



Universidad de las Ciencias Informáticas
Facultad 4

“Desarrollo de la versión 2.0 del módulo Resultados de la colección de software El Navegante en su versión multiplataforma”

**TRABAJO DE DIPLOMA PARA OPTAR AL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autores:

Claudia Murillo Díaz

Jorge Martínez García

Tutores:

Ing. Osdalme Fuentes Colina

Ing. Angel Alberto Vázquez Sanchez

La Habana, junio de 2011

“Año 53 de la Revolución”

Módulo Resultados

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ___ días del mes de junio del año 2011.

Claudia Murillo Díaz

Autor

Jorge Martínez García

Autor

Ing. Osdalme Fuentes Colina

Tutor

Ing. Angel Alberto Vázquez Sanchez

Tutor

Módulo Resultados

Es de importancia para quien desee alcanzar una certeza en su investigación, el saber dudar a tiempo.

Aristóteles

Agradecimientos

De Claudia

A mis padres por ser un digno ejemplo a seguir, por enseñarme a ser una mejor persona y saber caminar por la vida sin tener que poner un pie en falso, a mi mamá por ser tan especial y siempre estar ahí y a mi papá por darme tantas lecciones de la vida. Gracias a los dos por estar siempre a mi lado, por permitirme ser una lucecita para sus corazones pero sobre todas las cosas por depositar toda su confianza en mí.

A mi hermana Glenda, que la he visto crecer y ya hoy es casi una profesional, por su comprensión, confianza y apoyo incondicional. A toda mi familia, mis abuelos Mami y Papi que siempre me han querido como si fuese su propia hija, siempre los tengo en mi mente. A mis tíos Idania y Boris los quiero mucho.

A mi novio, por animarme desde el día en que lo conocí, con su cariño, amor y optimismo. Porque sobre el recayeron mis peores momentos, por aguantarme cuando ya ni yo lo podía hacer. Por su ayuda tan fuerte en cada una de las etapas de la tesis. Porque muy valioso y espero poder retribuirte a lo largo de nuestra vida todo lo que me diste, todo lo que me das...

Al mejor grupo que he tenido en toda la vida, el 8106, por compartir conmigo alegrías y tristezas. Por convertirse en mi familia.

A Osdalme por su paciencia y esfuerzo incondicional, desde que nos conocimos en el viaje a Venezuela. Gracias por confiar en mí.

A todos mis amigos en general por preocuparse y ayudarme de una forma u otra en estos 5 años de universidad, nunca los olvidaré.

Agradecimientos

De Jorge

A mi madre, por dar todo de sí para que nunca me faltara nada y poder ser un hombre de bien, por ser ejemplo de madre, de amiga, de mujer y de abuela, por apoyarme en los momentos más difíciles, por ser ese escudo que me protege a diario, por ser una guerrera en tiempos difíciles, por hacerme sentir feliz en los momentos más tristes, por ser la luz que me ilumina.

A mi hombrecito, por ser mi mayor fuente de inspiración, por hacerme feliz día a día con su sonrisa y gritar con sus mordidas, por ser mi remedio ante cualquier enfermedad, por ser el motor impulsor que me permite seguir adelante, por darme fuerzas cuando no las tengo, por recordarme que no hay nada más lindo en el mundo que estar junto a las personas que amas.

A mi novia, por estar a mi lado estos años a pesar de las dificultades, por traer al mundo ese ser tan lindo que es nuestro hijo, por ser una buena madre a pesar de tan corta edad y poca experiencia, por amarme.

A mis tutores, Osdalme y Angel por su gran ayuda, dedicación e incondicionalidad durante la realización de la tesis.

A los grupos 8103 y 8305 especialmente por ayudarme y brindarme su apoyo en momentos difíciles, a cada uno de sus integrantes, los llevare siempre conmigo.

A Lesdie, Yureydis, Yoanis, Miguel, Kenier, Yolaida, Yamek, Aylén, Daymis, Yuleysis, Roberto y muchos más, excelentes compañeros de grupo y amigos incondicionales.

A mis tantas amistades profesores y estudiantes que me han ayudado, apoyado y enseñado durante estos 5 años, a los cuales les debo mucho y estoy eternamente agradecido.

A Mayliuvis y Yudanis, por ser mis amigos y de mi familia, por sus detalles con mi hijo y por toda la ayuda que me han brindado de manera incondicional.

Módulo Resultados

Agradecimientos

Al profe Dosagües, por acogerme con los brazos abiertos en 2do año cuando ingresé al Proyecto Historia Universal.

A Deiler, por ayudarme y apoyarme en un momento muy difícil, por ofrecerme su amistad.

A Yunior, por fajarnos tanto y al finar aprender de sus virtudes, por ser compañeros de aula y de trabajo, por los resultados obtenidos durante nuestra carrera.

A Adrián, por su ayuda incondicional cada vez que la necesité.

A Yudislandry, por recibirme en el Comité Primario, por ayudarme cuando lo necesité.

A Yusdel, por toda la ayuda y apoyo brindado en tan poco tiempo.

Al colectivo de “El Navegante” por permitirnos ser parte de él, especialmente a Ramiro por trabajar de forma directa en la realización del módulo.

A Jorge Antonio por motivarme durante mi primer curso de Flash, el cual me permitió obtener buenos resultados y despertar en mí la semilla del querer aprender más y más.

A la Universidad y en particular a la Facultad por ser mi casa durante estos 5 años, por permitir que me formara como persona y profesional, por darme la posibilidad de conocer tanta gente buena.

A todos los que de una forma u otra me ayudaron y contribuyeron a que hoy pueda ser una mejor persona.

Módulo Resultados

Dedicatoria

Con todo el amor del mundo a mi familia y en especial a mis padres que siempre han sido mi inspiración y apoyo para hacer este sueño realidad, brindándome su cariño y confianza incondicional.

Claudia

A mi madre por su esfuerzo, sacrificio y apoyo incondicional, a mi novia por estar a mi lado y traer al mundo la persona más especial de mi vida, a mi hijo que desde su nacimiento ha sido mi fuente de motivación e inspiración.

Jorge

Módulo Resultados

Resumen

El desarrollo de software educativo en Cuba como medio de enseñanza – aprendizaje se ha visto incrementado en los últimos años producto a los avances de las Tecnologías de la Información y las Comunicaciones. En la X Mixta Cuba-Venezuela fue firmado el proyecto “Colecciones de Software Educativo Multisaber-El Navegante” con el objetivo de desarrollar una aplicación orientada a la enseñanza secundaria, multiplataforma y contextualizada a Venezuela.

Con el presente trabajo se persigue el desarrollo del módulo Resultados de la colección El Navegante, que aportará funcionalidades para llevar a cabo un control de la trayectoria de los estudiantes durante su interacción con los restantes módulos de la aplicación. Con este fin se realizó un estudio exhaustivo de las soluciones similares existentes, metodologías, estándares y herramientas, así como de las tendencias y tecnologías actuales. Este análisis permitió seleccionar la metodología RUP para apoyar el proceso de desarrollo de software. Seguidamente se transitó por los flujos de trabajo de RUP: modelo del negocio, requisitos, análisis, diseño, implementación y pruebas; obteniéndose cada uno de los artefactos propuestos.

A través del desarrollo del presente trabajo se obtuvo un módulo Resultados, que brinda la posibilidad de acceder a las evaluaciones obtenidas por lo estudiantes en la aplicación y permite mostrar la evolución de los mismos en las asignaturas y contenidos especificados. De esta manera el profesor puede contar con una herramienta capaz de consultar las evaluaciones y de mostrar el recorrido de sus estudiantes.

Módulo Resultados

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica.....	5
1.1 Introducción	5
1.2 Análisis de soluciones existentes	5
1.3 Estudio de la metodología para el desarrollo de software.....	8
1.4 Tecnologías y herramientas a utilizar	11
1.4.1 Lenguaje de modelado.....	11
1.4.2 Herramienta Case	12
1.4.3 Lenguajes de desarrollo.....	12
1.4.4 Frameworks (Plataformas de desarrollo)	14
1.4.5 Servidor web	16
1.4.6 Sistema de gestión de bases de datos.....	17
1.4.7 Entorno de desarrollo integrado (IDE)	19
1.5 Patrón arquitectónico Modelo-Vista-Controlador.....	20
1.6 Conclusiones	21
Capítulo 2: Descripción del sistema.	22
2.1 Introducción	22
2.2 Modelo de dominio	22
2.2.1 Análisis de los conceptos del dominio	22
2.3 Descripción del sistema propuesto.....	24
2.4 Especificación de requerimientos	25
2.5 Definición de los casos de uso	28
2.5.1 Actores del sistema	28

Módulo Resultados

2.5.2 Diagrama de casos de uso del sistema.....	29
2.5.3 Descripción de los casos de uso del sistema	29
2.6 Conclusiones	33
Capítulo 3: Análisis y Diseño	34
3.1 Introducción	34
3.2 Modelo de análisis	34
3.3 Patrones de diseño.....	38
3.3.1 Patrón Mediator.....	38
3.3.2 Patrón Controlador frontal	38
3.4 Modelo de diseño	39
3.5 Diseño de la base de datos	46
3.5.1 Modelo Entidad-Relación.....	46
3.6 Conclusiones	47
Capítulo 4: Implementación y Pruebas.....	49
4.1 Introducción	49
4.2 Estándar de codificación.....	49
4.2.1 Convenciones de nombres (Funciones, Variables Globales, Clases).....	49
4.2.2 Indentación y espacios en blanco	50
4.2.3 Comentarios.....	50
4.3 Modelo de implementación.....	51
4.3.1 Diagrama de despliegue	51
4.3.2 Diagrama de componentes	52
4.4 Pruebas de software.....	57
4.4.1 Pruebas de caja negra	58

Módulo Resultados

4.5 Conclusiones	60
Conclusiones generales	61
Recomendaciones	62
Referencias bibliográficas.....	63
Bibliografía.....	65
Glosario de términos.....	66

Introducción

En la actualidad se evidencia un ritmo acelerado en el avance tecnológico, estrechamente relacionado con el impacto producido por el empleo de las Tecnologías de la Información y las Comunicaciones (TIC) que se ha transformado en el medio común del camino de la vida social. Los avances tecnológicos han revolucionado el modo de vivir de los seres humanos, pues se encuentra fuertemente relacionado con los grandes adelantos que se conciben en forma de instrumentos y se insertan en la sociedad, originando un mayor nivel y calidad de vida. Como consecuencia a dicho desarrollo, las empresas, organizaciones y gobiernos le prestan un gran interés, centrado sus objetivos con fines de explotar las facilidades que ofrece para su desempeño.

El auge de las TIC ha transformado totalmente el papel social de la información y del aprendizaje. En Cuba, a pesar del bloqueo económico, las Tecnologías de Informática y Comunicaciones se están convirtiendo en una esfera esencial de las estrategias cubanas para el avance socioeconómico de la Revolución. Debido a los beneficios que brindan, se evidencia un ascenso en su utilización en la sociedad, la cultura, la educación, la medicina y la economía cubana, con la cual interactúan gran cantidad de profesionales influyendo en la población en general. El estado revolucionario cubano, es reconocido a nivel mundial como insignia indiscutible en la esfera de la educación, ha enfatizado en el proceso de informatización de la sociedad, el cual se sitúa en orientar todo el desarrollo tecnológico al alcance del pueblo. Dentro del proceso de informatización de la sociedad cubana, la dirección del Ministerio de Educación (MINED) conformó grupos de docentes con motivaciones informáticas para elaborar software educativo, con el objetivo de elevar la calidad del proceso de enseñanza-aprendizaje en el país.

El Gobierno Bolivariano, por su proceso revolucionario, se encuentra inmerso en un proceso constante de cambios en la esfera de la educación, para ello es sumamente importante la introducción de las TIC en los centros educacionales. Como parte del convenio integral de cooperación Cuba-Venezuela en la X Mixta, se realizó un contrato para el desarrollo del proyecto “Colecciones de Software Educativo Multisaber y El Navegante para planteles de los niveles de Primaria y Secundaria del Sistema Educativo” cuyo objetivo es el desarrollo de las colecciones Multisaber y El Navegante para su instalación en las escuelas venezolanas. Estas colecciones consisten en el desarrollo de 24 productos basados en las colecciones educativas, “Multisaber” con 14 y “El Navegante” con 10, ambas utilizadas actualmente en las escuelas

Módulo Resultados

cubanas. La facultad 4 que es reconocida en la UCI por su experiencia y excelencia en el desarrollo de software educativo, se le asignó la tarea de participar en este proyecto, con el objetivo de crear varias colecciones para distintos niveles educativos.

La colección “El Navegante” en su versión multiplataforma, está orientada a la enseñanza secundaria conformada por diez productos especializados varias materias. Cada uno de estos productos está integrado por seis módulos, que brindan la facilidad de realizar diversas acciones mezclando diferentes elementos que representan las disímiles tipologías de software educativo. El módulo Resultados es sin dudas de gran importancia, ya que brinda la información necesaria para medir el avance de los estudiantes y a la vez puede ser utilizado por los profesores para llevar a cabo el proceso de enseñanza-aprendizaje. Este módulo es el encargado de gestionar todo lo referente a los reportes de los resultados obtenidos por los estudiantes, mostrando una serie de datos e informaciones referentes a cómo se ha comportado el proceso de aprendizaje, a partir de la interacción de los alumnos con cada uno de los restantes módulos de la Colección.

Durante el curso académico 2009-2010, se desarrollaron varios trabajos de diploma con el objetivo de obtener un software desarrollado con herramientas libres, multiplataforma y contextualizado a Venezuela. El desarrollo del módulo Resultados fue el principal objetivo de uno de estos trabajos de diploma, cuyo fin era permitir la visualización de las evaluaciones realizadas por los estudiantes en las actividades de los diferentes módulos de la Colección.

En el año 2010 se realizó un nuevo levantamiento de requisitos, donde el módulo Resultados concebido en un principio no contaba con las nuevas funcionalidades requeridas por el Ministerio del Poder Popular para la Educación (MPPE) de la República Bolivariana de Venezuela. Esta primera versión no presenta un conjunto de reportes a través de los cuales se puedan realizar análisis estadísticos utilizando gráficas de efectividad, a partir de los resultados de los estudiantes en las distintas materias, además no ofrece opciones de búsqueda y filtrado para conocer los datos personales de los estudiantes. Se hace necesario entonces rehacer el software existente, de manera que satisfaga los nuevos requerimientos.

Partiendo de lo anteriormente planteado, unido a la necesidad de migrar la colección El Navegante a software libre, es preciso declarar el siguiente problema científico: ¿Cómo desarrollar un módulo Resultados con herramientas libres, multiplataforma, que cumpla con los requisitos del Ministerio del

Módulo Resultados

Poder Popular para la Educación, facilitando al profesor el control sobre todas las acciones realizadas por los estudiantes?

Del problema a resolver expresado anteriormente, se deriva el objeto de estudio: proceso de producción de software educativo, teniendo como campo de acción: módulo Resultados de la colección El Navegante en su versión multiplataforma.

Por tal razón se plantea como objetivo general para este trabajo de diploma desarrollar la versión 2.0 del módulo Resultados de la colección de software “El Navegante”, con herramientas libres, multiplataforma, que cumpla con los requisitos del Ministerio del Poder Popular para la Educación, facilitando al profesor el control sobre todas las acciones realizadas por los estudiantes.

Para dar cumplimiento al objetivo general se definen como objetivos específicos:

- Elaborar el marco teórico de la investigación.
- Realizar el análisis y diseño de la nueva versión del módulo Resultados, teniendo en cuenta los requisitos del cliente.
- Desarrollar la nueva versión del módulo Resultados, sobre la base del software desarrollado durante el curso anterior.
- Validar que el software cumpla con los requisitos especificados.

Como idea a defender se expone, que si se desarrolla la versión 2.0 del módulo Resultados perteneciente a la colección El Navegante, con herramientas libres, multiplataforma, que cumpla con los requisitos del Ministerio del Poder Popular para la Educación, se podrá contar con una herramienta que le facilite al profesor llevar un control debidamente organizado sobre todas las acciones realizadas por los estudiantes en los módulos de cada uno de los productos de la colección El Navegante.

Las tareas que se proponen para dar solución a los objetivos anteriormente planteados son:

1. Levantamiento del estado del arte.
2. Análisis de la versión anterior del módulo Resultados.
3. Análisis de los requisitos funcionales y no funcionales especificados por el MPPE para el módulo Resultados.

Módulo Resultados

4. Selección y fundamentación de las herramientas y tecnologías que serán utilizadas durante el desarrollo de la nueva versión del módulo Resultados.
5. Identificación de los cambios necesarios a la estructura de la base de datos.
6. Identificación de las clases necesarias para el funcionamiento de la nueva versión del módulo.
7. Implementación de las clases identificadas para el funcionamiento de la nueva versión del módulo.
8. Comprobación del funcionamiento del módulo dentro de la línea base del producto general.
9. Validación y cumplimiento de los requisitos especificados por el MPPE.

Estructura capitular

Capítulo 1 Fundamentación teórica: En este capítulo se abordan los elementos teóricos más importantes que sirven de base para la realización de toda la investigación en general, tales como: los sistemas similares existentes vinculados al campo de acción, además de una descripción y selección de las herramientas a utilizar.

Capítulo 2 Propuesta de la solución: En este capítulo se identifican y describen los requisitos funcionales y no funcionales, a la vez que se presentan los actores del sistema y diagrama de casos de uso del mismo, acompañado de la descripción textual de cada uno de estos.

Capítulo 3 Análisis y diseño del sistema: Este capítulo contiene una descripción de cómo deberá ser realizado el módulo partiendo de lo descrito en el capítulo 2. Además, se presentan los diagramas de clases del análisis y los diagramas de clases del diseño, que reflejan de una forma más clara cómo va a funcionar el módulo y qué clases están presentes en el mismo.

Capítulo 4 Implementación y pruebas: En este capítulo se describen los aspectos relacionados con la construcción de la solución propuesta a través de los diagramas de componentes y el diagrama de despliegue. Se definen los casos de prueba con los que se comprobará la funcionalidad del módulo.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se describen los elementos fundamentales que constituyen la base teórica de la investigación. Se realiza un análisis del estado del arte mediante un estudio de los sistemas existentes en la UCI y a nivel nacional con funcionalidades que muestren los resultados de la interacción de los usuarios con las aplicaciones. Además, se realiza una descripción de las herramientas y metodologías usadas para la creación de la nueva versión del módulo Resultados de la colección El Navegante en su versión multiplataforma.

1.2 Análisis de soluciones existentes

El software educativo constituye una herramienta indispensable para el desarrollo de los procesos de enseñanza, ya que brinda la posibilidad de tratar diferentes formas de aprendizaje muy diversas. Muchas de estas aplicaciones tienen integrado un módulo capaz registrar las trazas de los contenidos presentes en el software educativo, facilitándole al profesor un control de las actividades y las evaluaciones realizadas por los estudiantes, entre ellos se encuentran:

Colección Multisaber (Versión instalada en las escuelas primarias cubanas)

En las escuelas primarias cubanas se encuentra instalada la colección Multisaber desarrollada con herramientas propietarias. El módulo Resultados forma parte de esta colección y presenta una interfaz que permite explorar los resultados de las interactividades de los estudiantes con el software, plasmándose todos los detalles de la interacción de los mismos en cada una de las actividades que realizan. Este análisis posibilita que el profesor realice valoraciones de la trayectoria alcanzada por los estudiantes, lo que contribuye al enriquecimiento del diagnóstico de los escolares. Este módulo brinda la facilidad de que los estudiantes controlen los resultados de sus actividades y determinen los logros alcanzados, así como las deficiencias que presentan y cómo corregirlas; es decir, a tomar conciencia de sus propias acciones, trazarse nuevas metas y por tanto, adjudicarse un rol protagónico en su proceso de aprendizaje.(1)

Módulo Resultados

Colección El Navegante (Versión instalada en las escuelas secundarias cubanas)

La colección El Navegante constituye un sistema informático basado en tecnología hipertexto que contiene un conjunto de elementos representativos de diversas tipologías de software educativo. Esta colección está compuesta por diez software que responden a las diferentes asignaturas que integran el currículo de la escuela secundaria cubana y cada software está compuesto por seis módulos. Estos son: Temas, Ejercicios, Juegos, Biblioteca, Profesor y Resultados. El módulo Resultados permite observar el desempeño de los estudiantes con el software en cada una de las sesiones de trabajo. En este módulo se obtiene la información de cada una de las acciones realizadas por los estudiantes, de una forma secuencial en un texto, lo que trae como consecuencia que no permita la realización de análisis estadísticos o búsquedas especializadas. (2)

Colección Futuro (Versión instalada en las escuelas preuniversitarias cubanas)

Esta colección, destinada a la educación media superior de las escuelas cubanas, está constituida por 16 productos que abarcan las diferentes áreas de conocimiento, además de materiales dirigidos a la preparación de profesores y directivos.

El módulo Resultados se encuentra en cada uno de los dieciséis productos, permitiendo el análisis de trazas para el establecimiento de diagnósticos y el control de tareas en las que, el profesor, pudiera inclusive no estar presente. El análisis de la traza se realiza a través de cuatro niveles de reporte: (3)

- **Traza del estudiante:** Este reporte presenta la información de la interacción del estudiante con el software a nivel de sesión, o sea, los elementos que la componen son: el nombre y los apellidos del estudiante, el grado, el grupo, la fecha, la hora de autenticación, los puntos visitados, tiempos de estancia y resultados de cuestionarios.
- **Análisis de contenidos específicos:** Permite realizar análisis estadísticos de contenidos específicos de manera grupal o individual, además realiza análisis de los resultados de la interacción de los estudiantes con los cuestionarios interactivos del módulo.
- **Historial del estudiante:** Realiza un análisis gráfico de línea donde se observa la evolución del estudiante con la aplicación en sus diferentes interacciones.

Módulo Resultados

- **Análisis integral:** Brinda la posibilidad de realizar un análisis del recorrido del estudiante en todas las asignaturas, en sus diferentes interacciones con los productos de la colección.

El módulo Resultados permite observar el desempeño de los estudiantes con el software en cada una de las sesiones de trabajo. En este módulo se obtiene la información de cada una de las acciones realizadas por los estudiantes, de una forma secuencial en un texto, lo que impide la realización de análisis estadísticos o búsquedas especializadas. (4)

Colección Multisaber versión Multiplataforma (“La caja mágica”)

La colección “La caja mágica” está integrada por varios productos que se especializan en la enseñanza de diferentes materias, conformada por varios módulos, donde cada uno de estos brinda la posibilidad de realizar un conjunto de acciones diferentes. Los módulos son: Temas, Ejercicios, Mediateca, Profesor, Resultados y Juegos.

El módulo Resultados, tiene como objetivo brindar al profesor un informe detallado sobre el desempeño de cada uno de los estudiantes que interactúan con los restantes módulos de la aplicación, y a su vez, cada estudiante puede visualizar la trayectoria que ha tenido hasta el momento. Estos datos se pueden visualizar a través de las siguientes opciones:

- Selección del estudiante: En esta opción se determina el estudiante a analizar, permitiendo seleccionar la escuela, grado y grupo o sección. A partir de esto se muestra una lista de estudiantes con la fecha de la última sesión abierta por cada uno de ellos y muestra las opciones Registro e Historial.
- Opción de Registro: Esta opción permite al profesor agregar, editar o eliminar información de los estudiantes.
- Opción de Historial: Permite mostrar el historial de cada estudiante, seleccionando previamente la sesión de trabajo que se quiera analizar.

SadHEA-Web

SADHEA-Web es un Sistema de Autor para el desarrollo de hiperentornos de aprendizajes para la web, desarrollado por el Centro de Estudios de Software y sus Aplicaciones Docentes (CESOFTAD), de la

Módulo Resultados

Universidad de Ciencias Pedagógicas “José de la Luz y Caballero” de Holguín. Esta aplicación brinda la posibilidad de producir software educativo on-line, en forma de hiperentornos de aprendizajes y tiene como características que: no exige conocimientos de programación para los usuarios, permite actualización sistemática, el producto resultante es un software libre multiplataforma, incorpora conceptos y servicios propios de la web, distribución flexible (Internet, Intranet, Laboratorio escolar y Computadora local), posibilita el desarrollo de diferentes prototipos y diseños de la interfaz y permite la creación diferentes roles para el usuario. (5)

SadHEA-Web contiene un módulo Resultados que brinda al usuario una herramienta para analizar la trayectoria del estudiante por el software educativo, este cuenta con opciones como: (6)

- **Vínculos visitados:** Se almacena el nombre y el tiempo que el usuario estuvo en cada uno de los nodos de información a los que accedió (imágenes, sonidos, videos, animaciones, diaporamas, términos del glosario o palabras calientes del módulo temas, entre otros).
- **Ejercicios:** En esta sección se muestran los resultados generales del entrenamiento, ofreciendo la cantidad de ejercicios que respondió el usuario y clasificándolos en Bien, Regular, Mal y los No evaluados; así como la posibilidad de volver a consultar el ejercicio, tal y como el usuario lo respondió, y en los casos en que la respuesta fue incorrecta, posibilita visualizar la respuesta correcta.
- **Juegos:** Se almacena, de cada uno de los usuarios, los juegos en los que participó y los resultados obtenidos.

El módulo Resultados trae consigo muchos beneficios ya que brinda una herramienta capaz de registrar las trazas de todos los usuarios que utilicen el producto, permitiendo mostrar errores cometidos en las respuestas a los ejercicios resueltos y el análisis progresivo de cada una de las acciones realizadas.

1.3 Estudio de la metodología para el desarrollo de software

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar un software con calidad. Define las etapas para el desarrollo de un software, las tareas que se deben llevar a cabo en cada etapa, las restricciones que deben aplicarse, las técnicas y herramientas que deben emplearse, y la forma de controlar y gestionar un proyecto. (7)

Módulo Resultados

Producto a la constante innovación tecnológica se cuenta con diversas metodologías para el desarrollo de software, las cuales han sido clasificadas en dos grupos: las metodologías ágiles y las tradicionales o robustas. La gran diferencia entre estos dos grupos consiste en que las metodologías ágiles tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso y las tradicionales intentan conseguir el objetivo común mediante el orden y la documentación(7).Dentro del grupo de las metodologías tradicionales se encuentran entre otras: Proceso Unificado de Desarrollo de Software (RUP), Microsoft Solutions Framework (MSF) y Métrica; mientras que por las ágiles se encuentran: Extreme Programming (XP), SCRUM, Cristal Methods, entre otras.

RUP se selecciona como metodología de desarrollo de software para el módulo Resultados de la colección El Navegante, pues es la que está definida para todos los módulos de la plataforma, esto representa que toda la información de la aplicación está bajo las instrucciones de esta metodología, además de no exigir una constante interacción con el cliente y de presentar ventajas significativas en el desarrollo de la vida del software.

Esta metodología captura las mejores prácticas del conocimiento de líderes en Ingeniería de Software y proporciona a los equipos de desarrollo guías, estándares y recomendaciones para la construcción de software de alta calidad. Además intenta integrar todos los puntos a tener en cuenta durante todo el ciclo de vida del software, con el objetivo de abarcar tanto pequeños como grandes proyectos; proporcionando un conjunto de herramientas para realizar las actividades y la documentación. (8)

Las características principales de RUP son: (9)

Centrado en la arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas, software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Es como una radiografía del sistema, lo suficientemente completa como para que todos los implicados en el desarrollo tengan una idea clara de qué es lo que están construyendo. (10)

Iterativo e incremental: Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de

Módulo Resultados

cualquier proceso de desarrollo. RUP propone que cada fase se desarrolle en iteraciones, una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente unos más que otros. (11)

Dirigido por casos de uso: Se define un caso de uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los casos de uso representan los requisitos funcionales del sistema. En RUP los casos de uso además de ser una herramienta para especificar los requisitos del sistema, también se utilizan para guiar el análisis, diseño, implementación y prueba del proceso de desarrollo del software.

El desarrollo del software en RUP está dividido en cuatro fases ([Ve Anexo1](#)), estas son: (9)

Inicio: Se realiza una planificación de fases, se identifican los principales casos de uso, se llega a un acuerdo entre todos los interesados respecto a los objetivos del ciclo de vida para el proyecto y se identifican los riesgos. Se define el alcance del proyecto.

Elaboración: Se establece la línea base de la arquitectura, se realiza un plan de proyecto, se completan los casos de uso y se mitigan la mayoría de los riesgos.

Construcción: Esta fase se centra en la elaboración de un producto totalmente operativo basado en la línea base de la arquitectura con todas las funcionalidades del software.

Transición: El objetivo es garantizar que el software esté listo para entregar e instalar al cliente. En esta fase pueden surgir nuevos elementos que implicarán nuevos desarrollos.

Dentro de estas fases, RUP propone ocho flujos de trabajo, de los cuales cinco son obligatorios estos son.

Modelo del negocio: Se busca tener un mejor entendimiento de la organización donde se va a implantar el producto, conociendo así el negocio y buscando un lenguaje común para todos. Describe cuales son los procesos de negocio, identificando quiénes participan y que actividades requieren automatización.

Requerimientos: Se identifica qué es lo que el sistema debe hacer, para lo cual se capturan las funcionalidades requeridas y las restricciones que se imponen.

Módulo Resultados

Análisis y Diseño: En esta etapa se describe cómo el sistema será realizado a partir de las funcionalidades previstas. Se definen los diagramas de clases, los diagramas de colaboración y de interacción, se realizan las tablas de la base de datos y sus relaciones, en resumen se indica con precisión lo que se debe programar.

Implementación: Se realiza la implementación de las clases y sus relaciones en un lenguaje de programación. Se define cuáles nodos se utilizarán y la estructura de la aplicación.

Pruebas: Se evalúa la calidad de producto, se verifica que todos los requerimientos hayan sido correctamente implementados y que los errores sean informados antes de la entrega final.

Despliegue: Se lleva el producto a un entorno de usuario y se verifica que funcione, se realizan los manuales de usuarios, se empaqueta, distribuye e instala el software.

1.4 Tecnologías y herramientas a utilizar

En este epígrafe se describen las tecnologías y herramientas a utilizar para llevar a cabo el desarrollo del módulo. Las mismas están definidas por la arquitectura del proyecto Multisaber-El Navegante.

1.4.1 Lenguaje de modelado

UML (Lenguaje Unificado de Modelado) es un lenguaje proporciona un vocabulario y una regla para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. (12)

El Lenguaje Unificado de Modelado tiene muchas ventajas ya que permite representar el sistema de una forma gráfica de una manera que se pueda entender y especificar las características del sistema antes de su construcción. Es importante resaltar que UML se puede aplicar en el desarrollo de software proporcionando gran variedad de formas para dar soporte a metodologías de desarrollo de software como RUP. (12)

1.4.2 Herramienta Case

Visual Paradigm

Visual Paradigm es una herramienta Case que provee el modelado de procesos de negocios, creada para ayudar en el proceso de desarrollo de software. Soporta los principales estándares de la industria, tales como UML, SysML, BPMN y XMI. (13)

Esta herramienta posibilita la diagramación visual y el diseño de los proyectos, además ofrece un entorno fácil para la creación de diagramas UML. Se selecciona Visual Paradigm como herramienta case a utilizar pues es la que se aplica para el modelado de la colección El Navegante y permite la modelación de procesos de desarrollo de software que sigan la filosofía de software libre.(13)

1.4.3 Lenguajes de desarrollo

Un lenguaje de desarrollo está diseñado para describir un conjunto de acciones consecutivas que un equipo de cómputo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. Es un lenguaje artificial que puede ser usado para controlar el comportamiento de una computadora y está compuesto por un conjunto de reglas sintácticas y semánticas permitiendo expresar instrucciones que luego serán interpretadas. (14)

En el desarrollo de aplicaciones web se clasifican dos tipos de lenguajes principales. (14)

1. **Lenguajes de programación del lado del cliente:** Son los lenguajes que basan su procesamiento en el cliente web, es decir que se ejecutan en el navegador web del usuario.
2. **Lenguajes de programación del lado del servidor:** Son los lenguajes que se procesan en el lado del servidor y que generan la página antes de enviarla al cliente. Este proceso consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para generar páginas XHTML dinámicamente como respuesta.

Lenguajes de programación del lado del cliente

XHTML

HTML (HyperText Markup Language) es un lenguaje de marcado para la construcción de páginas web. XHTML es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de

Módulo Resultados

reemplazar al mismo ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. Al estar orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en lo que a código se refiere. Entre estos requisitos básicos se puede mencionar una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados. (15)

CSS 2

Cascading Style Sheets (Hojas de Estilo en Cascada) es un lenguaje usado para definir la presentación de un documento HTML. El objetivo del desarrollo con CSS es separar la estructura de un documento de su presentación (color, tipos y tamaño de fuentes, justificación de textos, fondos, alineación, margen). Con el fin de crear una interfaz más amigable para los usuarios, en CSS2 se incluyen nuevas propiedades que hacen eficiente el trabajo y permiten que los documentos sean estandarizados. (16)

JavaScript

Es un lenguaje de programación orientado a objetos. El corazón del lenguaje se embebe en los navegadores para interpretar los scripts incorporados en las páginas. Estos códigos pueden hacer que exista interactividad dinámica entre el usuario y la página, controlar el navegador o crear páginas completas HTML sin que sea necesario realizar nuevas peticiones al servidor. (17)

Entre las acciones típicas que se pueden realizar en script se tienen dos vertientes. Por un lado los efectos especiales de páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que es posible crear páginas interactivas. (18)

Características generales: (18)

- Es simple, no hace falta tener profundos conocimientos de programación para poder hacer un programa en JavaScript.
- Maneja objetos dentro de la página web y sobre ese objeto se definen diferentes eventos, los cuales facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la computadora del usuario, tales como formateo de unidades y modificaciones de archivos.
- Es dinámico, responde a eventos en tiempo real.

Lenguajes de programación del lado del servidor

PHP 5

PHP, acrónimo de Hypertext Pre-processor, es un lenguaje interpretado de programación del lado del servidor gratuito y multiplataforma, rápido, con una gran librería de funciones. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. (18)

Características generales: (18)

- Es un lenguaje orientado a objetos, que posee variedad de funciones implementadas y de mucha utilidad para los desarrolladores.
- Posee soporte para múltiples bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo de dato que estén manejando en tiempo de ejecución.
- Es libre, por lo que se presenta como una alternativa de fácil acceso.
- Es completamente orientado al desarrollo de aplicaciones web dinámicas, con acceso a información almacenada en una base de datos.
- Cuenta con una biblioteca de funciones sumamente amplia.

1.4.4 Frameworks (Plataformas de desarrollo)

Un Framework es a grandes rasgos un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software los Frameworks son librerías de código que contienen procesos o rutinas listos para usar. Los programadores lo utilizan para no tener que desarrollar las tareas más básicas, ya que estos tienen incluido implementaciones que no se necesitan volver a programar.

Módulo Resultados

Un Framework para el desarrollo web asiste al trabajo de un programador en dos niveles: en el lado del cliente, ofreciendo funciones Scripts para enviar peticiones al servidor. En el lado del servidor, el cual procesa las peticiones, busca información, y la transmite al navegador. (19)

Frameworks del lado del cliente

En la actualidad han surgido varias alternativas para el uso de aplicaciones AJAX, dentro de estas se incluyen múltiples Frameworks para el desarrollo del lado del cliente con JavaScript los cuales presentan un conjunto de funcionalidades que permiten hacer más eficiente el trabajo en una página web, dentro de estos se encuentran: jQuery, Dojo toolkit, ExtJS, Mootools y Prototype.

JQuery 1.5

Utilizando JavaScript se puede crear una infraestructura que permita la creación de aplicaciones complejas del lado del cliente. JQuery es una biblioteca de JavaScript que permite la creación de interfaces de usuario, efectos dinámicos y aplicaciones que hacen uso de AJAX.

La ventaja del uso de jQuery, es que la página se puede manipular en cuanto se ha cargado el código HTML (y por tanto, se ha construido el árbol DOM de la página) mientras que la función de JavaScript espera a que se carguen todos los elementos de la página, incluyendo todas las imágenes. De esta forma, las aplicaciones realizadas con jQuery pueden responder de forma más rápido que las aplicaciones JavaScript tradicionales. (20)

Esta potente biblioteca encabeza las listas de bibliotecas JavaScript más utilizadas gracias a su sencillez, su flexibilidad, su potencia y su rendimiento. Se caracteriza por tener una amplia aceptación por parte de los programadores y un grado de penetración en el mercado muy amplio, es un producto serio, estable y bien documentado. (20)

Framework del lado del servidor

Los Frameworks del lado del servidor presentan grandes ventajas y funcionalidades y resultan de gran eficiencia y ahorro de tiempo en la construcción de aplicaciones web.

Dentro de las principales ventajas se puede encontrar que traen implementados patrones de arquitectura. Además, automatizan las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (21).

Módulo Resultados

Existen muchos Frameworks para el trabajo con PHP dentro de los que se encuentran: Symfony, CodeIgniter, CakePHP y Zend Framework.

Symfony 1.4.3

Symfony es un Framework multiplataforma del lado del servidor desarrollado completamente en PHP5. Trae consigo un conjunto de herramientas y clases que reducen considerablemente el tiempo de duración en el desarrollo de las aplicaciones. Una de las ventajas de este es que es Open Source (Código Abierto), además, automatiza la mayoría de elementos comunes de los proyectos web. También posee una amplia documentación para que más desarrolladores se incorporen al proyecto. (21)

Características generales: (21)

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

1.4.5 Servidor web

Es un programa que permite atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS. Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente ciclo:

- Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).

- Busca el recurso.
- Envía el recurso utilizando la misma conexión por la que recibió petición.
- Vuelve al segundo punto.

Dentro de los principales servidores web se encuentran Apache, Microsoft IIS y Google GFE. (22)

Apache

Es un servidor web Open Source y el más utilizado a nivel mundial. Es multiplataforma, extensible y posee una amplia documentación. El servidor consta de una sección *core* y diversos módulos que aportan muchas de las funcionalidades que podrían considerarse básicas para un servidor web. (28)

Características generales: (23)

- Es robusto y seguro.
- Posee una configuración simple y potente a través de ficheros.
- Tiene soporte para autenticación HTTP.
- Cuenta con un sistema de notificación de errores HTTP altamente configurable.
- Posibilita el chequeo avanzado de sintaxis de la *URL*, incluyendo caracteres falsos que son fuentes frecuentes de ataques como SQL Injection, Command Injection y XSS.

1.4.6 Sistema de gestión de bases de datos

DBMS acrónimo de (Database Management System), en español *sistema de gestión de bases de datos* (SGBD), es un software que sirve de interfaz entre una base de datos, el usuario y la aplicación que la utiliza.

Un SGBD ayuda a realizar las siguientes acciones: (24)

- Definición de los datos.
- Mantenimiento de la integridad de los datos dentro de la base de datos.
- Control de la seguridad y privacidad de los datos.
- Manipulación de los datos.

Módulo Resultados

Características generales: (24)

- **Abstracción de la información:** Los detalles de almacenamiento son transparentes al usuario.
- **Redundancia mínima:** Evitan la aparición de información repetida o redundante.
- **Consistencia:** La información repetida se actualiza de forma coherente una vez que se obtiene redundancia nula.
- **Seguridad:** Habitualmente, poseen un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos, ya que la información almacenada puede tener un gran valor, por lo que se debe garantizar que esta se encuentre segura frente a los intrusos.
- **Integridad:** Protege los datos introducidos por usuarios descuidados, o cualquier otra ocurrencia capaz de dañar la información almacenada ante fallos de hardware.
- **Respaldo y recuperación:** Proporciona una forma eficiente de realizar copias de respaldo de la información almacenada y los cambios realizados.

Actualmente existen múltiples SGBD dentro de los que se encuentran PostgreSQL, MySQL, Oracle, Microsoft SQL Server y Apache Derby.

PostgreSQL 8.4

Es un SGDB relacional libre orientado a objetos, está diseñado para administrar grandes cantidades de datos. Se ejecuta en los Sistemas Operativos más utilizados en el mundo, incluyendo Linux, varias versiones de UNIX y Windows. Es compatible con el almacenamiento de objetos binarios, incluye imágenes, sonidos y videos. Tiene interfaces de programación con diferentes lenguajes como PHP, C, C++, Java, .NET, Perl, Python, entre otros. Además brinda una amplia documentación lo que hace más fácil trabajar con el mismo. (25)

Características generales: (25)

- Presenta claves ajenas también denominadas llaves ajenas o claves foráneas en inglés (*foreign keys*).
- Tiene integridad transaccional.

Módulo Resultados

- Posee herencia de tablas.
- Cuenta con tipos de datos y operaciones geométricas.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Presenta diferentes herramientas de administración dentro de las que destacan:
 - ✓ PgAdmin3: Entorno de escritorio visual, con licencia Open Source.
 - ✓ PgAccess: Entorno de escritorio visual.
 - ✓ PhpPgAdmin: Entorno web.
 - ✓ PSQL: Cliente de consola.
 - ✓ Database Master: Entorno de escritorio visual.

1.4.7 Entorno de desarrollo integrado (IDE)

IDE acrónimo de (Integrated Development Environment) es un software compuesto por un conjunto de herramientas de programación que puede dedicarse en exclusiva a un sólo lenguaje de programación o varios. Los IDEs pueden ser aplicaciones por si solos o pueden ser parte de aplicaciones existentes. Están compuestos por un editor de texto, un compilador, un intérprete, herramientas de automatización y un depurador. (26)

Dentro de los IDEs más completos existentes para el desarrollo de aplicaciones se encuentran Netbeans, Zend Studio, Eclipse, Aptana y Komodo IDE.

Netbeans 6.9

Netbeans es un IDE de código abierto, gratuito sin restricciones de uso y una plataforma de aplicaciones que permiten a los desarrolladores crear de manera rápida diferentes tipos de aplicaciones.

Módulo Resultados

Es una herramienta realizada en java pensada para escribir, compilar, depurar y ejecutar programas de diferentes lenguajes, ya que cuenta con un número importante de módulos. (26)

Características generales: (25)

- Administración de las interfaces de usuario.
- Administración de las configuraciones del usuario.
- Administración del almacenamiento guardando y cargando diferentes tipos de datos.
- Administración de ventanas.

NetBeans 6.9 permite crear aplicaciones Web con PHP 5, además tiene soporte para trabajar con Symfony, Framework que brinda disímiles ventajas para crear aplicaciones web.

1.5 Patrón arquitectónico Modelo-Vista-Controlador

El patrón arquitectónico Modelo-Vista-Controlador (MVC) es muy utilizado en el desarrollo de aplicaciones web, por la posibilidad que brinda de organizar el código de una aplicación separando los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. ([Ver Anexo2](#))

Este patrón representa un factor fundamental para el trabajo organizado y eficiente para el desarrollo de aplicaciones web. Fue creado para disminuir el esfuerzo al momento de programar, permitiendo una clara separación entre los componentes de un programa.

Modelo: Es el modelo de datos, este sirve como representación específica de toda la información con la cual el sistema va a trabajar. La lógica de datos puede llegar a asegurar la integridad de ellos y permite derivar nuevos datos.

Vista: Es la página HTML y el código que provee de datos dinámicos a la página. Presenta el modelo con el que interactúa el usuario y es más conocida como interfaz.

Controlador: Responde a eventos, son acciones que el usuario invoca, implica cambios en el modelo y también en la vista. (21)

1.6 Conclusiones

En el presente capítulo se efectuó un análisis bibliográfico, exponiendo los principales temas de interés en cuanto a herramientas similares existentes en Cuba, que aunque ninguna cumple con el soporte tecnológico establecido para la versión 2.0 del módulo Resultados, si aportaron ideas para su creación. Para llevar a cabo el proceso de desarrollo de software es necesario definir de forma precisa las herramientas, metodologías y lenguajes de programación a utilizar. Se selecciona RUP como metodología a emplear ya que facilitará la documentación, el seguimiento y control de las actividades a desarrollar. Se utilizará Visual Paradigm como herramienta de modelado, que emplea UML como lenguaje de modelado, además de ser libre y multiplataforma. Dentro de los lenguajes y tecnologías seleccionadas se encuentran XHTML, JavaScript, PHP, CCS y AJAX. La aplicación a desarrollar será web y el patrón arquitectónico a utilizar es el MVC. Lo expuesto anteriormente aporta los elementos necesarios para el desarrollo del módulo Resultados de la colección El Navegante.

Capítulo 2: Descripción del sistema.

2.1 Introducción

El presente capítulo tiene como objetivo principal dar una visión de las características del módulo a desarrollar, teniendo como punto de partida los conceptos más importantes del dominio, y llegando de esta manera a la confección del modelo de dominio. Para facilitar la comprensión y el análisis de dicho modelo se exponen los requerimientos funcionales y no funcionales, que son las características y cualidades que el sistema debe cumplir. Además, se obtienen los actores involucrados en el sistema, el diagrama de casos de uso del sistema y la descripción de cada uno de estos.

2.2 Modelo de dominio

RUP como metodología de desarrollo de software propone realizar un modelo de negocio, pero para ello se hace necesaria la identificación de procesos del negocio, siendo este un conjunto de actividades que responden directamente a un proceso en sí (12). Puesto que fue definido por la arquitectura del proyecto Multisaber-Navegante que no es necesario un modelo completo del negocio se realiza un modelo de dominio.

Un modelo de dominio explica los conceptos más significativos en un problema. Representa cosas del mundo real, no componentes del software y tiene como objetivo entender y describir los objetos más importantes dentro del contexto del sistema y las relaciones entre ellos que. (9)

2.2.1 Análisis de los conceptos del dominio

Colección Navegante: Conjunto de Software que representan hiperentornos de enseñanza-aprendizaje orientada a una educación o área en común.

Producto multimedia: Producto que combina diversos tipos de medias.

Media: Se utiliza para referirse tanto a texto, imágenes, sonidos o videos.

Texto: Es una composición de signos codificado en un sistema de escritura (como un alfabeto) que forma una unidad de sentido.

Diaporama: Proyección simultánea de diapositivas sobre una o varias pantallas.

Módulo Resultados

Video: Producto audiovisual capturado de la realidad objetiva, mediante cámaras.

Sonido: Sensación producida en el órgano del oído por el movimiento vibratorio de los cuerpos, transmitido por un medio elástico, como el aire.

Imagen: Representación gráfica de una expresión.

Módulo: Conjunto de elementos fundamentales que componen el producto.

Juegos: Módulo encargado de que los estudiantes puedan divertirse a través de juegos mientras emplean sus habilidades y conocimientos.

Mediateca: Módulo responsable de gestionar todo lo referente a los recursos multimedia, mediante este el estudiante puede interactuar con textos, imágenes y videos.

Ejercicios: Módulo encargado de controlar el aprendizaje a través de baterías de preguntas o cuestionarios que emiten un criterio a cada una de las respuestas.

Resultados: Es el encargado de almacenar y llevar el control de las trazas del estudiante para visualizar su comportamiento durante su interacción con cada producto de la Colección.

Contenidos: Módulo en el que se representan los contenidos de las asignaturas o temas a los que responde el producto. Esto se presenta en forma de hipermedia permitiendo la navegación no lineal por los contenidos.

Sección: Conjunto de opciones o secciones que brinda el módulo Resultados.

Análisis de contenidos: Esta sección permite realizar un análisis de los resultados de uno o más estudiantes sobre los ejercicios realizados de cada uno de los contenidos del producto mostrando datos como: nombre del tema y cantidad de evaluaciones realizadas.

Traza del estudiante: Presenta información resumida de la interacción del estudiante en una sesión de trabajo con el software mostrando Nombre, Apellido, Grado y Sección del estudiante y la fecha, hora de entrada, hora de salida y cantidad de veces que interactuó en cada sesión.

Historial del estudiante: Esta sección permite realizar un análisis de los resultados de los estudiantes en los temas del producto tomando en cuenta sus resultados en función de las fechas en que los estudiantes

Módulo Resultados

interactuaron con los ejercicios, mostrando datos como: nombre del tema y cantidad de evaluaciones realizadas.

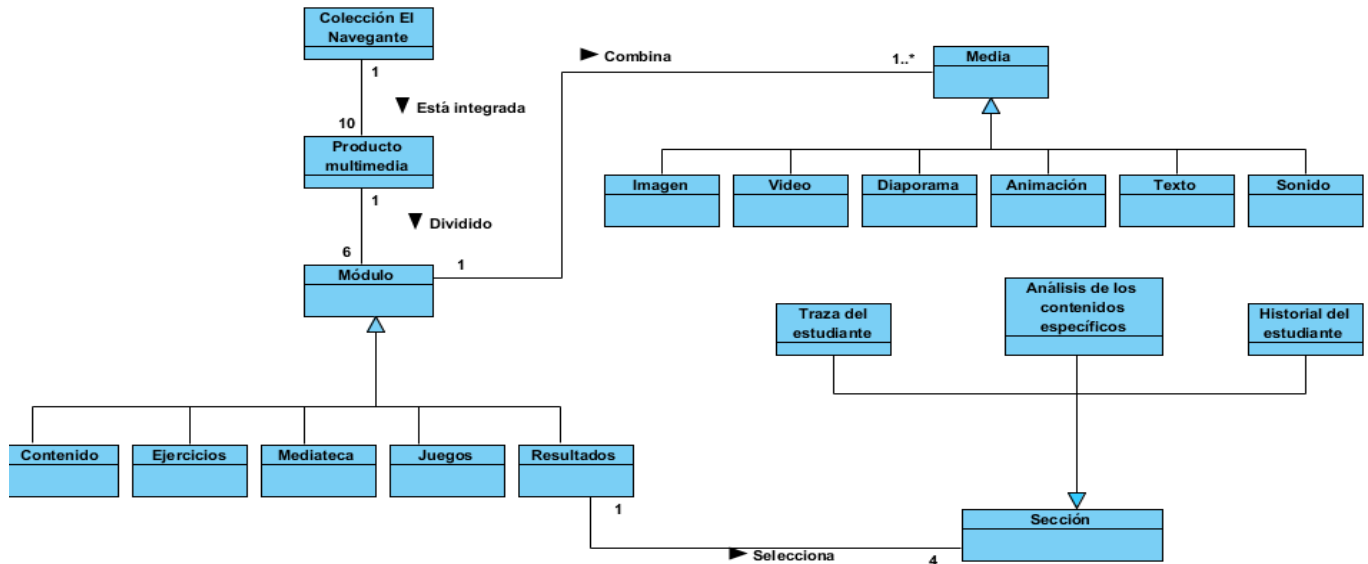


Figura 1 Modelo de dominio

2.3 Descripción del sistema propuesto

El módulo Resultados posibilita llevar un control debidamente organizado, sobre las acciones realizadas por los estudiantes; brindando la posibilidad de obtener un conjunto de reportes (traza del estudiante, análisis de los contenidos específicos, historial del estudiante y análisis integral del estudiante); ofreciendo opciones de búsqueda y filtrado, para conocer los datos personales de los estudiantes y mostrando la evolución de cada uno en las asignaturas y contenidos específicos, a través de análisis mediante gráficas de efectividad.

La propuesta del módulo Resultados se divide en 4 secciones. La sección de *Trayectoria del estudiante* presenta información consolidada de la interacción del estudiante en una sesión de trabajo con la Colección. La sección de *Análisis de contenidos* facilita la consulta de las calificaciones que el estudiante va obteniendo de forma paulatina, con los cuestionarios interactivos del módulo ejercicios, en función de los contenidos. Dentro de la sección *Análisis integral* se analizará a nivel general los resultados de la interacción con los ejercicios de todos los productos de la Colección en función de las asignaturas. Por

último, se encuentra la sección *Historial del estudiante* que presentará un análisis de los resultados de la interacción de los estudiantes con los contenidos, tomando en cuenta sus resultados en función de las fechas en que los estudiantes interactuaron con los ejercicios.

2.4 Especificación de requerimientos

Un proceso de desarrollo de software no puede ser exitoso sin una descripción detallada, correcta y exhaustiva de los requerimientos, ya que es necesario saber qué debe hacer dicho sistema, además de tener bien claro cuáles son las expectativas del cliente con respecto al sistema a desarrollar. (7)

Un requisito (o requerimiento) es una cualidad o capacidad especificada por el cliente y que debe de cumplir el sistema. Estos se pueden clasificar en: (12)

- Requerimientos funcionales: Definen las funcionalidades que el sistema tiene que realizar, para lo cual se identifican las funcionalidades requeridas.
- Requerimientos no funcionales: Son propiedades o cualidades que el producto debe tener. Tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, entre otras.

A continuación se presentan los requisitos funcionales y no funcionales definidos por el proyecto Multisaber-Navegante, los cuales se pueden encontrar en el documento *“Especificación de requisitos de El Navegante”*.

Listado de los Requisitos Funcionales (RF)

RF1: Mostrar la pantalla principal del módulo Resultados.

RF2: Generar reportes del historial del estudiante.

RF2.1: Mostrar las fechas en las que se han realizado ejercicios.

RF2.2: Permitir la selección de las fechas que se deseen analizar.

RF2.3: Mostrar las fechas en las que se han realizado ejercicios.

RF2.4: Permitir la selección de las fechas que se deseen analizar.

Módulo Resultados

RF2.5: Mostrar los contenidos realizados según las fechas seleccionadas.

RF2.6: Permitir la selección de los contenidos que se deseen analizar.

RF2.7: Mostrar los resultados de los contenidos seleccionados.

RF2.8: Mostrar una gráfica de comparación de efectividad de los contenidos seleccionados.

RF3: Consultar la traza de un estudiante.

RF3.1: Seleccionar categoría.

RF3.2: Mostrar resultados de la categoría seleccionada.

RF4: Configurar el tipo de análisis a realizar.

RF4.1: Listar estudiantes en el software.

RF5: Generar reportes del análisis de contenidos específicos.

RF5.1: Mostrar los contenidos con ejercicios realizados.

RF5.2: Permitir la selección de los contenidos que se deseen analizar.

RF5.3: Mostrar los resultados de los contenidos seleccionados.

RF5.4: Mostrar una gráfica de comparación de efectividad de los contenidos seleccionados.

RF6: Generar reportes del análisis integral.

RF6.1: Mostrar los Software con ejercicios realizados.

RF6.2: Permitir la selección de los Software que se deseen analizar.

RF6.3: Mostrar los resultados de los Software seleccionados.

RF6.4: Mostrar una gráfica de comparación de efectividad de los Software seleccionados.

Listado de los Requisitos no Funcionales (RNF)

RNF 1 Requisitos de software.

- ✓ Computadora Personal con navegador Mozilla Firefox 3.0 o superior.

RNF 2 Requisitos de hardware.

- ✓ Procesador Pentium 233 MHz (recomendado 500 MHz o mayor).
- ✓ 64 MB de RAM (recomendado 128 MB de RAM o mayor).
- ✓ 1 GB de espacio en disco duro.
- ✓ Soporte de video que admita resolución de al menos 800x600 y 24 bits.
- ✓ Dispositivo de red de al menos 10 Mbits.
- ✓ El sistema interactuará con una impresora que permita imprimir los diferentes contenidos como respuesta a las funcionalidades del sistema.

RNF 3 Requisitos de restricciones en el diseño y en la implementación.

- ✓ JQuery 1.5
- ✓ PHP 5.2
- ✓ Symfony 1.4
- ✓ Apache 2.x
- ✓ PostgreSQL 8.4

RNF 4 Requisitos de apariencia e interfaz externa.

- ✓ El diseño de las interfaces debe ser amigable y sencilla.
- ✓ El sistema proporcionará claridad y correcta organización de la información, permitiendo la interpretación correcta e inequívoca de ésta.
- ✓ El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.

RNF 5 Requisitos de soporte.

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

Módulo Resultados

- ✓ Requerimientos de ayuda y documentación en línea.

RNF 6 Requisitos de portabilidad.

- ✓ El sistema podrá ser utilizado bajo los sistemas operativos Linux y Windows.

RNF 7 Requisitos de usabilidad

- ✓ La utilización del módulo por parte de los usuarios sin avanzados conocimientos de computación, no requiere de previa preparación, debido al diseño sencillo y estándar que tiene.

RNF 8 Requisitos de interfaces de usuario

- ✓ Las interfaces se desarrollarán teniendo en cuenta elementos de las aplicaciones con tecnología multimedia diseñadas para escritorio y las aplicaciones web, logrando equilibrar la eficiencia del software sobre la web con la interactividad y el diseño visual que exige la aplicación.

2.5 Definición de los casos de uso

Un caso de uso (CU) es una secuencia de actividades que se desarrollan entre un sistema y sus actores, en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas, describiendo las funcionalidades propuestas de la nueva aplicación. En resumen los casos de usos representan los requerimientos funcionales del sistema capturado con anterioridad, formando un conjunto de actividades que llevan implícito funcionalidades básicas propiamente del sistema. (7)

2.5.1 Actores del sistema

Un actor es cualquier usuario externo que interactúe con el sistema, ya que define acciones y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina. (12)

Actor	Descripción
Usuario	Representa a una persona que va interactuar con el sistema principalmente estudiante, invitado o profesor.

Tabla 1 Determinación y descripción de los actores del sistema.

Módulo Resultados

2.5.2 Diagrama de casos de uso del sistema

El diagrama de casos de uso del sistema es un conjunto de acciones que un sistema ejecuta y que produce un resultado observable para un actor. Es decir, son funcionalidades que el sistema brinda a los actores que interactúan con el mismo. (7)

A continuación se muestra el diagrama de casos de uso del sistema, quedando representadas gráficamente las funcionalidades del sistema y su interacción con los actores.

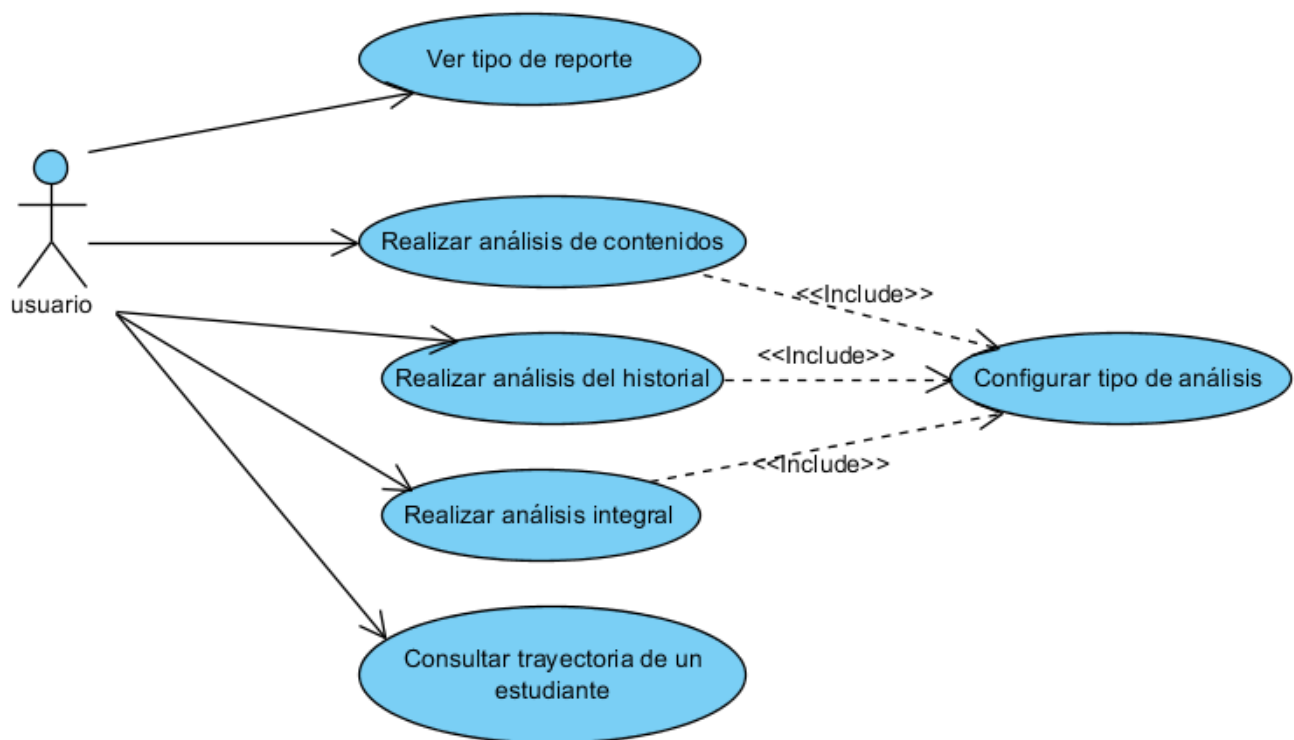


Figura 2 Diagrama de casos de uso del sistema

2.5.3 Descripción de los casos de uso del sistema

En el siguiente epígrafe se realiza una descripción de los casos de uso presentes en el diagrama de casos de uso del sistema. Se define el objetivo general de cada caso de uso, el actor que le da inicio, un

Módulo Resultados

resumen de su funcionamiento así como las condiciones necesarias para que el caso de uso en sí pueda realizarse.

A continuación se muestra una descripción de los casos de uso de forma resumida, para verla más ampliada consultar el [Anexo 3](#).

CU-1	Ver tipo de reporte
Actor	Usuario
Propósito	Mostrar las distintas opciones de reporte que realiza el <i>módulo Resultados</i> .
Resumen	El caso de uso se inicia cuando el usuario selecciona la imagen que representa al módulo Resultados, el sistema muestra una interfaz permitiéndole al usuario seleccionar el tipo de reporte que desee generar, terminando así el caso de uso.

Tabla 2 Resumen del CU-1: Ver tipo de reporte.

CU-2	Configurar tipo de análisis
Actor	Usuario
Propósito	Configurar el tipo de análisis que se desea realizar en los reportes de <i>análisis de contenidos específicos, historial del estudiante y análisis integral</i> .
Resumen	El caso de uso inicia cuando el usuario selecciona la opción de realizar <i>análisis integral, análisis de contenidos o historial del estudiante</i> , el sistema muestra una interfaz permitiéndole al usuario la selección del tipo de análisis a realizar (<i>individual, grupos, subgrupos, general</i>). El usuario selecciona la institución, año y sección de los estudiantes que se desea analizar y el sistema muestra un listado de

Módulo Resultados

	los que cumplen con esos parámetros, terminando así el caso de uso.
--	---

Tabla 3 Resumen del CU-3: Configurar tipo de análisis.

CU-3	Realizar análisis de contenidos
Actor	Usuario
Propósito	Permitir generar los reportes de los contenidos específicos para cada uno de los tipos de análisis permitidos.
Resumen	El caso de uso inicia cuando el usuario selecciona la acción realizar <i>análisis de contenidos específicos</i> . El sistema muestra una interfaz para la selección del tipo de análisis a realizar. El usuario selecciona los contenidos que se desea analizar, mostrando el sistema una tabla con los resultados de los mismos y una gráfica con la comparación de la efectividad de los contenidos, terminando así el caso de uso.

Tabla 4 Resumen del CU-4: Realizar análisis de contenidos

CU-4	Realizar análisis integral
Actor	Usuario
Propósito	Permitir generar los reportes de los Software seleccionados para cada uno de los tipos de análisis permitidos.
Resumen	El caso de uso inicia cuando el usuario selecciona la acción realizar <i>análisis integral</i> . El sistema muestra una interfaz para la selección del tipo de análisis a realizar. El usuario selecciona los Software que se desea analizar, mostrando el sistema una tabla con los resultados de los mismos y una gráfica con la comparación de la efectividad de los

Módulo Resultados

	contenidos, terminando así el caso de uso.
--	--

Tabla 5 Resumen del CU-5: Realizar análisis integral.

CU-5	Realizar análisis del historial
Actor	Usuario
Propósito	Permitir generar los reportes de los contenidos seleccionados en una fecha determinada, para cada uno de los tipos de análisis permitidos.
Resumen	El CUS se inicia cuando el usuario desea ver el <i>historial</i> de uno o más estudiantes en los contenidos que han realizado ejercicios en un rango de fecha. El sistema muestra los resultados de los mismos y una gráfica con la comparación de la efectividad de los contenidos, terminando así el caso de uso.

Tabla 6 Resumen del CU-6: Realizar análisis del historial

CU-6	Consultar trayectoria del estudiante
Actor	Usuario
Propósito	Mostrar la trayectoria de un estudiante en una categoría seleccionada por el usuario.
Resumen	El CUS inicia cuando el usuario desea ver la trayectoria de algún estudiante, permitiendo el sistema seleccionar una Categoría (<i>Datos Generales, Itinerario, Temas, Ejercicios, Juegos, Imágenes, Videos, Información de Interés, Glosario</i>) a consultar. El sistema muestra los resultados del estudiante seleccionado, terminando así el caso de uso.

Tabla 7 Resumen del CU-7: Consultar trayectoria del estudiante.

2.6 Conclusiones

En el presente capítulo se realizó una descripción de la propuesta de solución, quedando definidos los principales conceptos asociados al dominio del problema, permitiendo realizar su modelación, logrando un mejor entendimiento de lo requerido. Los requisitos funcionales identificados se concretaron en seis casos de uso, representados en el diagrama de caso de uso del sistema, quedando definidas las funcionalidades básicas de la aplicación. De esta forma es posible comenzar una fase posterior donde se realizará el análisis y diseño del sistema propuesto.

Capítulo 3: Análisis y Diseño

3.1 Introducción

El capítulo que se presenta a continuación contiene los principales artefactos generados a raíz del flujo de trabajo análisis y diseño propuesto por la metodología RUP. Se expone el resultado de la realización del modelo de análisis, basándose en la realización de los diagramas de clases de análisis correspondientes a los casos de uso identificados. Se realizan además los diagramas vinculados al diseño de la aplicación y se detalla todo lo referente al modelo de datos.

3.2 Modelo de análisis

El modelo de análisis contiene clases y objetos, proporcionando una comprensión detallada de los requisitos. En la realización de este es necesario identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas, con toda esta información se construye el diagrama de clases del análisis. (27)

El diagrama de clases del análisis tiene como objetivo visualizar la información a través de una representación visual de los objetos, exponiendo al usuario un diseño donde se evidencian las interrelaciones que se producen entre actores y clases, utilizando específicamente tres estereotipos básicos, para la representación de las clases. (9)

3.2.1 Diagramas de clases del análisis

En el análisis se muestran los siguientes estereotipos de clases: (11)

Clase Interfaz: Estas clases se encargan de manejar la interacción entre el sistema y sus actores. Ejemplos: ventanas, formularios, comunicación con otros sistemas o dispositivos.

Clase Controladora: Coordinan el trabajo para la realización y la especificación de los casos de uso, en otras palabras, encapsulan el comportamiento de un CU. Estas clases tienen un control sobre todas las acciones a realizar.

Módulo Resultados

Clase Entidad: Representa la información que generalmente es de larga duración y a menudo persistente. También reflejan entidades del mundo real que resultan necesarias para realizar tareas internas del sistema.

A continuación se muestran los diagramas de clases del análisis, que se organizan en el mismo orden en que se listaron los casos de uso anteriores.



Figura 3 Diagrama de clases del análisis del CU Ver tipo de reportes

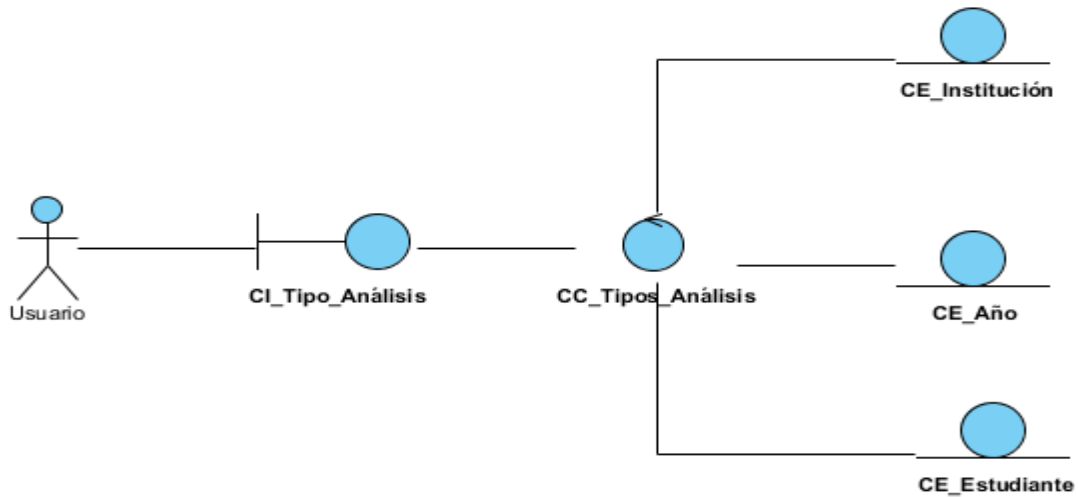


Figura 4 Diagrama de clases del análisis del CU Configurar tipo de análisis

Módulo Resultados

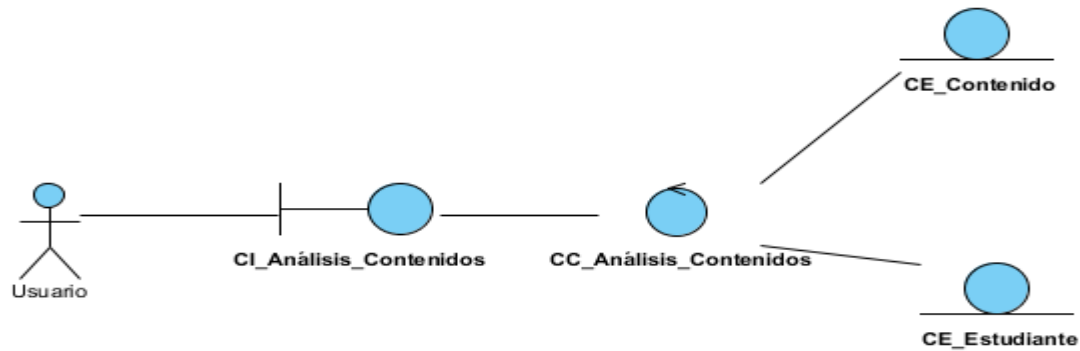


Figura 5 Diagrama de clases del análisis del CU Realizar análisis de contenidos

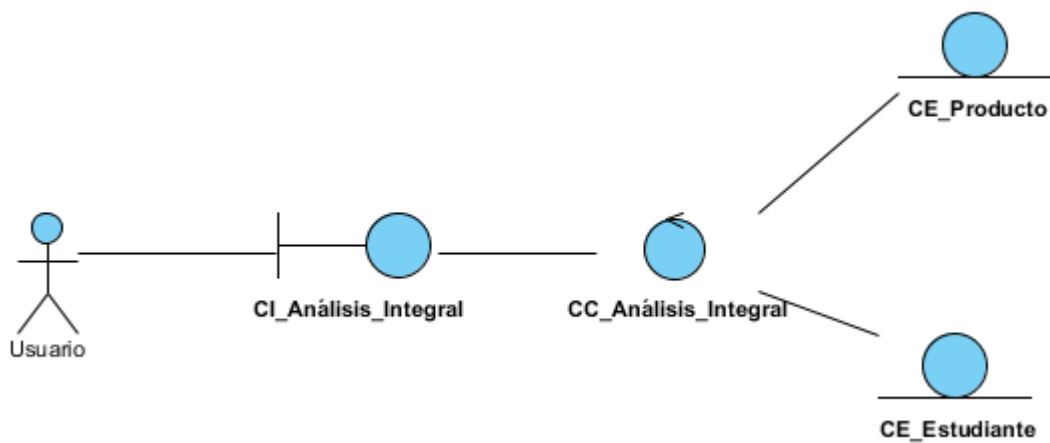


Figura 6 Diagrama de clases del análisis del CU Realizar análisis integral

Módulo Resultados

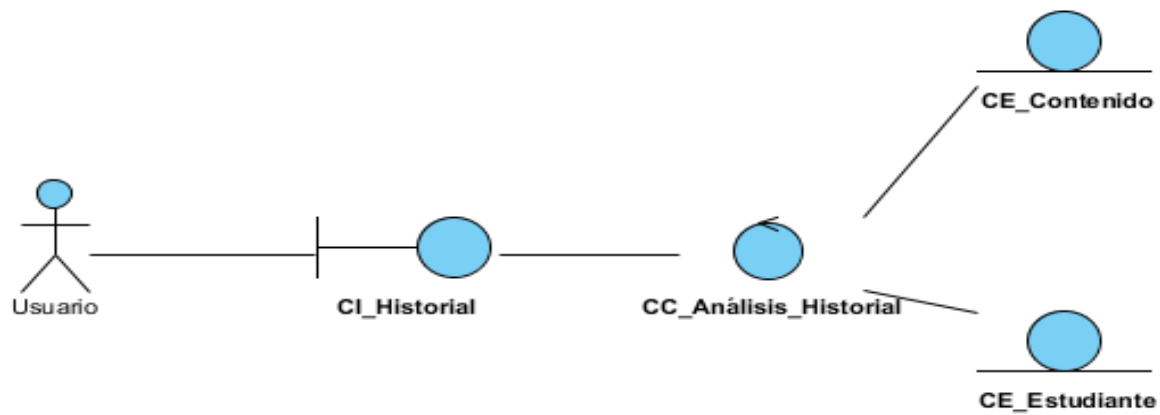


Figura 7 Diagrama de clases del análisis del CU Realizar análisis del historial

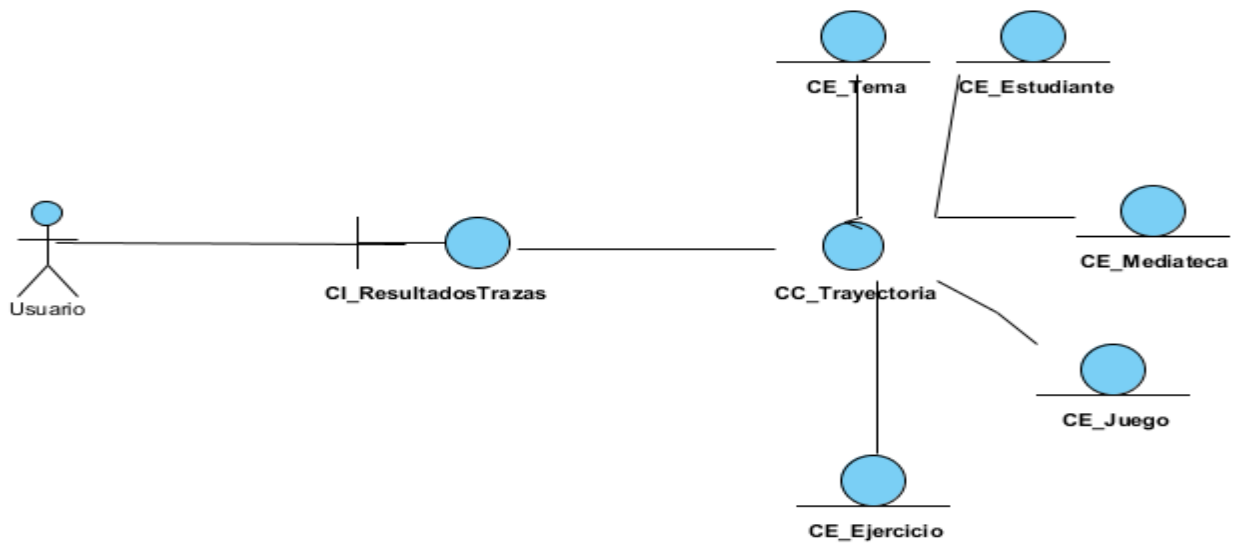


Figura 8 Diagrama de clases del análisis del CU Consultar la trayectoria del estudiante

3.3 Patrones de diseño

Los patrones tratan los problemas del diseño que se repiten y que se presentan en situaciones particulares, con el objetivo de proponer soluciones a ellas. Por lo tanto, los patrones de diseño son soluciones exitosas a problemas comunes.

3.3.1 Patrón Mediador

Un Mediador es un patrón que define un objeto y encapsula cómo interactúan un conjunto de objetos. El Mediador estimula la pérdida de acoplamiento, ocultando las referencias explícitas entre los objetos, permitiendo variar su interacción de forma independientemente. Este patrón se utiliza cuando: (28)

- Un conjunto grande de objetos se comunica de una forma bien definida, pero compleja.
- Reutilizar un objeto se hace difícil porque se relaciona con muchos objetos.
- El comportamiento de muchos objetos que está distribuido entre varias clases.

En la implementación del módulo Resultados el patrón Mediador se ve reflejado en dos de las clases JavaScript. La primera, denominada “mediador” se encarga de controlar la interacción entre los diferentes módulos de la aplicación. La segunda denominada, “reMediadorResultados” se encarga de dirigir todas las acciones de las clases JavaScript dentro del módulo.

3.3.2 Patrón Controlador frontal

El patrón de diseño Controlador frontal se basa en utilizar un controlador como punto inicial para la gestión de las peticiones. El controlador gestiona estas peticiones, y realiza algunas funciones como: comprobación de restricciones de seguridad, manejo de errores, mapear y delegación de las peticiones a otros componentes de la aplicación que se encargarán de generar la vista adecuada para el usuario (21).

Las aplicaciones Symfony utilizan el patrón de diseño Controlador frontal, con el objetivo de garantizar un único punto de entrada a cada aplicación. De esta manera todas las peticiones que realiza el usuario se controlan mediante un único script de PHP, que se encarga de redirigir las peticiones a cada módulo y acciones correspondientes en cada caso (21).

3.4 Modelo de diseño

El Modelo de diseño es un modelo de objetos que describe la realización de los casos de uso. En el diseño se modela el sistema, incluida la arquitectura, para que soporte todos los requisitos, incluyendo los no funcionales, constituyendo entrada esencial a las actividades de implementación y prueba. (11)

3.4.1 Diagramas de clases del diseño

Los diagramas de clases muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. El diagrama de clases del diseño es un modelo de objetos que describe la realización de los casos de uso, centrándose en los requisitos funcionales y no funcionales. Para la modelación de la solución se proponen como estereotipos del diseño los estándares para modelar aplicaciones web, ya que estas se diferencian un poco del resto de las aplicaciones, representando así, las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. (9)

En la siguiente tabla se muestran las terminologías utilizadas en los diagramas de clases del diseño: (11)




Clases	Estereotipos	Función
Server Page [SP] (en español Página Servidora)		Representa la página web que tiene código que se ejecuta en el servidor. Su principal función es construir la Página cliente.
Client Page [CP] (en español Página Cliente)		Representa una página Web, con formato HTML. Son interpretadas por el navegador, su función es visualizar, interactuar y mostrar lo que el usuario necesita.
Form (en español Formulario)		Colección de elementos de entrada que son parte de una página cliente. Sus atributos son los elementos de entrada del formulario. Su principal función es enviar los datos entrados por el usuario a la Página servidora.

Tabla 8 Clases del Diseño

Módulo Resultados

A continuación se presentan los diagramas de clases del diseño para cada uno de los casos de uso del sistema. Con el objetivo de simplificar los Diagramas de Diseño se representaron las clases JavaScript de manera ampliada en la figura 29 ([Anexo 4](#)):

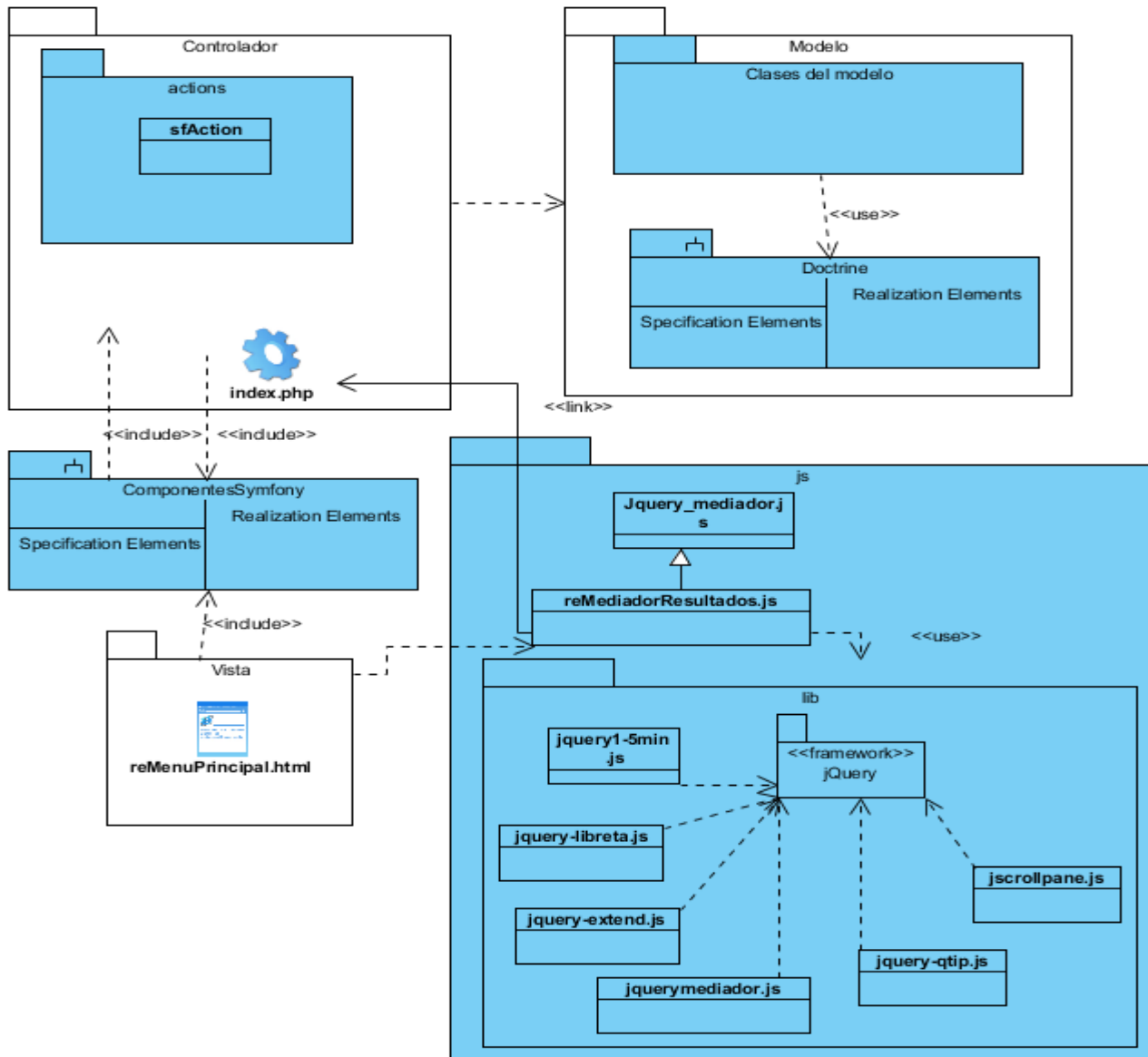


Figura 9 Diagrama de clases del diseño CU_Ver tipo de reportes.

Módulo Resultados

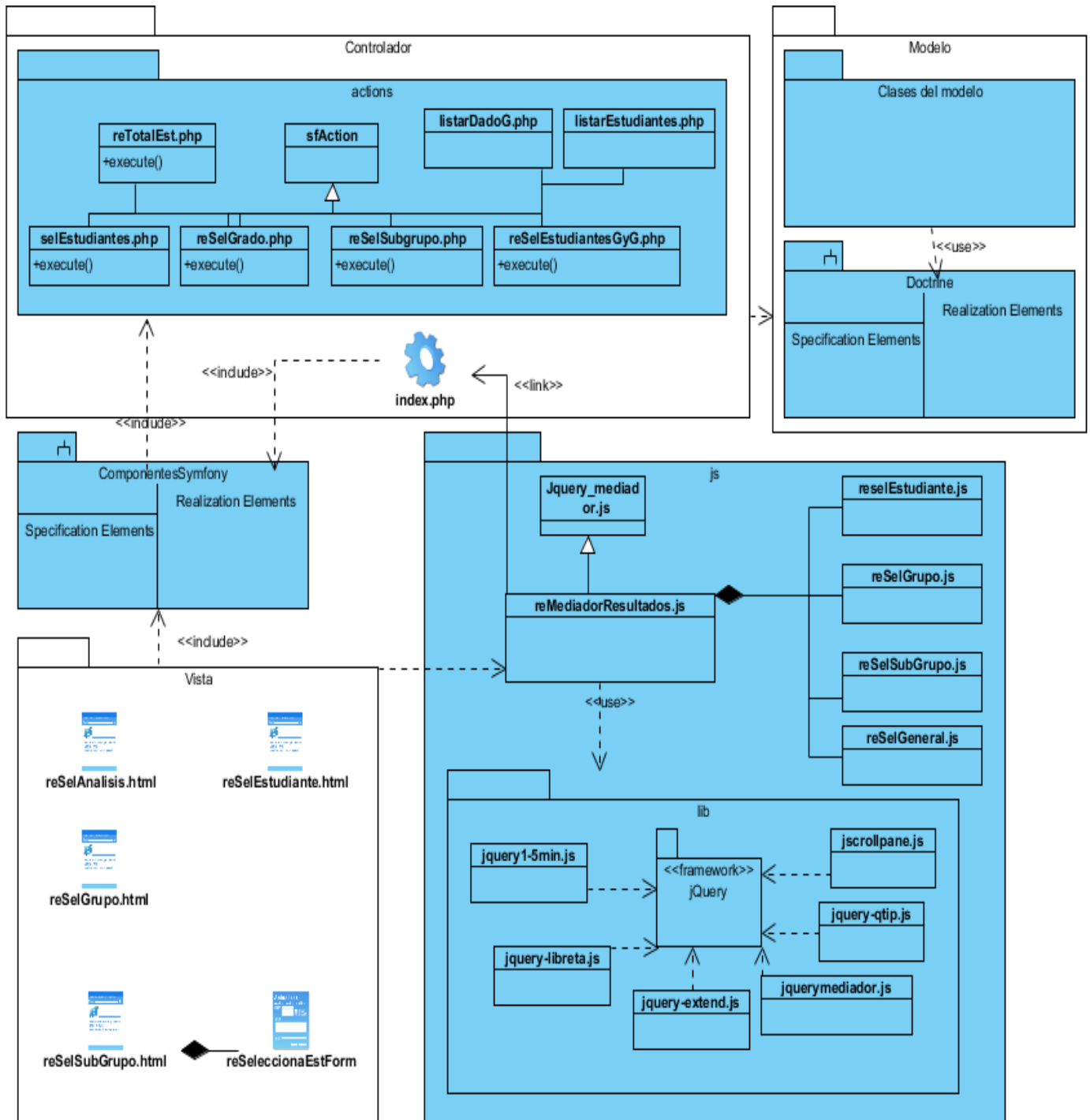


Figura 10 Diagrama de clases del diseño CU_Configurar tipo de análisis.

Módulo Resultados

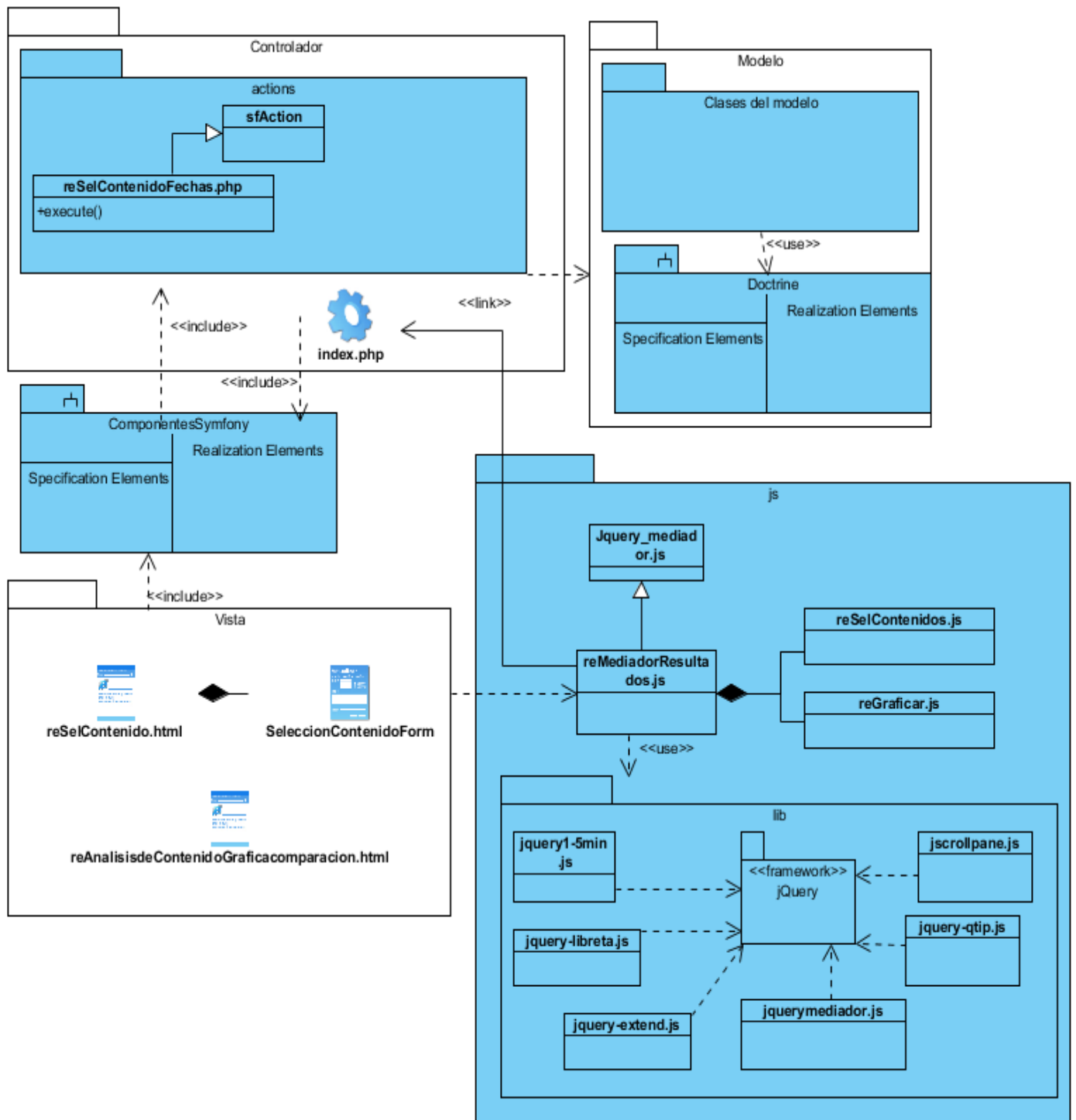


Figura 11 Diagrama de clases del diseño CU_Realizar análisis de contenidos

Módulo Resultados

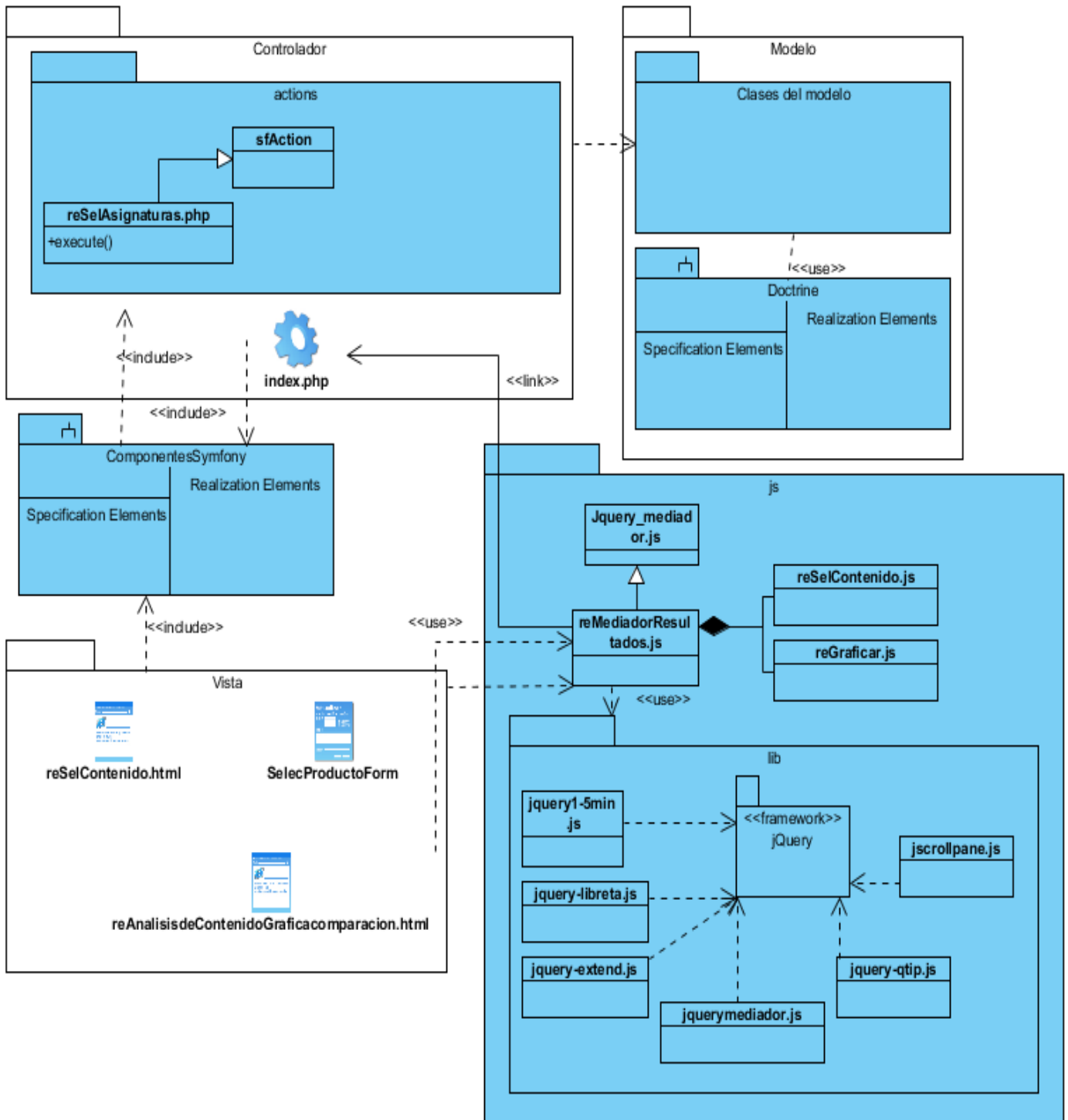


Figura 12 Diagrama de clases del diseño CU_ Realizar análisis integral

Módulo Resultados

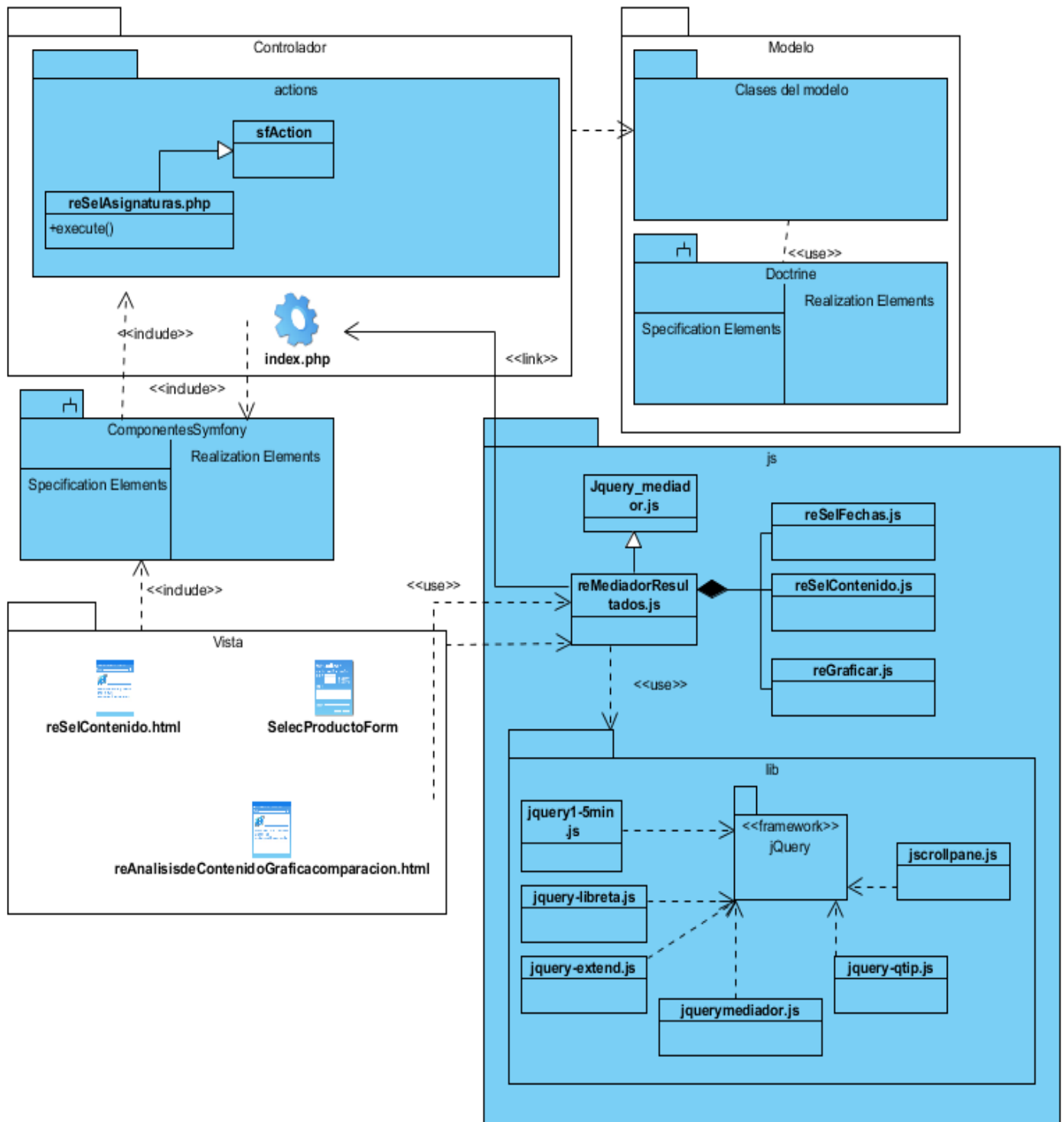


Figura 13 Diagrama de clases del diseño CU_Realizar historial del estudiante

Módulo Resultados

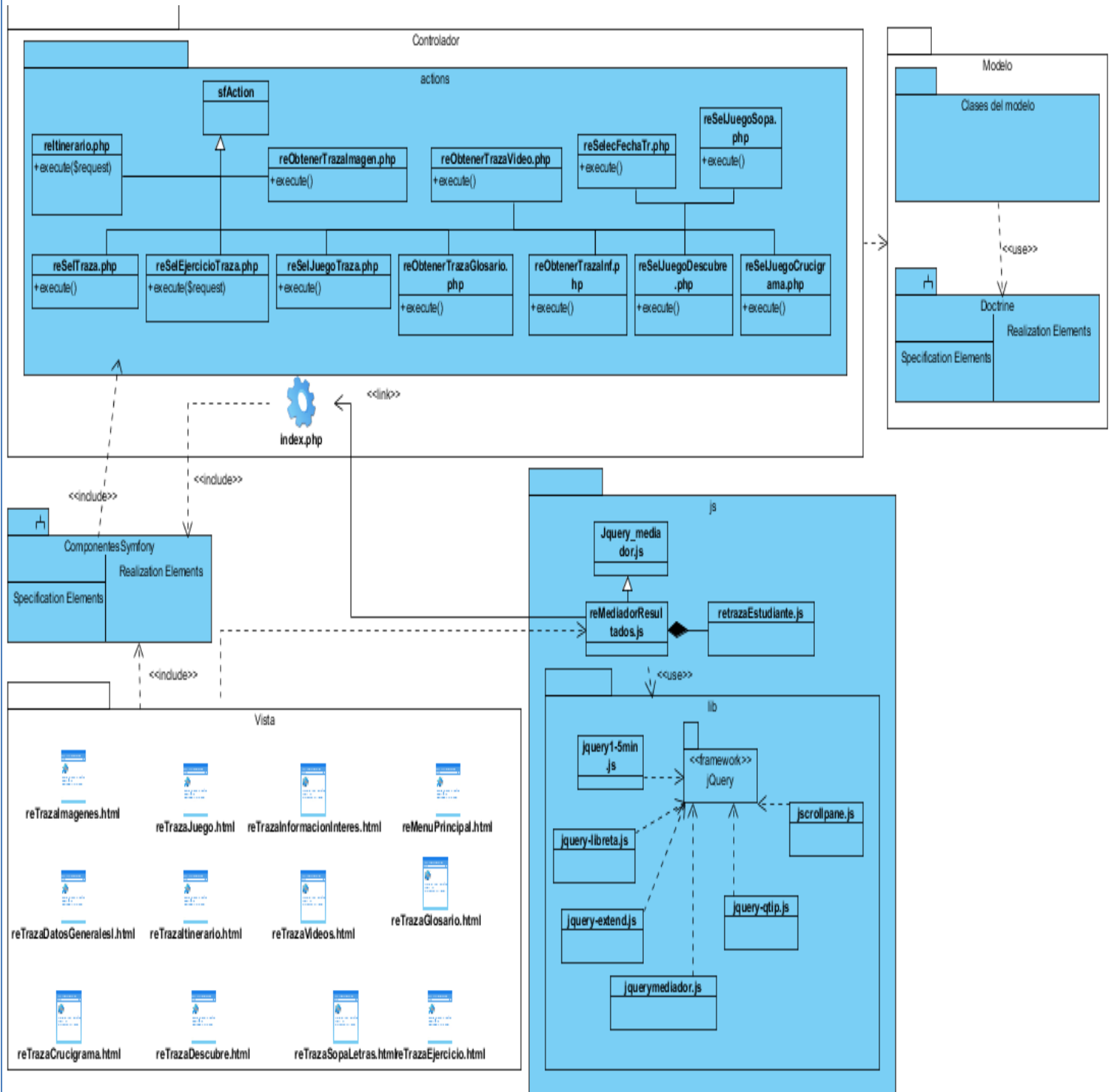


Figura 14 Diagrama de clases del diseño CU_Consultar trayectoria del estudiante

3.5 Diseño de la base de datos

El diseño de la base de datos juega un papel fundamental para cualquier aplicación que gestione información, sobre todo para las aplicaciones web. El propósito de este flujo de trabajo es estructurar la base de datos, controlando que la información no presente redundancia y no existan datos innecesarios que no cumplen ningún objetivo al ser guardados. (22)

3.5.1 Modelo Entidad-Relación

El modelo de datos se utiliza para describir la estructura lógica y física de la información persistente gestionada por el sistema. Este artefacto describe las representaciones lógicas y físicas de datos persistentes utilizados por la aplicación. (11)

El Framework utilizado en el desarrollo del módulo (Symfony) trae incorporado dos potentes ORM (Object Relation Mapper): Doctrine y Propel, con el fin de poder manipular los datos con mayor facilidad. Doctrine está escrito en PHP y proporciona persistencia para objetos PHP. Este ORM es utilizado en el módulo y está por encima de la capa de abstracción a la base de datos. Una de sus características es la posibilidad de escribir consultas a la base de datos a partir del tratamiento con objetos en PHP. (21)

En el actual desarrollo, la base de datos que se representa está vinculada a los datos de los estudiantes que interactúan con la aplicación, también se presenta los datos referentes a los módulos Ejercicio, Mediateca y Juegos, quedando plasmadas las relaciones entre ellos para mantener la organización, el almacenamiento y control de acceso de la información que se maneja, de una forma eficiente.

A continuación se muestra el Modelo Entidad-Relación referente a la base de datos del sistema a desarrollar.

Módulo Resultados

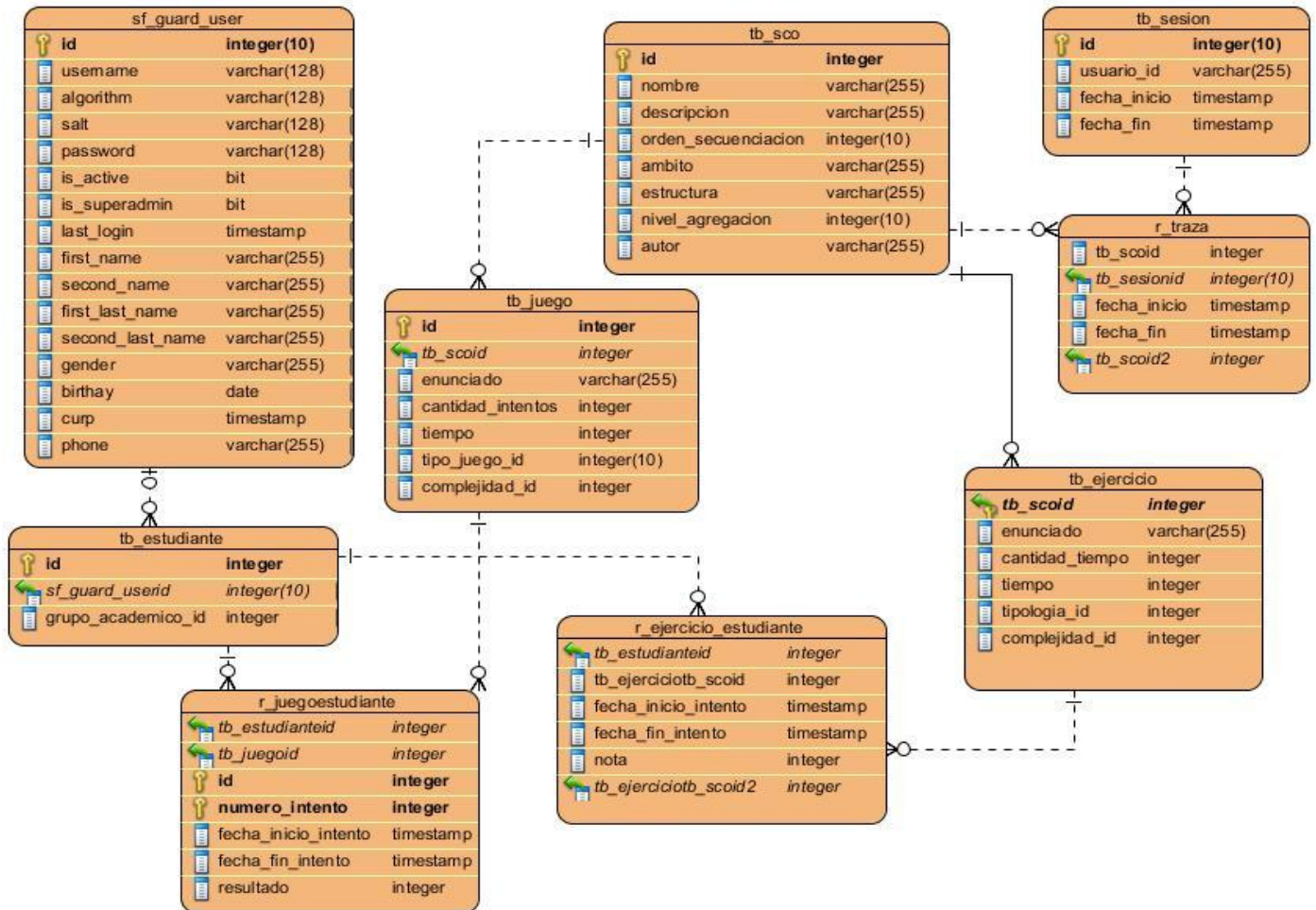


Figura 15 Modelo Entidad-Relación

Las descripciones de cada tabla del Modelo Entidad-Relación se localizan en los anexos del trabajo, específicamente en el [Anexo 5](#).

3.6 Conclusiones

En el capítulo se generaron diferentes artefactos, como resultado del flujo de trabajo análisis y diseño. En tal sentido se documentaron los diagramas de clases del análisis, proporcionando un mayor entendimiento de los requisitos. A través del modelo de diseño se logró preparar el sistema para la posterior implementación y prueba. La descripción de la base de datos contribuyó a establecer una relación directa

Módulo Resultados

con las entidades que aparecen en dicho modelo, en aras de lograr una implementación exitosa del módulo Resultados.

Capítulo 4: Implementación y Pruebas

4.1 Introducción

En el capítulo que se presenta a continuación se documenta el proceso de implementación y la posterior fase de pruebas del producto final. Para dar cumplimiento a estas tareas, se presenta el diagrama de componentes y el diagrama de despliegue, además se diseñan las pruebas de caja negra, a través de la confección de los casos de prueba.

4.2 Estándar de codificación

Los estándares de codificación son reglas que se utilizan para la escritura del código fuente. Estas permiten asegurarse de que todos los programadores del proyecto trabajen de forma coordinada y en un vocabulario común. El uso de los mismos trae consigo innumerables de ventajas, ya que permite asegurar la legibilidad del código entre distintos programadores, provee una guía para el encargado del mantenimiento o actualización del sistema, con código claro y bien documentado y además facilita la portabilidad entre plataformas y aplicaciones.

4.2.1 Convenciones de nombres (Funciones, Variables Globales, Clases)

Los nombres que se utilizan son cortos, y de forma abreviada, siempre que su contexto sea específicamente local y su lectura sea intuitiva. Para los nombres se establecen las siguientes reglas:

Funciones y métodos

Se utiliza la notación Camel que consiste en escribir los identificadores con la primera letra de cada palabra en mayúsculas y el resto en minúscula: EndOfFile. Las funciones y métodos se nombran usando el formato CamelCase (notación camello) que es una variante de la notación Camel donde la primera letra es minúscula.

• Clases

Los nombres de las clases comienzan con el sufijo del módulo en este caso del módulo Resultados seguidos del formato CamelCase ejemplo reMenuPrincipal.

Las clases que se generan en el Modelo-Entidad-Relación comienzan con:

- Tb_: las clases de entidad.

- Nom_: las clases de nomencladores.
- R_: las clases que representan relaciones muchos-a-muchos.

4.2.2 Indentación y espacios en blanco

- **Alineación y longitud de líneas**

Indentar con dos espacios, sin tabulador, para que cualquier editor de texto reconozca correctamente la indentación. Utilizar líneas entre 75-80 caracteres de longitud, de esta manera se maneja mejor algunas herramientas y terminales.

- **Condicionales if, for, while, switch**

Debe existir un espacio entre la palabra clave y el paréntesis de apertura. Esto mejora la legibilidad y disminuye la posibilidad de errores lógicos al agregar nuevas líneas de código.

- **Líneas plegadas**

Cuando una expresión no cabe en una línea simple debido a su extensión se divide en más de una línea, siguiendo las siguientes precisiones:

- Dividir después de una coma.
- Dividir después de un operador.
- Alinear la nueva línea al inicio de la expresión en el mismo nivel que la línea anterior.

4.2.3 Comentarios

El uso de comentarios en línea se utiliza para facilitar la comprensión del código, sobre todo en procedimientos complejos. Los comentarios pueden ser con fin documental o para ayudar a la memoria del programador. Se recomienda utilizar el estilo (`/* */`).

En el caso de los bloques de comentarios tienen solo un asterisco al inicio de cada línea exceptuando la primera. Ejemplo:

```
/**  
 *Comentario  
*/
```

Módulo Resultados

Los comentarios cortos pueden aparecer en una sola línea alineados al mismo nivel del código que lo sigue, además debe estar precedido de una línea en blanco. Ejemplo:

```
/* Comentario*/
```

4.3 Modelo de implementación

El flujo de trabajo de Implementación en la fase de Elaboración describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo éstos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Este flujo está fuertemente determinado por el lenguaje de programación y va encaminado a implementar la arquitectura que se ha definido (29).

4.3.1 Diagrama de despliegue

Un diagrama de despliegue representa las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución. Se puede decir además que, un diagrama de despliegue es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. (29)

A continuación se muestran cómo están distribuidos los nodos y dispositivos para este producto. En este caso están distribuidos en dos variantes, según lo solicitado por el cliente.

- 1- Instalar localmente el producto en la PC del usuario el cual va a tener instalado el servidor web y la base de datos. La impresora se utiliza como dispositivo para posibilitar la impresión de documentos.

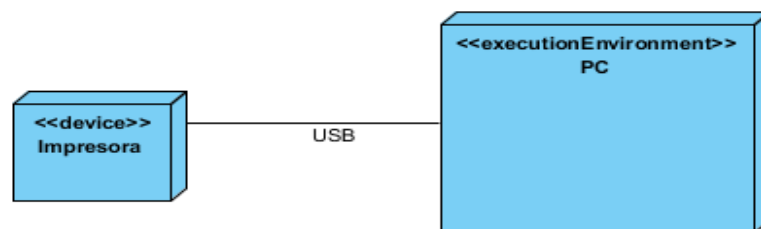


Figura 16 Diagrama de despliegue variante 1

Módulo Resultados

2- El Diagrama de despliegue 2 está compuesto por tres nodos, donde en el nodo Servidor se tiene una computadora como servidor central donde reside el servidor web conjunto con el de bases de datos. El nodo PC cliente, es en el cual el usuario va a poder interactuar con la aplicación y para ello debe tener instalado algunos de los navegadores propuestos en los requerimientos no funcionales del software y la impresora como dispositivo utilizado por el sistema para posibilitar la impresión de documentos.

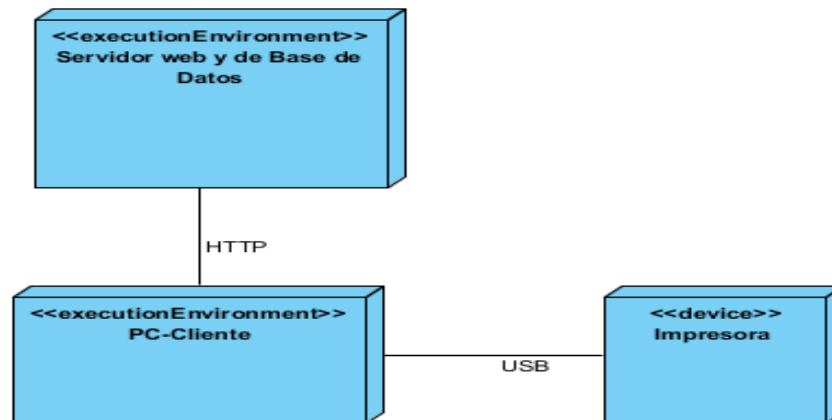


Figura 17 Diagrama de despliegue variante 2

4.3.2 Diagrama de componentes

Un diagrama de componentes muestra un conjunto de elementos de los modelos como componentes, subsistemas de implementación y sus relaciones. El uso más importante es mostrar la estructura de alto nivel del modelo de implementación, especificando los subsistemas de implementación y sus dependencias a la hora de importar código, organizándolos en capas (9).

Módulo Resultados

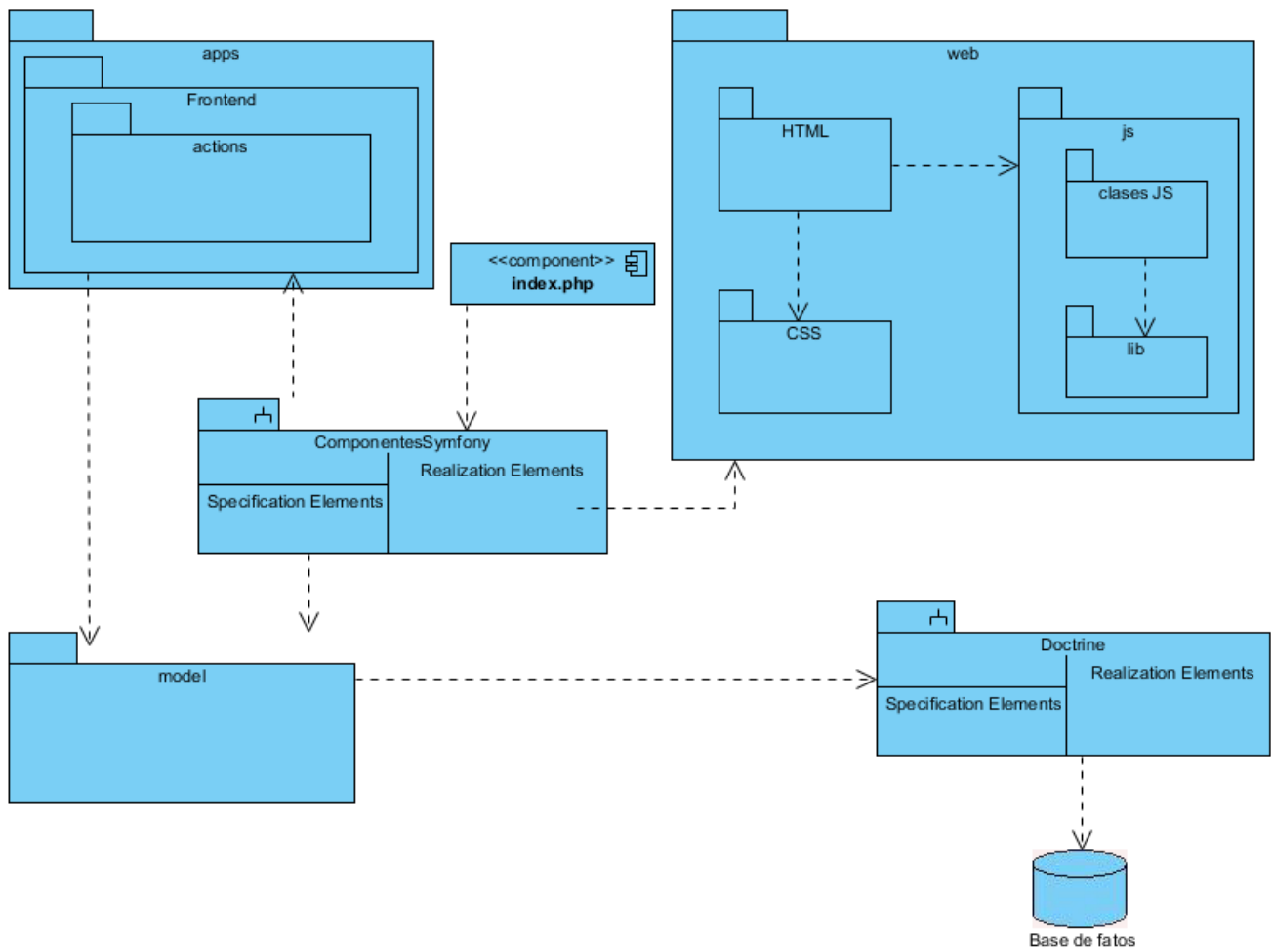


Figura 18 Diagrama de componentes del módulo Resultados

Módulo Resultados

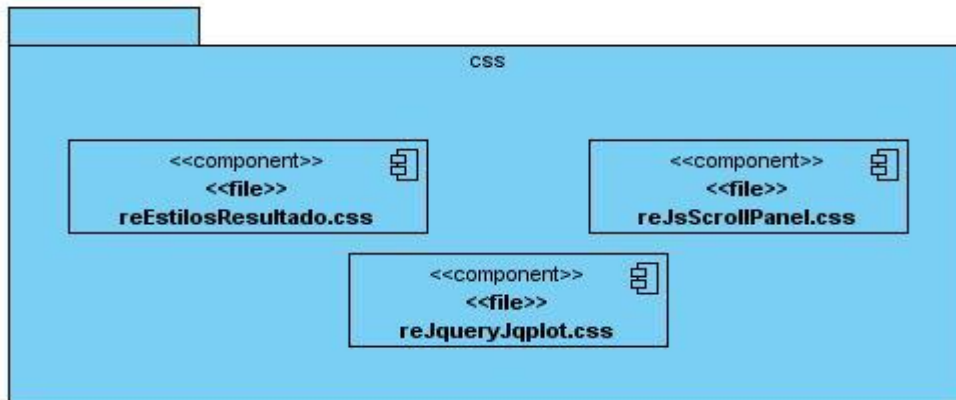


Figura 19 Diagrama de componentes del paquete CSS

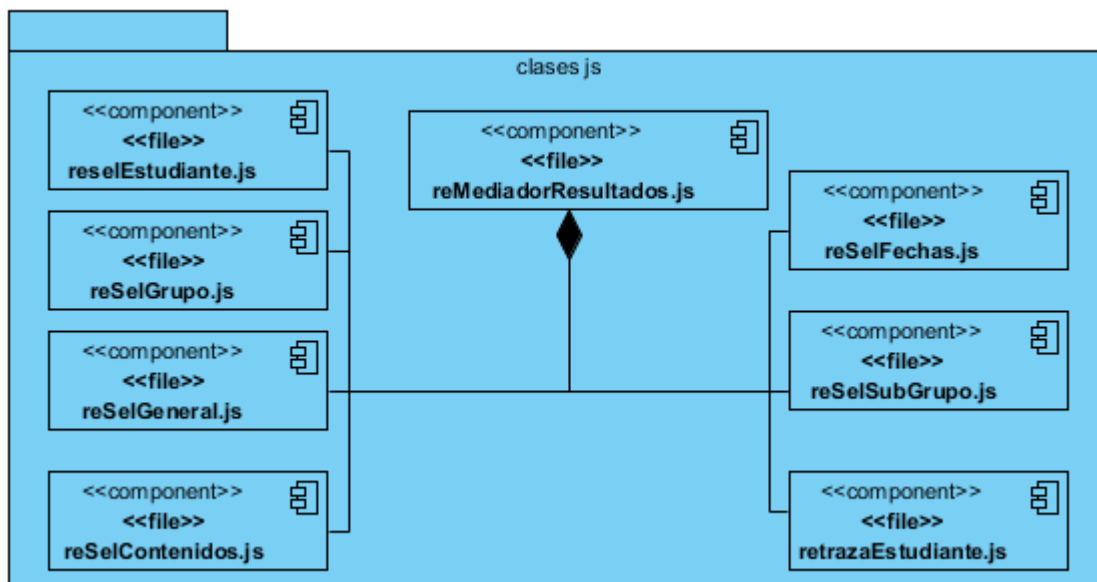


Figura 20 Diagrama de componentes del paquete vista

Módulo Resultados

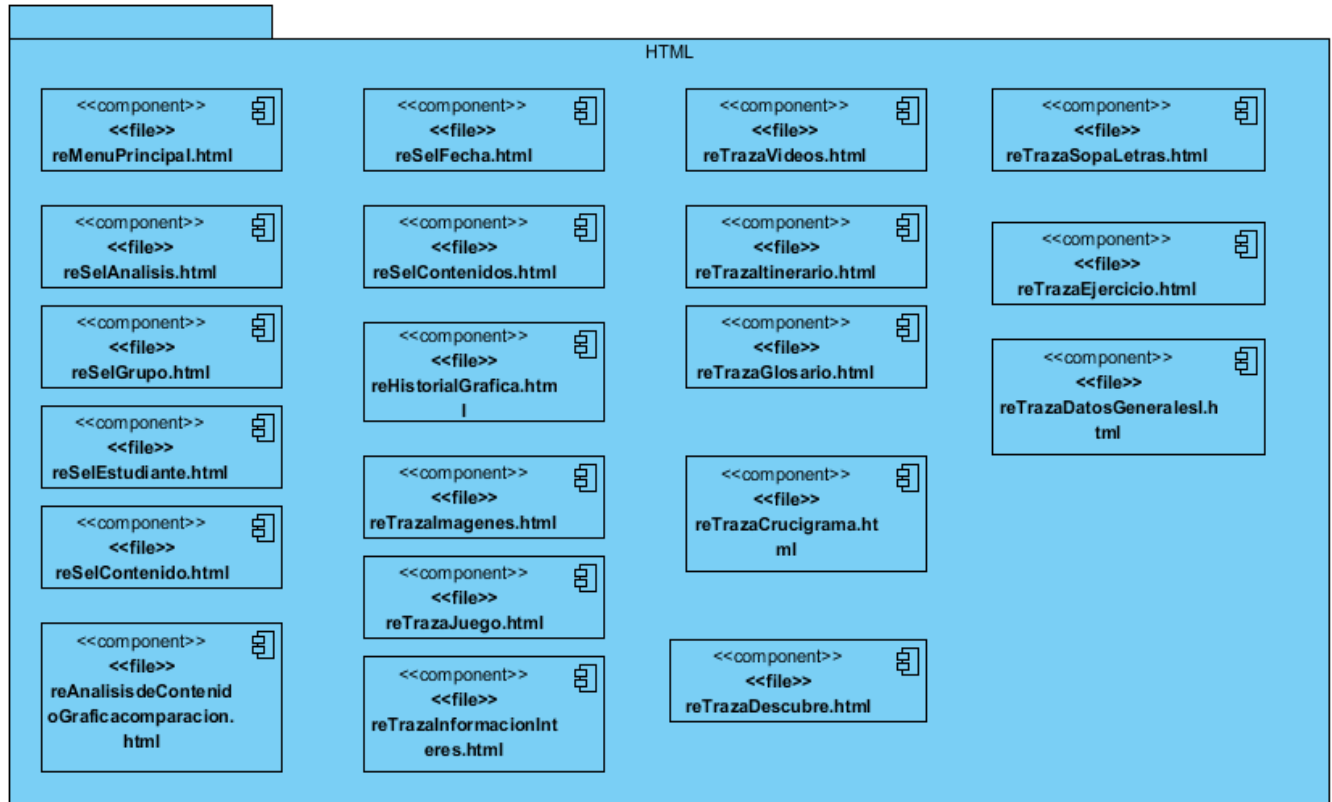


Figura 21 Diagrama de componentes del paquete HTML

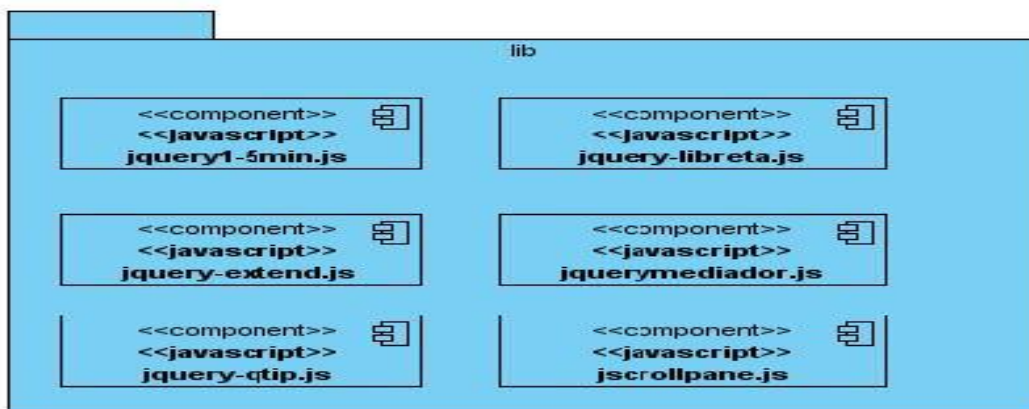


Figura 22 Diagrama de componentes del paquete lib.

Módulo Resultados

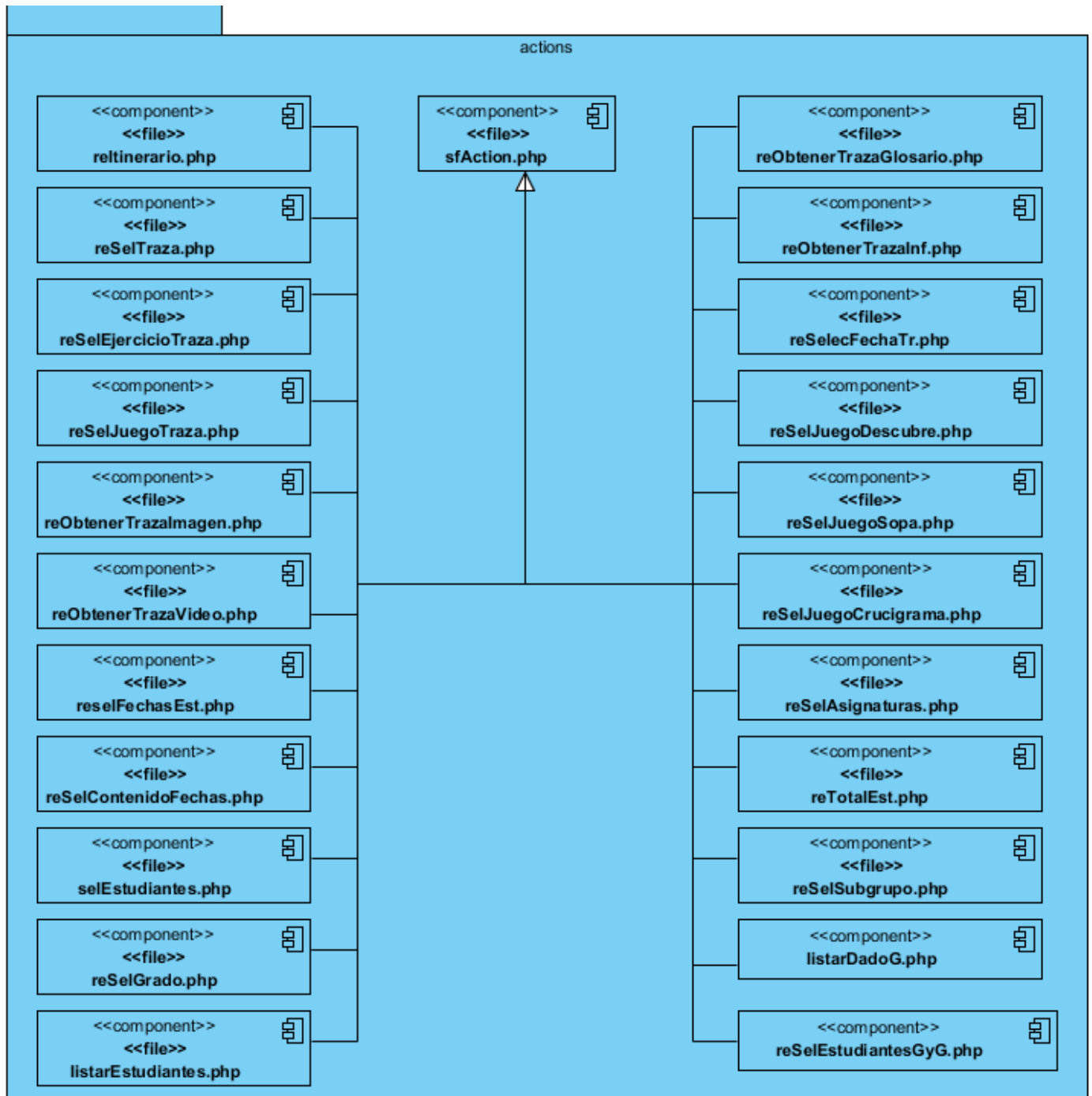


Figura 23 Diagrama de componentes del paquete actions.

Módulo Resultados

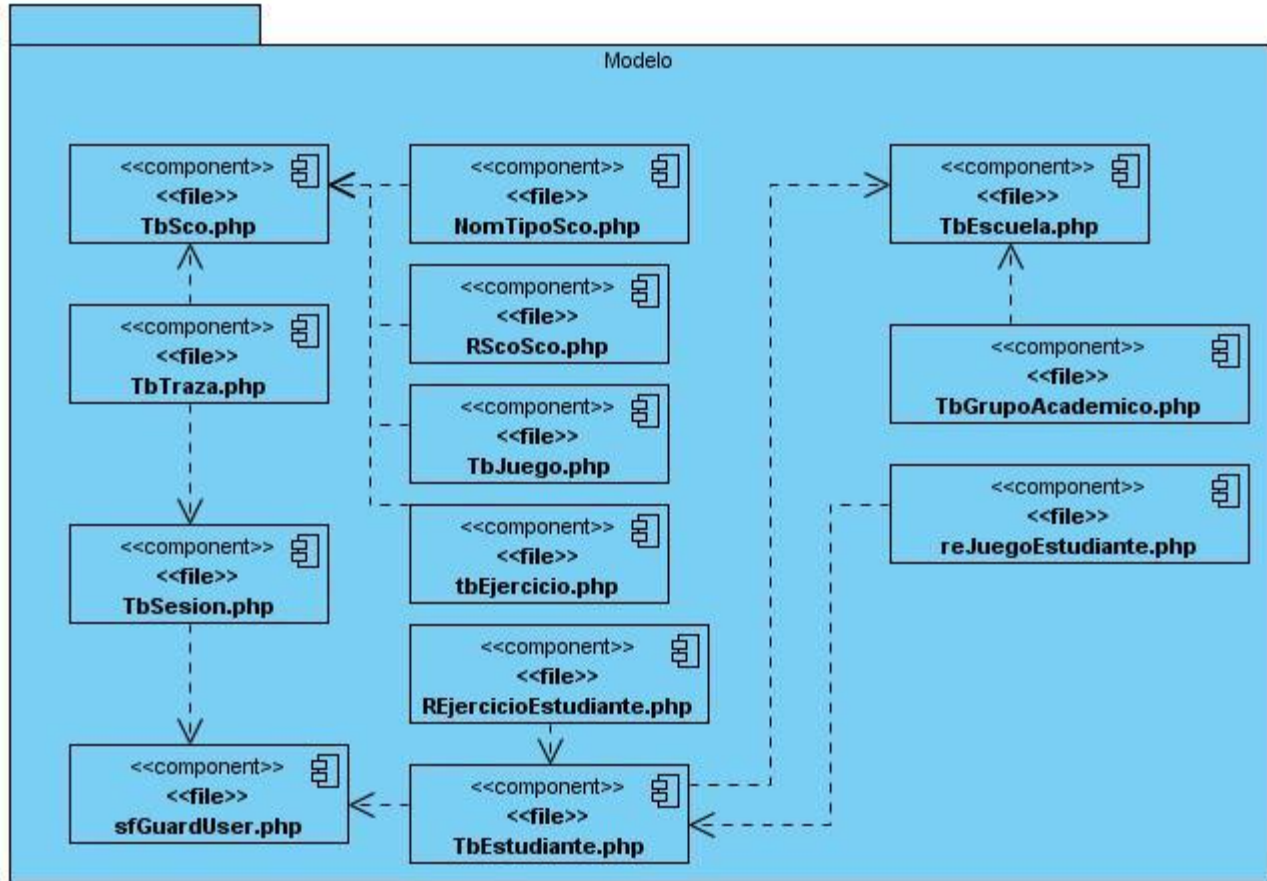


Figura 24 Diagrama de componentes del paquete Modelo

4.4 Pruebas de software

Para que la solución se considere lista es muy importante tener una vía para validar que lo que se realizó realmente es correcto o cumple con las expectativas del cliente o usuario final. Por esta razón en el proceso de desarrollo de software se define una etapa de pruebas, con el objetivo de analizar si el producto final fue implementado correctamente. Para ello es necesario conocer las funcionalidades específicas y determinar si realmente cumplen el objetivo para lo cual fueron diseñadas, demostrando que las funciones son operativas. De igual manera se pueden realizar pruebas internas en el código, que requieren del conocimiento de la estructura interna del programa y aseguran que todos los elementos existentes en el mismo cumplan una función. Todo esto se resume en las pruebas de caja negra y las pruebas de caja blanca. (30)

Las pruebas de caja blanca son realizadas mediante un seguimiento del código fuente, analizando detenidamente cada bloque de instrucciones, mientras que las pruebas de caja negra se llevan a cabo sobre la interfaz del software, verificando que las funcionalidades son operativas. Partiendo de lo planteado se decide realizar pruebas de cajas negras con el objetivo de validar que la versión 2.0 del módulo Resultados cumple con los requerimientos definidos por el cliente.

4.4.1 Pruebas de caja negra

Las pruebas de caja negra se centran fundamentalmente en los requisitos funcionales del software. Estas se realizan sobre la interfaz de la aplicación, con el objetivo de demostrar que las funciones son correctas, que la entrada se acepta de forma adecuada y que se produce una salida correcta. A través de las mismas es muy usual encontrar funciones incorrectas o ausentes, errores de interfaz, errores de rendimiento, así como errores de inicialización o terminación. (7)

Las pruebas de aceptación son pruebas de caja negra ejecutadas por el cliente o el equipo de desarrollo, con el objetivo de garantizar que los requerimientos han sido cumplidos y descartar los posibles errores. (12) Debido a la importancia de estas pruebas, para la solución implementada se definen casos de prueba por casos de uso ([Anexo 6](#)), con el objetivo de comprobar que cumple con las funcionalidades descritas.

Para la realización de las pruebas de cajas negras, se utilizan varias técnicas, dentro de ella se encuentran: Técnica de la Partición de Equivalencia, Análisis de Valores Límites y Grafos de Causa-Efecto. La técnica que se seleccionó para realizar los casos de prueba es la variante de particiones equivalentes, cuyo objetivo fundamental es dividir el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Esta técnica consta de dos pasos fundamentales: (7)

1. Identificación de las clases de equivalencia.
2. Identificación de casos de prueba.

En el primer paso, las clases de equivalencia son identificadas tomando cada condición de entrada (generalmente una oración o una frase en la especificación) y repartiéndola en dos o más grupos, cada combinación será un escenario del caso de uso. En el segundo paso se identifican las variables y las clases de equivalencia para la confección de los casos de prueba.

Módulo Resultados

Los casos de prueba son un conjunto de condiciones o variables bajo los cuáles se determina si las funcionalidades de la aplicación son correctas. Estos incluyen una descripción de la funcionalidad que se probará, que es tomada de los requisitos o de los casos de uso. Para la realización de los casos de pruebas se utiliza un número de datos de entrada y salida que apoyan a la ejecución de los mismos y que permiten que el sistema se ejecute en todas sus variantes. (30)

A partir de los casos de prueba se realizaron tres iteraciones, donde se utilizaron los métodos y técnicas de pruebas expuestas anteriormente, en el que se encontraron un total de 15 No conformidades, que fueron resueltas, aportando mayor calidad al módulo Resultados. A continuación se muestra una tabla con el resumen de las tres iteraciones realizadas y un gráfico de los errores encontrados.

Tipos No conformidades	1ra iteración	2da iteración	3ra iteración
Formato(A y D)	3	1	
Error técnico			
Ortografía		2	
Redacción	4		
Validación			
Opciones que no funcionan			
Error de interfaz		1	
Error de idioma			
Funcionalidad			
Excepciones	3		
Otros			
Recomendación			
Total	10	5	

Tabla 9 Resumen de No conformidades encontradas

Módulo Resultados

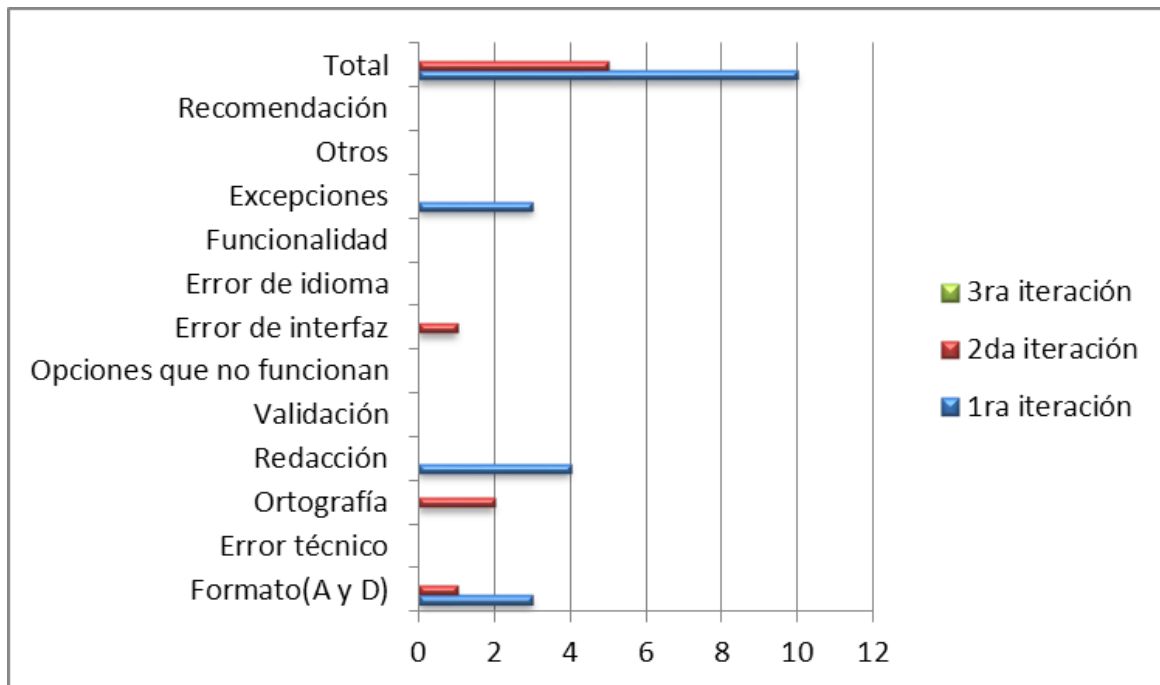


Figura 25 No conformidades detectadas durante las pruebas

4.5 Conclusiones

En el presente capítulo se generaron diferentes artefactos como resultado de la implementación del módulo Resultados, quedando confeccionado el diagrama de componentes y el diagrama de despliegue. Se realizaron además los casos de prueba para llevar a cabo las pruebas al módulo Resultados y asegurando el correcto funcionamiento del mismo.

Conclusiones generales

Como resultado del trabajo desarrollado se arribaron a las siguientes conclusiones:

- Las soluciones similares existentes estudiadas no soportan el modelo pedagógico establecido para la nueva versión de la colección El Navegante y no cumplen con los requerimientos tecnológicos definidos en la arquitectura de la Colección, por lo que se demostró la necesidad de desarrollar una aplicación utilizando herramientas libres y multiplataforma, que fuese capaz de mostrar un conjunto de informaciones actualizadas de los usuarios que interactúan con la Colección.
- Mediante el análisis, diseño e implementación de la versión 2.0 del módulo Resultados, fue posible obtener una aplicación que muestra la información de los estudiantes en su interacción con los restantes módulos de la Colección, facilitando por esta vía que el profesor pueda emitir un mejor criterio acerca del desempeño de los mismos.
- La solución propuesta fue validada utilizando pruebas de caja negra, basadas en los casos de prueba, aportando una mayor calidad al producto. Todo esto permitió erradicar deficiencias encontradas y entregar un módulo que cumple con las funcionalidades requeridas por el cliente.

Recomendaciones

Aún con el cumplimiento del objetivo de la aplicación y teniendo como base el estudio efectuado durante la realización de esta investigación, se proponen como recomendaciones:

- Incorporar nuevas funcionalidades que permitan realizar comparaciones de resultados entre estudiantes, grupos o instituciones.
- Agregar otras tipologías de gráficas, con el objetivo de facilitarle al profesor el análisis de los resultados obtenidos por los estudiantes.

Referencias bibliográficas

1. RIMED. Características-generales-de-la-coleccion-multisaber. [En línea] <http://blogs.rimed.cu/multisaber>.
2. RIMED. El navegante. [En línea] <http://rimed.cu/navegante>.
3. **Matos, Walfrido Terrero**. Procedimientos metodológicos para el uso de los software educativos de la Colección Futuro. Cuba.
4. RIMED. Preuniversitario institucional coleccion-futuro. [En línea] <http://www.rimed.cu>.
5. Sadhea. [En línea] <http://www.sadhea.rimed.cu>.
6. Sadhea Web. [En línea] <http://www.enfermeria.sld.cu/docencia/sadhea.html>.
7. **Pressman, Roger S**. Ingeniería del Software, un enfoque práctico.
8. **Tarazona, Ivon y Gómez, Oriana**. Uso de UML en el modelado de Datos. [En línea] <http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/download/exposiciones2005-2006/uml.pdf>.
9. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James**. El Proceso Unificado de Desarrollo de Software. Madrid: Addison Wesley, 2000.
10. **Dávila, Vela José A**. Ambientes de Desarrollo de Software Basado en Componentes. [En línea] <http://www.cimat.mx/Eventos/seminariotecnologias08/javd.pdf>
11. **Martínez, Alejandro y Martínez, Raúl**. Proceso Unificado de Desarrollo. Escuela Politécnica Superior de Albacete – Universidad de Castilla la Mancha. [En línea] <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-Guia%20RUP.pdf>
12. **Orallo, Enrique Hernández**. El Lenguaje Unificado de Modelado (UML).
13. Visual Paradigm for UML. [En línea] <http://www.visual-paradigm.com>.
14. Lenguajes de Programación. [En línea] <http://www.lenguajes-de-programacion.com>.
15. **Kennedy, Bill y Musciano, Chuck**. HTML La Guía Completa. Mexico: Litografica Ingramex, 1999.
16. CSS2. [En línea] <http://www.w3c.es/divulgacion/guiasreferencia/css21>
17. **Issi, Lázaro**. Java Script. España: ANAYA MULTIMEDIA, 2002.
18. **Monso, Martín Ramos**. Programación PHP. Argentina: MP. Ediciones SA, 2005.
19. **Warden, Shane y Shore, James**. The Art of Agile Development. 2007.

Módulo Resultados

20. JQuery. [En línea] <http://www.jquery.com>
21. **Potencier, Fabien y Zaninotto, Francois.** Symfony, la guía definitiva. 2008.
22. Servidores Web. [En línea] <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente>.
23. **Kabir, Mohammed J.** Servidor Apache 2. Anaya Multimedia, 2003.
24. **Alvarez, Sara.** Sistemas gestores de bases de datos. [En línea] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
25. PostgreSQL. [En línea] <http://www.postgresql.org>.
26. Netbeans. [En línea] <http://www.netbeans.org>.
27. **Guerrero, L.A.** Análisis y Diseño Orientado a Objetos. [En línea] <http://www.dcc.uchile.cl/~luguerre/cc40b/clase4>
28. **Marca, Hugo.** Diagrama de Despliegue. [En línea] <http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc>.
29. **Sommerville, Ian.** Ingeniería del Software. Pearson Educación. Enero, 2005.
30. **Binder, Robert.** Object-Oriented Software Testing. Communications of the ACM. September, 1994.

Bibliografía

1. **Alejandro, Novoa Vazquez.** [En línea] http://www.ecured.cu/index.php/Colecci%C3%B3n_Multisaber.
2. **Amo, Alonso y Martínez, Loïc Normand.** Introducción a la Ingeniería de Software. Delta Publicaciones, 2005.
3. Colección de Software Educativo. [En línea] <http://blogs.rimed.cu/multisaber/2009/07/03/caracteristicas-generales-de-la-coleccion-multisaber>.
4. **Cuevas, Carlos.** Hacia una Clasificación de la Computación en la Enseñanza de las Matemáticas. Grupo Editorial Iberoamérica S.A. de C.V, 1998.
5. **Duval, Raymond.** Sémiosis y Pensamiento Humano, Registros Semióticos y aprendizajes intelectuales. Santiago de Cali, Colombia, 1999.
6. Ecured. [En línea] http://www.ecured.cu/index.php/Colecci%C3%B3n_Multisaber.
7. **GALLEGO, JUAN PABLO GOMEZ.** Fundamentos de la Metodología RUP. Universidad Tecnológica de Pereira. 2007.
8. **IPN-UPICSA.** Visual Paradigm For Uml. [En línea] <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
9. **Jesús, Areba Barranco.** Metodología del análisis estructurado de sistemas. 2001.
10. **Matedu.** [En línea] <http://www.matedu.cinvestav.mx/ccuevas/SoftwareEducativo.htm>.
11. Metodología XP Vs. Metodología Rup . Metodología XP Vs. Metodología Rup. [En línea] <http://metodologiexpvsmetodologiarup.blogspot.com>.
12. publicalpha. [En línea] <http://publicalpha.com/%C2%BFque-es-el-software-educativo>.
13. Scribd. [En línea] <http://www.scribd.com/doc/2050925/metodologias-de-desarrollo-software>.
14. SOFTWARE, DEFINICIÓN Y CARACTERÍSTICAS. [En línea] <http://tecnopeducativa.blogspot.com/2007/03/software-definicion-y-caracteristicas.html>.

Glosario de términos

AJAX (JavaScript Asíncrono y XML): No es una tecnología por sí misma, es un término que describe un nuevo modo de utilizar conjuntamente varias tecnologías existentes. Esto incluye: HTML o XHTML, CSS, JavaScript, DOM, XML, XSLT, y el objeto XML Http Request. Cuando estas tecnologías se combinan en un modelo AJAX, es posible lograr aplicaciones web capaces de actualizarse continuamente sin tener que volver a cargar la página completa. Esto permite crear aplicaciones más rápidas y con mejor respuesta a las acciones del usuario.

Aplicación web: son aquellas aplicaciones informáticas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una red interna mediante un navegador web.

Efectividad: Porcentaje de eficiencia en la realización de ejercicios en el tema calculado a partir de la fórmula siguiente:

$$E = ((\text{Cantidad_R} + 2 * \text{Cantidad_B}) * 100) / (\text{Total} * 2).$$

Cantidad_R: Cantidad de evaluaciones Regular que tiene un estudiante.

Cantidad_B: Cantidad de evaluaciones de Bien que tiene un estudiante.

Total: Total de ejercicios realizados.

Hipermedia: Hipertexto que no se limita a documentos escritos, sino también otros medios como imágenes, audio y vídeo.

HTTP: Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto), es el método más común de intercambio de información en la World Wide Web, mediante el cual se transfieren las páginas web a un ordenador.

Multimedia: Es el término que se utiliza para referirse a cualquier objeto o sistema que maneja múltiples medios de expresión (físicos o digitales) para presentar o comunicar información. De allí la expresión

Módulo Resultados

"multi-medios". Los medios pueden ser variados, desde texto e imágenes, hasta animaciones, sonidos y videos.

Multiplataforma: Programa o aplicación que puede utilizarse sin inconvenientes en distintas plataformas de hardware y sistemas operativos.

Navegador web: Un navegador o navegador web (del inglés, web browser) es un programa que permite visualizar la información que contiene una página web, ya esté alojada en un servidor dentro de la World Wide Web o en uno local.

ORM: Es una técnica de programación que permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de la base de datos pasan a ser clases y los registros objetos que se pueden manejar con facilidad.

Recursos multimedia: es la integración de video, imágenes, animación, texto y sonido. Estos constituyen un elemento crucial en la calidad de los software educativo, lo que posibilita un gran apoyo al proceso de enseñanza – aprendizaje.

Software educativo: es un conjunto de elementos informáticos, diseñado para ser utilizado en el proceso de enseñanza – aprendizaje. Se caracteriza por ser una aplicación altamente interactiva y por contener recursos multimedia, como videos, sonidos e imágenes; además, puede contener información sobre diferentes materias, diccionarios especializados, ejercicios y juegos instructivos que apoyan los diagnósticos y las funciones de evaluación.

URL: La sigla URL (en inglés "Uniform Resource Locator") se refiere a la dirección única que identifica a un sitio web en Internet.