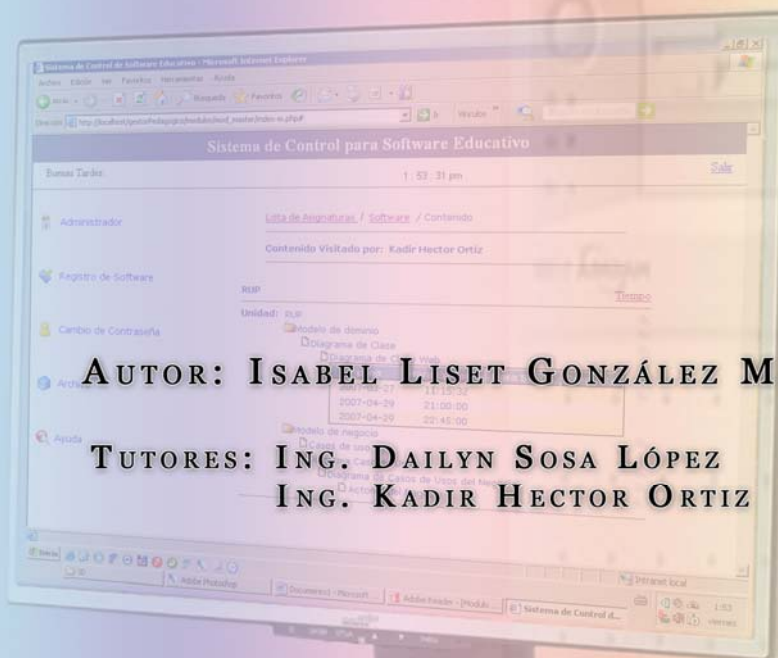


UNIVERSIDAD DE CENFUEGOS
"CARLOS RAFAEL RODRÍGUEZ"
FACULTAD DE INFORMÁTICA

TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERO EN INFORMÁTICA

TÍTULO: SISTEMA DE CONTROL DEL USO DE SOFTWARES EDUCATIVOS



AUTOR: ISABEL LISET GONZÁLEZ MONZÓN

**TUTORES: ING. DAILYN SOSA LÓPEZ
ING. KADIR HECTOR ORTIZ**

- 2007 -

"AÑO 49 DE LA REVOLUCIÓN"

*“Lo que hace crecer el mundo
no es el descubrir cómo está hecho,
sino el esfuerzo de cada uno para descubrirlo.”*

José Martí

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Informática de la Facultad de Informática en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que hagan el uso que estimen pertinente con el trabajo de diploma.

Para que así conste firmo la presente a los ____ días del mes de ____ del ____.

Isabel Liset González Monzón

Nombre completo del primer autor

Ing. Dailyn Sosa López

Nombre completo del primer tutor

Ing. Kadir Hector Ortiz

Nombre completo del segundo tutor

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

Ing. Dailyn Sosa López

Firma Tutor

Firma ICT

Ing. Kadir Hector Ortiz

Firma Tutor

Firma Vicedecano

opinión del tutor

*A todos los que piensan con amor y viven con el corazón; a ti abuela aunque no
podamos disfrutarlo juntas,*

a ti madre, por dedicar todo tu sacrificio a mi educación,

a ti mi esposo, por tu amor,

a mi papá, mi hermano y a todos, por apoyarme e

incitarme a continuar.

A mis tutores Dailyn y Kadir por su apoyo y amistad día tras día, a mis profesores que fueron cómplices del logro de este resultado científico, mi tesis de graduación.

Por su constancia, amor, compañerismo y solidaridad agradezco a todos mis compañeros de aula por haber sabido estar en cada circunstancia de estos largos y difíciles cinco años.

A todos los que confiaron en mí,

Gracias

Índice

Introducción.....	1
Capítulo I: “Fundamentación Teórica”	5
I.1 Tecnologías de la Información y las Comunicaciones (TIC)	5
I.1.1 Software Educativo	5
I.2 Descripción de los sistemas existentes	9
I.3 Metodologías Utilizadas	10
I.3.1 Lenguaje de Modelación Unificado (UML)	10
I.3.2 Proceso Unificado de Desarrollo (RUP)	11
I.4 Paradigmas de programación.	12
I.4.1 Programación por capas.	12
I.5 Tecnologías a utilizar	14
I.5.1 Interfaz de usuario (Capa de Presentación).....	14
I.5.2 Lógica del negocio (Capa de la Lógica del Negocio)	15
I.5.2.1 Lenguajes de programación web	15
I.5.2.2 Servicios web	17
I.5.2.3 XML.....	18
I.5.2.4 SOAP	21
I.5.2.5 Servidores de Aplicaciones Web.....	21
I.5.3 Capa de acceso a datos.	22
Conclusiones	23
Capítulo II: “Descripción del problema”	25
II.1 Descripción del modelo del dominio.	25
II.1.1 Definición de los objetos y los conceptos principales.....	25
II.1.2 Representación del Modelo de Objetos del Dominio.	26
II.1.3 Reglas del negocio a considerar.	26
II.2 Descripción del sistema propuesto.	27
II.2.1 Concepción general del sistema.	27
II.2.2 Requerimientos Funcionales.....	27
II.2.3 Requerimientos no Funcionales.....	29

II.3	Modelo de Casos de Uso del Sistema.....	32
II.3.1	Actores del sistema.....	32
II.3.2	Casos de Uso del Sistema.....	33
II.3.3	Diagramas de casos de uso del sistema.....	33
II.3.4	Descripción de los Casos de Uso del Sistema.....	36
II.4	Factibilidad Económica.....	54
II.4.1	Planificación por puntos de función.....	56
II.4.2	Determinación de los costos	62
II.4.3	Beneficios tangibles e intangibles	67
II.4.4	Análisis de costos y beneficios	67
	Conclusiones	67
Capítulo III:	“Diseño de la Solución Propuesta”	68
III.1	Principios de Diseño del Sistema	68
III.1.1	Estándares en la Interfaz de la Aplicación	68
III.1.2	Concepción General de la Ayuda	69
III.1.3	Tratamiento de Excepciones.....	69
III.2	Modelo de Clases Web.....	70
III.3	Diseño de la Base de datos.	72
III.3.1	Diagrama del modelo lógico de datos.	73
III.3.2	Diagramas del modelo físico de datos.	73
III.4	Diagramas de Implementación	73
	Conclusiones	75
	Conclusiones.....	76
	Recomendaciones	77

Índice de figuras

Fig 1 Diagrama de clases del modelo de objetos del dominio.	26
Fig 2 Diagrama de Paquetes de Casos de Usos.	33
Fig 3 Diagrama de caso de Uso del Sistema: Paquete Administrativo.	34
Fig 4 Diagrama de caso de Uso del Sistema: Paquete Reportes.	35
Fig 5 Diagrama de Implementación	74

Índice de Tablas

Tabla 1	Sistemas Gestores de Bases de Datos.	23
Tabla 2	Actores del Sistema.....	32
Tabla 3	Descripción del caso de uso de sistema: Crear usuario	36
Tabla 4	Descripción del caso de uso de sistema: Habilitar usuario	37
Tabla 5	Descripción del caso de uso de sistema: Inhabilitar usuario.....	38
Tabla 6	Descripción del caso de uso de sistema: Asignar permisos a usuarios.....	39
Tabla 7	Descripción del caso de uso de sistema: Asignar asignatura al profesor. ...	40
Tabla 8	Descripción del caso de uso de sistema: Buscar usuario.	40
Tabla 9	Descripción del caso de uso de sistema: Cambiar contraseña.....	41
Tabla 10	Descripción del caso de uso de sistema: Autenticar usuario	42
Tabla 11	Descripción del caso de uso de sistema: Cambiar contraseña usuario registrado	43
Tabla 12	Descripción del caso de uso de sistema: Asignar SE a asignatura.	44
Tabla 13	Descripción del caso de uso de sistema: Editar nombre de SE.....	44
Tabla 14	Descripción del caso de uso de sistema: Eliminar SE del sistema.	45
Tabla 15	Descripción del caso de uso de sistema: Eliminar asignatura por profesor .	46
Tabla 16	Descripción del caso de uso de sistema: Consultar ayuda.....	47
Tabla 17	Descripción del caso de uso de sistema: Visualizar listado de estudiantes que visitaron un SE por Grupo.....	47
Tabla 18	Descripción del caso de uso de sistema: Visualizar el contenido visitado por un estudiante en el SE.	48
Tabla 19	Descripción del caso de uso de sistema: Visualizar fecha de entrada y salida de un estudiante en un contenido de un SE.	49
Tabla 20	Descripción del caso de uso de sistema: Visualizar cantidad de visitas al SE. 49	
Tabla 21	Descripción del caso de uso de sistema: Visualizar cantidad de visitas del estudiante al SE.....	50
Tabla 22	Descripción del caso de uso de sistema: Visualizar el tiempo que estuvo un estudiante en un SE por fecha.	51

Tabla 23 Descripción del caso de uso de sistema: Visualizar evaluación de un estudiante por contenido en un SE.	52
Tabla 24 Descripción del caso de uso de sistema: Visualizar respuesta de un estudiante en una pregunta de un contenido de un SE.	53
Tabla 25 Descripción del caso de uso de sistema: Visualizar fecha de entrada y salida de un estudiante al SE.	53
Tabla 26 Descripción del caso de uso de sistema: Visualizar SSEE de una asignatura.	54
Tabla 27 Planificación: Entradas Externas	56
Tabla 28 Planificación: Salidas Externas.....	57
Tabla 29 Planificación: Peticiones	58
Tabla 30 Planificación: Ficheros Internos.	61
Tabla 31 Planificación: Puntos de Función.....	61
Tabla 32 Planificación: Miles de Instrucciones Fuertes	62
Tabla 33 Costos: Factores de escalas.....	63
Tabla 34 Costos: Factores de Ajuste.....	65
Tabla 35 Diagrama de Clases Web. Paquete Administrativo	71
Tabla 36 Diagrama de Clases Web. Paquete Reportes.	72

Resumen

El presente trabajo nombrado “Sistema para el Control del Uso de los Softwares Educativos”, fue desarrollado en la Universidad de Cienfuegos como parte de una investigación acerca del uso de las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC) en el ámbito educativo. Su propósito es desarrollar una interfaz para la Plataforma para el Control del Uso de Softwares Educativos (PCUSE) que permite hacer uso de la información que dicha plataforma manipula. PCUSE está basado en el modelo de tres capas de programación con tecnología de servicios web y XML.

Para el desarrollo de este proyecto se utilizaron la metodología RUP para la descripción del proceso ingenieril, UML como lenguaje de modelado orientado a objeto y PHP como lenguaje de programación. Apache es el servidor web elegido para ejecutar la aplicación. Además se utilizó el modelo de tres capas y se procesa la información recibida de PCUSE en formato XML.

Se realizó un estudio que involucró a profesores y estudiantes de la Facultad de Ciencias Informáticas para determinar la información de interés del comportamiento de los estudiantes en el uso de los Softwares Educativos.

Al finalizar la primera etapa del proyecto (Análisis del Sistema), se realizó un estudio de factibilidad, utilizando la metodología COCOMO nivel II.

Introducción

Los adelantos en la rama de la informática y los avances multiplicados van demostrando que la concepción de la ciencia está en constantes cambios significativos, apoyados en el surgimiento de diferentes tecnologías lo cual requiere una permanente preparación profesional y cambios en la concepción social, económico y cultural de la sociedad, abriendo espacios de búsquedas científicas dentro de los elementos que tienen que ver con el desarrollo y como centro el conocimiento que se genera y aplica en el progreso de la Revolución Tecnológica.

En la actualidad los cambios realizados en el proceso de enseñanza-aprendizaje son necesarios y responden a los problemas relativos en la educación, al nuevo contexto, relacionados, en menor medida, con limitaciones tecnológicas y con la creatividad del hombre, por ello sigue siendo su formación y desarrollo la premisa fundamental de la propuesta cubana en la Tercera Revolución Educacional.

Por todos es conocido el surgimiento y desarrollo de los Softwares Educativos (SSEE) en la actualidad y la aplicación de las diferentes herramientas asociadas a los requerimientos que impone nuestra educación. El producto nacional en este sentido ha engendrado el potencial humano existente en los centros de Enseñanza, lo que resulta una contribución significativa al proceso renovador que emprende el país en esta esfera del saber.

El desarrollo de la informática ha permitido la creación de SSEE con los que se logra individualizar el proceso de enseñanza-aprendizaje, permiten un vínculo estrecho entre profesor y estudiante, posibilitan la retroalimentación particularizada en dependencia de las características de cada estudiante, la utilización de novedosas técnicas y todo esto redundando en una mejor preparación de los profesionales.

En el caso de nuestro país se han desarrollado innumerables SSEE, pero la posibilidad de valorar individualmente al estudiante en unos casos es pésima y en otros nula.

En la Universidad de Cienfuegos se desarrolló una Plataforma para el Control del Uso de los Softwares Educativos (PCUSE), basada en las nuevas tecnologías de

acceso a las redes informáticas que permite mantener un control estricto y centralizado del acceso de los estudiantes a los SSEE. Esta plataforma es independiente del sistema operativo y/o lenguaje de programación. Fue desarrollada en PHP, basándose en el uso de los Servicios Web, XML, SOAP y utiliza bibliotecas PEAR. Teniendo en cuenta las necesidades para la inserción de diferentes herramientas que respondan a los intereses del nuevo proyecto educativo que se aplica hace solo unos años. En aras de perfeccionar el trabajo de los diferentes profesores a la hora de la aplicación de Softwares Educativos en las diferentes asignaturas, especialidades o niveles de la enseñanza lo cual se hace necesario pensar en un sistema que nos posibilite el control del uso de los Softwares Educativos (SSEE) por parte de los estudiantes, desde el punto de vista informático y posibilite dar o emitir diferentes criterios evaluativos de los estudiantes por parte del profesor, por lo que se plantea el siguiente **problema a resolver**: la nueva Plataforma para el Control del Uso de los Softwares Educativos (PCUSE) no cuenta con una herramienta que le facilite al profesor la interacción con la información recopilada por el mismo.

Para realizar este sistema de control de SSEE se tuvo en cuenta otros trabajos realizados y la importancia que hoy en día han tomado los Softwares Educativos hasta convertirse en un elemento importante en los planes de perfeccionamiento de los sistemas de enseñanza y han extendido su uso a otras esferas de la actividad humana en las que la gestión del conocimiento ocupa un lugar preponderante, además se consultó diferentes temas relacionados al sistema en cuestión, lo que pudo permitir un mejor entendimiento de la materia y de las herramientas a utilizar para la realización de la investigación. En este proyecto aboga por la articulación de la tecnología computacional (desde la perspectiva de la comunicación visual) con la educación (las variables tecnológicas frente a la comunicación educativa), no como mera capacitación, sino desde un punto de vista fuertemente didáctico, que atienda a la diversidad; esto es, no descuidar la singularidad psíquica del educando.

En la concepción del trabajo se proyectó como **campo de acción** la información generada en el proceso de interacción de los estudiantes con los Softwares

Educativos, por lo que se definió como **objeto de estudio** la Plataforma para el Control del Uso de Softwares Educativos (PCUSE).

Desde el punto de vista científico se definió como **idea a defender**: La creación de una herramienta que utilice los nuevos paradigmas de servicios web permitirá acceder a la información recopilada por el entorno PCUSE desde cualquier estación de trabajo que esté conectada a dicho entorno.

Para dar cumplimiento al problema científico planteado anteriormente nos trazamos el **Objetivo General** siguiente:

Desarrollar un sistema informático que permita al profesor tener un control del uso de los Softwares Educativos por parte de los estudiantes y poder emitir un criterio evaluativo.

Los **Objetivos Específicos** del sistema responden a:

- Estudiar el uso de los SSEE por parte de los estudiantes.
- Analizar las necesidades elementales para el control de un SE por parte del profesor.
- Diseñar un sistema multiplataforma en cuanto a Sistema Operativo y Gestor de Base de Datos se refiere.

Para dar cumplimiento a los diferentes objetivos propuestos nos trazamos las siguientes tareas a resolver.

1. Analizar el funcionamiento, características y las diferentes especificidades de los SSEE.
2. Entrevistar a los profesores para conocer la información de interés sobre el comportamiento de un estudiante en un SE.
3. Valorar las tecnologías y/o metodologías a emplear y escoger las herramientas que permitan solucionar el problema científico planteado.
4. Analizar y diseñar la Base Datos que utilizará el sistema.
5. Elaborar un prototipo que de la posibilidad a los clientes de satisfacer las necesidades que brinda el sistema.

6. Documentar la información referente al análisis, diseño e implementación del sistema.

La puesta en práctica de dicho sistema brindará el siguiente **aporte práctico**: Mediante la herramienta informática propuesta el profesor podrá obtener la información necesaria acerca del comportamiento del estudiante en un Software Educativo.

El siguiente trabajo se adecuó de forma tal que el orden propiciara un entendimiento del argumento teórico, cumpliendo con la metodología de la investigación y facilitando el cumplimiento de los objetivos propuestos para el proyecto, por lo que su estructuración es de la manera siguiente:

Capítulo I “Fundamentación Teórica”: Donde se hace un análisis crítico de las bibliografías utilizadas. Se describen las metodologías, softwares y lenguajes de programación además de hacer alusión al por qué de su utilización.

Capítulo II “Descripción del Problema”: Se hace una descripción lógica del diseño del modelo del dominio y del sistema propuesto, se definen los objetos y conceptos principales dentro del sistema, además de los diferentes casos de uso utilizados con sus relaciones y los correspondientes diagramas. Se realizan los cálculos correspondientes a los costos, análisis de las planificaciones y de las escalas.

Capítulo III “Diseño de la Solución Propuesta”: Se basa en los diferentes principios aplicados en el sistema, el tratamiento de ellos y el modelo de clases Web así como el diseño de trabajo de la base de datos con los diagramas de modelo lógico y físico de los datos y su implementación.

Capítulo I: "Fundamentación Teórica"

En este capítulo se abordarán aspectos teóricos del tema que se va a analizar, exponiendo los principales conceptos asociados al dominio del mismo. Se describe el contexto donde se enmarca, las características y dificultades que lo acompañan, así como las metodologías y tecnologías a utilizar.

I.1 Tecnologías de la Información y las Comunicaciones (TIC)

Existen innumerables conceptos para referirse a las TICs, en [1] se usa el siguiente: Las Tecnologías de Información y las Comunicaciones "son un conjunto de aparatos, redes y servicios que se integran o se integrarán a la larga, en un sistema de información interconectado y complementario. La innovación tecnológica consiste en que se pierden las fronteras entre un medio de información y otro".

Hoy en día los maestros que deciden emplear la computación como medio de enseñanza tienen a su disposición una amplia gama de programas que pueden ser empleados por ellos con propósitos educativos.

I.1.1 Software Educativo

Sánchez J. (1999), en su Libro "Construyendo y Aprendiendo con el Computador", define el concepto genérico de Software Educativo como cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar.

Según Rodríguez Lamas (2000), es una aplicación informática, que soportada sobre una bien definida estrategia pedagógica, apoya directamente el proceso de enseñanza aprendizaje constituyendo un efectivo instrumento para el desarrollo educacional del hombre del próximo siglo.

Los Softwares Educativos (SSEE) se pueden considerar como el conjunto de recursos informáticos diseñados con la intención de ser utilizados en el contexto del proceso de enseñanza – aprendizaje. [2]

Se caracterizan por ser interactivos, a partir del empleo de recursos multimedia, como videos, sonidos y fotografías. Todo esto es usado para construir diccionarios especializados, profundizar en las explicaciones de experimentados profesores, ejercicios y juegos instructivos que sirven de apoyo a las funciones de enseñanza, evaluación y diagnóstico.

Los softwares educativos pueden tratar las diferentes materias (Matemáticas, Idiomas, Geografía, Dibujo, etc.), de formas muy diversas a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos, y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los alumnos y rico en posibilidades de interacción; pero todos comparten las siguientes características: **[3]**

- ✓ Permiten la interactividad con los estudiantes, retroalimentándolos y evaluando lo aprendido.
- ✓ Facilitan las representaciones animadas.
- ✓ Inciden en el desarrollo de las habilidades a través de la ejercitación.
- ✓ Permiten simular procesos complejos.
- ✓ Reducen el tiempo que se dispone para impartir gran cantidad de conocimientos facilitando un trabajo diferenciado, introduciendo al estudiante en el trabajo con los medios computarizados.
- ✓ Facilitan el trabajo independiente y a la vez un tratamiento individual de las diferencias.
- ✓ Permiten al usuario introducirse en las técnicas más avanzadas.

El uso de los softwares educativos en el proceso de enseñanza - aprendizaje puede ser:

Por parte del alumno. Esto se evidencia cuando el estudiante opera directamente el SE, pero en este caso es de vital importancia la acción dirigida por el profesor.

Por parte del profesor. Esto se manifiesta cuando el profesor opera directamente con el software y el estudiante actúa como receptor del sistema de información. La generalidad plantea que este no es el caso más productivo para el aprendizaje.

El uso de los SSEE por parte del docente proporciona numerosas ventajas, entre ellas:

- ✓ Enriquecen el campo de la Pedagogía al incorporar la tecnología de punta que revoluciona los métodos de enseñanza - aprendizaje.
- ✓ Constituyen una nueva, atractiva, dinámica y rica fuente de conocimientos.
- ✓ Pueden adaptarse a las características y necesidades de un grupo, teniendo en cuenta el diagnóstico en el proceso de enseñanza - aprendizaje.
- ✓ Permiten elevar la calidad del proceso docente - educativo.
- ✓ Permiten controlar las tareas docentes de forma individual o colectiva.
- ✓ Proporcionan la interdisciplinariedad de las asignaturas.
- ✓ Marcan las posibilidades para una nueva clase más desarrolladora.

Los Softwares Educativos se presentan con unas características muy diversas, a pesar de tener rasgos esenciales básicos y una estructura general común. Estos rasgos son: **[4]**

- ✓ aparentan ser un laboratorio o una biblioteca,
- ✓ se limitan a ofrecer una función instrumental del tipo máquina de escribir o calculadora,
- ✓ se presentan como un juego o como un libro,
- ✓ constituyen exámenes,
- ✓ se utilizan como expertos.

Es común encontrar algunos Softwares Educativos que combinan más de uno de los rasgos anteriores.

Para poner orden a esta disparidad, se elaboraron múltiples tipologías que los clasifican a partir de diferentes criterios.

Por ejemplo, hasta el año 2003, existían dos tipos de Softwares Educativos, según los polos en los cuales se ha movido la educación: Algorítmicos y Heurísticos.

Los Softwares Educativos considerados Algorítmicos son aquellos donde predomina el aprendizaje vía transmisión del conocimiento, pues el rol del alumno es asimilar el máximo de lo que se le transmite. [5]

Los Softwares Educativos Algorítmicos se pueden clasificar considerando la función educativa en:

Sistemas Tutoriales,

Sistemas Entrenadores,

Libros Electrónicos.

Los Softwares Educativos considerados Heurísticos son aquellos donde el estudiante descubre el conocimiento interactuando con el ambiente de aprendizaje que le permita llegar a él.

Considerando la función educativa se pueden clasificar en:

Simuladores,

Juegos Educativos,

Sistemas Expertos,

Sistemas Tutoriales Inteligentes de Enseñanza.

A partir del 2004 surge una nueva tendencia, que es la de integrar en un mismo producto, todas o algunas de estas tipologías de softwares educativos. A este nuevo modelo de software se le ha denominado HIPERENTORNO EDUCATIVO o HIPERENTORNO DE APRENDIZAJE, lo cual no es más que un sistema informático basado en tecnología hipermedia que contiene una mezcla de elementos representativos de diversas tipologías de software educativo.

I.2 Descripción de los sistemas existentes

En la esfera educativa han surgido muchas herramientas para ayudar al profesor en su trabajo. Dichas herramientas son utilizadas tanto en la educación presencial, como a distancia. Esto se logra a través del uso de las nuevas tecnologías en la educación. Ejemplo de estas plataformas son: CLAROLINE, TELEDUC, ILIAS, GANESHA, FLE3, MOODLE, MICROCAMPUS.

Moodle

Es una plataforma de aprendizaje a distancia basada en software libre. Constituye un sistema de gestión de la enseñanza, es decir, una aplicación diseñada para ayudar a los educadores a crear cursos de calidad en línea. En términos de arquitectura, se trata de una aplicación web que puede funcionar en cualquier computadora, pero no permite retroalimentarse sobre el uso de cualquier tipo de software educativo.

Microcampus

Es un gestor de cursos de última generación para el apoyo de la enseñanza presencial y una herramienta esencial para la enseñanza a distancia a través de Internet. Perteneciente a una red de centros en Iberoamérica, que desarrollan proyectos educativos basados en la educación a distancia con el uso de las nuevas tecnologías aplicadas a la formación. Permite conocer en detalles el uso por parte de los estudiantes de la plataforma, pero no contempla el uso de otro software desarrollado en plataformas diferentes.

CLAROLINE

Es un gestor de educación a distancia que ofrece facilidades de trabajo a estudiantes, profesores y trabajadores de centros educativos, incluye vínculos a sistemas utilizados en el centro donde se usa la plataforma pero no contiene ninguna herramienta que permita conocer el desarrollo de los estudiantes en los softwares educativos.

Se encontraron otras plataformas en el mercado como TELEDUC, ILIAS, GANESHA, FLE3, entre otras, pero ninguna permite salvar información referente al uso y consumo de softwares instalados en la red.

I.3 Metodologías Utilizadas

I.3.1 Lenguaje de Modelación Unificado (UML)

El Lenguaje de Modelado Unificado (UML - Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Este lenguaje fue creado por un grupo de estudiosos de la Ingeniería de Software formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995. Desde entonces, se ha convertido en el estándar internacional para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. Con este lenguaje, se pretende unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes. [6]

Entre sus objetivos fundamentales se encuentran: [7]

1. Ser tan simple como sea posible, pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir.
2. Necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como los mecanismos de la ingeniería de software, como son el encapsulamiento y el uso de componentes.
3. Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
4. Imponer un estándar mundial.

I.3.2 Proceso Unificado de Desarrollo (RUP)

El Proceso Unificado de Desarrollo (RUP, por su denominación en inglés Rational Unified Process), fue creado por el mismo grupo de expertos que crearon UML, Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998. El objetivo que se perseguía con esta metodología era producir software de alta calidad, es decir, que cumpliera con los requerimientos de los usuarios dentro de una planificación y presupuestos establecidos. Como se expresaba anteriormente, esta metodología concibió desde sus inicios el uso de UML como lenguaje de modelado.

Es un proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

A continuación se muestran estas prácticas. [8]

- Desarrollo de software en forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente
- Verifica la calidad del software.
- Controla los cambios.

Para apoyar el trabajo con esta metodología ha sido desarrollada por la compañía norteamericana Rational Corporation la herramienta CASE (Computer Aided Software Engineering) Rational Rose. Esta herramienta integra todos los elementos que propone la metodología para cubrir el ciclo de vida de un proyecto.

I.4 Paradigmas de programación.

I.4.1 Programación por capas.

La **programación por capas** es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio, solo se modifica el nivel requerido sin tener que revisar todo el código. Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, simplemente es necesario conocer la Interfaz de Programación de Aplicaciones (Application Programming Interface - API por sus siglas en Inglés) que existe entre los niveles.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o Programación por capas, teniendo en cuenta las ventajas que esta ofrece. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables, es decir, que pueden ampliarse con facilidad en caso de que las necesidades aumenten.

El diseño más usado en la actualidad es el de tres capas: **[9]**

1. Capa de presentación,
2. Capa de lógica de negocio,
3. Capa de datos.

1.- Capa de presentación: Esta capa provee la interfaz de usuario (IU). Aquí es donde la aplicación presenta información a los usuarios y acepta entradas o visualiza respuestas. Idealmente, la IU no desarrolla ningún procesamiento de negocios o reglas de validación de negocios. Por el contrario, la IU debería relegar sobre la capa de negocios para manipular estos asuntos. Esto es importante, especialmente hoy en día, debido a que es muy común para una aplicación tener múltiples IU. **[10]**

2.- Capa de negocio: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio - LN) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados y con la capa de datos, para solicitar al gestor de base de datos el almacenamiento o la recuperación de los mismos. [11]

3.- Capa de datos: La capa de acceso a datos (AD) está formada por bibliotecas de acceso a los motores de bases de datos, así como por uno o más Sistemas Gestores de Bases de Datos (SGBD) que se encargan del almacenamiento de los datos.

Todas estas capas pueden residir en un mismo equipo. Lo más usual es que haya una multitud de computadoras donde resida la capa de presentación, es decir, los clientes de la arquitectura cliente/servidor, y una computadora central que ejerce la función de servidor en dicha arquitectura. Las capas de negocio y de datos pueden residir en un mismo servidor y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más servidores. Así mismo, si el tamaño o la complejidad de la base de datos aumentaran, se puede separar en varios servidores de datos, los cuales recibirán las peticiones del servidor donde reside la capa de negocio. [12]

Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más computadoras que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de datos y otra serie de ordenadores sobre los cuales corre la base de datos. En este caso estamos haciendo referencia a Sistemas de Aplicaciones Distribuidas.

En una arquitectura de tres niveles, los términos Capas o Niveles no significan lo mismo ni son similares. El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos. El término nivel, corresponde a la forma como las capas lógicas, se encuentran distribuidas de forma física.

I.5 Tecnologías a utilizar

I.5.1 Interfaz de usuario (Capa de Presentación)

Para desarrollar la interfaz de usuario de cualquier aplicación existen varias herramientas como Borland Delphi, Visual Fox Pro, Microsoft FrontPage, Macromedia Dreamweaver entre otras.

Generalmente para el desarrollo de interfaz de intranets se utilizan herramientas web por las comodidades que ofrecen. Se realizó un análisis de las características fundamentales de Macromedia Dreamweaver MX 2004 para justificar el por qué de su selección para la realización del diseño de la interfaz de este proyecto.

Macromedia Dreamweaver

Macromedia Dreamweaver MX es uno de los editores de desarrollo Web más utilizado a nivel profesional para la creación de sitios Web. Su amplio abanico de herramientas permite crear desde la más simple página Web personal hasta el sitio Web más completo y complejo para una gran empresa y utilizar casi todos los recursos de la Web. Este editor de HTML profesional para el diseño, codificación y desarrollo de páginas, sitios y aplicaciones Web; permite la edición visual, o sea, crear páginas rápidamente sin escribir código, así como también la codificación manual. Dreamweaver ayuda además a construir aplicaciones Web dinámicas apoyadas en bases de datos, es completamente personalizable. Se pueden crear objetos y comandos propios, modificar los accesos directos de teclado, e incluso, escribir código *script* (guión) para extender las capacidades de las páginas web creadas con nuevos comportamientos. [13]

Dreamweaver soporta varias tecnologías del servidor para la construcción de aplicaciones Web, tales como: Macromedia ColdFusion, Microsoft ASP, Microsoft ASP.NET, Sun JavaServer Pages (JSP) y PHP.

AJAX

Es un acrónimo de Asynchronous Javascript And XML (Javascript Asincrónico y XML) no es una tecnología web, sino el uso coordinado de distintas tecnologías que

en conjunto permiten una mayor rapidez y eficacia para las aplicaciones basadas en la Web. [14]

Hasta hace muy poco tiempo una de las principales diferencias entre las aplicaciones Web y las aplicaciones de escritorio era sin dudas su rapidez de ejecución, en el momento en que el volumen de datos a gestionar era algo elevado, la ralentización era la normal. Una de las principales aportaciones de AJAX, es acelerar la velocidad de ejecución de estas aplicaciones web haciéndolas muy similares a las aplicaciones de escritorio. Para ello la clave esta en modificar existencialmente el tipo de relaciones que se crean entre el navegador Web y el servidor.

I.5.2 Lógica del negocio (Capa de la Lógica del Negocio)

En la capa de lógica del negocio es necesario seleccionar un lenguaje de programación y un servidor de aplicaciones web. Teniendo en cuenta que existen varios lenguajes de programación web del lado del servidor (*Server Side Scripts* - SSS) como ASP, JSP, PHP entre otros, se analizaron las características fundamentales que nos permitieron seleccionar cual utilizar y en el caso de los servidores web se analizaron las características fundamentales de los dos servidores más difundidos en el mundo: Internet Information Services (IIS) y Apache. [15]

I.5.2.1 Lenguajes de programación web

ASP

Active Server Pages (ASP) es una tecnología de Microsoft que funciona del lado servidor permite generar páginas web dinámica, ha sido comercializada como un anexo a Internet Information Services (IIS), está gratuito en el mercado, pero depende de la plataforma Windows. [16]

JSP

JSP es un acrónimo de Java Server Pages, (Páginas de Servidor Java). Es una tecnología orientada a crear páginas web con programación en Java. Con JSP pueden crearse aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. [17]

PHP

PHP (Profesional Home Page Tools) es un lenguaje de programación el cual se ejecuta en los servidores Web y que permite crear contenido dinámico en las páginas HTML, con un lenguaje propietario derivado del Perl. [18]

Al principio, PHP solo estaba compuesto por algunas macros que facilitaban el trabajo a la hora de crear una página Web. Hacia mediados de 1995 se creó el analizador sintáctico y se llamó PHP/F1 Versión 2, y sólo reconocía el texto HTML y algunas directivas de MySQL. A partir de este momento, la contribución al código fue pública. El crecimiento de PHP desde entonces ha sido exponencial y han surgido versiones nuevas como las actuales, PHP3 y PHP4.

Dispone de múltiples herramientas que permiten acceder a bases de datos de forma sencilla, por lo que es ideal para crear aplicaciones para Internet.

Es multiplataforma, funciona tanto para Unix como para Windows de forma que el código que se haya creado para una de ellas no tiene por qué modificarse al pasar a la otra.

El lenguaje PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, funciones, entre otras. La sintaxis que utiliza la toma de otros lenguajes muy extendidos como C y Perl. [19]

El lenguaje PHP, como todos los del tipo "guiones del lado del servidor", envía al navegador código HTML, por lo que es transparente al usuario final.

PHP se encuentra libre en el mercado y se puede acceder a él por medio de Internet. Cuenta además con un repositorio muy completo de clases, PEAR (Repositorio de Aplicaciones y Extensiones de PHP), en el cual se puede encontrar desde clases para manejar ecuaciones matemáticas o para generar gráficos, hasta clases para generar hojas de cálculo en Excel, de una forma fácil y con funciones pocas veces vista con PHP. PEAR cuenta con 273 clases (hasta el momento de la revisión bibliográfica), está dividido en dos partes principales: PEAR y PCL, el primero lo constituyen paquetes escritos completamente en PHP, el segundo está compuesto por clases escritas en C o C++. [20]

I.5.2.2 Servicios web

Un servicio web es una entidad programable que proporciona alguna funcionalidad determinada y es accesible a cualquier número de sistemas que usen las normas de Internet. Un servicio web puede ser usado internamente por una aplicación o ser publicado hacia Internet. Estos servicios permiten la ejecución de sus funcionalidades sin importar la plataforma, sistema operativo, o lenguaje en el cual estén implementados. [21]

Los servicios web se pueden utilizar para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. [22]

Los servicios web tienen una interfaz descrita en un formato que puede ser procesado por una máquina (específicamente WSDL) y otros sistemas interactúan con el servicio web utilizando mensajes SOAP.

Los servicios web brindan grandes ventajas dentro de las aplicaciones distribuidas como son:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en textos, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados. [23]

Atendiendo a las ventajas, antes mencionadas, que brindan el uso de webservice se incluye su utilización en el desarrollo de este proyecto.

Los servicios web pueden ser utilizados a través de dos protocolos fundamentalmente: RPC y SOAP. El protocolo RPC tiene una dependencia de la plataforma y la tecnología que se utiliza para ejecutar los servicios web, en cambio SOAP es un protocolo que permite la independencia de la plataforma y de la tecnología.

I.5.2.3 XML

El Lenguaje Extensible de Marcas, abreviado XML (eXtensible Markup Language), describe una clase de objetos de datos llamados documentos XML y describe parcialmente el comportamiento de los programas de computadora que los procesan. XML es un "perfil de aplicación" o una forma restringida de SGML, el Lenguaje Estándar Generalizado de Marcación [ISO 8879]. Por construcción, los documentos XML son documentos SGML conformados. [24]

Los documentos XML están compuestos por unidades de almacenamiento llamadas entidades, que contienen tanto datos analizados como no analizados. Los datos analizados están compuestos de caracteres, algunos de los cuales, de la forma datos carácter, y otros de la forma etiquetas. Las etiquetas codifican una descripción de la estructura de almacenamiento del documento y su estructura lógica.

XML fue desarrollado por un Grupo de Trabajo XML (originalmente conocido como "*SGML Editorial Review Board*") formado bajo los auspicios del Consorcio World Wide Web (W3C), en 1996. Fue presidido por Jon Bosak de Sun Microsystems con la participación activa de un Grupo Especial de Interés en XML (previamente conocido como Grupo de Trabajo SGML) también organizado en el W3C. Dan Connolly sirvió como contacto entre el GT y el W3C.

Los objetivos de diseño de XML son [25]:

1. XML debe ser directamente utilizable sobre Internet.
2. XML debe soportar una amplia variedad de aplicaciones.
 - XML debe ser compatible con SGML.
 - Debe ser fácil la escritura de programas que procesen documentos XML.
 - El diseño de XML debe ser formal y conciso.
 - Los documentos XML deben ser fácilmente creables.

XML es la solución a un problema de comunicación entre programas de computadoras. XML intenta ser un formato absolutamente genérico, con el que se puede describir cualquier tipo de archivo.

Características de XML

- Es una arquitectura más abierta y extensible. No se necesitan versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en internet/intranet por medio de un validador de documentos (parser).
- Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML (los RDF Resource Description Framework), en comparación a los atributos de la etiqueta <META> del HTML.

- Integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en el propio PC como en una red local o extensa.
- Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje nos permitirá agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- Gestión y manipulación de los datos desde el propio cliente web.
- Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte más accesible.
- Se desarrollarán de manera extensible las búsquedas personalizables y subjetivas para robots y agentes inteligentes. También conllevará que los clientes web puedan ser más autónomos para desarrollar tareas que actualmente se ejecutan en el servidor.
- Se permitirá un comportamiento más estable y actualizable de las aplicaciones web, incluyendo enlaces bidireccionales y almacenados de forma externa.
- El concepto de "hipertexto" se desarrollará ampliamente (permitirá denominación independiente de la ubicación, enlaces bidireccionales, enlaces que pueden especificarse y gestionarse desde fuera del documento, hiperenlaces múltiples, enlaces agrupados, atributos para los enlaces, etc. Creado a través del Lenguaje de enlaces extensibles, abreviado XLL (eXtensible Linking Language).
- Exportabilidad a otros formatos de publicación (papel, web, cd-rom, etc.). El documento maestro de la edición electrónica podría ser un documento XML que se integraría en el formato deseado de manera directa. **[26]**

I.5.2.4 SOAP

SOAP, siglas de *Simple Object Access Protocol*, es un protocolo estándar creado por Microsoft, IBM y otros, está actualmente bajo el auspicio de la W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web. [27]

A diferencia de DCOM y CORBA, que son binarios, SOAP usa el código fuente en XML, que facilita la eliminación de errores, pero es menos efectivo. El intercambio de mensajes se realiza mediante tecnología de componentes. El término *Object* en el nombre significa que se adhiere al paradigma de la programación orientada a objetos.

SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí.

SOAP corre sobre cualquier protocolo de Internet, generalmente HTTP. SOAP tiene como base XML, con un diseño que cumple el patrón Cabecera-Desarrollo de diseño de software, como otros muchos diseños, verbigracia HTML. La cabecera (*header*) es opcional y contiene metadatos sobre enrutamiento (*routing*), seguridad o transacciones. El desarrollo (*body*) contiene la información principal, que se conoce como carga útil (*payload*). La carga útil se acoge a un esquema de XML propio.

I.5.2.5 Servidores de Aplicaciones Web.

Se realizó un estudio de dos de los servidores de aplicaciones web más difundidos para determinar cuál será usado en el proyecto.

Apache

El **servidor Apache** es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se

basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. [28]

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, carece de una interfaz gráfica que ayude en su configuración.

En la actualidad, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

Internet Information Services

IIS engloba una serie de herramientas administrativas que permite controlar sitios Web, FTP, SMTP (correo saliente) y Servicio de noticias (NNTP). Dispone también del soporte para crear páginas dinámicas (ASP), tecnología para el desarrollo de aplicaciones para Internet ampliamente extendida.

Después de analizar las características fundamentales de Apache e IIS se decide usar como servidor de aplicaciones web Apache teniendo en cuenta que se encuentra gratis en Internet y posee una alta compatibilidad con PHP.

I.5.3 Capa de acceso a datos.

Sistemas Gestores de Base de Datos

Una Base de Datos (BD) es un conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo. [29]

Un **Sistema de Gestión de Base de Datos** (SGBD) 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. [30]

El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de

forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

En el mercado existen hoy un sinnúmero de gestores de base de datos, todos con sus características que lo hacen una opción a escoger, pero la elección, la mayoría de las veces, depende del cliente y no del desarrollador.

A continuación mostramos una tabla comparativa de algunos de los SGBD en cuanto a si son Multiplataforma o no, Gratis o no y en cuanto a la disponibilidad del código fuente.

	Multiplataforma	Gratis	Disponibilidad Código Fuente
MSSQL	No	No	No
MSDE	No	Si	No
MySQL	Sí	Si	Sí
Postgres	Sí	Si	Sí
Firebird	Sí	Si	Si
Interbase	Si	No	Sí
SyBase	Si	No	No
INFORMIX	Sí	No	No
Oracle	Sí	no	No

Tabla 1 Sistemas Gestores de Bases de Datos.

Conclusiones

A partir de la información recopilada en este capítulo detectamos la necesidad de crear una aplicación que permita interaccionar con el entorno PCUSE. Para el desarrollo de esta herramienta utilizamos la metodología RUP por ser esta sobre la

cual se tiene mayor conocimiento, además de que se ajusta a las necesidades del proyecto en cuestión. Para la modelación de este sistema se utilizó UML por ser el lenguaje más relacionado con RUP y permitirnos la modelación orientada a objetos.

Dicha aplicación debe desarrollarse utilizando el modelo de tres capas y basar su funcionamiento en el uso de la tecnología de los servicios web.

Las herramientas seleccionadas para la programación de dicha aplicación son las siguientes:

- Para la Interfaz de Usuario (IU) se utilizará Macromedia Dreamweaver, Adobe Photoshop.
- Para la capa de la Lógica del Negocio (LN) se utilizará Apache como Servidor de Aplicaciones WEB, debido fundamentalmente a su gratuidad en el uso y la posibilidad multiplataforma del mismo. La elección del lenguaje de programación es evidente. Se seleccionará PHP por ser este uno de los lenguajes de código abierto (open source) más difundidos, ser independiente de plataforma y estar muy ligado al Servidor de Aplicación WEB seleccionado. Se utilizará PHP Expert Editor como herramienta de programación en este lenguaje. Para el manejo de los datos en esta capa se recomienda utilizar XML por ser independiente de plataforma y ampliamente utilizado por la mayoría de los sistemas modernos de comunicaciones para el intercambio de datos.
- En la capa de acceso a datos se utilizará las bibliotecas PEAR escritas en PHP y como motor de base de datos MySQL, por ser multiplataforma y de código abierto.
- Para la comunicación entre capas se utilizará XML como lenguaje de intercambio de información.

Capítulo II: “Descripción del problema”

En el presente capítulo se describe el análisis de la solución propuesta utilizando UML como lenguaje de modelado y RUP como Metodología empleando uno de los artefactos que brinda: el Modelo del Dominio, además se exponen los requerimientos funcionales y no funcionales, descripción general del producto que se propone, así como el modelo de casos de uso del sistema y la descripción de cada uno de estos. En este capítulo, además, se hace un análisis de la factibilidad para la aplicación desarrollada, utilizando como metodología COCOMO II.

II.1 Descripción del modelo del dominio.

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio. [31]

El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema.

Para una mayor comprensión del contexto en que se desarrolla el sistema se definen los principales conceptos relacionados con el entorno del problema.

II.1.1 Definición de los objetos y los conceptos principales

En el modelo de dominio se definen las siguientes clases principales: Profesor, Estudiante, Asignatura, Software Educativo.

Profesor: Usuario interesado en obtener del sistema el desempeño del estudiante en el SE.

Estudiante: Es el usuario que utiliza el SE y se registra su comportamiento en el sistema para ser analizado por el profesor.

Asignatura: Materia que imparte un profesor a un grupo de estudiantes.

Software Educativo: Herramienta que utiliza el profesor como complemento educativo de la asignatura.

II.1.2 Representación del Modelo de Objetos del Dominio.

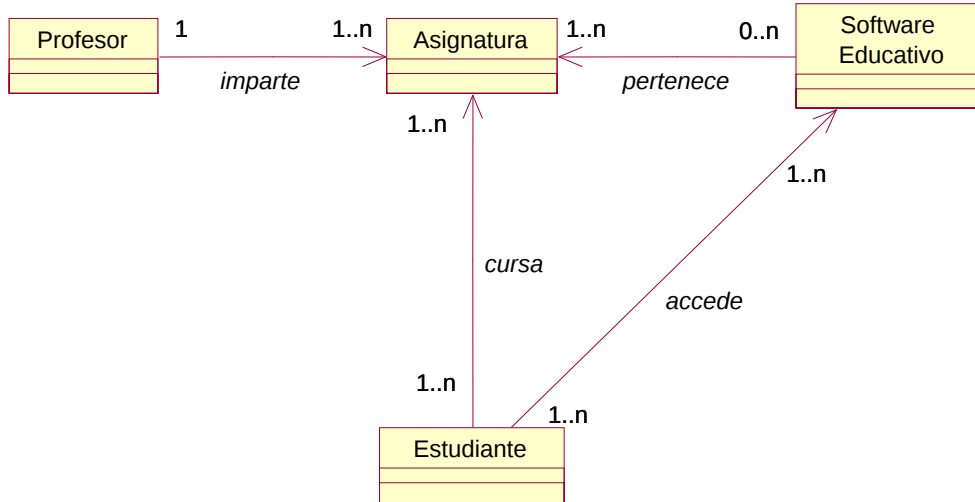


Fig 1 Diagrama de clases del modelo de objetos del dominio.

II.1.3 Reglas del negocio a considerar.

- ✓ Los usuarios del sistema son: Administrador y Profesor.
- ✓ Los usuarios del sistema pueden estar activos o inactivos temporalmente.
- ✓ El administrador del sistema puede asignar permisos al usuario según su rol en el mismo.
- ✓ Un profesor puede tener más de una asignatura asignada.
- ✓ Un profesor solo tiene acceso a las asignaturas que tiene asignada y a su vez a los SSEE de estas.
- ✓ Solo los profesores pueden acceder a la información de los estudiantes.
- ✓ Un profesor solo puede asignar SE a las asignaturas que el imparte.
- ✓ Los profesores solo tendrán acceso a los grupos donde imparten asignaturas.

- ✓ Un profesor puede utilizar más de un SE en cada asignatura.
- ✓ Un profesor puede acceder al comportamiento del estudiante en tantos SSEE como tenga asignada la asignatura.
- ✓ Un profesor puede impartir más de una asignatura a un mismo grupo.
- ✓ El administrador del sistema puede cambiar la contraseña de todos los usuarios.

II.2 Descripción del sistema propuesto.

El Sistema de Control de Uso de SE, surge por la necesidad de brindarle al profesor una herramienta que le permita evaluar el desempeño del estudiante en los SSEE.

II.2.1 Concepción general del sistema.

El resultado que se pretende alcanzar en esta investigación es la obtención de un producto de software que permita al profesor tener un control y criterio del uso del software educativo por parte del estudiante, lo que resuelve la problemática planteada en la introducción, brindándole la posibilidad a los profesores de utilizar el sistema propuesto con el objetivo de poder realizar un análisis del comportamiento de cada estudiante en los SSEE y el tiempo utilizado en los diferentes contenidos así como otras posibilidades que nos brinda el sistema.

II.2.2 Requerimientos Funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo. [32]

Los requerimientos funcionales del sistema propuesto son los siguientes:

1. Crear usuario.

2. Habilitar usuario.
3. Inhabilitar usuario.
4. Asignar permisos a usuario.
5. Asignar asignatura al profesor.
6. Buscar usuario.
7. Cambiar contraseña.
8. Autenticar usuario.
9. Cambiar contraseña usuario registrado.
10. Asignar SE a asignatura.
11. Editar nombre de SE.
12. Eliminar SE del sistema.
13. Eliminar asignatura por profesor.
14. Consultar ayuda.
15. Visualizar listado de estudiantes que visitaron un SE por grupo.
16. Visualizar el contenido visitado por un estudiante en el SE.
17. Visualizar fecha de entrada y salida de un estudiante en un contenido de un SE.
18. Visualizar cantidad de visitas al SE.
19. Visualizar cantidad de visitas del estudiante al SE.
20. Visualizar el tiempo que estuvo un estudiante en un SE por fecha.
21. Visualizar evaluación de un estudiante por contenido de un SE.
22. Visualizar Respuesta de un estudiante en una pregunta de un contenido de un SE.
23. Visualizar SSEE de una asignatura.

II.2.3 Requerimientos no Funcionales

“Los requerimientos no funcionales especifican cualidades, propiedades del sistema; como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad.” [33]

Los requerimientos no funcionales del sistema propuesto son los siguientes:

Requerimientos de apariencia o interfaz externa

El sistema debe tener una interfaz sencilla, amigable, muy legible y simple de usar, el producto debe ser autoritario e interactivo para que los usuarios se sientan confiados.

Se debe informar al usuario dónde está y qué puede hacer desde allí, al proporcionar señales de navegación que conduzcan al usuario hasta el contenido que desea y evitarle navegar a través de muchas áreas para ello. El contenido debe ser mostrado de manera comprensible para el usuario.

Requerimientos de Usabilidad

El sistema estará dirigido a profesores, interesados en conocer el comportamiento de los estudiantes en los Softwares Educativos. El mismo será utilizado solo por usuarios registrados, estos pueden ser administradores o profesores, a los cuales se les asignan privilegios, que pueden ser: Administración, Registro de Software, Cambio de Contraseña y Archivo. Si el usuario es un profesor solo puede tener acceso a las asignaturas que él imparte y a su vez a los softwares de las mismas, así como a los grupos que tiene asignado y la información individual de los estudiantes sobre el uso de los softwares. El sistema está diseñado para ser utilizado por personas con mínimos conocimientos en el manejo de la computadora y el ambiente Web en sentido general, debido a que contará con una ayuda a fin de documentar al usuario en su utilización. La ejecución de los comandos debe ser posible por el uso del teclado u otros dispositivos como el Mouse.

Los mensajes de error deben ser reportados por la propia aplicación en la medida de las posibilidades y no por el Sistema Operativo. Los mensajes del sistema deben estar en el idioma apropiado.

Requerimientos de Rendimientos

El sistema deberá ser rápido ante las solicitudes de los usuarios y en el procesamiento de la información. La eficiencia de la aplicación estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas a la base de datos. Se realizará la validación de los datos y la manipulación de eventos en el cliente y en el servidor aquellas que por cuestiones de seguridad, o de acceso a los datos lo requieran. Lográndose así un tiempo de respuesta más rápido, una mayor velocidad de procesamiento y un mayor aprovechamiento de los recursos.

Requerimientos de Soporte

Se documentará la aplicación con una ayuda dirigida a los diferentes tipos de usuarios para garantizar el soporte de la herramienta. Se realizará mantenimiento al sistema a fin de aumentar las funcionalidades del mismo a través de versiones posteriores. Los servicios de instalación y mantenimiento del sistema será responsabilidad del administrador del sistema en la entidad que sea utilizado.

Requerimientos de Seguridad

La información estará protegida contra accesos no autorizados utilizando mecanismos de autenticación y autorización que puedan garantizar el cumplimiento de esto: cuenta, contraseña y nivel de acceso, de manera que cada uno pueda tener disponible solamente las opciones relacionadas con su actividad y tenga datos de acceso propios, garantizando así la confidencialidad.

Se usarán mecanismos de encriptación de los datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, como es el caso de las contraseñas. Se guardará encriptada esta información en la base de datos utilizando para ello MD5 como algoritmo de encriptación.

Se harán validaciones de la información tanto en el cliente como en el servidor.

Se crearán usuarios con diferentes niveles de acceso al sistema. Se limitarán los permisos de los usuarios que ejecutan sentencias o consultas SQL. Se utilizarán, además, procedimientos almacenados dado que el modo en que se pasan los parámetros, evita el uso de inyección de código SQL.

No obstante, los usuarios accederán de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para ellos.

Requerimientos Políticos

La aplicación debe cumplir con lineamientos, políticos y/o regulaciones de la entidad que utilice el sistema.

Requerimientos de Confiabilidad

El sistema debe ser tolerante ante los fallos y las operaciones a realizar deben ser transaccionales.

Ayuda y documentación en línea

El sistema tendrá ayuda en la que se realizarán ciertas aclaraciones sobre algunas opciones del sistema que garantizarán el buen desempeño de los usuarios a la hora de interactuar con el mismo.

Requerimientos de Software

La aplicación debe poderse ejecutar en entornos *Windows*, *Linux*, *etc.* (Multiplataforma), para su ejecución del lado del servidor necesita MySQL como sistema gestor de base de datos y Apache como servidor Web, del lado del cliente cualquiera de los exploradores Web existentes en el mercado.

Requerimientos de Hardware

Se requiere de una computadora como servidor de base de datos con los requerimientos de hardware que necesita MySQL. Las terminales clientes solo requerirán de una computadora conectada a la red, para poder ejecutar los navegadores de Web al menos deben cumplir los requisitos mínimos (que requiera el navegador en cuestión)

II.3 Modelo de Casos de Uso del Sistema.

“El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas.” [34]

“Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones.”

“El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario.”

II.3.1 Actores del sistema.

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interaccionan con estos Casos de Uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez identificado los actores del sistema, queda identificado el entorno externo del sistema. [35]

Actor	Justificación
Usuario General	Persona que puede registrarse en el sistema, puede ser: profesor o administrador.
Profesor	Es la persona interesada en conocer el comportamiento de un Estudiante en un Software Educativo.
Administrador	Es quien crea las cuentas de acceso al sistema, le asigna a cada usuario sus permisos en dependencia al rol a desarrollar y cambia la contraseña.

Tabla 2 Actores del Sistema

II.3.2 Casos de Uso del Sistema.

"La forma en que los actores usan el sistema se representa con un caso de uso. Los casos de uso son "fragmentos" de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia." [36]

Por el número de casos de uso se introducen paquetes al modelo de casos de uso del sistema con el objetivo de disminuir la complejidad y así aumentar en comprensión.



Fig 2 Diagrama de Paquetes de Casos de Usos.

II.3.3 Diagramas de casos de uso del sistema.

El Paquete Administrativo (Figura 3) contiene los casos de usos referentes a la administración del sistema:

1. Crear usuario.
2. Habilitar usuario.
3. Inhabilitar usuario.
4. Asignar permisos a usuario.
5. Asignar asignatura al profesor.
6. Buscar usuario.
7. Cambiar contraseña.

8. Autenticar usuario.
9. Cambiar contraseña usuario registrado.
10. Asignar SE a asignatura.
11. Editar nombre de SE.
12. Eliminar SE del sistema.
13. Eliminar asignatura por profesor.

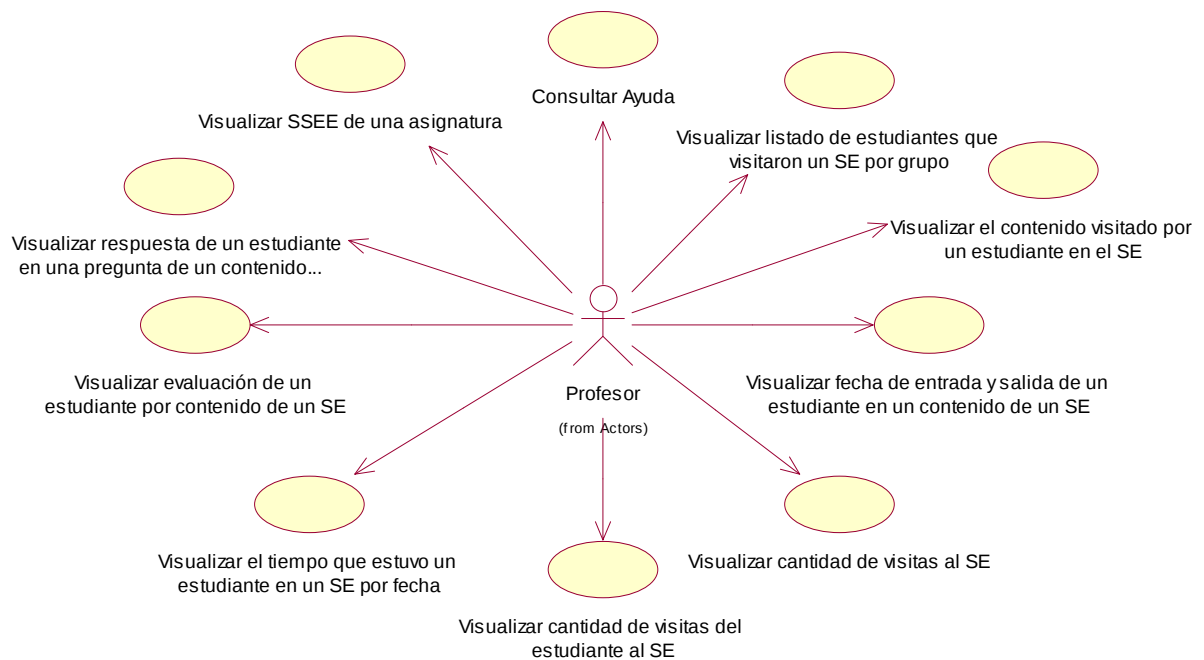


Fig 3 Diagrama de caso de Uso del Sistema: Paquete Administrativo.

El Paquete Reportes (Figura 4) contiene los casos de usos referentes a la información que podrá obtener el profesor a través del sistema, teniendo en cuenta el comportamiento de los estudiantes en cada SE:

1. Consultar ayuda.
2. Visualizar listado de estudiantes que visitaron un SE por grupo.
3. Visualizar el contenido visitado por un estudiante en el SE.

4. Visualizar fecha de entrada y salida de un estudiante en un contenido de un SE.
5. Visualizar cantidad de visitas al SE.
6. Visualizar cantidad de visitas del estudiante al SE.
7. Visualizar el tiempo que estuvo un estudiante en un SE por fecha.
8. Visualizar evaluación de un estudiante por contenido de un SE.
9. Visualizar pregunta y respuesta de un estudiante por contenido en un SE.
10. Visualizar SSEE de una asignatura.

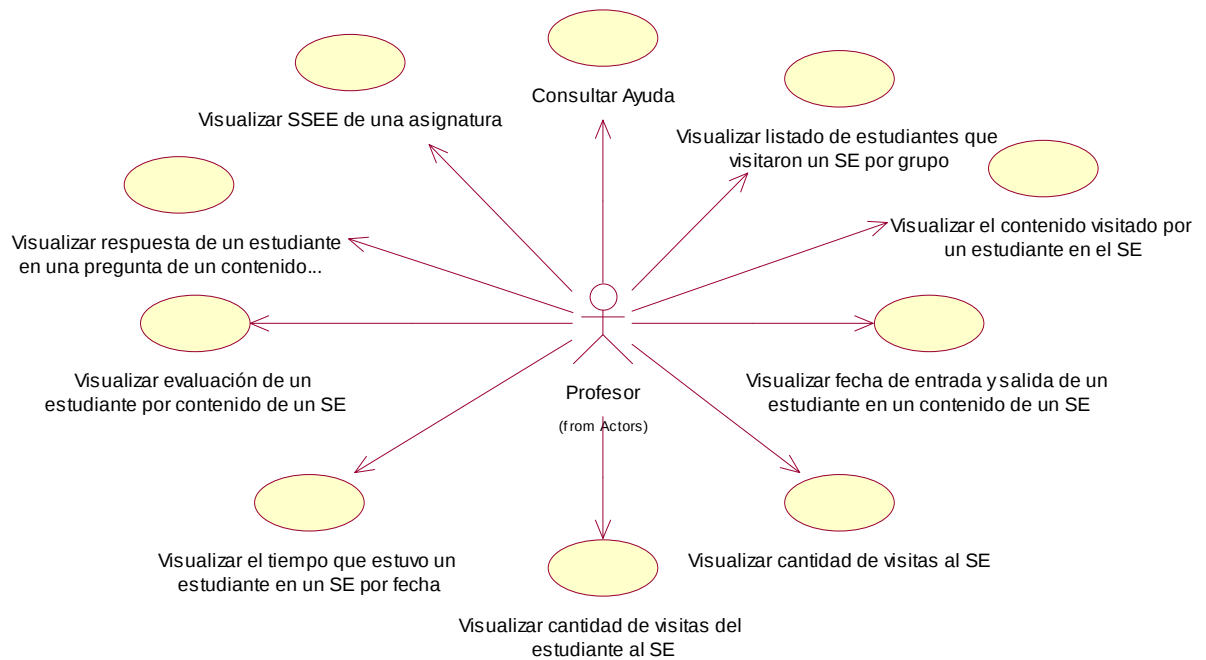


Fig 4 Diagrama de caso de Uso del Sistema: Paquete Reportes.

II.3.4 Descripción de los Casos de Uso del Sistema.

Caso de uso: Crear usuario
Actores: Administrador (inicia).
Propósito: Añadir nuevo usuario al sistema.
Resumen: El caso de uso se inicia cuando el administrador crea un nuevo usuario que necesite trabajar con la información que brinda el sistema, inserta el carnet de identidad, nombre y apellidos, dirección, cuenta, clave y la confirmación de la clave, para la creación de la cuenta de dicho usuario, si el usuario existe y no tiene cuenta creada inserta los datos para crearle la cuenta, si ya es un usuario del sistema, el mismo brinda una notificación de cuenta.
Precondiciones:
Referencias: R1
Poscondiciones: Queda registrado el usuario y listo para asignarle los permisos ante el sistema.
Prototipo: Ver Anexo 2.2

Tabla 3 Descripción del caso de uso de sistema: Crear usuario

Caso de uso: Habilitar usuario
Actores: Administrador (inicia).

Propósito: Activar la cuenta de un usuario del sistema.
Resumen: El caso de uso se inicia cuando el administrador desea habilitar la cuenta del usuario para el uso del sistema. Busca el usuario deseado por cualquiera de los criterios posibles nombre, primer apellido, carnet de identidad y cuenta, al mostrarse selecciona el usuario y lo activa terminando así el caso de uso.
Precondiciones: El usuario a habilitar debe estar previamente registrado como usuario del sistema.
Referencias: R3
Poscondiciones: Queda activada la cuenta del usuario.
Prototipo: Ver Anexo 2.4

Tabla 4 Descripción del caso de uso de sistema: Habilitar usuario

Caso de uso: Inhabilitar usuario
Actores: Administrador (inicia).
Propósito: Inhabilitar la cuenta de un usuario del sistema.
Resumen: El caso de uso se inicia cuando el administrador desea desactivar la cuenta del usuario para el uso del sistema. Busca el usuario deseado por cualquiera de los criterios posibles nombre, primer apellido, carnet

de identidad y cuenta, al mostrarse selecciona el usuario y lo inhabilita terminando así el caso de uso.
Precondiciones: El usuario a inhabilitar debe estar previamente registrado como usuario del sistema.
Referencias: R3
Poscondiciones: Queda desactivada la cuenta del usuario.
Prototipo: Ver Anexo 2.4

Tabla 5 Descripción del caso de uso de sistema: Inhabilitar usuario.

Caso de uso: Asignar permisos a usuarios
Actores: Administrador (inicia).
Propósito: Dar a los usuarios privilegios para que puedan tener acceso a las funcionalidades del sistema.
Resumen: El caso de uso se inicia cuando el administrador le va a asignar permisos a los usuarios del sistema, se muestran los permisos posibles, estos permisos pueden ser de Administración, Registro de Software Educativo, Cambio de Contraseña y Archivo, selecciona los permisos deseados para ese usuario y quedan habilitados solamente los seleccionados.
Precondiciones:

El usuario al que se le van a asignar los permisos debe estar previamente registrado como usuario del sistema.
Referencias: R4
Poscondiciones: El usuario al que le fueron otorgados los permisos solo podrá acceder a la información a la cual se le seleccionó como permisos.
Prototipo: Ver Anexo 2.4

Tabla 6 Descripción del caso de uso de sistema: Asignar permisos a usuarios.

Caso de uso: Asignar asignatura al profesor
Actores: Administrador (inicia).
Propósito: Asignar asignatura al profesor
Resumen: El caso de uso se inicia cuando el Administrador le va asignar una asignatura al profesor. Se muestran las asignaturas que tiene asignada en cada carrera y si desea agregar una nueva, selecciona la carrera y el año y se muestran las asignaturas correspondientes a estas para seleccionar y así asignarla al profesor.
Precondiciones: El profesor tiene que ser usuario del sistema.
Referencias: R5
Poscondiciones:

El profesor puede utilizar los softwares educativos pertenecientes a la asignatura asignada.
Prototipo: Ver Anexo 2.6

Tabla 7 Descripción del caso de uso de sistema: Asignar asignatura al profesor.

Caso de uso: Buscar usuario
Actores: Administrador (inicia).
Propósito: Buscar Usuario.
Resumen: El caso de uso se inicia cuando el administrador desea buscar un usuario para trabajar con sus datos propios. Luego aparece la información solicitada culminando así el caso de uso.
Precondiciones: La persona a buscar debe ser usuario del sistema.
Referencias: R6
Poscondiciones:
Prototipo: Ver Anexo 2.3

Tabla 8 Descripción del caso de uso de sistema: Buscar usuario.

Caso de uso: Cambiar contraseña
Actores: Administrador (inicia).

Propósito: Cambiar la contraseña del sistema a usuarios que no tengan asignados ese privilegio.
Resumen: El caso de uso se inicia cuando el administrador desea cambiar la contraseña de un usuario del sistema que no tiene permiso a cambiar su contraseña, el sistema le da la posibilidad de poner la cuenta del usuario, la contraseña nueva y confirmarla, si los datos son correctos la contraseña queda cambiada, finalizando así el caso de uso; de lo contrario se muestra un mensaje de error.
Precondiciones: Debe existir información almacenada del usuario que se desea cambiar la contraseña
Referencias: R7
Poscondiciones: Queda actualizada la contraseña del usuario
Prototipo: Ver Anexo 2.7

Tabla 9 Descripción del caso de uso de sistema: Cambiar contraseña

Caso de uso: Autenticar usuario
Actores: Profesor o Administrador (inicia).
Propósito: Registrarse en el sistema.
Resumen: El caso de uso se inicia cuando el profesor o administrador desean

acceder a información o servicios que se brindan de acuerdo a los permisos asignados a los mismos, para esto debe introducir su cuenta y contraseña, el sistema verifica en la base de datos si los mismos son correctos y pasa a mostrar su perfil iniciando su sesión, culminando así el caso de uso.
Precondiciones: Debe existir un usuario con cuenta y contraseña correcta.
Referencias: R8
Poscondiciones: El usuario puede acceder a toda la información a la que tiene permiso.
Prototipo: Ver Anexo 2.1

Tabla 10 Descripción del caso de uso de sistema: Autenticar usuario

Caso de uso: Cambiar contraseña usuario registrado
Actores: Administrador o Profesor (inicia).
Propósito: Cambiar la contraseña.
Resumen: El caso de uso se inicia cuando el administrador o el profesor desea cambiar su clave de acceso al sistema, introduciendo su cuenta, la clave anterior, la nueva contraseña y la confirmación de la misma, si los datos son correctos la contraseña queda cambiada, de lo contrario se muestra un mensaje de error, finalizando así el caso de uso.
Precondiciones:

- Debe existir información almacenada de este usuario. - El usuario debe tener asignado el permiso de cambio de contraseña
Referencias: R9
Poscondiciones: Queda actualizada la contraseña del usuario.
Prototipo: Ver Anexo 2.8

Tabla 11 Descripción del caso de uso de sistema: Cambiar contraseña usuario registrado

Caso de uso: Asignar SE a asignatura
Actores: Profesor (inicia).
Propósito: Asignar un Software Educativo a una asignatura
Resumen: El caso de uso se inicia cuando el profesor desea registrar un nuevo SE a las asignaturas a las que tiene acceso. Se muestran las posibles asignaturas a seleccionar para asignar un SE, el profesor debe introducir el nombre del nuevo SE, además se muestran los que tiene asignado cada asignatura.
Precondiciones: El profesor debe tener acceso a la asignatura en la que desea .registrar el software.
Referencias: R10

Poscondiciones:
Queda registrado un nuevo SE en la asignatura seleccionada.
Prototipo: Ver Anexo 2.9

Tabla 12 Descripción del caso de uso de sistema: Asignar SE a asignatura.

Caso de uso: Editar nombre de SE
Actores: Administrador o Profesor (inicia).
Propósito: Cambiar el nombre del Software Educativo.
Resumen: El caso de uso se inicia cuando el administrador o el profesor desean cambiar el nombre de un software educativo registrado. Seleccionan el SE a editar le cambian el nombre y lo salvan.
Precondiciones: El software educativo que se va a editar tiene que estar previamente registrado.
Referencias: R11
Poscondiciones: Queda actualizado el nombre del software educativo.
Prototipo: Ver Anexo 2.10

Tabla 13 Descripción del caso de uso de sistema: Editar nombre de SE

Caso de uso: Eliminar SE del sistema.
Actores: Profesor (inicia).
Propósito: Eliminar Software Educativo.
Resumen: El caso de uso se inicia cuando el profesor desea eliminar un software educativo registrado. Selecciona el SE y lo elimina, culminando así el caso de uso.
Precondiciones: El software educativo que se va a eliminar tiene que estar previamente registrado.
Referencias: R12
Poscondiciones: El Software Educativo queda eliminado de la asignatura en la que estaba registrado.
Prototipo: Ver Anexo 2.11

Tabla 14 Descripción del caso de uso de sistema: Eliminar SE del sistema.

Caso de uso: Eliminar asignatura por profesor.
Actores: Administrador (inicia).
Propósito: Denegar el acceso de un profesor a una asignatura.
Resumen:

El caso de uso se inicia cuando el administrador desea quitar el acceso de un profesor a una determinada asignatura, en la que no va a impartir clases. Se muestran las asignaturas que tiene asignada en cada carrera y se selecciona la que se va a eliminar.
Precondiciones: El profesor tiene que tener permiso a la asignatura que se le desea eliminar el acceso.
Referencias: R13
Poscondiciones: El profesor queda sin acceso a la asignatura que se le quito el permiso.
Prototipo: Ver Anexo 2.5

Tabla 15 Descripción del caso de uso de sistema: Eliminar asignatura por profesor

Caso de uso: Consultar ayuda
Actores: Profesor o Administrador (inicia).
Propósito: Consultar ayuda general del sistema.
Resumen: El caso de uso se inicia cuando un usuario del sistema desea informarse acerca del uso general del sistema, se muestra la ayuda referente al tema solicitado culminado así el caso de uso.
Precondiciones:
Referencias: R14

Poscondiciones:
Prototipo: Ver Anexo 2.18

Tabla 16 Descripción del caso de uso de sistema: Consultar ayuda

Caso de uso: Visualizar listado de estudiantes que visitaron un SE por Grupo.
Actores: Profesor (inicia).
Propósito: Mostrar los estudiantes por grupos que visitaron un SE.
Resumen: El caso de uso se inicia cuando un profesor solicita visualizar el listado de estudiantes por grupos que visitaron un software educativo, culminando el caso de uso mostrando el listado con el correo de cada estudiante.
Precondiciones: -El grupo del que se desea visualizar el listado tiene que tener acceso al software educativo solicitado.
Referencias: R15
Poscondiciones:
Prototipo: Ver Anexo 2.13

Tabla 17 Descripción del caso de uso de sistema: Visualizar listado de estudiantes que visitaron un SE por Grupo.

Caso de uso: Visualizar el contenido visitado por un estudiante en el SE.
Actores: Profesor (inicia).
Propósito: Mostrar el contenido que visitó un estudiante en un SE.
Resumen: El caso de uso se inicia cuando un profesor desea visualizar el contenido visitado por un estudiante en un Software Educativo.
Precondiciones: -El estudiante debe haber visitado al menos un contenido del SE.
Referencias: R16
Poscondiciones:
Prototipo: Ver Anexo 2.14

Tabla 18 Descripción del caso de uso de sistema: Visualizar el contenido visitado por un estudiante en el SE.

Caso de uso: Visualizar fecha de entrada y salida de un estudiante en un contenido de un SE.
Actores: Profesor (inicia).
Propósito: Mostrar la fecha de entrada y salida de un estudiante en un contenido de un SE.
Resumen: El caso de uso se inicia cuando un profesor desea visualizar la fecha de la entrada y salida de un estudiante a un contenido de un SE.
Precondiciones:

-El estudiante debe haber visitado al menos un contenido del SE.
Referencias: R17
Poscondiciones:
Prototipo: Ver Anexo 2.15

Tabla 19 Descripción del caso de uso de sistema: Visualizar fecha de entrada y salida de un estudiante en un contenido de un SE.

Caso de uso: Visualizar cantidad de visitas al SE.
Actores: Profesor (inicia).
Propósito: Mostrar la cantidad de veces que ha sido visitado un SE.
Resumen: El caso de uso se inicia cuando un profesor solicita visualizar las veces que han visitado el software los estudiantes.
Precondiciones:
Referencias: R18
Poscondiciones:
Prototipo: Ver Anexo 2.12

Tabla 20 Descripción del caso de uso de sistema: Visualizar cantidad de visitas al SE.

Caso de uso: Visualizar cantidad de visitas del estudiante al SE
Actores: Profesor (inicia).

Propósito: Mostrar la cantidad de veces que un estudiante visitó el software educativo.
Resumen: El caso de uso se inicia cuando un profesor solicita visualizar las veces que accedió un estudiante a un software educativo de una asignatura, culminando el caso de uso mostrando lo solicitado.
Precondiciones: El estudiante debe haber visitado al menos una vez el SE.
Referencias: R19
Poscondiciones:
Prototipo: Ver Anexo 2.14

Tabla 21 Descripción del caso de uso de sistema: Visualizar cantidad de visitas del estudiante al SE

Caso de uso: Visualizar el tiempo que estuvo un estudiante en un SE por fecha.
Actores: Profesor (inicia).
Propósito: Mostrar el tiempo que se demoró un estudiante en recorrer el software educativo.
Resumen: El caso de uso se inicia cuando un profesor desea visualizar el tiempo, que estuvo un Estudiante en el SE en cada fecha que lo visitó culminando el caso de uso imprimiendo en pantalla la fecha, hora de

entrada y el tiempo de cada visita.
Precondiciones: El estudiante debe haber navegado en el software educativo.
Referencias: R20
Poscondiciones:
Prototipo: Ver Anexo 2.17

Tabla 22 Descripción del caso de uso de sistema: Visualizar el tiempo que estuvo un estudiante en un SE por fecha.

Caso de uso: Visualizar evaluación de un estudiante por contenido en un SE.
Actores: Profesor (inicia).
Propósito: Mostrar la Evaluación de un estudiante en un SE.
Resumen: El caso de uso se inicia cuando un profesor desea visualizar la evaluación de un estudiante en un contenido de un SE. Selecciona la asignatura de la carrera, mostrándose los SE de dicha asignatura, los grupos en los que el profesor imparte clases, seleccionando el grupo se muestran los estudiantes de ese grupo que visitaron el SE seleccionado, escogiendo el estudiante a analizar se visualizan los contenidos visitados por el mismo, las fechas de cada contenido y finalmente la evaluación para cada fecha en que el estudiante respondió la pregunta de un contenido determinado, en caso que el contenido presente preguntas evaluativas.

Precondiciones: El estudiante debe haber navegado en una SE evaluativo como mínimo una vez.
Referencias: R21
Poscondiciones:
Prototipo: Ver Anexo 2.16

Tabla 23 Descripción del caso de uso de sistema: Visualizar evaluación de un estudiante por contenido en un SE.

Caso de uso: Visualizar respuesta de un estudiante en una pregunta de un contenido de un SE.
Actores: Profesor (inicia).
Propósito: Mostrar la pregunta y respuesta de un estudiante por contenido en un SE.
Resumen: El caso de uso se inicia cuando un profesor desea visualizar la respuesta de un estudiante en una pregunta de un contenido de un SE. Mostrándose la pregunta, la respuesta ideal y real terminando así el caso.
Precondiciones: El estudiante debe haber navegado en una SE evaluativo como mínimo una vez.
Referencias: R22

Poscondiciones:
Prototipo: Ver Anexo 2.16

Tabla 24 Descripción del caso de uso de sistema: Visualizar respuesta de un estudiante en una pregunta de un contenido de un SE.

Caso de uso: Visualizar fecha de entrada y salida de un estudiante al SE.
Actores: Profesor (inicia).
Propósito: Mostrar la fecha de entrada y salida de un estudiante en un SE.
Resumen: El caso de uso se inicia cuando un profesor desea visualizar la fecha de la entrada y salida de un estudiante a un SE de una asignatura, culminando el caso de uso mostrando lo solicitado.
Precondiciones: El estudiante debe haber navegado en el software educativo como mínimo una vez.
Referencias: R23
Poscondiciones:
Prototipo: Ver Anexo 2.23

Tabla 25 Descripción del caso de uso de sistema: Visualizar fecha de entrada y salida de un estudiante al SE.

Caso de uso: Visualizar SSEE de una asignatura.
Actores: Profesor. (inicia)
Propósito: Mostrar los Softwares Educativos que tiene asignado una asignatura.
Resumen: El caso de uso se inicia cuando un profesor desea visualizar los Software Educativos que tiene asignado una asignatura en la que él imparte clases, culminando el caso de uso mostrando lo solicitado.
Precondiciones: La asignatura debe tener asignado al menos un SE.
Referencias: R24
Poscondiciones:
Prototipo: Ver Anexo 2.12

Tabla 26 Descripción del caso de uso de sistema: Visualizar SSEE de una asignatura.

II.4 Factibilidad Económica

“Para llevar a cabo un buen proyecto de desarrollo de software, debemos comprender el ámbito del trabajo a realizar, los recursos requeridos, las tareas a ejecutar, las referencias a tener en cuenta, el esfuerzo (COSTE) a emplear y la agenda a seguir”. **[37]**

Para determinar si la construcción de un software es factible o no se han introducido en la Ingeniería del Software una serie de técnicas utilizadas dentro de

las tareas de planificación, que ayudan a planificar y controlar el esfuerzo y el tiempo necesario de desarrollo:

- Técnicas de estimación del esfuerzo (costo) de desarrollo. Dentro de las cuales se sitúa el Modelo Constructivo de Costes (Constructive Cost Model (COCOMO)).
- Técnicas de planificación y seguimiento de proyectos. **[38]**

“La estimación es una de las primeras actividades de la gestión de proyectos informáticos. Se le define como la predicción del personal, del esfuerzo, de los costos y del tiempo que se requerirán para realizar todas las actividades y construir todos los productos asociados con el proyecto. Su objetivo es conocer en etapas tempranas y de manera aproximada, el costo, la duración y los recursos necesarios para el desarrollo de proyectos de software. Se trata de una apreciación del futuro y la exactitud con la que ésta se realice, depende la mayoría de las veces de una buena herramienta de estimación, de la experiencia del estimador y del acceso a una base de información histórica de los proyectos.”**[39]**

II.4.1 Planificación por puntos de función.

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Insertar nuevo usuario	2	9	Medio
Buscar usuario	2	8	Medio
Insertar permisos usuario	2	5	Medio
Insertar permisos profesor	2	5	Medio
Insertar asignatura al profesor	8	15	Alto
Eliminar asignatura al profesor	1	2	Bajo
Iniciar sesión	4	14	Alto
Modificar contraseña	1	1	Bajo
Modificar contraseña usuario registrado	1	1	Bajo
Insertar SE	2	3	Bajo
Modificar SE	1	2	Bajo
Eliminar SE	1	2	Bajo

Tabla 27 Planificación: Entradas Externas

Nombre de la salida externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Mostrar permiso usuario	2	8	Medio
Mostrar permiso profesor	2	8	Medio
Mostrar asignatura del profesor	1	2	Bajo
Mostrar SE	5	6	Alto
Mostrar listado de estudiantes por grupo por SE	4	10	Alto
Mostar asignatura por año	5	7	Bajo
Mostrar contenido visitado por estudiante	2	6	Medio
Mostrar cantidad de visitas por estudiante	1	2	Bajo
Mostar cantidad de visitas por SE	1	1	Bajo
Mostrar pregunta y respuesta	3	16	Medio
Mostrar fecha de entrada y salida al contenido por estudiante	1	5	Bajo
Mostar el tiempo de cada vez que el estudiante visita un contenido	1	5	Bajo
Mostrar evaluación por contenido por estudiante	3	16	Bajo

Tabla 28 Planificación: Salidas Externas

Nombre de la petición	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Obtener carreras	1	3	Bajo
Obtener años por carrera	3	5	Bajo
Obtener asignatura por año	5	7	Alto
Obtener asignatura por profesor	5	8	Alto
Obtener asignatura por carrera	5	8	Alto
Obtener grupo por carrera	1	4	Bajo
Obtener software por asignatura	3	4	Bajo
Obtener si hay pregunta	1	8	Bajo

Tabla 29 Planificación: Peticiones

Nombre del fichero interno	Cantidad de records	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
cod_actividadlaboral	1	2	Bajo
cod_cargo	1	2	Bajo

cod_claseestudiante	1	2	Bajo
cod_conceptoingreso	1	2	Bajo
cod_municipio	1	4	Bajo
cod_nacionalidad	1	2	Bajo
cod_niveleducacional	1	2	Bajo
cod_paises	1	2	Bajo
cod_procedenciaescolar	1	2	Bajo
cod_provincia	1	3	Bajo
cod_raza	1	2	Bajo
cod_sexos	1	2	Bajo
cod_situacionescolar	1	2	Bajo
cod_tipoevaluacion	1	2	Bajo
cod_viaingreso	1	2	Bajo
estado	1	2	Bajo
subestado	1	3	Bajo
tb_annos	1	3	Bajo
tb_area	1	4	Bajo
tb_asignatura	1	7	Bajo
tb_asignaturaxmultimedia	1	2	Bajo

tb_asignaturaxprofesor	1	2	Bajo
tb_carreras	1	3	Bajo
tb_contenido	1	5	Bajo
tb_contenidoxestudiante	1	8	Bajo
tb_disciplinas	1	4	Bajo
tb_estudiante	1	3	Bajo
tb_estudiantexmultimedia	1	6	Bajo
tb_grupo	1	4	Bajo
tb_modulos	1	3	Bajo
tb_multimedia	1	2	Bajo
tb_niveles	1	2	Bajo
tb_operadores	1	4	Bajo
tb_operadorxmodulos	1	3	Bajo
tb_persona	1	15	Bajo
tb_planestudio	1	5	Bajo
tb_preasignatura	1	5	Bajo
tb_pregunta	1	3	Bajo
tb_profesor	1	1	Bajo
tb_semestre	1	4	Bajo

tb_sesiones	1	4	Bajo
tb_subarea	1	5	Bajo
tb_trabajador	1	2	Bajo

Tabla 30 Planificación: Ficheros Internos.

Elementos	Bajos	X Peso	Medios	X Peso	Altos	X Peso	Subtotal de puntos de función
Entradas externas	6	3	4	4	2	6	46
Salidas externas	7	4	4	5	2	7	62
Peticiones	5	3	0	4	3	6	
Ficheros lógicos internos	43	7	0	10	0	15	301
Ficheros de interfaces externas	0	5	0	7	0	10	0
Total de Puntos de Función No Ajustados (TPFNA)							409

Tabla 31 Planificación: Puntos de Función

Características			Valor
Puntos de función desajustados			409
Lenguaje	MySQL	PHP	Java Script
Instrucciones fuentes por puntos de función (SLOC*PFNA)	40	44	63
Por ciento de la aplicación en cuanto a requerimientos funcionales	30%	60%	10%
Instrucciones fuentes	122.7*40=4908	245.4*44=10797.6 10797	40.9*63=2576.7=2576
Total de instrucciones fuentes	18 281		

Tabla 32 Planificación: Miles de Instrucciones Fuertes

II.4.2 Determinación de los costos

Determinación de los valores de las variables de costos utilizadas en el cálculo de costos en la producción de software, como el cálculo del esfuerzo, el tiempo de desarrollo, la cantidad de hombres y el costo total del sistema.

Factores de Escala SF _j	Valor	Justificación
Precedentes (PREC)	3,72	La experiencia previa del equipo de desarrollo en este tipo de proyectos es casi sin precedentes, pues se han desarrollado aplicaciones para la Web pero no utilizando el modelo de tres capas

		(Nominal).
Flexibilidad (FLEX)	3,04	Refleja en el proceso de desarrollo flexibilidad entre la relación de las especificaciones de los requerimientos preestablecidos y las de la interfaz externa (Nominal).
Arquitectura/resolución del riesgo (RESL)	4, 24	Al analizar la amplitud de análisis de riesgo del producto a desarrollar y teniendo en cuenta la experiencia de nuestros profesionales en investigaciones de este tipo podemos decir que existen algunos factores de riesgo (Nominal).
Cohesión del Equipo (TEAM)	1,10	La relación entre los miembros del equipo que trabajarán en el desarrollo del producto es altamente cooperativa. (Muy alto)
Madurez del Proceso (PMAT)	6,24	El desarrollo del sistema se encuentra a un Nivel I (Alto), por lo que podemos decir que está en su primera etapa un poco avanzada. (Bajo)

Tabla 33 Costos: Factores de escalas

Factores de Ajuste	Valor	Justificación
Fiabilidad y complejidad del producto (RCPX)	1	El producto será fiable y tendrá una moderada complejidad pues estará basada en la Web y la base de datos (Nominal)

		BD moderada, no se requiere de amplia documentación. La aplicación Web tiene una moderada complejidad. (Nominal)
Requerimientos de reusabilidad (RUSE)	1,07	Se utilizara código reusable en toda la aplicación (Alto).
Dificultad de la plataforma (PDIF)	1	La plataforma de aplicación tiene gran estabilidad. No tiene grandes restricciones en cuanto al tiempo de ejecución ya que el software podrá estar trabajando varias horas. EL Software no tiene limitación de memoria impuesta (Nominal).
Capacidad del personal (PERS)	0,83	El personal dispone de una alta capacidad.
Experiencia del personal (PREX)	0,87	El equipo que desarrollará el software posee un buen dominio y conocimiento del lenguaje de programación y el diseño de base de datos. Con una experiencia de aproximadamente algunos años (Alto).
Facilidades para el desarrollo (FCIL)	0,87	Para el desarrollo del software se utilizarán en el diseño, la herramienta CASE Rational Rose 2003 utilizando UML como lenguaje de modelado y para la programación del mismo PHP como lenguaje, como SGBD se escogió MYSQL, servidor de aplicaciones web Apache, la utilización de estas herramientas facilitará el desarrollo del software (Alto).

Esfuerzo de calendario (SCED)	1	Para el desarrollo del producto se ha realizado una planificación del esfuerzo de calendario alto, por parte del equipo que desarrollará el sistema (Alto).
----------------------------------	---	---

Tabla 34 Costos: Factores de Ajuste

Multiplicador de esfuerzos

$$EM = \prod_{i=1}^n E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED$$

i=1

$$EM = 1 * 1,07 * 1 * 0,83 * 0,87 * 0,87 * 1 = 0,67$$

Factores de Escala

$$SF_j = \sum SF_j = PREC + FLEX + RESL + TEAM + PMAT$$

$$SF_j = 3,72 + 3,04 + 4,24 + 1,10 + 6,24 = 18,34$$

Valores de los coeficientes

$$A = 2,94; B = 0,91; C = 3,67; D = 0,24$$

$$E = B + 0,01 * SF_j$$

$$F = D + 0,2 * (E - B)$$

$$E = 0,91 + 0,01 * 18,34$$

$$F = 0,24 + 0,2 * (1,0934 - 0,91)$$

$$E = 1,0934$$

$$F = 0,27668$$

Esfuerzo

$$PM = A * (MF)^E * EM$$

Donde: MF es miles de instrucciones fuentes

$$MF = 18,281 \approx 18,2$$

$$PM = 2,94 * (18,2)^{1,0934} * 0,67$$

$$PM = 2,94 * 23,86 * 0,67$$

$$PM = 46,99$$

Cálculo del tiempo de desarrollo

$$TDEV = C * PM^F$$

$$TDEV = 3,67 * (46,99)^{0,27668}$$

$$TDEV = 10,64 \approx 11 \text{ meses}$$

Cálculo de la cantidad de hombres

$$CH = PM / TDEV$$

$$CH = 46,99 / 10,64$$

$$CH = 4,41 \approx 4 \text{ hombres}$$

Costo

Se asume como salario promedio mensual \$ 225

$$CHM = 4 * \text{Salario Promedio}$$

$$CHM = 4 * 225$$

$$CHM = 900 \$/\text{mes}$$

$$\text{Costo} = CHM * PM$$

$$\text{Costo} = \$ 900 * 46,99$$

$$\text{Costo} = \$ 42.291$$

II.4.3 Beneficios tangibles e intangibles

Los beneficios obtenidos con el desarrollo del software son fundamentalmente intangibles, pues permite al profesor tener un control del uso de los Softwares Educativos por parte de los estudiantes y poder emitir un criterio evaluativo para cada uno de ellos además posibilita realizar un tratamiento individualizado atendiendo a las especificidades de cada estudiante. Por otra parte el sistema estimulará su uso ya que se diseñó con tecnologías de punta teniendo en cuenta el empleo de las nuevas tecnologías de informática y las comunicaciones aplicadas a la educación.

II.4.4 Análisis de costos y beneficios

Al desarrollo de todo producto informático va asociado a un costo, el justificarlo depende de los beneficios tangibles e intangibles que produce. En el caso de la aplicación a desarrollar se llegó a la conclusión que se logra aproximadamente en 11 meses con una fuerza de trabajo de 4 hombres con salario promedio de \$225 pesos por trabajador para un costo total de \$42 291.

Conclusiones

En el presente capítulo se representa y describe la solución propuesta del sistema, utilizando el modelo del dominio donde se definen los diferentes conceptos referente a este y sus relaciones, lo que llevó a lograr una mejor comprensión del problema a resolver y permitió con una mayor claridad hacer mención a los requerimientos funcionales y no funcionales así como a los casos de uso con su respectiva descripción.

La herramienta propuesta dispone de una serie de beneficios y posibilidades, sobre todo intangibles, que propician una eficiencia en la organización y control de la información, lo cual contribuirá al desarrollo y perfeccionamiento de la educación por lo que se considera factible implementar la herramienta propuesta.

Capítulo III: “Diseño de la Solución Propuesta”

En este capítulo se muestra el diseño de clases web que se propone para desarrollar el sistema y los diagramas necesarios para la implementación de la solución propuesta.

III.1 Principios de Diseño del Sistema

El **diseño de sistemas** se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física. [40]

III.1.1 Estándares en la Interfaz de la Aplicación

La primera impresión del usuario cuando visita una aplicación web la brinda el diseño de la interfaz. Es por ello que, para lograr la apariencia adecuada y que el usuario se sienta confortable, se tienen en cuenta varios aspectos, sobre todo relacionados con tipografía, colores, gráficos, navegación, composición del sitio, etc., que a continuación se detallan. En el sistema, el diseño de la interfaz está basado en páginas Web, se utilizan las tonalidades suaves y refrescantes. El vocabulario manejado es lo menos técnico posible, acercándose al utilizado por los usuarios.

Se utilizan imágenes identificativas como vínculos para la navegación dentro del sitio web. La letra utilizada en todo el sistema es Times New Roman (12, 16) lográndose un diseño estándar en todo el sitio. Los mensajes de error son pequeños y en Español.

Se utilizan pequeños íconos para una mayor comprensión de las acciones, aunque se seleccionaron imágenes consecuentes con el significado que se quiere transmitir.

El fondo de las páginas es de color blanco para mayor frescura de la vista. Todo esto se ha hecho con el objetivo de que el uso del sitio brinde comodidad y confort al usuario.

III.1.2 Concepción General de la Ayuda

La ayuda constituye una parte imprescindible en todo sistema. En el menú principal aparece una opción *Ayuda* que explicará de forma detallada cómo funciona el sistema, tratando de aclarar los puntos que podría causar duda al usuario. Este menú aparece en todas las páginas para mayor comodidad del usuario.

Cada una de las opciones del sistema, así como las consideraciones que se asumen en la ejecución de ellas están propiamente documentadas para evitar cualquier tipo de confusión por parte del usuario. Cada aspecto de la ayuda ha sido diseñado con el objetivo de expresar explícitamente cómo y en qué orden debe operar el usuario.

III.1.3 Tratamiento de Excepciones

El diseño de la interfaz ha estado dirigido a evitar errores, teniendo en cuenta paralelamente la creación de interfaces útiles y amigables. Se ha buscado simplificar la validación de los datos garantizando una validación intrínseca de los mismos, procurando facilitar la corrección de errores lógicos tanto en la introducción de la información como en cualquier otro momento del tratamiento de la misma.

La técnica para el manejo de los errores en el sistema se concebirá de manera que cuando ocurra un error se genere una excepción; es decir, la ejecución normal se detenga y se transfiera el control a la zona de tratamiento de excepciones. Las excepciones internas se generan automáticamente por el sistema.

Los mensajes de error que emita el sistema ya sea de la base de datos o de la aplicación cliente se captarán y se traducirán a un lenguaje comprensible para el usuario.

Los formularios manejan los datos en memoria y solo se actualiza en la base de datos cuando se indique salvarlos.

III.2 Modelo de Clases Web

Un diagrama de clases Web representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. [41] Al tratar de utilizar el diagrama de clases tradicionales para modelar aplicaciones Web surgen varios problemas, por lo cual los especialistas del Rational plantearon la creación de una extensión al modelo de análisis y diseño que permitiera representar el nivel de abstracción adecuado y la relación con los restantes artefactos de UML.

Caso de Uso	Diagrama de Clases
Crear usuario.	Anexo 3.1
Habilitar usuario.	Anexo 3.2
Inhabilitar usuario.	Anexo 3.3
Asignar permisos a usuario.	Anexo 3.4
Asignar asignatura al profesor.	Anexo 3.5
Buscar usuario.	Anexo 3.6
Cambiar contraseña.	Anexo 3.7
Autenticar usuario.	Anexo 3.8
Cambiar contraseña usuario registrado.	Anexo 3.9
Asignar SE a asignatura.	Anexo 3.10
Editar nombre de SE.	Anexo 3.11
Eliminar SE del sistema.	Anexo 3.12

Eliminar asignatura por profesor.	Anexo 3.13
-----------------------------------	------------

Tabla 35 Diagrama de Clases Web. Paquete Administrativo

Caso de Uso	Diagrama de Clases
Consultar ayuda.	Anexo 3.14
Visualizar listado de estudiantes que visitaron un SE por grupo.	Anexo 3.15
Visualizar el contenido visitado por un estudiante en el SE.	Anexo 3.16
Visualizar fecha de entrada y salida de un estudiante en un contenido de un SE.	Anexo 3.17
Visualizar cantidad de visitas al SE.	Anexo 3.18
Visualizar cantidad de visitas del estudiante al SE.	Anexo 3.19
Visualizar el tiempo que estuvo un estudiante en un SE por fecha.	Anexo 3.20
Visualizar evaluación de un Estudiante por contenido de un SE.	Anexo 3.21
Visualizar respuesta de un estudiante en una pregunta de un contenido de un SE.	Anexo 3.22
Visualizar fecha de entrada y salida de un estudiante al SE.	Anexo 3.23
Visualizar SSEE de una asignatura.	Anexo 3.24

Consultar ayuda.	Anexo 3.25
Visualizar listado de Estudiantes que visitaron un SE por Grupo.	Anexo 3.26
Visualizar el contenido visitado por un Estudiante en el SE.	Anexo 3.27
Visualizar fecha de entrada y salida de un Estudiante en un contenido de un SE.	Anexo 3.28
Visualizar cantidad de visitas al Software Educativo.	Anexo 3.29
Visualizar cantidad de visitas del estudiante al SE.	Anexo 3.30
Visualizar el tiempo que estuvo un estudiante en un SE por fecha.	Anexo 3.31
Visualizar evaluación de un estudiante por contenido de un SE.	Anexo 3.32
Visualizar Pregunta y Respuesta de un Estudiante por contenido en un SE.	Anexo 3.33
Visualizar Softwares Educativos de una Asignatura.	Anexo 3.34

Tabla 36 Diagrama de Clases Web. Paquete Reportes.

III.3 Diseño de la Base de datos.

En este epígrafe se muestra el diseño de la base de datos del sistema propuesto a través de los diagramas de clases persistente y el esquema de la base de datos generados a partir de este, con el modelo de datos.

III.3.1 Diagrama del modelo lógico de datos.

El diagrama de clases persistentes muestra todas las clases capaces de mantener su valor en el espacio y en el tiempo [42]. Se muestra en el Anexo 3.24.

III.3.2 Diagrama del modelo físico de datos.

El modelo de datos que muestra la estructura física de las tablas de la base de datos, obtenido a partir del diagrama de clases persistentes [43], es mostrado en el Anexo 3.25.

III.4 Diagrama de Implementación

El modelo de implementación denota la implementación del sistema en términos de componentes y subsistemas de implementación. Describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de la implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros. [44]

La figura 5 muestra el diagrama de implementación.

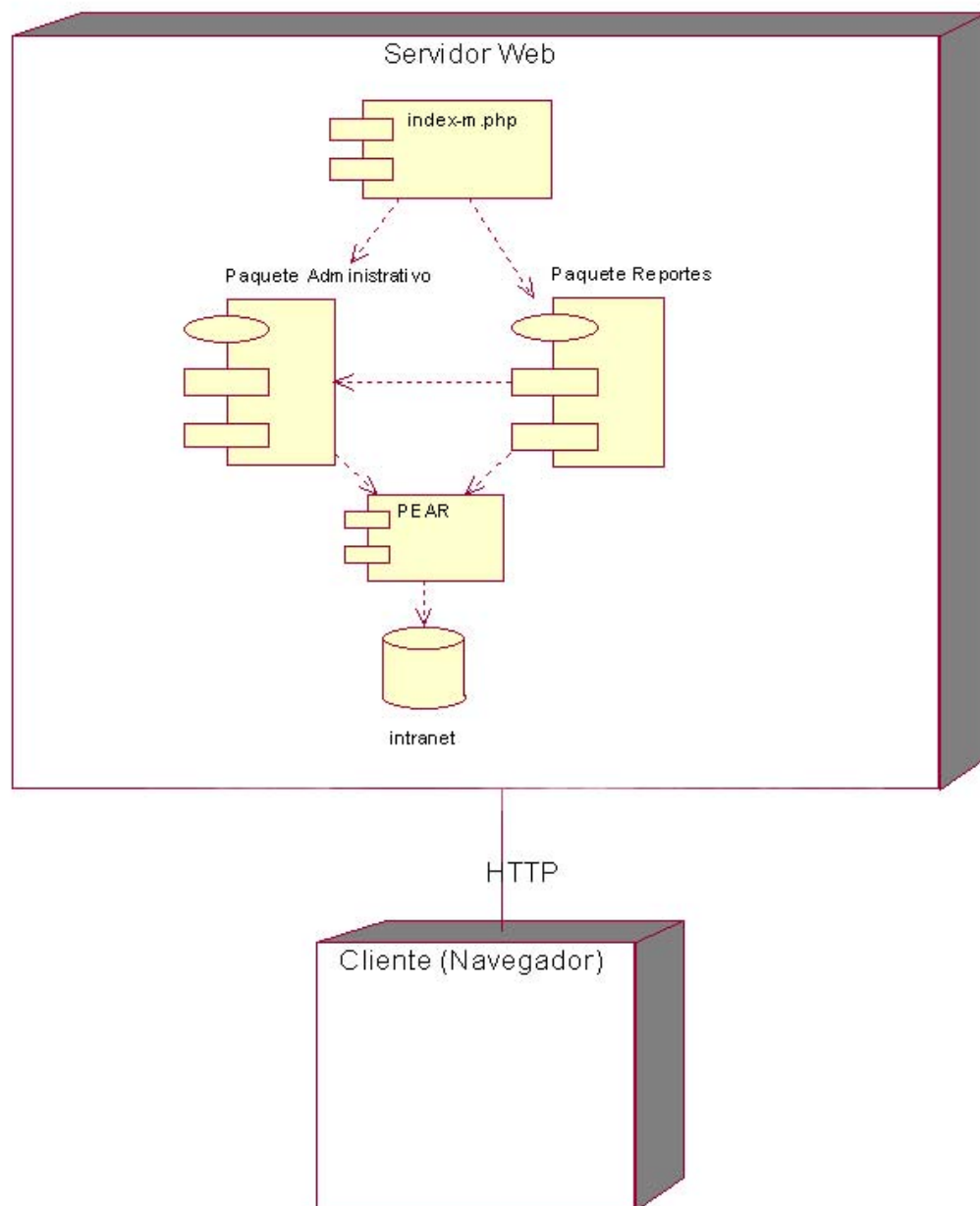


Fig 5 Diagrama de Implementación

Conclusiones

En el presente capítulo se muestran los diagramas de clases de la aplicación realizada, se describe el diseño de la base de datos y el diagrama de implementación, además, los principios de diseño seguidos, específicamente, los temas de estándares de la interfaz, concepción del tratamiento de excepciones, sistema de ayuda y principios de codificación. Se mostraron también los resultados de las etapas de diseño e implementación del sistema propuesto.

Conclusiones

Al finalizar el proyecto podemos llegar a las siguientes conclusiones:

- ✓ Se desarrolló un sistema informático que permite al profesor tener un control del uso de los Softwares Educativos disponibles en la intranet, y con dicho control el profesor puede obtener criterios que le permitan valorar el desempeño de los estudiantes, de esta manera puede llegar a conclusiones evaluativas con mayor fundamentación.
- ✓ Se realizó un estudio en detalles del uso de los Softwares Educativos por parte de los estudiantes, así como de las necesidades informativas de los profesores, permitiendo esto llegar a determinar los requisitos necesarios para este tipo de aplicación.
- ✓ Se diseñó un sistema independiente de la plataforma que puede ser usado en la intranet de cualquier centro educativo.
- ✓ Se utilizaron los conocimientos adquiridos durante los estudios realizados en la carrera de Ingeniería Informática tales como: metodología RUP para el desarrollo del proyecto, lenguaje UML para el modelado orientado a objeto y técnicas de programación con el lenguaje PHP.
- ✓ Adicionalmente se estudió a profundidad el modelo de tres capas, el uso y creación de servicios web y el lenguaje de descripción de datos XML.

Recomendaciones

A pesar de que los objetivos del trabajo se cumplieron se sugiere tomar esta propuesta como principio de un proyecto más abarcador y continuar esta investigación, por lo que se recomienda:

- ✓ Añadir técnicas de Inteligencia Artificial.
- ✓ Aplicar el sistema en otros centros de educación como escuelas primarias, secundarias y pre-universitarios.
- ✓ Extender la investigación para añadir otros datos de interés de los profesores sobre el uso de los Softwares Educativos por parte de los estudiantes.
- ✓ Aplicar el software utilizando otros Sistemas Gestores de Bases de Datos.
- ✓ Mantener la idea de que el sistema debe ser desarrollado sobre la base de la multiplataforma, lo que permitirá adaptarse a las nuevas tecnologías de la informática moderna.

[ACO 02] Acosta Duarte, Dionel A. *Registros Médicos Pediátricos*. Tesis presentada en opción al título de Master en Informática, Villa Clara, 2002.

[ARO 93] Aronow, D.B. *Information technology applications in quality assurance and quality improvement*, The Joint Commision Journal on Quality Improvement, Vol 19, No 9, EE.UU, Enero 1993.

[BEN 92] Bennatan, E.M. *Software Project Management: A Practitioner's Approach*. Editorial McGraw Hill, EE.UU, 1992.

[BOE 81] Boehm, B. *Software Engineering Economics*. Prentice Hall, EE.UU, 1981.

[BOO 96] Booch, Grady. *Análisis y Diseño Orientado a Objetos*. Ed. Addison-Wesley, EE.UU, 1996.

[DIA 97] Díaz Ferrera, Carlos N. *Sistema de Almacenamiento de la Información*. Editorial ENPES, Ciudad de la Habana, 1982.

[FAI 96] Fairley, Richard. *Ingeniería del Software*. Mc Graw-Hill, Mexico, 1996.

[FRO 89] Frost. *Bases de datos y sistemas expertos*, Bases de datos. Editorial "Evelio Rodríguez Curbelo", La Habana, 1989.

[JON 86] Jones, C. *Programming Productivity*. McGraw Hill, EE UU., 1986.

[PRE 98] Pressman, Roger. *Ingeniería de Software, un enfoque práctico*. Mc Graw-Hill, EE.UU., 1998.

[RUM 00] Rumbaugh, J. *El Lenguaje Unificado de Modelado*. Manual de Referencia. Pearson Education, Madrid, 2000.

SQL Server Books Online. Microsoft Press, EE.UU., 1998.

[TAY 89] Taylor, E.S. *An Interim Report on Engineering Design*. Massachuett Institute of Technology, EE.UU., Cambridge, 1989.

[ULL 83] Ullman, J.D. *"Principles of Database Systems"*. Computer Science Press, EE.UU., 1983.

[VIZ 98] Vizcarro, C. *Nuevas tecnologías para el aprendizaje*. Pirámide, Madrid, 1998.

[TEJ 96] Tejada Fernández, José. *"Informática e innovación educativa"*. Comunicación Educativa y Nuevas Tecnologías, Praxis, Barcelona, 1996.

[TEJ 1999] Tejada Fernández, José. *"Nuevas tecnologías y educación: consideraciones psicopedagógicas de selección, diseño y aplicación"*. Comunicación y Pedagogía, 1999.

[SIM 83] Simon, J. *La educación y la informatización de la sociedad*. Nancea, Madrid, 1998.

[SEV 98] Sevillano, María Luisa. *Nuevas Tecnologías, medios de comunicación y educación*, CCS, Madrid, 1998.

[SAN 96] Sancho Gil, Joana M^a. *"Educación en la era de la información"*. Revista Cuadernos de Pedagogía, Vol 253, No 253, Marzo, 1996.

[ROI 02] Roig, Rosabel. *Las Nuevas Tecnologías aplicadas a la educación*. Elementos para una articulación didáctica de las Tecnologías de la Información y la Comunicación. Alcoy, Marfil, 2002.

[ROD 94] Rodríguez Diéguez, José Luis. *"Nuevas Tecnologías para la Educación"*. Alfar, Sevilla, 1994.

[PÉR 98] Pérez Pérez, R. *"Nuevas tecnologías y nuevos modelos de enseñanza"*. CCS, Madrid, 1998.

[GUT 98] Gutierrez Martin, Alfonso. *"El profesor ante las nuevas tecnologías multimedia"*. Comunicación y Pedagogía, Vol 2, No 153,1998.

[FER 98] Fernández Muñoz, Ricardo. *"Nuevas tecnologías, educación y sociedad"*. Editorial CCS, Madrid, 1998.

Conceptos y evolución de la ingeniería del software <http://mondragon.angeltowns.net>. (10/01/07).

Software educativo. Metodología y criterio para su elaboración y evaluación. Mg. Mirtha Ramos <http://www.uned.ac.cr>. (10/01/07)

Declaración de Luis Ignacio Gómez en la página Web: <http://www.cubaminrex.cu> sobre los softwares educativos. (10/01/07).

,

- [1] Bartolomé, A. Nuevas Tecnologías aplicadas a la educación. EDUTEC.La Habana,1995.—p.295. Bartolomé, A. *Nuevas Tecnologías aplicadas a la educación*. EDUTEC, La Habana, 1995.

- [2] Melendez Sara, Guerrero *Informática Educativa*. <http://www.fmmeduacion.com.ar/Informatica/infoeduc.htm>. (12/01/2007)

- [3] Ramos, Mirtha. *Software educativo. Metodología y criterio para su elaboración y evaluación*. [http:// www.uned.ac.cr](http://www.uned.ac.cr) (7/01/2007)

- [4] Osuna Acedo Sara. *La clave para el paso de milenio*. <http://www.multimedia.com> (2/02/2007)

- [5] Declaración de Luis Ignacio Gómez. <http://www.cubaminrex.cu> (21/01/2007)

- [6] Torres Patricio, Letelier. *Desarrollo de Software Orientado a Objetos usando UML*. <http://www.creangel.com/uml/intro.php>.(12/05/2007)

- [7] Grau Xavier, Ferrá. *Desarrollo orientado a objetos con UML*. <http://www.clikear.com/manuales/uml/introduccion.asp>. (10/04/2007)

- [8] Díaz Antón, Maria Gabriela. *Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática*. <http://www.academia-interactiva.com/ise.pdf>. (5/5/2007)

- [9] Francia H, Joel. *Desarrollo de una Aplicación en tres Capas con VS .NET*.
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art140.asp>
(1/12/2006)

- [10] Francia H, Joel. *Desarrollo de una Aplicación en tres Capas con VS .NET*.
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art140.asp>
(1/12/2006)

- [11] Francia H, Joel. *Desarrollo de una Aplicación en tres Capas con VS .NET*.
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art140.asp>
(1/12/2006)

- [12] Francia H, Joel. *Desarrollo de una Aplicación en tres Capas con VS .NET*.
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art140.asp>
(1/12/2006)

- [13] Macromedia Dreamweaver MX. Getting Started. Ayuda. (20/02/07)

- [14] Conde, Jesús. AJAX. <http://www.illasaron.com/html/modules.php>

- [15] Sosa D, Hector K, *Desarrollo de un prototipo de intranet para una Facultad de un Centro de Educación Superior: Modulo Docente*. Trabajo de diploma para optar por el titulo de Ingeniero Informático. Universidad Carlos Rafael Rodríguez, Cienfuegos, junio 2006.

- [16] *Desarrollo ASP a fondo*. <http://www.desarrolloweb.com/asp/> (15/12/2006)

- [17] *Desarrollo ASP a fondo*. <http://www.desarrolloweb.com/jsp/> (15-12-2006)

- [18] *Introducción a php.* <http://www.ciberteca.net/webmaster/php>.(20/01/2007)
- [19] Rodríguez Luis, *Conoce a Pear: El Repositorio de Aplicaciones y Extensione de PHP* http://programacion.com/blogs/60_php_extensiones/php.htm .(4/01/07)
- [20] Sosa D, Hector K, *Desarrollo de un prototipo de intranet para una Facultad de un Centro de Educación Superior: Modulo Docente.* Trabajo de diploma para optar por el titulo de Ingeniero Informático. Universidad Carlos Rafael Rodríguez, Cienfuegos, junio 2006.
- [21] Pérez Sacristán, Juan Ignacio. *Programación. Soap.* <http://www.programacion.com/tutorial/xmlrpcsoap/2/> (1/02/2007)
- [22] Ibídem.
- [23] Ibídem.
- [24] Lichtmaier, Nicolás. *Introducción a XML.* <http://www.reloco.com.ar/xml.html> Montero (10/04/2007)
- [25] Ibídem.
- [26] Ibídem.
- [27] SOAP. Simple Object Access Protocol. <http://www.desarrolloweb.com/articulos/1853.php> (15/01/2007)

- [28] Sosa D, Hector K, *Desarrollo de un prototipo de intranet para una Facultad de un Centro de Educación Superior: Modulo Docente*. Trabajo de diploma para optar por el titulo de Ingeniero Informático. Universidad Carlos Rafael Rodríguez, Cienfuegos, junio 2006.

- [29] Sosa D, Hector K, *Desarrollo de un prototipo de intranet para una Facultad de un Centro de Educación Superior: Modulo Docente*. Trabajo de diploma para optar por el titulo de Ingeniero Informático. Universidad Carlos Rafael Rodríguez, Cienfuegos, junio 2006.

- [30] Matos, Rosa María. *Introducción al trabajo con Base de Datos. Asignatura de Sistemas de Gestión de Base de Datos*. CUJAE, Ciudad Habana, 2004.

- [31] Jacobson, I. *El Proceso Unificado de Desarrollo de software*. Addison-Wesley, EE.UU., 2000.

- [32] Ibídem.

- [33] Ibídem.

- [34] Ibídem.

- [35] Ibídem.

- [36] Ibídem.

- [37] COCOMO v2 *Modelo de Estimación de Costes para proyectos software*.
- [38] Ibídem.
- [39] Ovejero, Jose Daniel. *Estimación de proyectos para sistemas basados en conocimiento*. Universidad de Cienfuegos, Cienfuegos, 2006.
- [40] Jacobson, I. *El Proceso Unificado de Desarrollo de software*. Addison-Wesley, EE.UU., 2000.
- [41] Ibídem.
- [42] Ibídem.
- [43] Ibídem.
- [44] Ibídem.

Anexo 2.1 Prototipo Autenticar usuario.

Introduzca su clave de acceso

Cuenta:

Clave:



Solo para personal autorizado

Anexo 2.2 Prototipo Crear usuario.

Carnet de Identidad: *

Nombre: *

Primer Apellido: *

Segundo Apellido: *

Dirección: *

Cuenta: *

Clave: *

Confirmar Clave: *

(*) Campos Obligatorios

Anexo 2.3 Prototipo Buscar usuario

Buscar Usuario

Buscar por

- Buscar por
- Nombre
- Carnet Identidad
- PrimerApellido
- Cuenta

Anexo 2.4 Prototipo Asignar permisos a usuario, Habilitar e inhabilitar usuario.Nombre : **Isabel Liset González Monzón**Cuenta : **liset**Activo: ☒

Asignar Pemisos:

- ☒ Administrador
- ☒ Registro de Multimedia
- ☒ Cambio de Contraseña
- ☒ Archivo

Asig. Asignatura (s)

Salvar

Anexo 2.5 Prototipo Eliminar asignatura por profesor.**Asignatura(s) de: Kadir Hector Ortiz**

Asignatura(s)	Carrera(s)
☐ Programación I	Informática
☐ Inteligencia Artificial I	Informática
☐ Programación	Informática
☒ Historia	Informática

Agregar Asignatura

Eliminar

Anexo 2.6 Prototipo Asignar asignatura al profesor.**Kadir Hector Ortiz**

Carreras: Informatica

Año: 1

Asignaturas	
☒	Programación I
☐	Programación II

Asignar

Anexo 2.7 Prototipo Cambiar contraseña.

Nombre: Isabel Monzón Hernández

Cuenta: isabel

Clave: *

Confirmar Clave: *

Cambiar Contraseña

Anexo 2.8 Prototipo cambiar contraseña usuario registrado.

Nombre: Kadir Hector Ortiz

Cuenta: khector

Clave Anterior: *

Clave: *

Confirmar Clave: *

Cambiar Contraseña

Anexo 2.9 Prototipo Asignar Software Educativo a asignatura.

Asignatura:

Nombre Software:

Anexo 2.10 Prototipo Editar nombre de Software Educativo.

Lista de Software

Id	Nombre de Software(s)	Asignatura(s)
☐ 1	Delphi en 21 días Salvar Cancelar	Programación I
☐ 2	Algoritmos de Búsqueda	Inteligencia Artificial I
☐ 6	Historia de Cuba	Historia
☐ 7	Elementos Informáticos	Programación I
☐ 8	Análisis y Diseño	Programación I
☐ 9	RUP	Programación
☐ 10	Historia de Cienfuegos	Historia

Anexo 2.11 Prototipo Eliminar Software Educativo del sistema.

Lista de Software

Id	Nombre de Software(s)	Asignatura(s)
☐ 1	Delphi	Programación I
☐ 2	Algoritmos de Búsqueda	Inteligencia Artificial I
☐ 6	Historia de Cuba	Historia
☐ 7	Elementos Informáticos	Programación I
☐ 8	Análisis y Diseño	Programación I
☒ 9	RUP	Programación

Anexo 2.12 Prototipo Visualizar Softwares Educativos de una asignatura, Visualizar cantidad de visitas al software Educativo.

[Lista de Asignaturas / Software](#)

Carrera: Informática

Asignatura: Programación

Grupos:

Software(s) Educativo(s)	Visitas
RUP	4

Anexo 2.13 Prototipo Visualizar listado de estudiantes que visitaron un Software Educativo.

Nombre y Apellidos	Correo
Kadir Hector Ortiz	khector

Anexo 2.14 Prototipo Visualizar el contenido visitado por un estudiante en el SE, Visualizar cantidad de visitas del estudiante al Software Educativo.

[Lista de Asignaturas](#) / [Software](#) / [Contenido](#)

Contenido Visitado por: Kadir Hector Ortiz

RUP	Tiempo
Unidad: RUP	
- Modelo de dominio - Diagrama de Clase - Diagrama de Clase Web - Modelo de negocio - Casos de uso - Diagrama Caso de Uso - Diagrama de Casos de Usos del Negocio - Actores del Negocio	
Cantidad de Visitas al SE	3

Anexo 2.15 Prototipo Visualizar fecha de entrada y salida de un estudiante en un contenido de un Software Educativo.

Contenido Visitado por: Kadir Hector Ortiz

RUP

Tiempo

Unidad: RUP

Modelo de dominio

Fecha	Hora Entrada	Hora Salida	Pregunta
2007-03-27	11:33:54	11:46:54	
2007-04-29	02:45:12	03:45:12	

Diagrama de Clase

Diagrama de Clase Web

Modelo de negocio

Casos de uso

Diagrama Caso de Uso

Diagrama de Casos de Usos del Negocio

Actores del Negocio

Cantidad de Visitas al SE

3

Anexo 2.16 Prototipo Visualizar respuesta de un estudiante en una pregunta de un contenido de un Software Educativo, Visualizar evaluación de un estudiante por contenido de un Software Educativo.

Kadir Hector Ortiz

RUP

Modelo de dominio

Hora Entrada	02:45:12
Hora Salida	03:45:12
Pregunta:.....	¿Qué es el modelo de Casos de Uso?
Respuesta Ideal	Es un modelo del sistema que contiene actores, casos de uso y sus relaciones.
Respuesta Real.....	Es un modelo del sistema que contiene actores, casos de uso y sus relaciones.
Evaluación.....	5

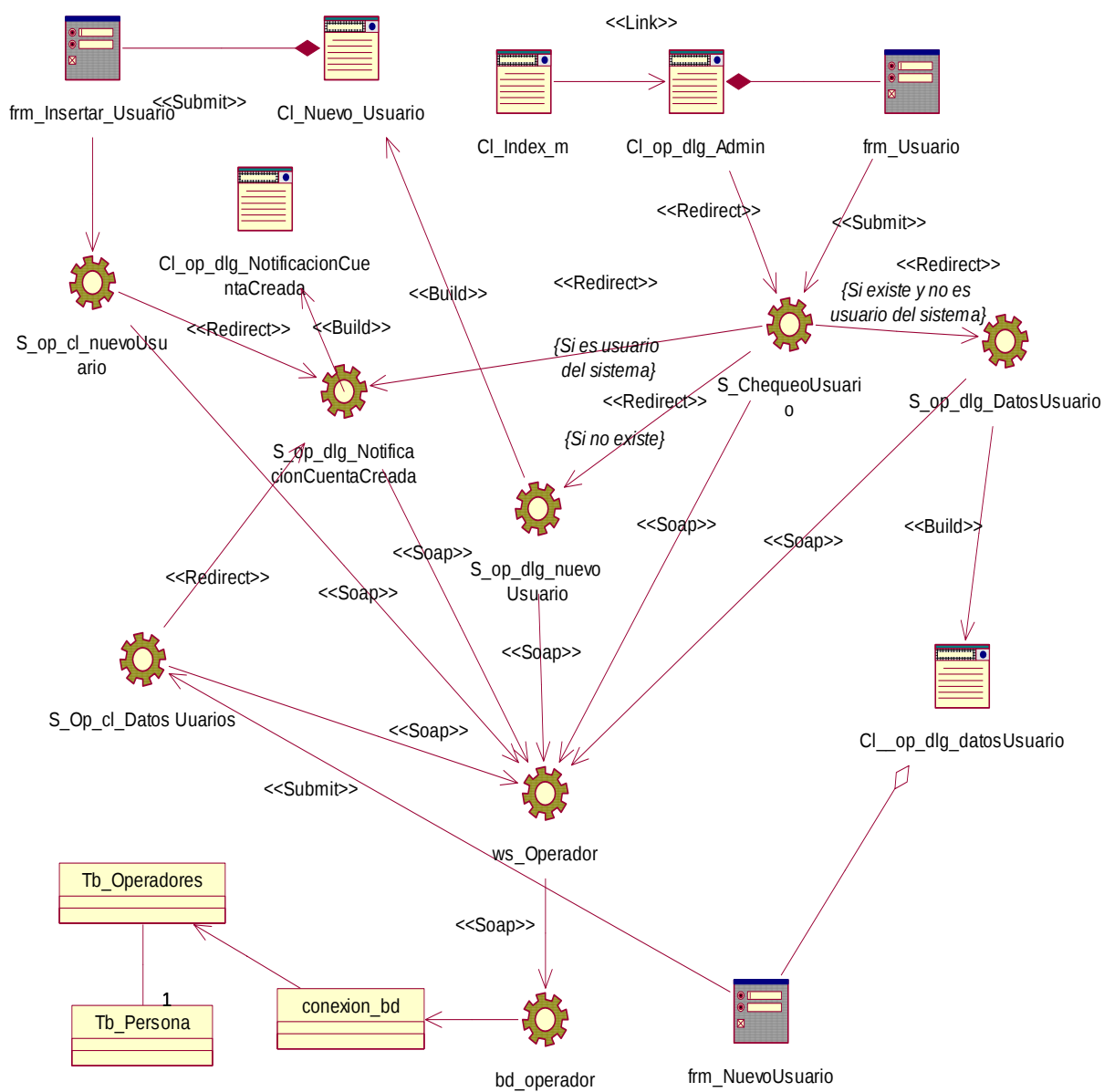
Anexo 2.17 Prototipo Visualizar tiempo que estuvo un estudiante en un Software Educativo por fecha.

Tiempo Total por Fecha de: Kadir Hector Ortiz en RUP

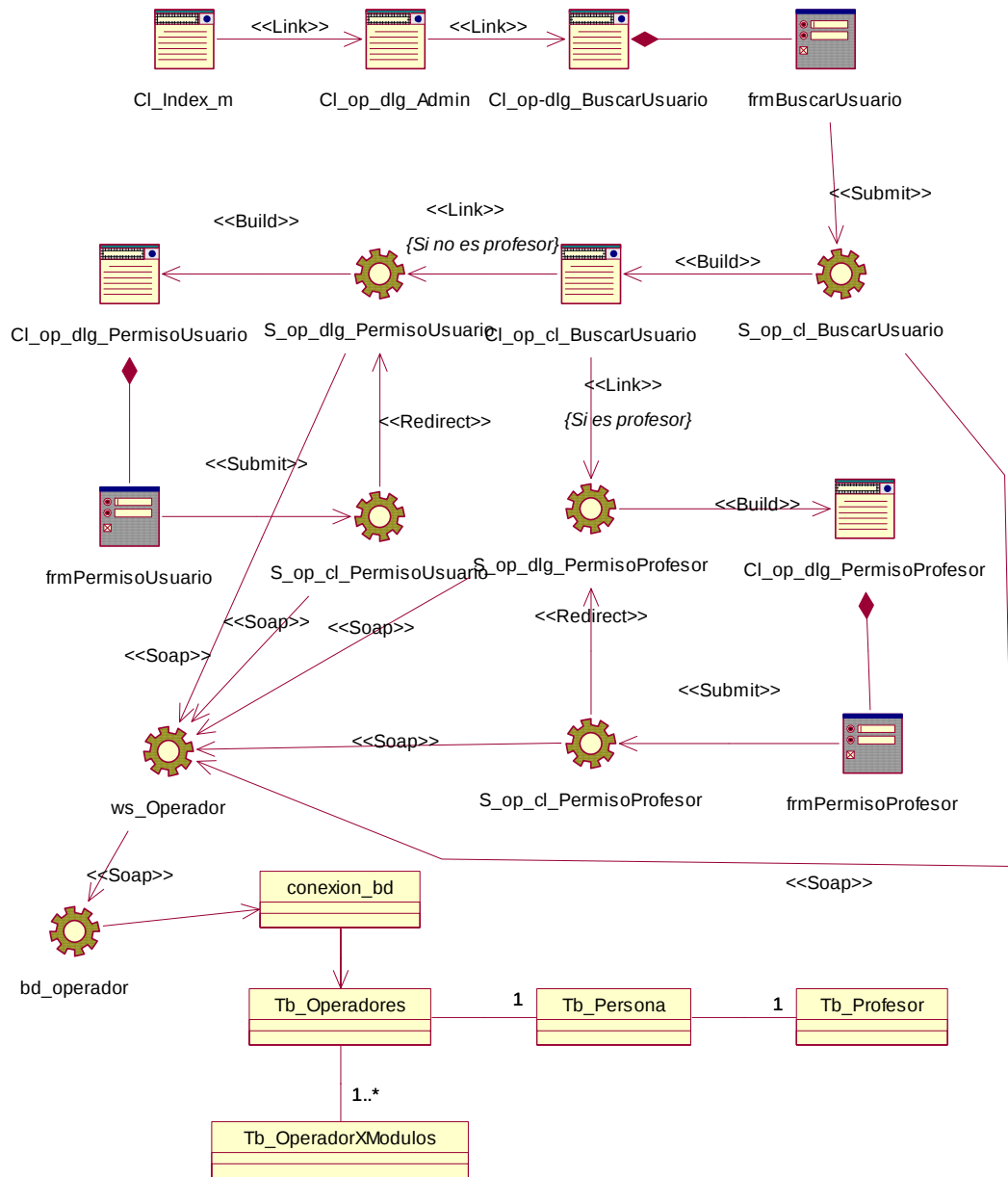
	Fecha Entrada(s)	Fecha Salida(s)	Tiempo(s)
1	2007-03-27	2007-03-27	00:54:00
2	2007-04-22	2007-04-22	00:05:00
3	2007-04-29	2007-04-29	08:14:48

Anexo 2.18 Prototipo Consultar ayuda.**Ayuda**[Administrador](#)[Crear Usuario](#)[Buscar Usuarios](#)[Habilitar, Inhabilitar y Asignar Permisos a Usuario.](#)[Asignar Asignatura al Profesor.](#)[Cambiar Contraseña.](#)[Registro de Software](#)[Cambiar Contraseña Usuario Registrado.](#)[Archivo.](#)

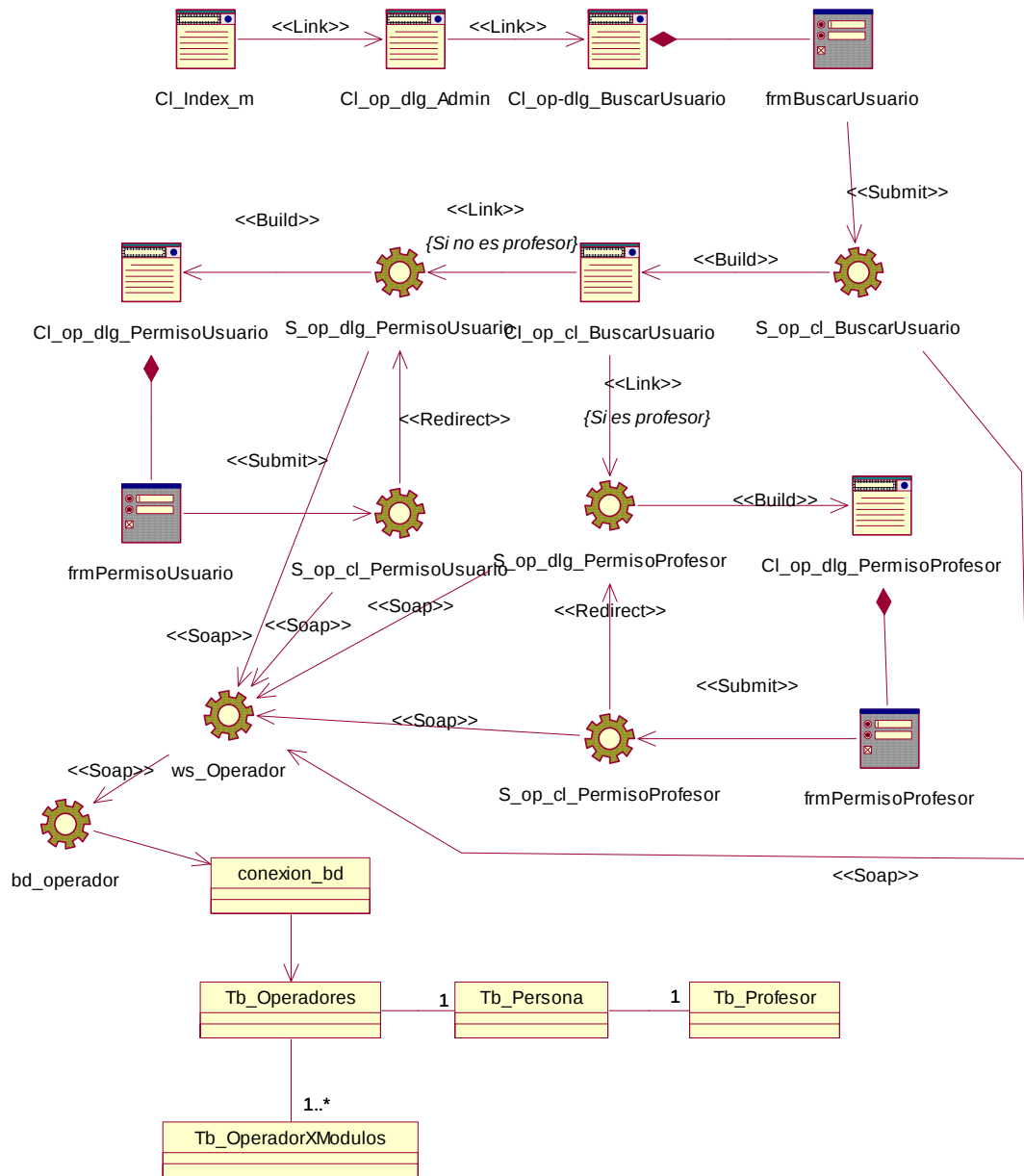
Anexo 3.1 Diagrama de Clases Web Crear usuario.



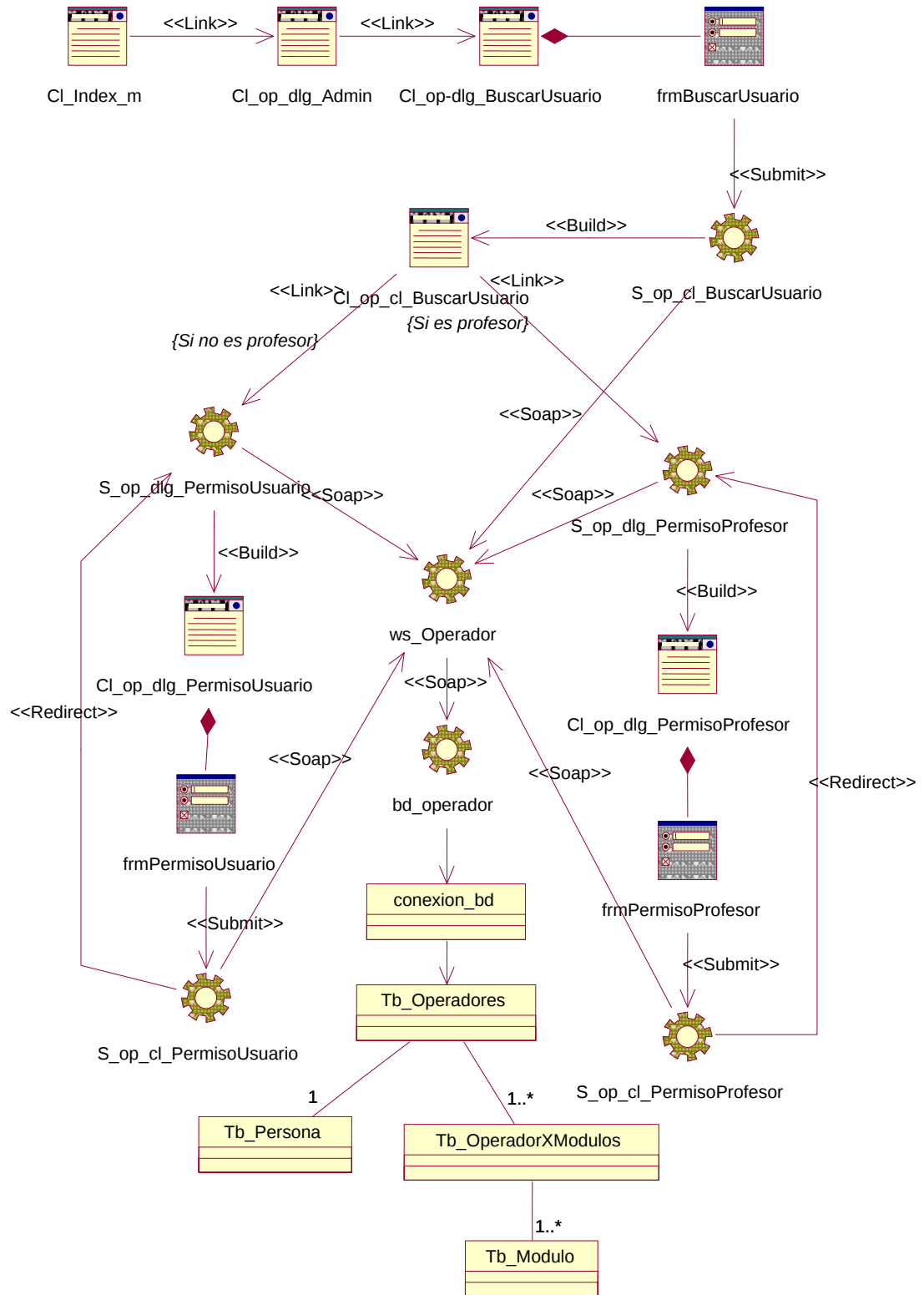
Anexo 3.2 Diagrama de Clases Web Habilitar usuario.



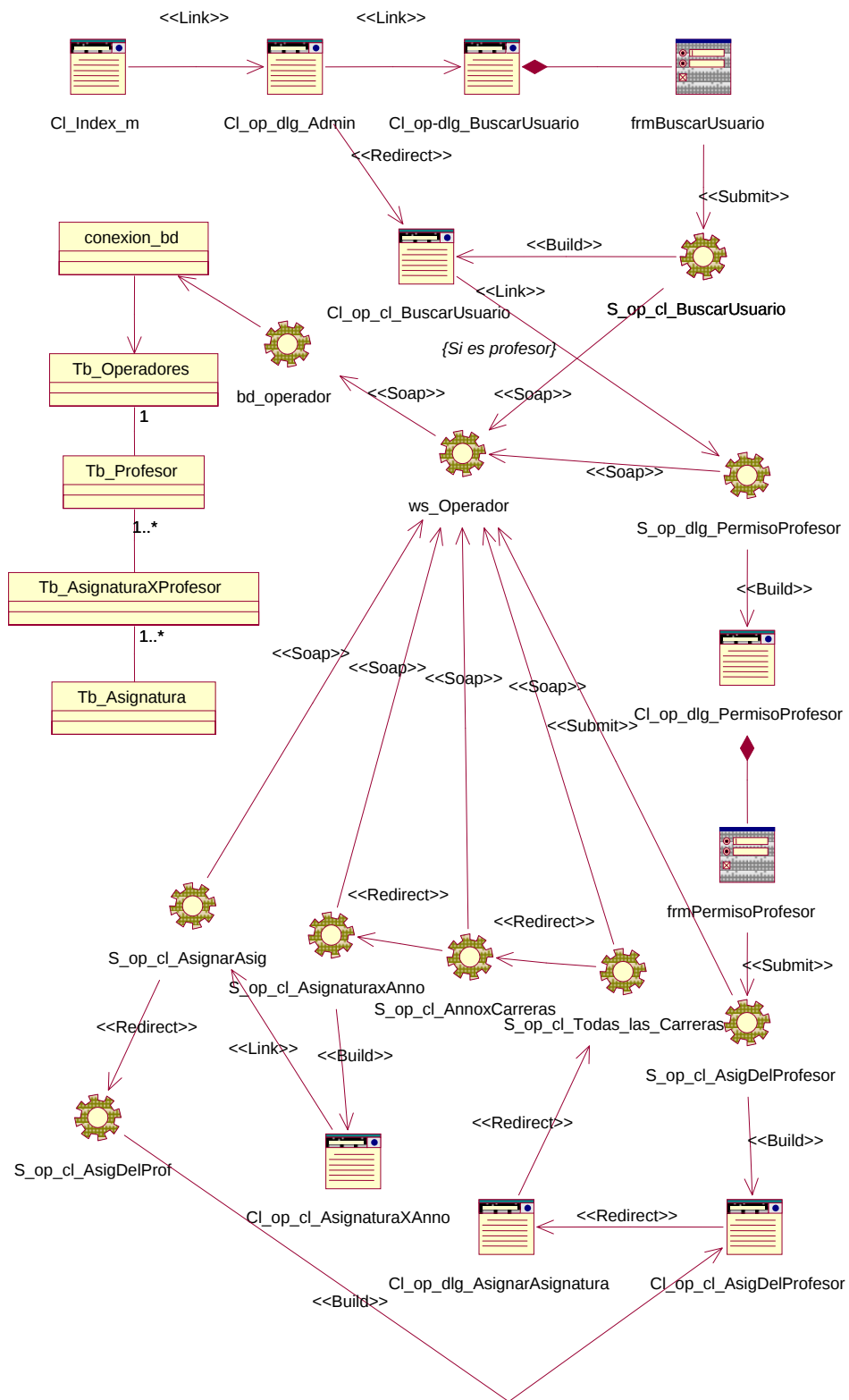
Anexo 3.3 Diagrama de Clases Web Inhabilitar usuario.

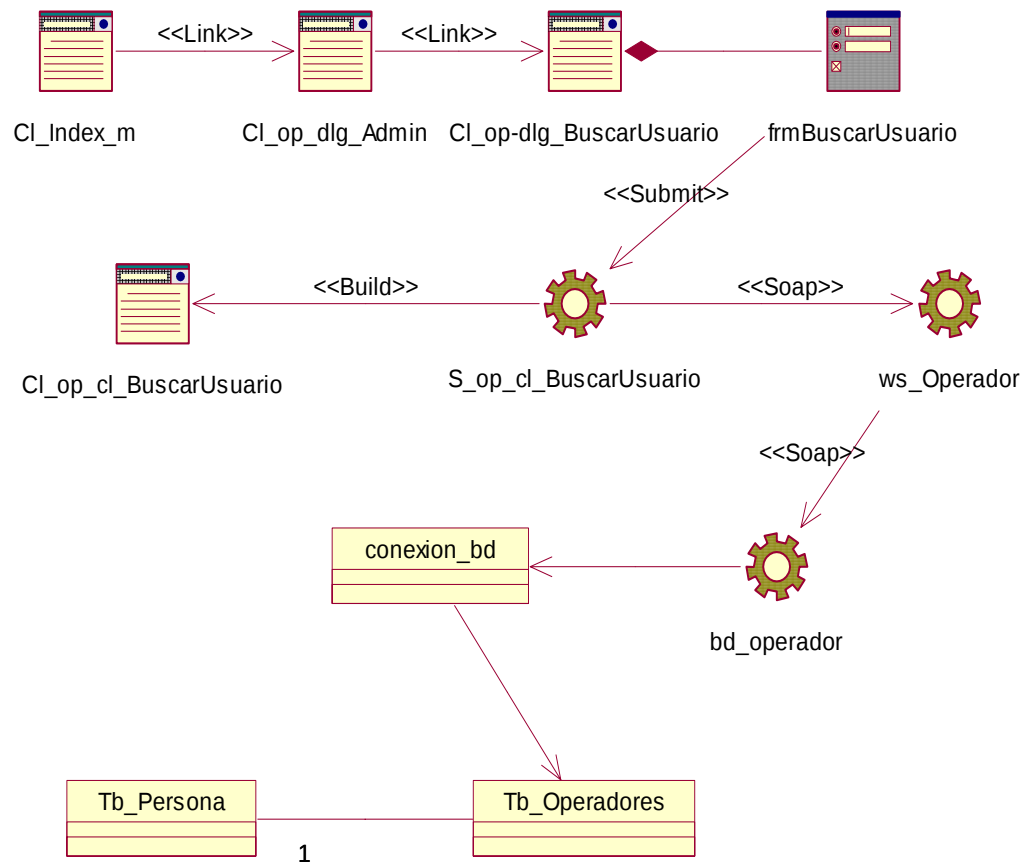


Anexo 3.4 Diagrama de Clases Web Asignar permisos a usuarios.

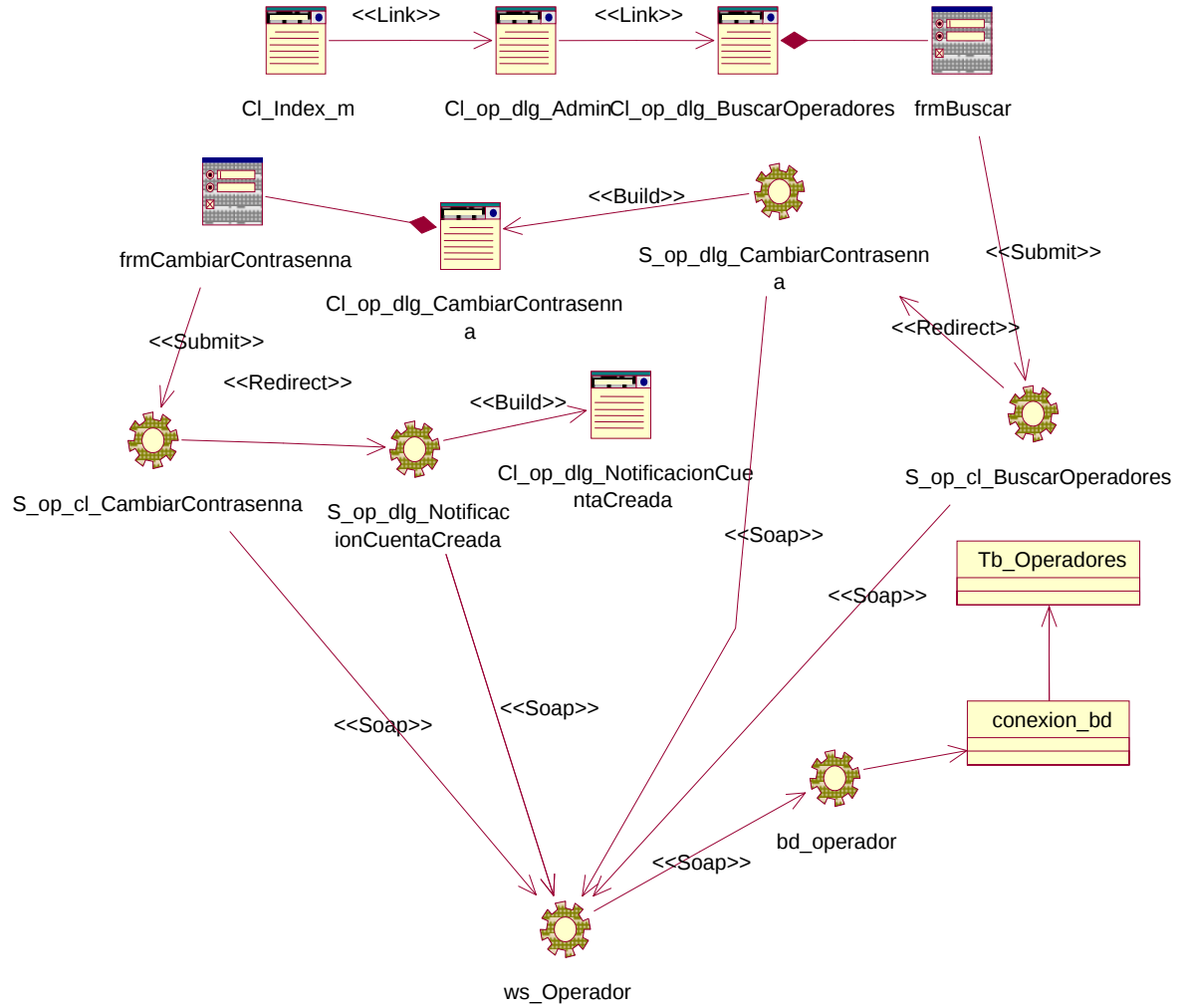


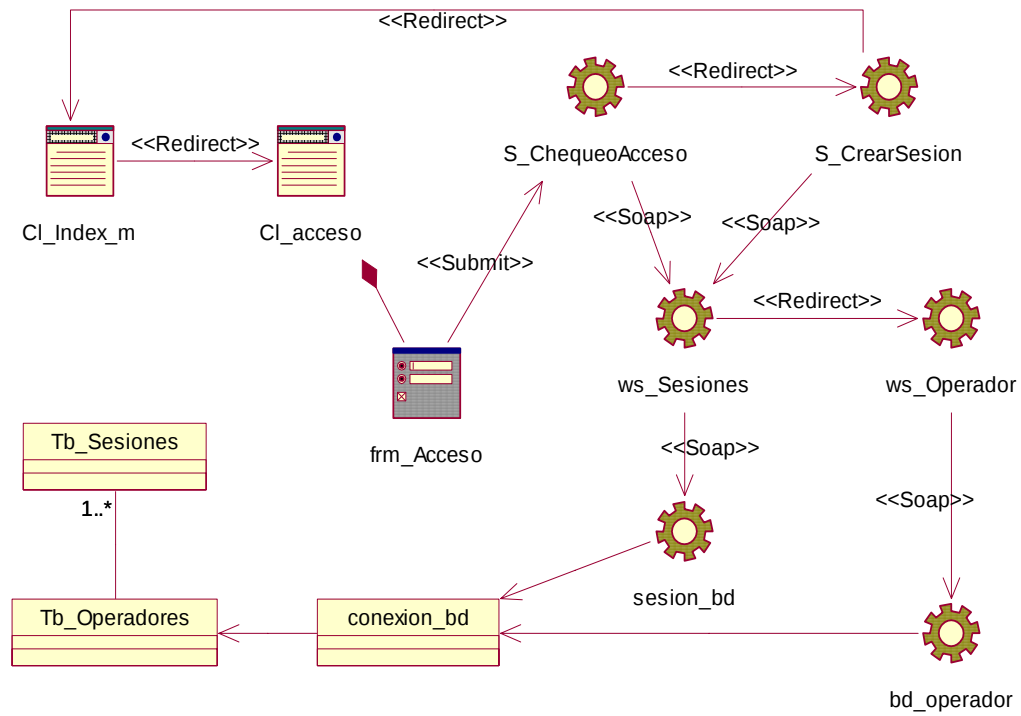
Anexo 3.5 Diagrama de Clases Web Asignar asignatura al profesor.



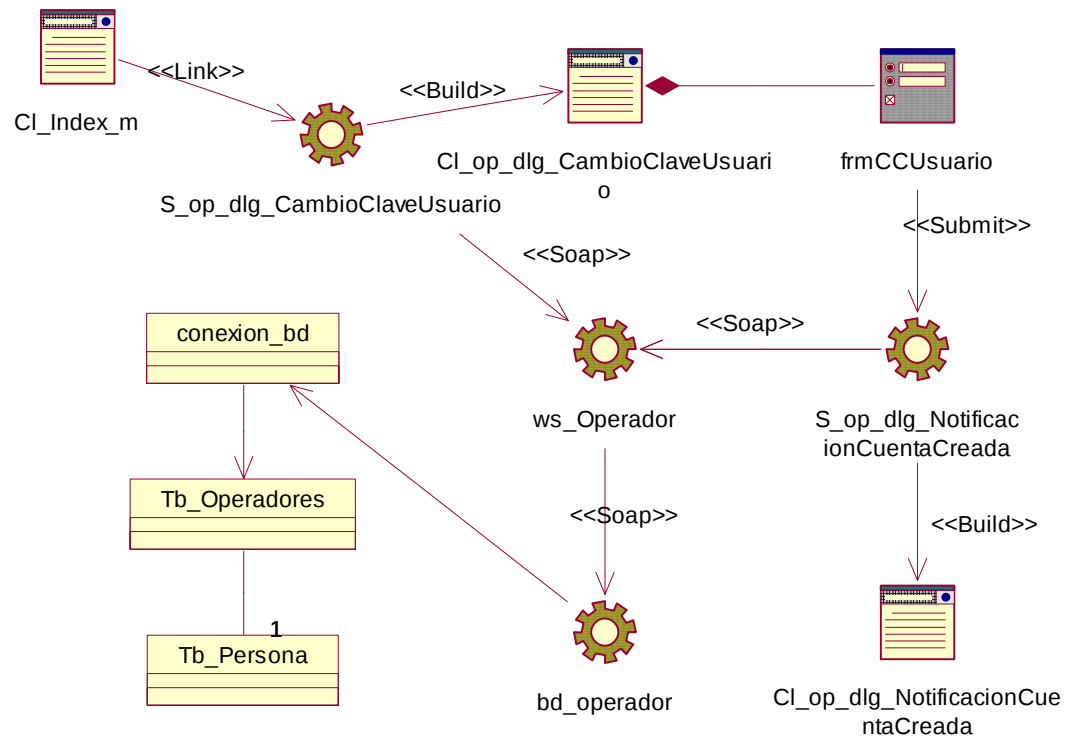
Anexo 3.6 Diagrama de Clases Web Buscar usuario.

Anexo 3.7 Diagrama de Clases Web Cambiar contraseña.

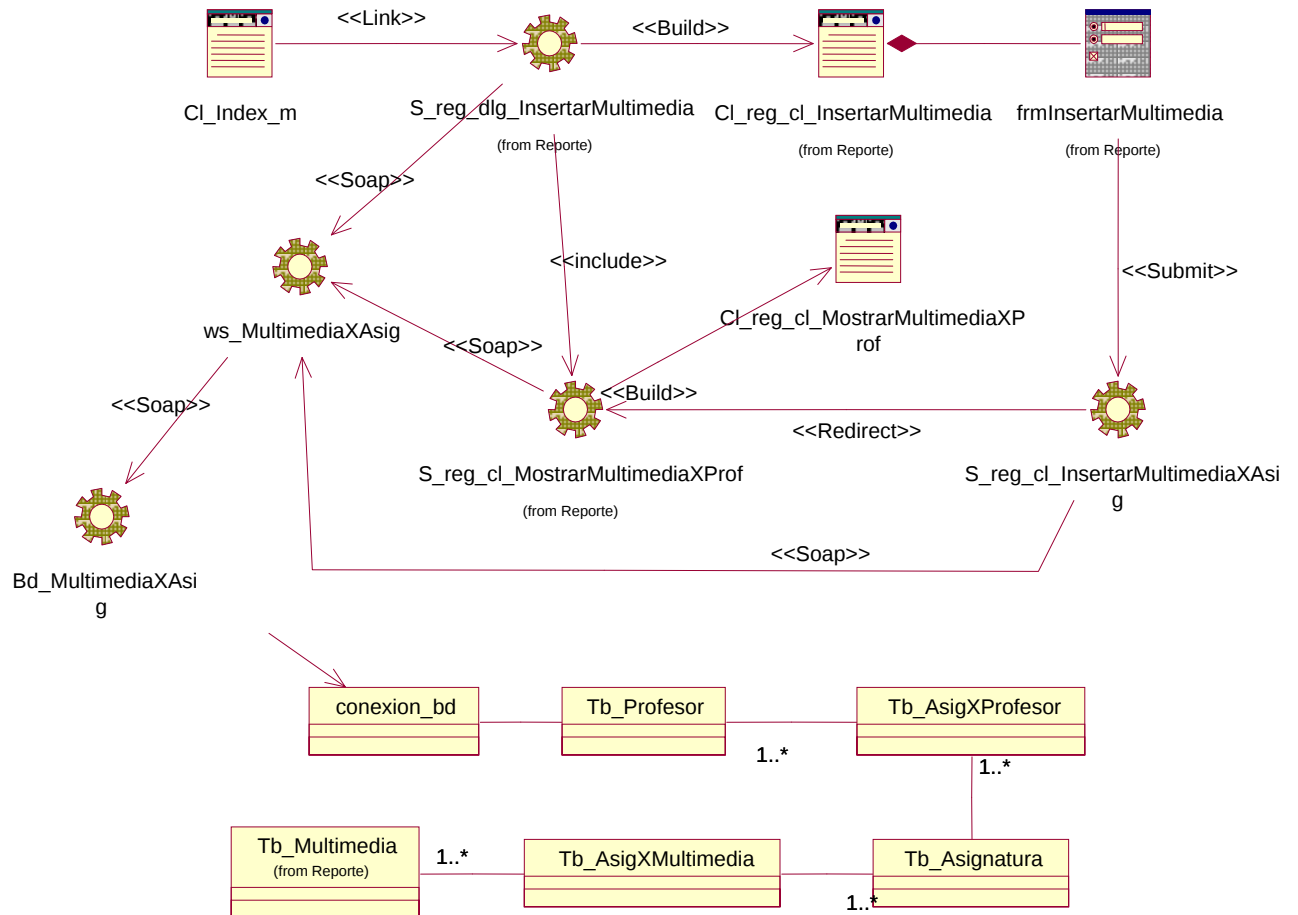


Anexo 3.8 Diagrama de Clases Web Autenticar usuario.

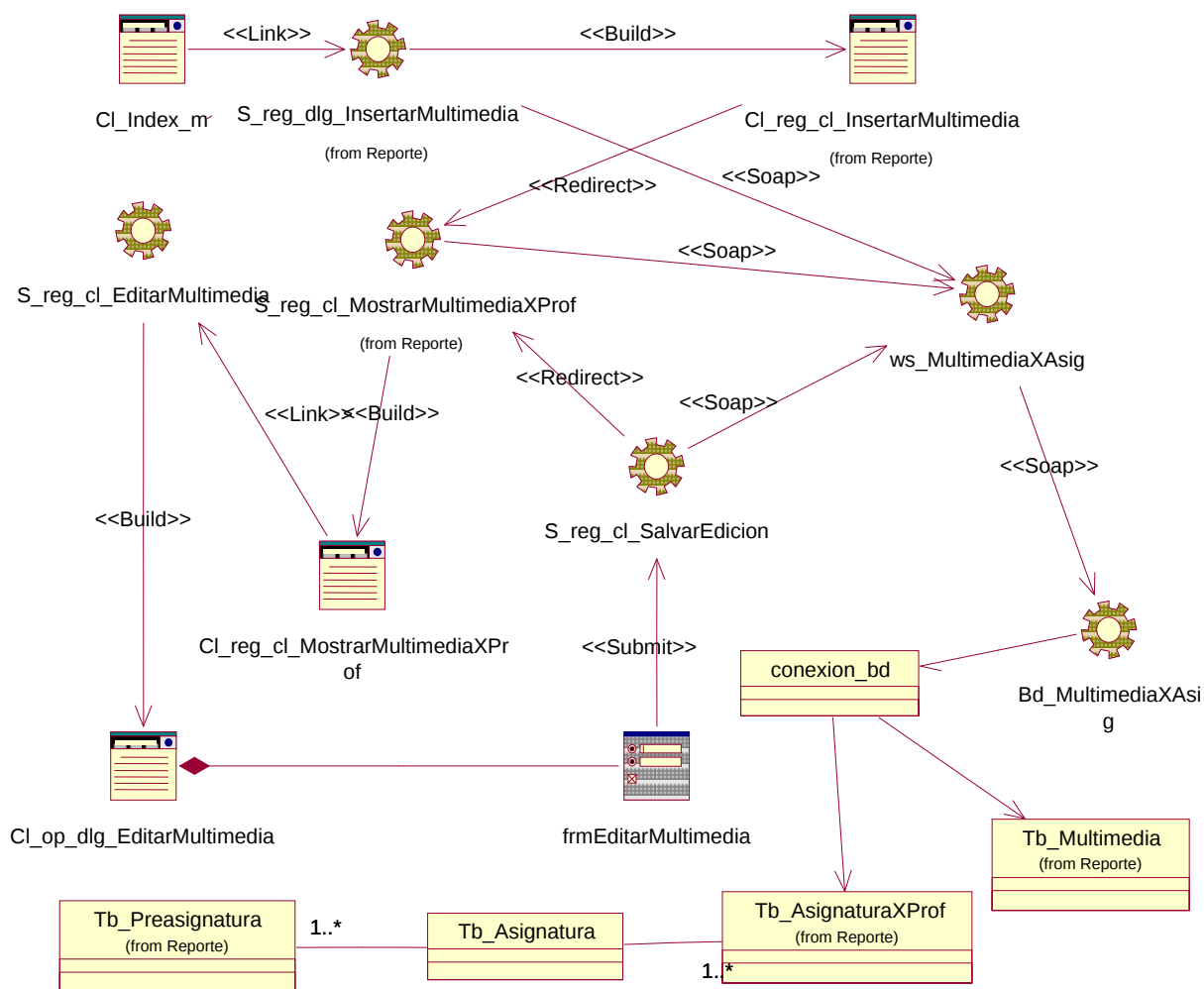
Anexo 3.9 Diagrama de Clases Web Cambiar contraseña usuario registrado.



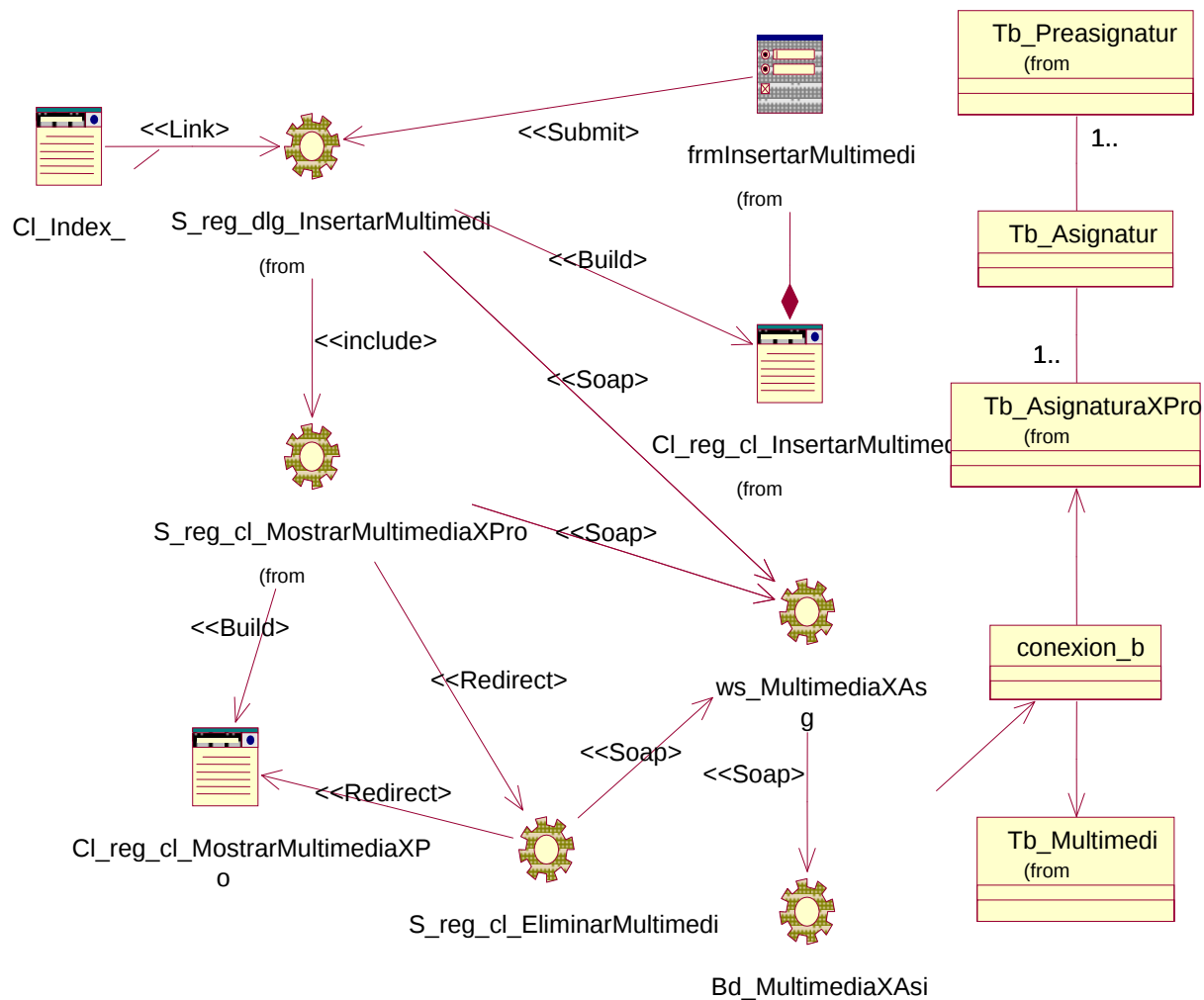
Anexo 3.10 Diagrama de Clases Web Asignar Software Educativo a asignatura.

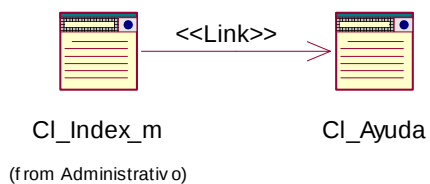
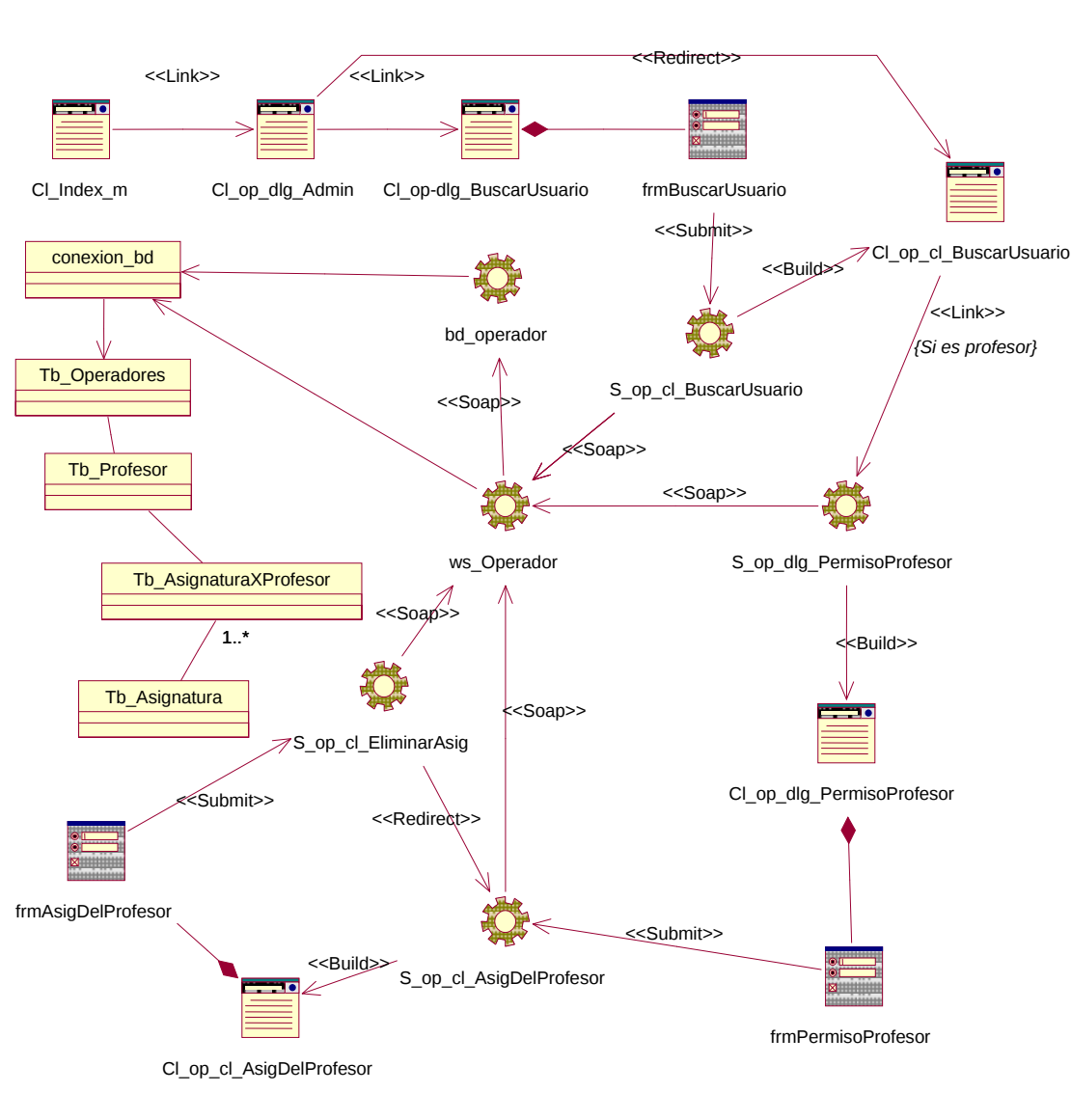


Anexo 3.11 Diagrama de Clases Web Editar nombre de Software Educativo.

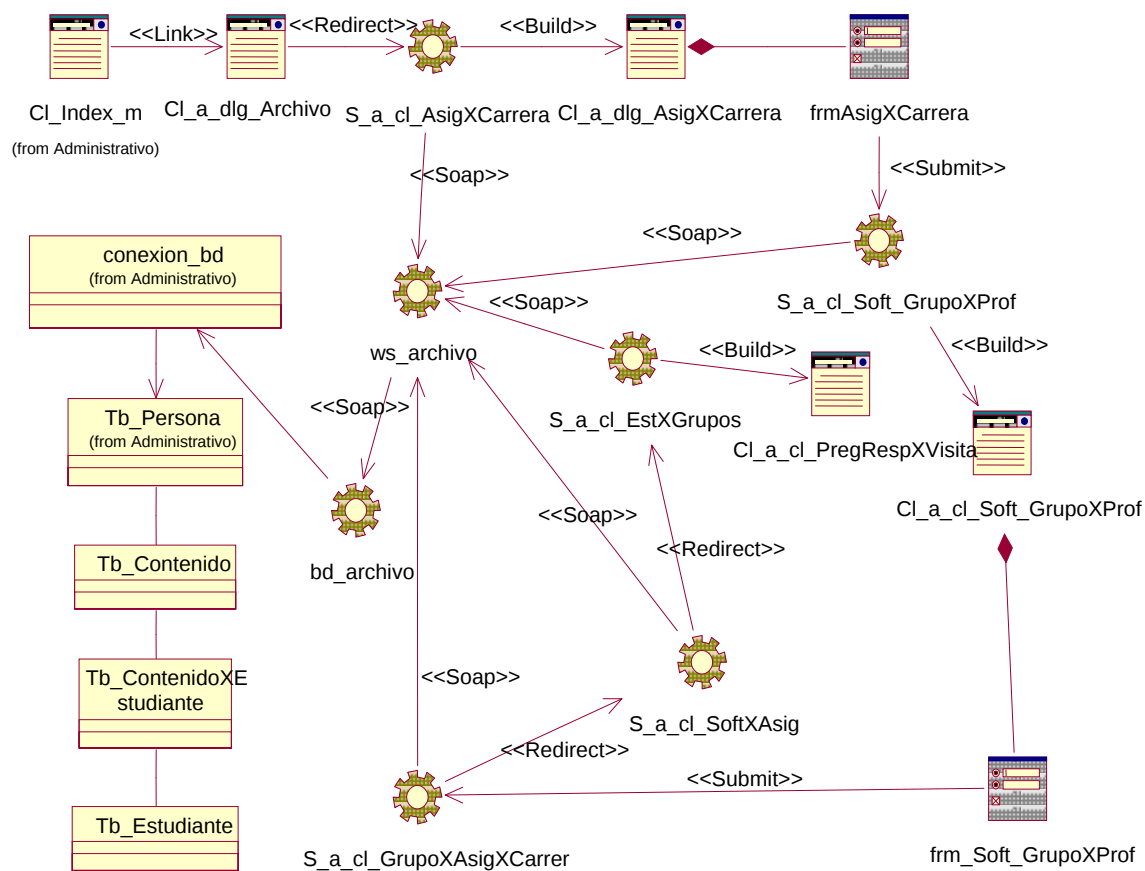


Anexo 3.12 Diagrama de Clases Web Eliminar Software Educativo del sistema.

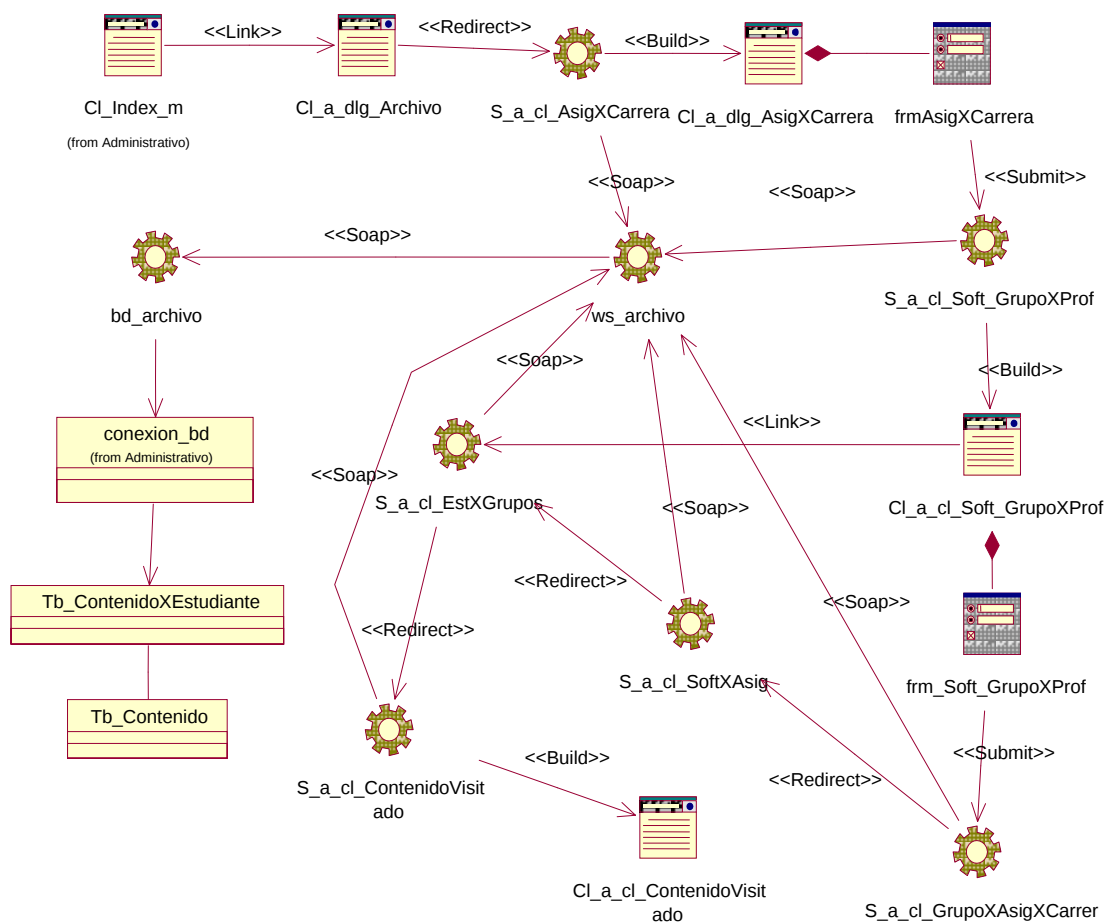




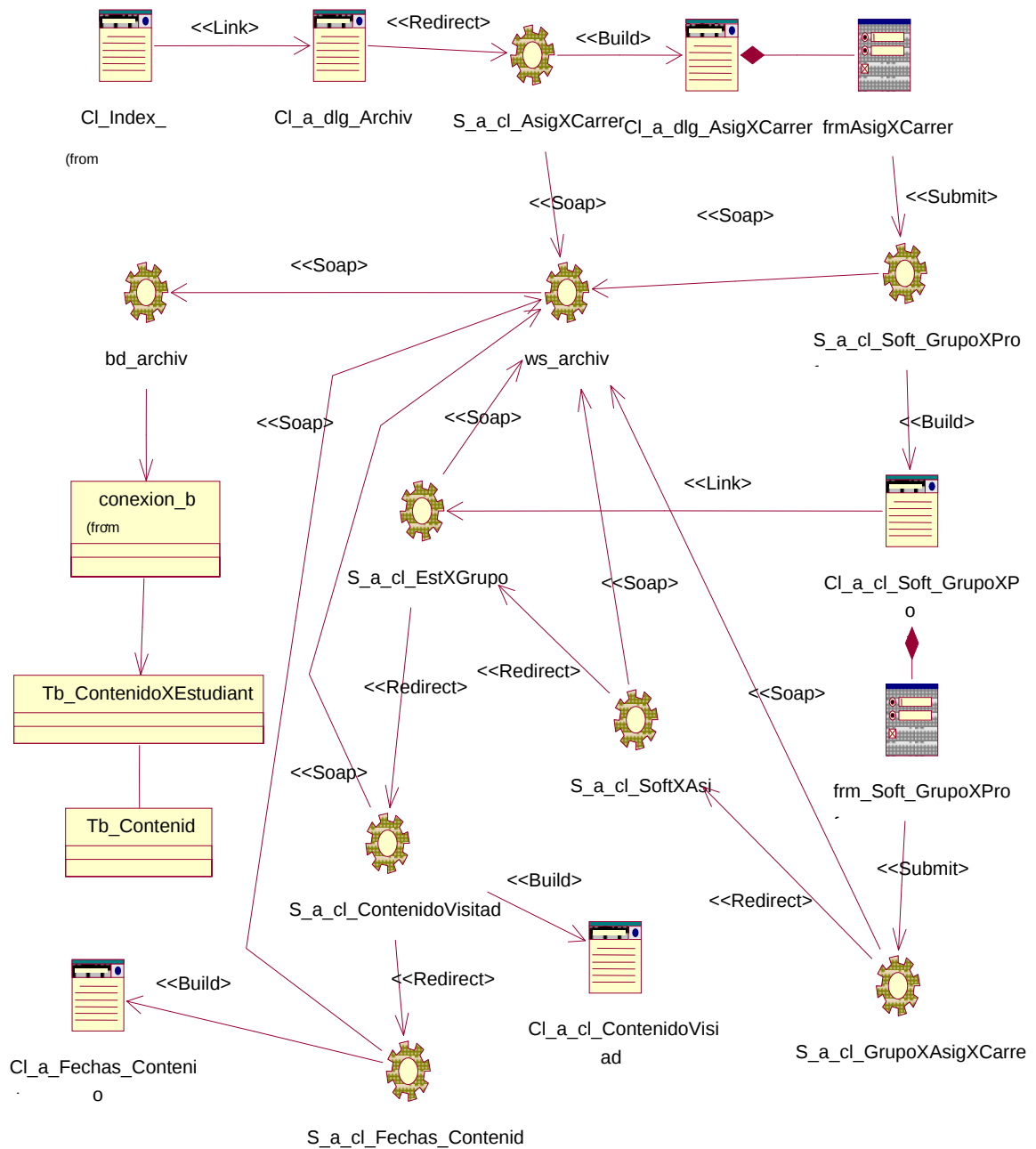
Anexo 3.15 Diagrama de Clases Web Visualizar listado de estudiantes que visitaron un Software Educativo por grupo.



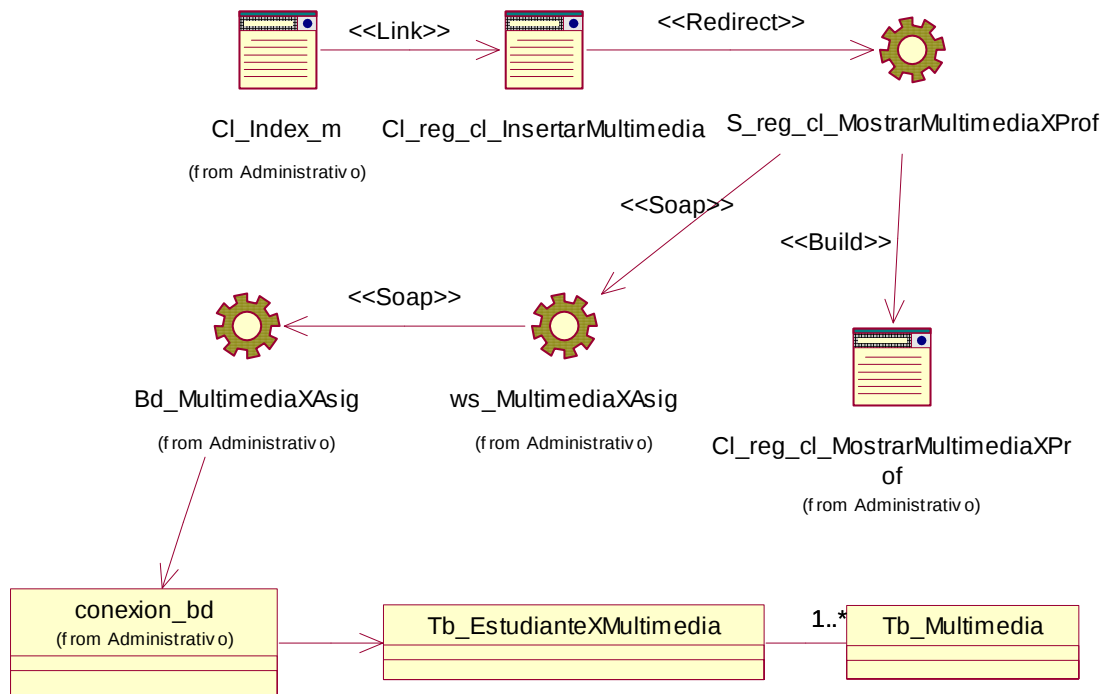
Anexo 3.16 Diagrama de Clases Web Visualizar el contenido visitado por un estudiante en el Software Educativo.



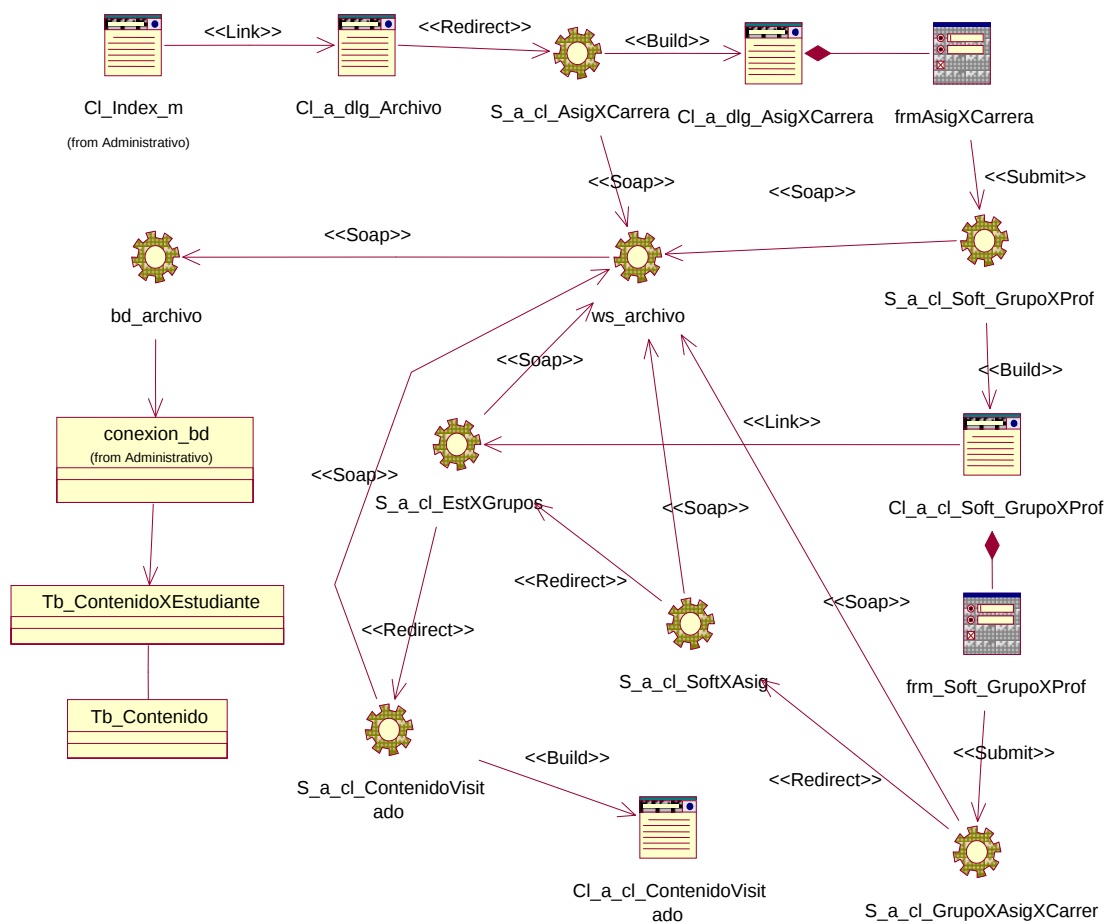
Anexo 3.17 Diagrama de Clases Web Visualizar fecha de entrada y salida de un estudiante en un contenido de un Software Educativo.



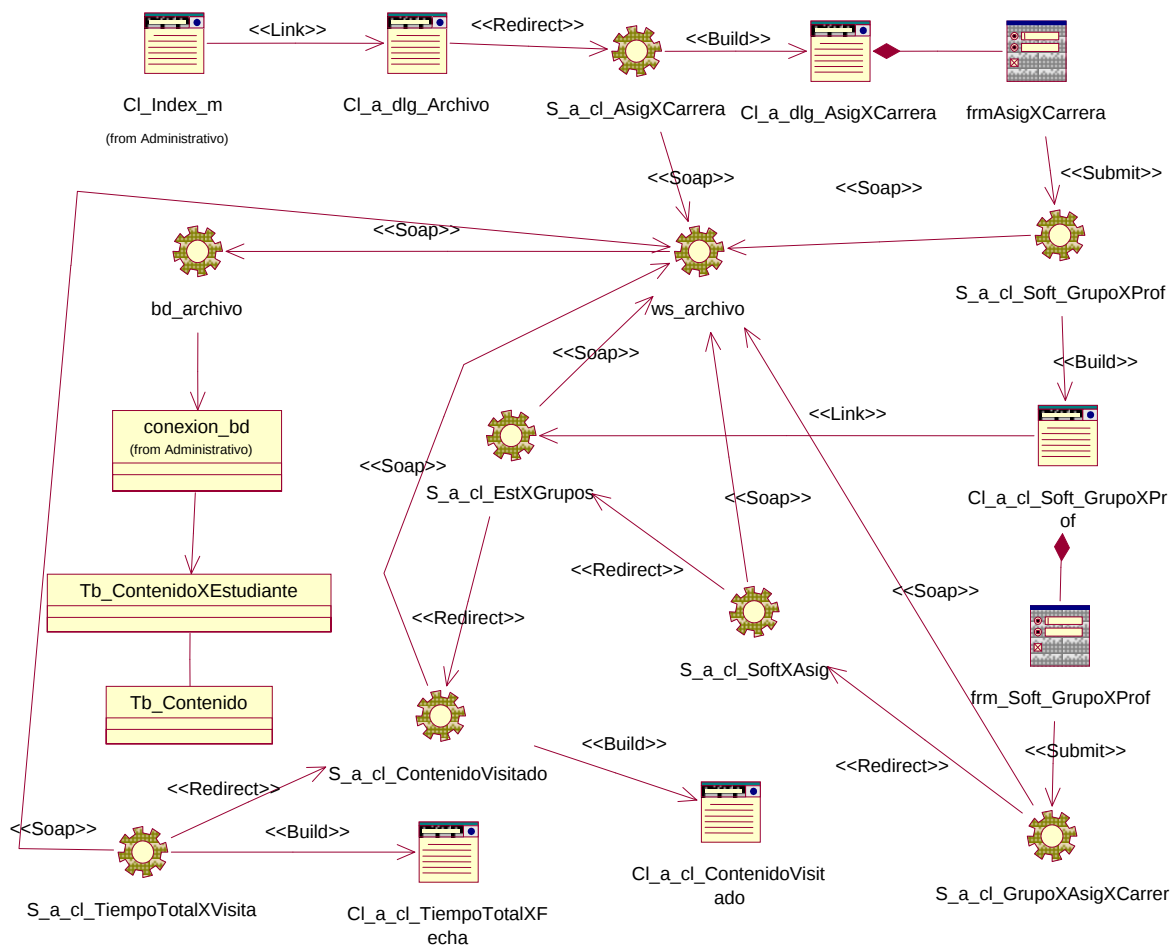
Anexo 3.18 Diagrama de Clases Web Visualizar cantidad de visitas al Software Educativo.



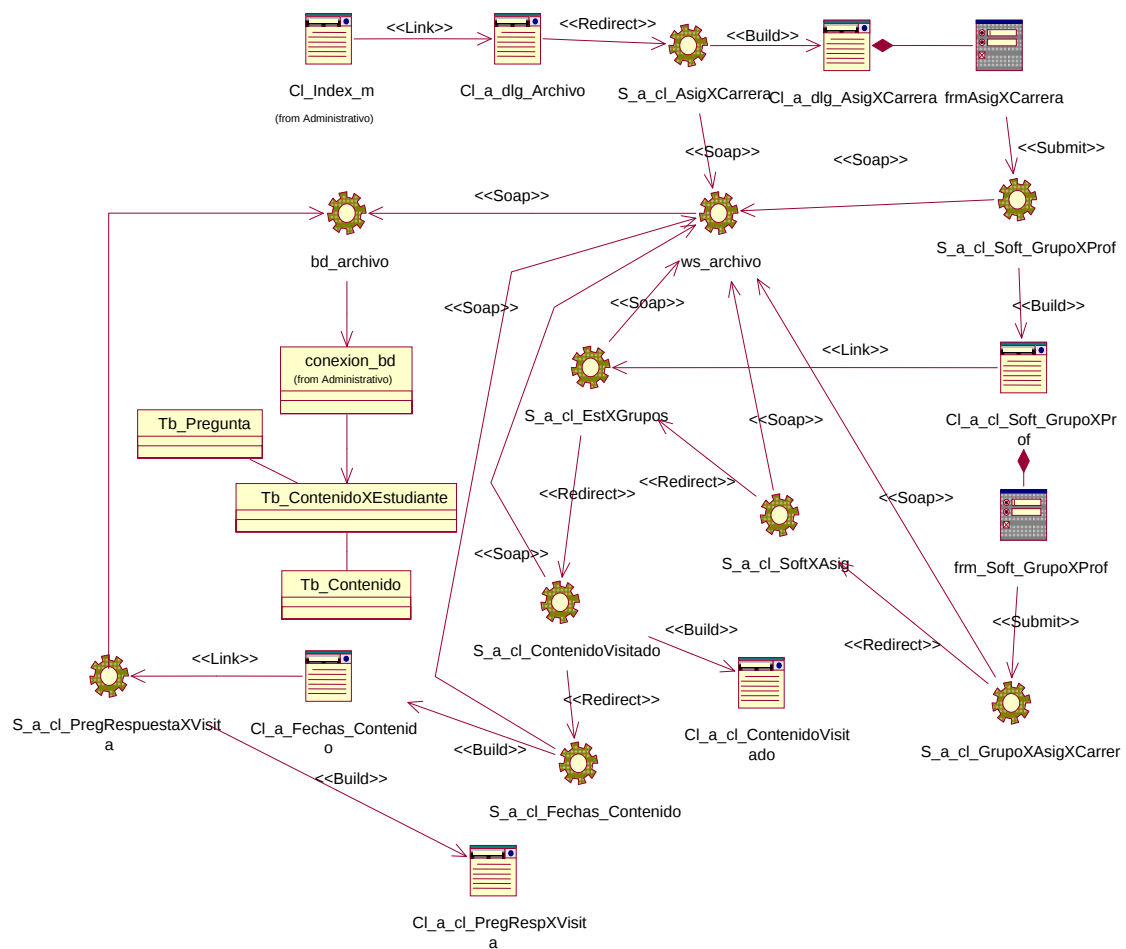
Anexo 3.19 Diagrama de Clases Web Visualizar cantidad de visitas del estudiante al Software Educativo.



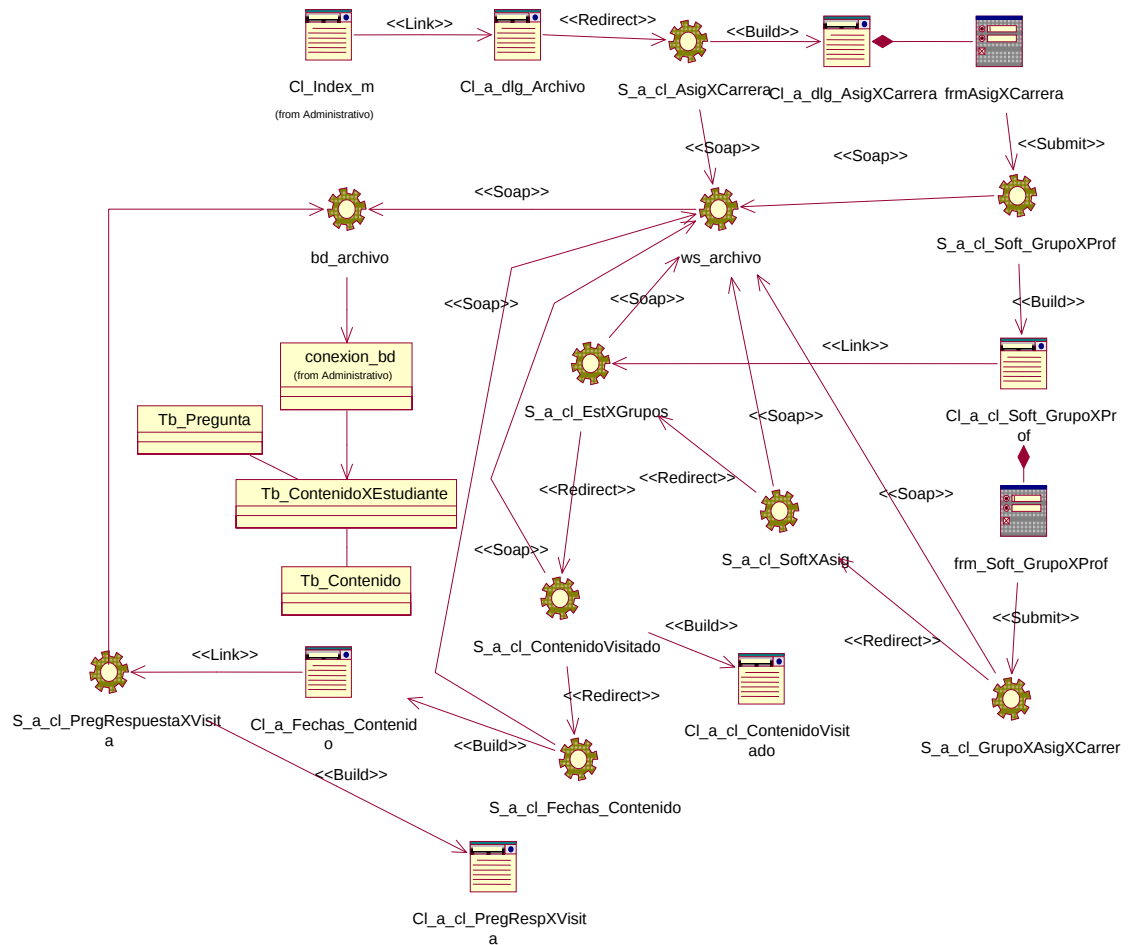
Anexo 3.20 Diagrama de Clases Web Visualizar el tiempo que estuvo un estudiante en un Software Educativo por fecha.



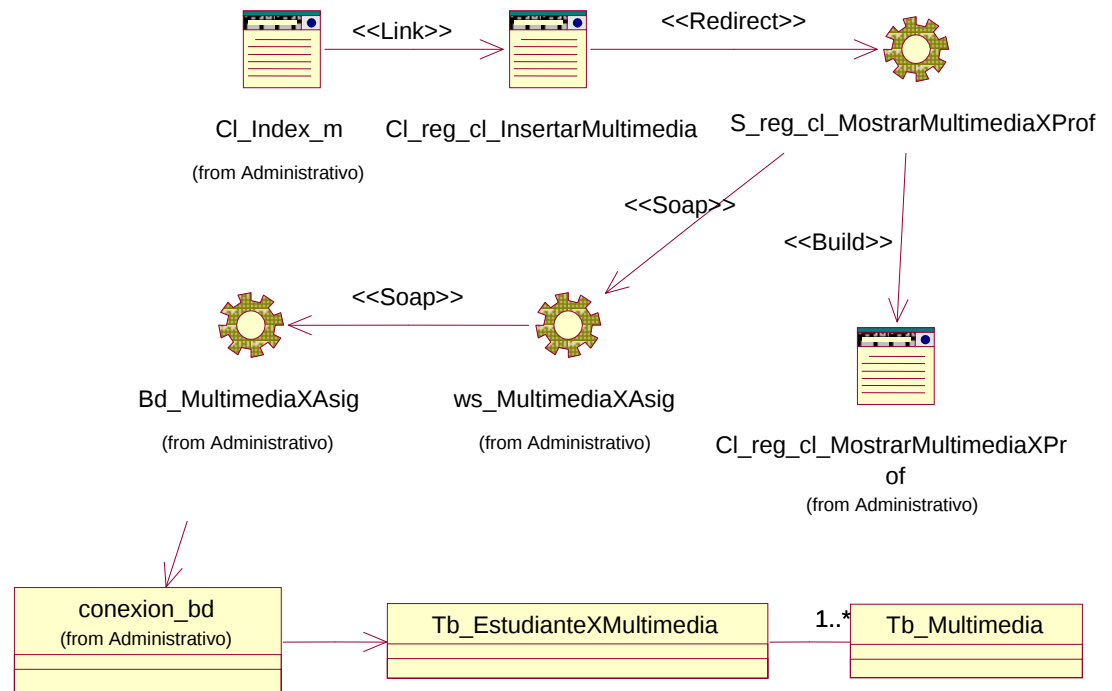
Anexo 3.21 Diagrama de Clases Web Visualizar evaluación de un estudiante por contenido de un SE.

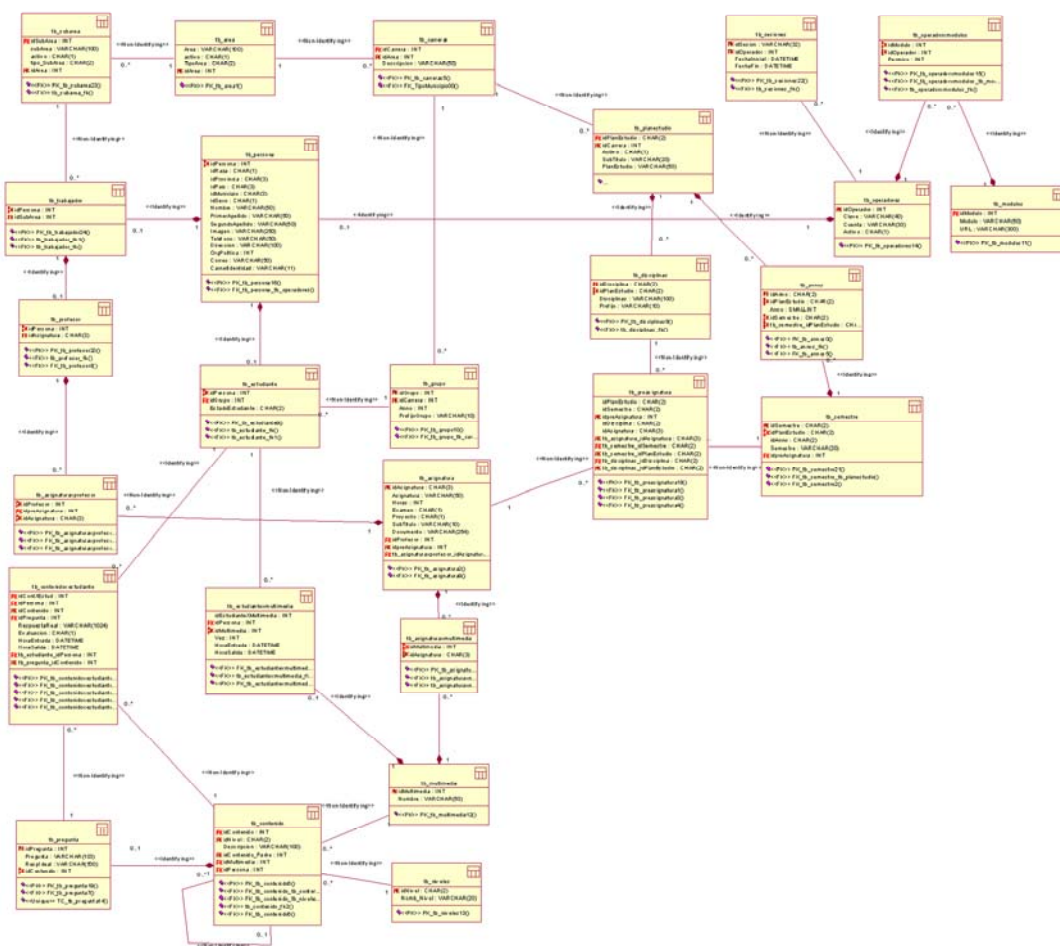


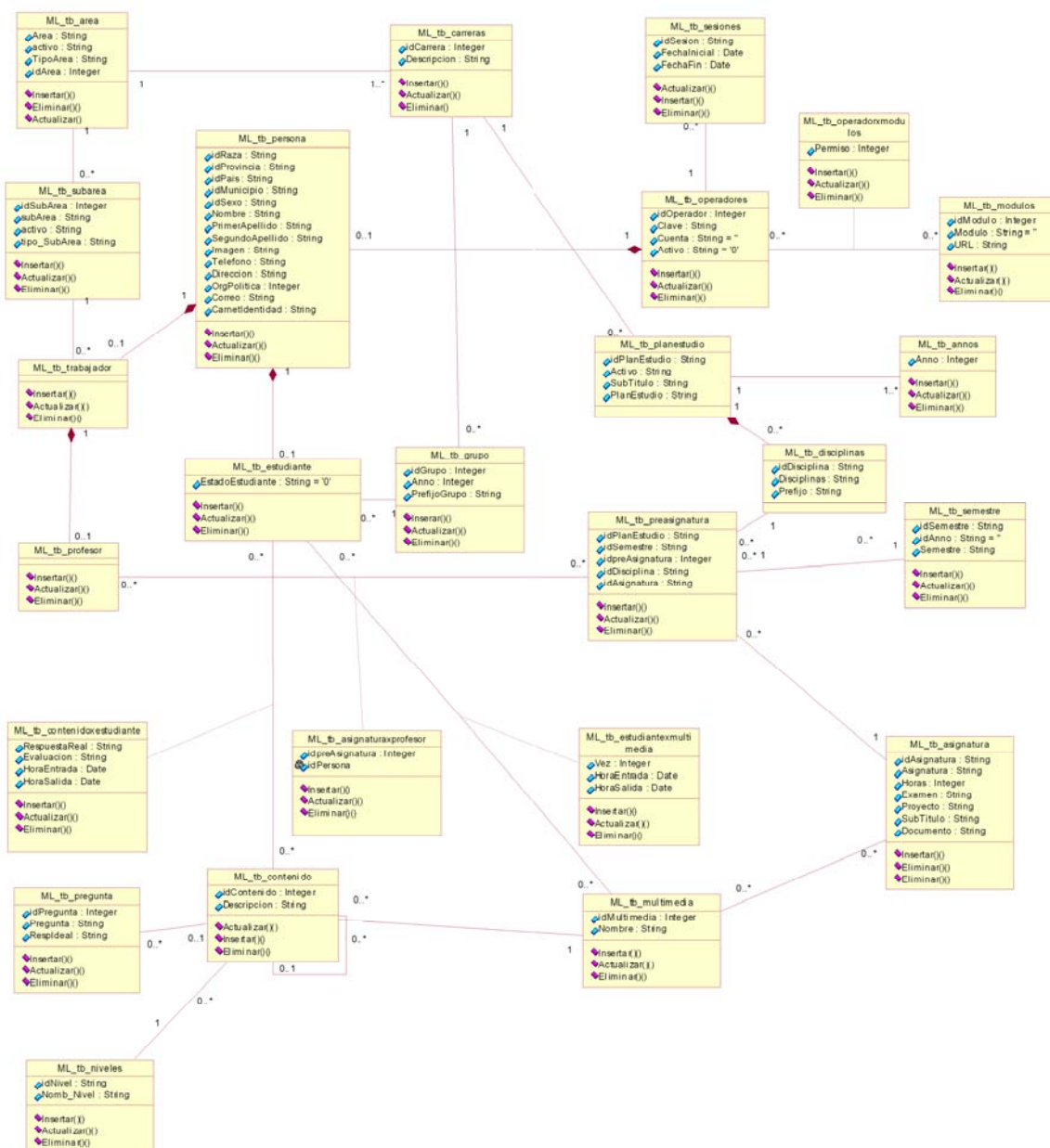
Anexo 3.22 Diagrama de Clases Web Visualizar pregunta y respuesta de un estudiante por contenido en un SE.



Anexo 3.23 Diagrama de Clases Web Visualizar Softwares Educativos de una asignatura.







Anexo 3.26 Consulta a la EBSCO.

Lista de carpetas

Elementos de la lista de resultados (2)

▶ [Imágenes](#) (0)

▶ [Videos](#) (0)

[Vínculos persistentes a la búsqueda](#) (0)

▶ [Búsquedas guardadas](#) (0)

▶ [Alertas de búsqueda](#) (0)

▶ [Alertas de publicación](#) (0)

▶ [Páginas Web](#) (0)

Elementos de la lista de resultados

1 - 2 de 2 Páginas: 1  [Imprimir](#)  [Enviar](#)  [Guardar en disco](#) [Eliminar todos](#)

1. [SRW/U with OAI: Expected and Unexpected Synergies.](#) By: Sanderson, Robert; Young, Jeffrey; LeVan, Ralph. D-Lib Magazine, Feb2005, Vol. 11 Issue 2, p1-1, 1p; DOI: 10.1045/february2005-sanderson; (AN **16556763**)

2. [Potential of Webservices to enable smart business networks.](#) By: van Hillegersberg, Jos; Boeke, Ruurd; van den Heuvel, Willem-Jan. Journal of Information Technology (Palgrave Macmillan), Dec2004, Vol. 19 Issue 4, p281-287, 7p, 6 diagrams; DOI: 10.1057/palgrave.jit.200027; (AN **15908178**)

1 - 2 de 2 Páginas: 1