

*“Análisis, Diseño e Implementación del módulo Resultados de la
Colección El Navegante en su versión multiplataforma”*

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

Autores:

Jorge Morgado Novo

Erianne Carmenate Cabrera

Tutores:

Ing. Ismael Armando Nodarse Mora

Ing. Ana Margarita Rojas Riverón

Ciudad de La Habana, mayo de 2010.

“Año 52 de la Revolución”.

No hay más que asomarse a las puertas de la tecnología y la ciencia contemporánea para preguntarnos si es posible vivir y conocer ese mundo del futuro sin un enorme caudal de preparación y conocimiento.

Fidel Castro

AGRADECIMIENTOS

Durante el desarrollo de este trabajo de diploma hemos recibido la ayuda desinteresada de muchas personas que han contribuido a nuestra formación. Reciban nuestros más sinceros agradecimientos:

-A la Revolución y a Fidel Castro por ser gestor y creador de este proyecto de ensueño:

-A nuestros tutores Ana Margarita e Ismael por su orientación constante, la crítica oportuna y el aliento sistemático.

-A los profesores Liana Isabel, Maifyn y Agustín por asesorarnos.

-A nuestros amigos y compañeros de aula por estar siempre a nuestro lado, por darnos su apoyo y ayuda incondicional.

-A todos nuestros profesores de la carrera por contribuir con sus vastos conocimientos, al cumplimiento de este sueño.

-A nuestros vecinos, compañeros de apartamento, amigos de la universidad y los de la casa, por preocuparse por nosotros y darnos su apoyo.

Erianne

-A mis padres por su apoyo, por ser mi soporte y razón de ser, por haber formado una familia que me ha dado lo mejor del mundo; han sacrificado todo por mí, por ser las personas más importantes en mi vida, sin ellos este sueño no hubiese sido posible.

-A mis hermanos Arianna y Carlos, por su comprensión, confianza, apoyo incondicional y ser mi inspiración. Ejemplo a seguir.

-A mis sobrinitos Elbys, Yan Carlos y Kevin, por darme tanto amor y cariño, por cada momento de alegría que me dan cuando estoy a su lado y por ser los peques que más quiero en todo “el universo”, quisiera ser su ejemplo y que se sientan orgullosos de su” tatica”.

-A mis cuñados Orelbys y Mariela, por ser dos hermanos más, por su apoyo, cariño y ser además miembro de un gran tesoro llamado familia.



-A mi novio y compañero de tesis Jorge, por su paciencia como novio y compañero de tesis, por haber estado conmigo minuto a minuto, al convertir los malos momentos en dulces recuerdos.

-A mi tía Teresa por quererme como una hija y estar ahí apoyándome siempre que la he necesitado.

-A mis suegros y cuñados por acogerme como un miembro más de la familia.

Jorge

-A mis padres Mirta y Jorge por su apoyo y cariño.

-A mis hermanos Desiré y Yulien por su amor y apoyo.

Le agradecemos a toda nuestra familia por su cariño y comprensión, sin cada uno de ellos no seríamos las personas que hoy somos.

A todos gracias...



DEDICATORIA

Hay personas que por su ejemplo y su andar cotidiano marcan pautas y huellas indelebles. A ellos dedicamos esta investigación.

Erienne

A mis padres, esas dos personitas maravillosas que me dieron el don de la vida, por tanto amor, apoyo y sacrificio cuando yo dudaba. Por guiarme siempre hacia el camino correcto, darme fuerzas para seguir adelante, ayudándome a pasar los obstáculos que me ha puesto la vida. A mis hermanos, sobrinos y cuñados por complementar mi felicidad con su existencia.

Jorge

A mis padres por su apoyo, comprensión y cariño durante toda mi vida. Mis hermanos por darme siempre el consejo apropiado y a mis abuelos por estar siempre al tanto de cada prueba, evaluación y cada nuevo paso que daba en mi vida.



DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos 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 _____ del año _____.

Firma del Autor

Jorge Morgado Novo

Firma del Autor

Erienne Carmenate Cabrera

Firma del Tutor

Ing. Ismael Armando Nodarse Mora

Firma del Tutor

Ing. Ana Margarita Rojas Riverón

RESUMEN

Las tecnologías de la información y las comunicaciones han alcanzado grandes avances dado el desarrollo que las mismas manifiestan en la actualidad, al favorecer el aumento de la productividad en las actividades de la sociedad y ser los productos multimedia los que han obtenido una indudable importancia debido a la gran cantidad de audiencia y utilidades que poseen. Estas tecnologías son empleadas en el desarrollo de software, sobre todo en el software educativo. El presente trabajo es una investigación realizada sobre el análisis, diseño e implementación de la colección de software educativo El Navegante, específicamente del módulo Resultados. Está dirigida a desarrollar una aplicación que brinde información de manera consolidada de los reportes de los resultados obtenidos por los estudiantes durante su interacción con la aplicación. Para el desarrollo del módulo se efectúa un análisis de las tendencias, tecnologías y herramientas a utilizar. Se efectúa el levantamiento de los requerimientos especificados por el cliente, el análisis, diseño e implementación de la solución propuesta siguiendo la metodología Proceso Unificado de Desarrollo del Software (RUP) y modelando con UML (Lenguaje Unificado de Modelado). Este módulo permite mostrar la evolución del estudiante en las asignaturas y contenidos especificados mediante gráficas de efectividad así como las consultas de reportes generados por el mismo.



Tabla de contenidos

INTRODUCCIÓN	1
CAPÍTULO 1: Fundamentación teórica	5
1.1 Introducción	5
1.2 Análisis de otras soluciones existentes	5
1.3 Tecnologías y herramientas a utilizar	7
1.3.1 Metodología de desarrollo del software	7
1.3.2 Lenguaje de modelado	10
1.3.3 Herramienta CASE	10
1.3.4 Lenguajes de desarrollo	11
1.3.5 Framework	14
1.3.6 Gestores de bases de datos	17
1.3.7 Tipo de aplicación	18
1.3.8 Servidor web	19
1.3.9 Entorno de desarrollo integrado	20
1.3.10 Patrón Modelo-Vista-Controlador (MVC)	20
1.4 Conclusiones	21
CAPÍTULO 2: Propuesta de la solución	22
2.1 Introducción	22
2.2 Valoración del módulo a implementar	22
2.3 Modelo de Dominio	22
2.3.1 Descripción del Modelo de Dominio	22
2.4 Especificación de requerimientos	24
2.4.1 Requisitos funcionales	25
2.4.2 Requisitos no funcionales	26
2.5 Modelo de Casos de Uso del Sistema	27
2.5.1 Actores del sistema	27
2.5.2 Diagrama de Casos de Uso del Sistema	28
2.5.3 Descripción de Casos de Uso del Sistema	28
2.6 Conclusiones	45
CAPÍTULO 3: Análisis y Diseño del sistema	46
3.1 Introducción	46
3.2 Diagramas de Clases del Análisis	46
3.3 Diagrama de Clases del Diseño	51
3.4 Diseño de la base de datos	60
3.5 Conclusiones	60

CAPÍTULO 4: Implementación del sistema	61
4.1 Introducción	61
4.2 Modelo de Implementación	61
4.2.1 Diagrama de Despliegue	61
4.2.2 Diagramas de Componentes.....	62
4.3 Conclusiones	66
CONCLUSIONES GENERALES	67
RECOMENDACIONES	68
REFERENCIAS BIBLIOGRÁFICAS	69
BIBLIOGRAFÍA	71
GLOSARIO	74
ANEXOS	76

Figuras

Figura 1. Modelo de Dominio.....	24
Figura 2. Diagrama de Casos de Uso del Sistema	28
Figura 3. DCA_CU Ver tipo de reporte	47
Figura 4. DCA_CU Controlar navegación	47
Figura 5. DCA_CU Configurar tipo de análisis	47
Figura 6. DCA_CU Realizar análisis del historial	48
Figura 7. DCA_CU Realizar análisis integral.....	48
Figura 8. DCA_CU Realizar análisis de los contenidos	49
Figura 9. DCA_CU Registrar traza	49
Figura 10. DCA_CU Realizar análisis de trazas	50
Figura 11. DCA_CU Consultar historial del estudiante	50
Figura 12. DCD_CU Ver tipo de reporte.....	51
Figura 13. DCD_CU Controlar navegación	52
Figura 14. DCD_CU Configurar tipo de análisis.....	53
Figura 15. DCD_CU Realizar análisis del historial.....	54
Figura 16. DCD_CU Realizar análisis integral	55
Figura 17. DCD_CU Registrar traza.....	56
Figura 18. DCD_CU Realizar análisis de trazas	57
Figura 19. DCD_CU Consultar historial del estudiante	58
Figura 20. DCD_CU Realizar análisis de los contenidos.....	59
Figura 21. Modelo Entidad Relación.....	60
Figura 22. Diagrama de Despliegue 1	62
Figura 23. Diagrama de Despliegue 2	62
Figura 24. Diagrama de Componentes General	63
Figura 25. Diagrama de Componentes del paquete CSS.....	63
Figura 26. Diagrama de Componentes del paquete Controlador	64
Figura 27. Diagrama de Componentes del subsistema Vista.....	65
Figura 28. Diagrama de Componentes del subsistema Modelo.....	65
Figura 29. Diagrama de Componentes del paquete Lib	66

INTRODUCCIÓN

Actualmente las Tecnologías de la Información y las Comunicaciones (TIC) están sufriendo un desarrollo vertiginoso, afectando prácticamente todos los campos de la sociedad, la educación no es una excepción. Ellas se presentan “cada vez más”, como una necesidad en el contexto de la sociedad donde los rápidos cambios, el aumento de los conocimientos y las demandas de una educación de alto nivel constantemente actualizada, se convierten en una exigencia permanente.

La Universidad de las Ciencias Informáticas (UCI) es un centro que desarrolla la actividad productiva como complemento de la docencia, permitiéndole realizar convenios con empresas cubanas y contratos con empresas extranjeras, convirtiéndose en parte importante de la industria cubana del software, cuya prioridad actual es satisfacer las exigencias cada vez mayores de los clientes, lo que obliga a elevar la calidad de las producciones.

Como parte del fortalecimiento de las relaciones entre los gobiernos de Cuba y Venezuela y siguiendo las políticas establecidas a partir de diciembre del 2004, donde fue promulgado el Decreto 3390, el cual establece el uso prioritario del Software Libre en los centros de la administración pública venezolana, se realizó el contrato de un nuevo proyecto. La Colección El Navegante, actualmente instalada en las escuelas de nivel medio de Cuba, será la base del nuevo proyecto: Colecciones de software educativo Multisaber y El Navegante para planteles de los niveles de primaria y secundaria del sistema educativo, como resultado de la firma de este convenio en la X Mixta Cuba-Venezuela, con el objetivo de instalarla en todos los Liceos Bolivarianos.

La Colección El Navegante, está conformada por diez productos, cada uno se especializa en la enseñanza de una materia determinada y a su vez están compuestos por seis módulos básicos, estos brindan la posibilidad de realizar un conjunto de acciones diferentes, como es el caso del módulo Temas, el cual presenta los contenidos de las asignaturas o temas correspondientes al software. El módulo Juegos es el encargado de que los estudiantes puedan divertirse mientras emplean sus habilidades y conocimientos. El módulo Mediateca gestiona todo lo referente a los recursos multimediales, mediante el cual el estudiante puede interactuar con textos, imágenes y videos, medios de vital importancia para su aprendizaje. El módulo Ejercicios encargado de controlar el aprendizaje a través de baterías de preguntas o cuestionarios que emiten un criterio a cada respuesta. Esta Colección

cuenta además con el módulo Maestro, con la finalidad de brindar la posibilidad de configurar varias opciones relativas al producto en cuestión y por último, pero no menos importante, el módulo Resultados, encargado de supervisar todo lo referente a los reportes de los resultados obtenidos por los estudiantes durante su interacción con el software.

Dicha Colección presenta entre sus características su desarrollo con herramientas propietarias; no es multiplataforma, limitando su utilización al sistema operativo Windows, su contenido se encuentra contextualizado a Cuba, lo que hace imposible el uso en entornos educativos de otros países y el módulo Resultados a pesar de permitir la interacción del usuario con los reportes de los resultados, no provee todas las funcionalidades necesarias para la adecuada supervisión de los mismos en la aplicación. Dadas estas características presentadas por el producto es necesario desarrollar una nueva versión del mismo para poderlo comercializar con Venezuela.

A partir de lo planteado y la necesidad de migrar cada uno de los productos que conforman la Colección El Navegante a Software Libre, se define como **problema a resolver**: ¿Cómo llevar un control sobre la interacción de cada uno de los estudiantes, con los diferentes módulos de los productos que conforman la Colección El Navegante en su versión multiplataforma?.

El **objetivo general** de esta investigación es: analizar, diseñar e implementar el módulo Resultados de los productos que conforman la Colección El Navegante en su versión multiplataforma.

El problema a resolver identificado anteriormente se enmarca en el **objeto de estudio**: proceso de producción de software educativo teniendo como **campo de acción**: el desarrollo del módulo Resultados de la Colección El Navegante.

Para dar solución al problema se plantean los siguientes **objetivos específicos**:

1. Realizar el estudio de las principales tendencias y tecnologías actuales posibles a utilizar, para darle cumplimiento al problema planteado.
2. Realizar el análisis y diseño del módulo Resultados perteneciente a los nuevos productos que conforman la Colección El Navegante en su versión multiplataforma.
3. Implementar el módulo Resultados perteneciente a los nuevos productos que conforman la Colección El Navegante en su versión multiplataforma.

Se plantean como **resultados esperados**: Si se desarrolla el módulo Resultados perteneciente a los nuevos productos que conforman la Colección El Navegante en su versión multiplataforma, se podrá contar con un medio que facilite llevar el registro de todo lo referente a los reportes de resultados obtenidos por estudiantes durante su interacción con el software.

Métodos Científicos

Teóricos

Análisis histórico lógico: Con la ayuda de este método se realizará el estudio de bibliografías, posibilitando tener conocimiento de otros trabajos relacionados al tema que servirán de referencias y permitirán el desarrollo del módulo Resultados.

Análítico Sintético: Mediante la utilización de este método se examinarán los distintos documentos que están vinculados con el registro de evaluación. De esta forma, se podrá realizar un análisis de las metodologías empleadas en la versión actual de El Navegante y utilizar un procedimiento adecuado para el desarrollo del módulo en cuestión.

Con vista a darle cumplimiento a los objetivos trazados, se ha decidido desarrollar las siguientes **tareas**:

1. Investigar otras soluciones existentes.
2. Estudiar las diferentes tecnologías que serán usadas en el proyecto para implementar la Colección.
3. Estudiar la metodología y las herramientas seleccionadas por el proyecto para el desarrollo de la aplicación.
4. Realizar el levantamiento de requisitos.
5. Realizar el análisis de la solución propuesta.
6. Realizar el diseño de la solución propuesta.
7. Implementar la capa de presentación del módulo.
8. Integrar el módulo a la Colección el Navegante.

Este trabajo está estructurado en 4 capítulos, en los cuales se tratan las cuestiones siguientes:

Capítulo 1. Fundamentación Teórica: Este capítulo aborda los aspectos teóricos más importantes que servirán de base para el desarrollo del módulo, tales como estudio

de las herramientas, lenguaje y metodología definidas para desarrollar aplicaciones con características similares al módulo que se desea implementar.

Capítulo 2. Propuesta de la solución: En este capítulo se definen los requisitos funcionales y no funcionales, también cuenta con una descripción de los actores del sistema y el Diagrama de Casos de Uso del Sistema.

Capítulo 3. Análisis y diseño del sistema: Contiene una descripción de cómo deberá ser realizado el módulo partiendo de lo descrito en el capítulo anterior. Esto se realizará con la ayuda de la presentación de los Diagramas de Clases del Análisis y Diagramas de Clases del Diseño.

Capítulo 4. Implementación del sistema: Con la ayuda de los Diagramas de Componentes y Despliegue que están presentes en este capítulo se puede obtener una descripción de la implementación del sistema.

CAPÍTULO 1: Fundamentación teórica

1.1 Introducción

El presente capítulo aborda la fundamentación teórica, la cual incluye el estado del arte de la investigación a nivel internacional y nacional, para desarrollar el módulo Resultados de los productos que conforman la Colección El Navegante en su versión multiplataforma, y se detallan las diferentes tecnologías a utilizar identificadas por el equipo de proyecto que tiene la misión de desarrollar la Colección.

1.2 Análisis de otras soluciones existentes

Actualmente existen en el mundo varias aplicaciones que tienen integrado un módulo que permite registrar la evaluación de los contenidos presentes en software educativos, entre ellos se encuentran:

Moodle

Moodle (Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular), es un Sistema de Gestión de Cursos (CMS), un paquete de software diseñado para ayudar al profesor a crear fácilmente cursos en línea de calidad. Estos sistemas e-learning también se llaman Sistemas de Gestión de Aprendizaje (LMS) o Ambientes Virtuales de Aprendizaje (VLE). (1)

Es una plataforma de software libre muy fácil de manejar y con gran potencialidad ya que continuamente se están realizando actualizaciones. Moodle es un paquete de software para la creación de cursos y sitios web basados en Internet. Presenta los resultados como una secuencia de acciones del usuario con la plataforma y en el caso de los cuestionarios se muestra la puntuación obtenida por un estudiante y las respuestas dadas a cada una de las preguntas que lo componen.

Colección Multisaber

La Colección Multisaber es una multimedia educativa destinada a la enseñanza primaria. Está compuesta por catorce productos, incluye en su modelo pedagógico seis módulos multimedia. Estos son: Temas, Ejercicios, Mediateca, Profesor, Resultados y Juegos.

El módulo Resultados presenta una información consolidada de la interacción del estudiante, en una sesión de trabajo del software, con las siguientes pantallas:

Pantalla de selección del estudiante: En esta pantalla se selecciona el estudiante a analizar, permitiendo el filtrado por escuela, grado y grupo o sección. Se muestra la lista de estudiantes con la fecha de la última sesión abierta por cada uno de ellos en el formato (dd/mm/aaaa), en caso de que los estudiantes no tengan sesión en el software se muestra en el campo fecha el valor (--/--/----). En la parte inferior se muestran los botones: Registro e Historial.

Pantalla de Registro: Solo será visible si el usuario es el profesor y permite agregar, editar o eliminar los estudiantes de la base de datos.

Pantalla de Historial: Permite ir a la pantalla de historial la cual muestra el historial de cada estudiante, seleccionando la sesión de trabajo que será analizada.

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

El Navegante es una colección de software con tecnología multimedia del sistema educacional en Cuba dirigida a la enseñanza media. Está constituido por diez productos los cuales contienen seis módulos cada uno. Estos son: Temas, Ejercicios, Juegos, Biblioteca, Profesor y Resultados.

El módulo Resultados permite analizar cada una de las sesiones de trabajo de los estudiantes en el software, las cuales se presentan en forma de una bitácora de todas las acciones realizadas por el estudiante en su interacción con el software de una forma secuencial en un texto, lo que trae como desventaja que no acepta ningún tipo de filtrado que permita consolidar la información ni realizar análisis estadísticos.

Colección Futuro

Esta Colección está destinada a la educación media superior. Constituida por dieciséis productos, en su modelo pedagógico se incluyen los seis módulos similares a la Colección El Navegante. Los productos de esta Colección aportan un tratamiento más especializado al módulo Resultados y se caracterizan por tener cuatro tipos de reporte de resultados:

Traza del Estudiante: En este tipo de reporte se presenta una información de forma consolidada de la interacción del estudiante con el software en una sesión de trabajo donde se muestra el tiempo y la cantidad de veces que fue visitado cada módulo del mismo, además se especifica dentro de los contenidos del módulo cuántas veces el

estudiante consultó algún artículo o recurso y el tiempo de consulta. Para el caso de los cuestionarios, se muestran por cada contenido específico la cantidad de ejercicios realizados y cuántos fueron evaluados de Bien, Regular o Mal y el porcentaje de la eficiencia.

Análisis de contenidos específicos: Permite realizar un análisis de los resultados de la interacción de los estudiantes con los cuestionarios interactivos del módulo Ejercicios en función de los contenidos específicos en que están agrupados.

Historial del estudiante: Presenta un análisis diacrónico de los resultados de la interacción de los estudiantes con los contenidos específicos al tomar en cuenta sus resultados en función de las fechas en que los estudiantes interactuaron con los ejercicios.

Análisis Integral: Este es exactamente igual al análisis de los contenidos específicos, solo que en vez de analizar los contenidos específicos del software en cuestión, se analizará a nivel general los resultados de la interacción con los ejercicios de todo el software en función de las asignaturas a las que corresponde cada uno.

Para la realización del módulo Resultados de la Colección El Navegante, en su versión multiplataforma, se tomaron elementos del modelo pedagógico de la traza de la Colección Futuro que se encuentra instalada en las escuelas de la enseñanza media superior de Cuba y de la Colección Multisaber, esta última en su versión multiplataforma. Se aprovechan las potencialidades de estos dos modelos, para el desarrollo del módulo, se toman los tres tipos de análisis de los resultados (Análisis de los contenidos específicos, Análisis integral, Historial del estudiante) y de Multisaber específicamente el modelo de traza del estudiante. Esto se hace con el objetivo de que no tengan muchas diferencias al tener en cuenta que se van a utilizar las dos colecciones en el mismo contexto educativo.

1.3 Tecnologías y herramientas a utilizar

1.3.1 Metodología de desarrollo del software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. (6)
En la actualidad se cuenta con diversas metodologías de software las cuales se clasifican en dos grupos: las tradicionales o robustas y las ágiles. Dentro del grupo de las metodologías tradicionales se encuentran entre otras: Proceso Unificado de

Rational (RUP), Microsoft Solutions Framework (MSF) y Métrica, mientras que por las ágiles se encuentran: Extreme Programming (XP), SCRUM, Cristal Methods, entre otras. Entre estos dos grupos existe una gran diferencia y es que mientras las metodologías tradicionales intentan conseguir el objetivo común mediante el orden y la documentación, 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. (7)

Diferencias entre metodologías ágiles y tradicionales. (8)

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Luego de analizar todas estas diferencias antes expuestas se ha llegado a la conclusión de utilizar una metodología tradicional y en este caso sería la nombrada: Proceso Unificado de Rational, dado que para el uso de una metodología ágil es necesario entre muchos otros aspectos una constante interacción con el cliente de forma directa y este requerimiento no se cumple ya que la interacción con el mismo será a través de terceros, también se puede decir que RUP es un estándar para el desarrollo de software de alta calidad, ofrece una mejor comprensión del sistema

gracias a la gran cantidad de artefactos que se documentan en el mismo, mientras que las ágiles no se documentan con la misma profundidad, por tal motivo el cliente no podrá entender el desarrollo del software con la misma claridad.

Proceso Unificado de Desarrollo de Software

El Proceso Unificado de Desarrollo del Software (RUP) captura las mejores prácticas del conocimiento de líderes en Ingeniería de Software y proporciona a los equipos de desarrollo guías, estándares y recomendaciones para la construcción de software de alta calidad. Las mejores prácticas de desarrollo de software están documentadas como principio clave, donde cada una de ellas corresponde a distintos aspectos del desarrollo de software que generalmente requieren habilidades específicas, esto se refleja en los roles y productos de trabajo de cada uno de ellos. (9)

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML y trabajo de muchas metodologías utilizadas por los clientes. En un proyecto de desarrollo de software define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

RUP es un proceso y en su modelación define como sus principales elementos: (10)

- ✚ Trabajadores ("quién"): Define el comportamiento y responsabilidades de un individuo, grupo de individuos, sistemas automatizados o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- ✚ Actividades ("cómo"): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- ✚ Artefactos ("qué"): Producto tangible del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- ✚ Flujo de actividades o de trabajo ("cuándo"): Secuencia de actividades realizadas por trabajadores y que producen un resultado de valor observable.

El ciclo de vida del RUP se caracteriza por tres aspectos fundamentales

- ✚ **Dirigido por caso de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual es aceptado cuando se modela el negocio y se representa a través de los requerimientos. De aquí en adelante los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen,

como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

- ✚ **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto debe estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlos, desarrollarlos y producirlos económicamente.
- ✚ **Iterativo e incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. (23)

1.3.2 Lenguaje de modelado

UML (Lenguaje Unificado de Modelado) es un lenguaje usado para especificar, visualizar y documentar los diferentes aspectos relativos a un sistema de software bajo desarrollo, así como para modelado de negocios y almacenamiento de datos. (11)

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. (12)

Es un lenguaje que permite modelar, construir y documentar los elementos que forman un software educativo orientado a objetos. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Tiene partes estáticas, dinámicas, de entorno y organizativas. Divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo. Permite describir un sistema en diferentes niveles de abstracción, simplificando la complejidad sin perder información, para que los usuarios y desarrolladores comprendan las características de la aplicación.

1.3.3 Herramienta CASE

Las herramientas CASE (Computer Aided Software Engineering o Ingeniería de Software Asistida por Computación) son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completa o en alguna de sus fases. Pueden mejorar la productividad

en el desarrollo de una aplicación de base de datos. Permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraerse del código fuente, a un nivel donde la arquitectura y el diseño se tornan más fáciles de entender y modificar. (13)

Estas herramientas son un conjunto de software desarrollados con el objetivo de aumentar la productividad en el desarrollo del software, se logra así una reducción en el costo de los mismos en términos de tiempo y dinero. Existen varias herramientas CASE para el modelado visual mediante UML de sistemas software, entre estas se encuentran Visual Paradigm, Enterprise Architect y Rational Rose.

Visual Paradigm

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Es una herramienta multiplataforma que ayuda a una rápida construcción de aplicaciones de mejor calidad y a un menor costo. (14)

Visual Paradigm es una de las herramientas más utilizadas a nivel mundial, es multiplataforma, ayuda a construir productos de mejor calidad, de forma rápida y con poco costo. Es fácil de instalar y actualizar. Cuenta con una interfaz amigable, está integrada con distintos (IDE) y es gratuita y comercial.

1.3.4 Lenguajes de desarrollo

Un lenguaje de programación es una serie de comandos que permiten codificar instrucciones de manera que sean entendidas y ejecutadas por una computadora. (15)

Las máquinas en general, y las computadoras en particular, necesitan de un lenguaje propio para poder interpretar las instrucciones que se les dan y para que el desarrollador pueda controlar su comportamiento. Ese lenguaje que permite esta relación con las computadoras es el lenguaje de programación.

Estos lenguajes se dividen en dos grandes grupos: lenguajes del lado del servidor y lenguajes del lado del cliente. Los lenguajes del lado del servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes del lado del

cliente son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento. (16)

1.3.4.1 Lenguaje de desarrollo del lado del cliente

HTML

HTML (Lenguaje de Marcas de Hipertexto) es un lenguaje que se utiliza comúnmente para establecer la estructura y contenido de un sitio web, tanto de texto, objetos e imágenes. Los archivos desarrollados en HTML usan la extensión .htm o .html. (17)

Es un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web. Los archivos son pequeños y admitidos por todos los navegadores. La funcionalidad del HTML es tan sencilla que puede ser creado y editado en cualquier editor de textos básicos, como el Bloc de Notas del sistema operativo Windows. Los programas Microsoft FrontPage o el Adobe Dreamweaver son típicos para la edición en HTML, que sirven para crear sitios y páginas web. La mayoría de estos software incluyen una versión WYSIWYG ("Lo que ves es lo que tienes"), que permite editar en vivo el HTML de manera rápida y sencilla, lo que admite previsualizar los resultados mientras que el programa genera paralelamente la versión en código HTML.

CSS

Las CSS (**Hojas de Estilo en Cascada**) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). (18)

La principal ventaja de CSS sobre el lenguaje HTML o similar, es que el estilo se puede guardar completamente por separado del contenido, siendo posible almacenar todos los estilos de presentación para una web de muchas páginas en un sólo archivo de CSS. Permite la separación de los contenidos de los documentos descritos en HTML, XML entre otros, de la presentación de documentos con las hojas de estilo, incluye elementos tales como color, imágenes, tipos de letras, modificando la apariencia de una página web de una forma más sencilla.

JavaScript

JavaScript es un lenguaje de script multiplataforma orientado a objetos. Es pequeño y ligero; no es útil como un lenguaje independiente, sino que está diseñado para una

fácil incrustación en otros productos y aplicaciones, tales como los navegadores web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos. (19)

Tiene como una de sus principales características que es un lenguaje interpretado por lo que no requiere compilación, donde el navegador que utilice el usuario es el encargado de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas de forma adecuada. También es un lenguaje orientado a eventos ya que cuando el usuario realiza una acción en la página, como dar clic sobre un enlace se produce un evento y mediante JavaScript se desarrollan script que ejecuten acciones en respuestas a estos eventos. Por último, pero no menos importante, mencionar que es un lenguaje orientado a objetos donde su modelo de objetos incluye elementos necesarios para que los scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.

1.3.4.2 Lenguaje de desarrollo del lado del servidor

Los lenguajes del lado del servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor en el momento que se realiza la petición y que se envían al cliente en un formato comprensible para él. Como resultado de la ejecución de la página del lado del servidor, el cliente recibe solamente una página con el código XHTML. Estas páginas ejecutadas en el lado del servidor pueden efectuar consultas a base de datos, conexiones en red, así como otras tareas con el fin de crear la página final que verá el cliente. Son muchos los lenguajes que son utilizados en este lado, entre los que se encuentran: PHP, Ruby, Python, Perl, entre otros.

PHP

Es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web sin costo alguno. (20)

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, ejecutadas en el servidor, de él existe una gran documentación y es muy fácil de aprender y utilizar. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl, con algunas características específicas lo

cual propicia agilidad en el trabajo ya que en este caso la mayoría de los desarrolladores tienen noción del mismo por lo que no será necesaria una capacitación. Se caracteriza por ser un lenguaje rápido. Posee capacidad de conexión con la mayoría de los manejadores de datos y es un lenguaje multiplataforma. No necesita ser compilado para ejecutarse.

Algunas de las ventajas del uso del lenguaje PHP (20)

1. Es un lenguaje multiplataforma.
2. Completamente orientado a la web.
3. Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
4. Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
5. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no está forzado a pagar actualizaciones anuales para tener una versión que funcione.
6. Permite las técnicas de Programación Orientada a Objetos.
7. Biblioteca nativa de funciones sumamente amplia e incluida.
8. No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
9. Tiene manejo de excepciones (desde PHP5).
10. Rapidez. PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
11. Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros.

1.3.5 Framework

Un framework es una aplicación semicompleta que contiene componentes estáticos y dinámicos que pueden ser personalizados para obtener aplicaciones de usuario específicas. (21)

Los framework ayudan a desarrollar aplicaciones con mayor rapidez, pues poseen una estructura definida y una organización para el desarrollo y mantenimiento del

software desarrollado. Permite la reutilización del código y facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

1.3.5.1 Framework del lado del cliente

En la actualidad han surgido varias alternativas para el uso de aplicaciones AJAX para lograr la creación de aplicaciones web dinámicas del lado del cliente, ya que incorpora varios efectos visuales para obtener un mejor uso por parte del cliente. Estas alternativas han sido resultados del notable crecimiento de desarrollo basado en tecnología web a nivel mundial. El empleo de framework facilita el desarrollo de la aplicación a la hora de desarrollar varios de estos efectos desde cero, ya resulta un poco complejo sobre todo si hay poco tiempo para presentar la solución.

En estos momentos existen varias librerías que cumplen con los requerimientos de usabilidad que el usuario necesita para lograr crear aplicaciones web dinámicas del lado del cliente, incorporando varios efectos visuales entre los que se encuentran JQuery, Dojo y ExtJs, entre otros.

JQuery es una librería rápida y concisa de JavaScript que simplifica el documento HTML, manejo de eventos, animación y las interacciones AJAX para el desarrollo de una aplicación web en poco tiempo. JQuery está diseñada para cambiar la forma en que se escribe JavaScript. (22)

Las aplicaciones JQuery pueden responder de manera mucho más rápida que las aplicaciones JavaScript y esto se puede ver por ejemplo con respecto a sus funciones, ya que precisamente la función JQuery tiene entre sus ventajas que la página se puede manipular en cuanto se carga su código, a diferencia de la función JavaScript la cual tiene que esperar a que se carguen todos los elementos de la página. Esta librería es muy utilizada dada su sencillez, flexibilidad así como rendimiento. Se caracteriza por ser un producto estable y bien documentado logrando de esta manera una notable aceptación por parte de los programadores.

La librería JQuery en resumen aporta las siguientes ventajas

1. Ahorra muchas líneas de código.
2. Hace transparente el soporte de la aplicación para los principales navegadores web.
3. Provee de un mecanismo para la captura de eventos.

4. Provee un conjunto de funciones para animar el contenido de la página en forma muy sencilla.
5. Integra funcionalidades para trabajar con AJAX.
6. Permite tener acceso a los selectores CSS de los elementos contenidos dentro del árbol DOM.
7. Permite manipular los atributos de los elementos de la página HTML. (23)

1.3.5.2 Framework del lado del servidor

Por otra parte, del lado del servidor han surgido numerosos framework paralelamente al crecimiento existente de los framework del lado del cliente, entre los cuales se pueden mencionar: Symfony, CakePHP y CodeIgniter.

Anteriormente se especificó como lenguaje de desarrollo del lado del servidor a PHP, por las características que este presentaba. Esta decisión trae como resultado el uso de un framework implementado sobre el lenguaje PHP, para lograr mayor agilidad en el proceso de desarrollo. La selección de Symfony como framework de apoyo a la implementación en el lado del servidor fue la opción por la que se optó en este caso.

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. (24)

Características de Symfony

1. Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
2. Independiente del Sistema Gestor de Bases de Datos.
3. Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
4. Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solamente debe configurar aquello que no es convencional.
5. Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
6. 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.

7. Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Este framework ayuda a mejorar la calidad de las aplicaciones web aplicando las buenas prácticas de diseño que se han definido para la web. Cuenta con una amplia distribución de libros de forma gratuita y decenas de tutoriales. Se encarga de todos los aspectos comunes de las aplicaciones web, permite de esta forma que el desarrollador aporte las características únicas de cada proyecto.

1.3.6 Gestores de bases de datos

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (Data Base Management System) no es más que una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición, manipulación y de consulta de datos. Un SGBD permite definir los datos a distintos niveles de abstracción y manipularlos, garantizando la seguridad e integridad de los mismos. (25)

Un SGBD tiene entre sus características fundamentales:

- ✚ **Abstracción de la información:** Los detalles de almacenamiento son transparentes al usuario.
- ✚ **Redundancia mínima:** Evitar la aparición de información repetida o redundante.
- ✚ **Consistencia:** Es necesario tener en cuenta que aquella información que aparece repetida se actualice de forma coherente una vez logrado obtener una 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 información se encuentre segura frente a usuarios con otras intenciones.
- ✚ **Integridad:** Proteger los datos introducidos por usuarios descuidados, o cualquier otra ocurrencia capaz de dañar la información almacenada ante fallos de hardware.
- ✚ **Respaldo y recuperación:** Deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada, y de restaurar a partir de estas copias los datos que se hayan podido perder.

- ✚ **Control de la concurrencia:** Controlar los accesos concurrentes a la información para evitar las inconsistencias. (4)

Entre los diferentes tipos de SGBD se pueden encontrar el PostgreSQL, Oracle, MySQL y Microsoft SQL Server. La mayoría de estos gestores son propietarios, su costo en el mercado es muy elevado en la mayoría de los casos, una alternativa sería el uso de gestores libres, entre los que se encuentra PostgreSQL y MySQL, los cuales han logrado alcanzar un nivel de madurez equivalente a los gestores propietarios en cuanto a la eficiencia y seguridad en los últimos años. Sin embargo, el SGBD PostgreSQL es uno de los más usados en el mundo de software libre por ser rápido y poseer facilidad de uso además de su fácil instalación y configuración.

PostgreSQL es un sistema diseñado para administrar grandes cantidades de datos. Es compatible con el almacenamiento de objetos binarios, incluye imágenes, sonidos o vídeos. Tiene interfaces de programación nativa de C / C ++, Java, .NET, Perl, Python, PHP, Ruby, entre otros, y posee una amplia documentación. Incluye la mayoría de tipos de datos, como INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL y TIMESTAMP. El PostgreSQL se ejecuta en la mayoría de los Sistemas Operativos más utilizados en el mundo incluyendo, Linux, varias versiones de UNIX y Windows. (26)

1.3.7 Tipo de aplicación

Una aplicación web es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o su red interna (intranet). Las aplicaciones web tienen varias ventajas sobre los programas de software descargables tradicionales, entre las que se encuentran:

- ✚ **Compatibilidad multiplataforma:** Las aplicaciones web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables.
- ✚ **Actualización:** Están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación.
- ✚ **Inmediatez de acceso:** Las aplicaciones basadas en web no necesitan ser descargadas, instaladas y configuradas.

- ✚ **Menos requerimientos de memoria:** Las aplicaciones basadas en web al residir y correr en los servidores del proveedor usa en muchos casos la memoria de las computadoras que ellos corren, dejan más espacio para correr múltiples aplicaciones del mismo tiempo sin incurrir en frustrantes deterioros en el rendimiento.
- ✚ **Precio:** No requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional.
- ✚ **Múltiples usuarios concurrentes:** Puede realmente ser utilizada por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta. (27)

Otras ventajas

1. Se facilita el trabajo a distancia.
2. No se requieren complicadas combinaciones de Hardware/Software para utilizar estas aplicaciones. Solo un computador con un buen navegador web.
3. Las aplicaciones web son fáciles de usar (no requieren conocimientos avanzados de computación).
4. Con una aplicación web se podrá tener alta disponibilidad, ya que se puede realizar consultas desde cualquier parte del mundo donde se tenga acceso a Internet.

1.3.8 Servidor web

El servidor web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor web buscará una página web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. (28)

Entre los servidores web más utilizados se encuentran Apache, Microsoft IIS y Google GFE.

Apache

Está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a

través de su diseño modular. Este diseño permite a los administradores de sitios web elegir qué características van a ser incluidas en el servidor, seleccionando qué módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. (29)

Este servidor es uno de los más seguros y puede soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos también puede personalizarse según las necesidades de cada sitio web. Entre sus características se puede mencionar que es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto y trabaja con páginas Java y páginas JavaScript, teniendo todo el soporte que necesita para tener páginas dinámicas.

1.3.9 Entorno de desarrollo integrado

Un (IDE) Entorno de Desarrollo Integrado (Integrated Development Environment) es un programa compuesto por un conjunto de herramientas para el uso del programador que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por si solos o pueden ser parte de aplicaciones existentes. (30)

Entre los IDEs se encuentran: Aptana Studio, Zend Studio y NetBeans, entre otros.

NetBeans IDE

Es un premiado IDE disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores la creación rápida de diferentes tipos de aplicaciones utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy y Grails, y C / C + +. (31)

Tiene soporte para varios lenguajes como PHP, JavaScript, HTML, CSS. Esta herramienta ofrece una amplia documentación y recursos. Es un producto de código abierto, libre y gratuito sin restricciones de uso, en su versión 6.8 tiene soporte para el framework Symfony. En su integración con este framework permite desarrollar aplicaciones de forma más sencilla y productiva. Proporciona a los programadores mayor flexibilidad a la hora de modificar el IDE con cada nueva versión del NetBeans. Por estas y otras ventajas, fue seleccionado por el equipo de proyecto para el desarrollo de la nueva Colección.

1.3.10 Patrón Modelo-Vista-Controlador

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Tiene como objetivo intentar repetirse lo menos posible y tenerlo todo organizado, es decir, realizar una distinción entre la lógica y la presentación de la aplicación. Los desarrolladores se enfrentan con frecuencia a problemas comunes como la reutilización del código que ya han implementado. Esta actividad se facilita mucho si a la hora de programar los desarrolladores tienen la precaución de separar el código en varias partes de forma tal que puedan ser reutilizadas sin modificaciones, ya que hay momentos en que tienen que resolver un problema parecido a algo que ya han implementado como por ejemplo, mejorar el algoritmo o el aspecto de un determinado 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 permitirá 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 va a interactuar el usuario, más conocida como interfaz.
- ✚ **Controlador:** Es el Sistema de Gestión de Base de Datos, responde más bien a eventos, normalmente, son acciones que el usuario invoca, implica cambios en el modelo y también en la vista. (32)

1.4 Conclusiones

Durante el desarrollo de este capítulo se analizó el proceso de gestión del registro de evaluación en varios sistemas con el objetivo de obtener conocimiento de sus tendencias actuales. Se fundamentó todo lo necesario para darle una solución eficaz al problema a resolver. Se realizó un análisis de algunas de las soluciones existentes. Con respecto a las tecnologías a utilizar se seleccionó PostgreSQL como SGBD, Apache como servidor web, UML como lenguaje modelado. Con relación a los lenguajes de desarrollo, tanto del lado del cliente como del lado del servidor, se propuso el uso de CSS, HTML, JavaScript y PHP. Se propone JQuery como framework del lado del cliente y Symfony del lado del servidor. NetBeans es la propuesta como Entorno de Desarrollo Integrado y Visual Paradigm como herramienta CASE.

CAPÍTULO 2: Propuesta de la solución

2.1 Introducción

En el presente capítulo se describe y modela la solución propuesta, con la ayuda del RUP. Se realiza una descripción del Modelo de Dominio y de sus conceptos asociados. Además, se enumeran los requerimientos funcionales y no funcionales del sistema, lo cual permite hacer una concepción general de la aplicación e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema y la descripción de cada caso de uso.

2.2 Valoración del módulo a implementar

El módulo Resultados tiene como objetivo mostrar los reportes de la interacción de los estudiantes con la Colección, permitiendo obtener la información de una manera consolidada, ésta se agrupa según los módulos del software con los que interactúan los estudiantes. Cada uno de estos módulos tiene su propia estructura de reporte de información según sus características:

- ✚ **Temas:** Presenta el tiempo y la cantidad de veces que estuvo el estudiante en un contenido.
- ✚ **Ejercicios:** Se agrupan por temas y según la evaluación obtenida.
- ✚ **Juegos:** Resultados de la iteración de los juegos.
- ✚ **Mediateca:** Acceso y tiempo que estuvo en cada recurso.

2.3 Modelo de Dominio

2.3.1 Descripción del Modelo de Dominio

Debido a la poca estructuración de los procesos de negocio se propone un Modelo de Dominio, pues captura los tipos más importantes de objetos en el contexto del sistema. Los objetos representan las “cosas” que existen o los eventos que suceden en el entorno del problema. El Modelo de Dominio permitirá una mejor comprensión de los conceptos del entorno donde se desarrolla el problema planteado. Para su realización se describe el Modelo del Dominio a través de un Diagrama de Clases de UML, el cual define las principales clases conceptuales que intervienen en el desarrollo del módulo Resultados de los productos que conforman la Colección El Navegante en su versión multiplataforma.

Análisis de los conceptos del dominio

- ✚ **Liceo:** Institución educativa centrada en la formación de estudiantes, que tiene como objetivo fundamental organizar la enseñanza y el aprendizaje de contenidos básicos y capacitar al alumno para proseguir estudios superiores o bien para incorporarse al mundo laboral.
- ✚ **Profesor:** Persona responsable de educar, guiar y supervisar a los estudiantes en el proceso de enseñanza - aprendizaje.
- ✚ **Estudiante:** Persona que cursa estudios en las instituciones de nivel secundario.
- ✚ **Colección Navegante:** Constituye un conjunto de hiperentornos de enseñanza-aprendizaje que está 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.
- ✚ **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:** Es el encargado de gestionar todo lo referente a los recursos multimediales, mediante este el estudiante puede interactuar con textos, imágenes y videos, lo cual es de vital importancia para su aprendizaje.
- ✚ **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.
- ✚ **Temas:** Módulo en el que se representan los contenidos de las asignaturas o temas a los que responde el software. Esto se presentan en una forma hipertextual permitiendo la navegación no lineal por los contenidos.

- ✚ **Profesor:** Módulo encargado de proveer a los docentes de los recursos y orientaciones sobre un tema educativo y le permite un mejor aprovechamiento de los hiperentornos.

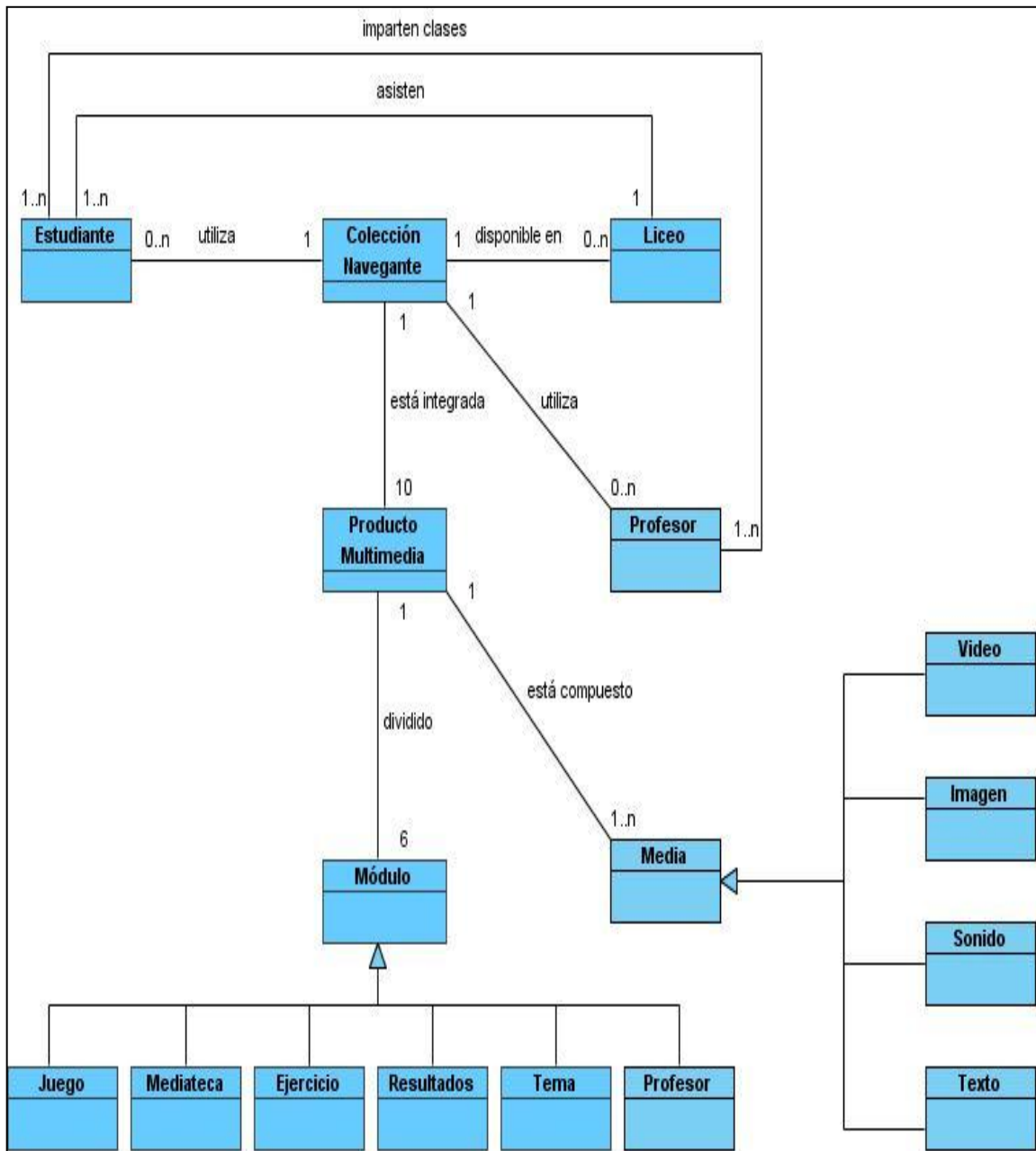


Figura 1. Modelo de Dominio

2.4 Especificación de requerimientos

2.4.1 Requisitos funcionales

RF1: Mostrar las opciones de los 4 tipos de reportes de resultados a los que tendrá acceso el usuario (Traza de estudiante, Análisis de contenidos específicos, Historial del estudiante, Análisis integral).

RF2: Mostrar tipo de análisis: Individual, Grupal, Subgrupo, General.

RF3: Realizar análisis de los contenidos específicos.

RF3.1: Mostrar las asignaturas y permitir la selección de las que se quiera analizar.

RF3.2: Mostrar los resultados de los contenidos específicos.

RF3.3: Mostrar una gráfica de comparación de efectividad de los contenidos específicos.

RF4: Realizar análisis del historial del estudiante a partir de la fechas de interacción con los ejercicios.

RF4.1: Mostrar las fechas y permitir la selección de las que se quiera analizar.

RF4.2: Mostrar los contenidos realizados según las fechas seleccionadas anteriormente y permitir la selección de los que se desean analizar.

RF4.3: Mostrar los resultados de los contenidos específicos por fechas. Utilizar una recta numérica para la selección de las fechas.

RF4.4: Mostrar gráfica de análisis de efectividad.

RF5: Realizar análisis integral.

RF5.1: Mostrar los contenidos y permitir la selección de los que se quiera analizar.

RF5.2: Mostrar los resultados de las asignaturas seleccionadas.

RF5.3: Mostrar una gráfica de comparación de efectividad de las asignaturas.

RF6: Permitir la navegabilidad entre cada una de las pantallas a través de las opciones Siguiente y Terminar.

RF7: Buscar estudiantes por escuela, grado y sección.

RF8: Mostrar de cada estudiante el número de la lista, nombre(s), apellidos, cantidad de sesiones iniciadas y última sesión.

RF9: Registrar la interacción de los estudiantes con los siguientes módulos o secciones del producto: Temas, Ejercicios, Juegos, Mediateca.

RF10: Mostrar el historial de cada estudiante en las sesiones de trabajo iniciadas.

2.4.2 Requisitos no funcionales

Los requisitos no funcionales están vinculados a requisitos funcionales, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades son características que hacen al producto atractivo. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto.

Requisitos de Software

Navegador Internet Explorer 8 o Mozilla Firefox 3.x.

Requisitos de Hardware

Todas las computadoras deben contar con:

1. Procesador Pentium 2,93 GHz
2. 20 GB de espacio en disco duro como mínimo.

Requisitos de Hardware para la PC cliente

3. 256 MB de RAM (recomendado 512 MB de RAM o mayor).
4. Soporte de video que admita resolución de al menos 1024x600 y 24 bits.
5. Dispositivos de audio.
6. Dispositivo de red de al menos 10 Mbits.
7. El sistema interactuará con una impresora que permita imprimir los diferentes contenidos como respuesta a las funcionalidades del sistema.

Requisitos de Hardware para la PC servidor

1. 1 GB de RAM para la PC servidor.
2. Dispositivo de red de al menos 10 Mbits.

Requisitos de apariencia o interfaz externa

1. El sistema proporcionará claridad y correcta organización de la información, permite la interpretación correcta e inequívoca de ésta.
2. El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.

Requisitos de Portabilidad

La aplicación podrá ser utilizada bajo cualquier Sistema Operativo.

2.5 Modelo de Casos de Uso del Sistema

El Modelo de Casos de Uso describe la funcionalidad propuesta del nuevo sistema. Los casos de uso se utilizan para modelar el funcionamiento del sistema, son una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje y realice la captura de requisitos funcionales del sistema utilizando UML y representándolo mediante un Diagrama de Caso de Uso, garantizando de esta forma su correcto funcionamiento de acuerdo con las exigencias del cliente. Para ello se definen cuáles serían los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades del mismo.

2.5.1 Actores del sistema

Actores	Descripción
Usuario	Representa a una persona (principalmente estudiante de nivel secundario, invitado o profesor) que va interactuar con el sistema.
Estudiante	Representa a una persona (principalmente estudiante de nivel secundario) que va a interactuar con el sistema.

Tabla 1. Determinación y descripción de los actores del sistema.

2.5.2 Diagrama de Casos de Uso del Sistema

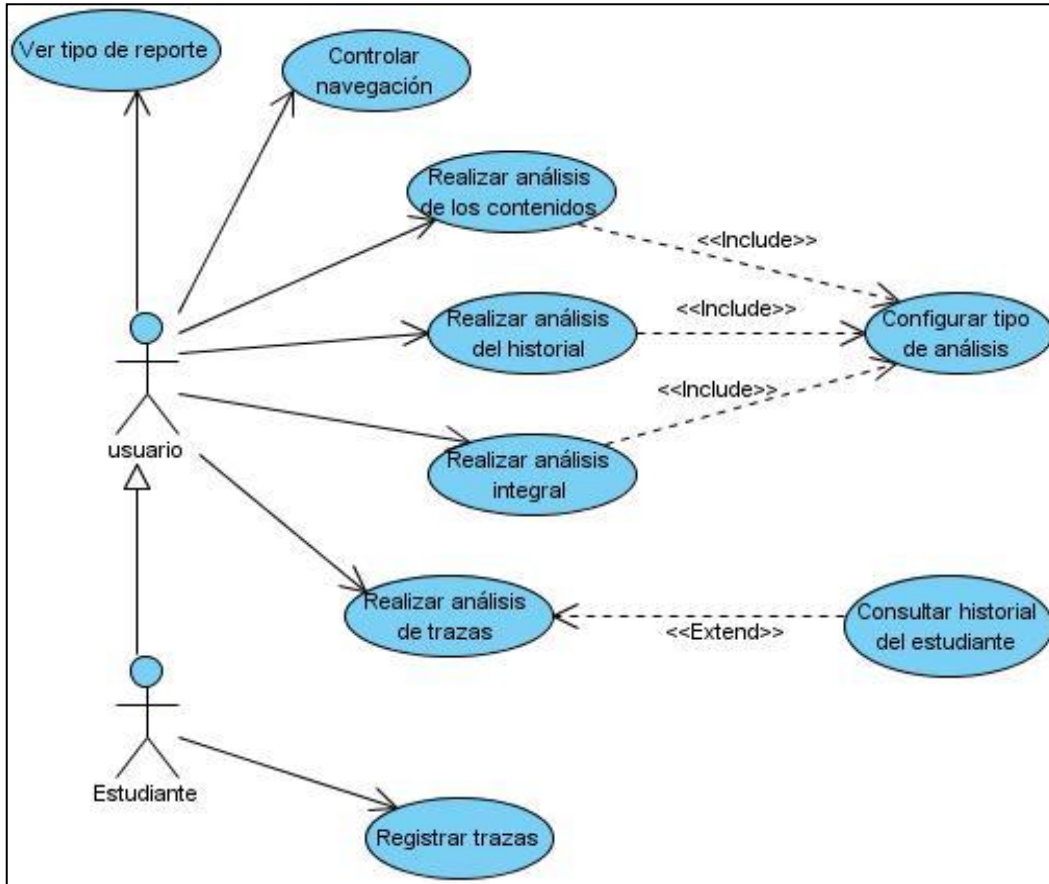


Figura 2. Diagrama de Casos de Uso del Sistema

2.5.3 Descripción de Casos de Uso del Sistema

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

Caso de Uso:	Ver tipo de reporte
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario selecciona la imagen que metafóricamente representa al módulo Resultados, el sistema muestra una interfaz permitiéndole al usuario seleccionar el reporte a consultar.
Precondiciones	El usuario debe estar registrado en el sistema.
Referencia	RF1
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1. El caso de uso inicia cuando el usuario selecciona la imagen que metafóricamente representa al módulo Resultados.	2. El sistema muestra una interfaz permitiéndole al usuario la selección del reporte a consultar: -Traza del estudiante -Análisis de los contenidos específicos -Análisis integral -Historial del estudiante
Poscondiciones	Se realiza la inicialización de los resultados.

Tabla 3. Descripción del CU-2: Configurar tipo de análisis.

Caso de Uso:	Configurar tipo de análisis	
Actores:	Usuario	
Resumen:	El caso de uso se inicia cuando el usuario interactúa con alguno de los dos reportes de análisis que ofrece la aplicación o el historial del estudiante, el sistema muestra una interfaz permitiéndole al usuario la selección del tipo de análisis a realizar, efectúa la búsqueda de los estudiantes que han realizado ejercicios mediante un filtrado por la escuela, grado y grupo y un tamaño de muestra para la selección aleatoria.	
Precondiciones	El usuario debe estar registrado en el sistema.	
Referencias	RF2	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el usuario selecciona alguno de los tres reportes de análisis que ofrece la aplicación: Análisis integral. Análisis de los contenidos específicos. Historial del estudiante.	2. El sistema muestra una interfaz permitiéndole al usuario la configuración del tipo de análisis a realizar: -Individual: ir a la Sección 1 . -Por grupo: ir a la Sección 2 . -Por subgrupo: ir a la Sección 3 . -General: ir a la Sección 4 .	
Sección 1 “Individual”: Flujo Normal de los Eventos		
Acción del Actor	Respuesta del Sistema	

	1. El sistema muestra una interfaz permitiendo el filtrado por escuela, grado y grupo o sección para la selección del estudiante donde solo aparecen los que tienen reportes de ejercicios realizados.
2. El usuario selecciona una escuela, grado y grupo o sección.	3. El sistema muestra una lista donde solo aparecen los estudiantes que tienen reportes de ejercicios realizados.
Sección 2 “Por grupo”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra una interfaz en la que se puede seleccionar el grupo.
2. El usuario selecciona un grupo o sección.	
Sección 3 “Por subgrupos”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra una interfaz que permite el filtrado de los estudiantes por grado y grupo. Presenta además dos listas, en la lista de la izquierda van a aparecer todos los estudiantes que han realizado ejercicios dada la selección especificada. La lista de la derecha estará vacía para que se puedan mover a ella los estudiantes que se desean analizar.
2. El usuario mueve a la lista de la derecha los estudiantes que se desean analizar.	3. El sistema muestra en la lista de la derecha los estudiantes seleccionados para analizar.
Sección 4 “General”: Flujo Normal de los Eventos	

	1. El sistema muestra una interfaz donde tiene habilitado una caja de texto para que el estudiante entre el tamaño de la muestra y una opción para que el usuario pueda seleccionar a todos los estudiantes sin tener que entrar el número en la caja de texto.
2. El usuario entra un número menor que la cantidad máxima de estudiantes en la caja de texto habilitada para seleccionarlo como muestra a analizar.	
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Sistema
2.1. El usuario selecciona a todos los estudiantes como muestra.	
Poscondiciones	Se realiza la selección del tipo de análisis.

Tabla 4. Descripción del CU-3: Realizar análisis de los contenidos

Caso de Uso:	Realizar análisis de los contenidos
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario selecciona la acción análisis de los contenidos específicos. El sistema muestra una interfaz para la selección del tipo de análisis a realizar, así como la selección de los contenidos a analizar, muestra una tabla con los resultados de los mismos y una gráfica con la comparación de la efectividad de los contenidos.
Precondiciones	El usuario debe estar registrado en el sistema.
Referencias	RF3, RF3.1, RF3.2 , RF3.3
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

<p>1. El caso de uso inicia cuando el usuario selecciona la opción análisis de los contenidos específicos.</p>	<p>2. El sistema muestra una interfaz permitiéndole al usuario la selección del tipo de análisis a realizar. <i>Ver CU Configurar tipo de análisis.</i></p>
	<p>3. El sistema muestra una interfaz donde se muestran dos listas, en la lista de la izquierda aparecen los contenidos en las que se han realizado ejercicios por los estudiantes. La lista de la derecha estará vacía para que se puedan mover a ella los contenidos en los que se han realizado ejercicios para analizar.</p>
<p>4. El usuario selecciona un contenido específico.</p>	
<p>5. El usuario mueve a la lista de la derecha los contenidos específicos a los que pertenecen los ejercicios realizados por los estudiantes para analizarlo.</p>	<p>6. El sistema muestra en la lista de la derecha los contenidos en los que se han realizado ejercicios.</p>
	<p>7. El sistema muestra una tabla con los resultados del análisis realizado sobre los estudiantes en los contenidos específicos seleccionados con los siguientes campos: nombre del contenido (Contenido), cantidad de ejercicios con la máxima calificación (Bien u otro término), cantidad de ejercicios evaluados de regular (Regular u otro término), cantidad de ejercicios evaluados de mal (Mal u otro término), cantidad total de ejercicios realizados (Total) y el por ciento de eficiencia en la realización</p>

	de los ejercicios (Efectividad).
	8. El sistema muestra una gráfica de barra en la que cada columna representará la efectividad de cada contenido específico. Cada columna presentará un color diferente para diferenciar en la leyenda.
Poscondiciones	Se realiza el análisis de los contenidos específicos.

Tabla 5. Descripción del CU-4: Realizar análisis del historial.

Caso de Uso:	Realizar análisis del historial	
Actores:	Usuario	
Resumen:	El caso de uso se inicia cuando el usuario selecciona la acción realizar análisis del historial. El sistema muestra una interfaz para la selección del tipo de análisis a realizar, también permite la selección de las fechas en que desea analizar los contenidos, muestra además una tabla con los resultados del análisis y una gráfica con el análisis de la efectividad.	
Precondiciones	El usuario debe estar registrado en el sistema.	
Referencias	RF4, RF4.1, RF4.2, RF4.3, RF4.4,	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el usuario selecciona la acción análisis del historial.	2. El sistema muestra una interfaz permitiéndole al usuario la selección del tipo de análisis a realizar. <i>Ver CU Configurar tipo de análisis.</i>	

	3. El sistema muestra una interfaz presentando dos listas, donde en la lista de la izquierda aparece las fechas en las que se han realizado ejercicios por los estudiantes. La lista de la derecha estará vacía para que se puedan mover a ella las fechas a analizar.
4. El usuario selecciona una fecha a analizar.	
5. El usuario mueve a la lista de la derecha las fechas seleccionadas para analizar.	6. El sistema muestra en la lista de la derecha las fechas seleccionadas para analizar los contenidos en los que se han realizado ejercicios.
	7. El sistema muestra una interfaz presentando dos listas, en la lista de la izquierda aparece los contenidos en los que se han realizado ejercicios realizados por los estudiantes en las fechas seleccionadas a analizar. La lista de la derecha estará vacía para que se puedan mover a ella los contenidos a analizar, permitiéndole al usuario seleccionar los contenidos específicos en los que se han realizado ejercicios en las fechas seleccionadas anteriormente.
8. El usuario selecciona un contenido específico.	
9. El usuario mueve a la lista de la derecha los contenidos en los que se han realizado ejercicios en las fechas seleccionadas anteriormente.	10. El sistema muestra en la lista de la derecha los contenidos seleccionados para analizar.

	11. El sistema muestra una interfaz la cual presenta una recta numérica mostrando las fechas seleccionadas.
12. El usuario mueve el indicador de selección.	13. El sistema muestra una tabla con los resultados del análisis realizado sobre los estudiantes en los contenidos específicos en esa fecha con los siguientes campos: nombre del contenido (Contenido), cantidad de ejercicios con la máxima calificación (Bien u otro término), cantidad de ejercicios evaluados de regular (Regular u otro término), cantidad de ejercicios evaluados de mal (Mal u otro término), cantidad total de ejercicios realizados (Total) y el por ciento de eficiencia en la realización de los ejercicios (Efectividad).
	14. El sistema muestra una interfaz que contiene una gráfica de línea en la cual por las X se representan las fechas seleccionadas y por las Y el valor de la efectividad. Cada contenido específico está representado por una línea de un color diferente y se traza a través de los puntos formados por el par (fecha, efectividad).
Poscondiciones	Se realiza el análisis del historial.

Tabla 6. Descripción del CU-5: Realizar el análisis integral.

Caso de Uso:	Realizar análisis integral
Actores:	Usuario

Resumen:	El caso de uso se inicia cuando el usuario selecciona la acción análisis integral. El sistema muestra una interfaz para la selección del tipo de análisis a realizar, también permite la selección de las asignaturas que desea analizar, muestra además una tabla con los resultados de este análisis y una gráfica de comparación de efectividad de las asignaturas.	
Precondiciones	El usuario debe estar registrado en el sistema.	
Referencias	RF5; RF5.1; RF5.2; RF5.3	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el usuario selecciona la acción análisis integral.	2. El sistema muestra una interfaz permitiéndole al usuario la selección del tipo de análisis a realizar. <i>Ver CU Configurar tipo de análisis.</i>	
	3. El sistema muestra una interfaz donde se muestran dos listas, en la lista de la izquierda aparecen las asignaturas en las que se han realizado ejercicios por los estudiantes. La lista de la derecha estará vacía para que se puedan mover a ella las asignaturas en las que se han realizado ejercicios para analizar.	
4. El usuario selecciona las asignaturas a analizar.		
5. El usuario mueve a la lista de la derecha las asignaturas específicas a las que pertenecen los ejercicios realizados por los estudiantes para analizarla.	6. El sistema muestra en la lista de la derecha las asignaturas en las que se han realizado ejercicios.	
	7- El sistema muestra una tabla con los resultados del análisis realizado sobre los estudiantes en las asignaturas seleccionadas con los	

	siguientes campos: nombre de la asignatura (Asignatura), cantidad de ejercicios con la máxima calificación (Bien u otro término), cantidad de ejercicios evaluados de regular (Regular u otro término), cantidad de ejercicios evaluados de mal (Mal u otro término), cantidad total de ejercicios realizados (Total) y el por ciento de eficiencia en la realización de los ejercicios (Efectividad).
	8. El sistema muestra una gráfica de barra en la que cada columna representará la efectividad de cada asignatura. Cada columna presentará un color diferente para diferenciar en la leyenda.
Poscondiciones	Se realiza el análisis integral.

Tabla 7. Descripción del CU-6: Controlar navegación.

Caso de Uso:	Controlar navegación	
Actores:	Usuario	
Resumen:	El caso de uso se inicia cuando el usuario decide cambiar de interfaz. El sistema muestra en la interfaz dos opciones permitiéndole al usuario realizar la navegación deseada (siguiente y terminar).	
Precondiciones	El usuario debe estar registrado en el sistema.	
Referencias	RF6	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	

1. El caso de uso inicia cuando el usuario desea cambiar de interfaz.	2. El sistema muestra en cada interfaz dos opciones permitiéndole al usuario seleccionar la de su preferencia: - Siguiete - Terminar
3. El usuario selecciona "Siguiete".	4. El sistema le muestra la pantalla siguiente a la que se encuentra en esos momentos permitiéndole al usuario consultar nuevos resultados.
Flujo Alternativo de Eventos	
Acción del Actor	Respuesta del Sistema
3.1. El usuario selecciona "Terminar".	4.1. El sistema le muestra la interfaz principal permitiéndole al usuario consultar otro tipo de reporte de resultado , estos son: -Traza del estudiante -Análisis de los contenidos específicos -Análisis integral -Historial del estudiante
Poscondiciones	Se controla la navegabilidad entre interfaz.

Tabla 8. Descripción del CU-7: Realizar análisis de trazas.

Caso de Uso:	Realizar análisis de trazas
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario selecciona el reporte Traza del estudiante. El sistema muestra el listado de todos los estudiantes que interactuaron con el sistema y termina el caso de uso.
Precondiciones	El usuario debe estar registrado en el sistema.
Referencias	RF7; RF8
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1. El caso de uso inicia cuando el usuario selecciona Trazas del estudiante.	2. El sistema muestra la interfaz “Resultados de los estudiantes” con las siguientes opciones de búsqueda: -Escuela -Grado -Sección
3. El usuario selecciona una o varias opciones de búsqueda.	4. El sistema busca y muestra un listado con los siguientes datos: - Número de lista - Nombre(s) - Apellido(s) - Sesiones - Fecha de la última sesión iniciada
Poscondiciones	Se realiza el análisis de trazas.

Tabla 9. Descripción del CU-8: Registrar traza.

Caso de Uso:	Registrar traza	
Actores:	Estudiante	
Resumen:	El caso de uso se inicia cuando un estudiante selecciona algún módulo de los existentes (Temas, Ejercicios, Juegos, Mediateca). El sistema registra los datos de la interacción del mismo con el módulo y termina el caso de uso.	
Precondiciones	El estudiante debe estar registrado en el sistema.	
Referencias	RF9	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el estudiante selecciona la imagen que metafóricamente representa algún módulo en específico, estos son: -Temas -Ejercicios -Juegos	2. El sistema muestra los datos específicos correspondiente al módulo seleccionado por el estudiante, para que el mismo pueda realizar la selección deseada.	

-Mediatecas	
3. El estudiante realiza una selección específica.	4. El sistema muestra los datos específicos de esta selección realizada por el estudiante, permitiéndole al mismo la interacción con el sistema.
	5. El sistema registra las trazas del estudiante durante su interacción con cada producto de la Colección para luego visualizar su comportamiento.
Poscondiciones	Se registra la traza de los módulos con los que el estudiante interactúa.

Tabla 10. Descripción del CU-9: Consultar historial del estudiante.

Caso de Uso:	Consultar historial del estudiante	
Actores:	Usuario	
Resumen:	El caso de uso se inicia cuando el usuario desea ver el historial de algún estudiante. El sistema muestra los resultados del estudiante seleccionado y termina el caso de uso.	
Precondiciones	Debe haber sido seleccionado un estudiante.	
Referencias	RF10	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el usuario selecciona la opción "Historial".	2. El sistema busca los resultados del estudiante elegido por el usuario y muestra una interfaz con la fecha de la primera sesión de trabajo iniciada por el estudiante y las siguientes categorías de consulta: Datos de la sesión	

	<ul style="list-style-type: none"> • Identificación • Itinerario • Contenidos visitados • Ejercicios • Juegos <p>Mediateca</p> <ul style="list-style-type: none"> • Imágenes • Videos • Sonidos • Glosario
	<p>3. El sistema permite seleccionar cualquiera de las categorías mencionadas anteriormente, además de poder elegir la fecha de la sesión de trabajo que desea.</p>
<p>4. El usuario selecciona la fecha de la sesión de trabajo que desea, además de la categoría relacionada con los resultados obtenidos por los estudiantes.</p>	<p>5. El sistema busca las solicitudes realizadas por el usuario y muestra un panel de visualización que contiene datos diferentes para cada categoría:</p> <ul style="list-style-type: none"> - Identificación: ir a la Sección 1. - Itinerario: ir a la Sección 2. - Contenidos visitados: ir a la Sección 3. - Ejercicios: ir a la Sección 4. - Juegos: ir a la Sección 5. - Imágenes: ir a la Sección 6. - Videos: ir a la Sección 7. - Sonidos: ir a la Sección 8. - Glosario: ir a la Sección 9.
<p>Sección 1 “Identificación”: Flujo Normal de los Eventos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>1. El sistema muestra los datos generales del estudiante seleccionado, estos son:</p>

	<ul style="list-style-type: none"> - Nombre(s) y Apellido(s) - Grado - Grupo - Escuela - Fecha y Hora de entrada al software
Sección 2 “Itinerario”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra dos tablas que contienen varios datos acerca del recorrido realizado por el estudiante, estos son:</p> <p>Resumen del Itinerario:</p> <ul style="list-style-type: none"> -Fecha, hora de entrada, hora de salida y tiempo general en el software. -Módulos visitados, cantidad de veces que visitó cada módulo y tiempo total en cada módulo.
Sección 3 “Contenidos visitados”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra dos tablas que contienen datos acerca del resultado de la interacción del estudiante con el módulo “Temas”, estos son:</p> <p>-Resumen de Temas: temas visitados, cantidad de veces que visitó cada tema y tiempo total en cada tema.</p> <p>-Detalles de Temas: temas visitados, tiempo total en cada tema, epígrafes visitados por temas y tiempo en cada epígrafe.</p>
Sección 4 “Ejercicios”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra dos tablas que contienen datos acerca del resultado</p>

	<p>de la interacción del estudiante con el módulo “Ejercicios”, estos son:</p> <p>-Resumen de Ejercicios: contenidos, cantidad total de ejercicios realizados por contenidos, cantidad de ejercicios evaluados de B, R o M por contenidos, % de efectividad por contenidos.</p> <p>-Detalles de Ejercicios: números de los ejercicios realizados, calificación, tiempo, % de efectividad e intentos consumidos.</p>
Sección 5 “Juegos”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra los siguientes datos para todos los juegos:</p> <ul style="list-style-type: none"> -Título de los juegos -Cantidad de veces que jugó -Tiempo de permanencia
Sección 6 “Imágenes”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra dos tablas que contienen varios datos sobre la interacción del estudiante con la galería de imágenes, estos son:</p> <p>-Resumen de Imágenes: cantidad de imágenes con las que se interactuó y tiempo.</p> <p>-Detalles de Imágenes: título de la imagen, cantidad de veces que la observó y tiempo</p>
Sección7 “Videos”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema

	<p>1. El sistema muestra dos tablas que contienen varios datos sobre la interacción del estudiante con la galería de videos, estos son:</p> <p>-Resumen de Videos: cantidad de videos con los que se interactuó y tiempo.</p> <p>-Detalles de Videos: título del video, cantidad de veces que lo observó y tiempo.</p>
Sección 8 “Sonidos”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra dos tablas que contienen varios datos sobre la interacción del estudiante con la información de interés que se presenta en la Mediateca, estos son:</p> <p>-Resumen de Sonidos: cantidad de sonidos con las que se interactuó y tiempo.</p> <p>-Detalles de Sonidos: título del sonido, cantidad de veces que lo escuchó y tiempo.</p>
Sección 9 “Glosario”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
	<p>1. El sistema muestra una tabla que contienen varios datos sobre la interacción del estudiante con las palabras del glosario, estos son:</p> <p>-Resumen del Glosario: cantidad de palabras consultadas, tiempo total, palabras consultadas y tiempo dedicado a cada palabra.</p>
Poscondiciones	Se realiza la consulta del historial.

2.6 Conclusiones

En este capítulo, a la poca claridad en los procesos del negocio, se realizó el Modelo de Dominio del mismo, donde se identificaron los conceptos principales que se asocian al entorno del problema. Se realizó la identificación de los requisitos funcionales y no funcionales. Se diseñó el DCUS y se realizó la descripción de cada caso de uso lo cual crea una base para el desarrollo del siguiente capítulo.

CAPÍTULO 3: Análisis y Diseño del sistema

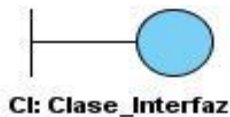
3.1 Introducción

En el presente capítulo se expone el análisis y diseño del sistema propuesto, donde el resultado del Modelo del Análisis es esencial para la concepción general del diseño del sistema ya que tiene como objetivo mantener un modelo eficiente de la solución propuesta que sirva de base para el diseño, el cual, con la ayuda de los Diagramas de Clases del Diseño se detalla la interacción de las distintas páginas y clases diseñadas. También se presenta el Modelo de Datos.

3.2 Diagramas de Clases del Análisis

El Diagrama de Clases del Análisis constituye una vista estática de las clases que conforman el Modelo del Análisis y las asociaciones entre las mismas. Es una vista de la futura composición de clases de software. Con la información recopilada, luego de identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellos, se construye el Diagrama de Clases del Análisis.

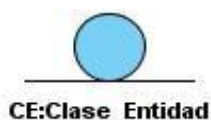
Las clases del Análisis están siempre identificadas con uno de los tres estereotipos existentes, los cuales son:



Se encargan de la modelación de toda la interacción que puede existir entre los actores y el sistema. Ejemplos: ventanas, formularios, comunicación con otros sistemas o dispositivos.



Coordinan los eventos necesarios para la realización o especificación del caso de uso, en otras palabras, son las que ejecutan el caso de uso. Usualmente son dependientes de la aplicación, además de tener un control sobre todas las acciones a realizar.



Representa la información de larga duración y a menudo persistente que se maneja en el sistema.



Figura 3. DCA_CU Ver tipo de reporte

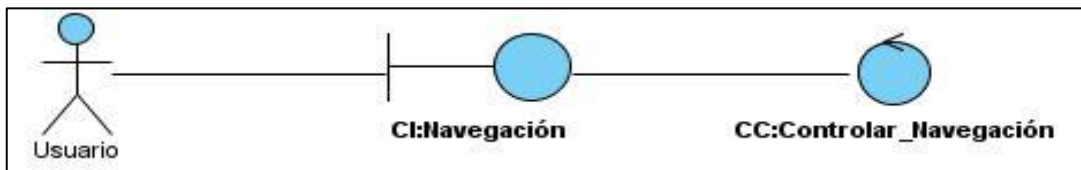


Figura 4. DCA_CU Controlar navegación

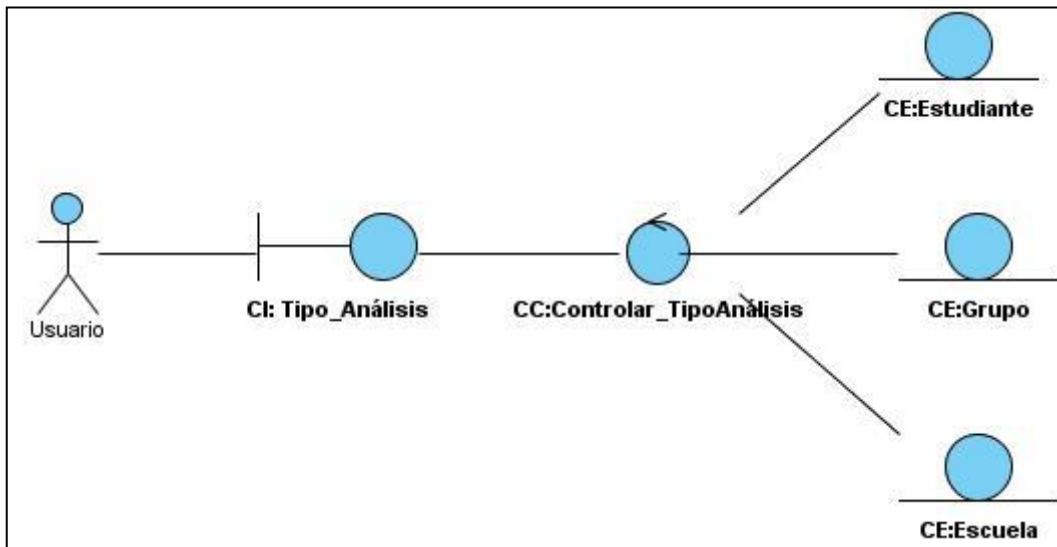


Figura 5. DCA_CU Configurar tipo de análisis

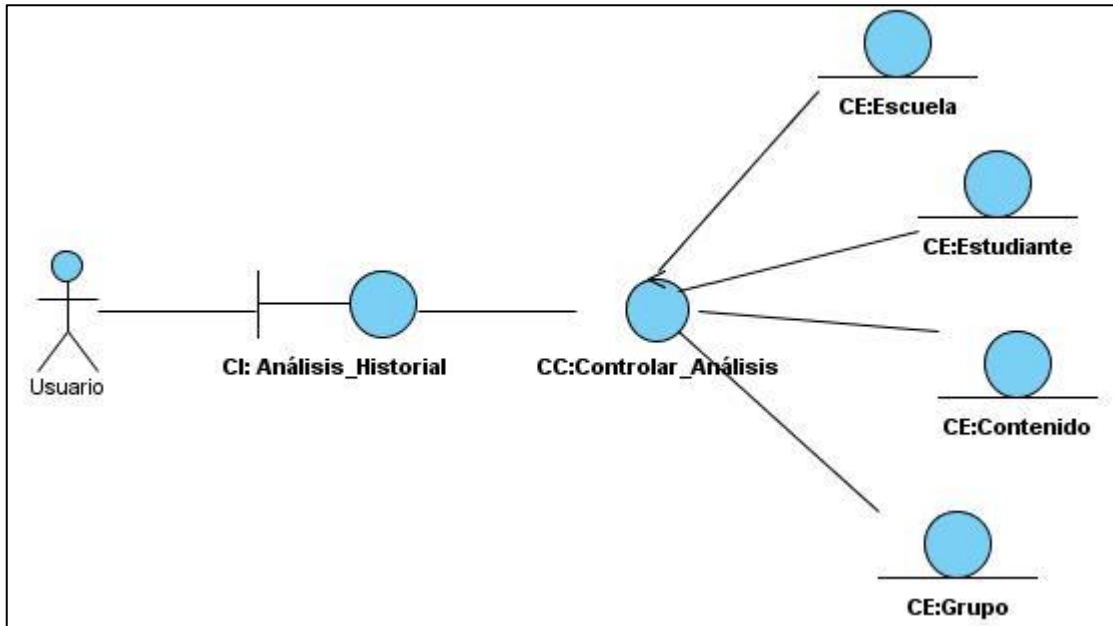


Figura 6. DCA_CU Realizar análisis del historial

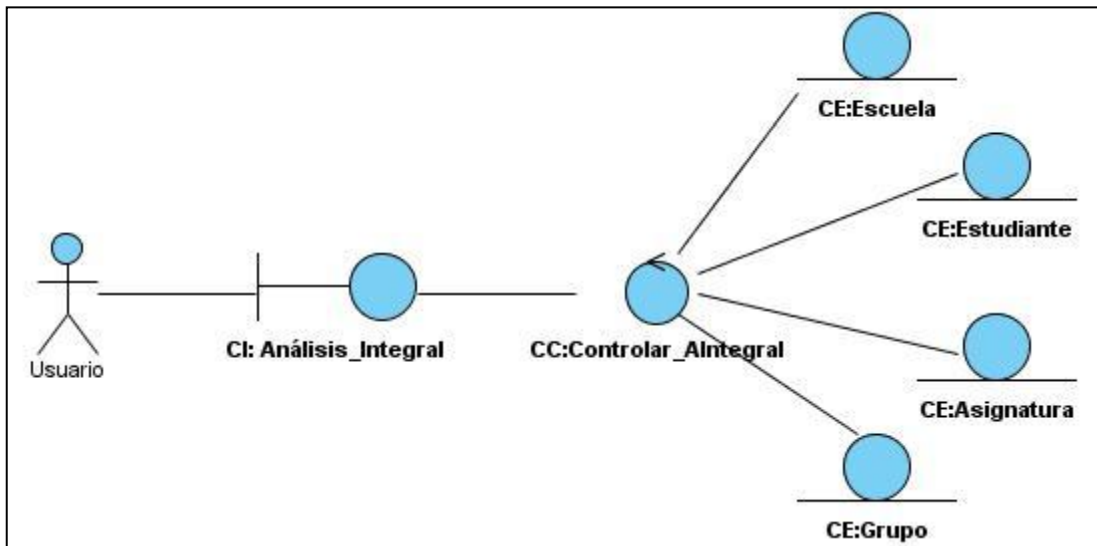


Figura 7. DCA_CU Realizar análisis integral

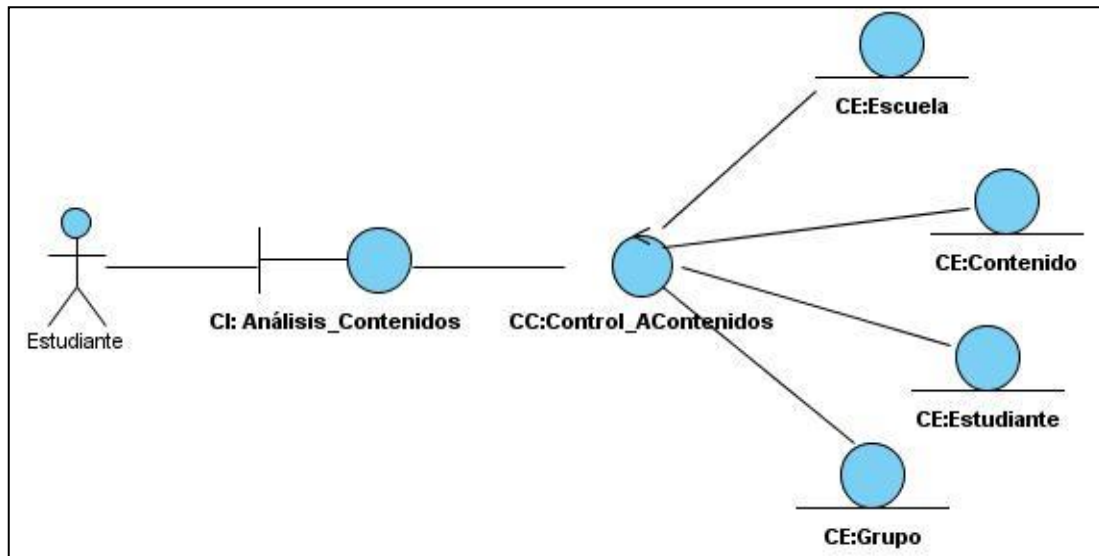


Figura 8. DCA_CU Realizar análisis de los contenidos

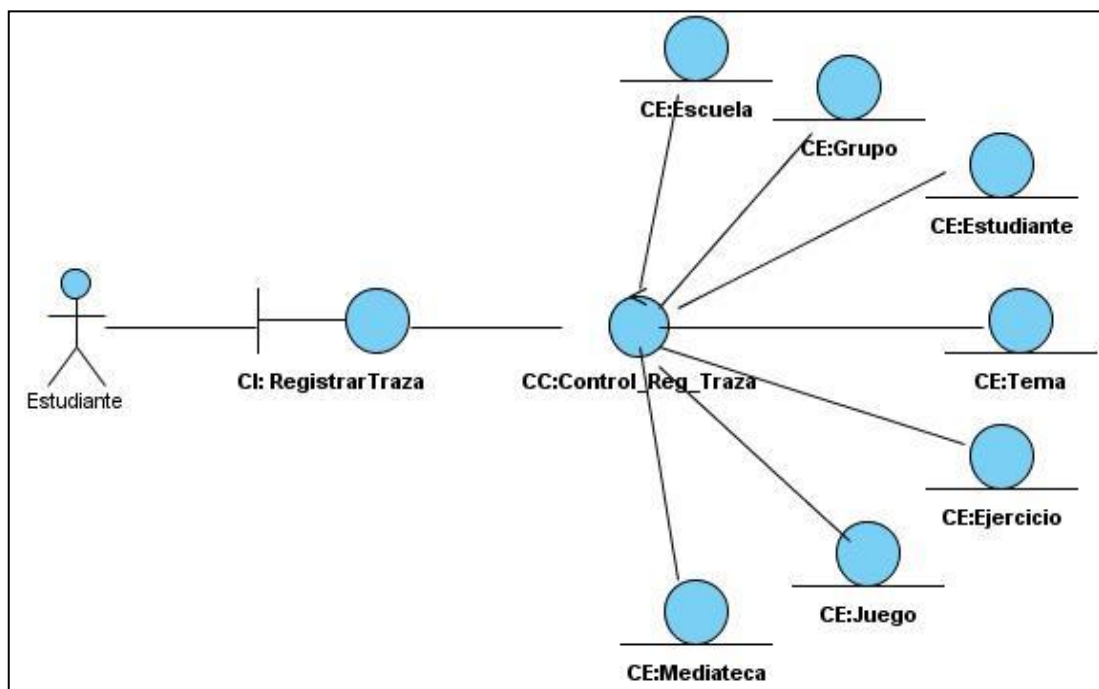


Figura 9. DCA_CU Registrar traza

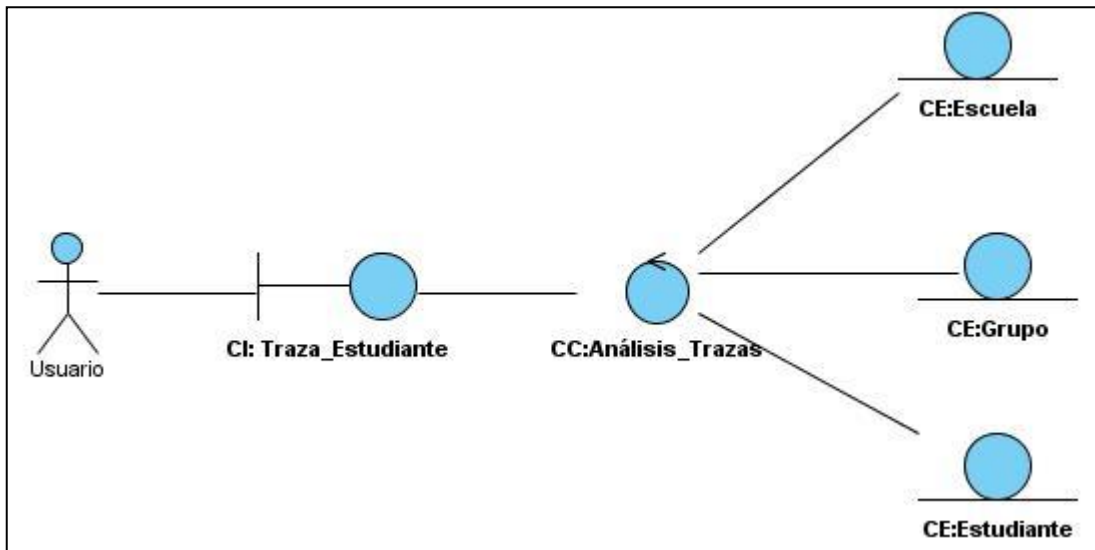


Figura 10. DCA_CU Realizar análisis de trazas

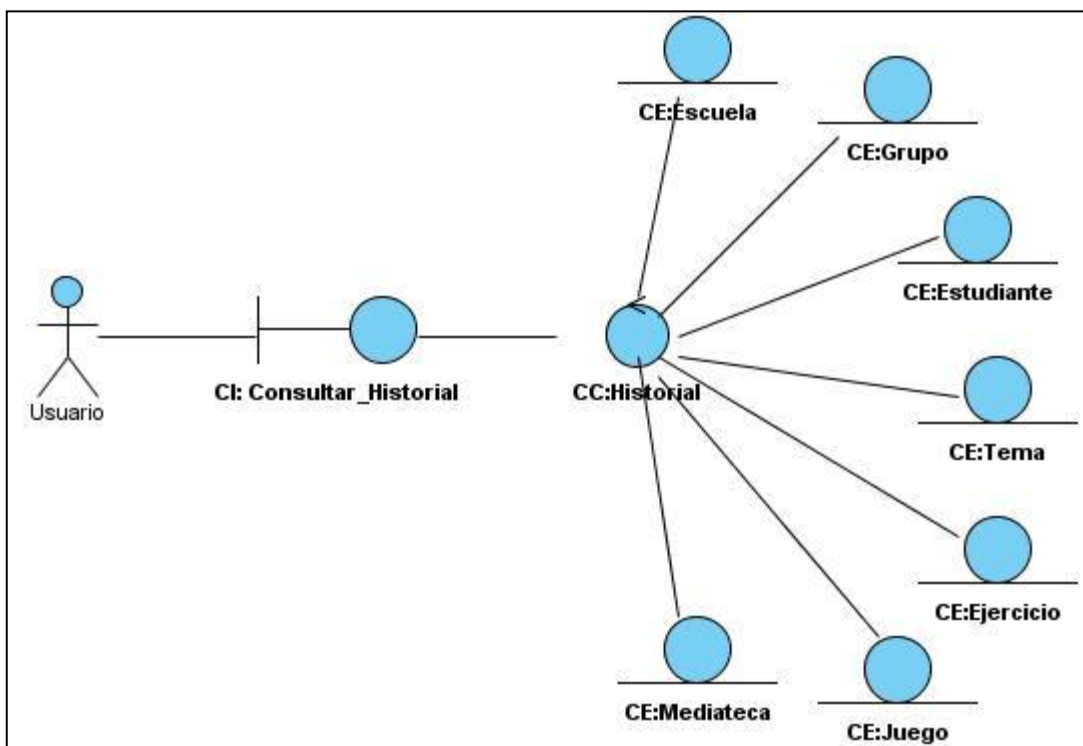


Figura 11. DCA_CU Consultar historial del estudiante

3.3 Diagrama de Clases del Diseño

Durante el Diseño se toman decisiones para darle cumplimiento a los requerimientos funcionales y no funcionales de un sistema. El Diagrama de Clases para las aplicaciones web se diferencia un poco del resto de las aplicaciones que se construyen, ya que en ella son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación. Un Diagrama de Clases presenta las clases del sistema con sus relaciones estructurales y de herencia. En el caso de las aplicaciones web, el Diagrama de Clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase.

A continuación se muestran algunos de los Diagramas de Clases del Diseño con estereotipos web para cada uno de los Casos de Uso del Sistema.

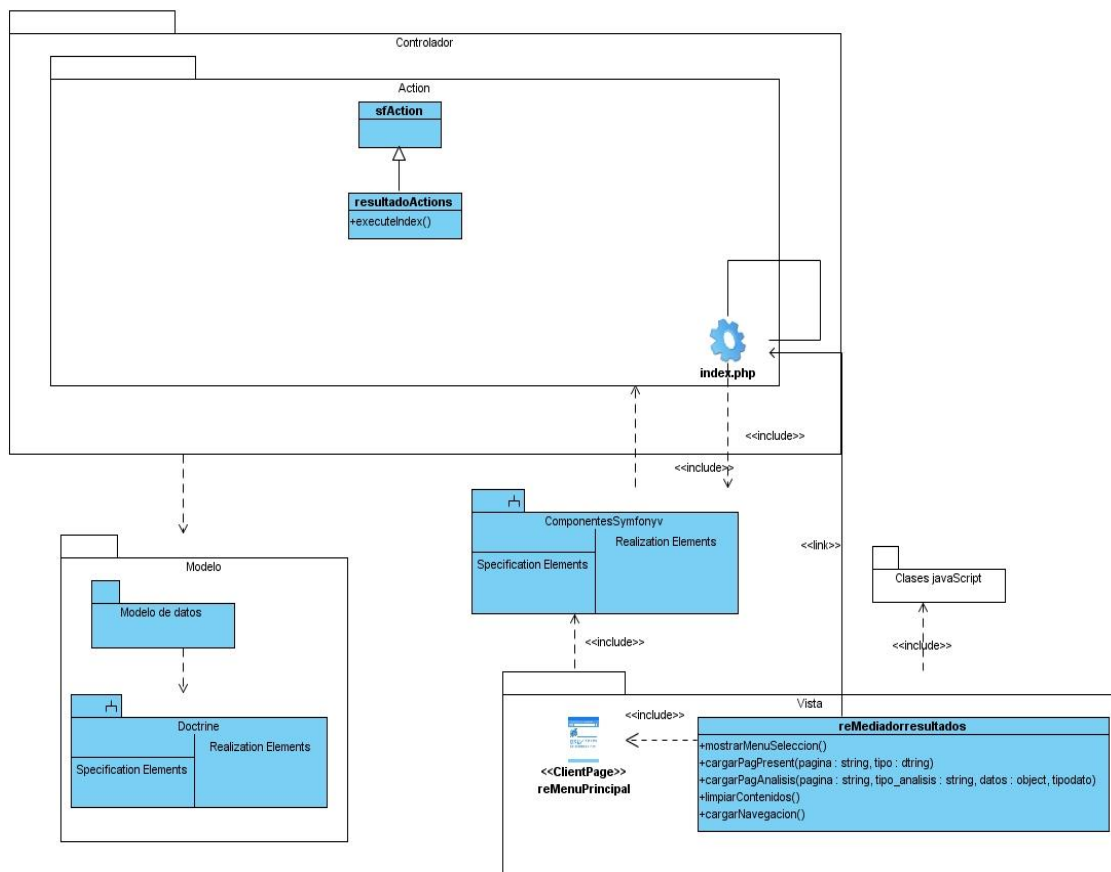


Figura 12. DCD_CU Ver tipo de reporte

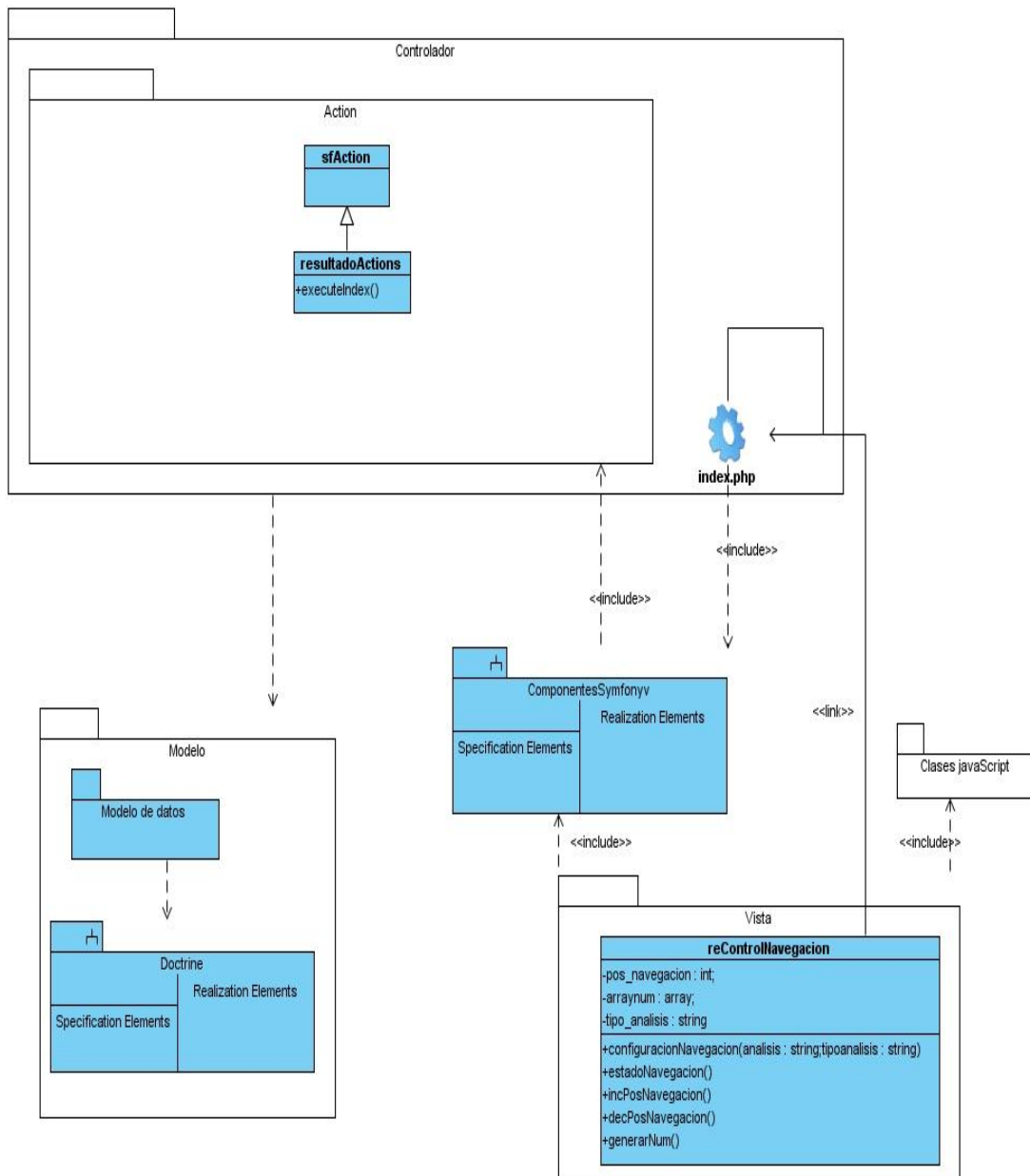


Figura 13. DCD_CU Controlar navegación

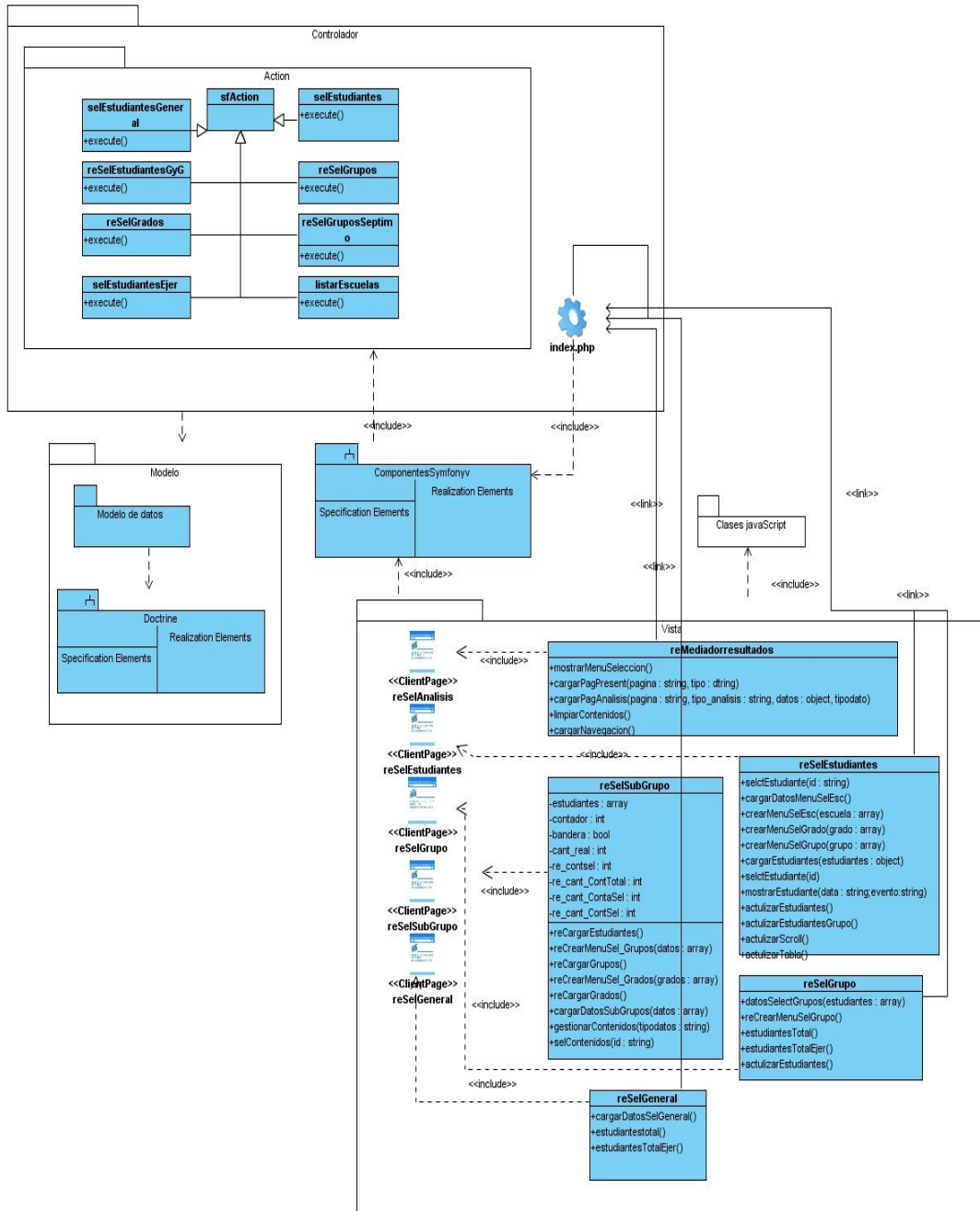


Figura 14. DCD_CU Configurar tipo de análisis

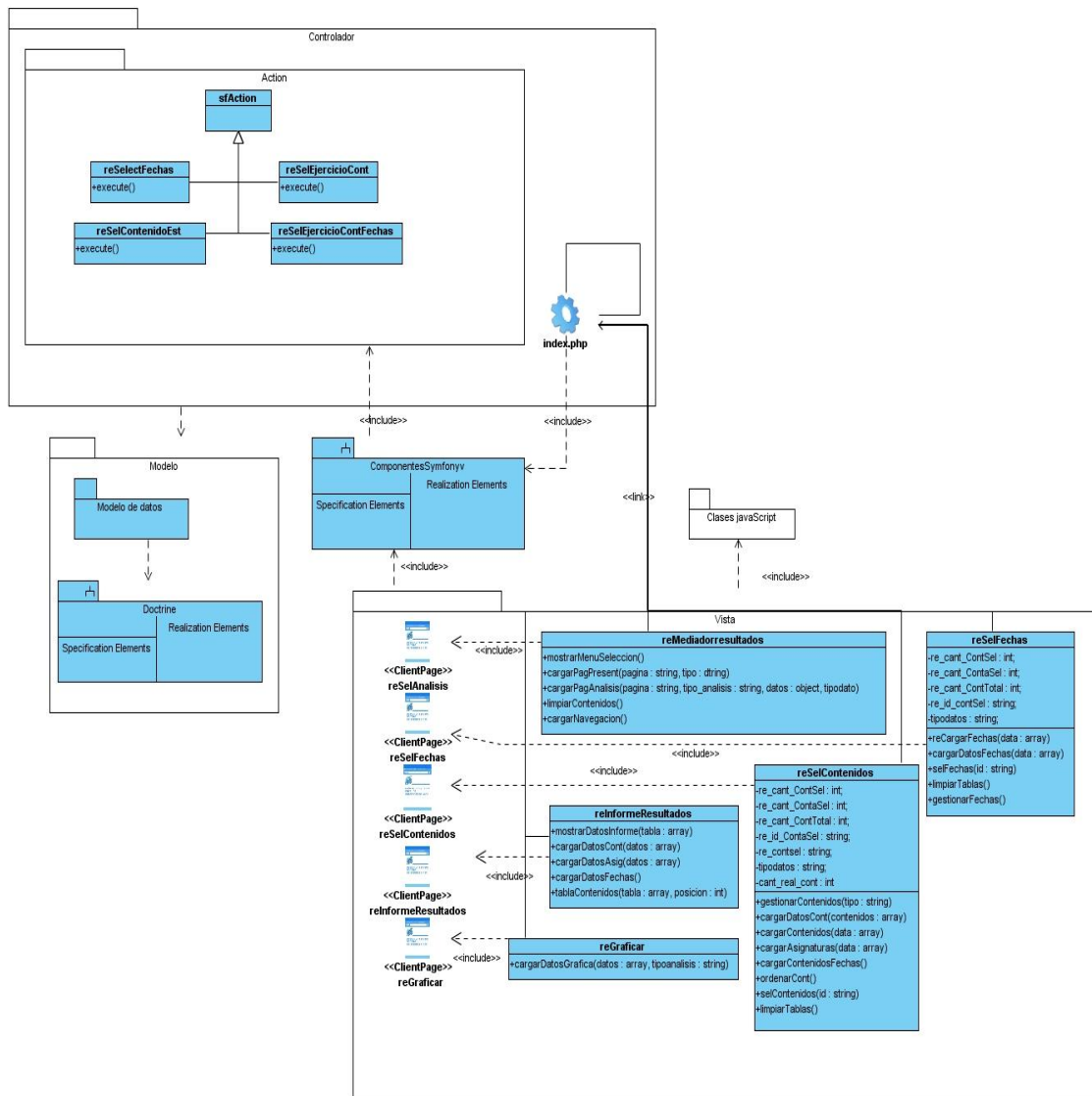


Figura 15. DCD_CU Realizar análisis del historial

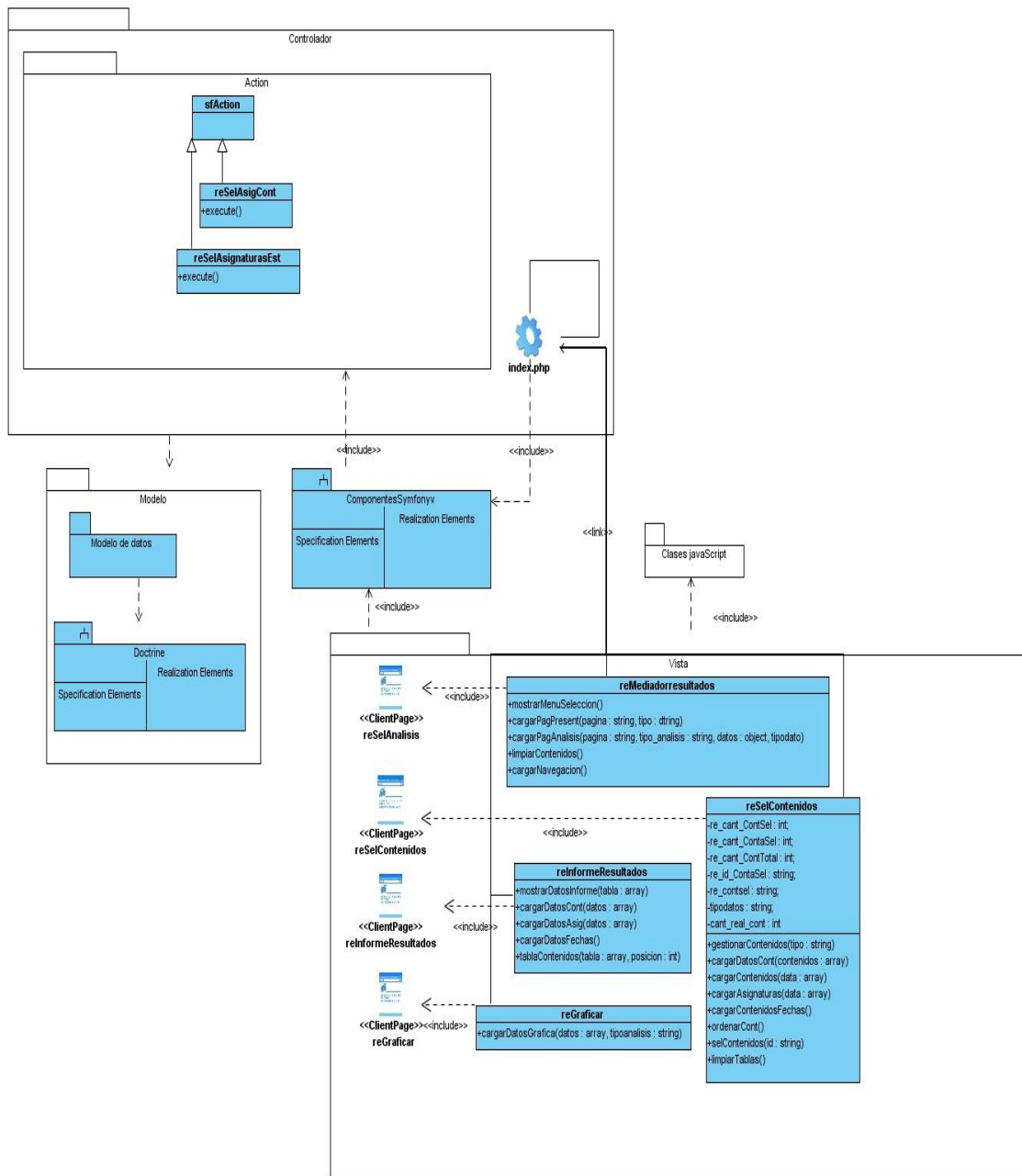


Figura 16. DCD_CU Realizar análisis integral

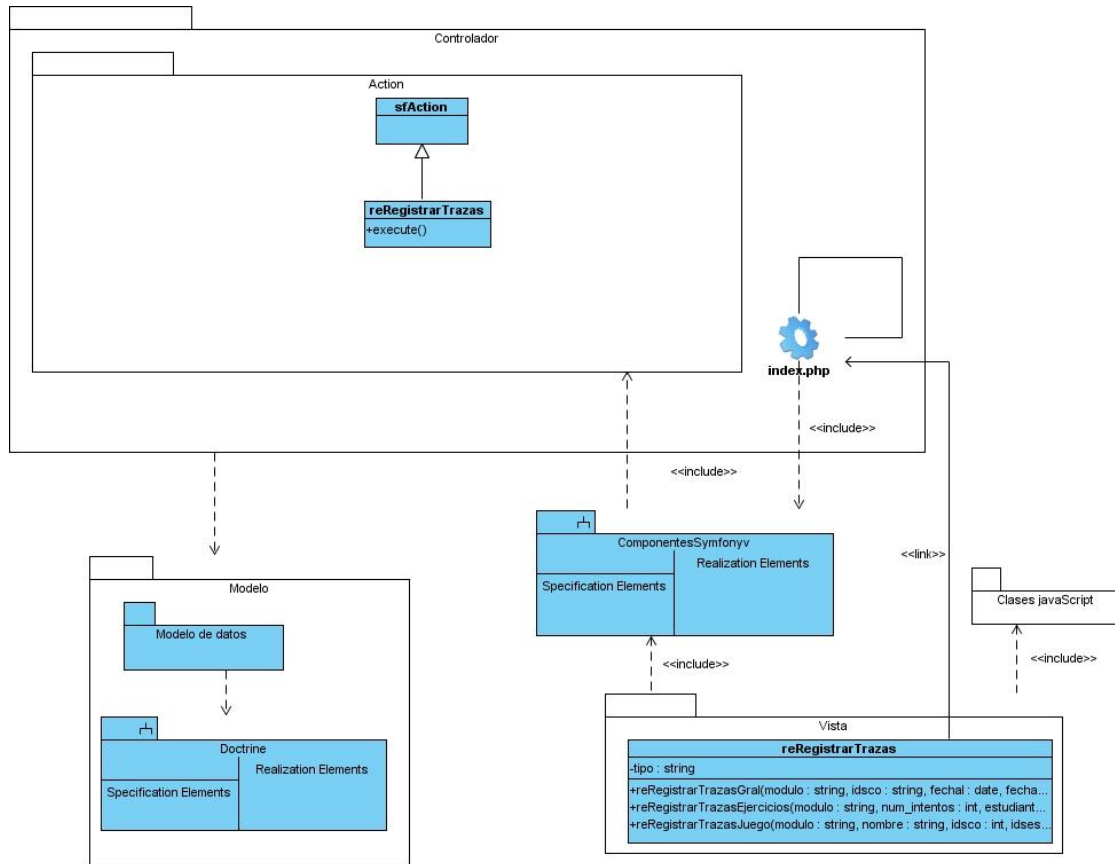


Figura 17. DCD_CU Registrar traza

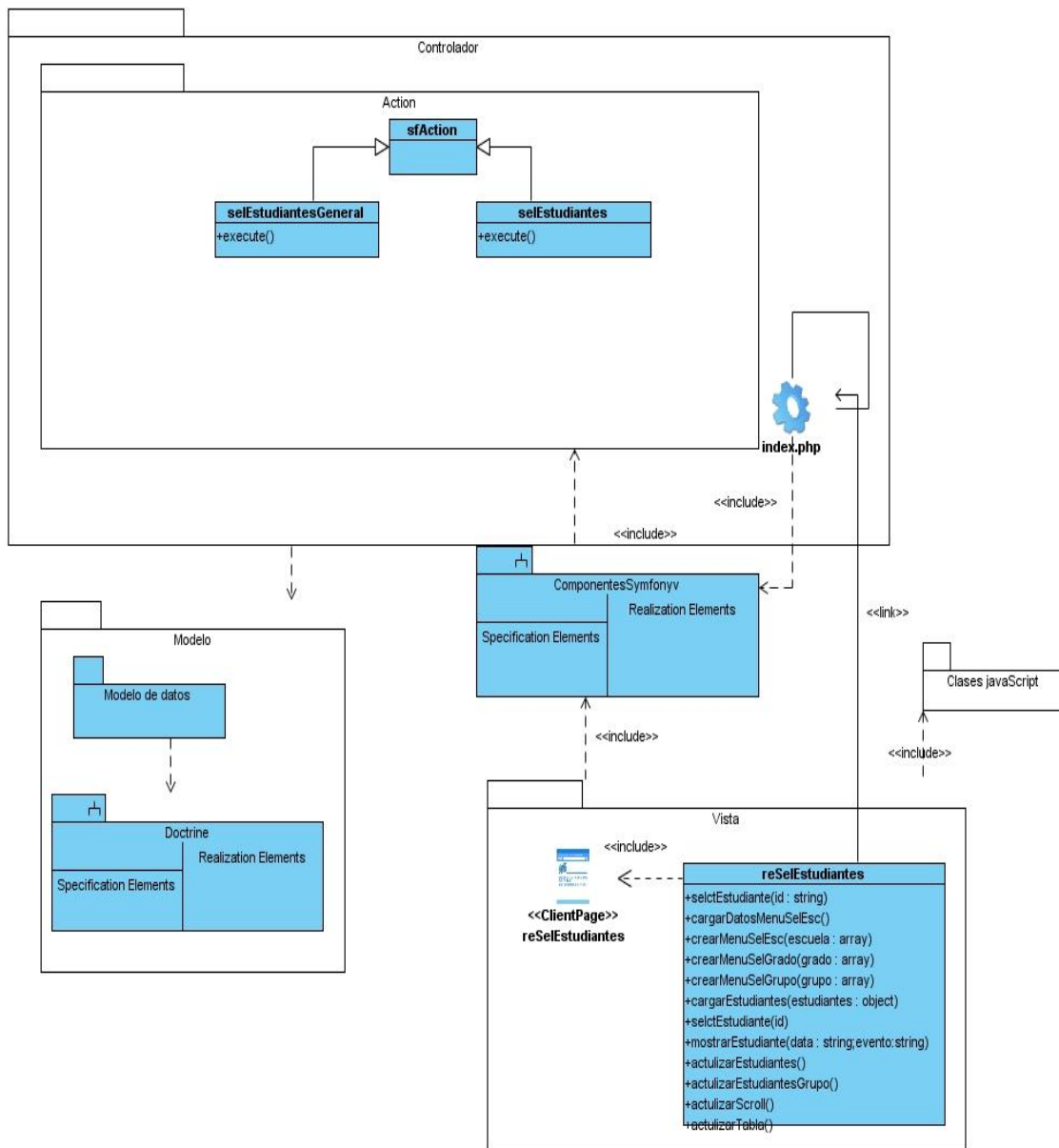


Figura 18. DCD_CU Realizar análisis de trazas

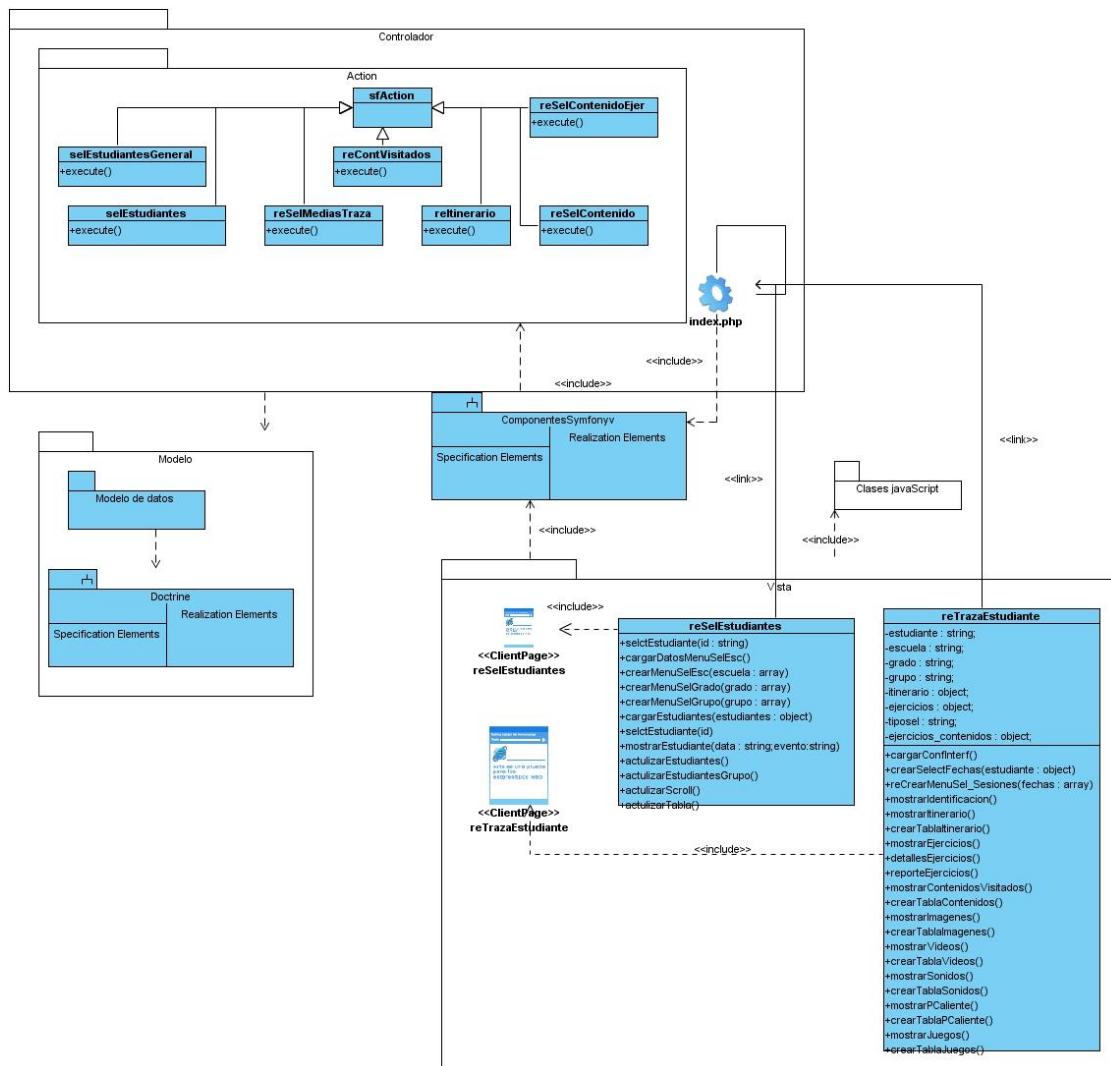


Figura 19. DCD_CU Consultar historial del estudiante

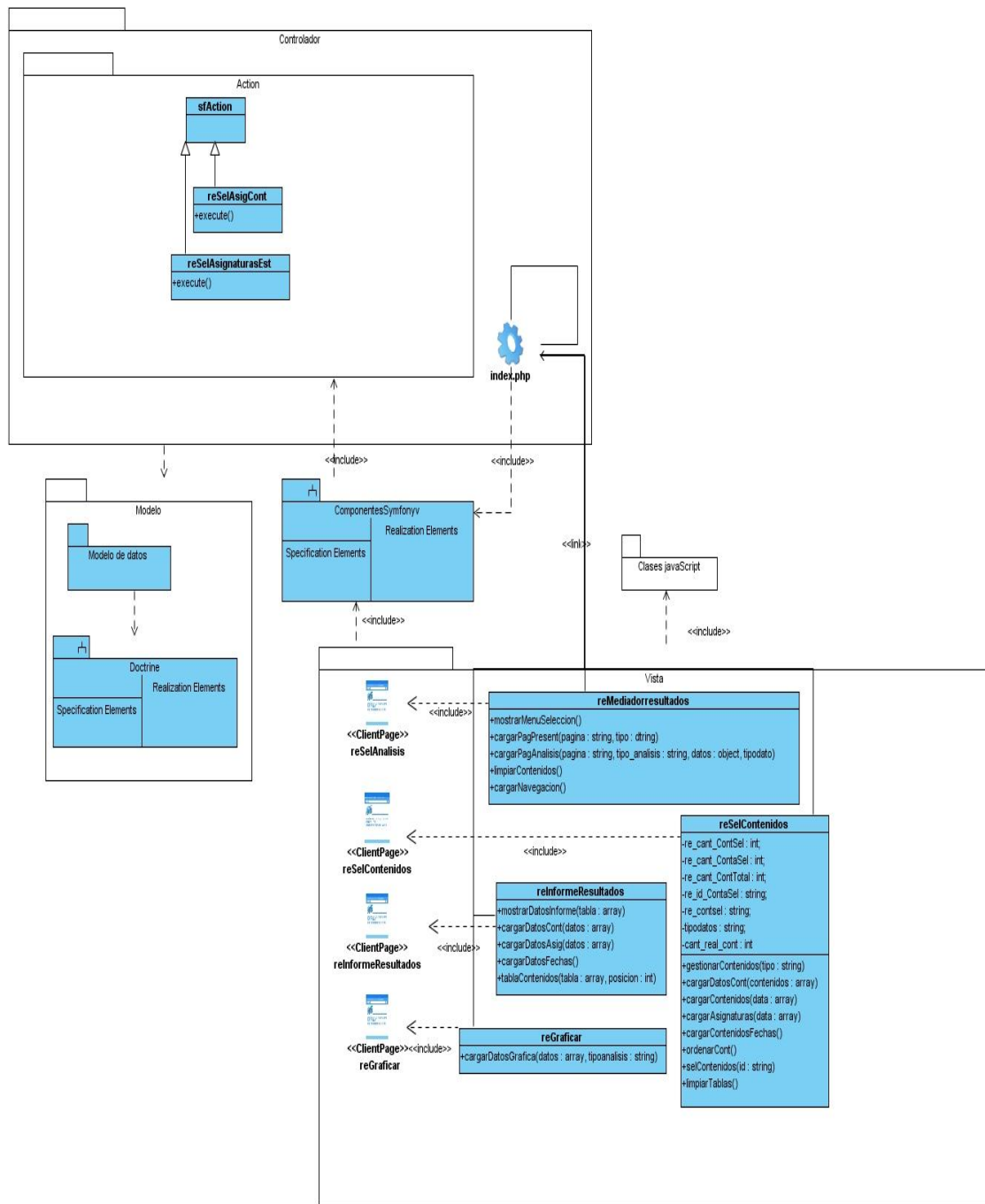


Figura 20. DCD_CU Realizar análisis de los contenidos

3.4 Diseño de la base de datos

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

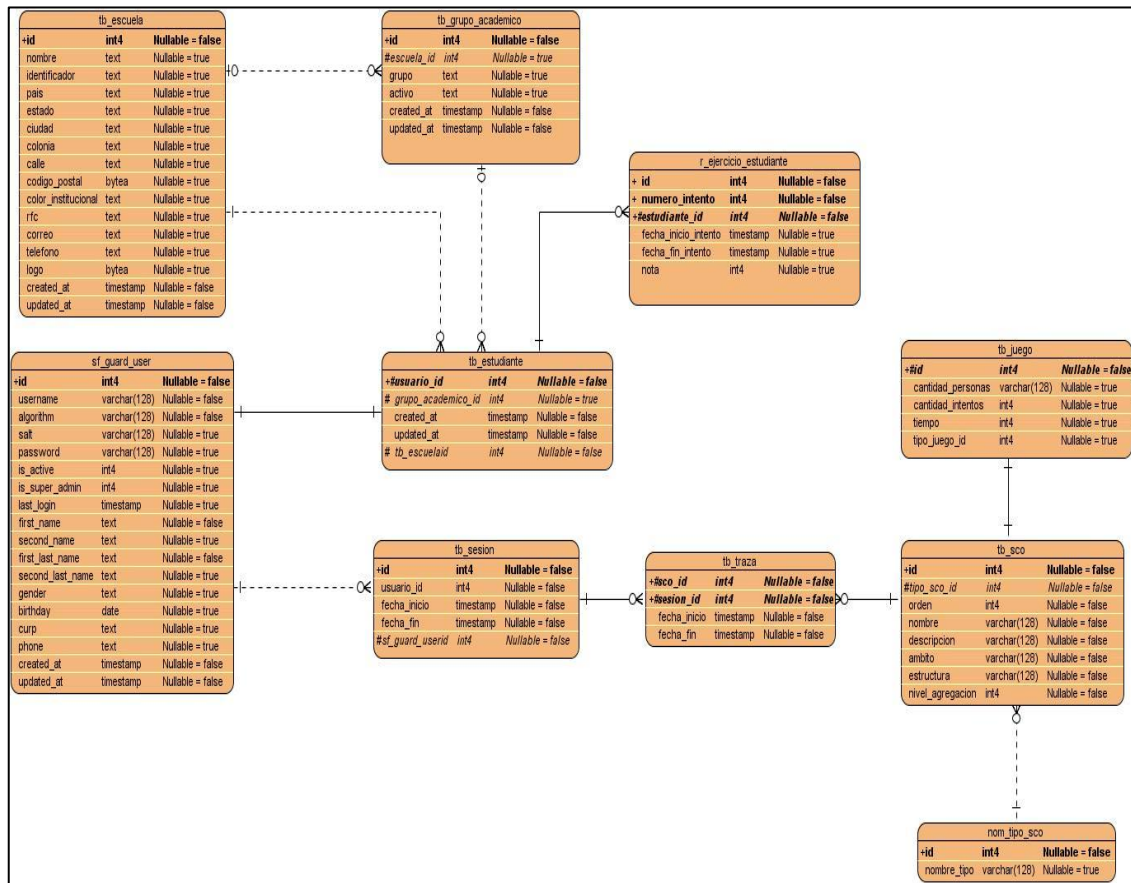


Figura 21. Modelo Entidad Relación

3.5 Conclusiones

En el presente capítulo se diseñó el Modelo del Análisis, dentro del cual se modelaron los diferentes Diagramas de Clases del Análisis de cada caso de uso. Se diseñaron los Diagramas de Clases del Diseño de cada funcionalidad y el Modelo Entidad Relación propuesto.

CAPÍTULO 4: Implementación del sistema

4.1 Introducción

En el presente capítulo se expondrán los Diagramas de Despliegue y los Diagramas de Componentes, los cuales tienen gran importancia dentro del flujo de trabajo de Implementación.

4.2 Modelo de Implementación

El flujo de trabajo de Implementación en la fase de Elaboración va encaminado a implementar la arquitectura que se ha definido. Está fuertemente determinado por el lenguaje de programación. También describe cómo los elementos del Modelo del Diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo con los nodos específicos en el Modelo de Despliegue. Los Diagramas de Despliegue y Componentes conforman lo que se conoce como un Modelo de Implementación al describir los componentes, su organización y la dependencia entre los nodos físicos en los que funcionará la aplicación.

4.2.1 Diagrama de Despliegue

En un Diagrama de Despliegue se representan las relaciones físicas entre los componentes que existen en el software y el hardware, que no son más que los procesos que se ejecutan en los mismos. Un Diagrama de Despliegue es un grafo de nodos conectados por enlaces de comunicación.

A continuación se presenta el Modelo de Despliegue del sistema, donde en este caso son dos, según lo solicitado por el cliente, por lo que se utilizará algunas de estas dos variantes. Estos modelos están compuestos por dos o tres nodos según el que se seleccione. El Diagrama de Despliegue 1 está compuesto por tres nodos, donde en el nodo Servidor se instala el servidor web Apache y SGBD PostgreSQL. El nodo PC cliente, mediante 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.

El Diagrama de Despliegue 2 representa al nodo PC cliente, el cual va a tener instalado el servidor web y la base de datos, además cuenta con todas las funcionalidades necesarias para lograr un adecuado funcionamiento de la aplicación

y la impresora como dispositivo utilizado por el sistema para posibilitar la impresión de documentos.

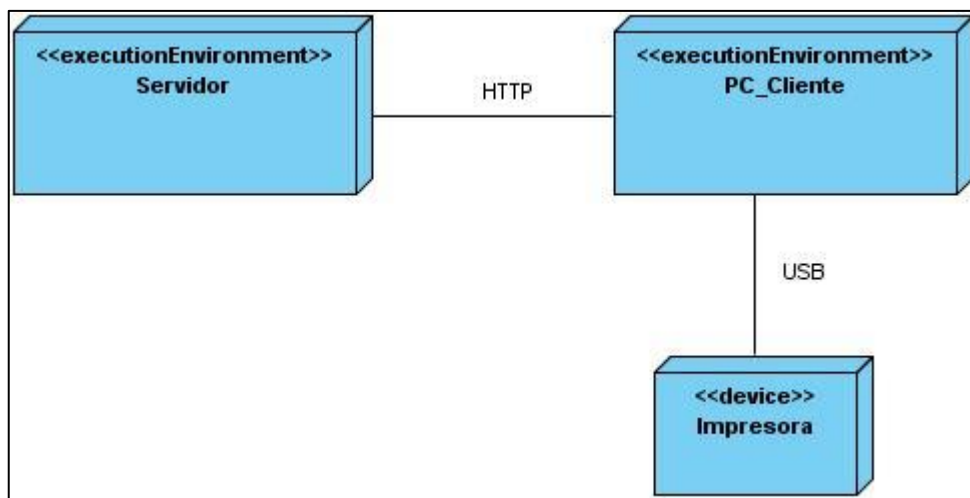


Figura 22. Diagrama de Despliegue 1

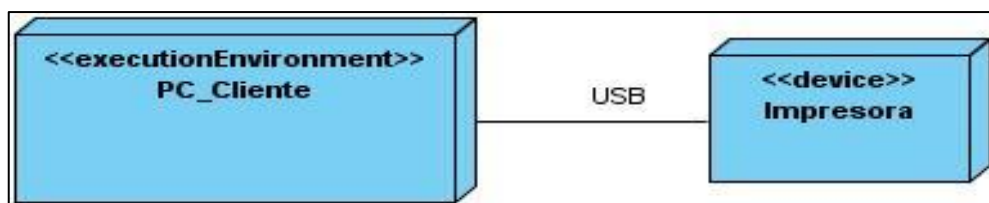


Figura 23. Diagrama de Despliegue 2

4.2.2 Diagramas de Componentes

Un Diagrama de Componentes no es más que un grafo de componentes unidos a través de relaciones que pueden ser de compilación o de ejecución, y también se pueden representar las interfaces de esos componentes. Las interfaces son las descripciones de las operaciones que se realizan en esos componentes y en esos subsistemas. Estos diagramas muestran un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. Son utilizados para estructurar el Modelo de Implementación en términos de Subsistemas de Implementación y mostrar las relaciones entre los elementos de implementación.

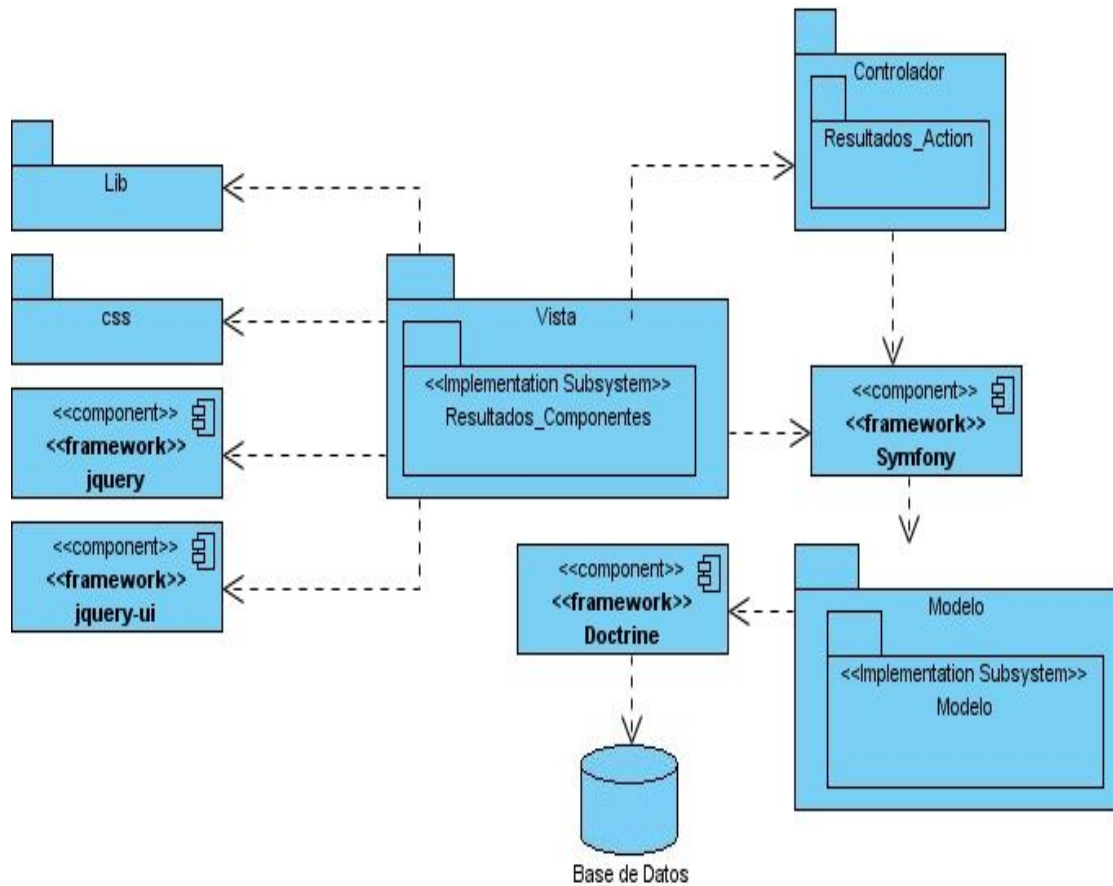


Figura 24. Diagrama de Componentes General

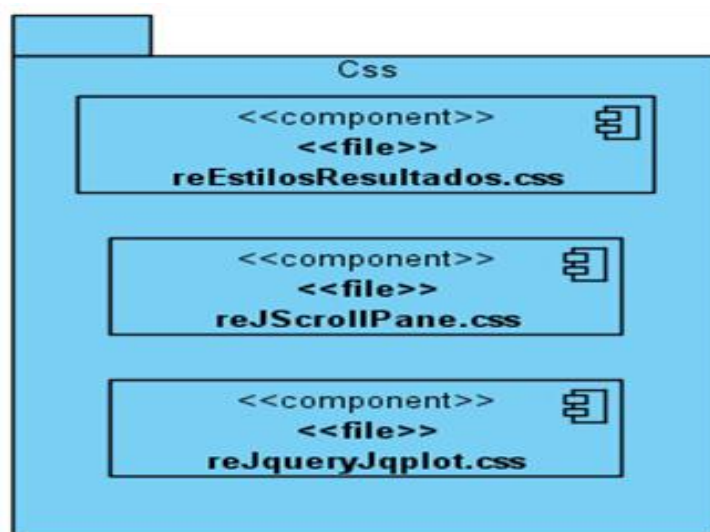


Figura 25. Diagrama de Componentes del paquete CSS



Figura 26. Diagrama de Componentes del paquete Controlador

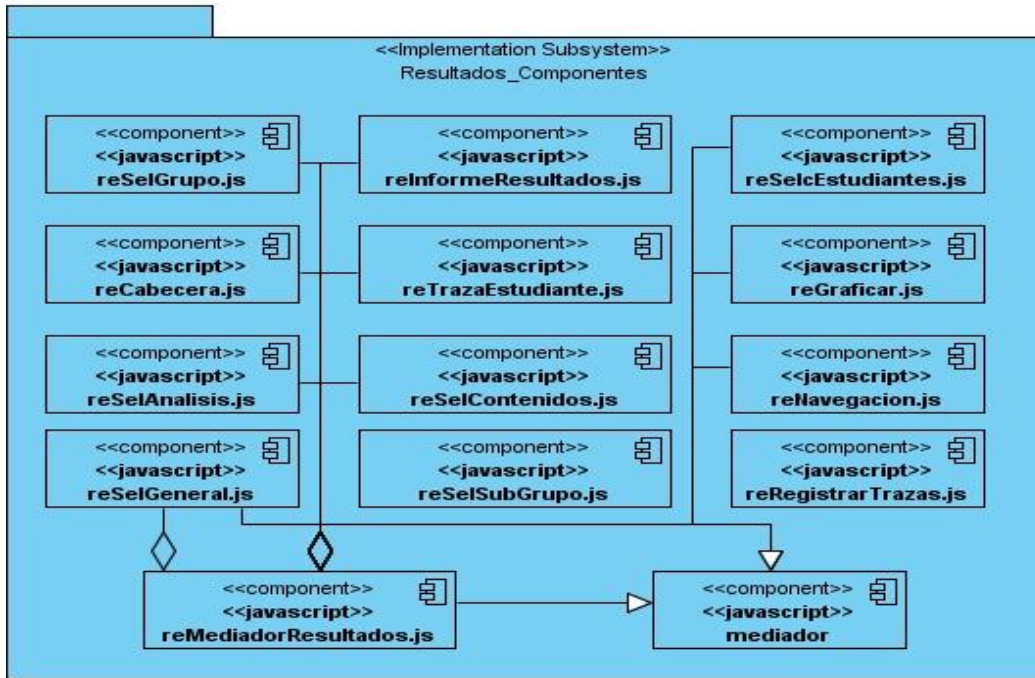


Figura 27. Diagrama de Componentes del subsistema Vista

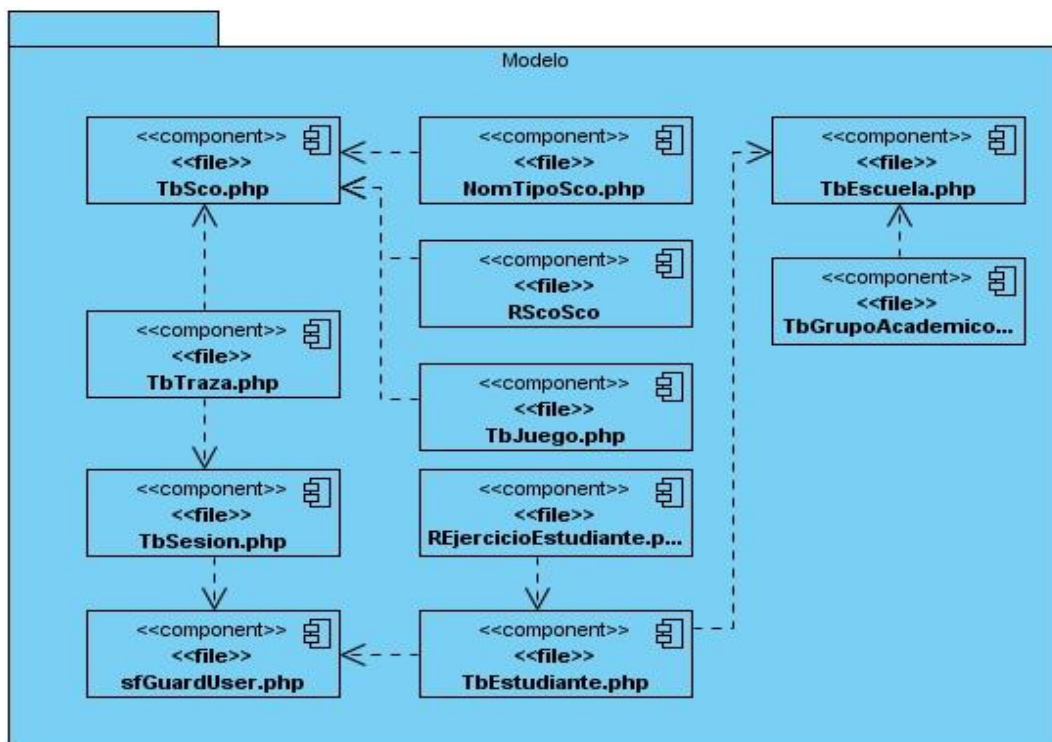


Figura 28. Diagrama de Componentes del subsistema Modelo

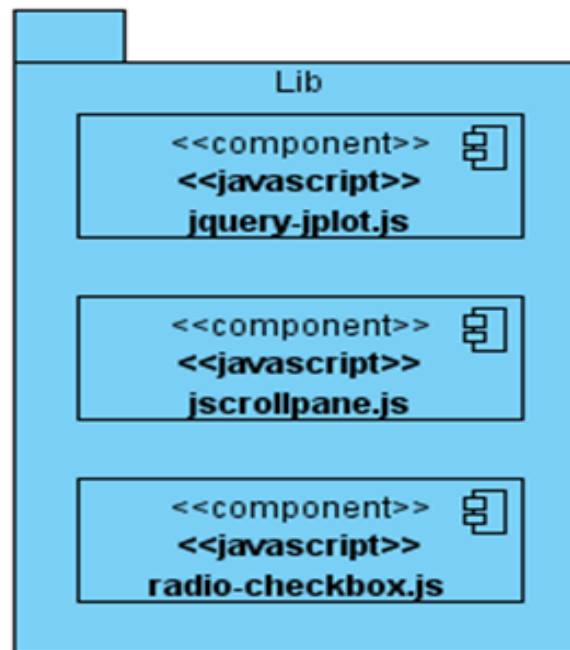


Figura 29. Diagrama de Componentes del paquete Lib

4.3 Conclusiones

En este capítulo fue presentado cómo está construido el sistema a partir de los Diagramas de Despliegue, ilustrando cuáles serán los nodos que se utilizarán para la implantación de la aplicación. También fue mostrado el Diagrama de Componentes del sistema en general, así como, los Diagramas de Componentes de cada paquete y subsistema.

CONCLUSIONES GENERALES

Luego de haber realizado las investigaciones y estudios requeridos para el desarrollo del módulo y una vez concluido con las fases de Concepción, Elaboración y Construcción:

- ✚ Se analizaron varias herramientas que contienen un módulo para llevar el registro de los estudiantes durante su interacción con la aplicación, pero ninguna de ellas soporta el modelo pedagógico establecido para la Colección El Navegante y algunas no cumplen con los requerimientos tecnológicos definidos en la arquitectura establecida en el proyecto.
- ✚ Al seguir las fases que define la metodología RUP, se transitó por la Concepción y Elaboración del módulo Resultados perteneciente a los nuevos productos que conforman la Colección El Navegante en su versión multiplataforma, obteniendo toda la documentación necesaria para garantizar el futuro mantenimiento del mismo.
- ✚ Se implementó un módulo que permite llevar un control sobre la interacción de cada uno de los estudiantes, con los diferentes módulos de los productos que conforman la Colección El Navegante, en su versión multiplataforma.

RECOMENDACIONES

Teniendo como base el estudio efectuado durante la realización de esta investigación, se propone como recomendación, realizar las actividades propuestas por la metodología de desarrollo RUP en el flujo de trabajo Pruebas con vista a garantizar la calidad del producto.

REFERENCIAS BIBLIOGRÁFICAS

1. Proyecto CITA2/ALFA. [En línea] 8 de 11 de 2005. [Citado el: 03 de 02 de 2010.] <http://www.iered.org/ev/mod/resource/view.php?id=364>.
2. UM. [En línea] [Citado el: 03 de 02 de 2010.] <http://www.um.es/docencia/barzana/LAGP/lagp2.html>.
3. Molpeceres, Alberto. Willydev. [En línea] 15 de 12 de 2002. [Citado el: 02 de 02 de 2010.] <http://www.willydev.net/descargas/articulos/general/cualxpddrup.PDF>.
4. José H. Canós, Patricio Letelier y M^a Carmen Penadés. Willydev. [En línea] [Citado el: 02 de 02 de 2010.] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
5. Itera. Itera. [En línea] [Citado el: 03 de 02 de 2010.] http://www.iteraprocess.com/index.php?option=com_content&task=view&id=18&Itemid=42.
6. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. España : Addison Wesley.
7. Ivon Tarazona y Oriana Gomez. FACYT. [En línea] [Citado el: 01 de 02 de 2010.] <http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/download/exposiciones2005-2006/uml.pdf>.
8. Dávila, Jose A Vela. Ponencia en Seminario. [En línea] 2008. [Citado el: 01 de 02 de 2010.] <http://www.cimat.mx/Eventos/seminariotecnologias08/javd.pdf>.
9. Rincondelvago. [En línea] 17 de 09 de 2003. [Citado el: 04 de 02 de 2010.] http://html.rincondelvago.com/herramientas-case_2.html.
10. Free Download Manager. [En línea] [Citado el: 03 de 02 de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
11. Gustavo F. Torrealday. Torrealday Informática. [En línea] [Citado el: 04 de 02 de 2010.] <http://www.torrealday.com.ar/articulos/articulo006.htm>.
12. Anibal de la Torre. PHPNuke. [En línea] 2006. [Citado el: 02 de 02 de 2010.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
13. Victoria. Definición ABC. [En línea] 12 de 01 de 2009. [Citado el: 04 de 02 de 2010.] <http://www.definicionabc.com/tecnologia/html.php>.
14. Valdéz, Damián Pérez. Maestros del web. [En línea] [Citado el: 05 de 02 de 2010.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.

15. Cruciatus. Mozilla Developer Center. [En línea] [Citado el: 06 de 02 de 2010.] https://developer.mozilla.org/index.php?title=Es/Gu%C3%ADa_JavaScript_1.5/.
16. Ventajas y desventajas del Personal Home Page. [En línea] 05 de 02 de 2010. <http://www.creargratisunapaginaweb.com/PHP/Ventajas-y-desventajas-del-Personal-Home-Page-4/>.
17. Warden, James Shore y Shane. *The Art of Agile Development*. 2007.
18. JQuery. [En línea] [Citado el: 01 de 02 de 2010.] <http://www.jquery.cl/>.
19. Gomez, Diego. DosIdeas. [En línea] 19 de 09 de 2008. [Citado el: 02 de 02 de 2010.] <http://www.dosideas.com/noticias/java/238-aplicaciones-ria-y-jquery.html>.
20. Librosweb. [En línea] [Citado el: 03 de 02 de 2010.] http://librosweb.es/symfony_1_0/capitulo1/symfony_en_pocas_palabras.html.
21. CAVSI. [En línea] [Citado el: 01 de 02 de 2010.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd>.
22. PostgreSQL. [En línea] [Citado el: 02 de 02 de 2010.] <http://www.postgresql.org/about/>.
23. Solcre Technology Solutions. [En línea] [Citado el: 03 de 02 de 2010.] http://www.solcre.com/files/ventajas_de_las_aplicaciones_web.pdf.
24. Jesús Vega. Informática. [En línea] 31 de 03 de 2002. [Citado el: 02 de 02 de 2010.] <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>.
25. LinuxParaTodos. [En línea] 12 de 11 de 2008. [Citado el: 02 de 02 de 2010.] <http://www.linuxparatodos.net/portal/staticpages/index.php?page=servidor-web>.
26. Babylon. [En línea] [Citado el: 04 de 02 de 2010.] http://www.babylon.com/definicion/entorno_de_desarrollo/Spanish.
27. Netbeans. [En línea] [Citado el: 02 de 02 de 2010.] <http://netbeans.org>.
28. Potencier, François Zaninotto y Fabien. *Symfony, la guía definitiva*.
29. Integrax. MultiEducacion. [En línea] Integrax. [Citado el: 01 de 02 de 20010.] <http://multieducacion.com/SCORM.aspx>.

BIBLIOGRAFÍA

1. Proyecto CITA2/ALFA. [En línea] 8 de 11 de 2005. [Citado el: 03 de 02 de 2010.]
<http://www.ired.org/ev/mod/resource/view.php?id=364>.
2. UM. [En línea] [Citado el: 03 de 02 de 2010.]
<http://www.um.es/docencia/barzana/LAGP/lagp2.html>.
3. Molpeceres, Alberto. Willydev. [En línea] 15 de 12 de 2002. [Citado el: 02 de 02 de 2010.] <http://www.willydev.net/descargas/articulos/general/cualxpddrup.PDF>.
4. José H. Canós, Patricio Letelier y M^a Carmen Penadés. Willydev. [En línea] [Citado el: 02 de 02 de 2010.] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
5. Itera. Itera. [En línea] [Citado el: 03 de 02 de 2010.]
http://www.iteraprocess.com/index.php?option=com_content&task=view&id=18&Itemid=42.
6. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. España : Addison Wesley.
7. Ivon Tarazona y Oriana Gomez. FACYT. [En línea] [Citado el: 01 de 02 de 2010.]
<http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/download/exposiciones2005-2006/uml.pdf>.
8. Dávila, Jose A Vela. Ponencia en Seminario. [En línea] 2008. [Citado el: 01 de 02 de 2010.] <http://www.cimat.mx/Eventos/seminariotecnologias08/javd.pdf>.
9. Rincondelvago. [En línea] 17 de 09 de 2003. [Citado el: 04 de 02 de 2010.]
http://html.rincondelvago.com/herramientas-case_2.html.
10. Free Download Manager. [En línea] [Citado el: 03 de 02 de 2010.]
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
11. Gustavo F. Torrealday. Torrealday Informática. [En línea] [Citado el: 04 de 02 de 2010.] <http://www.torrealday.com.ar/articulos/articulo006.htm>.
12. Anibal de la Torre. PHPNuke. [En línea] 2006. [Citado el: 02 de 02 de 2010.]
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
13. Victoria. Definición ABC. [En línea] 12 de 01 de 2009. [Citado el: 04 de 02 de 2010.] <http://www.definicionabc.com/tecnologia/html.php>.
14. Valdéz, Damián Pérez. Maestros del web. [En línea] [Citado el: 05 de 02 de 2010.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.

15. Cruciatus. Mozilla Developer Center. [En línea] [Citado el: 06 de 02 de 2010.]
https://developer.mozilla.org/index.php?title=Es/Gu%C3%ADa_JavaScript_1.5/.
16. Ventajas y desventajas del Personal Home Page. [En línea] 05 de 02 de 2010.
<http://www.creargratisunapaginaweb.com/PHP/Ventajas-y-desventajas-del-Personal-Home-Page-4/>.
17. Warden, James Shore y Shane. *The Art of Agile Development*. 2007.
18. JQuery. [En línea] [Citado el: 01 de 02 de 2010.] <http://www.jquery.cl/>.
19. Gomez, Diego. DosIdeas. [En línea] 19 de 09 de 2008. [Citado el: 02 de 02 de 2010.] <http://www.dosideas.com/noticias/java/238-aplicaciones-ria-y-jquery.html>.
20. Librosweb. [En línea] [Citado el: 03 de 02 de 2010.]
http://librosweb.es/symfony_1_0/capitulo1/symfony_en_pocas_palabras.html.
21. CAVSI. [En línea] [Citado el: 01 de 02 de 2010.]
<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd>.
22. PostgreSQL. [En línea] [Citado el: 02 de 02 de 2010.]
<http://www.postgresql.org/about/>.
23. Solcre Technology Solutions. [En línea] [Citado el: 03 de 02 de 2010.]
http://www.solcre.com/files/ventajas_de_las_aplicaciones_web.pdf.
24. Jesús Vega. Informática. [En línea] 31 de 03 de 2002. [Citado el: 02 de 02 de 2010.] <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>.
25. LinuxParaTodos. [En línea] 12 de 11 de 2008. [Citado el: 02 de 02 de 2010.]
<http://www.linuxparatodos.net/portal/staticpages/index.php?page=servidor-web>.
26. Babylon. [En línea] [Citado el: 04 de 02 de 2010.]
http://www.babylon.com/definicion/entorno_de_desarrollo/Spanish.
27. Netbeans. [En línea] [Citado el: 02 de 02 de 2010.] <http://netbeans.org>.
28. Potencier, François Zaninotto y Fabien. *Symfony, la guía definitiva*.
29. Iván Garcerant. [En línea] 10 de 07 de 2008. [Citado el: 10 de 02 de 2010.]
<http://synergix.wordpress.com/2008/07/10/modelo-de-dominio>.
30. Maestrosdelweb. [Citado el: 04 de 02 de 2010.]
<http://www.maestrosdelweb.com/editorial/javascript-facil-y-rapido-con-jquery>
31. NetBeans Community. [Citado el: 01 de 02 de 2010.]
http://www.netbeans.org/index_es.html
32. Javier J. Gutiérrez. Lsi. [En línea] [Citado el: 02 de 02 de 2010.]
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf34.
33. Juliana .DesarrolladorWeb. [En línea] [Citado el: 02 de 02 de 2010.] HYPERLINK
<http://www.desarrolloweb.com/contacta/87.html>

33. Monteiro Lazaro .DesarrolladorWeb. [En línea] [Citado el: 02 de 02 de 2010.]
http://www.desarrolloweb.com/articulos/26.php
34. Miguel Angel Alvarez .DesarrolladorWeb. [En línea] [Citado el: 02 de 02 de 2010.]
HYPERLINK *http://www.desarrolloweb.com/contacta/1.html*.
- 35.DesarrolladorWeb. [En línea] [Citado el: 02 de 02 de 2010.]
http://www.desarrolloweb.com/articulos/que-es-html.html
36. MiTecnologico. [En línea] [Citado el: 01 de 02 de 2010.]
http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion
37. Lcc. [En línea] [Citado el: 05 de 02 de 2010.]
http://www.lcc.uma.es/~eat/services/html-js/manual14.html
38. Ciberneta. [En línea] [Citado el: 03 de 02 de 2010.]
http://www.ciberneta.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php
39. Symfony.es. [En línea] [Citado el: 02 de 02 de 2010.]
http://www.symfony.es/que-es-symfony
40. HYPERLINK "http://www.symfony.es/que-es-symfony"
41. CubaMinRex. [En línea] [Citado el: 02 de 02 de 2010.]
http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Masivo_TIC.htm
42. GSInnova. [En línea] [Citado el: 02 de 02 de 2010.]
http://www.rational.com.ar/herramientas/rup.html
43. IBM. [En línea] [Citado el: 06 de 02 de 2010.] *http://www-01.ibm.com/software/awdtools/rup/*
44. FreeDownloadManager. [En línea] [Citado el: 06 de 02 de 2010.]
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/
45. Gadgetcabaret´s Weblog. [En línea] [Citado el: 04 de 02 de 2010.]
http://gadgetcabaret.wordpress.com/2007/12/05/netbeans-60-instalacion-y-caracteristicas/
46. Ángel Ciudad Ricardo. Monografias. [En línea] [Citado el: 02 de 02 de 2010.]
http://www.monografias.com/trabajos43/patron-modelo-vista/patron-modelo-vista.shtml
47. Javier Vidal. [En línea] [Citado el: 08 de 02 de 2010.] *http://javiervidal.net/los-servidores-web-mas-utilizados/*
48. Javier Vidal. [En línea] [Citado el: 08 de 02 de 2010.]
http://linux.ciberaula.com/articulo/linux_apache_intro.

GLOSARIO

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 crea aplicaciones más rápidas y con mejor respuesta a las acciones del usuario.

HTTP/ HTTPS: Protocolo de transferencia Hipertexto / Protocolo seguro de transferencia de Hipertexto; Protocolo utilizado en cada transacción de la web, el segundo es la versión segura de HTTP.

IDE: Integrated Development Environment (Ambiente Integrado de Desarrollo). Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

Linux: Sistema operativo libre tipo Unix.

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 "multi-medios". Los medios pueden ser variados, desde texto e imágenes, hasta animación, sonido, video.

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

Software Educativo: Se refiere a los programas educativos o didácticos, conocidos también como programas por ordenador, creados con la finalidad específica de ser utilizados para facilitar los procesos de enseñanza y aprendizaje.

UNIX: Sistema operativo multitarea, multiusuario. Gran parte de las características de otros sistemas más conocidos como MS-DOS están basadas en este sistema, muy extendido para grandes servidores.

WYSIWYG: (Viene del inglés "What you see is what you get"). Característica que tienen algunos editores de texto, o editores en general, que lo que ves en la pantalla

es el resultado final que obtendrás en la impresión o en el visionado final. Generalmente usado en editores de texto o HTML.

XML: Extensible Markup Language (Lenguaje de Etiquetado Extensible), conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes.

XSLT: Extensible Stylesheet Language Transformation (Transformación de Lenguaje de Hojas de Estilos Extensible), es transformar el contenido de un documento XML de origen en otro documento con un formato o estructura diferentes. Por ejemplo, se puede emplear para transformar el documento XML en HTML con el fin de usarlo en un sitio web o transformarlo en un documento que sólo contenga los campos necesarios para una aplicación.

Anexo_1

