



Universidad de las Ciencias Informáticas

Facultad 8

*Análisis, Diseño e Implementación del Módulo
Resultados de la Colección Multisaber, en su
versión multiplataforma*



multisaber

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

Autores: Daylén Delgado Fernández

Yasniel Benavides Triana

Tutores: Ing. Fidel Alberto Curbelo Rosell

Lic. Liana Isabel Araujo Pérez

Co_Tutor: Ing. Mailyn Cabrera Torres

Ciudad de La Habana, junio de 2010

“Año 52 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo “Análisis, Diseño e Implementación del Módulo Resultados de la Colección Multisaber, en su versión multiplataforma” y autorizamos a la Facultad 8 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autores:

Daylén Delgado Fernández

Yasniel Benavides Triana

Tutores:

Ing. Fidel Alberto Curbelo Rosell

Lic. Liana Isabel Araujo Pérez

PENSAMIENTO



“Podrán cortar todas las flores, pero nunca terminarán con la primavera”

Che Guevara

Dedicatoria

A mi mamá, por la oportunidad de existir, por su sacrificio en algún tiempo incomprendido, por su comprensión y confianza, por su amor y aliento, por ser la persona que siempre me lo ha dado todo y que ha logrado que sea quién soy hoy, porque sin su apoyo no hubiese sido posible la culminación de mi carrera profesional.

A mi abuelita Níca, que aunque ya no esté siempre la llevaré en mi corazón, y fue quien nunca dejó de decirme que mis estudios, en esta etapa, eran lo más importante. Gracias abuela por todo lo que me enseñaste. Este triunfo es también tuyo.

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

Daylén

A mi abuelita mami, que aunque ya no está sé que hubiese disfrutado mucho de este momento.

A mi mamá y mi hermanito, que son lo lindo de mi vida.

A la Revolución por hacer posibles los sueños de tantas personas.

Yasniel

Agradecimientos

Al término de esta etapa de nuestras vidas, queremos expresar un profundo agradecimiento a quienes con su ayuda, apoyo y comprensión nos alentaron a lograr esta hermosa realidad, ellos son:

A mi pequeña familia, tía Tamara por darme siempre mucho cariño y ser su sobri, tío Ché y Neri que siempre preguntaban por mí, mis primos Arian, Baby y el más pequeñito Albertico. A Iván ¿pensaste que te dejaría fuera?, a ti mil gracias por todos los consejos que siempre me distes y me sigues dando, como si fueras mi padre y acompañar a mi mamá cuando yo no estaba. A tía chiquitica por ser tan atenta conmigo y demás familiares.

A un grupo de personas que siempre me brindaron su ayuda cuando la necesité, Anabel por ser como una madre, Jorgito, Coqui, José Luis por estar siempre ayudándome en cada viaje que hacía a la casa, Maribel por su atención en cada momento de apuro, Idalmis y Pedro por estar siempre preocupándose por mi, Niurka, Ana Melba, Mabel; a todos por esto y más... Gracias.

A todos los profes que tuve durante estos cinco años de la carrera, de los cuales siempre aprendí algo útil para la vida, muy en especial a la profe Maily, por su orientación, sus consejos, paciencia y dedicación, a Agustín por escucharme y aclararme todas las dudas referentes al desarrollo de esta investigación, a la profe y tutora Liana por sus sugerencias, a la profe Yisell por dedicar parte de su tiempo a ayudarme con la ortografía.

A todos mis compañeros de grupo, en especial a Yadira, por compartir conmigo muchos momentos, tanto alegres como tristes y ser una gran amiga. A todos los del apartamento, Pedro, Morilla, Pipiolo, Isabel, Yunesti y Damir.

A mis compañeros del laboratorio por hacer que cada pedazo de tiempo fuera ameno y por toda su colaboración, Ana María, Danay, Jorge Martínez, Osmany, Ramiro, Jacobo, Héctor, Lavin y Eriane; en especial a mi compañero de tesis Yasniel por su preocupación y comprensión.

A todas mis tías de Gastex, por todos los momentos alegres que pasamos juntas, a Rosa por ser tan especial, a la gorda de Magela por hacerme reír tanto, a Elenita porque a pesar de ser poco el tiempo que nos conocemos ha demostrado ser una gran persona y sobre todo una gran amiga.

Daylén

A mi mamá Rosilda y a mi hermanito Yusniel por ser parte indispensable de mi vida.

A mi papá Juan José por sus buenos consejos.

A mi tía Niña y Nungo, a mi tío Andrés y mis primos: Tatín, Adrián, Ary y Andy que nunca me defraudaron y están siempre ahí para lo que necesite.

A toda mi familia, pues todos han mostrado interés y preocupación por mí.

A mis tutores Liana y Fidel por asesorarme y dar esa gran ayuda incondicional.

A mi compañera de tesis Daylén por haber tenido tanta paciencia conmigo.

A mis compañeros de aula, proyecto y todos los que me ayudaron de una u otra forma.

Gracias.

Yasniel

A la Revolución, por habernos dado la posibilidad de estudiar y obtener los conocimientos necesarios para la realización de esta investigación.



RESUMEN

En el presente trabajo se describe el proceso de desarrollo del módulo “Resultados”, perteneciente a los nuevos productos que conforman la Colección Multisaber, en su versión multiplataforma. Esta última está compuesta por varios software, que tienen la finalidad de servir de apoyo a los profesores de las escuelas primarias venezolanas en la impartición de las diferentes materias. En particular, el módulo “Resultados”, permite conocer todo cuanto ha hecho cada estudiante durante su interacción con los restantes módulos de un producto en específico.

A lo largo de este trabajo se expone en detalles cómo fue construido este módulo, cuáles son las funcionalidades que brinda y qué importancia posee el mismo, tanto para el profesor como para los alumnos.

PALABRAS CLAVES:

Colección Multisaber, módulo Resultados, versión multiplataforma.



Tabla de contenidos

INTRODUCCIÓN	1
Fundamentación Teórica.....	6
1.2 Sistemas similares existentes vinculados al campo de acción	6
1.3 Tendencias y tecnologías actuales	7
1.3.1 Lenguaje de programación.....	7
1.3.2 Servidor Web Apache.....	9
1.3.3 Content Management System (CMS).....	10
1.3.4 Integrated Development Environment (IDE)	13
1.3.5 Sistema Gestor de Base de Datos	14
1.3.6 Framework	15
1.3.7 Proceso de desarrollo de software	17
1.3.8 Metodologías de desarrollo de software	18
1.3.9 Herramienta Case de Modelado con UML.....	21
1.4 Conclusiones	24
Características del Sistema.....	26
2.1 Introducción	26
2.2 Especificación del contenido.....	26
2.3 Descripción del Modelo de Dominio.....	27
2.4 Solución propuesta.....	29
2.5 Especificación de requerimientos	30
2.5.1 Requisitos Funcionales del Sistema	30
2.5.2 Requisitos no Funcionales del Sistema	31
2.6 Modelo de Casos de Uso del Sistema	32
2.6.1 Actores del sistema	32
2.6.2 Diagrama de Casos de Uso del Sistema	33
2.6.3 Descripción de los Casos de Uso del Sistema.....	33
2.7 Conclusiones	55
Análisis y Diseño del Sistema	57
3.1 Introducción	57
3.2 Modelo de Análisis	57
3.2.1 Diagramas de clases del análisis.....	57
3.3 Modelo de Diseño	60
3.3.1 Diagramas de clases del diseño.....	61
3.4 Patrones	65



3.4.1 Patrones de diseño.....	65
3.5 Diseño de la base de datos.....	68
3.5.1 Diagramas de clases persistentes	68
3.5.2 Modelo de Datos.....	69
3.6 Conclusiones.....	70
Implementación	71
4.1 Introducción	71
4.2 Modelo de Implementación.....	71
4.2.1 Diagrama de despliegue.....	71
4.2.2 Diagrama de componentes	72
4.3 Conclusiones.....	73
Conclusiones	74
Recomendaciones	75
Referencias Bibliográficas	76
Bibliografía	78
Glosario de Términos y Siglas.....	79
Anexos.....	83



INTRODUCCIÓN

En la actualidad, el empleo de las nuevas Tecnologías de la Información y las Comunicaciones (TICs en lo adelante), ha devenido en un paso de avance significativo en todo el quehacer de una sociedad. Una de las esferas donde ha tenido un mayor impacto, es sin dudas, la educación, a partir del uso de diferentes medios que sirven de apoyo a todo el proceso de enseñanza-aprendizaje, trayendo consigo un aumento considerable de la calidad con la cual se desarrollan cada una de estas actividades.

A partir del año 2000 en Cuba, se pusieron en práctica un conjunto de transformaciones en los centros de enseñanza primaria, secundaria y preuniversitaria, con vista a elevar la calidad del proceso que, en materia de educación y formación, se venía desarrollando hasta el momento. Una de las que más ha incidido de manera positiva en el aprendizaje del estudiantado, ha sido sin dudas, el incremento del número de computadoras en todas las escuelas del país, lo cual ha posibilitado la diversificación de la enseñanza, a partir de la utilización de los diferentes software y materiales didácticos, que hacen posible que los estudiantes se apropien de los conocimientos necesarios de una manera más amena. Por tal razón, el Ministerio de Educación se dio a la tarea de reunir varios profesores que tuviesen motivaciones por la informática, con el objetivo de desarrollar productos con carácter educativo, para ser instalados en las computadoras de los diferentes centros educacionales del país. Fue así como surgió la “Colección Multisaber” para la enseñanza primaria, la “Colección El Navegante” para la enseñanza secundaria y la “Colección Futuro” para la enseñanza preuniversitaria. Estas colecciones de software están constituidas por un conjunto de productos con tecnología multimedia, para la enseñanza de las diferentes materias, tienen la finalidad de motivar a los estudiantes y con ello facilitar la asimilación de los diferentes contenidos, a partir de una mayor interacción hombre-máquina.

La Colección Multisaber en particular está compuesta por varios productos de similar modelo pedagógico, pero con contenidos diferentes, es decir, cada uno de los productos que forman parte de dicha colección se especializa en la enseñanza de una materia determinada, aunque todos poseen la misma estructura, que se divide a su vez, en varios módulos, donde cada uno de estos brinda la posibilidad de realizar un conjunto de acciones diferentes. Por ejemplo el módulo “Temas”, donde el alumno puede estudiar el contenido referente a un tema específico. También se encuentra el módulo “Ejercicios”, que le ofrece la posibilidad al estudiante de realizar una ejercitación variada del conocimiento adquirido. El módulo “Juegos”, que tiene la finalidad de crear en el alumno, un grupo de habilidades en el manejo de los diferentes dispositivos de la computadora, además de propiciar que los mismos adquieran el conocimiento de una forma más atractiva. El módulo “Mediateca” que recoge un conjunto de imágenes, videos, palabras y textos



relativos a los diferentes contenidos con los que cuenta el producto multimedia. Posee también el módulo “Maestro”, que es para uso solamente del profesor, y tiene la finalidad de brindarle al mismo la posibilidad de configurar varias opciones relativas al producto en cuestión. Por último se encuentra el módulo “Resultados”, el cual tiene como objetivo brindarle al profesor un informe pormenorizado sobre el desempeño de cada uno de sus estudiantes durante su interacción con los restantes módulos de la aplicación, así como, que cada estudiante a su vez, pueda visualizar la trayectoria que ha tenido hasta el momento.

A partir de la llegada de Chávez a la presidencia de Venezuela y del fortalecimiento de las relaciones entre ambos países, es que surge la idea, por medio de un convenio de colaboración, de facilitarle a dicho país la Colección Multisaber, con el objetivo de ser instalada en todas las escuelas primarias venezolanas. Por tal motivo, surge la necesidad de rehacer en su totalidad la colección completa, debido a 2 razones fundamentales: la Colección existente en estos momentos en las escuelas primarias cubanas fue desarrollada utilizando software propietario, es decir, software que para poder hacer uso del mismo es necesario primero, pagar cuantiosas sumas de dinero por la adquisición de su licencia, lo cual constituye una gran limitante a la hora de desplegarla en los centros educacionales de Venezuela, debido a que, como política de ese país, a partir del 23 de diciembre del año 2004 fue promulgado el Decreto 3390, el cual establece el uso prioritario del Software Libre en los centros de la administración pública nacional. La segunda razón es la necesidad de adecuar todos los contenidos y materiales con los que cuenta dicha colección, al contexto venezolano, es decir, a la cultura e idiosincrasia de ese país, debido a que, la que se encuentra en explotación en las escuelas primarias cubanas había sido construida utilizando rasgos típicos de la cultura de este pueblo, que iban desde sus atributos nacionales, hasta el lenguaje, la historia, las costumbres, entre otras.

De los 6 módulos que componen cada uno de los productos de la Colección Multisaber, uno que sin dudas reviste una gran importancia para el profesor, es el de “Resultados”, debido a que le brinda una serie de datos e informaciones referentes a cómo se ha comportado el proceso de aprendizaje de sus alumnos, a partir de la interacción de estos con cada uno de los restantes módulos con los que cuenta el producto. Lo que se persigue con el módulo “Resultados” es que el profesor pueda contar con una “Traza”, como también se le suele llamar, del itinerario de sus estudiantes durante el paso de estos por cada una de las secciones que conforman los restantes módulos. No obstante, durante el proceso de reconstrucción o migración de los productos pertenecientes a la Colección Multisaber, se pudo apreciar que el módulo “Resultados” que estos poseen, no satisface totalmente las necesidades de los profesores, es decir, la información que brinda en algunos productos es insuficiente, ya sea porque la misma está incompleta, o porque



no se encuentra debidamente organizada, lo cual trae notables dificultades al profesor a la hora de poder medir el desempeño de sus estudiantes durante la interacción de estos con los restantes módulos de la aplicación, y por ende, el grado de avance que van teniendo los mismos, con relación al aprendizaje de una materia específica.

Lo planteado anteriormente, unido a la necesidad de migrar cada uno de los productos que conforman la Colección Multisaber a Software Libre, conlleva a enunciar el siguiente **problema científico**: ¿Cómo lograr que la información que se muestra en el módulo “Resultados” de los nuevos productos que conforman la Colección Multisaber en su versión multiplataforma, esté debidamente organizada y con la calidad requerida?

Por tal razón, el **objetivo general** de esta investigación es: Desarrollar el módulo “Resultados” de los nuevos productos que conforman la Colección Multisaber, en su versión multiplataforma.

De lo expresado anteriormente, se deriva el **objeto de estudio**, el cual está determinado por el proceso de desarrollo de los nuevos productos que conforman la Colección Multisaber, en su versión multiplataforma.

Como **objetivos específicos** se tienen los siguientes:

1. Realizar el estudio de las principales tendencias y tecnologías actuales posibles a utilizar, para darle cumplimiento al problema planteado.
2. Realizar el Análisis, Diseño e Implementación del módulo “Resultados” de los nuevos productos pertenecientes a la Colección Multisaber, en su versión multiplataforma.
3. Conformar la documentación en la que se recoja todo el proceso de desarrollo de este módulo, para que sirva de apoyo al desarrollo de posteriores versiones.

El **campo de acción** está enmarcado en el proceso de desarrollo del módulo “Resultados” perteneciente a los nuevos productos que conforman la Colección Multisaber, en su versión multiplataforma.

Se plantea como **idea a defender**, que si se realiza el Análisis, Diseño e Implementación del módulo “Resultados”, perteneciente a los nuevos productos que conforman la Colección Multisaber en su versión multiplataforma, se podrá contar con un medio que le facilite al profesor llevar un control detallado y debidamente organizado sobre cada una de las acciones realizadas por sus estudiantes, durante la interacción de los mismos con los restantes módulos de la aplicación, propiciando con ello que el profesor pueda conocer el grado de avance que van



adquiriendo sus alumnos durante el aprendizaje de una materia específica. Además de permitirle a los estudiantes visualizar el resultado de todas y cada una de las actividades que han realizado en cada producto de la colección.

A continuación, se mencionan los **métodos teóricos** utilizados:

1. **Método de Análisis Histórico – Lógico:** Para comprender la historia del desarrollo de las investigaciones realizadas anteriormente, además de su esencia y necesidad. Es decir, para conocer la trayectoria de la Colección Multisaber realizada para las escuelas primarias cubanas y entender así, los detalles más importantes existentes para el desarrollo posterior de los nuevos productos relacionados con dicha colección en su versión multiplataforma.
2. **Método de Modelación:** Para modelar el sistema en cuestión y realizar todos los modelos correspondientes a cada uno de los flujos de trabajo. Debido a que este método justamente crea abstracciones con el objetivo de explicar la realidad, o sea, que permite comprender mejor el fenómeno u objeto de estudio, puesto que para obtener el conocimiento es necesario observar lo real, construir modelos y analizar la estructura.

(...) “El método debe ser visto como la estrategia concreta integral de la investigación, ya que no existe un método único y eterno para crear ciencia, partiendo del hecho de que cada investigación científica es un acto original y creativo y que todo investigador verdadero, pueda crear su propio método” (...).[1]

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

1. Realizar el diseño teórico de la investigación.
2. Fundamentar las herramientas y tecnologías que serán usadas durante el desarrollo del módulo “Resultados”.
3. Realizar un estudio del Documento “Acuerdo #30 sobre el módulo Resultados”.
4. Determinar los requisitos funcionales y no funcionales del módulo “Resultados”.
5. Implementar la capa de Acceso a Datos.
6. Diseñar la interfaz de usuario correspondiente al módulo “Resultados”.
7. Conformar los artefactos correspondientes al flujo de trabajo de “Análisis y Diseño”.



8. Implementar cada una de las funcionalidades que contendrá el módulo “Resultados”.
9. Conformar los artefactos correspondientes al flujo de trabajo de “Implementación”.
10. Elaborar un informe de la investigación.

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

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

Capítulo 2. Características del Sistema: En dicho capítulo se realiza una descripción del objeto de estudio, además de que se definen cuáles son los requerimientos funcionales y no funcionales, a la vez que se presentan los actores del sistema y diagrama de casos de uso del mismo, acompañado de la descripción textual de cada uno de estos.

Capítulo 3. Análisis y Diseño del Sistema: En este capítulo se presentan los diagramas de clases del análisis, además de los diagramas de clases Web, que reflejan de una forma más clara cómo va a funcionar dicho sistema y qué clases están presentes en el mismo, acompañado además del diagrama de clases persistentes y el modelo de datos.

Capítulo 4. Implementación: En este capítulo se describe cómo será implementado el sistema, a través de los diagramas de componentes y el diagrama de despliegue.



Capítulo

1

Fundamentación Teórica

1.1 Introducción

El siguiente capítulo aborda la fundamentación teórica del módulo “Resultados” de la Colección Multisaber, en su versión multiplataforma. En el mismo se realiza un análisis de los sistemas similares existentes vinculados al campo de acción, además de que se fundamentan las herramientas y tecnologías que serán utilizadas durante el desarrollo de la solución que se propone.

1.2 Sistemas similares existentes vinculados al campo de acción

En la actualidad se encuentra instalada en las escuelas primarias de Cuba la antigua versión de la Colección Multisaber, la cual se desarrolló con herramientas propietarias.

En particular el módulo “Resultados”, que se encuentra en los productos de estas colecciones, se caracteriza por llevar un registro exhaustivo de cada sesión de trabajo de los estudiantes, plasmándose todos los detalles de la interacción de los mismos en cada uno de los software; es decir, se registraba la hora de entrada al producto, la hora de entrada a cada módulo, a cada tema, la consulta a cada recurso mediático o definición, los resultados de la evaluación de las preguntas, entre muchas otras cosas. Todo esto se realizaba de manera cronológica según iba sucediendo, lo que a simple vista parece ser acertado, pero en la práctica se demostró que generaba gran cantidad de información, la cual para poseer un sentido práctico primero debía ser decodificada por el usuario, después consolidada o resumida, para entonces obtener un criterio cualitativo o cuantitativo del desempeño del estudiante en una sesión de trabajo.

A continuación se enumeran las principales deficiencias encontradas al módulo “Resultados” de dicha Colección.

1. Presentación de excesiva cantidad de información, y en ocasiones redundante, que en lugar de aportar beneficios conducen a oscurecer la lectura e interpretación de los datos.
2. La traza sólo se organiza por sesiones de trabajo y no por estudiantes, por lo que hace difícil su acceso, así como un seguimiento de la evolución del desempeño del alumno.
3. No se ofrecen de forma directa resultados o datos para realizar una evaluación cualitativa o cuantitativa del desempeño.



4. Los usuarios en la autenticación deben escribir siempre su nombre y apellidos, lo que trae el inconveniente que un mismo usuario puede entrar al producto con nombres diferentes, impidiendo que el profesor pueda darle un seguimiento adecuado.
5. Dificultades en el mantenimiento de los productos, pues se crean gran cantidad de ficheros de traza que periódicamente deben ser eliminados manualmente, o de lo contrario pueden llegar a llenarse los discos duros.
6. Sólo se ofrece información textual y de forma secuencial, es decir, no existe un mecanismo que permita acceder directamente a las distintas secciones dentro del propio reporte de la traza.

1.3 Tendencias y tecnologías actuales

A continuación se describen las tendencias y tecnologías presentes en la actualidad, que serán utilizadas con vista a darle solución a los problemas planteados anteriormente de manera eficiente, teniendo en cuenta las necesidades existentes y el entorno donde se aplicará la solución propuesta.

1.3.1 Lenguaje de programación

Un lenguaje de programación, es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos de hardware y software existentes. [2]

Con otras palabras, es un lenguaje artificial que puede ser usado para controlar el comportamiento de una máquina, específicamente una computadora y que está compuesto por un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas.

PHP: Acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de "código abierto" interpretado de alto nivel, especialmente pensado para Aplicaciones Web, el cual puede ser embebido en páginas HTML. Su sintaxis es similar a C, Java y Perl y es fácil de aprender. [3]

¿Qué se puede hacer con PHP?

Con PHP se puede hacer, tanto cuanto sea posible realizar con un script Common Gateway Interface (en lo adelante CGI), como procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies, entre otras cosas. [3]



Existen tres campos en los que se usan scripts escritos en PHP.

- Scripts del lado del servidor. Este es el campo más tradicional y el principal foco de trabajo. Se necesitan tres cosas para que esto funcione: el intérprete PHP (CGI ó módulo), un servidor Web y un navegador. Es necesario ejecutar el servidor Web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectándose con el servidor Web.
- Scripts en la línea de comandos. Se puede crear un script PHP y ejecutarlo sin ningún servidor Web o navegador. Solamente se necesita el intérprete PHP para usarlo de esta manera. También pueden ser usados para tareas simples de procesamiento de texto.
- Escribir aplicaciones de interfaz gráfica. Probablemente PHP no sea el lenguaje más apropiado para escribir aplicaciones gráficas, no obstante, posee una extensión llamada "PHP-GTK" la cual incluye varios componentes gráficos implementados con este lenguaje accesibles al usuario. [3]

¿Cómo funciona PHP?

A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por tal razón permite acceder a los recursos que se encuentren en el mismo, como por ejemplo una base de datos. Cuando el cliente hace una petición al servidor, para que le envíe una página Web generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica y regresa el resultado al servidor, que a su vez se encarga de regresárselo al cliente.

A continuación, se muestran varias de las características principales que posee PHP, y que lo convierten en uno de los lenguajes para la Web más usados en la actualidad:

- Soporte para diversas bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader), Excel, XML, entre otros.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.



- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se actualiza continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. [4]

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, entre los que se encuentran: Linux, diversas variantes de Unix, Microsoft Windows y Mac OS X. Además, soporta la mayoría de servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape, O'Reilly Website Pro server, entre otros. Tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI. [3]

De manera general, las principales características que distinguen a PHP son: su rapidez; su facilidad de aprendizaje; su soporte multiplataforma tanto de diversos Sistemas Operativos, como de servidores HTTP y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta. A lo cual se puede agregar, que es un lenguaje orientado a objetos y que posee variedad de funciones ya implementadas y de mucha utilidad para los desarrolladores.

1.3.2 Servidor Web Apache

Apache es un servidor Web gratuito desarrollado por el Apache Server Project (Proyecto Servidor Apache), cuyo objetivo es la creación de un servidor Web fiable, eficiente y fácilmente extensible con código fuente abierto gratuito. Este proyecto es conjuntamente manejado por un grupo de voluntarios localizados alrededor del mundo, que a través de Internet planean y desarrollan el servidor y la documentación relacionada con éste. Estos voluntarios son conocidos como el grupo Apache. [5]

A menos de un año después de que el grupo Apache se constituyera, el servidor que lleva su nombre pasó a ser considerado como el servidor número uno de Internet. En la actualidad, según los datos publicados por Netcraft,[6] el servidor Web Apache es más usado que todos los demás servidores Web juntos.

Las principales ventajas que tiene la utilización de este servidor Web son:

- Su licencia: Esta es de código abierto del tipo BSD que permite el uso comercial y no comercial de Apache.



- Una talentosa comunidad de desarrolladores siguiendo un proceso abierto de desarrollo.
- Portabilidad: Apache trabaja sobre todas las versiones recientes de UNIX y Linux, Windows, entre otros.
- Es robusto y seguro.
- Arquitectura modular. [5]

Esta última característica se refiere a que Apache es un software que está estructurado en módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor Web. Esta modularidad es intencionada ya que la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del mismo. Los módulos del Apache se pueden clasificar en tres categorías: [6]

- Módulos Base: Módulos con las funciones básicas del Apache.
- Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender las peticiones.
- Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de las funcionalidades del servidor, se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software. [6]

1.3.3 Content Management System (CMS)

Un CMS o Sistema Gestor de Contenidos por sus siglas en inglés, es un software que permite organizar y facilitar la creación de documentos y otros contenidos de un modo cooperativo. Con frecuencia, un CMS es una aplicación Web usada para gestionar sitios y contenidos Web.

JOOMLA: Es un CMS de código abierto, liberado bajo la versión 2 de la licencia GPL, el cual se ha convertido rápidamente en uno de los más usados de su tipo, como lo prueban sus numerosos premios y la masiva comunidad en línea que posee. [7]



Varios de los aspectos que han hecho a JOOMLA tan popular son: el gran número de libertades y las extensiones disponibles en el mercado, que permiten a los usuarios hacer mucho más que administrar el contenido. La siguiente lista muestra un grupo de las funciones comunes que poseen dichas extensiones:

- Los anuncios de banner y Afiliados.
- Calendarios.
- Comunicación (Chats, Foros, Libros de Visitas, Listas de Correo, Boletines Informativos).
- Contenidos y Noticias (Blogs, Postales, Noticias).
- Documentación (Descargas, FAQs, Wikis).
- Comercio Electrónico (Anuncios, Carritos de compra).
- Formularios.
- Galería y Multimedia.
- Intranet.
- Búsqueda e indexación. [7]

Una vez que se ha instalado y configurado el CMS, resulta muy sencillo para usuarios sin conocimientos técnicos realizar las operaciones de añadir o editar algún contenido, así como, subir imágenes, documentos e información en general. Entre los beneficios que brinda este Sistema Gestor de Contenidos se encuentran los siguientes:

- **Organización del sitio Web:** Está preparado para organizar eficientemente los contenidos de su sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores. Desde el panel de administración de JOOMLA es posible crear, editar y borrar las secciones y categorías del sitio de la manera que más convenga.
- **Publicación de Contenidos:** Con este Sistema Gestor de Contenidos, se pueden crear páginas ilimitadas y editarlas desde un sencillo editor que permite formatear los textos con los estilos e imágenes deseados. Los contenidos son totalmente editables y modificables.

Escalabilidad e implementación de nuevas funcionalidades: Ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos, que agregarán servicios de valor a



los visitantes de su sitio Web, por ejemplo: galerías de imágenes, foros, boletines informativos, clasificados, entre otros.

- **Administración de usuarios:** Permite almacenar datos a los usuarios registrados y también la posibilidad de enviar E-mails masivos a todos los usuarios. La administración de usuarios es jerárquica, y los distintos grupos de usuarios poseen diferentes niveles de facultades/permisos dentro de la gestión y administración del sitio.
- **Diseño y aspecto estético del sitio:** Es posible cambiar todo el aspecto del sitio Web tan solo con un par de clic, gracias al sistema de plantillas que utiliza.
- **Navegación y menú:** Totalmente editables desde el panel de administración del mismo.
- **Administrador de Imágenes:** Utilidad que posee para subir imágenes al servidor y usarlas en todo el sitio.
- **Disposición de módulos modificable:** En un sitio creado con este CMS, la posición de los módulos puede acomodarse como se prefiera.
- **Encuestas:** Posee un sistema de votaciones y encuestas dinámicas con resultados en barras porcentuales.
- **Feed de Noticias:** Trae incorporado un sistema de sindicación de noticias por RSS/XMS de generación automática.
- **Publicidad:** Es posible hacer publicidad en el sitio usando el Administrador de Banners.
- **Estadísticas de visitas:** Con información del navegador, Sistema Operativo, y detalles de los documentos (páginas) más vistos. [8]

Otro aspecto a destacar de este CMS, es la manera en que publica cualquier tipo de información, a continuación se muestran varias características de este proceso:

- **Automatización en la publicación:** Las páginas y documentos que posee pueden programarse con fecha de publicación y fecha de caducidad. Es decir, un documento puede programarse para que se publique automáticamente al llegar una determinada fecha, y luego no publicarse de forma automática en otra fecha.
- **Archivo e historial:** Las páginas viejas o publicaciones que hayan perdido vigencia pueden enviarse a un "archivo" de almacenamiento, sin necesidad de tener que borrarlas. Esto permite también dar la posibilidad a los navegantes de consultar artículos viejos o documentos anteriores en un historial.



- **Formatos de lectura:** Cada documento es generado automáticamente por JOOMLA en formato PDF, en versión imprimible, y en XML.
- **Envío por E-mail:** Los usuarios del sitio podrán enviar automáticamente a un amigo por email cada documento publicado.
- **Valoración de contenidos:** Los visitantes del sitio podrán votar la calidad de lo publicado.
- **Comentarios:** Los usuarios podrán comentar sus opiniones o expresar sus inquietudes en la misma página de contenidos. [8]

1.3.4 Integrated Development Environment (IDE)

Un Entorno de Desarrollo Integrado o IDE (por sus siglas en inglés), es una herramienta que permite a los desarrolladores de software escribir sus programas en uno o más lenguajes. Consiste básicamente en una plataforma en la que se integran un editor de código, un compilador, un depurador y una interfaz gráfica de usuario.

NetBeans: Es una herramienta de código abierto diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java. NetBeans IDE dispone de soporte para crear interfaces gráficas de forma visual, así como, desarrollo de aplicaciones Web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis y por si fuera poco, sus funcionalidades son ampliables mediante la instalación de plugins. [9]

NetBeans IDE constituye la solución más completa para programar en Java, aunque soporta otros lenguajes como: C/C++, JavaScript, Ruby, Groovy, Python, y PHP. [10]

Esta poderosa herramienta está disponible para diferentes plataformas, como es el caso de: Windows XP/Vista, Linux, Mac OS, además de Solaris y OpenSolaris.

En particular, la versión 6.7 de NetBeans brinda la posibilidad de estar conectado al “Project Kenai”, entorno de colaboración que le permite a los desarrolladores hospedar su proyecto “open source” en Internet, y permitir que esté disponible para todo el mundo.

Gracias a esta integración, el equipo de desarrollo puede usar la misma interfaz para crear proyectos, así como subirlos, editarlos, depurarlos, construirlos, entre otras cosas. [10]

Si bien NetBeans fue creado originalmente por Sun Microsystems como IDE exclusivamente para Java, este brinda también la posibilidad de desarrollar en PHP desde la versión 6.5. En la



versión 6.7 se agregaron nuevos dispositivos de soporte a este lenguaje, como es el caso de “PHPUnit”, mediante el cual los desarrolladores pueden probar recursos PHP con facilidad. [11]

Otra característica importante de NetBeans 6.7 es que posee un buen completamiento de código para cada uno de los lenguajes de programación que soporta, como es el caso de PHP, para el cual proporciona incluso, completamiento de código para los métodos mágicos Get y Set. En el caso de JavaScript, brinda soporte y autocompletado de código para sus principales librerías, entre las que se destacan: jquery, dojo y prototype.

1.3.5 Sistema Gestor de Base de Datos

Un Sistema Gestor o Manejador de Bases de Datos (en lo adelante SGBD), es un conjunto de programas que permite a los usuarios crear y mantener una base de datos, por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular una base de datos para diversas aplicaciones.

MySQL: Es el SGBD de mayor popularidad en el mundo de las bases de datos de código abierto, debido a su gran rendimiento consistente, y de alta fiabilidad, además de su fácil uso. Se utiliza en todos los continentes, incluyendo la Antártida. La mayoría de las empresas y organizaciones lo utilizan para ahorrar tiempo y dinero, impulsando sus sitios Web de alto volumen, sistemas de negocio y paquetes de software, entre los que se incluyen líderes de la industria, como es el caso de: Yahoo, Google, Nokia, YouTube, Wikipedia, entre otros. Además de su gran popularidad mundial, MySQL se ha convertido en la base de datos de elección para una nueva generación de aplicaciones basadas en lo que se conoce como “LAMP” (Linux, Apache, MySQL, PHP/Perl/Python). [12]

Ventajas

- La velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Su bajo consumo lo hace apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.



- El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro y de buscadores de aplicaciones.
- Se ejecuta en más de 20 plataformas, entre las que se encuentran: Linux, Windows, Mac OS, Solaris, y muchas más. [13]

EMS SQL Manager for MySQL Professional for Linux

EMS SQL Manager for MySQL es una poderosa herramienta para la administración del servidor de Bases de Datos MySQL. Funciona con cualquier versión de MySQL superior a 3.23 y soporta todas las características y funcionalidades de MySQL, incluyendo las vistas, procedimientos almacenados, funciones definidas por el usuario, tablas transaccionales, los nuevos 5 protocolos de autenticación de MySQL, extensiones de privilegios de usuario, juegos de caracteres y colaciones de apoyo para las bases de datos y tablas. Ofrece una gran variedad de herramientas poderosas para usuarios avanzados, tales como: Visual Database Designer (Diseñador Visual de base de datos), Visual Query Builder (Constructor Visual de Consultas), y el espectador de BLOB, editor para satisfacer todas las necesidades. Tiene un nuevo estado de la interfaz gráfica de usuario (GUI, por sus siglas en inglés), la cual permite crear y editar los objetos de bases de datos de una manera simple y directa, además de ejecutar scripts SQL, administrar los privilegios de usuario, editar consultas SQL, extraer e imprimir metadatos, creación de informes de la estructura de base de datos en formato HTML (HyperText Markup Language, por sus siglas en inglés), la exportación e importación de datos, así como los suministros y otros tantos servicios que harán que el trabajo con el servidor MySQL sea fácil en gran medida. [14]

1.3.6 Framework

Es necesario entender que un framework es una abstracción de código común que provee funcionalidades genéricas que pueden ser utilizadas para desarrollar aplicaciones de manera rápida, fácil, modular y sencilla, ahorrando tiempo y esfuerzo. Entonces, un framework es concreto y también “incompleto”. Concreto porque es, desde un punto de vista simple, un conjunto de componentes; incompleto, porque por sí mismo no puede ser utilizado, ya que guía a la solución de problemas de programación recurrentes, pero por lo general, no constituye la solución específica completa. [15]

En su mayoría, los frameworks JavaScript proveen componentes para:

- **Compatibilidad:** Agregan la posibilidad de escribir código JavaScript totalmente compatible con todos los navegadores y motores JavaScript más utilizados. Esto aumenta



la portabilidad y elimina el “gran dolor de cabeza” de incompatibilidad entre navegadores y sus motores intérpretes JavaScript.

- **Comunicación asíncrona (AJAX):** Usando este acercamiento, es fácil utilizar XMLHttpRequest para manejar y manipular los datos en los elementos de un sitio, aumentando la interactividad y experiencia del usuario.
- **Modelo de Objetos del Documento (DOM).** Maximizan la capacidad de agregar, editar, cambiar y eliminar elementos de manera dinámica agregando librerías que facilitan usarlo.
- **Validación de Formularios.** Permiten de una manera relativamente fácil validar campos dentro de uno o varios formularios. Esto, desde el punto de vista del desarrollador, simplifica y reduce el código para procesar dichos formularios, ya que los datos llegan previamente validados, reduciendo los errores de tipos de datos.
- **Efectos visuales.** Utilizando la manipulación de los elementos, se pueden crear efectos visuales y animaciones. Entre los efectos se encuentran: aparecer y desaparecer, redimensionamiento, mover y más.
- **Almacenamiento del lado del cliente.** En adición provee funciones para leer y escribir cookies. También provee una abstracción de almacenamiento que permite a las aplicaciones Web guardar datos del lado del cliente, persistente y de manera segura.
- **Manejo JSON.** Incrementa al máximo el manejo de datos, que pueden ser utilizados para presentar informaciones de manera dinámica y en tiempo de ejecución.
- **Manejo de Eventos.** Esta característica agregada permite reaccionar de una manera u otra dependiendo de las acciones del usuario.
- **Recibidores de Datos.** Permiten utilizar diferentes formatos de datos como XML, HTML, Texto, JSON, ATOM, entre otros.
- **“Arrastra y Suelta”.** Mejor conocido como Drag and Drop. Es una funcionalidad que brinda la posibilidad de arrastrar elementos dentro de una misma página que interactúe con el resto de los elementos. [15]

Dojo Toolkit

Es un Framework compuesto por Widgets, que son componentes de código en JavaScript pre-empaquetados que pueden ser utilizados para enriquecer sitios Web con varias características, que trabajan en la mayoría de los navegadores, tales como: menú, tablas ordenables y más. [15]



Dojo contiene un sistema de empaquetado inteligente para los efectos de interfaz de usuario, arrastrar y soltar APIs, widget APIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX.

Dojo resuelve asuntos de usabilidad comunes como pueden ser: la navegación y detección del explorador y soportar cambios de direcciones en la barra de URL para luego regresar a ellas. Proporciona una gama más amplia de opciones en una sola biblioteca y hace un trabajo eficiente que apoya los nuevos y viejos navegadores. [16]

De manera general, Dojo incluye los siguientes aspectos:

- Un diseño e implementación estándar para los diferentes navegadores, permitiendo utilizar el mismo código fuente independiente del navegador o browser que se utilice.
- Biblioteca de funciones, que resume lo que en ocasiones resulta algo abstracto cuando se trabaja con el Modelo de Objetos del Documento.
- Un sistema de módulos junto con un sistema de construcción, que le permite dividir código en pequeños pedazos manejables.
- Bibliotecas independientes (o sea, que se cargan en dependencia de las necesidades) que implementan otras funcionalidades avanzadas. [17]

1.3.7 Proceso de desarrollo de software

A través de los años, la producción de software ha ido progresando en cuanto a cantidad, calidad y complejidad, debido a que cada vez son mayores las exigencias de los usuarios, por lo que se hace mayor la necesidad de un proceso que integre todas las fases de desarrollo de software y que sea perfectamente entendible por todos aquellos que participan en su elaboración.

Lenguaje Unificado de Modelado (UML)

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el mismo es un lenguaje, cuenta con reglas para combinar tales elementos.

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como “modelo”. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo. [18]



Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- **Visualizar:** UML permite expresar de una forma gráfica un sistema de manera que otro lo puede entender.
- **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado para su futura revisión. [19]

Aunque UML está pensado para modelar sistemas complejos, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- **Elementos:** Los elementos son abstracciones de entes reales o ficticios (objetos y sus acciones).
- **Relaciones:** Relacionan los elementos entre sí.
- **Diagramas:** Son colecciones de elementos con sus relaciones. [19]

Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas, entre los que se encuentran: diagrama de casos de uso, diagrama de clases, diagrama de objetos, diagrama de secuencia, diagrama de colaboración, diagrama de estados, diagrama de actividades, diagrama de componentes y diagrama de despliegue. [19]

1.3.8 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y pasos a seguir para construir un software. Por otra parte, una metodología de desarrollo de software define quién debe hacer qué, cuándo y cómo para alcanzar un determinado objetivo. También se puede decir que una metodología es un proceso, y en su modelación se definen como elementos principales los siguientes:



- **Trabajadores (quién):** Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado 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):** Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- **Flujo de actividades (cuándo):** Secuencia de actividades realizadas por trabajadores y que producen un resultado de valor observable.

Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés (Rational Unified Process), divide en 4 fases el desarrollo del software (*Ver Anexo 1*):

La siguiente tabla muestra sus fases y objetivos:

Fase	Objetivo
Inicio	Determinar la visión del proyecto.
Elaboración	Determinar la arquitectura.
Construcción	Llegar a alcanzar una funcionalidad operativa.
Transición	Llegar a obtener el entregable (release) del producto.

Tabla 1.1 Fases y objetivos de RUP

En cada una de estas fases tienen lugar iteraciones de varios flujos de trabajo (*Ver Anexo 1*), que son:

- **Modelo del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.



- **Análisis y Diseño:** Describe cómo el sistema será construido a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación o despliegue:** Produce entregable del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, entre otros.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un software que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, entre otros.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Características de RUP

1. **Dirigido por Casos de Uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí 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 (cómo se llevan a cabo).
2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben 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 comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas



UML (Vista de Casos de Uso, Vista Lógica, Vista de Procesos, Vista de Implementación, Vista de Despliegue).

3. **Iterativo e Incremental:** Aunque el *Anexo 1* puede sugerir que los flujos de trabajo se desarrollan en cascada, la “lectura” de este gráfico tiene que ser vertical y horizontal. RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos, aunque desarrolla fundamentalmente unos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

1.3.9 Herramienta Case de Modelado con UML

A medida que los sistemas que hoy se construyen se tornan más complejos, las herramientas de modelado con UML ofrecen muchos beneficios para todos los involucrados en un proyecto, por ejemplo, administrador del proyecto, analistas, arquitectos, desarrolladores y otros. Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) de modelado con UML permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraerse del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar. Cuanto más grande es un proyecto, es más importante utilizar una herramienta CASE. Por otro lado, al usar las herramientas CASE:

- Los analistas de negocio/sistemas pueden capturar los requisitos del negocio/sistema con un modelo de casos de uso.
- Los diseñadores/arquitectos pueden producir el modelo de diseño para articular la interacción entre los objetos o los subsistemas de la misma o de diferentes capas (los diagramas UML típicos que se crean son los de clases y los de interacción).
- Los desarrolladores pueden transformar rápidamente los modelos en una aplicación funcionando, buscar un subconjunto de clases y métodos y asimilar el entendimiento de cómo lograr interfaces con ellos. [6]

La estructura de las herramientas CASE se basan en las siguientes terminologías:

- **CASE de Alto Nivel:** Son aquellas herramientas que automatizan o apoyan las fases iniciales del ciclo de vida del desarrollo de sistemas, como la planificación, el análisis y el diseño.



- **CASE de Bajo Nivel:** Son aquellas herramientas que automatizan o apoyan las fases finales o inferiores del ciclo de vida, como es el caso del diseño detallado, la implantación y el soporte de sistemas.
- **CASE Cruzado de Ciclo de Vida:** Se aplica a aquellas herramientas que apoyan actividades que tienen lugar a lo largo de todo el ciclo de vida, se incluyen actividades como la gestión de proyectos y la estimación. [20]

Visual Paradigm for UML

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software de modelado, ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta CASE proporciona abundantes tutoriales de UML, demostraciones interactivas y proyectos UML. [20]

A continuación se mencionan varias de las características más notables que posee:

- Soporta las últimas versiones UML y la notación y modelado de procesos de negocios desde un grupo administrador de objetos.
- En adición al soporte de Modelado UML, esta herramienta provee el modelado de procesos de negocios, además de un generador de mapeo de objetos relacionales para los lenguajes de programación Java .NET y PHP.
- Para desarrolladores independientes existe una versión llamada Community Edition, la cual se caracteriza por ser de uso no comercial.
- Se integra con las siguientes herramientas de Java: Eclipse, JBuilder, NetBeans IDE, Oracle JDeveloper.
- Proporciona el código y la compatibilidad con alrededor de 10 lenguajes, tanto en la generación de código como en la ingeniería inversa, entre los cuales se pueden mencionar: Java, C++, C#, VisualBasic.Net, ActionScript, Delphi, Perl, Ruby, PHP, XML Schema, Ada y Python. [21]

Visual Paradigm for UML, se ha actualizado rápidamente en sintonía con el nuevo desarrollo de técnicas de modelado UML 2.1, con el propósito de generar un entorno de modelados visuales



en el que se reúnen hoy todas las necesidades, tanto de software y tecnología, como las de comunicación.

Esta versión de Visual Paradigm for UML emplea una rápida respuesta con poca memoria, utilizando moderadamente los tiempos del procesador, lo que le permite manejar grandes y complicadas estructuras de un proyecto de una forma muy eficiente y, que solo requiera de una configuración de escritorio. [21]

En esencia, Visual Paradigm for UML brinda la posibilidad de realizar las siguientes acciones:

- Modelado de procesos de Negocio.
- Modelado de Requisitos.
- Mapeo de objetos y relaciones.
- Modelado de Bases de Datos.
- Modelado UML.
- Generación de Documentación.
- Código de Ingeniería.
- Interoperabilidad.
- Colaboración en Equipo.
- Integración con diferentes IDE. [21]

1.3.10 Arquitectura

Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información.

Se selecciona y diseña con base en unos objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como: la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Algunas arquitecturas son más recomendables de implementar con ciertas tecnologías, mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura software de tres capas para implementar sistemas en tiempo real. [6]



La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura de software debe ser programable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea de computación.

La arquitectura de software tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad. [6]

Arquitectura en 3 Capas

La estrategia tradicional de utilizar aplicaciones compactas, causa gran cantidad de problemas de integración en sistemas de software complejos, como pueden ser: los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad e integración. [22]

Para resolver estos problemas se ha generalizado la división de las aplicaciones en capas. En el caso de que esta división se realice en tres capas: una de estas servirá para guardar los datos (base de datos), otra para centralizar la lógica de negocio (modelo) y por último una tercera para la interfaz gráfica que facilite al usuario el uso del sistema (*Ver Anexo 2*). [22]

Si se establece una separación entre la capa de interfaz gráfica, replicada en cada uno de los entornos de usuario, y la capa modelo, que quedaría centralizada en un servidor de aplicaciones, se obtendrá, sin dudas, una potente arquitectura que proporcionará las ventajas siguientes:

- Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo.
- No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento. [22]

1.4 Conclusiones

El módulo “Resultados” de la anterior versión de la Colección Multisaber, no brinda al profesor la posibilidad de llevar un control sobre las diferentes actividades realizadas por sus alumnos,



durante la interacción de estos con los restantes módulos de un producto en específico, debido a que la información que se muestra en el mismo, carece de la organización requerida, impidiendo en ocasiones que el profesor pueda contar con los datos necesarios, para poder realizar una valoración del grado de avance que van teniendo sus alumnos, durante el aprendizaje de una materia determinada. Con vista a darle solución a estos problemas, se ha decidido reconstruir en su totalidad este módulo, para lo cual se emplearán las herramientas descritas con anterioridad, debido a que ya habían sido definidas previamente por el proyecto al cual tributa esta investigación, además de que la mayoría son de código abierto y que están dentro de las más usadas a nivel mundial, lo cual brinda una serie de beneficios a la hora de buscar cualquier información sobre estas, tanto en Internet, como en la propia Universidad.



Capítulo

2

Características del Sistema



2.1 Introducción

El presente capítulo se centra en el Proceso Unificado de Desarrollo. Se hace referencia al Modelo de Dominio y la identificación de los principales conceptos que lo integran. También se identifican cuáles son los requisitos funcionales y no funcionales del sistema, que darán solución a los problemas existentes; a la vez que se presentan cuáles son las funcionalidades que brinda este sistema y quiénes son los actores que interactúan con el mismo.

2.2 Especificación del contenido

El primer paso en el proceso de desarrollo de software, es precisamente, alcanzar cierto nivel de conocimiento sobre el problema que se desea modelar. El contenido del problema en cuestión se encuentra estructurado de la siguiente manera:

En la primera pantalla de comunicación con el usuario, perteneciente al módulo “Resultados”, se muestra el listado de estudiantes (escuela, grado y sesión) seleccionado por el mismo; la cual, luego de haber elegido un estudiante determinado, permite acceder a la interfaz de comunicación: “Visualización del Historial del estudiante”, que tiene como objetivo mostrar un resumen de los resultados obtenidos por los alumnos en su interacción con el software o producto. Dicha interfaz va a contener un menú de categorías, las cuales se registran en el módulo. Estas se mencionan a continuación:

- **Identificador del estudiante:** Permite identificar al estudiante del que se desea visualizar los resultados, mostrando su nombre(s) y apellido(s).
- **Datos generales:** Muestra los datos generales del estudiante del que se visualizan los resultados: nombre(s) y apellidos(s), grado, grupo, escuela, fecha, hora de entrada al software y equipo de trabajo.
- **Itinerario:** Muestra un resumen del recorrido realizado por el estudiante en el software, es decir, la fecha y hora que entró al producto, además de la cantidad de veces que visitó cada módulo y el tiempo que permaneció en cada uno de ellos.



- **Temas:** Muestra los resultados de la interacción del estudiante con el módulo Temas, permitiendo visualizar cada uno de los epígrafes visitados y el tiempo que empleó en los mismos.
- **Ejercicios:** Muestra los resultados de la interacción del estudiante con el módulo Ejercicios, entre los que se encuentran: Ejercicios realizados, nivel de los mismos, calificación obtenida, tiempo que se demoró y efectividad.
- **Juegos:** Muestra los resultados de la interacción del estudiante con el módulo Juegos, brindando la posibilidad de conocer la cantidad de veces que interactuó con cada juego y el tiempo empleado en cada uno, además del lugar alcanzado y la efectividad obtenida.
- **Mediateca:** Muestra los resultados de la interacción del estudiante con el módulo Mediateca, agrupados en 4 epígrafes: Videos, Imágenes, Glosario e Información de Interés, en cada uno de los cuales se muestran los detalles de la información consultada.

2.3 Descripción del Modelo de Dominio

Debido a la poca estructuración y conocimiento de los procesos del negocio, se propone desarrollar un Modelo de Dominio, con el objetivo de ayudar a comprender los conceptos que utilizan los usuarios, que serán los mismos a utilizar en el desarrollo de la aplicación. Este modelo captura los tipos más importantes de objetos que existen, o los eventos que suceden en el entorno donde se encontrará el sistema. Los objetos, son los elementos básicos a utilizar para aclarar el dominio de un problema, algunos de ellos son: conceptos, atributos, asociaciones y roles. Una cualidad esencial que debe ofrecer un Modelo Conceptual o de Dominio es, representar un diagrama con los eventos reales relacionados con el sistema en cuestión; el cuál define las clases relevantes que intervienen en el contexto planteado.

A continuación se identifican los principales conceptos que se utilizarán en el diagrama, haciendo uso de un glosario de términos sobre los nombres:

Colección Multisaber: Colección de software educativo destinada a apoyar el proceso de enseñanza – aprendizaje de los niños en las escuelas primarias venezolanas.

Ejercicio: Módulo que permite ejercitar y comprobar las habilidades adquiridas por los estudiantes.



Escuela Primaria: Institución educativa centrada en la formación de menores entre 6 a 11 años de edad, que tiene como objetivo fundamental organizar la enseñanza y el aprendizaje de contenidos básicos.

Estudiante: Persona entre 6 y 11 años de edad que cursa estudios en las escuelas primarias.

Imagen: Representación gráfica de una expresión.

Juego: Módulo que contiene los juegos que el estudiante debe realizar para profundizar algún contenido indicado por el profesor.

Maestro: Módulo que le permite al profesor configurar algunas opciones del producto en cuestión.

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

Mediateca: Módulo que agrupa las medias presentes en el producto.

Módulo: Se le denomina al conjunto de elementos fundamentales que componen el producto.

Producto Multimedia: Producto que combina diversos tipos de medias.

Profesor: Persona responsable de educar, guiar y supervisar a los estudiantes en el proceso de enseñanza - aprendizaje.

Resultado: Es el módulo que visualiza la traza del estudiante durante la interacción de los mismos con los restantes módulos de la aplicación.

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

Tema: Es el módulo en el que se presentan los contenidos de las asignaturas o temas al cuál corresponde el software.

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

Video: Su objetivo radica en brindarle al usuario una información audiovisual, para que perciba imágenes y sonidos.



2.3.1 Diagrama de objetos del dominio

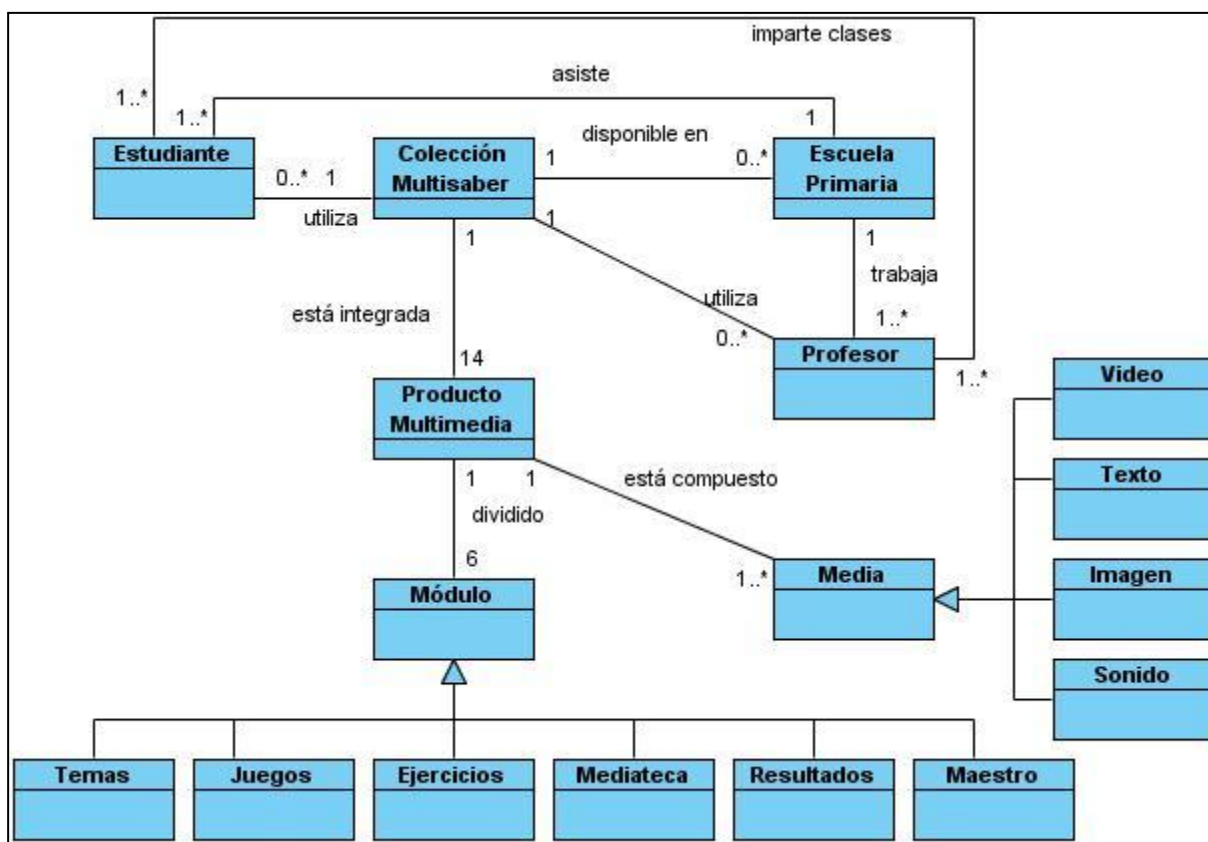


Figura 2.1 Diagrama de objetos del dominio

2.4 Solución propuesta

Con vista a darle cumplimiento a los problemas planteados con anterioridad, se propone reconstruir totalmente el módulo “Resultados” de los nuevos productos pertenecientes a la Colección Multisaber, en su versión multiplataforma, de manera que tanto el profesor como los alumnos, puedan contar con un medio que les permita visualizar el resultado de la interacción de estos últimos con los restantes módulos de un producto. La información que aquí se mostrará, podrá ser accesible en todo momento y constituirá una especie de historial que recogerá en detalles cada acción realizada por los estudiantes, como por ejemplo: la fecha y hora en que comenzó a interactuar con un producto, cuánto tiempo dedicó a realizar cada actividad, además de poder ver cada una de las sesiones que visitó.

En resumen, este nuevo módulo revestirá de gran importancia para los profesores, debido a que podrán contar con la información que necesitan de una forma más organizada y precisa, propiciando que los mismos puedan emitir una mejor valoración con relación al desempeño de sus alumnos.



2.5 Especificación de requerimientos

Un requisito es una descripción de necesidades o aspiraciones respecto a un producto. El objetivo principal de la actividad de definición de requisitos, consiste en identificar qué es lo que realmente se necesita. Esto se realiza de un modo que sirva de comunicación entre el cliente y el equipo de desarrollo.

Los requisitos se pueden clasificar en: funcionales y no funcionales. Los primeros son capacidades o condiciones que el sistema debe cumplir; y los segundos, son propiedades o cualidades que el producto debe tener.

2.5.1 Requisitos Funcionales del Sistema

RF 1- Registrar traza del módulo “Temas”.

RF 2- Registrar traza del módulo “Ejercicios”.

RF 3- Registrar traza del módulo “Juegos”.

3.1 Registrar traza de juegos con ejercicios.

3.2 Registrar traza de juegos sin ejercicios.

RF 4- Registrar traza del módulo “Mediateca”.

RF 5- Listar estudiantes.

5.1 Seleccionar los estudiantes por escuela, grado y sección.

5.2 Mostrar los siguientes datos de los estudiantes: nombre, apellidos, sesiones, última sesión.

5.3 Ordenar a los estudiantes por número de la lista, nombre (s), apellido (s), sesiones y fecha de la última sesión iniciada.

RF 6- Consultar historial de estudiantes.

RF 7- Gestionar estudiantes.

7.1 Adicionar estudiantes.

7.2 Eliminar estudiantes.

7.3 Modificar los datos de los estudiantes.



2.5.2 Requisitos no Funcionales del Sistema

Requerimientos de Software

RNF.1 Computadora personal con navegador Mozilla Firefox 3.x.

Requerimientos de Hardware

RNF.1 Procesador Pentium 233 MHz (recomendado 500 MHz o mayor).

RNF.2 64 MB de RAM (recomendado 128 MB de RAM o mayor).

RNF.3 GB de espacio en disco duro como mínimo.

RNF.4 Lector de CD-ROM.

RNF.5 Dispositivos de audio.

RNF.6 Soporte de video que admita resolución de al menos 800x600 y 24 bits.

RNF.7 Dispositivo de red de al menos 10 Mbits.

RNF.8 El sistema interactuará con una impresora que permita imprimir los diferentes contenidos como respuesta a las funcionalidades del sistema.

Restricciones en el diseño y la implementación

RNF.1 Joomla 1.0.12.

RNF.2 MySQL 5.0.

RNF.3 Apache 2.x.

RNF.4 Aptana como herramienta de desarrollo.

RNF.5 Dojo 1.1.

RNF.6 Lenguajes de programación: PHP 5, JavaScript, HTML.

RNF.7 Se recomienda el uso de la arquitectura de 3 capas con el fin de tener una mayor organización del código, además de hacer el software más extensible.

Requerimientos de Apariencia o interfaz externa

RNF.1 El sistema debe tener una interfaz sencilla, intuitiva, amigable, fácil de usar y agradable a la vista del usuario.

RNF.2 El sistema proporcionará claridad y correcta organización de la información, permitiendo la interpretación correcta e inequívoca de ésta.



RNF.3 El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.

Requerimientos de Soporte

RNF.1 Se realizará transferencia tecnológica de la colección a los clientes.

RNF.2 Se impartirán clases a los profesores venezolanos para explicar el funcionamiento y la utilidad del producto.

Requerimientos de Portabilidad

RNF.1 El sistema podrá ser utilizado bajo cualquier Sistema Operativo.

2.6 Modelo de Casos de Uso del Sistema

Este modelo permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Además describe lo que hace el sistema para cada tipo de usuario.

2.6.1 Actores del sistema

Los actores de un sistema son agentes externos, es decir, aquellas personas o sistemas que interactúan con él. En la siguiente figura se puede observar la representación UML de los actores del sistema propuesto.

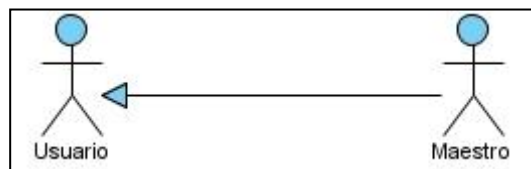


Figura 2.2 Actores del sistema

La siguiente tabla define los actores del sistema y su justificación.

Actor	Descripción
Usuario	Es toda persona que interactúa con la aplicación, es decir, el estudiante o maestro. Cualquier otro se le denominará invitado.
Maestro	Usuario que evalúa o chequea los contenidos visitados por los estudiantes, además de los ejercicios y juegos realizados por los mismos.

Tabla 2.1 Descripción de los actores del sistema



2.6.2 Diagrama de Casos de Uso del Sistema

Los Casos de Uso del Sistema (CUS) son un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable para un actor, es decir, son fragmentos de funcionalidad que el sistema ofrece a los actores que interactúan con el mismo, reportándoles algunos beneficios.

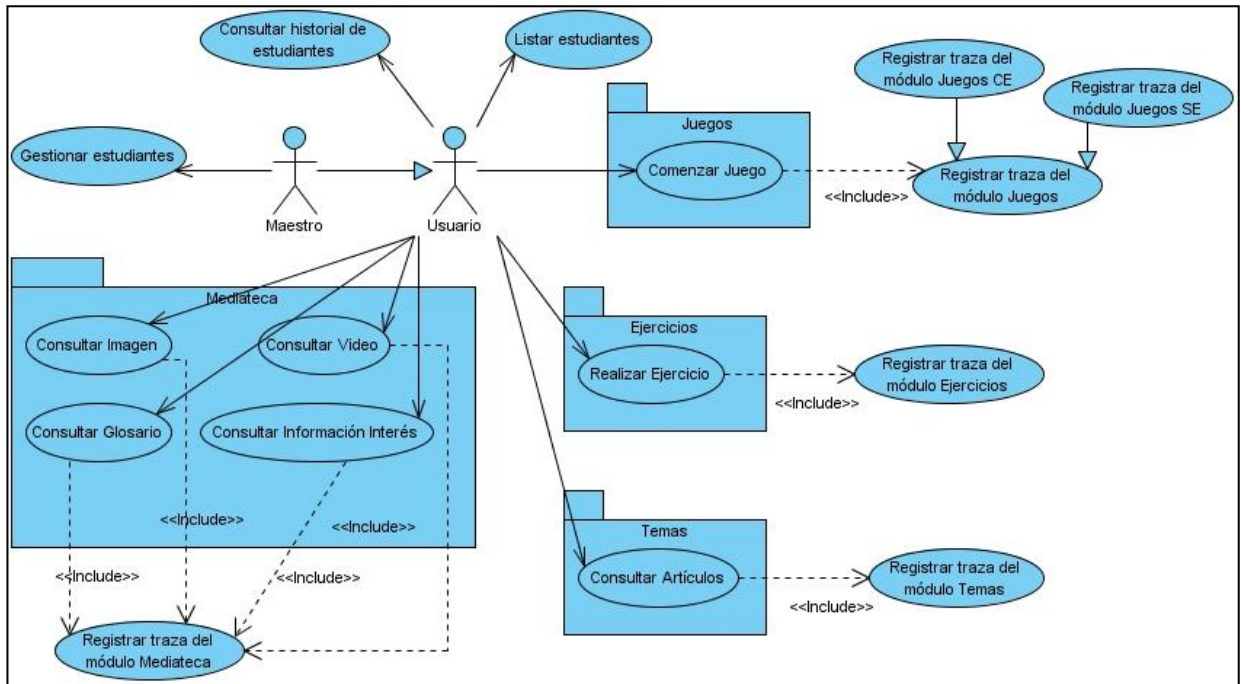


Figura 2.3 Diagrama de casos de uso del sistema

2.6.3 Descripción de los Casos de Uso del Sistema

La descripción de los casos de uso del sistema, detallan las acciones que tienen lugar durante la interacción actor-sistema, es decir, describe el flujo de actividades que realiza el actor al hacer uso del sistema y las correspondientes respuestas del mismo. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre qué es lo que el sistema debe hacer (requisitos).

2.6.3.1 Descripción del CUS “Registrar traza del módulo Temas”

Caso de Uso	Registrar traza del módulo Temas
Actores	Usuario
Propósito	Permitir registrar los datos del resultado de la interacción de los estudiantes con el módulo “Temas”.
Resumen	El caso de uso se inicia cuando el estudiante selecciona el módulo “Temas”. El sistema registra los datos de la interacción del estudiante con el módulo y termina el caso



	de uso.
Referencias	RF 1
Precondiciones	- El usuario debe haberse autenticado como estudiante.
Poscondiciones	- Se registra la traza de la interacción del estudiante con el módulo "Temas".
Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el estudiante selecciona la imagen que metafóricamente representa al módulo "Temas".	2. Registra la visita realizada al módulo. 3. Muestra todos los temas relacionados con el producto y los artículos correspondientes a cada uno.
4. Selecciona el tema y el artículo que desea leer.	5. Registra los siguientes datos: - Temas visitados - Tiempo en el tema - Nombre del artículo visitado por el estudiante - Tiempo de consulta del artículo.
	6. Finaliza el caso de uso.
Curso alternativo de los eventos	
4.a Selecciona la opción "Regresar a la pantalla principal".	
	4.a.1 Finaliza el caso de uso.
Prioridad	Crítico

Tabla 2.2 Descripción del caso de uso "Registrar traza del módulo Temas"

2.6.3.2 Descripción del CUS "Registrar traza del módulo Ejercicios"

Caso de Uso	Registrar traza del módulo Ejercicios
Actores	Usuario
Propósito	Permitir registrar los datos del resultado de la interacción de los estudiantes con el módulo "Ejercicios" del software.
Resumen	El caso de uso se inicia cuando el estudiante selecciona el módulo "Ejercicios". El sistema registra los datos de la interacción del estudiante con el módulo y termina el caso de uso.



Referencias	RF 2
Precondiciones	- El usuario debe haberse autenticado como estudiante.
Poscondiciones	- Se registra la traza de la interacción del estudiante con el módulo "Ejercicios".
Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el estudiante selecciona la imagen que metafóricamente representa al módulo "Ejercicios".	2. Muestra todos los temas relacionados con el producto y le permite seleccionar la cantidad de ejercicios a realizar por cada tema.
3. Selecciona el tema, la cantidad de ejercicios para hacer y la opción "Comenzar".	4. Registra el tiempo a partir del que inicia la realización del ejercicio.
5. Selecciona la opción "Terminar".	6. Registra los siguientes datos: <ul style="list-style-type: none"> - Contenidos - Ejercicios realizados - Calificación obtenida - Porcentaje de efectividad - Nivel - Intentos consumidos
	7. Finaliza el caso de uso.
Curso alternativo de los eventos	
3.a Selecciona la opción "Regresar a la pantalla principal".	
	3.a.1 Finaliza el caso de uso.
Prioridad	Crítico

Tabla 2.3 Descripción del caso de uso "Registrar traza del módulo Ejercicios"

2.6.3.3 Descripción del CUS "Registrar traza del módulo Juegos"

Caso de Uso	Registrar traza del módulo Juegos
Actores	Usuario
Propósito	Permitir registrar los datos del resultado de la interacción



	de los estudiantes con el módulo “Juegos” del software.
Resumen	El caso de uso se inicia cuando el estudiante accede al módulo “Juegos” del producto. El sistema registra los datos de la interacción del estudiante con el módulo y termina el caso de uso.
Referencias	RF 3
Precondiciones	- El usuario debe haberse autenticado como estudiante.
Poscondiciones	- Se registra la traza de la interacción del estudiante con el módulo “Juegos”.
Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el estudiante selecciona la imagen que representa metafóricamente al módulo “Juegos”.	2. Muestra todos los juegos relacionados con el producto y registra la hora de entrada al módulo.
3. El estudiante selecciona un juego.	4. Registra los siguientes datos: <ul style="list-style-type: none"> - Título del juego - Cantidad de veces que jugó - Tiempo de permanencia
	5. Si el juego seleccionado contiene ejercicios: <i>Ver CU Registrar Traza Juegos CE.</i> Si el juego seleccionado no contiene ejercicios: <i>Ver CU Registrar Traza Juegos SE.</i>
	6. Finaliza el caso de uso.
Prioridad	Crítico

Tabla 2.4 Descripción del caso de uso “Registrar traza del módulo Juegos”

2.6.3.4 Descripción del CUS “Registrar traza del módulo Juegos con ejercicios”

Caso de Uso	Registrar traza del módulo Juegos con ejercicios
Actores	Usuario
Propósito	Permitir registrar los datos del resultado de la interacción de los estudiantes con los juegos que contienen ejercicios.
Resumen	El caso de uso se inicia cuando el estudiante selecciona



	uno de los juegos que contienen ejercicios (Descubre la Imagen I, Rescatando al Turpial, Ensartando Aros, Ludo, Localiza y Aprende). El sistema registra los datos de la interacción del estudiante con el juego y termina el caso de uso.
Referencias	RF 3.1
Precondiciones	- El usuario debe haberse autenticado como estudiante.
Poscondiciones	- Se registra la traza de los juegos que contienen ejercicios.
Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el estudiante selecciona uno de los juegos que contienen ejercicios.	2. Registra los siguientes datos: <ul style="list-style-type: none"> - Contenidos - Calificación obtenida - Título del ejercicio - Nivel - Tiempo que se demoró - Lugar alcanzado - Porcentaje de efectividad
	3. Finaliza el caso de uso.
Curso alternativo de los eventos	
1.a Selecciona el juego “Localiza y Aprende”.	
	1.a.1 El sistema registra las respuestas incorrectas por el eje x y por el eje y.
	1. a.2 Regresa al punto 3 del flujo básico.
Prioridad	Crítico

Tabla 2.5 Descripción del caso de uso “Registrar traza del módulo Juegos con ejercicios”

2.6.3.5 Descripción del CUS “Registrar traza del módulo Juegos sin ejercicios”

Caso de Uso	Registrar traza del módulo Juegos sin ejercicios
Actores	Usuario
Propósito	Permitir registrar los datos del resultado de la interacción de los estudiantes con los juegos que no contienen ejercicios.
Resumen	El caso de uso se inicia cuando el estudiante selecciona



	para jugar un juego que no contiene ejercicios. El sistema registra los datos de la interacción del estudiante con el juego y termina el caso de uso.
Referencias	RF 3.2
Precondiciones	- El usuario debe haberse autenticado como estudiante.
Poscondiciones	- Se registra la traza de los juegos que no contienen ejercicios.
Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el estudiante selecciona el juego "Escalera de palabras".	2. Registra los siguientes datos: <ul style="list-style-type: none"> - Calificación - Lugar obtenido - Cantidad de palabras clasificadas - Errores cometidos - Palabras clasificadas de forma incorrecta
	3. Finaliza el caso de uso.
Curso alterno de los eventos	
1.a Selecciona el juego "El Tangram".	
	1.a.1 Registra los siguientes datos: <ul style="list-style-type: none"> - Calificación - Lugar obtenido - Cantidad de figuras - Cantidad de piezas - Cantidad de aciertos - Porcentaje de efectividad
	1.a.2 Regresa al punto 3 del flujo básico.
1.b Selecciona el juego "Rompecabezas".	
	1.b.1 Registra los siguientes datos: <ul style="list-style-type: none"> - Lugar obtenido - Tipo de corte - Si fue terminado - Cantidad de rompecabezas - Cantidad de piezas
	1.b.2 Regresa al punto 3 del flujo básico.
1.c Selecciona el juego "Conociendo el teclado".	



	<p>1.c.1 Registra los siguientes datos:</p> <ul style="list-style-type: none"> - Lugar obtenido - Cantidad de teclas correctas e incorrectas - Calificación - Porcentaje de efectividad
	1.c.2 Regresa al punto 3 del flujo básico.
1.d Selecciona el juego “Escribiendo con el teclado”.	
	<p>1.d.1 Registra los siguientes datos:</p> <ul style="list-style-type: none"> - Lugar obtenido - Cantidad de palabras correctas e incorrectas - Calificación - Porcentaje de efectividad
	1.d.2 Regresa al punto 3 del flujo básico.
1.e Selecciona el juego “Embellecer el patio”.	
	<p>1.e.1 Registra los siguientes datos:</p> <ul style="list-style-type: none"> - Lugar obtenido - Cantidad de objetos a recoger - Cantidad de objetos recogidos
	1.e.2 Regresa al punto 3 del flujo básico.
1.f Selecciona el juego “Buscando las herramientas”.	
	<p>1.f.1 Registra los siguientes datos:</p> <ul style="list-style-type: none"> - Lugar obtenido - Cantidad de herramientas correctas e incorrectas - Calificación - Porcentaje de efectividad
	1.f.2 Regresa al punto 3 del flujo básico.
1.g Selecciona el juego “Buscando Parejas”.	
	<p>1.g.1 Registra los siguientes datos:</p> <ul style="list-style-type: none"> - Lugar obtenido - Nivel
	1.g.2 Regresa al punto 3 del flujo básico.
1.h Selecciona el juego “Vamos a limpiar la pecera”.	
	<p>1.h.1 Registra los siguientes datos:</p> <ul style="list-style-type: none"> - Lugar obtenido



	<ul style="list-style-type: none"> - Cantidad de peces extraídos - Cantidad de peces no extraídos
	1.h.2 Regresa al punto 3 del flujo básico.
1.i Selecciona el juego “Mariposas en el campo”.	
	1.i.1 Registra los siguientes datos: <ul style="list-style-type: none"> - Lugar obtenido - Cantidad de mariposas volando - Cantidad de las mariposas que se encuentran posadas en las flores
	1.i.2 Regresa al punto 3 del flujo básico.
1.j Selecciona el juego “Adivinanzas”.	
	1.j.1 Registra los siguientes datos: <ul style="list-style-type: none"> - Cantidad de adivinanzas seleccionadas - Cantidad de aciertos y fallos - Palabras escritas con errores ortográficos
	1.j.2 Regresa al punto 3 del flujo básico.
1.k Selecciona el juego “Sopa de letras”.	
	1.k.1 Registra los siguientes datos: <ul style="list-style-type: none"> - Lugar obtenido - Nivel - Contenidos - Cantidad de sopas realizadas - Calificación - Porcentaje de efectividad - Palabras no descubiertas
	1.k.2 Regresa al punto 3 del flujo básico.
1.l Selecciona el juego “Crucigramas”.	
	1.l.1 Registra los siguientes datos: <ul style="list-style-type: none"> - Lugar obtenido - Contenidos - Cantidad de crucigramas realizados - Calificación - Porcentaje de efectividad - Palabras no descubiertas
	1.l.2 Regresa al punto 3 del flujo básico.



1.m Selecciona el juego “Descubre la imagen II”.	
	1.m.1 Registra los siguientes datos: <ul style="list-style-type: none"> - Lugar obtenido - Contenidos - Cantidad de juegos realizados - Calificación - Porcentaje de efectividad - Palabras o frases no descubiertas
	1.m.2 Regresa al punto 3 del flujo básico.
Prioridad	Crítico

Tabla 2.6 Descripción del caso de uso “Registrar traza del módulo Juegos sin ejercicios”

2.6.3.6 Descripción del CUS “Registrar traza del módulo Mediateca”

Caso de Uso	Registrar traza del módulo Mediateca
Actores	Usuario
Propósito	Permitir registrar los datos del resultado de la interacción de los estudiantes con el módulo “Mediateca”.
Resumen	El caso de uso se inicia cuando el estudiante selecciona el módulo “Mediateca”. El sistema registra los datos de la interacción del estudiante con el módulo y termina el caso de uso.
Referencias	RF 4
Precondiciones	- El usuario debe haberse autenticado como estudiante.
Poscondiciones	- Se registra la traza de la interacción del estudiante con el módulo “Mediateca”.
Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el estudiante selecciona la imagen que metafóricamente representa al módulo “Mediateca”.	2. El sistema muestra las siguientes secciones y permite seleccionar cada una de ellas: <ul style="list-style-type: none"> - Galería de Imágenes, ver Sección 1: “Galería de imágenes” - Galería de Videos, ver Sección 2: “Galería de videos” - Información de Interés, ver Sección 3:



	“Información de Interés” - Glosario, ver Sección 4: “Glosario”
Sección 1: “Galería de imágenes”: Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. Selecciona la opción de ver la Galería de imágenes.	2. Muestra una interfaz con las imágenes de la galería.
3. Selecciona la imagen que desea observar.	4. Registra los siguientes datos: - Título de la imagen - Cantidad de veces que interactuó con ella - Tiempo
	5. Finaliza el caso de uso.
Sección 2: “Galería de videos”: Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. Selecciona la opción de ver la Galería de videos.	2. Muestra una interfaz con los videos de la galería.
3. Selecciona el video que desea observar.	4. Registra los siguientes datos: - Título del video - Cantidad de veces que interactuó con el mismo - Tiempo
	5. Finaliza el caso de uso.
Sección 3: “Información de interés”: Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. Selecciona la opción de ver la Información de interés.	2. Muestra una interfaz perteneciente a la información de interés que se presenta en la Mediateca.
3. Selecciona la información de interés deseada.	4. Registra los siguientes datos: - Tipo de información consultada (Curiosidades, ¿Sabías qué?) - Visitas realizadas - Título del contenido visitado - Tiempo
	5. Finaliza el caso de uso.
Sección 4: “Glosario”: Curso normal de los eventos	
Acciones del actor	Respuesta del sistema



1. Selecciona la opción de ver el Glosario.	2. Muestra una interfaz con las palabras del glosario.
3. Selecciona las palabras que desea consultar.	4. Registra los siguientes datos: - Palabras consultadas - Tiempo que se demoró en cada palabra
	5. Finaliza el caso de uso.
Prioridad	Crítico

Tabla 2.7 Descripción del caso de uso “Registrar traza del módulo Mediateca”

2.6.3.7 Descripción del CUS “Listar estudiantes”

Caso de Uso	Listar estudiantes
Actores	Usuario
Propósito	Mostrar un listado de los estudiantes que interactuaron con el sistema.
Resumen	El caso de uso se inicia cuando el usuario selecciona el módulo “Resultados”. El sistema muestra el listado de todos los estudiantes que interactuaron con el sistema y termina el caso de uso.
Referencias	RF 5, RF 5.1, RF 5.2, RF 5.3
Precondiciones	- Debe haberse generado el escritorio de trabajo del usuario.
Poscondiciones	- Se muestra el listado de los estudiantes registrados en el software.
Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el usuario selecciona la imagen que metafóricamente representa al módulo “Resultados”.	2. El sistema muestra una interfaz 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)



	<ul style="list-style-type: none"> - Apellido(s) - Sesiones - Fecha de la última sesión iniciada Permite ordenar los resultados.
	5. Finaliza el caso de uso.
Curso alterno de los eventos	
1.a Selecciona algunos de los datos del listado.	
	1.a.1 El sistema ordena a los estudiantes por el criterio elegido por el usuario.
	1.a.2 Regresa al punto 5 del flujo básico.
3.a Selecciona la opción “Regresar a la pantalla principal”.	
	3.a.1 Finaliza el caso de uso.
Prioridad	Crítico

Tabla 2.8 Descripción del caso de uso “Listar estudiantes”

2.6.3.8 Descripción del CUS “Consultar historial de estudiantes”

Caso de Uso	Consultar historial de estudiantes
Actores	Usuario
Propósito	Buscar y listar de manera ordenada un resumen de los resultados obtenidos por los estudiantes en su interacción con el software.
Resumen	El caso de uso se inicia cuando el usuario selecciona ver el historial de un estudiante. El sistema muestra los resultados del estudiante seleccionado y termina el caso de uso.
Referencias	RF 6
Precondiciones	<ul style="list-style-type: none"> - Debe haberse generado el escritorio de trabajo del usuario. - El usuario debe haber seleccionado el estudiante del que desea consultar su historial.
Poscondiciones	- Se consultó el historial de un estudiante.
Curso normal de los eventos	
Acciones del actor	Respuesta del sistema



<p>1. El caso de uso se inicia cuando el usuario selecciona un estudiante para ver su historial.</p>	<p>2. Muestra la opción Historial. Y permite:</p> <ul style="list-style-type: none"> - Seleccionar Historial. - Regresar a la pantalla principal del producto.
<p>3. Selecciona la opción Historial.</p>	<p>4. El sistema busca los resultados del estudiante seleccionado y muestra una interfaz con los siguientes datos:</p> <ul style="list-style-type: none"> - Fecha y hora de la última sesión de trabajo iniciada. - Menú de categorías: <ul style="list-style-type: none"> • Datos Generales • Itinerario • Temas • Ejercicios • Juegos • Galería de Imágenes • Galería de Videos • Información de Interés • Glosario - La categoría de Datos Generales seleccionada y los datos que esta contiene: <ul style="list-style-type: none"> • Nombre(s) y Apellidos(s) del estudiante • Grado • Grupo • Escuela • Fecha y Hora de entrada al software • Equipo de trabajo <p>Y permite:</p> <ul style="list-style-type: none"> • Seleccionar la fecha y hora de la sesión de trabajo que se desea visualizar • Seleccionar una nueva categoría. • Regresar a la página anterior
<p>5. Selecciona una categoría.</p>	<p>6. Muestra los datos correspondientes a la categoría</p>



	seleccionada.
	7. Finaliza el caso de uso.
Curso alternativo de los eventos	
1.a Selecciona un estudiante sin sesiones de estudio.	
	1.a.1 Muestra el mensaje “El estudiante (Nombre(s) y Apellido(s)) no tiene sesiones de estudio en este programa educativo”.
	1.a.2 Regresa al punto 1 del flujo básico.
2.a Selecciona la opción “Regresar a la página anterior”.	
	2.a.1 Finaliza el caso de uso.
5.a Selecciona la categoría “Itinerario”.	
	5.a.1 El sistema muestra dos tablas que contienen varios datos acerca del recorrido realizado por el estudiante: 1. Resumen del Itinerario: <ul style="list-style-type: none"> - Fecha - Hora de entrada - Hora de salida - Tiempo general en el software 2. Detalles del Itinerario: <ul style="list-style-type: none"> - Módulos visitados - Cantidad de veces que visitó cada módulo - Tiempo total en cada uno
	5.a.2 Regresa al punto 7 del flujo básico.
5.b Selecciona la categoría “Temas”.	
	5.b.1 El sistema muestra dos tablas que contienen varios datos acerca del resultado de la interacción del estudiante con el módulo “Temas”: 1. Resumen de Temas: <ul style="list-style-type: none"> - Temas visitados - Cantidad de veces que visitó cada tema - Tiempo total en cada tema 2. Detalles de Temas: <ul style="list-style-type: none"> - Temas visitados - Tiempo total en cada tema



	<ul style="list-style-type: none"> - Epígrafes visitados por temas - Tiempo en cada epígrafe
	5.b.2 Regresa al punto 7 del flujo básico.
5.c Selecciona la categoría “Ejercicios”.	
	<p>5.c.1 El sistema muestra dos tablas que contienen datos acerca del resultado de la interacción del estudiante con el módulo “Ejercicios”:</p> <p>1. Resumen de Ejercicios:</p> <ul style="list-style-type: none"> - Contenidos - Cantidad total de ejercicios realizados - Calificación - Por ciento de efectividad <p>2. Detalles de Ejercicios:</p> <ul style="list-style-type: none"> - Números de los ejercicios realizados - Nivel - Calificación - Tiempo - Por ciento de efectividad - Intentos consumidos
	5.c.2 Regresa al punto 7 del flujo básico.
5.d Selecciona la categoría “Juegos”.	
	<p>5.d.1 El sistema muestra los siguientes datos para todos los juegos:</p> <ul style="list-style-type: none"> - Hora de entrada al módulo - Hora de salida del módulo - Título de los juegos - Cantidad de veces que jugó - Tiempo de permanencia
	<p>5.d.2 Si los juegos con los que interactuó el estudiante contienen ejercicios, el sistema muestra además dos tablas:</p> <p>1. Resumen del Juego:</p> <ul style="list-style-type: none"> - Calificación general - Por ciento de efectividad - Lugar alcanzado



	<ul style="list-style-type: none"> - Contenidos - Cantidad total de ejercicios realizados por temas <p>2. Detalles del Juego:</p> <ul style="list-style-type: none"> - Ejercicio - Nivel - Calificación - Tiempo
	<p>5.d.3 Si los juegos con los que interactuó el estudiante no contienen ejercicios, el sistema muestra información diferenciada para cada uno:</p> <p>Para el juego “Escalera de palabras”:</p> <ul style="list-style-type: none"> - Objetivo del juego - Calificación - Lugar obtenido - Cantidad total de palabras clasificadas - Cantidad total de errores cometidos. - Palabras clasificadas de forma incorrecta <p>Para el juego “El Tangram”:</p> <ul style="list-style-type: none"> - Calificación obtenida - Lugar obtenido - Cantidad de figuras armadas - Porcentaje de efectividad total <p>Además muestra otra tabla que contiene los detalles de los ejercicios realizados en el juego, en este caso:</p> <ul style="list-style-type: none"> - Figura armada - Cantidad total de piezas - Cantidad de aciertos - Tiempo <p>Para el juego “Rompecabezas”:</p> <ul style="list-style-type: none"> - Título del rompecabezas - Tiempo - Tipo de corte - Cantidad de piezas - Si fue terminado o no <p>Para el juego “Conociendo el teclado”:</p>



	<ul style="list-style-type: none"> - Objetivo del juego - Calificación obtenida - Lugar obtenido - Porcentaje de efectividad - Cantidad de teclas oprimidas de forma correcta e incorrecta <p>Además muestra otra tabla que contiene los detalles del juego, en este caso:</p> <ul style="list-style-type: none"> - Teclas oprimidas de forma correcta e incorrecta <p>Para el juego “Escribiendo con el teclado”:</p> <ul style="list-style-type: none"> - Objetivo - Lugar obtenido - Cantidad total de palabras escritas - Cantidad de palabras correctas e incorrectas - Calificación obtenida - Porcentaje de efectividad por contenidos <p>Además muestra otra tabla que contiene los detalles del juego, en este caso:</p> <ul style="list-style-type: none"> - Palabras escritas de forma correcta e incorrecta. <p>Para el juego “Embellecer el patio”:</p> <ul style="list-style-type: none"> - Objetivo - Lugar obtenido - Cantidad total de objetos a recoger - Cantidad total de objetos recogidos <p>Para el juego “Buscando las herramientas”:</p> <ul style="list-style-type: none"> - Objetivo - Lugar obtenido - Cantidad total de herramientas escogidas de forma correcta e incorrecta. - Calificación obtenida - Porcentaje de efectividad por contenidos <p>Además muestra otra tabla que contiene los detalles del juego, en este caso:</p> <ul style="list-style-type: none"> - Herramientas seleccionadas de forma correcta e incorrecta
--	---



	<p>Para el juego “Buscando Parejas”:</p> <ul style="list-style-type: none"> - Objetivo - Lugar obtenido - Título del nivel que se seleccionó <p>Para el juego “Vamos a limpiar la pecera”:</p> <ul style="list-style-type: none"> - Objetivo - Lugar obtenido - Cantidad total de peces extraídos de la pecera - Cantidad total de peces no extraídos de la pecera <p>Para el juego “Mariposas en el campo”:</p> <ul style="list-style-type: none"> - Objetivo - Lugar obtenido - Cantidad total de mariposas volando - Cantidad total de mariposas posadas en las flores <p>Para el juego “Adivinanzas”:</p> <ul style="list-style-type: none"> - Cantidad de adivinanzas seleccionadas por contenidos - Cantidad de aciertos - Cantidad de fallos <p>Además muestra otra tabla que contiene los detalles del juego, en este caso:</p> <ul style="list-style-type: none"> - Palabras escritas con errores ortográficos. <p>Para el juego “Sopa de letras”:</p> <ul style="list-style-type: none"> - Lugar obtenido - Nivel del juego - Contenidos - Cantidad total de sopas realizadas por contenidos - Calificación obtenida - Porcentaje de efectividad por contenidos <p>Además muestra otra tabla que contiene los detalles de los contenidos, en este caso:</p> <ul style="list-style-type: none"> - Contenido - Nivel - Palabras no descubiertas <p>Para el juego “Crucigramas”:</p>
--	---



	<ul style="list-style-type: none"> - Lugar obtenido - Contenidos - Cantidad total de crucigramas realizados por contenidos - Calificación obtenida - Por ciento de efectividad por contenidos <p>Además muestra otra tabla que contiene los detalles de los contenidos del juego, en este caso:</p> <ul style="list-style-type: none"> - Contenido - Palabras no descubiertas <p>Para el juego “Descubre la Imagen II”:</p> <ul style="list-style-type: none"> - Lugar obtenido - Contenidos - Cantidad total de Descubre la imagen realizado por contenidos - Calificación obtenida - Por ciento de efectividad por contenidos <p>Además muestra otra tabla debajo que contiene los detalles de los contenidos del juego, en este caso:</p> <ul style="list-style-type: none"> - Contenido - Palabras no descubiertas
<p>5.d.4 Interactúa con el juego “Localiza y Aprende”.</p>	
	<p>5.d.4 Si el estudiante interactuó con el juego “Localiza y Aprende”, el sistema muestra además en el resumen del juego los siguientes datos:</p> <ul style="list-style-type: none"> - Respuestas incorrectas por el eje x - Respuestas incorrectas por el eje y
	<p>5.d.5 Regresa al punto 7 del flujo básico.</p>
<p>5.e Selecciona la categoría “Galería de Imágenes”.</p>	
	<p>5.e.1 Muestra dos tablas que contienen varios datos sobre la interacción del estudiante con la galería de imágenes:</p> <p>1. Resumen de la Galería de Imágenes:</p> <ul style="list-style-type: none"> - Cantidad de imágenes con las que se interactuó - Tiempo



	<p>2. Detalles de la Galería de Imágenes:</p> <ul style="list-style-type: none"> - Título de la imagen - Cantidad de veces que la observó - Tiempo
	5.e.2 Regresa al punto 7 del flujo básico.
5.f Selecciona la categoría “Galería de Videos”.	
	<p>5.f.1 Muestra dos tablas que contienen varios datos sobre la interacción del estudiante con la galería de videos:</p> <p>1. Resumen de la Galería de Videos:</p> <ul style="list-style-type: none"> - Cantidad de videos con los que se interactuó - Tiempo <p>2. Detalles de la Galería de Videos:</p> <ul style="list-style-type: none"> - Título del video - Cantidad de veces que lo observó - Tiempo
	5.f.2 Regresa al punto 7 del flujo básico.
5.g Selecciona la categoría “Información de Interés”.	
	<p>5.g.1 Muestra dos tablas que contienen varios datos sobre la interacción del estudiante con la información de interés que se presenta en la Mediateca:</p> <p>1. Resumen de la Información de Interés:</p> <ul style="list-style-type: none"> - Tipo de información consultada - Cantidad de veces visitada - Tiempo <p>2. Detalles de la Información de Interés:</p> <ul style="list-style-type: none"> - Título del contenido visitado - Cantidad de veces que lo observó - Tiempo
	5.g.2 Regresa al punto 7 del flujo básico.
5.h Selecciona la categoría “Glosario”.	
	<p>5.h.1 Muestra una tabla que contienen varios datos sobre la interacción del estudiante con las palabras del glosario, estos son:</p> <p>1. Resumen del Glosario:</p>



	<ul style="list-style-type: none"> - Cantidad de palabras consultadas - Tiempo total <p>2. Detalles del Glosario:</p> <ul style="list-style-type: none"> - Palabras consultadas - Cantidad de veces que la observó - Tiempo dedicado a cada palabra
	5.h.2 Regresa al punto 7 del flujo básico.
Prioridad	Crítico

Tabla 2.9 Descripción del caso de uso “Consultar historial de estudiantes”

2.6.3.9 Descripción del CUS “Gestionar estudiantes”

Caso de Uso	Gestionar estudiantes
Actores	Profesor
Propósito	Permitir adicionar, modificar o eliminar los datos de un estudiante.
Resumen	El caso de uso se inicia cuando el profesor selecciona la opción que le permite realizar una acción sobre un estudiante. El profesor puede adicionar, modificar y eliminar estudiantes. El caso de uso termina.
Referencias	RF 7, RF 7.1, RF 7.2, RF 7.3
Precondiciones	<ul style="list-style-type: none"> - Debe haberse generado el escritorio de trabajo del profesor. - Para modificar un estudiante debe estar seleccionado previamente. - Para eliminar un estudiante debe estar seleccionado previamente.
Poscondiciones	- Se adicionó, modificó o eliminó estudiantes por el profesor.
Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el profesor selecciona la opción “Registro”.	2. El sistema muestra las siguientes secciones y permite seleccionar cada una de ellas: <ul style="list-style-type: none"> - Adicionar estudiantes al registro, ver Sección 1: “Adicionar estudiantes”



	<ul style="list-style-type: none"> - Modificar los datos del estudiante seleccionado, ver Sección 2: “Modificar estudiantes” - Eliminar los datos del estudiante, ver Sección 3: “Eliminar estudiantes”
	3. Finaliza el caso de uso.
Sección 1: “Adicionar estudiantes”: Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. Selecciona la opción de adicionar.	2. Muestra una interfaz para agregar a los estudiantes al registro con los siguientes datos: <ul style="list-style-type: none"> - Escuela - Grado - Sección - Nombre(s) y Apellido(s) - Y permite seleccionar el número de la lista - También permite cancelar la acción
3. Introduce los datos del nuevo estudiante.	
4. Selecciona la opción de agregar.	5. El sistema verifica que ningún campo haya quedado vacío.
	6. El sistema valida los datos.
	7. El sistema registra el nuevo estudiante y muestra un mensaje de confirmación “El estudiante ha sido agregado satisfactoriamente”.
	8. Finaliza el caso de uso.
Curso alterno de los eventos	
2.a “Selecciona la opción Cancelar”.	
	2.a.1 Regresa a la interfaz de usuario.
5.a “Entrada de datos incorrectos”.	
	5.a.1 El sistema emite un mensaje “Todos los campos deben de estar llenos”.
6.a “Comprobar datos del estudiante”.	
	6.a.1 Muestra un mensaje “El estudiante ya existe en el registro”.
6.b “Comprobar número de lista”.	
	6.b.1 Muestra un mensaje “Este número de lista no está



	disponible porque ya está asignado a un estudiante”.
Sección 2: “Modificar estudiantes”: Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. Selecciona la opción de modificar.	2. Muestra una interfaz con los datos del estudiante seleccionado. También permite cancelar la acción.
3. Modifica los datos que desea.	
4. Indica al sistema guardar los datos.	5. El sistema valida los datos.
	6. El sistema actualiza los datos del estudiante.
	7. Finaliza el caso de uso.
Curso alternativo de los eventos	
2.a “Selecciona la opción Cancelar”.	
	2.a.1 Regresa a la interfaz de usuario.
5.a “Entrada de datos incorrectos”.	
	5.a.1 El sistema emite un mensaje “Todos los campos deben de estar llenos”.
Sección 3 “Eliminar estudiantes”: Curso normal de los eventos	
Acciones del actor	Respuesta del sistema
1. Selecciona la opción de eliminar.	2. Muestra un mensaje de confirmación.
3. Selecciona la opción de aceptar.	4. El sistema elimina el estudiante de la base de datos.
	5. Finaliza el caso de uso.
Curso alternativo de los eventos	
3.a “Selecciona la opción Cancelar”.	
	3.a.1 Regresa a la interfaz de usuario.
Prioridad	Crítico

Tabla 2.10 Descripción del caso de uso “Gestionar estudiantes”

2.7 Conclusiones

Durante este capítulo fueron expuestas las características que contendrá el sistema. Se realizó el modelado del dominio en el que se reflejan los conceptos que fueron utilizados con posterioridad, además de describir la propuesta de solución. Se identificaron los requisitos funcionales y no funcionales que debe cumplir el sistema en cuestión, y con ello fueron



expuestos los casos de uso a tratar durante el desarrollo del mismo, con la correspondiente descripción textual de cada uno, lo cual provee de una visión general de qué es lo que el sistema debe hacer, por lo que se está en condiciones de pasar a ver cómo es que el mismo va a realizar las operaciones antes descritas y con ello, darle solución a los problemas planteados.



Capítulo

Análisis y Diseño del Sistema



3.1 Introducción

El siguiente capítulo presenta la concepción general referente al análisis y diseño del sistema propuesto, donde el análisis tiene como objetivo mantener un modelo eficiente de la solución propuesta que sirva de base para el diseño. Este último tiene a su vez la finalidad de mostrar cómo es que está construido el sistema, lo cual en este caso se realiza a partir de los diagramas de clases Web, que tienen el objetivo de describir la interacción entre las distintas páginas de la aplicación. Por otro lado, se exponen las características de los patrones de diseño utilizados. Finalmente, se presenta el modelo de datos y el diagrama de clases persistentes de la base de datos del sistema.

3.2 Modelo de Análisis

Este modelo esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluida aquella significativa para la arquitectura; sirve como una primera aproximación del diseño y está estructurado por clases y paquetes estereotipados. Su objetivo es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución.

3.2.1 Diagramas de clases del análisis

Un diagrama de clases de este tipo es un artefacto en el que se representan los conceptos en un dominio del problema.

En el análisis se presentan los siguientes estereotipos de clases:

Clase de frontera o interfaz: Modela la interfaz del sistema, y maneja la comunicación entre el entorno y el interior del mismo. Durante el diseño, estas clases son refinadas para tomar en consideración los mecanismos de interfaz seleccionados o implementados, además de facilitar la comunicación con otros sistemas.

Clases de entidad o sistema: Representan la información manejada en el caso de uso, además de que modelan información y comportamiento asociado que generalmente es de larga duración. Reflejan entidades del mundo real, que resultan necesarias para realizar tareas internas del sistema.



Clases de control o software: Coordinan los eventos necesarios para la realización o especificación del caso de uso, con 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.

A continuación se muestran los diagramas de clases del análisis para cada uno de los casos de uso del sistema:

3.2.1.1 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Temas”



Figura 3.1 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Temas”

3.2.1.2 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Ejercicios”

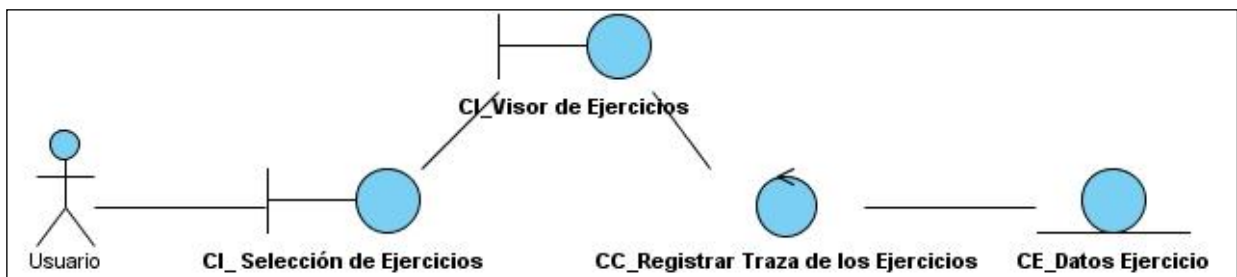


Figura 3.2 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Ejercicios”

3.2.1.3 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Juegos”

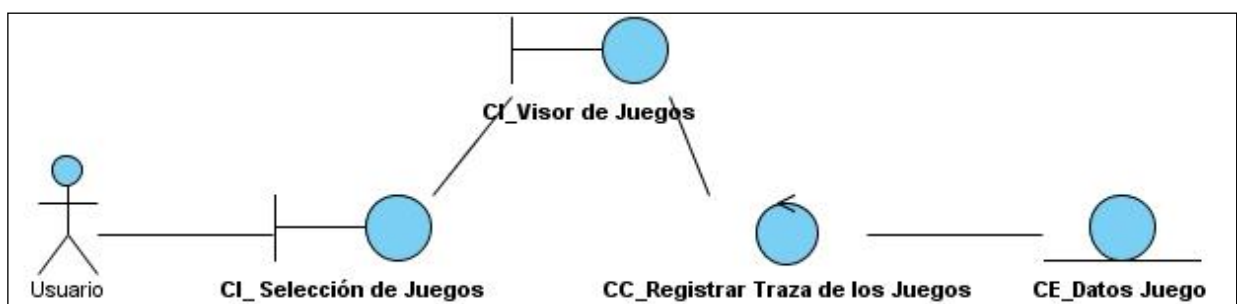


Figura 3.3 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Juegos”



3.2.1.4 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Juegos con ejercicios”

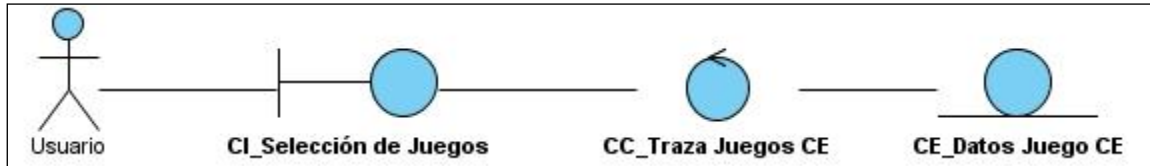


Figura 3.4 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Juegos con ejercicios”

3.2.1.5 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Juegos sin ejercicios”

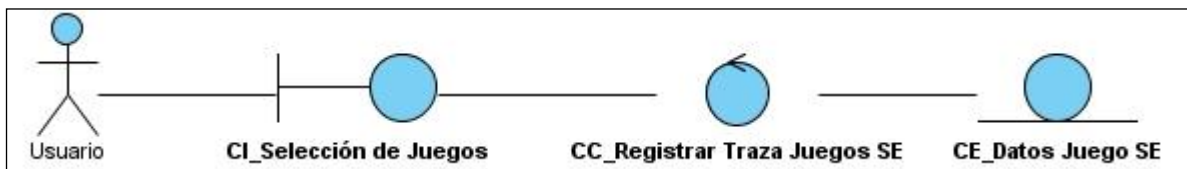


Figura 3.5 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Juegos sin ejercicios”

3.2.1.6 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Mediateca”

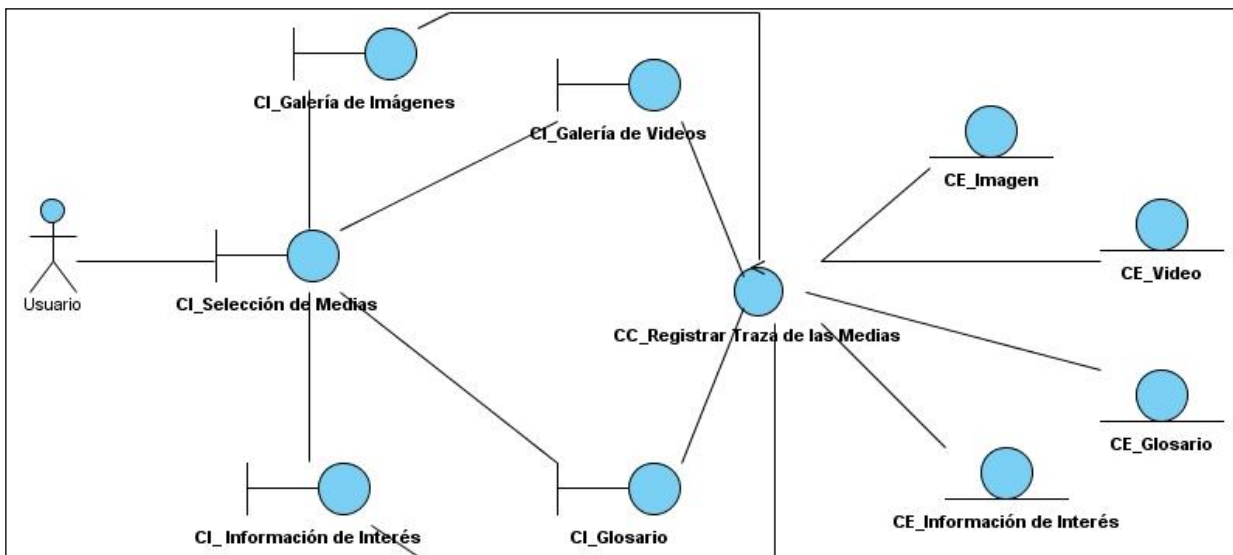


Figura 3.6 Diagrama de clases del análisis del caso de uso “Registrar traza del módulo Mediateca”

3.2.1.7 Diagrama de clases del análisis del caso de uso “Listar estudiantes”

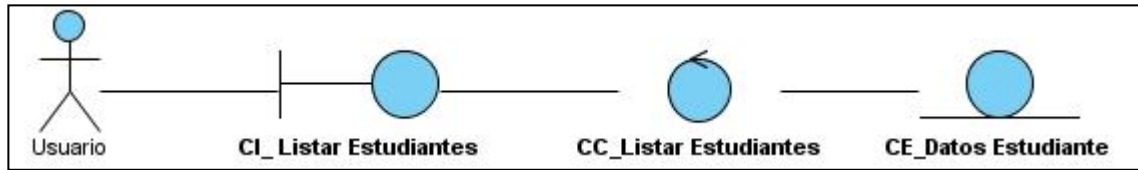


Figura 3.7 Diagrama de clases del análisis del caso de uso “Listar estudiantes”

3.2.1.8 Diagrama de clases del análisis del caso de uso “Consultar historial de estudiantes”

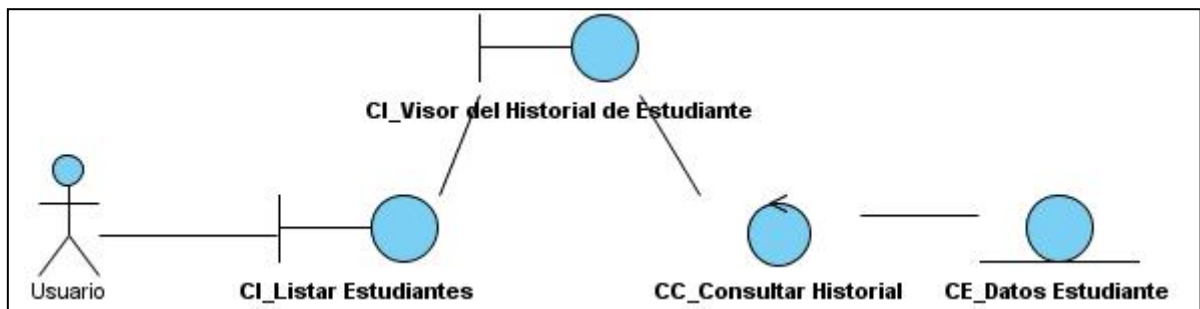


Figura 3.8 Diagrama de clases del análisis del caso de uso “Consultar historial de estudiantes”

3.2.1.9 Diagrama de clases del análisis del caso de uso “Gestionar estudiantes”

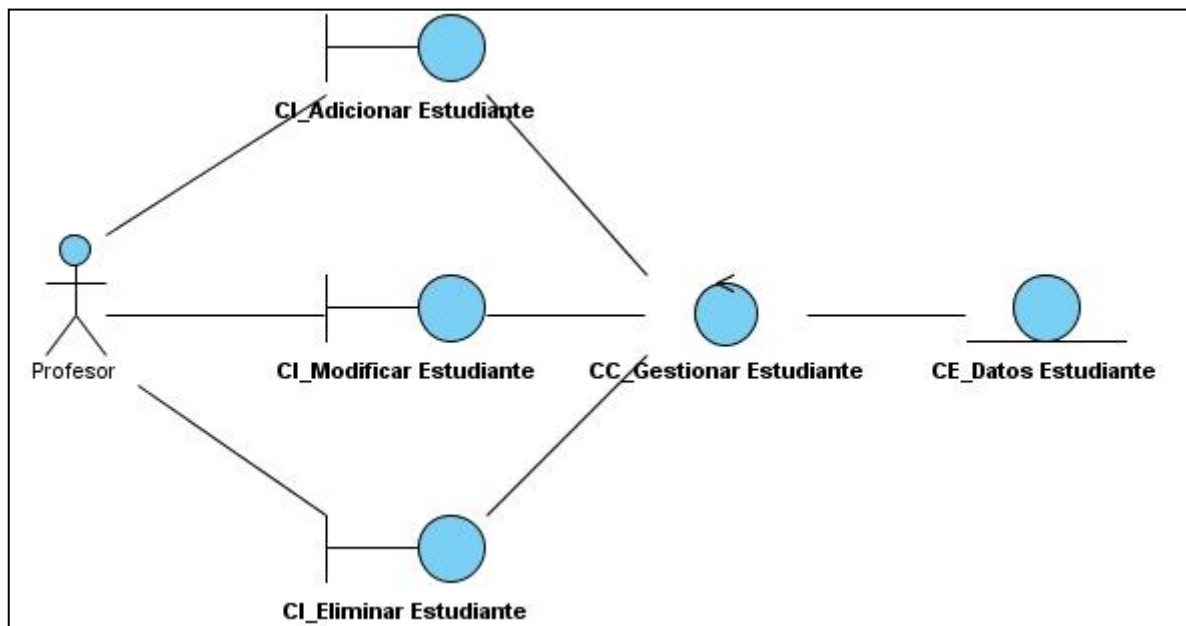


Figura 3.9 Diagrama de clases del análisis del caso de uso “Gestionar estudiantes”

3.3 Modelo de Diseño

Es un modelo de objeto que describe las realizaciones de los casos de uso y sirve como una abstracción del modelo de implementación y su código fuente. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, es decir, cómo

cumple el sistema sus objetivos. Es utilizado como una actividad esencial de entrada para la implementación.

3.3.1 Diagramas de clases del diseño

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 los diagramas de clases Web para cada uno de los casos de uso del sistema:

3.3.1.1 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Temas”

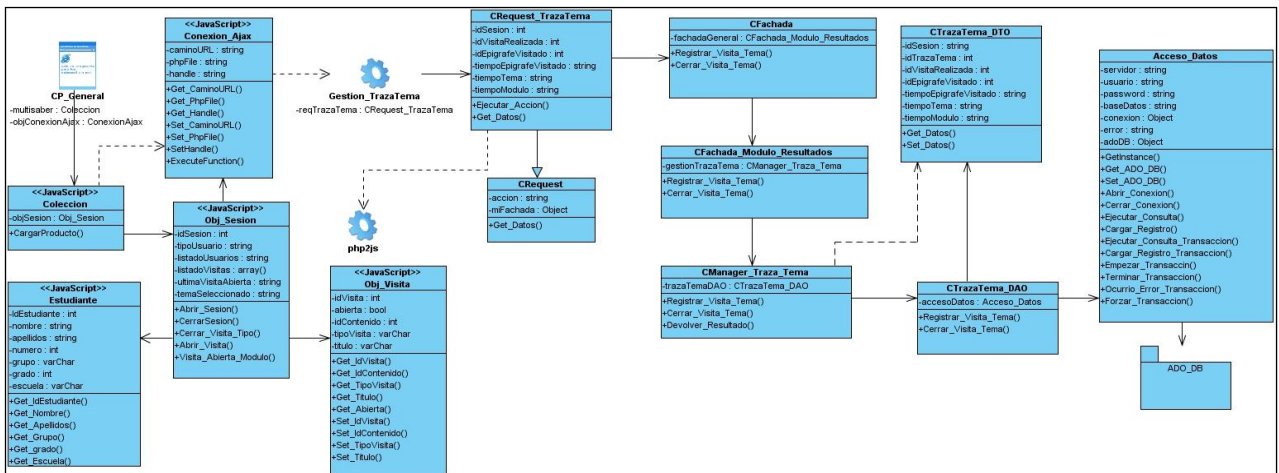


Figura 3.10 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Temas”

3.3.1.2 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Ejercicios”

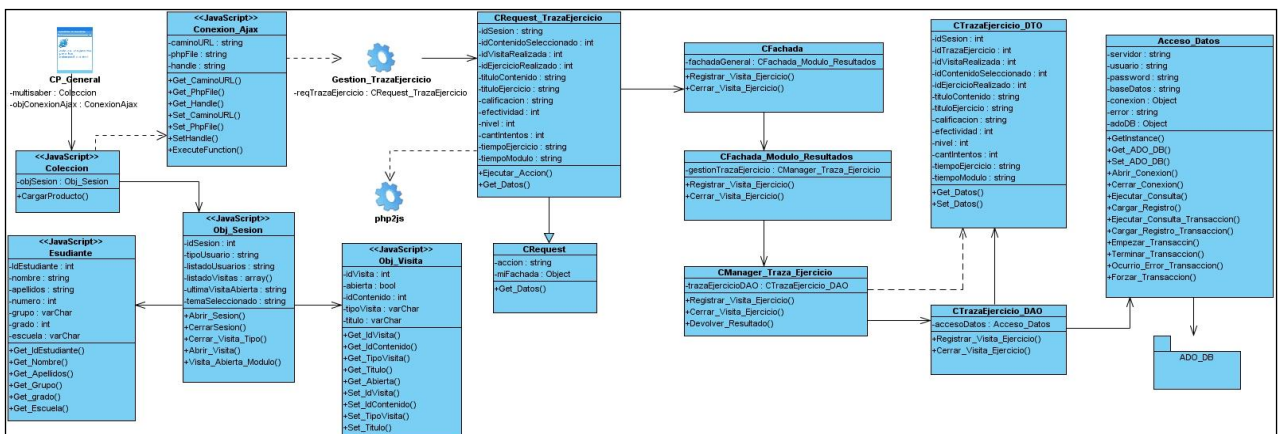


Figura 3.11 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Ejercicios”



3.3.1.3 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Juegos”

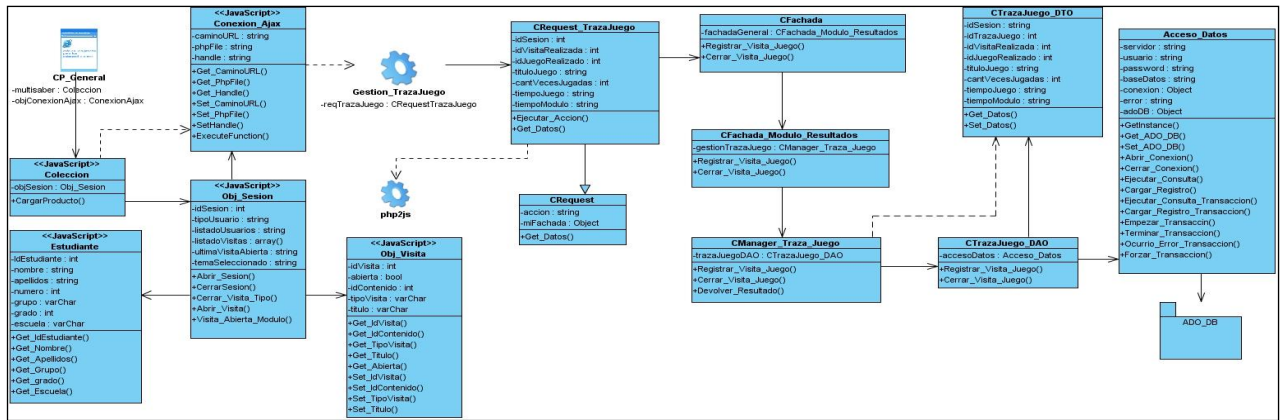


Figura 3.12 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Juegos”

3.3.1.4 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Juegos con ejercicios”

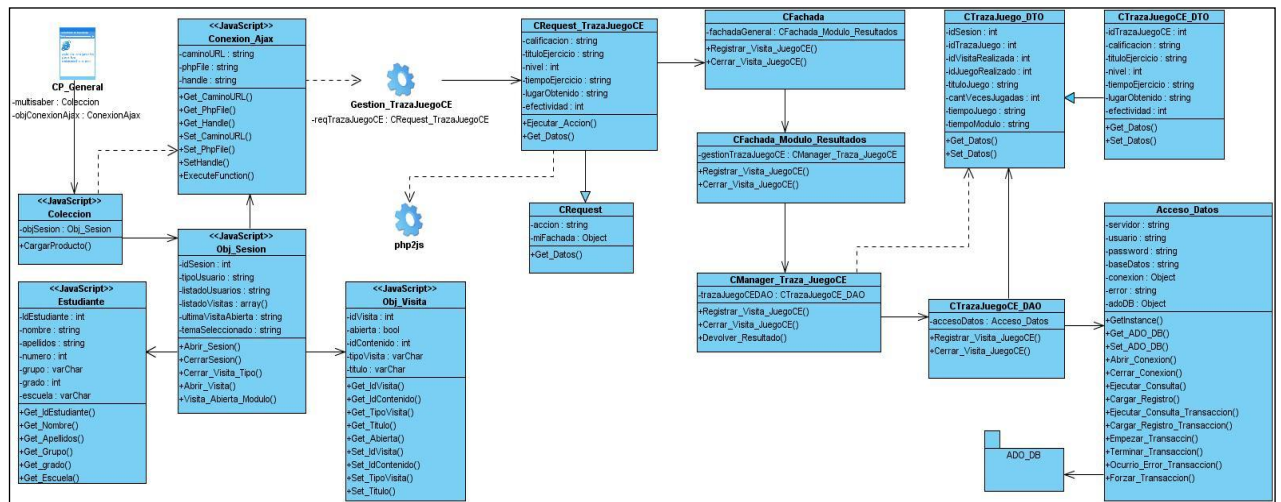


Figura 3.13 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Juegos con ejercicios”



3.3.1.5 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Juegos sin ejercicios”

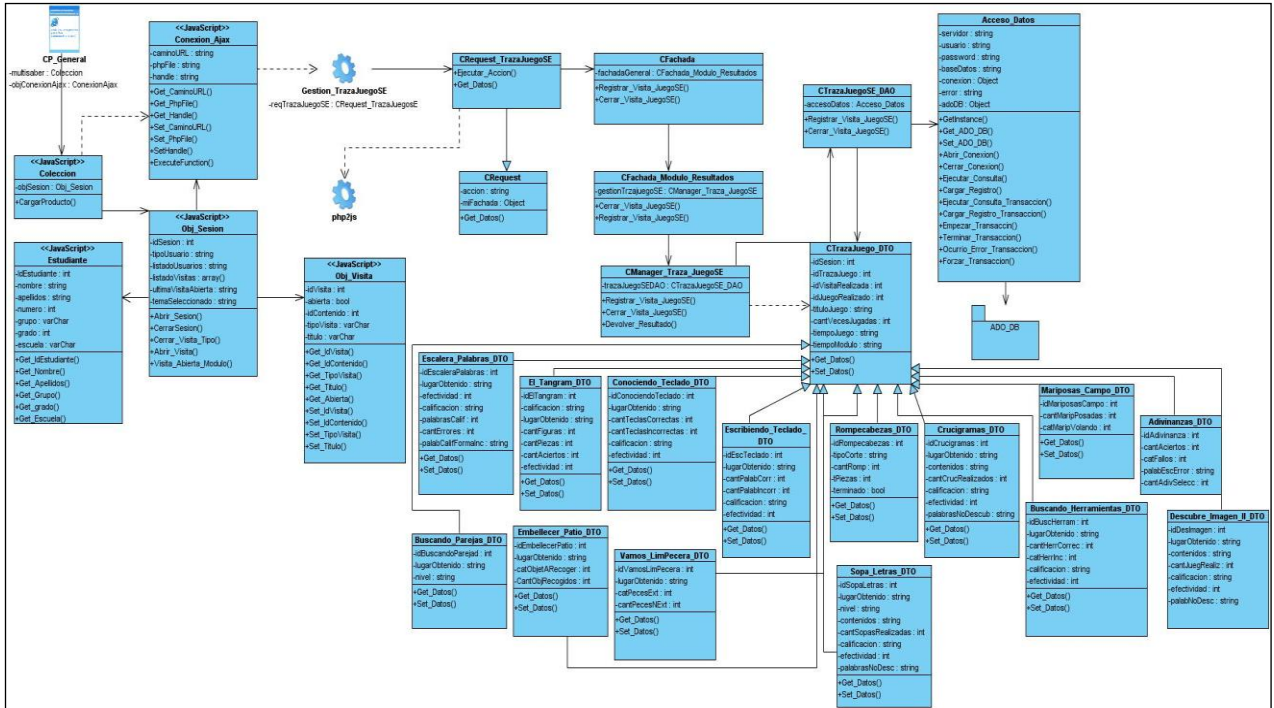


Figura 3.14 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Juegos sin ejercicios”

3.3.1.6 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Mediateca”

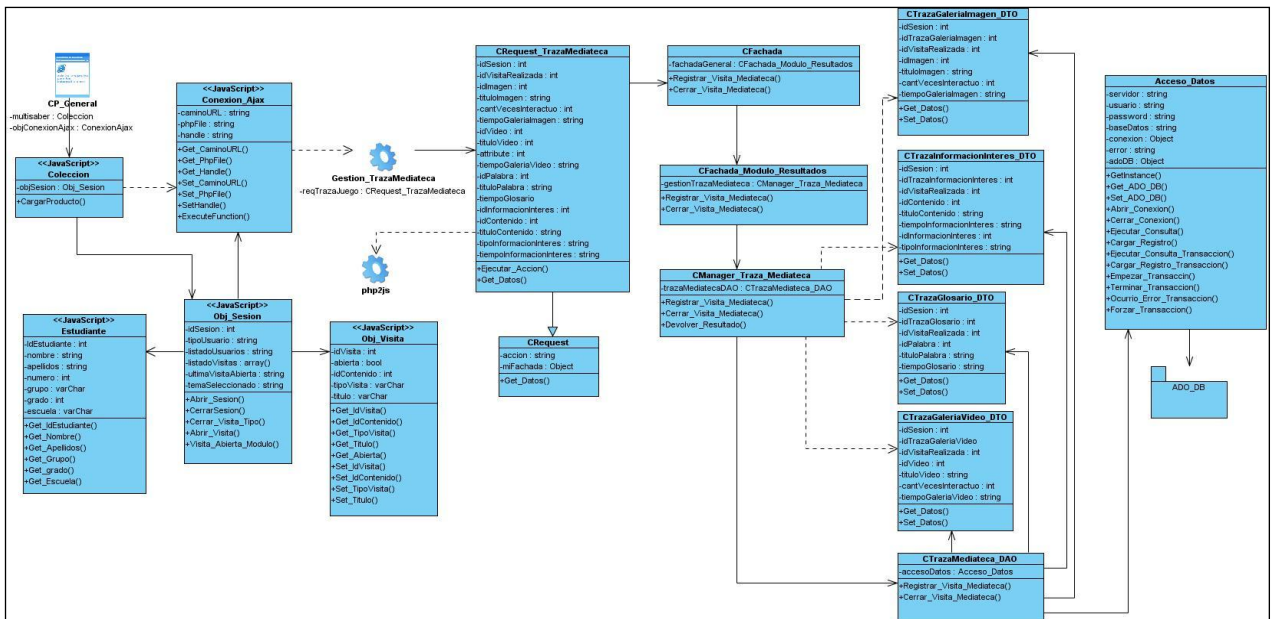


Figura 3.15 Diagrama de clases Web para el caso de uso “Registrar traza del módulo Mediateca”

1.3.1.7 Diagrama de clases Web para el caso de uso “Listar Estudiantes”

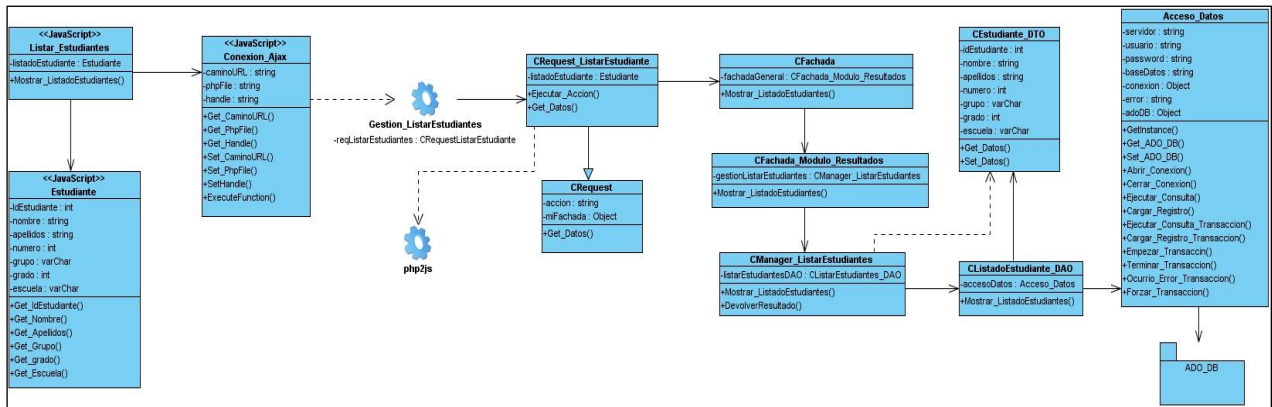


Figura 3.16 Diagrama de clases Web para el caso de uso “Listar Estudiantes”

3.3.1.8 Diagrama de clases Web para el caso de uso “Consultar historial de estudiantes”

Un caso de uso puede ser dividido en escenarios, los cuales representan funcionalidades específicas dentro del mismo y se utilizan cuando se tienen casos de uso que poseen varias funcionalidades, para lograr con ello una mayor comprensión de todas las actividades y procesos que este engloba.

Debido a que el caso de uso “Consultar historial de estudiantes” posee varios escenarios a continuación se muestra el diagrama de clases Web para el escenario “Consultar Datos Generales”, el resto de los mismos se pueden encontrar en el Anexo 3.

3.3.1.8.1 Diagrama de clases Web para el escenario “Consultar Datos Generales”

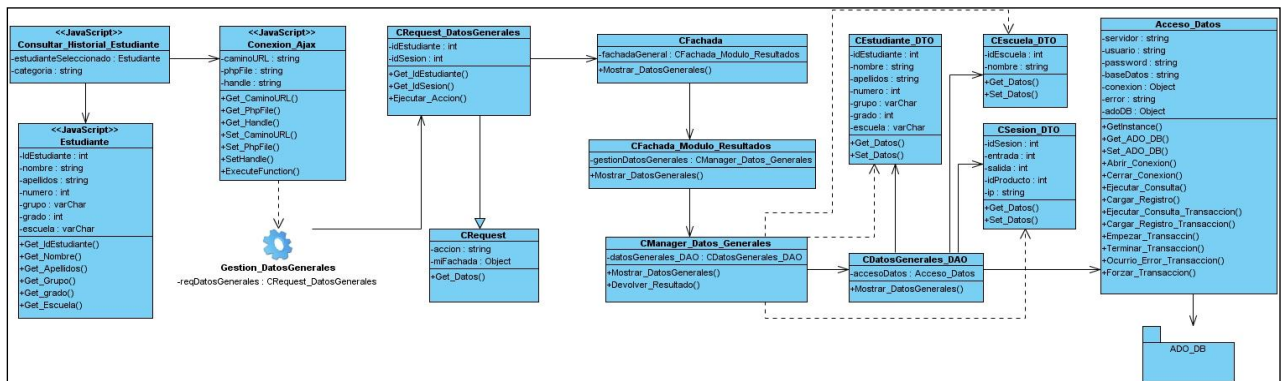


Figura 3.17 Diagrama de clases Web para el caso de uso “Consultar Datos Generales”

3.3.1.9 Diagrama de clases Web para el caso de uso “Gestionar Estudiantes”

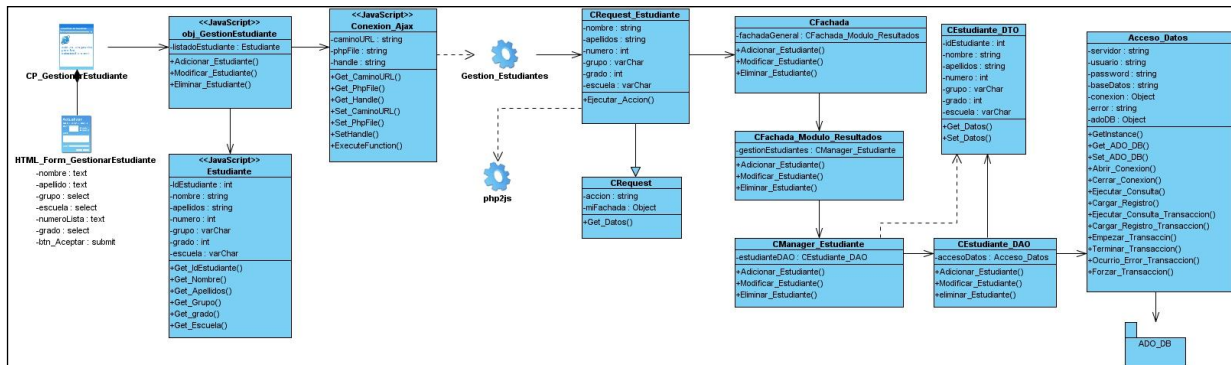


Figura 3.18 Diagrama de clases Web para el caso de uso “Gestionar Estudiantes”

3.4 Patrones

El patrón es una descripción nombrada de un problema recurrente y una solución para este, que se puede aplicar en varios contextos y es normalmente independiente de la implementación. Existen varios tipos de patrones y entre los más usados se encuentran los de diseño (GoF), arquitectura, responsabilidades (GRASP) y casos de uso.

3.4.1 Patrones de diseño

Los patrones de diseño (design patterns, en inglés) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Son utilizados principalmente en los modelos de clases y sus relaciones.

A continuación se describen los patrones utilizados en esta investigación:

“Facade” (Fachada): Es un patrón estructural que proporciona una interfaz unificada para un conjunto de interfaces de un sistema.

Entre los principales resultados obtenidos están los siguientes:

- Al separar al cliente de los componentes del subsistema, se reduce el número de objetos con los que el cliente trata, facilitando así el uso del subsistema.
- Se promueve un acoplamiento débil entre el subsistema y sus clientes, eliminándose o reduciéndose las dependencias.
- No existen obstáculos para que las aplicaciones usen las clases del subsistema que necesitan. De esta forma se puede elegir entre facilidad de uso y generalidad.



- Facade ofrece un punto de acceso al resto de las clases, si estas cambian o se sustituyen por otras solo hay que actualizar la clase Fachada sin que el cambio afecte a las aplicaciones cliente.

Estructura de Facade:

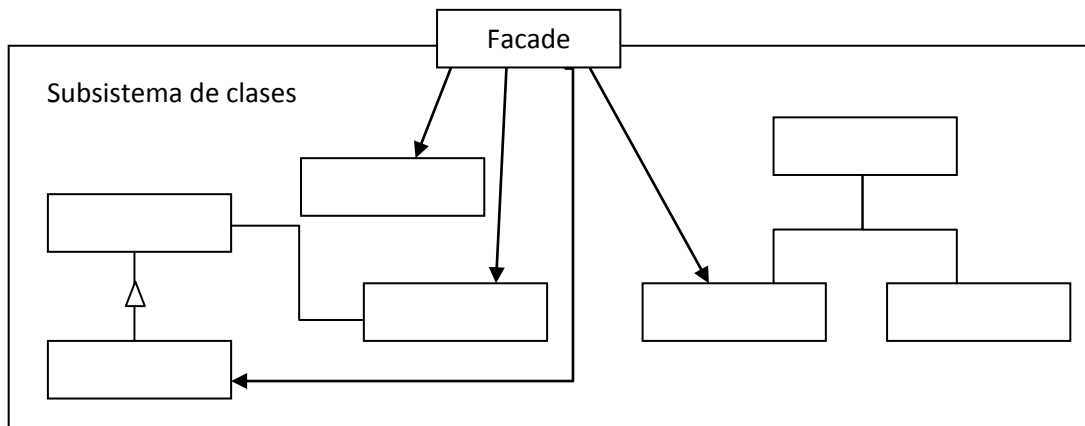


Figura 3.19 Estructura del patrón Facade

Facade: Es la clase que aísla la lógica compleja de los subsistemas, del resto de la aplicación.

Subsistema de clases: Son las clases que implementan la funcionalidad del sistema.

En la solución que se propone, se puede apreciar la utilización de este patrón por medio de la clase que lleva su mismo nombre, en la cual se encuentran todas las funcionalidades de cada uno de los módulos que conforman los nuevos productos de la Colección Multisaber. Dicha clase, es la que se encarga de distribuir el trabajo, en dependencia de las peticiones que le lleguen de la capa de presentación. Cada módulo, tiene asociado a su vez una clase Fachada, que tiene la finalidad de servir de interfaz con respecto a todas las funcionalidades que este encapsula. Por tanto, cuando a la clase Fachada más general le llega una petición relacionada con el módulo “Resultados”, esta se la transfiere a su correspondiente clase Fachada, la cual a su vez se la entrega a las clases encargadas de procesar la petición solicitada.

“Data Access Object” (Objeto de Acceso a Datos, en lo adelante DAO por sus siglas en inglés): El problema que resuelve este patrón es el de contar con diversas fuentes de datos (base de datos, archivos, servicios externos, entre otros). Permite que se encapsule la forma de acceder a la fuente de datos. Surge de la necesidad de gestionar una diversidad de fuentes de datos, aunque su uso se extiende al problema de encapsular no sólo la fuente de datos, sino además, ocultar la forma de acceder a los datos. Se trata de que el software cliente se centre en los datos que necesita y se olvide de cómo se realiza el acceso a los mismos o de cuál es la fuente de almacenamiento.



Estructura del patrón DAO:

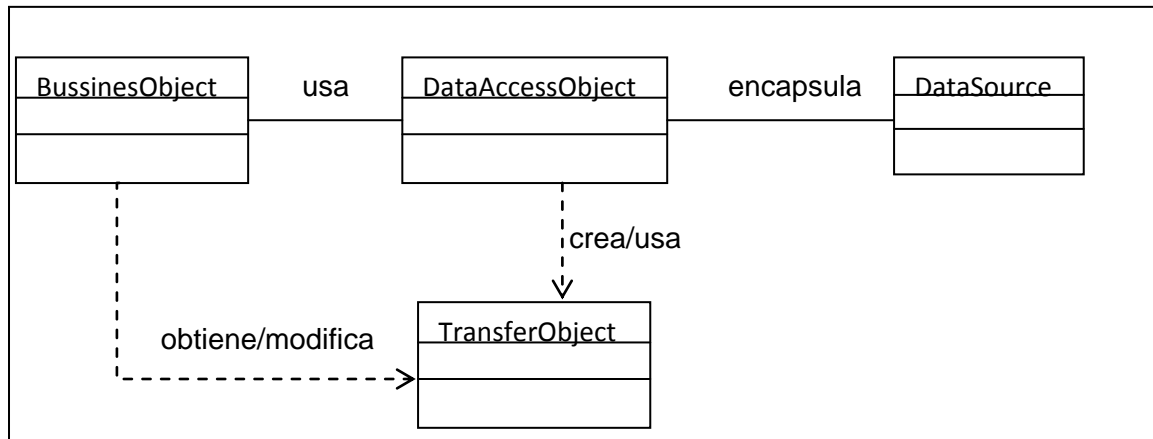


Figura 3.20 Estructura del patrón DAO

BusinessObject: Es el objeto que quiere acceder a la fuente de datos para poder almacenar o consultar datos.

DataAccessObject: Abstrae al BusinessObject de los detalles del acceso a la fuente de datos.

DataSource: Representa la implementación de la fuente de datos en sí.

TransferObject: Es un objeto intermedio entre el BusinessObject y el DataAccessObject.

En la solución que se propone se puede apreciar la utilización de este patrón, por medio de las clases cuyos nombres terminan con “DAO”. Las mismas tienen la finalidad de comunicarse con la fuente de datos, a fin de solicitarle la información que necesita la capa lógica del negocio. Esta operación se realiza de manera transparente a las clases que posteriormente procesan todos los datos, que en este caso son las que comienzan con “Manager”.

“Singleton” (Solitario): Dentro de la clasificación GoF, es un patrón creacional que garantiza que una clase sólo tenga una única instancia, proporcionando un punto de acceso global a la misma.

Algunas de sus características son:

- El acceso a la instancia única está más controlado. Puede tener un control estricto sobre cómo y cuando acceden los clientes a la instancia.
- Permite el refinamiento de operaciones y la representación. Se puede crear una subclase de Singleton.



- Permite un número variable de instancias.

Estructura de Singleton:

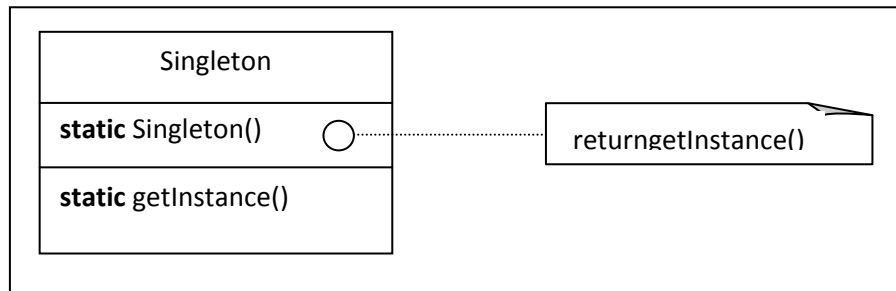


Figura 3.21 Estructura del patrón Singleton

El Singleton es la clase de la cual se tendrá solo una instancia a lo largo de toda la aplicación.

La manera en la cual está implícita la utilización de este patrón en la solución que se propone, es en la clase que lleva por nombre: "Acceso_Datos", con el objetivo de que exista sólo una instancia de la misma, a fin de proporcionar un punto común de acceso a la fuente de datos, y con ello evitar la sobrecarga de conexiones a una misma Base de Datos.

3.5 Diseño de la base de datos

El diseño de la base de datos juega un papel fundamental para cualquier aplicación que gestione información, sobre todo para las aplicaciones Web; es por ello que se hace necesario que los datos se almacenen de forma coherente y organizada, para evitar que dicha información se pierda. El principal objetivo para realizar el diseño de la base de datos en cuestión, es lograr la persistencia de los datos de los usuarios, así como la información de los recursos que se gestionan por los mismos, controlando que esta información no tenga redundancia y no existan datos innecesarios que no cumplen ningún objetivo al ser guardados.

3.5.1 Diagramas de clases persistentes

Las clases persistentes son aquellas que necesitan ser capaz de guardar su estado en un medio permanente; lo cual está dado por el almacenamiento físico y permanente de la información de la clase, para la copia de seguridad en caso del fracaso del sistema, o para el intercambio de información.



A continuación se muestra el diagrama de clases persistentes del sistema:

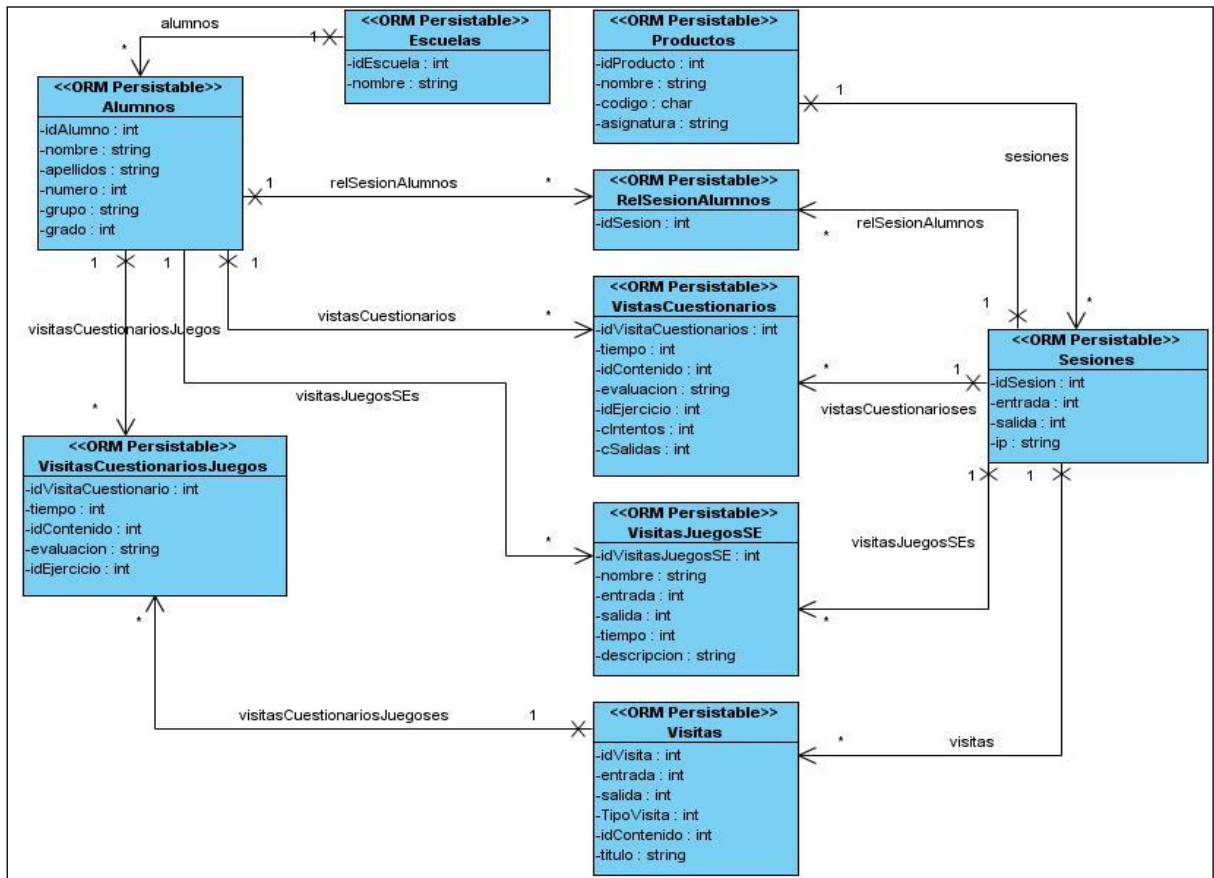


Figura 3.22 Diagrama de clases persistentes

3.5.2 Modelo de Datos

Es una serie de conceptos que puede utilizarse para describir un conjunto de datos y las operaciones para manipularlos. Hay dos tipos de modelos de datos: los modelos conceptuales y los modelos lógicos. Los modelos conceptuales se utilizan para representar la realidad a un alto nivel de abstracción. Mediante los modelos conceptuales se puede construir una descripción de la realidad que sea fácil de entender. En los modelos lógicos, las descripciones de los datos tienen una correspondencia sencilla con la estructura física de la base de datos.



A partir del diagrama de clases persistentes se obtuvo el modelo de datos que se muestra a continuación:

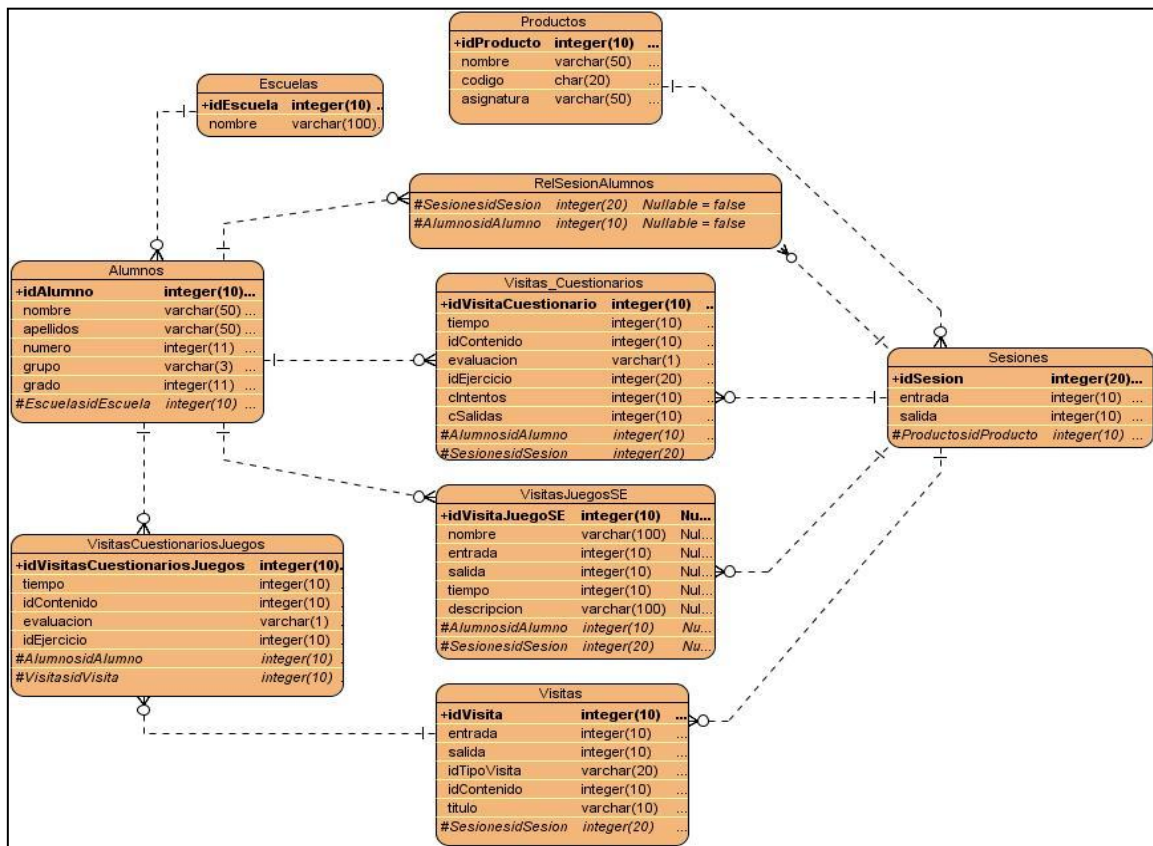


Figura 3.23 Modelo de Datos

3.6 Conclusiones

En este capítulo se representaron los diferentes elementos que ilustran cómo está construido el sistema, en términos de clases del análisis y del diseño. Este último permitió comprender la lógica del sistema en general. Se realizaron los diferentes diagramas de clases del diseño Web donde se representaron las clases y sus asociaciones. Se mencionaron los patrones de diseño utilizados que evitaban la reiteración de la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente. Por último se presentó el diseño de la base de datos, creando para ello el diagrama de clases persistentes y a partir de este, el modelo de datos.



Capítulo Implementación



4.1 Introducción

En el presente capítulo se modelan los artefactos que se recogen a partir del flujo de trabajo de implementación, es decir, se describe cómo los elementos del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

4.2 Modelo de Implementación

Este modelo describe cómo los elementos del diseño, como las clases, se implementan en términos de componentes, como son: ficheros fuentes, binarios, ejecutables y otros. Además de detallar cómo están organizados los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado, así como la dependencia que existe entre estos componentes.

4.2.1 Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes de hardware y de software, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software (procesos y objetos que se ejecutan en ellos). Se puede decir además que, un diagrama de despliegue es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.

A continuación se muestran cómo están distribuidos los nodos y dispositivos para la explotación del producto.

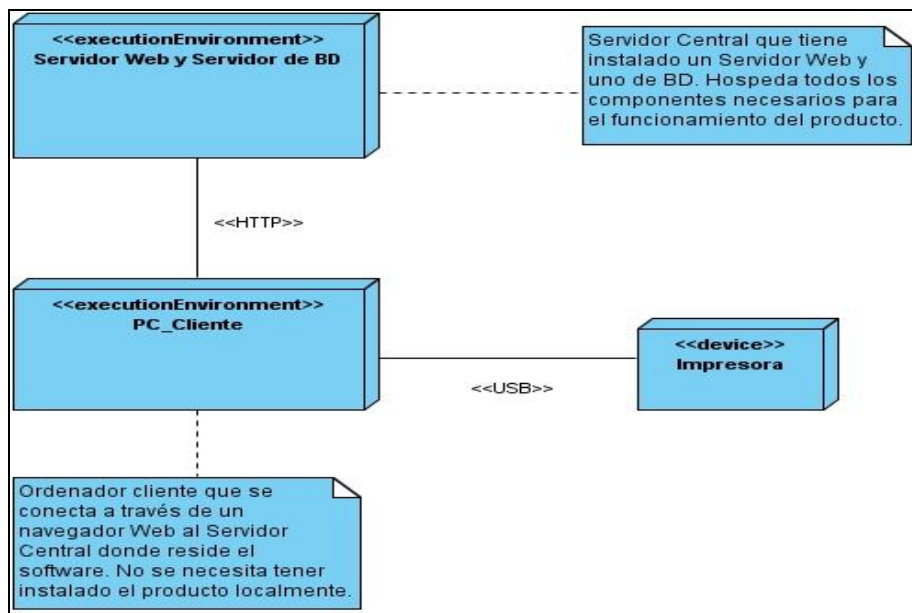


Figura 4.1 Diagrama de despliegue

4.2.2 Diagrama de componentes

El diagrama de componentes incluye a los componentes y archivos que se utilizan para ensamblar y hacer disponible el sistema físico. Se representa como un grafo de componentes de software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten.

4.2.2.1 Diagrama de componentes general

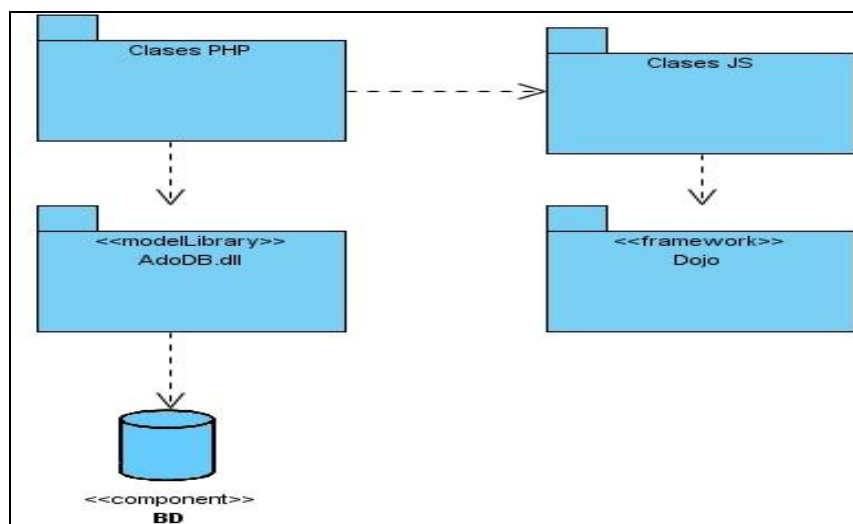


Figura 4.2 Diagrama de componentes general

4.2.2.2 Diagrama de componentes del paquete Clases JS

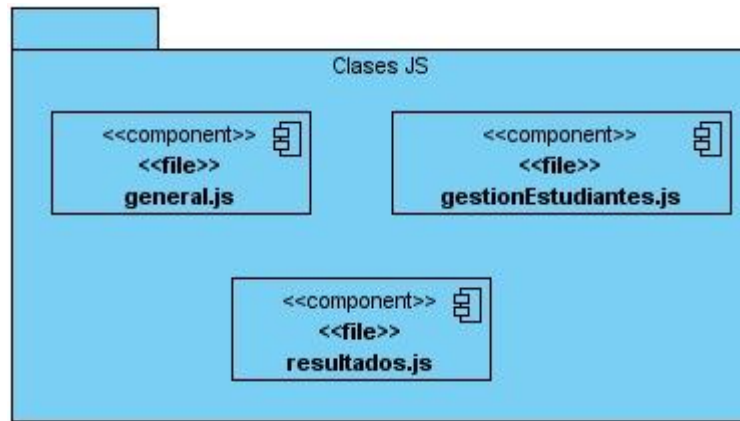


Figura 4.3 Diagrama de componentes del paquete Clases JS

4.2.2.3 Diagrama de componentes del paquete Clases PHP

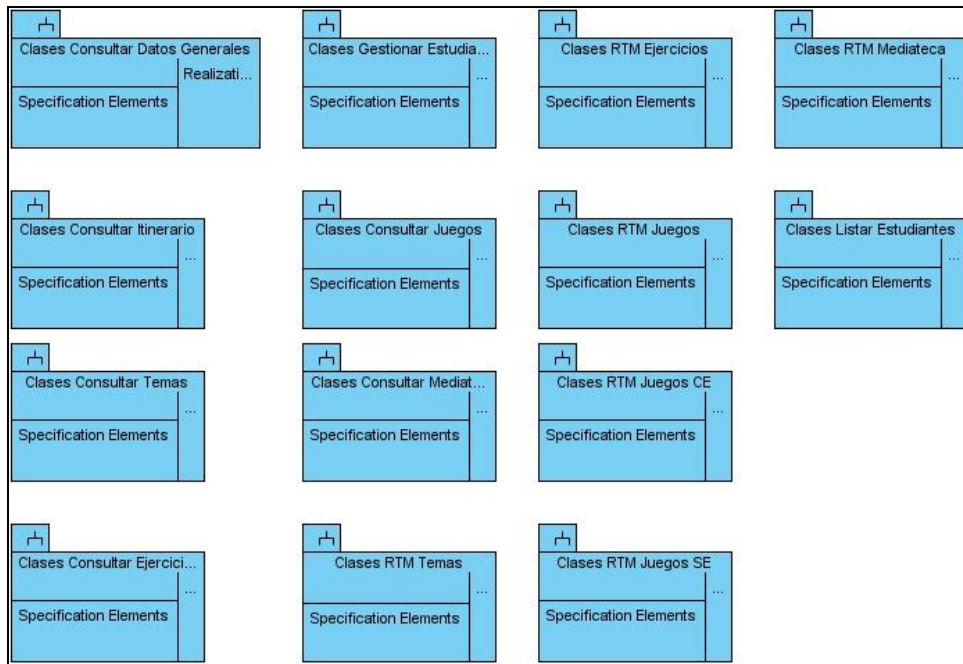


Figura 4.4 Diagrama de componentes del paquete Clases PHP

El resto de los diagramas de componentes se pueden encontrar en el *Anexo 4*.

4.3 Conclusiones

En este capítulo fue mostrado cómo puede ser distribuido el producto a través de la representación del diagrama de despliegue, ilustrando cuáles serán los nodos que se usarán para la implantación de la aplicación. Además fue presentado cómo está construido el sistema a partir de los diagramas de componentes, describiendo cada uno de ellos un apartado del sistema.



Conclusiones

Al término de esta investigación, se pudo arribar a las siguientes conclusiones:

- Se realizó un estudio de las tendencias y tecnologías actuales más utilizadas, a fines con la temática que se aborda, seleccionando y fundamentando aquellas que más se adecuaban al entorno donde se iban a aplicar, teniendo en cuenta siempre que fueran, en su mayoría, herramientas de código abierto y libres de cualquier tipo de restricción. Posibilitando que se obtuviera un módulo que se pudiera integrar sin muchas dificultades, a los restantes que conforman los nuevos productos de la Colección Multisaber, en su versión multiplataforma.
- Mediante el análisis, diseño e implementación del módulo “Resultados” perteneciente a los nuevos productos que conforman la Colección Multisaber, en su versión multiplataforma, fue posible contar con la información que se necesita de una manera más organizada, posibilitando que el profesor pueda emitir un mejor criterio acerca del desempeño de sus estudiantes, durante la interacción de estos con los restantes módulos de la aplicación.
- Quedó recogido en un documento todo el proceso de desarrollo de este módulo, en el que se explica en detalles cómo fue construido el mismo, a partir de los diferentes artefactos que propone la metodología de desarrollo RUP, en cada uno de sus flujos de trabajo, brindando con ello la posibilidad de contar con un material de apoyo para el desarrollo de posteriores versiones.



Recomendaciones

- Mejorar el diseño de la interfaz gráfica de usuario, buscando con ello una homogeneidad en todo el producto.
- Desarrollar el flujo de trabajo “Pruebas” de la metodología de desarrollo RUP.
- Implementar cierto nivel de seguridad, que impida que un estudiante pueda autenticarse haciendo uso de los datos personales de otro.
- Incorporar este módulo en los restantes productos de la Colección Multisaber.



Referencias Bibliográficas

1. Morales, V., *Planeamiento y análisis de investigaciones*. 1992: Caracas, Venezuela.
2. *Definición de lenguaje de programación*. [cited 2010 02/02/10]; Available from: <http://www.definicion.org/lenguaje-de-programacion>.
3. Achour, M., et al., *Información General*, in *PHP Manual*, G. Hojtsy, Editor. 2007, 05/05/07.
4. Hinostroza, R.R. *Usuarios y Grupos en Linux*. 2007 [cited 2010 17/02/10]; 22/02/2007:[Available from: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
5. José Márquez Díaz, L.S., Félix Vargas (2002) *Instalación y configuración de Apache, un servidor Web gratis*. Volume, 23p
6. Rosell, F.A.C., *Sistema de Gestión de Información de la Facultad 8. Módulo para la Gestión de la Residencia Estudiantil*. 2007, Universidad de las Ciencias Informáticas: Ciudad de La Habana. p. 127.
7. Kennard, J., *Mastering JOOMLA 1.5 Extension and Framework Development*. 2007: noviembre 2007. 486p.
8. Ayllapan, W.U. *Ventajas y características de JOOMLA CMS*. 2009 [cited 2010 16/02/10]; Available from: <http://www.joomlaos.net/caracteristicas-de-JOOMLA.php>.
9. Zaclive. 2009 [cited 2010 15 de febrero]; Available from: <http://www.svdescargas.info/2010/01/descarga-netbeans-67.html>.
10. Romo, A.R.V. 2009 [cited 2010 16 de febrero]; Available from: http://blogs.sun.com/AlanVargas/entry/netbeans_6_7.
11. Aguilar, C.A. *Sun libera su IDE NetBeans 6.7 para Java y PHP*. 2009 [cited 2010 16 de febrero]; Available from: <http://osum.sun.com/profiles/blogs/sun-libera-su-ide-netbeans-67>.
12. 2010 [cited; Available from: <http://www.mysql.com/why-mysql/>.
13. Batista, A.L. and D.G. Mora, *Sistema de Gestión de Información de la Facultad 8. Diseño de la Base de Datos*. 2007, Universidad de las Ciencias Informáticas: La Habana. p. 134.
14. *EMS SQL Manager for MySQL Professional for Linux 1.4.1.5*. 1998-2010 [cited; Available from: <http://fileforum.betanews.com/detail/EMS-SQL-Manager-for-MySQL-Professional-for-Linux/1007472749/2>.
15. Tavárez, D. *Comparación de Frameworks en Javascript*. [cited 2010 18 de febrero]; Available from: <http://www.maestrosdelweb.com/editorial/comparacion-frameworks-javascript/>.
16. Daniela. *Top 5 Javascript Frameworks*. 2007 [cited 2010 18 de febrero]; Available from: <http://www.elwebmaster.com/articulos/top-5-javascript-frameworks>.



17. Gill, R., C. Riecke, and A. Russell, *Mastering Dojo JavaScript and AJAX Tools for Great Web Experiences*. 2008. 545.
18. Schmuller, J., *APRENDIENDO UML EN 24 HORAS*. Editorial Mexicana Registro No. 1031 ed. 2000, México. 448.
19. Orallo, E.H. *El lenguaje Unificado de Modelado (UML)*. Volume, 74
20. *Paradigma Visual para UML*.
21. *VISUAL PARADIGM FOR UML 6.0*.
22. Sánchez, C., *Aplicaciones en capas Anterior Capítulo 3. Justificación*. 2004: Universidade da Coruña.



Bibliografía

Gill, R. (2008). Mastering Dojo.

Harmon, J. E. (2009). Dojo, Using the Dojo JavaScript Library. México.

Park, T. C. (2004). PHP y MYSQL Bible. Canadá.

Rusell, M. A. (2008). Dojo The Definitive Guide.

Schmuller, J. (2000). APRENDIENDO UML EN 24 HORAS. México.

Zammetti, F. W. (2008). Practical Dojo Projects.

Larman Craig, Philippe Kruchten. UML y Patrones. 2. Canadá : s.n. pág. 520.

Paradigma Visual para UML.

VISUAL PARADIGM FOR UML 6.0.

Kennard, J., Mastering JOOMLA 1.5 Extension and Framework Development. 2007: Noviembre 2007.

486p.

Orallo, E.H. El lenguaje Unificado de Modelado (UML). Volume, 74

Paradigma Visual para UML.

Sánchez, C., Aplicaciones en capas Anterior Capítulo 3. Justificación. 2004: Universidade da Coruña.

José Márquez Díaz, L.S., Félix Vargas (2002) Instalación y configuración de Apache, un servidor Web gratis.

Volume, 23p

Stefanov, Stoyan. 2008. Object-Oriented JavaScript. s.l. : Packt Publishing Ltd, 2008. pág. 354.



Glosario de Términos y Siglas

- **AJAX**: *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.
- **API**: Una interfaz de programación de aplicaciones o API (del inglés application programming interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Usados generalmente en las bibliotecas.
- **Auditabilidad**: En Informática es la revisión y evaluación de los controles, sistemas, procedimientos de informática, de los equipos de cómputo, su utilización, eficiencia y seguridad, de la organización que participan en el procesamiento de la información, a fin de que por medio del señalamiento de cursos alternativos se logre una utilización más eficiente y segura de la información que servirá para la adecuada toma de decisiones.
- **CASE**: *Computer Aided Software Engineering*. Ingeniería de software asistida por computadora
- **CGI**: *Common Gateway Interface*. En sí, es un método para la transmisión de información hacia un compilador instalado en el servidor. Su función principal es la de añadir una mayor interacción a los documentos Web que por medio del HTML se presentan de forma estática.
- **CMS**: *Content Management System*. Permite la creación y administración de contenidos principalmente en páginas Web.
- **DOM**: *Document Object Model*. Modelo de Objetos de Documento.
- **Encapsulamiento**: Mecanismo que se utiliza para ocultar los datos, la estructura interna y los detalles de implementación de algunos elementos, como un objeto o un subsistema. Todas las interacciones con un objeto se realizan a través de una interfaz pública de operaciones.
- **GUI**: *Graphical User Interface*. Sistema de interacción entre el ordenador y el usuario, caracterizado por la utilización de iconos y elementos gráficos en su concepción. Es un



paso más allá de los interfaces basados en caracteres, que sólo incluían líneas de texto para introducir comandos y conocer las respuestas del sistema.

- HTML: *Hypertext Markup Language*. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.
- HTTP: *HyperText Transfer Protocol*. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.
- Herramientas CASE: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.
- Instancia: Un miembro individual de una clase. En UML se denomina objeto.
- JavaScript: Es un lenguaje de script multiplataforma basado en objetos.
- JOOMLA: Es un sistema de administración de contenidos de código abierto construido con PHP bajo una licencia GPL. Este administrador de contenidos se usa para publicar en Internet e intranets utilizando una base de datos MySQL.
- JSON: (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos, es decir, es un formato de texto completamente independiente del lenguaje.
- Mantenibilidad: Es el proceso de modificar un componente o sistema de software después de su entrega para corregir fallos, mejorar su funcionamiento u otros atributos, o adaptarlo a cambios en el entorno.
- MySQL: Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.
- Oracle: Es una potente herramienta cliente/servidor para la gestión de bases de datos.



- PHP: *Hypertext Preprocessor*. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.
- PostgreSQL: Es un Sistema de Gestión de Bases de Datos de código abierto.
- Release: Este término se utiliza a veces de manera informal por muchas empresas de desarrollo, para describir el estado de un producto; el cuál debe estar preparado para su entrega a la comunidad de usuarios, a menos que aparezcan errores que lo impida.
- Registrar traza del módulo Juegos CE: Identificación del nombre del caso de uso “Registrar traza del módulo Juegos con ejercicios”.
- Registrar traza del módulo juegos SE: Identificación del nombre del caso de uso “Registrar traza del módulo Juegos sin ejercicios”
- RUP: *Rational Unified Process* (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.
- SGBD: *Sistema de Gestión de Bases de Datos*. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.
- Software: Programas de sistema, utilerías o aplicaciones expresadas en un lenguaje de máquina.
- SQL: *Structured Query Language*. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.
- TICs: Tecnologías de la Información y las Comunicaciones.
- Transaccionalidad: Es un grupo de instrucciones que se ejecutan con éxito en su totalidad, o no se ejecutan ninguna.
- UML: *Unified Modeling Language*. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

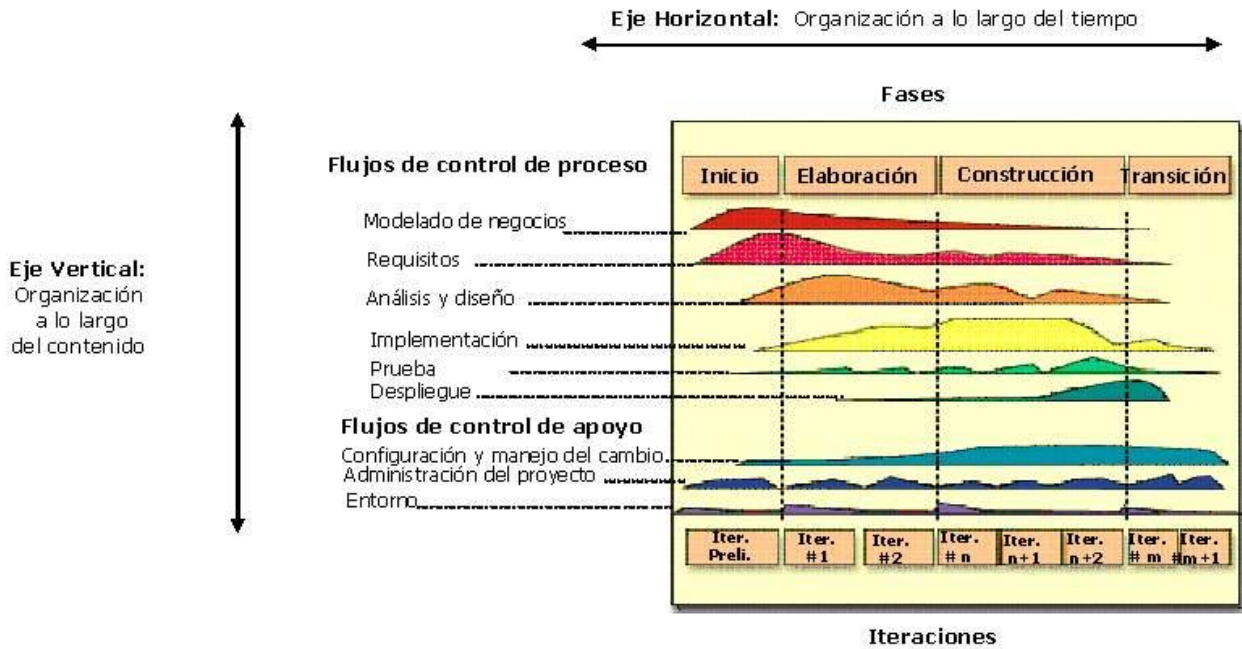


- URL: *Uniform Resource Locator*, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.
- XML: *Extensible Markup Language*. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.

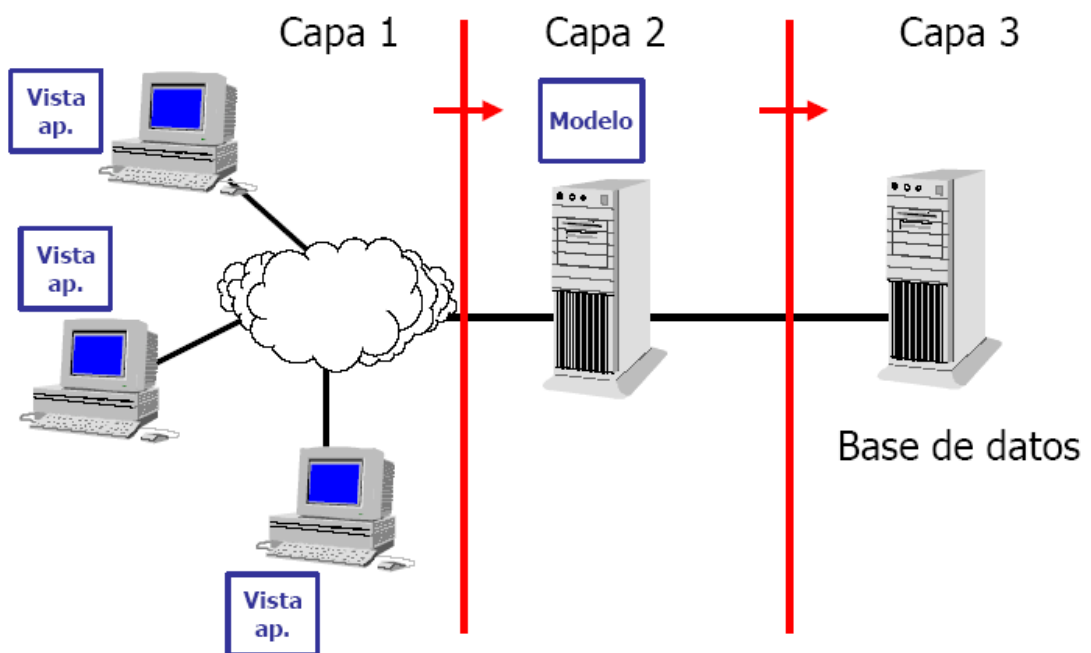


Anexos

Anexo 1: Fases y Flujos de trabajo de RUP



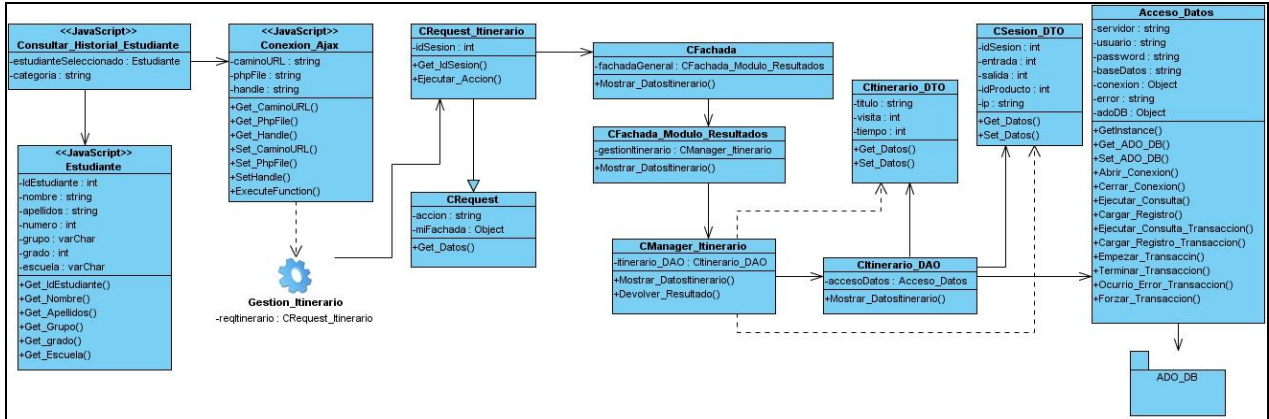
Anexo 2: Modelo de diseño en 3 capas



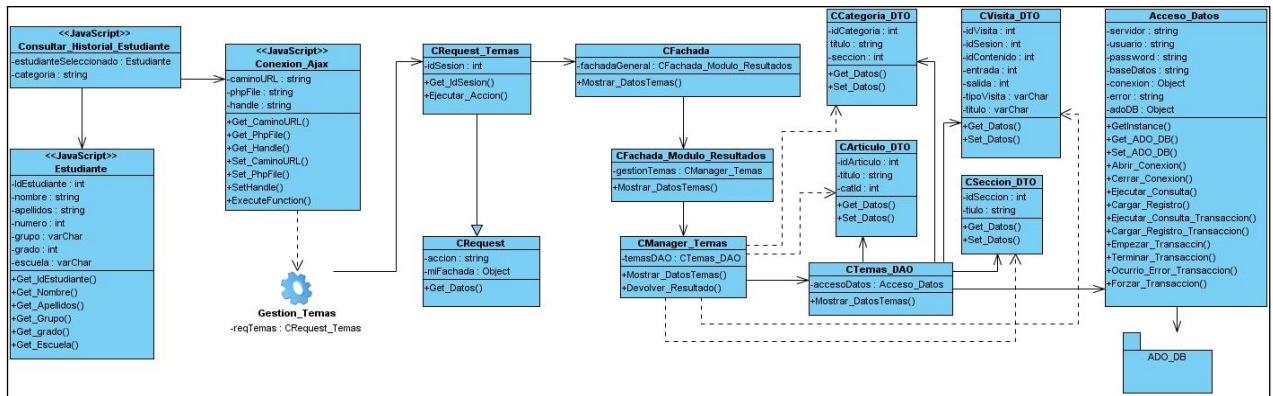


Anexo 3: Diagramas de clases web correspondientes al resto de los escenarios del caso de uso "Consultar historial de estudiantes"

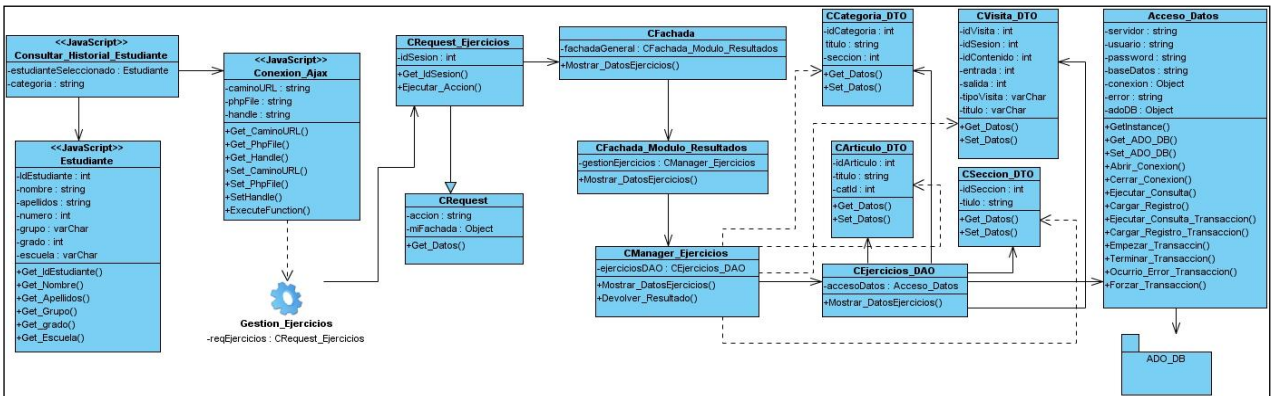
Escenario Consultar Itinerario



Escenario Consultar Temas

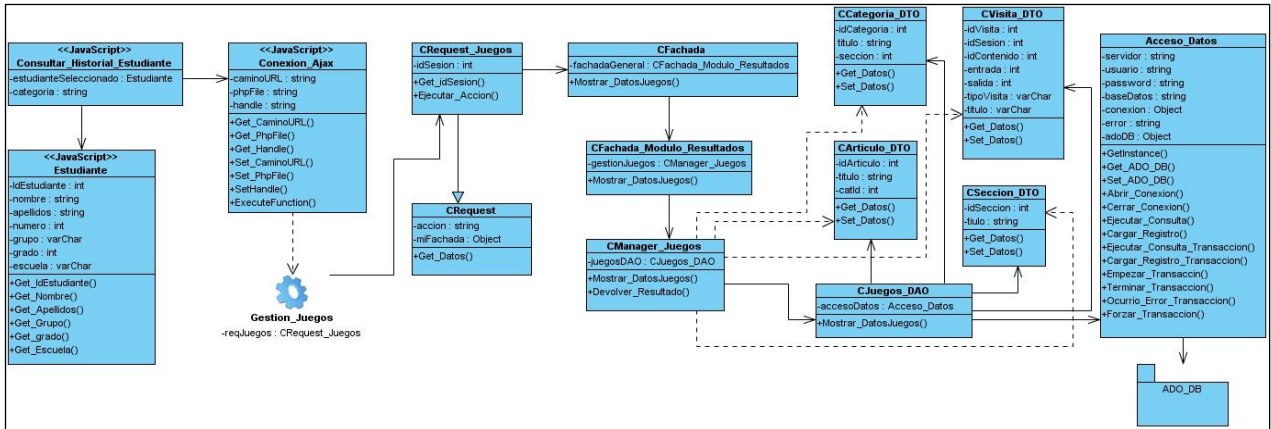


Escenario Consultar Ejercicios

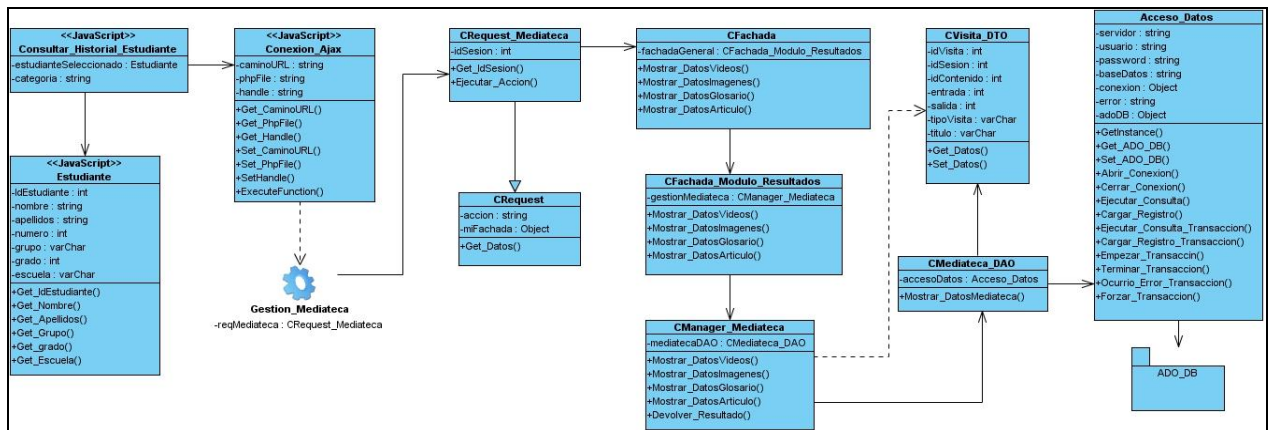




Escenario Consultar Juegos

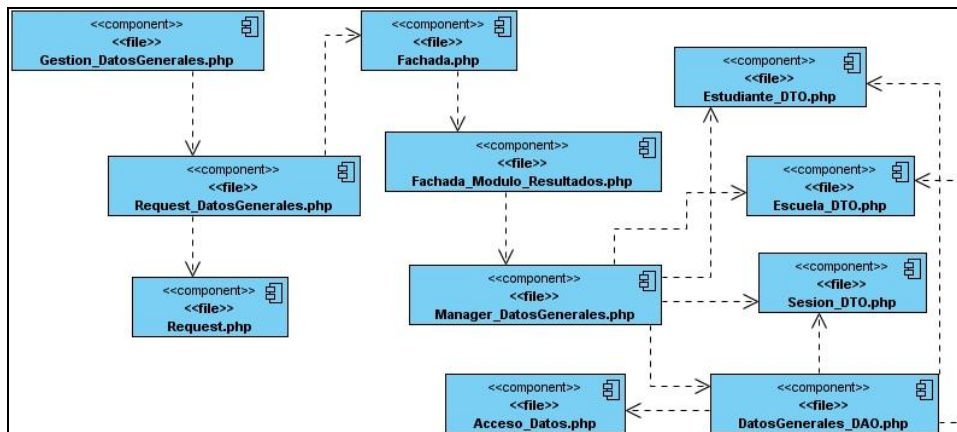


Escenario Consultar Mediateca



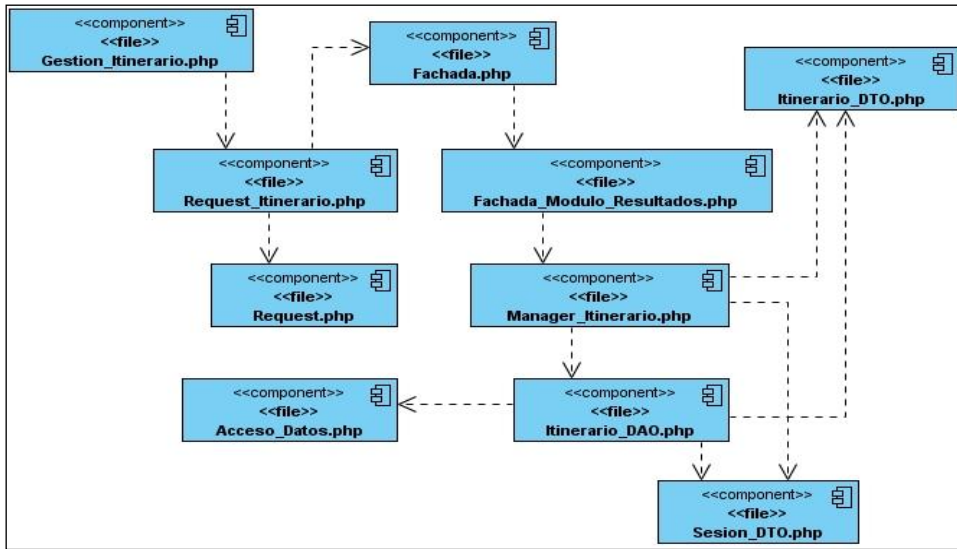
Anexo 4: Diagramas de componentes correspondientes al resto de los subsistemas del paquete Clases PHP

Subsistema Clases Consultar Datos Generales

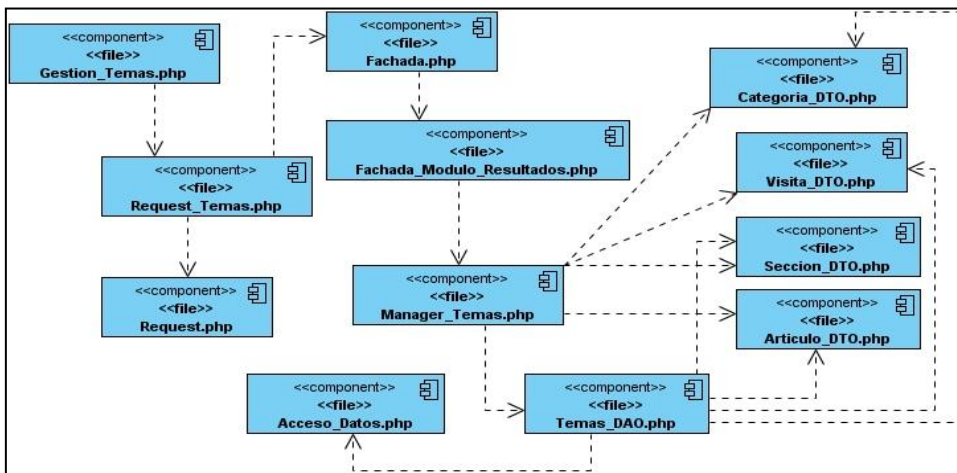




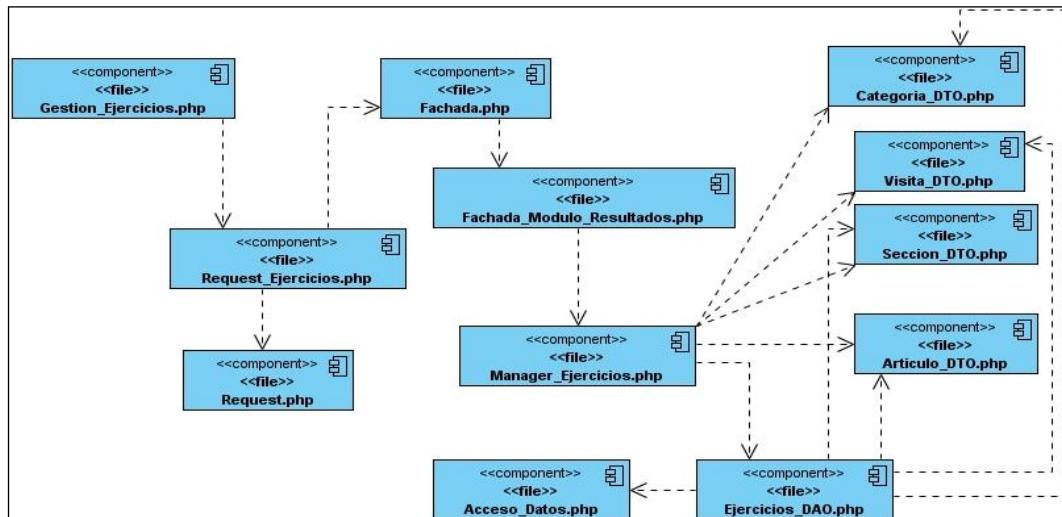
Subsistema Clases Consultar Itinerario



Subsistema Clases Consultar Temas

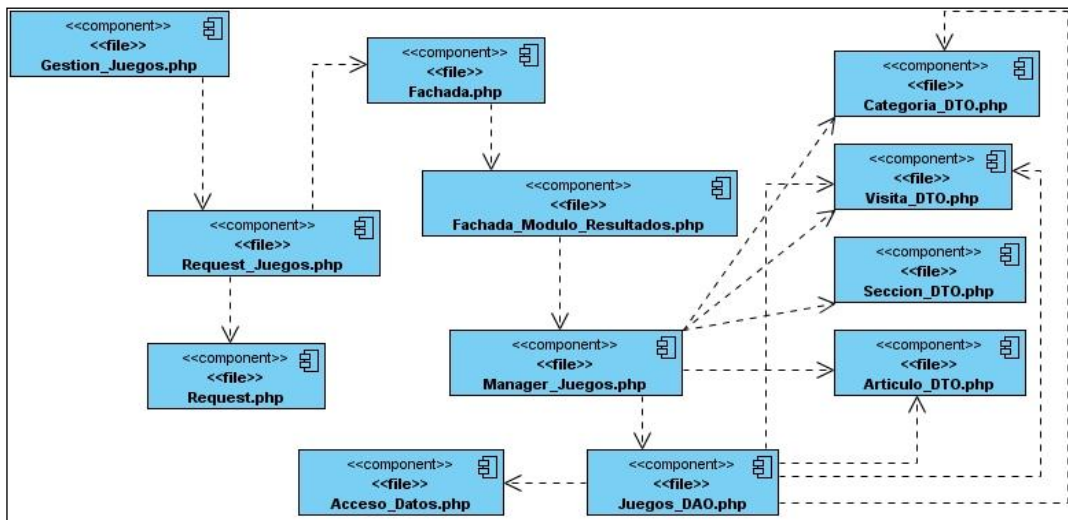


Subsistema Clases Consultar Ejercicios

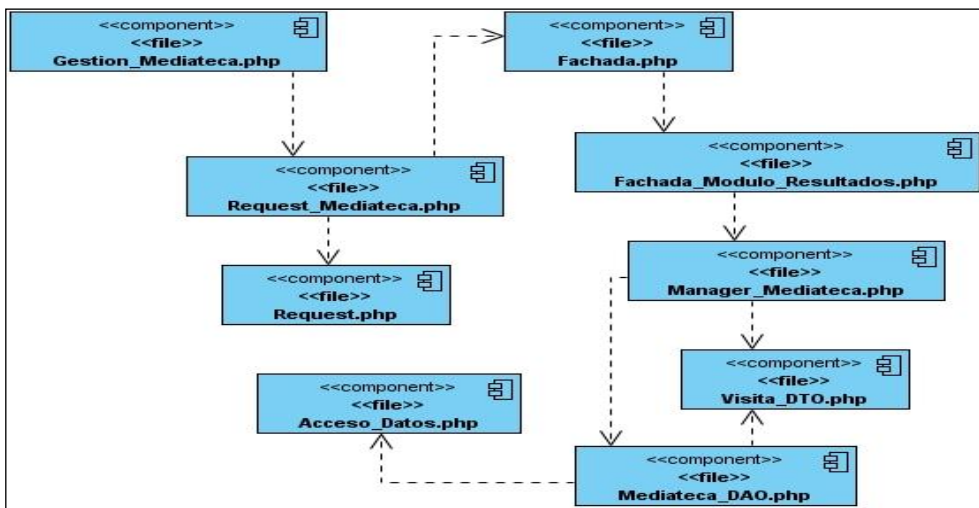




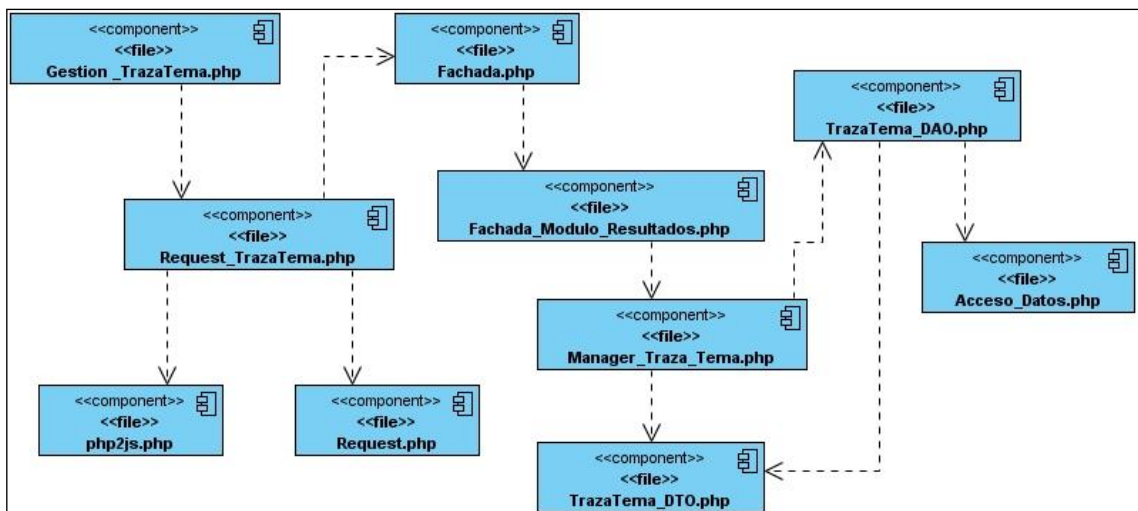
Subsistema Clases Consultar Juegos



Subsistema Clases Consultar Mediateca

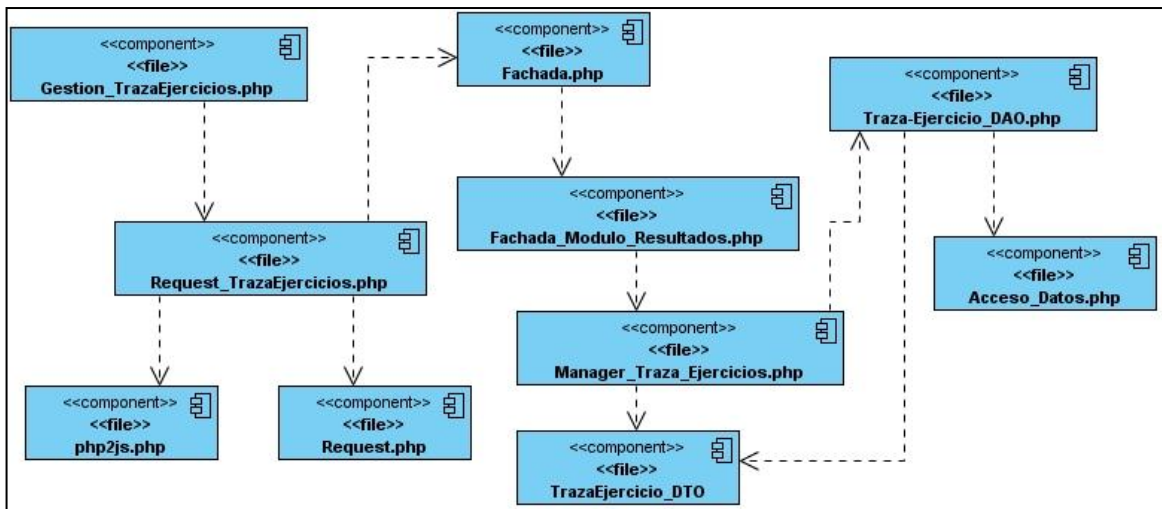


Subsistema Clases RTM Temas

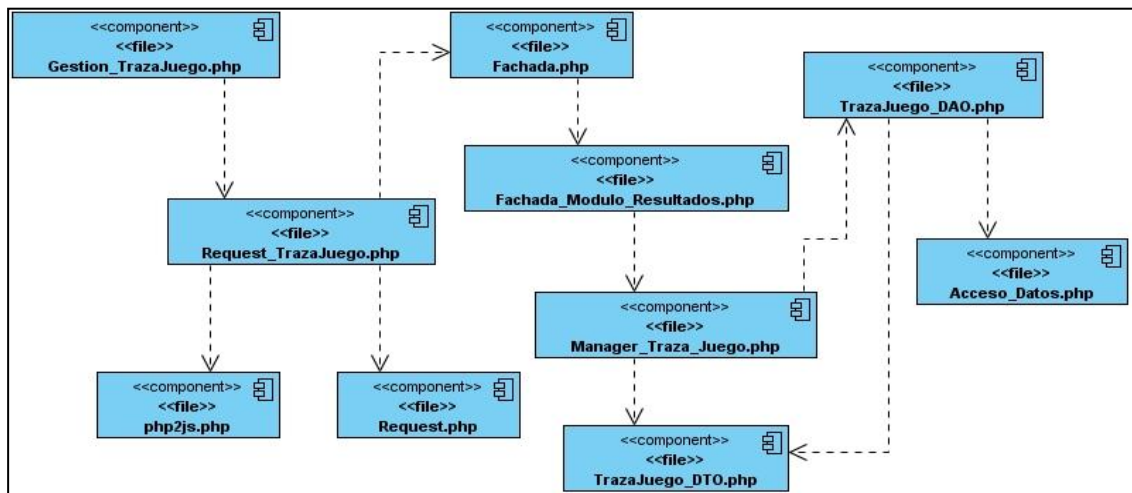




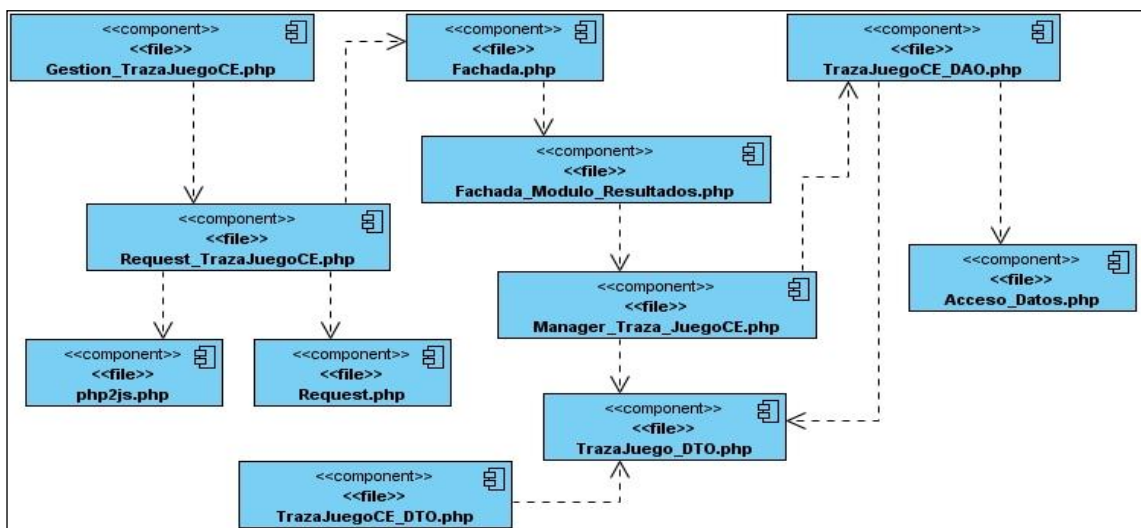
Subsistema Clases RTM Ejercicios



Subsistema Clases RTM Juegos

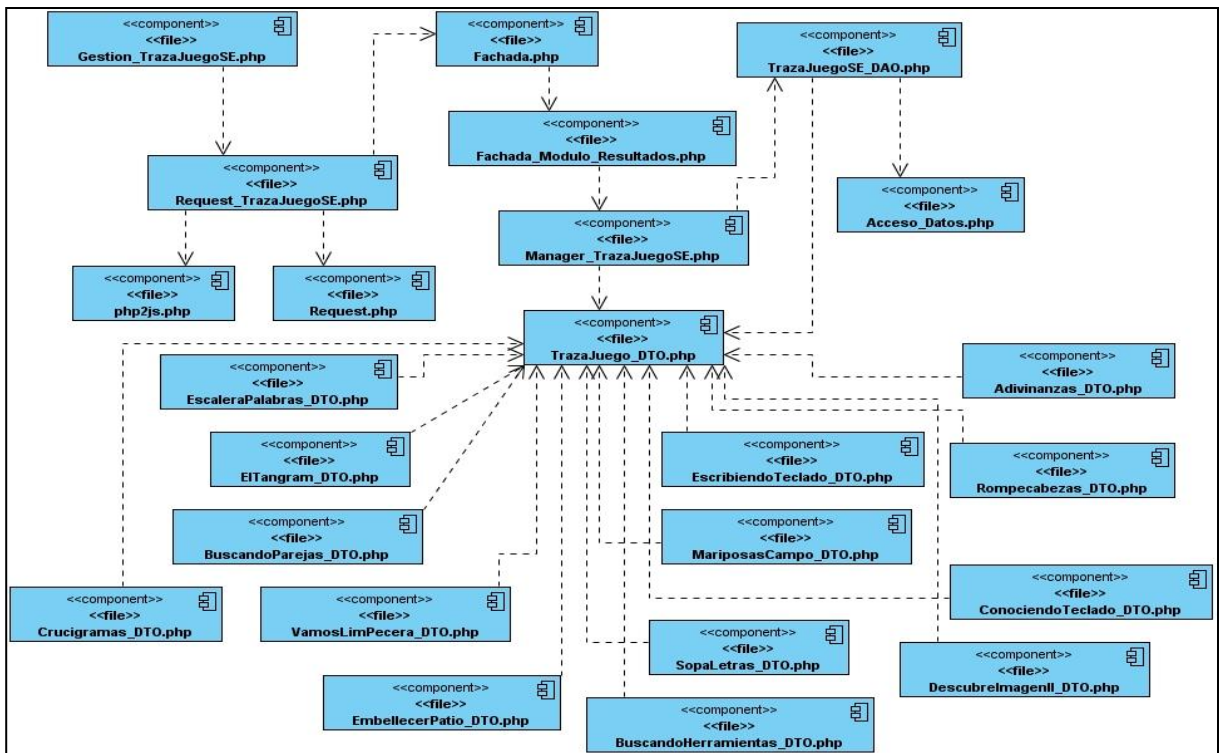


Subsistema Clases RTM Juegos CE

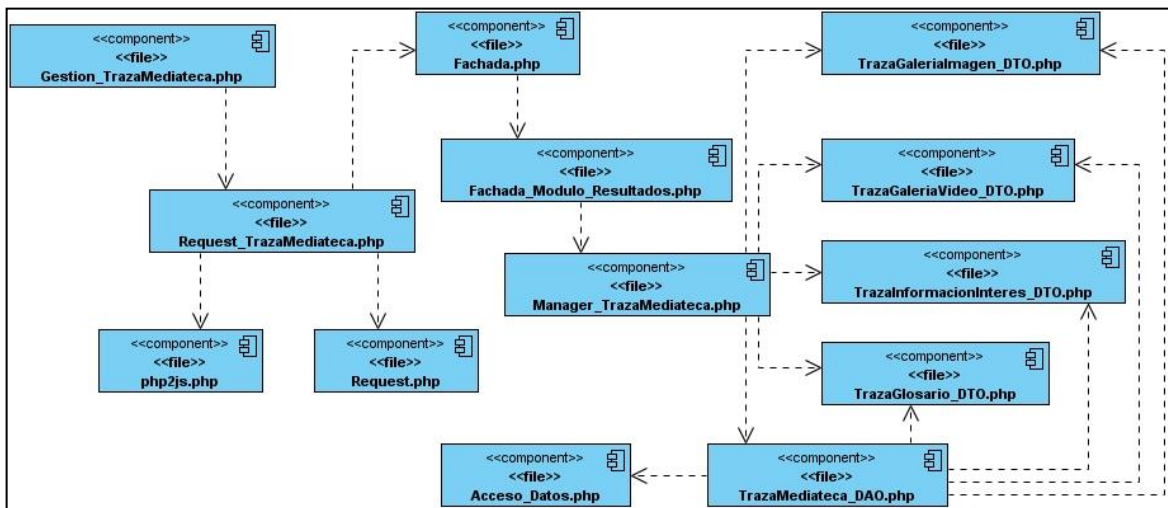




Subsistema Clases RTM Juegos SE

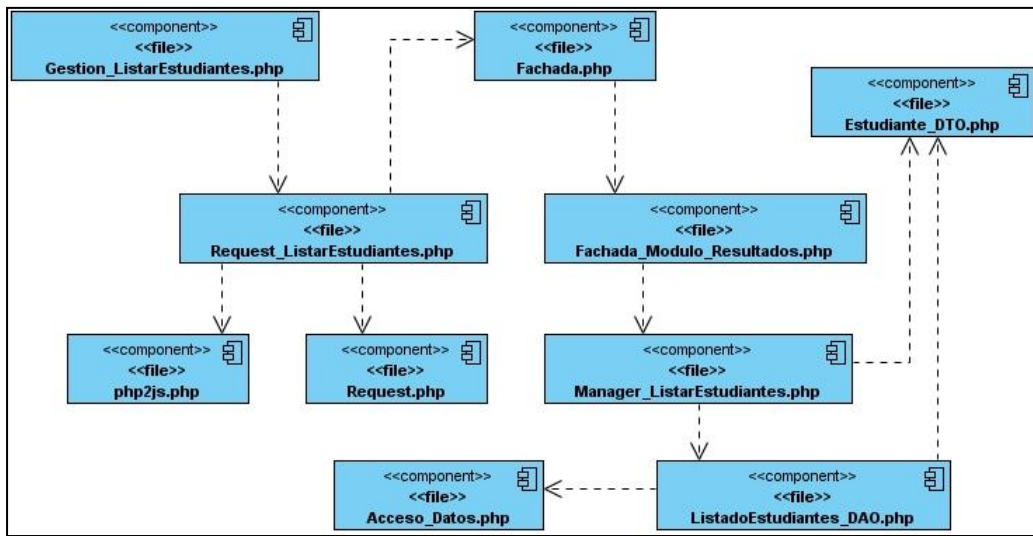


Subsistema Clases RTM Mediateca





Subsistema Clases Listar Estudiantes



Subsistema Clases Gestionar Estudiantes

