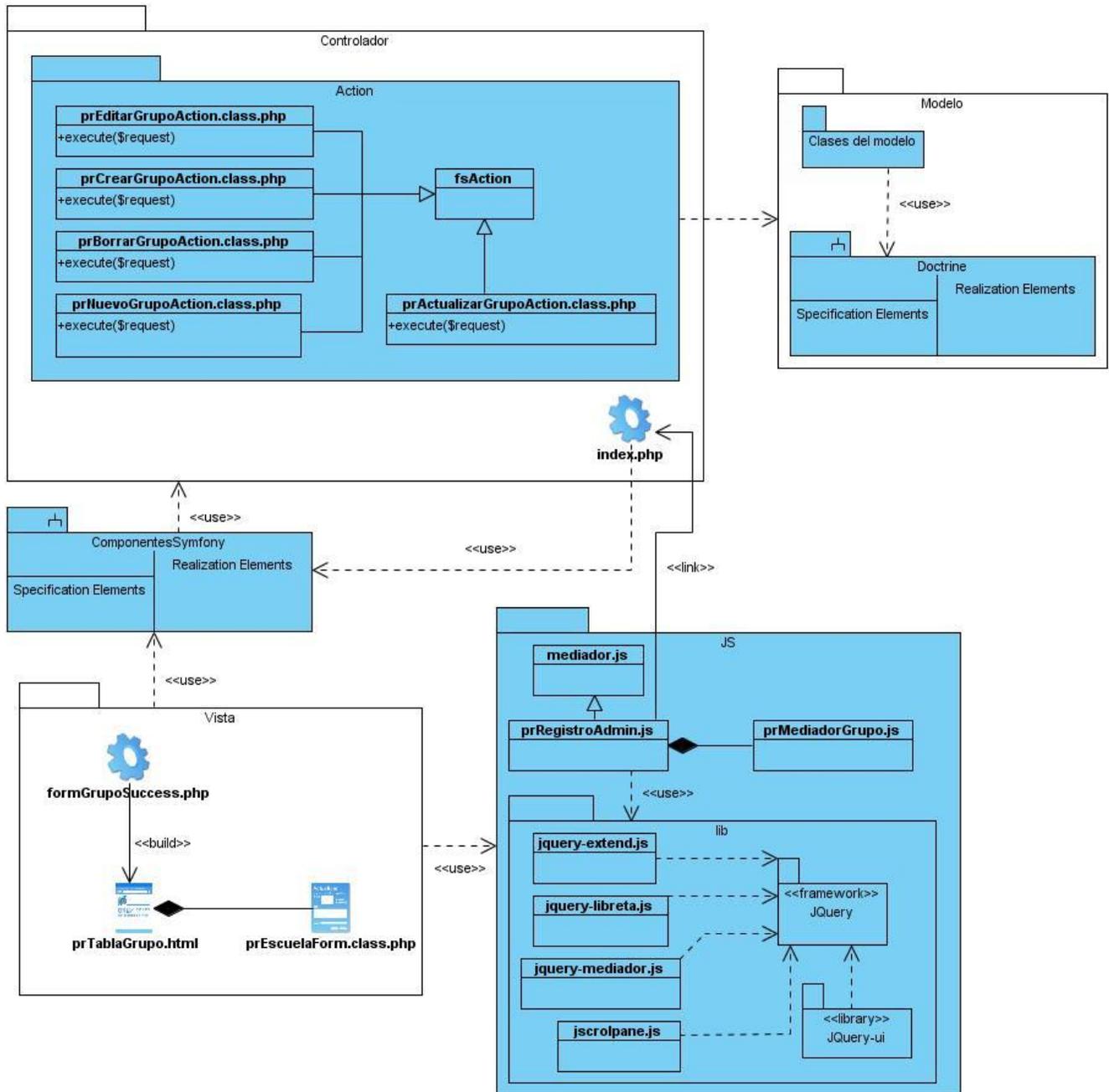


Figura 3.7. DCD_CU Gestionar Grupo Académico

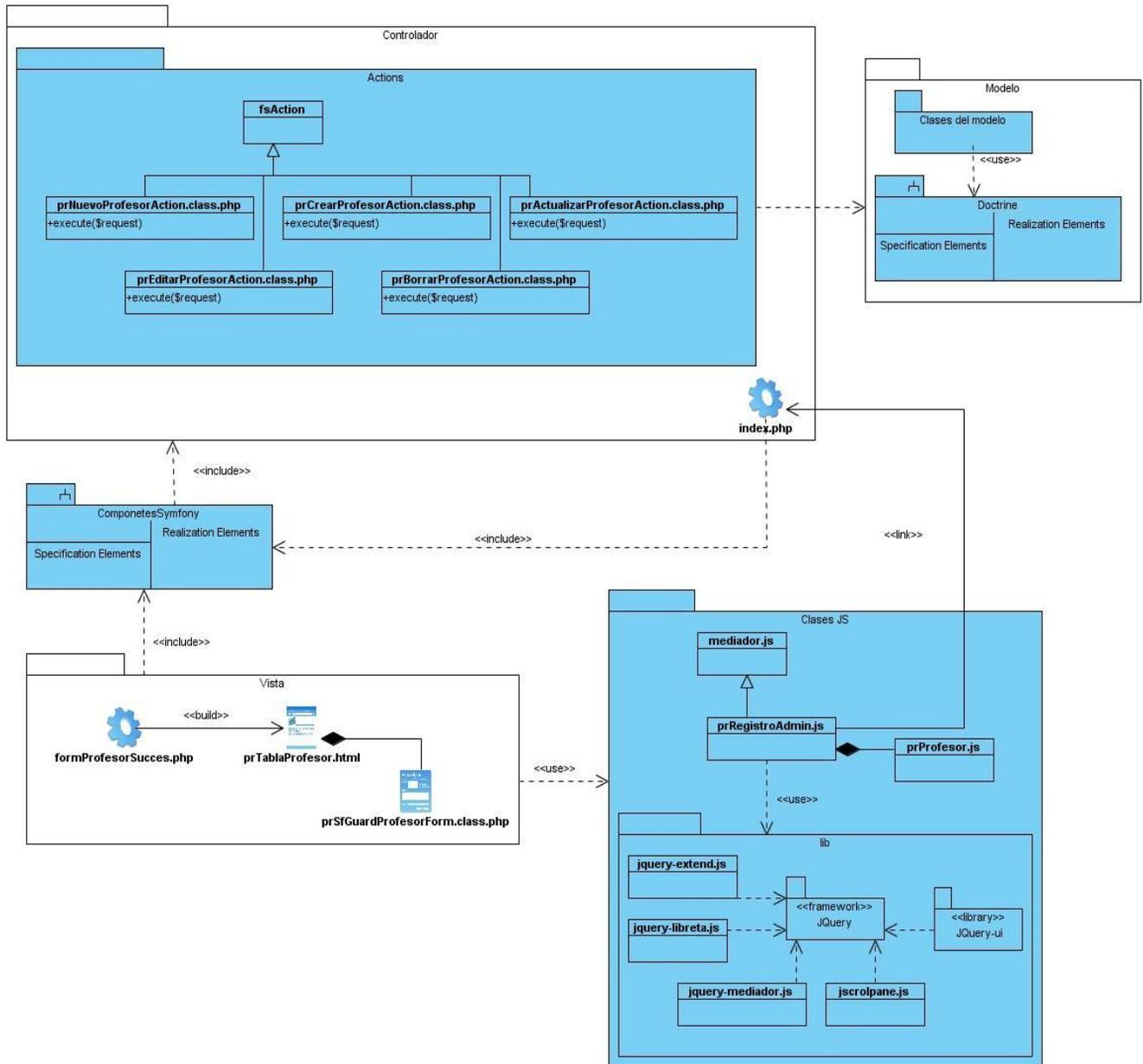


Figura 3.8. DCD_CU Gestionar Maestros

Los restantes DCD se pueden consultar en los anexos, específicamente en el [Anexo 4](#).

3.2.3. Descripción de las Clases del Diseño

Tabla 3.3. Descripción de las clases del diseño del caso de uso: Gestionar Escuela

Nombre de la clase: prNuevaEscuelaAction
Método de la clase: execute()
Descripción: este método es el encargado de mostrar los campos que debe llenar el actor para insertar una nueva escuela.

Nombre de la clase: prCrearEscuelaAction
Método de la clase: execute()
Descripción: realiza la acción de adicionar una nueva escuela.

Nombre de la clase: prBorrarEscuelaAction
Método de la clase: execute()
Descripción: elimina (o borra) una escuela.

Nombre de la clase: prEditarEscuelaAction
Método de la clase: execute()
Descripción: método encargado de mostrar los datos de una escuela y modificarlos.

Nombre de la clase: prActualizarEscuelaAction
Método de la clase: execute()
Descripción: realiza las acciones de validar y guardar los cambios realizados a los datos de una escuela.

Tabla 3.4. Descripción de las clases del diseño del caso de uso: Gestionar Estudiantes

Nombre de la clase: prBorrarEstudiantesAction
Método de la clase: execute()
Descripción: este método es el encargado de eliminar (borrar) un estudiante.

Nombre de la clase: prActualizarEstudianteAction
Método de la clase: execute()
Descripción: realiza las acciones de validar y guardar los cambios realizados a los datos de un estudiante.

Nombre de la clase: prEditarEstudianteAction
Método de la clase: execute()
Descripción: método encargado de mostrar los datos de un estudiante y modificarlos.

Nombre de la clase: prNuevoEstudianteAction
Método de la clase: execute()
Descripción: responsable de mostrar los campos que debe llenar el actor para insertar un nuevo estudiante.

Nombre de la clase: prCrearEstudianteAction
Método de la clase: execute()
Descripción: adiciona un nuevo estudiante.

Nombre de la clase: prCargarDatosEscuelaYGrupoAction
Método de la clase: execute()
Descripción: guarda los datos de la escuela y el grupo a donde pertenece un estudiante.

Nombre de la clase: prCargarCiEstudianteAction
Método de la clase: execute()
Descripción: guarda el número de carnet de identidad de un estudiante.

Tabla 3.5. Descripción de las clases de diseño del caso de uso: Gestionar Grupo Académico

Nombre de la clase: prNuevoGrupoAction
Método de la clase: execute()
Descripción: este método es el encargado de mostrar los campos que debe llenar el actor para insertar

un nuevo grupo académico.

Nombre de la clase: prCrearGrupoAction

Método de la clase: execute()

Descripción: adiciona un nuevo grupo académico.

Nombre de la clase: prBorrarGrupoAction

Método de la clase: execute()

Descripción: elimina (o borra) un grupo académico.

Nombre de la clase: prActualizarGrupoAction

Método de la clase: execute()

Descripción: se encarga de validar y guardar los cambios realizados a los datos de un grupo académico.

Nombre de la clase: prEditarGrupoAction

Método de la clase: execute()

Descripción: muestra los datos de un grupo académico y modificarlos.

Tabla 3.6. Descripción de las clases de diseño del caso de uso: Gestionar Maestros

Nombre de la clase: prNuevoProfesorAction

Método de la clase: execute()

Descripción: este método es el encargado de mostrar los campos que debe llenar el actor para insertar un nuevo profesor.

Nombre de la clase: prCrearProfesorAction

Método de la clase: execute()

Descripción: adiciona un nuevo profesor.

Nombre de la clase: prBorrarProfesorAction

Método de la clase: execute()

Descripción: realiza la acción de eliminar (borrar) un profesor.

Nombre de la clase: prActualizarProfesorAction

Método de la clase: execute()

Descripción: valida y guarda los cambios realizados a los datos de un profesor.

Nombre de la clase: prEditarProfesorAction

Método de la clase: execute()

Descripción: muestra los datos de un profesor y modificarlos.

3.3. Diagrama de clases persistentes

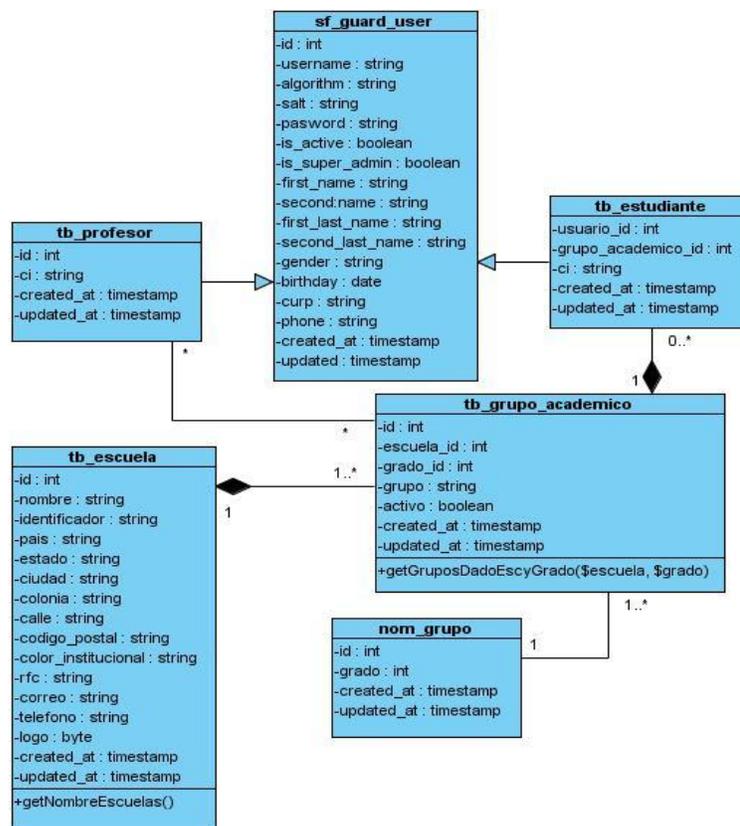


Figura 3.9. Diagrama de clases persistentes

3.4. Modelo de datos

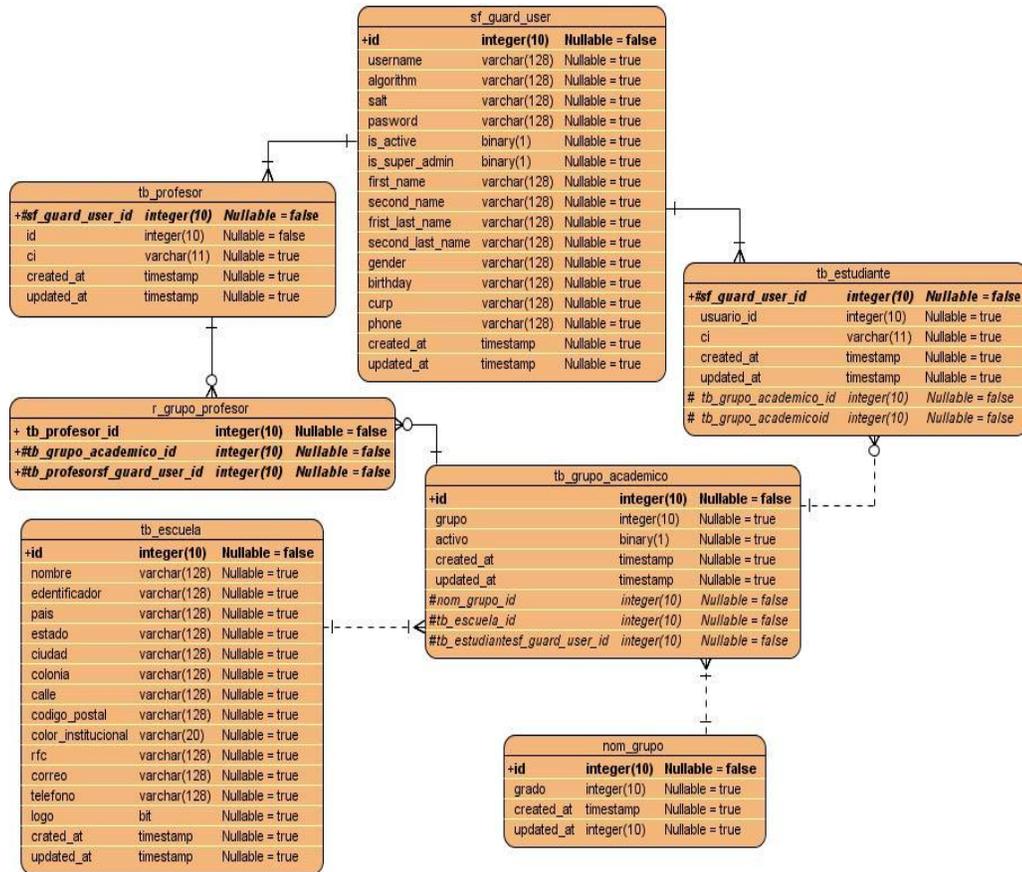


Figura 3.10. Modelo de datos

3.5. Patrones de diseño

Un patrón de diseño es considerado una descripción entre clases y objetos comunes entre sí, adaptada para resolver un problema de diseño general en un contexto particular; es una forma más práctica de describir ciertos aspectos de la organización de un programa. Un patrón de diseño es aplicable a diferentes problemas de diseño en distintas circunstancias.

Según la bibliografía consultada en el libro “Design Patterns: Elements of Reusable Object – Oriented Software”, existen tres tipos de patrones de diseño:

- Patrón de comportamiento: estos tipos de patrones, atañen a los algoritmos y a la asignación de responsabilidades entre objetos.
- Estructurales: este tipo de patrón, se ocupa de cómo clases y objetos son utilizados para

componer estructuras de mayor tamaño.

- Patrones de creación: este tipo de patrón, abstraen el proceso de creación de instancias.

Para la implementación de la solución propuesta, se hará uso de los siguientes patrones de diseño:

➤ **Patrón de diseño estructural:** Decorator (Decorador)

Decorator es un patrón de diseño estructural, cuyo objetivo principal es evitar un sinfín de clases heredadas. La idea de decorar un objeto (o envolver, como también se conoce), es agregarle nuevas funcionalidades o características al mismo. Dentro de los problemas que resuelve Decorator, se encuentran:

- Se tiene un objeto al que se le quiere aplicar una nueva característica.
- La característica se le debe aplicar dinámicamente.
- No se sabe que característica es la que se va a aplicar.

➤ **Patrón de diseño de comportamiento:** Mediador

El patrón de diseño de comportamiento Mediador, define un objeto que encapsula un conjunto de objetos. El mismo estimula la pérdida de acoplamiento ocultando las referencias explícitas entre los objetos, permitiendo así, variar su iteración de forma independiente. Es decir, se define un objeto (mediador) que se encarga de gestionar la comunicación entre distintos objetos (participantes), sin que éstos necesiten conocerse entre sí.

3.6. Conclusiones del capítulo

Una vez culminado el desarrollo del presente capítulo

- Se realizó el análisis y diseño de todos los CU del sistema en los cuales se relacionan los requisitos funcionales y no funcionales, en expresión de la programación; haciendo uso de las diferentes herramientas descritas en capítulos anteriores.
- Se preparó además, la etapa de implementación del sistema.

Capítulo 4. Implementación del sistema

Introducción

El objetivo principal del flujo de trabajo “Implementación”, es convertir los elementos del diseño en elementos de implementación. En el presente capítulo se describe y modela el sistema en términos de componentes a través de los Diagramas de despliegue y de Implementación. Se describen además, los componentes necesarios para el correcto funcionamiento de la aplicación.

4.1. Vista de despliegue

En un diagrama de despliegue, se muestran las relaciones que se establecen entre los componentes de software y hardware, representados mediante nodos estrechamente conectados.

A continuación se muestran y describen los diagramas de despliegue (DD) correspondiente al módulo propuesto. Estos pueden ser de dos variantes:

4.1.1. Variante 1 del DD

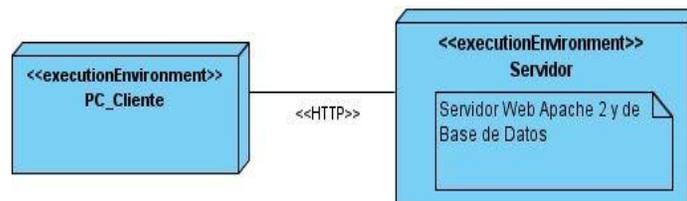


Figura 4.1. Diagrama de despliegue 1

Descripción de la funcionalidad:

- **PC_Cliente:** ordenador cliente, el cual se conecta a través de un navegador web a un servidor central donde se encuentra la aplicación.
- **Servidor:** servidor central donde se encuentra instalado un servidor web y uno de base de datos. Además, hospeda todos los componentes necesarios para garantizar un funcionamiento eficiente del software.
- **<<HTTP>>:** es un elemento de comunicación, el cual representa la conexión entre el navegador web y el servidor de aplicaciones.

4.1.2. Variante 2 del DD

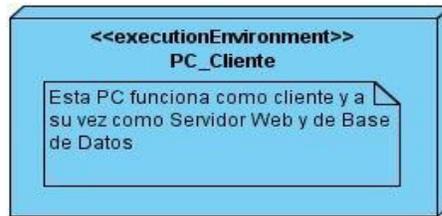


Figura 4.2. Diagrama de despliegue 2

Descripción de la funcionalidad:

- **PC_Cliente:** en esta variante, el mismo ordenador hace función de servidor y de cliente. En él, están instalados el servidor web y el de base de datos.

4.2. Vista de implementación

4.2.1. Diagrama de componentes

Un diagrama de componentes presenta un nivel más alto de abstracción que un diagrama de clases (usualmente un componente se implementa por una o más clases u objetos) en tiempo de ejecución. Los mismos muestran la relación existente entre los elementos de la implementación y se representan mediante grafos conectados por relaciones de compilación y ejecución.

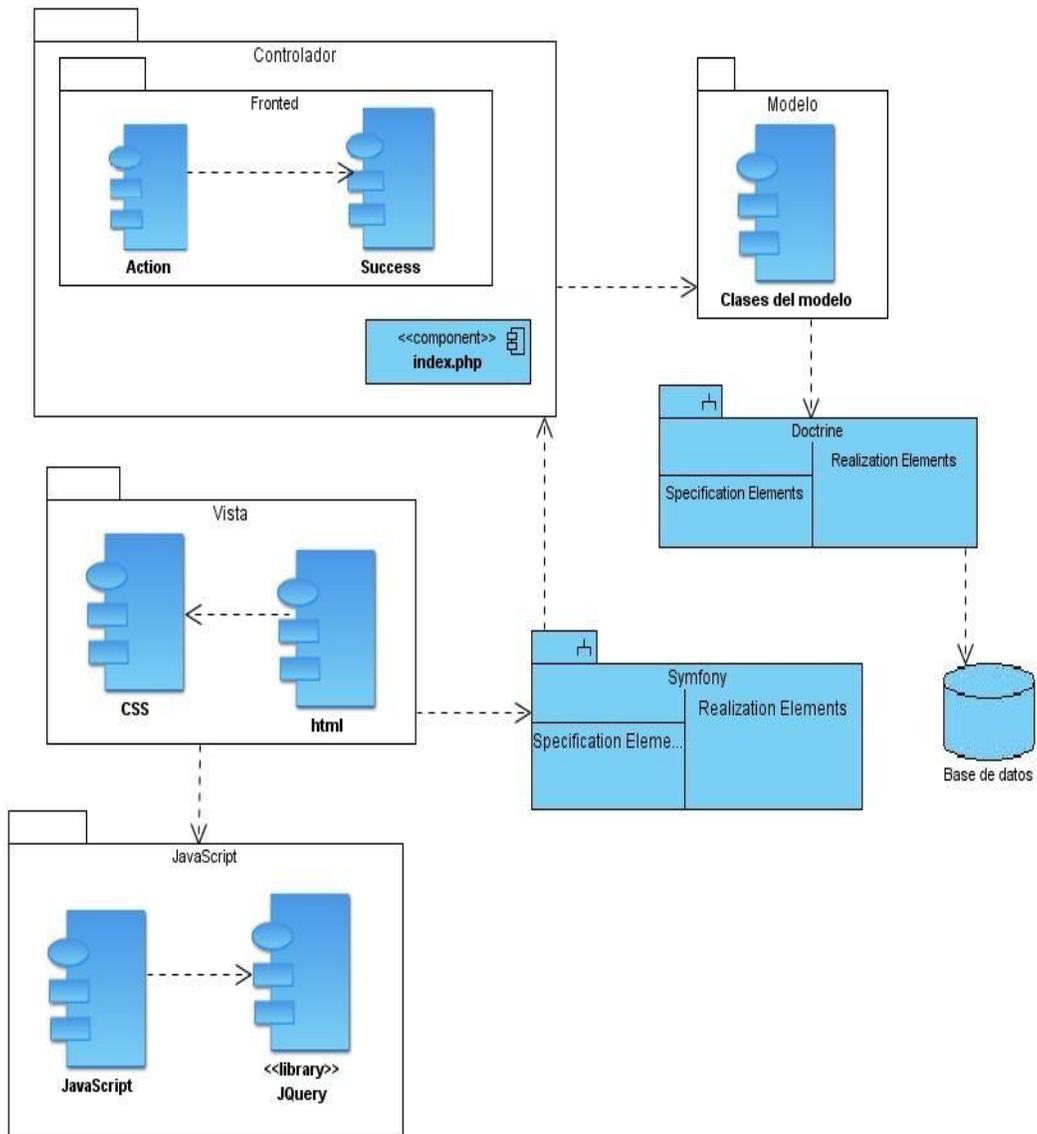


Figura 4.3. Diagrama de componentes del módulo Maestro

4.2.2. Componentes del subsistema controlador

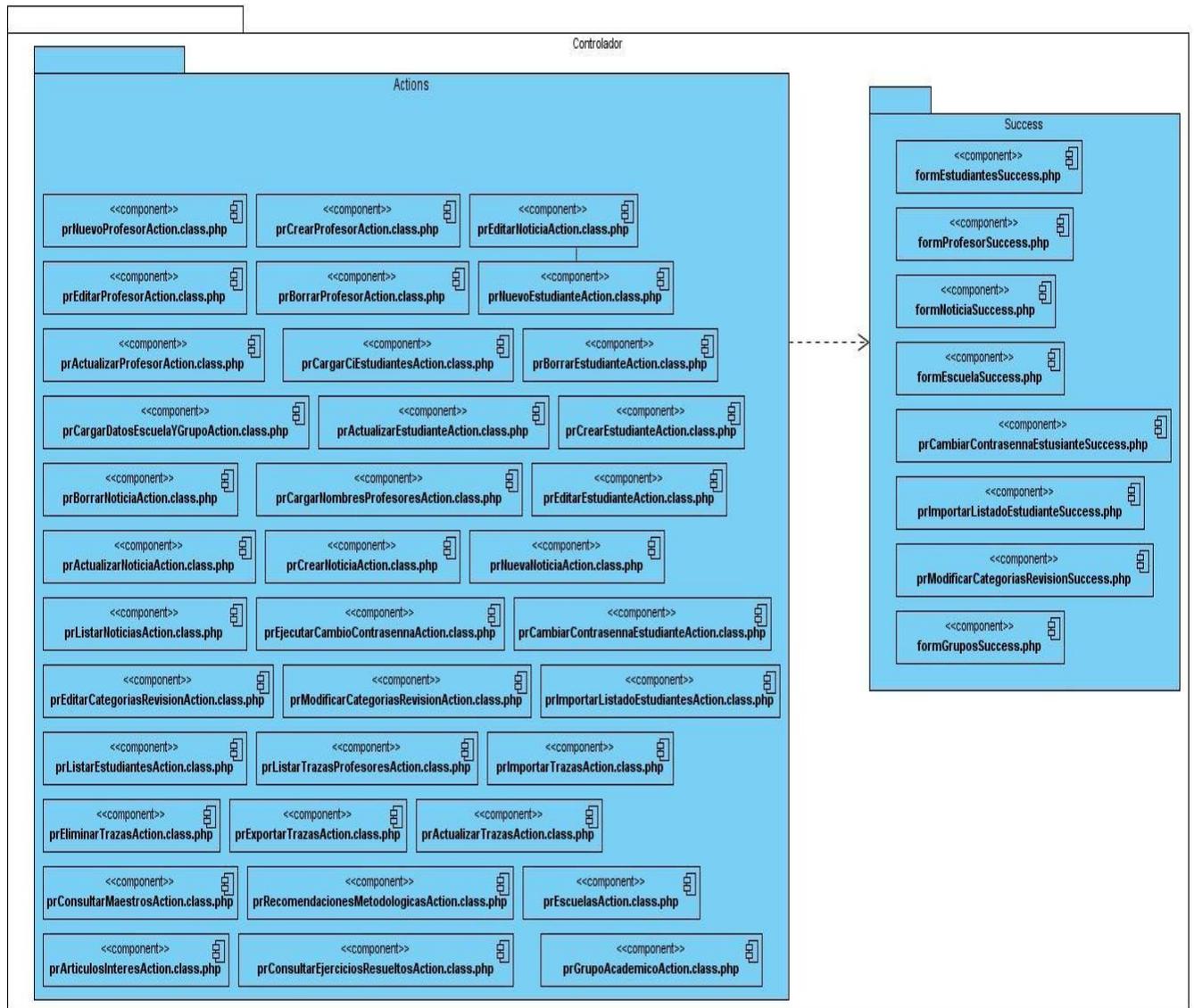


Figura 4.4. Diagrama de componentes del subsistema “controlador”

4.2.3. Componentes del subsistema modelo

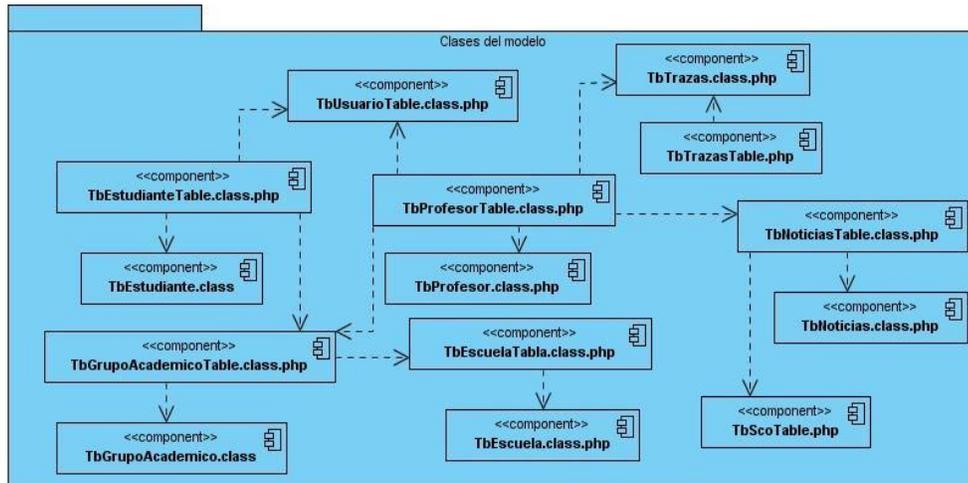


Figura 4.5. Diagrama de componentes del subsistema “modelo”

4.2.4. Componentes del subsistema vista

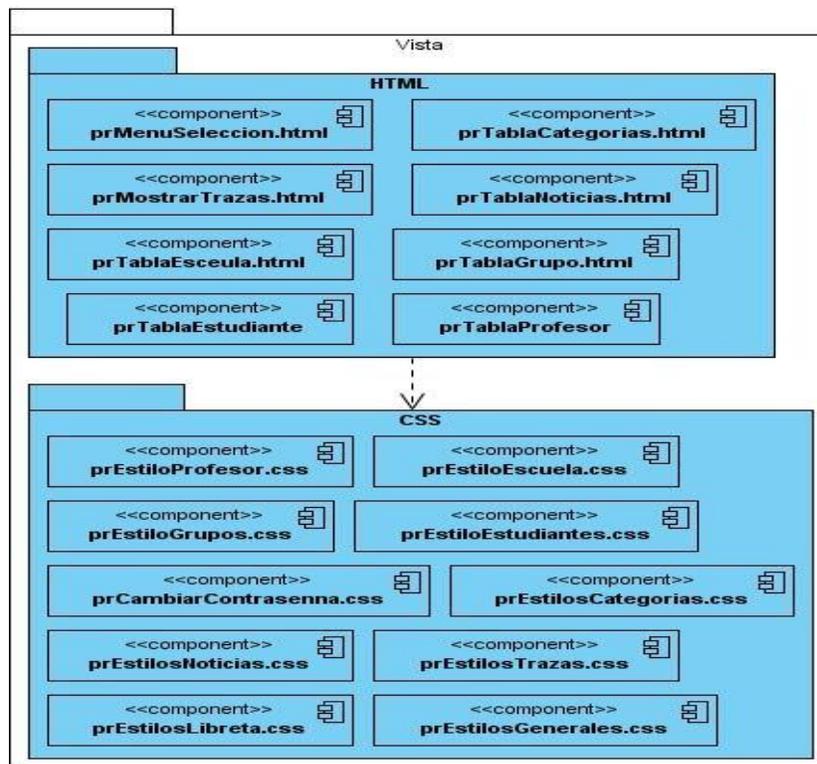


Figura 4.6. Diagrama de componentes del subsistema “vista”

4.2.5. Componentes del paquete JavaScript

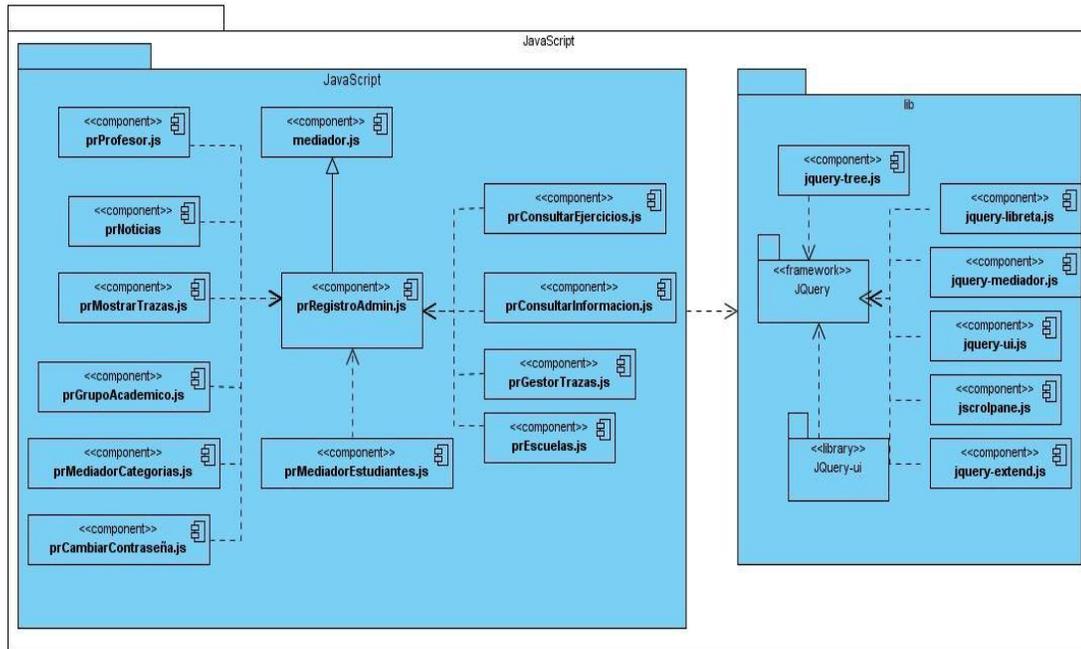


Figura 4.7. Diagrama de componentes del paquete JavaScript

4.3. Interfaz de la aplicación

Una interfaz es la parte que le permite a un usuario interactuar con una aplicación. El correcto diseño de interfaces de usuario, se ha convertido en una labor de suma importancia en el desarrollo de un software, ya que, estas permiten determinar el éxito o fracaso de un sistema, por este motivo se convierte en uno de los aspectos más significativos en la usabilidad de una aplicación. Siguiendo las reglas de una interfaz amigable, sencilla y agradable para el usuario, fueron obtenidas las siguientes interfaces del módulo:

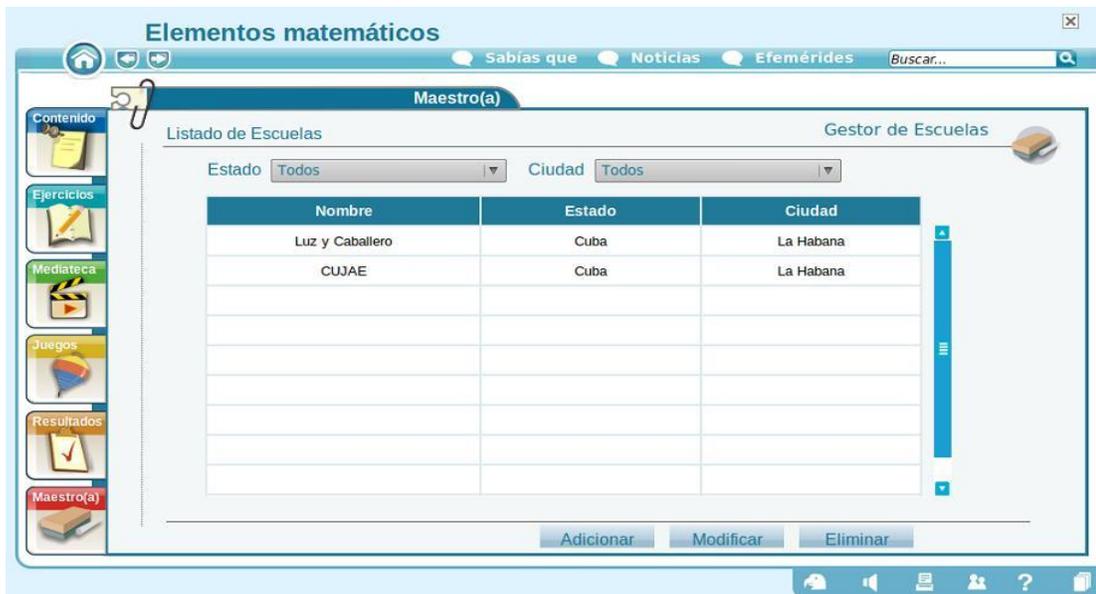


Figura 4.8. Interfaz para listar los datos de las escuelas



Figura 4.9. Interfaz para adicionar una escuela



Figura 4.10. Interfaz para modificar los datos de una escuela

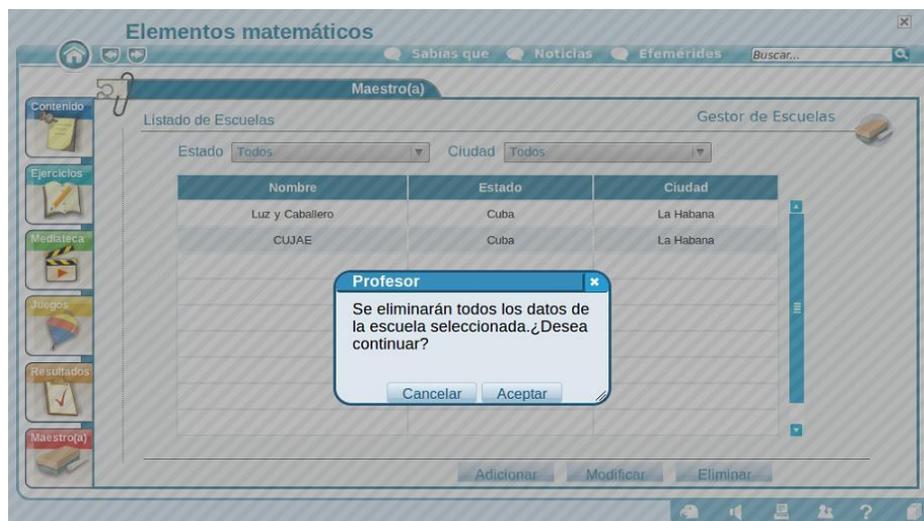


Figura 4.11. Interfaz para eliminar los datos de una escuela



Figura 4.12. Interfaz para listar los datos de los estudiantes

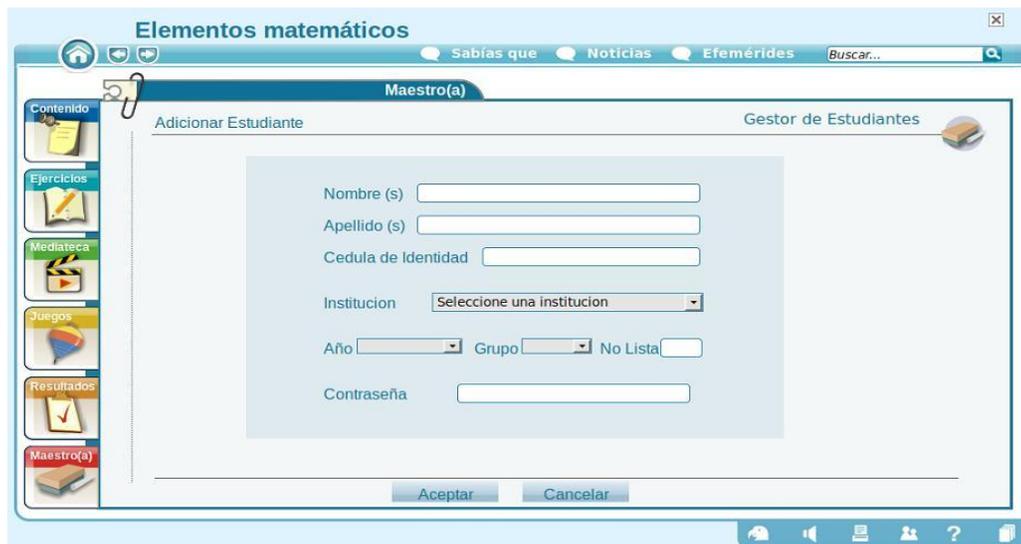


Figura 4.13. Interfaz para adicionar un estudiante



Figura 4.14. Interfaz para modificar los datos de un estudiante

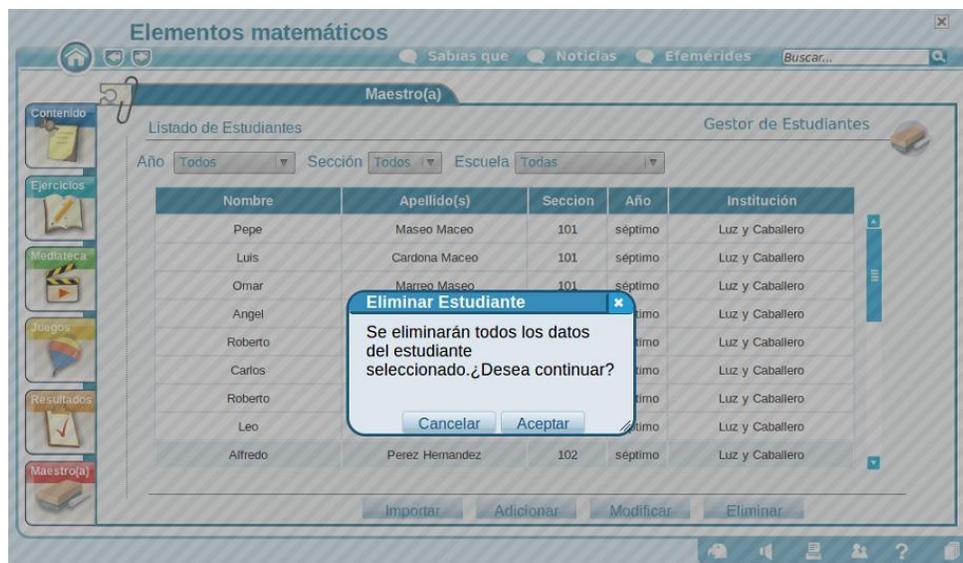


Figura 4.15. Interfaz para eliminar los datos de un estudiante



Figura 4.16. Interfaz para listar los datos de los maestros



Figura 4.17. Interfaz para adicionar maestros



Figura 4.18. Interfaz para modificar los datos de un maestro

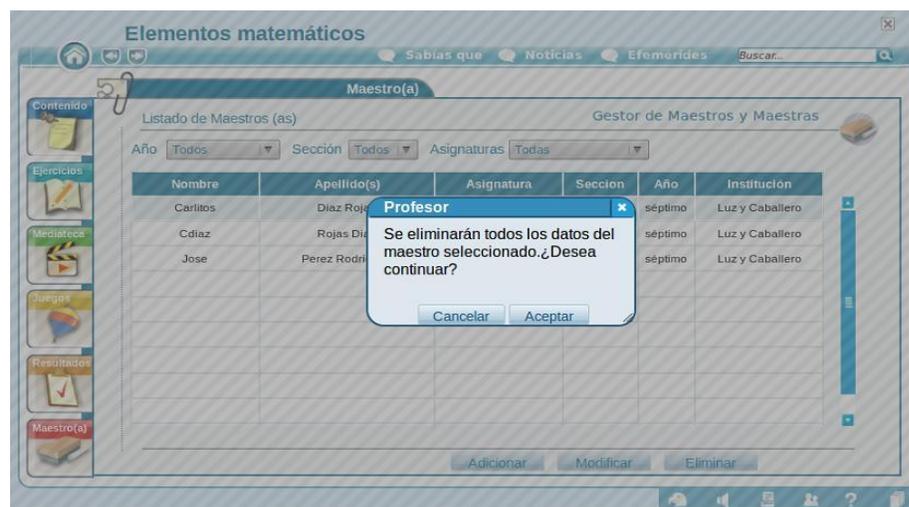


Figura 4.19. Interfaz para eliminar los datos de un maestro

Las restantes interfaces de usuario se pueden consultar en los anexos, específicamente en el [Anexo 5](#).

4.4. Conclusiones del capítulo

- Se conformaron los Diagramas de Componentes (DC) y Despliegue (DD). El DD se utilizó para modelar el hardware utilizado en la implementación de la aplicación y las relaciones entre sus componentes.
- Mediante los DC se muestra como está estructurado el sistema, así como, las dependencias que existen entre los componentes que lo conforman.

Conclusiones Generales

Con el desarrollo del presenta trabajo de diploma:

- Existen diferentes aplicaciones de software educativo en los cuales se encuentra un módulo a través del cual los profesores pueden, entre otras actividades, administrar toda la información referente a los usuarios que interactúan con la aplicación, pero ninguno cumple con las políticas de software educativo aplicadas en el desarrollo del módulo propuesto, y no satisfacen las necesidades del cliente. Por estas razones, se desarrolló el módulo propuesto que lleva por nombre: Módulo “Maestro” de la colección El navegante en su versión multiplataforma, siguiendo las funcionalidades especificadas anteriormente.
- A través de las fases que propone la metodología de desarrollo RUP, se evolucionó en la Concepción y Elaboración del módulo “Maestro” perteneciente a la colección El Navegante en su versión multiplataforma. Se generaron los artefactos fundamentales propuestos por dicha metodología, obteniendo la documentación necesaria para garantizar el futuro mantenimiento del sistema.
- Se obtuvo, toda la documentación del proceso de desarrollo de dicho módulo garantizando así el futuro mantenimiento del mismo y que sirva además de base para futuras versiones de la colección.

Por lo antes planteado, se puede concluir que se logró el cumplimiento satisfactorio de los objetivos planteados.

Recomendaciones

Una vez culminado el presente trabajo, se le recomienda al equipo de desarrollo del proyecto Multisaber y El Navegante:

- Continuar desarrollando la aplicación, brindándole soporte a futuras funcionalidades que pueden surgir.
- Realizar pruebas de unidad a la aplicación obtenida.
- Integrar la funcionalidad: “Visor de ejercicios”, una vez que esté terminada la implementación del módulo Ejercicios de la colección El Navegante en su versión multiplataforma.

Referencias Bibliográficas

1. Ms C Labañino, César. *Maestría Ciencias de la Educación*. La Habana : s.n, 2006.
2. Dr. Pere Marqués, Graells. Multimedia Educativo: Clasificación, funciones, ventajas, diseño de actividades. [En línea] Departamento de Pedagogía Aplicada, Facultad de Educación, 3 de 08 de 2010. [Citado el: 15 de 10 de 2010.] <http://www.peremarques.net/funcion.htm>
3. Multimedia educativa. Angel Fidalgo Blanco, Laboratorio de Innovación en Tecnologías de la Información. Departamento de Matemática aplicada y Métodos Informáticos. Univerisdad Politécnica de Madrid. [Citado el: 15 de 10 de 2010.] <http://innovacioneducativa.files.wordpress.com/2009/10/multimedia-educativa2.pdf>
4. Multimedia Educativo: clasificación, funciones, ventajas, diseño de actividades. Dr. Pere Marqués Graells. Departamento de Pedagogía Aplicada. Facultad de Educación UAB <http://www.peremarques.net/funcion.htm#ventajas>
5. e-cuaderno MyClass. [Citado el 15 de 10 de 2010 <http://www.my-class.es/software-cuaderno-de-profesor-ventajas-myclass.html>]
6. MSc. Carlos Martín Álvarez, MSc. Pedro Miguel Milián Vázquez, MSc. Oscar José Alejo Machado, Dr. C. María Magdalena López Rodríguez del Rey. Una herramienta informática para la superación profesional en Farmacología de los profesores de medicina en Cuba. [En línea] Universidad de Ciencias Médicas de Cienfuegos [Citado el 15 de 10 de 2010] <http://www.odiseo.com.mx/bitacora-educativa/2011/02/herramienta-informatica-para-superacion-profesional-farmacologia-profesor>
7. Agtica Edusoft. México [Citado el 15 de 10 de 2010] <http://www.netmedia.info/soluciones/software/crean-software-gestion-escuelas/>
8. Agtica Edusoft. México [Citado el 15 de 10 de 2010]. <http://www.sg.com.mx/guia/node/1212>
9. Colección de Osftware Educativos, Multisaber. [En línea] [Citado el: 15 de 10 de 2010.] <http://blogs.rimed.cu/multisaber/>.
10. PORTAL EDUCATIVO CUBANO, Ministerio de la Educación de la República de Cuba. [En línea] [Citado el 15 de 10de 2010] http://www.cubaeduca.rimed.cu/index.php?option=com_content&view=article&id=4700&Itemid=153
11. Entorno virtula de aprendizaje. [En línea] [Citado el: 15 de 10 de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=34099> .

12. Kioskea.net. [En línea] [Citado el: 16 de 12 de 2010.] <http://es.kioskea.net/contents/css/cssintro.php3> .
13. Maestros de la web. [En línea] [Citado el: 16 de 12 de 2010.] <http://www.maestrosdelweb.com/editorial/phpintro/> .
14. Booch, G.; Rumbaugh y J. y, Jacobson, I. *El lenguaje Unificado de Modelado*. s.l. : s,n, 2000.
15. Maestros de la Web. [En línea] [Citado el 16 de 12 de 2010] <http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/> .
16. Symfony.es. [En línea] [Citado el: 16 de 12 de 2010.] <http://www.symfony.es/10-razones-para-utilizar-symfony/> .
17. [En línea] [Citado el: 18 de 12 de 2010] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/ .
18. librosweb.es. [En línea] http://www.librosweb.es/symfony_1_2/capitulo2/el_patron_mvc.html
19. netBeans. [En línea] [Citado el: 18 de 12 de 2010.] <http://netbeans.org/community/releases/69/>.
20. sildeshare.net. [En línea] <http://www.slideshare.net/alexmerono/sistemas-gestores-de-bases-de-datos>.
21. postgresq. [En línea] [Citado el: 18 de 12 de 2010.] <http://www.postgresql.org/docs/8.4/interactive/intro-what-is.html> [En línea] .