

Universidad de Ciencias Informáticas.



Facultad 2

PSPSOFT, sistema de apoyo a la docencia para la asignatura de Gestión de Software.

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

Autores:

**Deivis Ricardo Álvarez Mendoza.
Aliennis Mercedes González Hurtado.**

Tutor:

Lic. Lianne Guillén Pérez.

Ciudad de La Habana, Cuba

Julio, 2007

DECLARACIÓN DE AUTORÍA

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

Para que así conste, firmamos la presente a los ___ días del mes de julio del 2007.

Firma del autor
Deivis Ricardo
Álvarez Mendoza

Firma del autor
Aliennis Mercedes
González Hurtado

Firma del tutor
Lic. Lianne
Guillén Pérez

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: PPSOFT, sistema de apoyo a la docencia para la asignatura Gestión de Software.

Autores: Deivis Ricardo Álvarez Mendoza.

Aliennis Mercedes González Hurtado.

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de

Firma

_____de julio del 2007

"No hay quien pueda parar los frutos, de tantas inteligencias trabajando en coordinación y tan estrechamente unidas."

Fidel Castro Ruz

Agradecimientos

Queremos agradecer a todas aquellas personas que de alguna forma u otra hicieron posible la realización de este trabajo.

Agradecemos en primer lugar a nuestra familia, de quienes hemos tenido en todo momento un apoyo significativo e incondicional en los momentos de intenso trabajo.

Estamos muy agradecidos de Lianne Pérez Guillén nuestra tutora que nos ayudó y apoyó mucho en la realización de nuestro trabajo.

Damos un reconocimiento especial a profesores y amigos que en un momento u otro contamos con su ayuda incondicional, ellos son: Anabel Parra Vázquez, Yadira Ruiz Constanten, Maypher Román Durán, Liuby Vega Garcés.

Queremos dar nuestros más profundos agradecimientos a nuestro mejor pensador el Comandante en Jefe Fidel Castro Ruz que sin él no hubiera sido posible este sueño hecho realidad.

Muchas gracias a todos por su paciencia y comprensión.

Dedicatoria

A mi madre Sila Álvarez y mi tía Estel Álvarez quienes me ayudaron a hacer mi sueño realidad con sus consejos, apoyo y cariño. Resto de mi familia y amigos.

Deivis

A mis padres Mercedes Hurtado y Abel González, a mis abuelas Caridad Lorenzo y Edelma Matienzo y mis tías Yusneisy y Belkis.

Aliennis

Muy en especial a nuestro Comandante Fidel Castro Ruz.

Resumen

EL Proceso de Software Personal es un marco de trabajo diseñado para enseñar a los ingenieros del software a hacer mejor su labor. Para lograr esto, se recogen datos que se guardan en el Cuaderno del Ingeniero. En la Universidad de las Ciencias Informáticas la recopilación de los datos por los estudiantes y profesores se hacía de forma manual, de ahí la necesidad de automatizar los procesos de creación, actualización y control sistemático del Cuaderno de Ingeniería.

Como objetivo se propone desarrollar un sistema informático para la automatización de la gestión y control del Cuaderno de Ingeniería para la asignatura de Gestión de Software.

Se utilizó como soporte teórico para el desarrollo de la aplicación el libro “Introducción al Proceso de Software Personal (PSP)” del autor Watts S. Humphrey.

Como resultado se obtuvo una aplicación web, sencilla, de fácil manejo, de interfaz amigable, flexible y amena, soportada sobre la tecnología PHP y usando los gestores de base de datos PostgreSQL o MySQL según sea conveniente para el almacenamiento y control de los datos.

Índice

Introducción.....	1
CAPÍTULO 1	5
Fundamentación Teórica.....	5
Introducción.....	5
1.2 Caracterización del Proceso de Software Personal y Cuaderno de Ingeniería	5
1.2.1 Caracterización del Proceso de Software Personal (PSP).....	5
1.2.2 Características del Cuaderno de Ingeniería.....	6
1.3 Tendencias y tecnologías actuales	7
1.3.1 Aplicaciones Web.....	7
1.3.2 Comparación entre aplicaciones Web y aplicaciones Desktop.....	7
1.3.3 Modelo Cliente Servidor	8
1.3.4 Lenguaje PHP	9
1.3.5 Servidor Web Apache	10
1.3.6 Lenguaje XML	11
1.3.7 Lenguaje JavaScript.....	12
1.3.8 AJAX	13
1.3.9 Gestores de base de datos	13
1.3.9.1 MySQL	13
1.3.9.2 PostgreSql.....	14
1.4 Patrones de arquitectura	15
1.5 Metodología empleada en la Ingeniería de Software.....	16
1.5.1 Herramienta y lenguaje aplicado en la metodología RUP.....	17
1.6 Herramientas empleadas para la programación y diseño web	18
1.7 Conclusiones.....	18
CAPÍTULO 2.....	20
Características del sistema	20
2.1 Introducción.....	20
2.2 Objeto de estudio	20

2.2.1 Sistema automatizado existente vinculado al campo de acción	21
2.3 Modelo del Negocio.....	23
2.3.1 Descripción de los actores y trabajadores del negocio	24
2.3.2 Representación del diagrama de casos de uso del negocio.....	24
2.3.3 Descripción de los casos de uso del negocio	24
2.3.4 Diagrama de actividades del negocio	25
2.3.5 Modelo de objeto del negocio	25
2.4 Requerimientos	25
2.4.1 Requerimientos Funcionales.....	25
2.4.2 Requerimientos no funcionales	27
2.5 Diagrama de casos de uso del sistema	28
2.5.1 Definición de los actores del sistema a automatizar	29
2.5.2 Representación del diagrama de casos de uso del sistema	29
2.5.3 Descripción de los casos de uso del sistema.....	30
2.6 Conclusiones.....	30
CAPÍTULO 3.....	31
Análisis y diseño del sistema	31
3.1 Introducción.....	31
3.2 Diagrama de clases del análisis.....	31
3.2.1 Representación de los diagramas de clase del análisis.....	31
3.3 Diagrama de Secuencia del diseño.....	39
3.3.1 Representación de los diagramas de secuencia.....	39
3.4 Diagramas de clases del diseño	40
3.4.1 Representación de los diagramas de clases del diseño	41
3.5 Diseño de la Base de datos	49
3.5.1 Modelo Lógico.....	49
3.5.1.1 Representación del modelo lógico de datos	50
3.5.2 Modelo Físico	51
3.5.2.1 Representación del modelo físico	52
3.6 Conclusiones.....	53

CAPÍTULO 4.....	54
Implementación y Prueba del sistema	54
4.1 Introducción.....	54
4.2 Modelo de Despliegue.....	54
4.2.1 Representación del diagrama de Despliegue	54
4.3. Diagrama de componentes	55
4.3.1 Representación del diagrama de Componentes	55
4.4 Casos de prueba de integración	56
4.4.1 Descripción de los casos de prueba de integración.....	57
4.5 Conclusiones.....	76
CAPÍTULO 5.....	77
Estudio de factibilidad	77
5.1 Introducción.....	77
5.2 Planificación	77
5.2.2 Puntos de función desajustados	80
5.3 Cálculo de las instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.	81
5.3.1 Cálculos.....	81
5.4 Beneficios tangibles e intangibles	83
5.5 Análisis de costo	83
5.6 Conclusiones.....	84
Conclusiones.....	85
Recomendaciones.....	86
Referencias bibliográficas.....	87
Bibliografía.....	90

Introducción

Desde hace aproximadamente veinte años, en diversas oleadas y desde diversas ideologías, numerosos autores anuncian el advenimiento de la sociedad de la información: un conjunto de transformaciones económicas y sociales que cambiarán la base material de nuestra sociedad. Tal vez uno de los fenómenos más espectaculares asociados a este conjunto de transformaciones sea la introducción generalizada de las nuevas tecnologías de la información y la comunicación en todos los ámbitos de nuestras vidas. Ellas están cambiando nuestra manera de hacer las cosas: de trabajar, de divertirnos, de relacionarnos y de aprender. De modo sutil también están cambiando nuestra forma de pensar. [1]

Las nuevas Tecnologías de la Información y Comunicación (TIC) son aquellas herramientas computacionales e informáticas que procesan, almacenan, sintetizan, recuperan y muestran información representada de las más variadas formas. [2]

Para todo tipo de aplicaciones educativas, las TIC son medios y no fines. Es decir, son herramientas y materiales de construcción que facilitan el aprendizaje, el desarrollo de habilidades y distintas formas de aprender, estilos y ritmos de los aprendices. Del mismo modo, la tecnología es utilizada tanto para acercar al aprendiz al mundo, como el mundo al aprendiz. [2]

Precisamente para favorecer este proceso de las TIC que se empieza a desarrollar desde los entornos educativos informales (familia, ocio...), la escuela debe integrar también la nueva cultura: alfabetización digital, fuente de información, instrumento de productividad para realizar trabajos, material didáctico, instrumento cognitivo. Obviamente la escuela debe acercar a los estudiantes la cultura de hoy, no la cultura de ayer. Por ello es importante la presencia en clase del ordenador (y de la cámara de vídeo, y de la televisión...) desde los primeros cursos, como un instrumento más, que se utilizará con finalidades diversas: informativas, comunicativas, instructivas. [3]

El impacto de las TIC en la educación trae como ventajas:

- ✓ Interés y motivación. Los alumnos están muy motivados al utilizar los recursos TIC y la motivación (el querer) es uno de los motores del aprendizaje, ya que incita a la actividad y al pensamiento. Por

otro lado, la motivación hace que los estudiantes dediquen más tiempo a trabajar y, por tanto, es probable que aprendan más. [3]

- ✓ Interacción. Continua actividad intelectual. Los estudiantes están permanentemente activos al interactuar con el ordenador e interactuar entre ellos a distancia. Mantienen un alto grado de implicación en el trabajo. La versatilidad e interactividad del ordenador, la posibilidad de "dialogar" con él, el gran volumen de información disponible en Internet..., les atrae y mantiene su atención. [3]
- ✓ Mayor comunicación entre profesores y alumnos. Los canales de comunicación que proporciona Internet (correo electrónico, foros, chat...) facilitan el contacto entre los alumnos y con los profesores. De esta manera es más fácil preguntar dudas en el momento en que surgen, compartir ideas, intercambiar recursos, debatir. [3]
- ✓ Aprendizaje cooperativo. Los instrumentos que proporcionan las TIC (fuentes de información, materiales interactivos, correo electrónico, espacio compartido de disco, foros...) facilitan el trabajo en grupo y el cultivo de actitudes sociales, el intercambio de ideas, la cooperación y el desarrollo de la personalidad. [3]

Cuba a pesar de ser un país subdesarrollado ha ido extendiendo la informatización por toda la isla durante algunos años cumpliendo gran relevancia en la educación, pues se han implantado nuevas tecnologías y nuevas formas de impartir las clases en el aula siendo un gran reto tanto para los profesores como para los estudiantes.

En las escuelas primarias, secundarias, preuniversitarios, y en universidades del país se ha experimentado este nuevo paradigma, lo cual ha mostrado resultados satisfactorios, constituyendo un ejemplo hoy la Universidad de las Ciencias Informáticas (UCI) donde se encuentran numerosas tecnologías y donde se imparten además asignaturas que se vinculan a esta nueva cultura tecnológica.

Dentro del plan de estudio de la universidad se incluye la asignatura Gestión de Software, cuyo objetivo es introducir al estudiante el Proceso de Software Personal (PSP), el cual fue diseñado para ayudar a los ingenieros del software a hacer bien su trabajo, a estimar y planificar su tiempo y actividades, y cómo mejorar la calidad de los programas que produce.

El PSP como herramienta brinda una serie de plantillas o formularios, donde el ingeniero registra su planificación y estimación, defectos de proyectos que son los elementos claves del Cuaderno de Ingeniería.

Sin embargo la creación y el llenado diario de las plantillas del cuaderno consumía mucho tiempo a los estudiantes ya que se realizaba manualmente, además se corría el riesgo de registrar datos erróneos, o se podía perder el cuaderno y con ello la información almacenada en él. Por parte del profesor se dificultaba llevar un control sistemático del desempeño de los estudiantes.

De ahí surge la **necesidad** de hacer un software de apoyo a la docencia que garantice la rápida y segura gestión, así como el control del Cuaderno de Ingeniería.

El **problema científico** se puede formular entonces de la siguiente manera: ¿Cómo facilitar la gestión y control del trabajo sistemático del Cuaderno de Ingeniería mediante un sistema automatizado?

Por tanto **el objeto de estudio** de este trabajo es el Proceso de gestión y control del Cuaderno de Ingeniería en la asignatura de Gestión de Software como parte del Proceso de Software Personal (PSP).

De ello se deriva que el **campo de acción** comprende la automatización de la gestión y control del Cuaderno de Ingeniería en la asignatura de Gestión de Software en la Universidad de las Ciencias Informáticas (UCI).

Y el **objetivo general** es desarrollar un sistema informático para la automatización de la gestión y control del Cuaderno de Ingeniería en la asignatura de Gestión de Software.

De esta forma se puede definir las siguientes **tareas de investigación**:

1. Estudio de las características y ventajas de aplicaciones Web.
2. Investigación de la metodología del Proceso de Software Personal (PSP).
3. Investigación sobre aplicaciones similares con el objetivo de mejorar funcionalidades.
4. Estudio de las principales características del Cuaderno de Ingeniería.

5. Investigación sobre tecnologías y herramientas para el desarrollo del sistema.

Para una mejor organización del trabajo, se decidió estructurar el contenido dividido en capítulos como se muestra a continuación:

Capítulo 1: Contiene una caracterización del PSP, cuaderno de ingeniería y la fundamentación teórica de las tendencias y tecnologías actuales escogidas para el desarrollo del sistema.

Capítulo 2: Aborda sobre el objeto de estudio que incluye situación problemática, objeto de automatización, así como, un enfoque general de sistemas automatizados existentes vinculados al campo de acción y el análisis comparativo de las soluciones existentes con la propuesta. En este capítulo además se da inicio a la Ingeniería de Software, mostrando lo referente a los flujos de trabajo del negocio y requerimientos.

Capítulo 3: Trata acerca del flujo de trabajo de análisis y diseño mostrando los diagramas correspondientes al mismo como son: diagramas de clases y diagrama de colaboración del análisis, diagramas de clases y diagramas de secuencia del diseño, diagrama de clases persistentes y el modelo de datos.

Capítulo 4: Se centra en los flujos de trabajo de implementación y prueba, mostrando el diagrama de despliegue, los diagramas de componentes y las descripciones de los casos de prueba de integración comprobando la funcionalidad del sistema.

Capítulo 5: Comprende el método de estimación Cocomo II para la planificación del sistema, obteniendo el esfuerzo y costo del proyecto. Se realiza además el análisis del beneficio tangible e intangible y el análisis del costo.

CAPÍTULO 1

Fundamentación Teórica

Introducción

Este capítulo contiene una caracterización sobre el PSP y Cuaderno de Ingeniería, además de abordar sobre la metodología, las tecnologías y herramientas actuales de desarrollo utilizadas para el análisis, diseño e implementación del sistema sobre las cuales se apoya la propuesta.

1.2 Caracterización del Proceso de Software Personal y Cuaderno de Ingeniería

Estudio de las características del Proceso de Software Personal en conjunto con el Cuaderno de Ingeniería.

1.2.1 Caracterización del Proceso de Software Personal (PSP)

El proceso de Software Personal (PSP, Personal Software Process) es un proceso de automejoramiento diseñado para ayudar a controlar, administrar y mejorar la forma en que se trabaja individualmente. Está estructurado por formularios, guías y procedimientos para desarrollar softwares de alta calidad. Además el PSP se convierte en una guía para fomentar la disciplina que deben mantener el ingeniero en el desarrollo de un software.

El PSP entre otras cosas muestra a los ingenieros como:

- ✓ manejar la calidad de sus proyectos
- ✓ mejorar, estimar y planificar su tiempo
- ✓ reducir los defectos en sus productos

Las ventajas que brinda el trabajo con el PSP son las siguientes:

- ✓ productos de más alta calidad [6]

- ✓ duraciones de ciclo reducidas [6]
- ✓ productividad creciente [6]
- ✓ seguridad mejorada [6]

1.2.2 Características del Cuaderno de Ingeniería

EL Cuaderno de Ingeniería es una herramienta que permite el PSP para gestionar el tiempo de un ingeniero, posibilita realizar un registro de los defectos encontrados en un proyecto durante el desarrollo del mismo para así obtener un producto de alta calidad. Además tiene gran importancia porque introduce el trabajo real que debe ejercer un ingeniero cuando desarrolla un software, enseña a tomar con seriedad la planificación del tiempo y de las actividades, lo que permite obtener como resultado un software con las características deseadas.

El Cuaderno de Ingeniería puede ser utilizado también para guardar los ejercicios, controlar los compromisos, tomar nota de clases y como un cuaderno de trabajo para anotar ideas de diseño y cálculos. [7]

Para ello existen plantillas o formularios, de los cuales mencionaremos algunos:

- ✓ Registro de tiempo, se registra la fecha de la realización de la actividad, el tiempo de inicio y fin de la actividad, tiempo dedicado a cada actividad, el tiempo de interrupción, la actividad, entre otros.
- ✓ Resumen semanal, proporciona un registro del tiempo dedicado a cada actividad y el tiempo medio, máximo y mínimo dedicado a la actividad.
- ✓ Estimación semanal de actividades, se registra los días de la semana, las actividades correspondientes a cada día y el tiempo que le dedicará a cada actividad la semana próxima.
- ✓ Compromisos, se registra los compromisos actuales, se incluye que es lo que hay que hacer y cuando, y se añade una estimación del tiempo que le dedicará a cada compromiso.
- ✓ Registro de defectos, se registra los defectos detectados en un proyecto en los distintos flujos de trabajo, lo cual constituye una gran ventaja pues para próximos proyectos estos errores no se vuelven a cometer, además son datos históricos que quedan registrados y que permiten comparar cuando se haga otros proyectos similares y verificar si se ha mejorado o no.

1.3 Tendencias y tecnologías actuales

Estudio de las tendencias y tecnologías actuales posibles a emplear, descritas a continuación.

1.3.1 Aplicaciones Web

El diseño de aplicaciones Web ha alcanzado en la actualidad un gran auge en la industria de la informatización, en el mercado de la publicidad en Internet, en el comercio electrónico, e incluso en la educación. Es un servicio que permite la estabilidad y disponibilidad para que el usuario pueda acceder a las diferentes informaciones a la hora que desee. Además constituye una interfaz entre el usuario y el otro lado de la red.

En ingeniería de software una aplicación web es aquella que los usuarios usan accediendo a un servidor web a través de Internet o de una Intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero. La habilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Aplicaciones como los webmails, weblogs, tiendas en línea y la Wikipedia son ejemplos bien conocidos de aplicaciones web. [8]

Aunque muchas variaciones son posibles, una aplicación web está comúnmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web es la primera capa, un motor usando alguna tecnología web dinámica (ejemplo: PHP o ASP) es la capa de en medio, y una base de datos como última capa. [9]

1.3.2 Comparación entre aplicaciones Web y aplicaciones Desktop

	Aplicaciones Desktop	Aplicaciones Web
Cobertura Global	Menos	Más
Requerimiento de hardware	Mayor	Menor
Fácil uso	Menos	Más

Mantenimiento	Menos	Más
Seguridad	Mayor	Menor
Tabla 1.1: Comparación entre aplicaciones Web y aplicaciones Desktop		

Por las funcionalidades que comprende una aplicación Web se decide optar por ella para la realización del sistema, ya que emplea una arquitectura cliente-servidor lo que hace que se pueda acceder a la información en cualquier momento y desde cualquier lugar, lo que facilita el trabajo de los usuarios. En fin la aplicación web brinda libertad de tiempo y lugar de trabajo.

1.3.3 Modelo Cliente Servidor

La arquitectura cliente-servidor, llamado, modelo cliente-servidor o servidor-cliente es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificarlas. [10]

En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes.

En la funcionalidad de un programa distribuido se pueden distinguir 3 capas o niveles:

- ✓ Manejador de Base de Datos (Nivel de almacenamiento). [10]
- ✓ Procesador de aplicaciones o reglas del negocio (Nivel lógico)
- ✓ Interfaz del usuario (Nivel de presentación). [10]

En la actualidad se suele hablar de arquitectura de tres niveles, donde la capa de almacenamiento y la de aplicación se ubican en (al menos) dos servidores diferentes, conocidos como servidores de datos y servidores de aplicaciones para lograr una mejor funcionalidad. [10]

Ventajas de la arquitectura cliente-servidor:

- ✓ El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes. [10]

- ✓ Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre para otra conexión. [10]

La decisión de soportar esta arquitectura surgió por la necesidad de una mayor organización del código, una funcionalidad más proporcionada, además de contar con una mayor fortaleza y menos riesgo de que intrusos corrompan el sistema. Proporciona también la construcción de un sistema distribuido.

1.3.4 Lenguaje PHP

PHP, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Este lenguaje está orientado para la realización de aplicaciones Web dinámicas. El fácil uso y la similitud con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores experimentados crear aplicaciones complejas con una curva de aprendizaje muy suave y les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas. [11]

Su interpretación y ejecución se da en el servidor, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página Web generada por un script PHP, el servidor ejecuta el intérprete de PHP procesando el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor que se encarga de regresárselo al cliente. Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos, entre otras cosas. [11]

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones Web muy robustas. [11]

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI. Está en el entorno código

abierto (open source) lo que permite de forma gratuita, una enorme cantidad de recursos: el lenguaje en sí, el servidor para ejecutarlo, manuales y tutoriales, y sobre todo códigos que una vez obtenidos puedes modificarlo y redistribuirlo libremente. Posee la capacidad de incrustar código PHP en las páginas HTML. Brinda estabilidad y bajo consumo de recursos. [11]

Otros lenguajes como Perl (Practical Extraction and Report Language), ASP (Active Server Pages) y JSP (Java Server Pages) tienen características similares al PHP aunque poseen rasgos que los marcan y por ello los distingue, entre ellos podemos encontrar:

- ✓ Características multiplataformas: Menos el ASP, que es solamente soportado por la plataforma Windows, los demás lenguajes están soportados en múltiples plataformas.
- ✓ Velocidad de ejecución: La velocidad es mayor en PHP, seguidos por PERL y JSP.
- ✓ Disponibilidad de recursos: Actualmente los más utilizados en la Internet son el PHP y el JSP, siendo más utilizado en la publicación de artículos y códigos de ejemplos. PHP tiene una de las comunidades más grandes en Internet, al igual que la de Java.
- ✓ Familiaridad con el lenguaje: En la universidad los lenguajes más utilizados por los programadores es el ASP y el PHP.

Teniendo en cuenta las comparaciones y características antes descritas se escoge al lenguaje PHP, la versión 5.0 para el desarrollo de la aplicación. Esta versión presenta las características necesarias para trabajar con una Programación Orientada a Objeto (POO), permitiendo la reutilización de funciones o métodos.

1.3.5 Servidor Web Apache

Un servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. [13]

Apache es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. [14]

Apache presenta entre otras características:

- ✓ Es un servidor altamente configurable de diseño modular. [12]
- ✓ Es muy sencillo ampliar las capacidades de este servidor. Actualmente existen muchos módulos que son adaptables a él, y están ahí para que sean instalados cuando se necesiten. [12]
- ✓ Trabaja con gran cantidad de lenguajes como Perl, PHP y otros lenguajes de script.
- ✓ Funciona sobre muchas plataformas (Unix, Linux, Win32, OS2). [12]
- ✓ Alto desempeño. [12]

Este servidor Web tiene amplia aceptación en la red: en el 2005, 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. [12]

Teniendo en cuenta las potencialidades que posee el servidor Web Apache es el más indicado para alojar la aplicación Web.

1.3.6 Lenguaje XML

XML (o lenguaje de anotación extensible) es desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del lenguaje SGML (sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial), estas las defines el usuario a diferencia de HTML que ya están predefinidas. Es válido aclarar que XML no es ni será nunca un reemplazo del HTML sino un complemento que sirve para manejar la información separada del formato. [15]

Se hizo uso de este lenguaje de etiquetas porque:

- ✓ Es facilitador de cargar dinámicamente los contenidos desde un fichero sin necesidad de almacenarlos en la base de datos. [16]

- ✓ Permite estructurar la información y luego aplicarle fácilmente transformaciones para su presentación, lo que ayuda a ganar en tiempo. [16]
- ✓ La mayoría de los navegadores actuales son capaces de entender XML. [16]
- ✓ La comunicación de datos. Cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella. [16]
- ✓ Permite la compatibilidad entre sistemas. Comparte la información de una manera segura, fiable y fácil. [16]

Aunque es una tecnología sencilla tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Por eso la idea de unirlo a PHP y a HTML, lo que permitió una mayor organización y seguridad en la aplicación, cargar los menús y mensajes dinámicamente.

1.3.7 Lenguaje JavaScript

Los lenguajes "Script" son versiones resumidas de otros lenguajes. Un código escrito en lenguaje script se incorpora dentro de un código HTML y se ejecuta sin que sea necesario compilarlo, son interpretados directamente cada vez que sea necesaria una ejecución. Permiten definir efectos de la página Web para hacer más atractivo su aspecto gráfico, e interactuar con el visitante. Los más utilizados son JavaScript y VBScript. [17]

El más popular de los lenguajes script es el "JavaScript", el cual es soportado por los navegadores Internet Explorer, Netscape Navigator, Mozilla, entre otros. [17]

El lenguaje de programación JavaScript soporta scripts del lado del cliente, permitiéndole a las páginas HTML ejecutar funciones y scripts. Estos scripts pueden ayudar a la implementación de comportamientos muy complejos en las páginas Web con un esfuerzo de programación relativamente poco. [17]

En la aplicación el uso de Javascript se basa en las validaciones de los formularios del lado del cliente, para permitir que los datos que se entren sean los correctos.

1.3.8 AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos, donde XML es un acrónimo de eXtensible Markup Language), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. [30]

Por las funcionalidades que brinda AJAX es usado en la aplicación para obtener datos del servidor de forma asincrónica sin necesidad de hacer un recargo de la página.

1.3.9 Gestores de base de datos

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos y el usuario. Se compone de un lenguaje de definición y manipulación de datos y de un lenguaje de consulta. [18]

Los SGBD poseen grandes ventajas como son:

- ✓ Facilidad de manejo de grandes volúmenes de información. [18]
- ✓ Gran velocidad en muy poco tiempo. [18]
- ✓ Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta. [18]
- ✓ No hay duplicidad de información, comprobación de información en el momento de introducir la misma. [18]

1.3.9.1 MySQL

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo, esto permite velocidad y flexibilidad. [20]

MySQL hoy en día es uno de los gestores de base de datos más utilizados en el mundo pues:

- ✓ Es un gestor de base de datos de fuente abierta, o sea, es posible para cualquier persona usarlo y modificarlo gratuitamente. [21]
- ✓ Es muy rápido, confiable y fácil de usar. [21]
- ✓ Puede ser usado por muchos lenguajes de programación como Perl, Java, C, C#, C++, Pascal, Delphi, Lisp, Ruby y PHP. [21]
- ✓ Es multihilo y multiusuario. [21]
- ✓ Posee buena escalabilidad, o sea, es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla. [21]
- ✓ Posee gran flexibilidad ya que trabaja en más de 20 plataformas incluyendo Linux, Windows, OS/X, HP-UX, AIX, Netware. [21]

1.3.9.2 PostgreSql

PostgreSql es un potente sistema de base de datos relacional libre. Funciona en todos los sistemas operativos importantes, incluyendo Linux, UNIX y Windows. Tiene soporte total para foreign keys, joins, views, triggers, y stored Procedures (procedimientos almacenados) en múltiples lenguajes. Tiene interfaces de programación nativos para C/ C++, Java, Perl, Python, Ruby, Tcl, ODBC, entre otros. [19]

PostgreSQL posee alta concurrencia pues permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo a la tabla. Es extensible, pues el código fuente está disponible para todos sin costo. [21]

A pesar de que este SGBD posee tan buenas características posee una pequeña deficiencia que no es tan notable o mejor dicho es bueno en proyectos grandes, ésta es la velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes, cosa que resulta loable. [22]

Para el desarrollo de la aplicación se hizo uso de dos gestores de base de datos muy utilizados hoy en día, éstos son: MySql y PostgreSQL, éste último para dar mayores opciones en correspondencia con la situación que presente el lugar donde se vaya a trabajar con el sistema.

El uso de MySql es factible si el lugar donde se pone en práctica no tiene muchos usuarios, ganando así en velocidad, un ejemplo son las Facultades Regionales pertenecientes a la UCI donde la matrícula es pequeña, por lo que la concurrencia no es muy grande.

Lo contrario ocurre con PostgreSQL, donde lo más importante es la concurrencia, por lo que es más recomendable para usar en lugares donde la cantidad de usuarios sea grande como por ejemplo aquí en la universidad, aunque se pierde en velocidad, pero se asegura la estabilidad del sistema.

1.4 Patrones de arquitectura

Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Ayudan a especificar la estructura fundamental de una aplicación. Cada patrón de arquitectura ayuda a conseguir una propiedad específica en el sistema global; por ejemplo, la adaptabilidad de la interfaz de usuario. Los patrones que dan soporte a características similares se agrupan en una misma categoría. [23]

Un ejemplo de patrón de arquitectura es el Modelo-Vista-Controlador (Model-View-Controller o MVC) fue introducido inicialmente en la comunidad de desarrolladores de Smalltalk-80. MVC divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- ✓ Modelo (Model): Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada. [23]
- ✓ Vista (View): Muestra la información al usuario. Obtiene los datos del modelo. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador. [23]
- ✓ Controlador (Controller): Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son

traducidos a solicitudes de servicio (“service requests” en el texto original) para el modelo o la vista. El usuario interactúa con el sistema a través de los controladores. [23]

El MVC es un patrón ampliamente utilizado en múltiples plataformas y lenguajes. El sistema se implementó basado en este patrón de arquitectura debido a la flexibilidad, adaptabilidad y confiabilidad que brinda al sistema, pues se distribuye el código de manera organizada en capas, cada una respondiendo a sus funcionalidades correspondientes. La Figura representada en el [\(Anexo I\)](#) muestra la estructura del patrón MVC.

1.5 Metodología empleada en la Ingeniería de Software

Se conoce que la construcción de un sistema de software implica la toma de decisiones sobre la arquitectura del mismo, así como definir los componentes y sus interacciones. Estas decisiones pueden ser cruciales para el éxito o fracaso del sistema resultante, por lo que se requiere seleccionar un proceso de desarrollo de software con el fin de obtener la calidad del sistema de software deseada y que cumpla con los requerimientos establecidos. [24]

Metodologías vigentes de ingeniería de software atienden muy bien estos requerimientos. Estas metodologías son: eXtreme Programming (XP), Microsoft Solution Framework (MSF), Proceso Unificado de Desarrollo (RUP), entre otras. [24]

RUP es la metodología que se escogió para trabajar en el análisis y diseño del sistema porque:

- ✓ Es adaptable a cualquier cambio, o sea, es un marco de trabajo genérico por lo que se pueden realizar las adecuaciones necesarias al proceso según la naturaleza del proyecto que se desea afrontar.[24]
- ✓ Está dirigido por casos de uso, ya que éstos no sólo inician el proceso de desarrollo, sino que le proporcionan un hilo conductor y siguiendo este hilo avanza a través de una serie de flujos de trabajo, pues los casos de uso se especifican, se diseñan y los casos de usos finales son la fuente a partir de la cual se construyen los casos de prueba. [24]

- ✓ Es un proceso centrado en la arquitectura por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. [24]
- ✓ Es un proceso iterativo e incremental, puesto que una iteración es un miniproyecto que resulta en un incremento. Las iteraciones hacen referencias a pasos en el flujo de trabajo y los incrementos, al crecimiento del producto. Cada iteración debe ser planificada, es por esto por lo que son miniproyectos. [24]
- ✓ Posibilita la captura de requisitos tanto funcionales como no funcionales, lo que constituye un paso decisivo en el desarrollo del sistema y permite además la verificación de la calidad del software. [24]
- ✓ Es un Proceso Integrado pues se establece una estructura que abarque los ciclos, fases, flujos de trabajo, mitigación de riesgos, control de calidad, gestión del proyecto y control de configuración. [24]

Por tanto la metodología RUP garantiza la obtención de software de calidad y esta es la razón fundamental por la cual se hizo uso de ella.

1.5.1 Herramienta y lenguaje aplicado en la metodología RUP

Rational Rose es una herramienta que permite visualizar, entender, y refinar los requerimientos y arquitectura antes del enfrentamiento al código. Esto permite evitar esfuerzos desperdiciados en el ciclo de desarrollo. [25]

Rational Rose tiene todas las características que como desarrolladores, analistas, y arquitectos se desea, ya que:

- ✓ Soporta UML. [26]
- ✓ Brinda completo soporte al equipo. [26]
- ✓ Presenta gran facilidad de uso, integración optimizada, y mucho más. [26]

Por todas estas facilidades que brinda esta herramienta en conjunto con el lenguaje UML se decidió trabajar con ella.

UML (Lenguaje Unificado de Modelado), es el lenguaje de modelado de sistemas estándar para el modelamiento de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática. [27]

UML permite definir y modelar el sistema con tecnología orientada a objeto (OO), posibilita visualizar el producto del trabajo en esquemas o diagramas estandarizados y permite documentar. Brinda además la posibilidad de generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con el que siempre se pueden mantener la visión del diseño de más alto nivel de la estructura de un proyecto. [27]

Todas estas posibilidades que brinda dicho lenguaje son importantes a la hora de hacer el modelado, ya que incluye todos los conceptos que se consideran necesarios para utilizar un proceso iterativo, basado en construir una sólida arquitectura para resolver requisitos que conforman los casos de uso.

1.6 Herramientas empleadas para la programación y diseño web

Las herramientas de programación sirven para soportar los diferentes lenguajes de programación para los cuales estén destinadas. Unas presentan interfaz gráfica, otras no las tienen, otras son para realizar aplicaciones desktop, otras para hacerlas Web, etc.

Para el desarrollo del sistema se utilizó:

- ✓ Dreamweaver
- ✓ Zend Studio
- ✓ Photoshop

1.7 Conclusiones

Cuando se desarrolla una aplicación Web, se debe tener en cuenta desde el levantamiento de requisitos hasta la implementación del código considerando, el lenguaje de programación, la herramienta a utilizar, gestor de base de datos, arquitectura y metodología. Estos aspectos fueron expuestos en este capítulo

justificando la elección de cada uno. También se abordó algunos temas de interés a los que está dirigido el desarrollo de la aplicación para un mejor entendimiento de la misma.

CAPÍTULO 2
Características del sistema**2.1 Introducción**

En este capítulo se plantea el objeto de estudio que incluye situación problemática, objeto de automatización, así como un enfoque general de sistemas automatizados existentes vinculados al campo de acción y el análisis comparativo de las soluciones existentes con la propuesta dada en este trabajo.

También se muestra el negocio para poder tener una visión clara del proyecto a realizar, definir además los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos. Se realiza el levantamiento de los requisitos funcionales y no funcionales para determinar las funcionalidades que tendrá el sistema y a partir de aquí se realizará el modelado del sistema.

Para la modelación del sistema, análisis y diseño se expondrán solamente los casos de uso críticos, o sea, los más importantes para el funcionamiento del sistema.

2.2 Objeto de estudio

La necesidad de que las situaciones se resuelvan lo más rápido posible, fuera del alcance de las funciones que tiene el hombre por naturaleza, va creciendo cada día. De ahí el surgimiento y desarrollo de las nuevas tecnologías que se van fortaleciendo en todo momento y han contribuido enormemente en la educación de los estudiantes.

La entrada y salida de datos del ordenador es de manera inmediata, lo cual resulta ser muy práctico. Por ello se hace muy importante la automatización de procesos que en la actualidad se trabajan de forma manual, así como el estudio y análisis detallado del almacenamiento y obtención de las informaciones.

Nuestra **situación problemática** se basa en las dificultades que presenta la realización manual del llenado del Cuaderno de Ingeniería, pues resulta muy trabajoso la actualización diaria y la creación de las plantillas del cuaderno. Si el estudiante se equivoca al poner algún dato siente la necesidad de borrar o tachar, lo cual no contribuye a la limpieza y cuidado del cuaderno. Además la realización de muchas de las plantillas conlleva parte del tiempo que se puede dedicar al registro de las tareas. Las posibilidades de errores de cálculo son mayores y en caso de que se extravíe el cuaderno se perderá toda la información contenida. En el caso de los profesores se dificulta llevar un control sistemático del trabajo realizado por cada uno de sus estudiantes.

Como **objeto de automatización** centramos la atención en dos procesos:

- ✓ Proceso de creación y actualización del Cuaderno de Ingeniería.

Este proceso comienza cuando el estudiante crea por primera vez su cuaderno, de ahí en lo adelante registrará cada tarea realizada con el fin de controlar su tiempo.

- ✓ Control de las actividades registradas en el Cuaderno de Ingeniería.

De esta manera el profesor tendrá un control sistemático sobre las actividades de sus estudiantes y así tener una calificación de los mismos.

Nuestra **propuesta** es un sistema informático capaz de automatizar los procesos de creación, actualización y control del Cuaderno de Ingeniería, logrando una mayor rapidez en el manejo de las informaciones.

2.2.1 Sistema automatizado existente vinculado al campo de acción

“Process dashboard” es un sistema que se vincula a nuestro campo de acción, cuyo objetivo es el de apoyar el trabajo de un ingeniero o un equipo de ingenieros desarrolladores de software en cuanto a planificación y estimación de su tiempo. Esta aplicación fue desarrollada originalmente en 1998 por la fuerza aérea de Estados Unidos, y ha continuado desarrollándose bajo modelo de código abierto. [4]

El Process Dashboard es un software que intenta mantenerse lo más pequeño posible en el desktop de la computadora, para no estorbar el trabajo del ingeniero en el desarrollo del software. En su pequeño espacio de la pantalla la aplicación, da acceso a un cronómetro para medir el tiempo de las actividades que realiza en su ordenador, un formulario para registrar la información de los defectos encontrados, otros scripts y formularios para seguir los procesos definidos, una "casilla de verificación" manual que permite una manera rápida y fácil de informar a la herramienta que una fase de desarrollo ha sido completada, menús que permiten la navegación a través del proyecto de trabajo bajo una estructura jerárquica, y un menú de configuración para permitir el acceso a otras opciones de la herramienta. [5]

El Process Dashboard es una aplicación que no está realizada con el fin de apoyar a los estudiantes en la docencia sino que está dirigido hacia los ingenieros de software con experiencia en su trabajo.

Sin embargo PSPSOFT permite una rápida y eficiente recolección de los datos, permite tomar acciones correctivas a tiempo y mejorar sucesivamente los procedimientos definidos. Está diseñado para que los estudiantes implementen el Cuaderno de Ingeniería y los profesores puedan darle seguimiento y evaluar esta actividad interactuando con los estudiantes a través de mensajes y chat, también el sistema brinda la posibilidad de recoger el estado de opinión de los estudiantes por medio de encuestas permitiendo una mejor interactividad y relación profesor-estudiante.

Es importante destacar que el PSPSOFT se hizo con el objetivo de apoyar a la asignatura Gestión de Software, sin embargo esto no es limitación para que pueda ser usado por estudiantes de los diferentes proyectos productivos en la universidad.

A continuación veremos algunas de las diferencias más significativas entre el Process dashboard y PSPSOFT:

	Process dashboard	PSPSOFT
Tipo de aplicación	Desktop	Web
Propósito	Apoyar el trabajo de ingenieros que utilizan el PSP y TSP.	Apoyo en la docencia a la asignatura Gestión de Software.
Arquitectura	-----	Cliente-Servidor.
Tecnología	Java.	PHP.
Docencia	El profesor debe pasar por cada máquina donde esté el estudiante para poder controlar su trabajo.	El profesor puede controlar el trabajo del estudiante desde cualquier computadora, siempre que tenga conexión.
Tabla 2.1: Comparación entre Process dashboard y PSPSOFT		

Se puede concluir diciendo que tanto el PSPSOFT como el Process dashboard constituyen sistemas centrados en el mejoramiento de la calidad del software.

2.3 Modelo del Negocio

El modelo del negocio describe el negocio en términos de casos de usos del negocio, que corresponde a lo que comúnmente se le llama procesos.

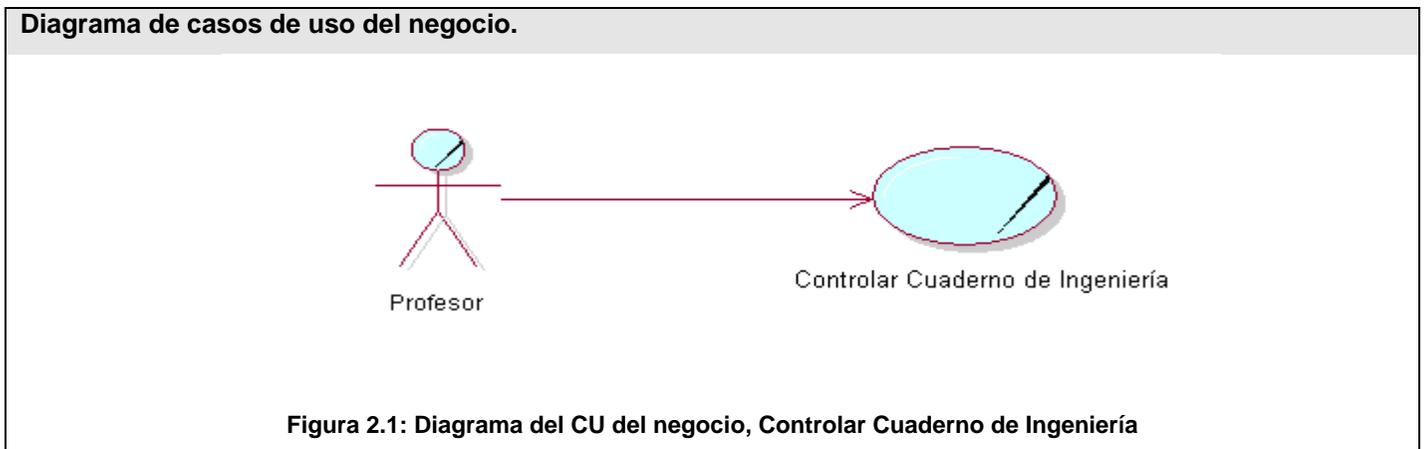
El modelo de Casos de Uso del Negocio es un modelo que describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como clientes, es decir, describe las funciones que el negocio realiza y su objetivo básico es describir cómo el negocio es utilizado por sus clientes.

2.3.1 Descripción de los actores y trabajadores del negocio

Actor del negocio	Justificación
Profesor	Interesado en controlar el registro de las actividades de los estudiantes es sus cuadernos de ingeniería.
Tabla 2.2: Descripción del Profesor, actor del negocio	

Trabajador del negocio	Justificación
Estudiante	Es el encargado de registrar en el cuaderno las actividades diarias.
Tabla 2.3: Descripción del estudiante, trabajador del negocio	

2.3.2 Representación del diagrama de casos de uso del negocio



2.3.3 Descripción de los casos de uso del negocio

La descripción de los casos de uso del negocio, describen detalladamente la interacción de los actores del negocio con el negocio, ([Ver anexo II](#)).

2.3.4 Diagrama de actividades del negocio

Los diagramas de actividades explican las actividades que ocurren dentro del proceso del negocio, separando las actividades que realizan los actores del negocio de las actividades que realizan los trabajadores del negocio, ([Ver anexo III](#)).

2.3.5 Modelo de objeto del negocio

El modelo de objeto representa las clases entidades del negocio y a los trabajadores que interactúan con ellas, ([Ver anexo IV](#)).

2.4 Requerimientos

Los requerimientos son las condiciones o capacidades que tiene que ser alcanzada por un sistema para satisfacer al cliente o usuario final.

Los requerimientos se clasifican en dos tipos: requerimientos funcionales y requerimientos no funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir mientras que los requerimientos no funcionales son características o propiedades que el sistema debe tener para ser un producto con mucha aceptación.

2.4.1 Requerimientos Funcionales

1- Autenticar usuario

- 1.1- Verificar usuario y contraseña
- 1.2- Otorgar los privilegios que le corresponde

2- Gestionar Cuaderno de Ingeniería

- 2.1- Registrar nueva actividad
- 2.2- Editar actividad
- 2.3- Imprimir plantilla del cuaderno

3- Gestionar profesor

- 3.1- Adicionar profesor
- 3.2- Eliminar profesor
- 3.3- Mostrar datos personales del profesor
- 3.4- Imprimir listado de profesores

4- Gestionar grupo

- 4.1- Agregar nuevo grupo
- 4.2- Editar grupo
- 4.3- Eliminar grupo
- 4.4- Mostrar listado de estudiantes
- 4.5- Imprimir listado de grupos

5- Gestionar estudiante

- 5.1- Agregar nuevo estudiante
- 5.2- Ver datos personales de estudiante
 - 5.2.1- Editar datos personales de estudiante
- 5.3- Eliminar estudiante
- 5.4- Imprimir listado de estudiantes

6- Gestionar semana

- 6.1- Agregar nueva semana
- 6.2- Editar semana
 - 6.2.1- Cambiar estado de la semana
- 6.3- Eliminar semana
- 6.4- Generar semana automáticamente
- 6.5- Cambiar estado de la semana
- 6.6- Imprimir listado de semanas

7- Controlar estudiante

- 7.1- Mostrar plantillas del Cuaderno de Ingeniería

7.1.1- Registrar calificación

7.1.2- Enviar mensaje al estudiante

7.2- Controlar visitas de estudiante

7.3- Imprimir datos

8- Gestionar calificación

8.1- Registrar calificación

8.2- Editar calificación

8.3- Eliminar calificación

2.4.2 Requerimientos no funcionales

- Apariencia o interfaz externa.

- ✓ Interfaz simple de usar
- ✓ Amigable
- ✓ Adaptable
- ✓ Legible

- Usabilidad.

- ✓ Fácil de aprender
- ✓ Flexibilidad
- ✓ Robustez
- ✓ Manual de usuario

- Rendimiento.

- ✓ Trabajo factible y con un gasto mínimo de recursos.

- Soporte.

- ✓ Fácil instalación
- ✓ Mantenimiento frecuente.

- Portabilidad.

- ✓ Multiplataforma

- Seguridad.

✓ Confidencialidad, pues la información manejada está protegida de acceso no autorizado y divulgación.

✓ Integridad, ya que la información manejada es protegida contra la corrupción y estado inconsistente.

✓ Disponibilidad, puesto que la información siempre está disponible y podemos obtener los datos deseados en un momento dado.

- ✓ Autenticidad, ya que el sistema presenta autenticación de usuario para su acceso.

- Ayuda y documentación en línea.

- ✓ Ayuda técnica sobre el sistema mediante Web.

- Software.

- ✓ Servidor Web Apache 2.0 con PHP 5.0.
- ✓ MySQL 5.0.
- ✓ PostgreSQL 8.2

- Hardware.

✓ PC o Servidor con microprocesador Pentium IV o superior, memoria RAM 256 Mb o superior, Memoria física superior a 500 Mb.

2.5 Diagrama de casos de uso del sistema

El diagrama de casos de uso del sistema muestra la relación existente entre los procesos o casos de uso del sistema y los actores del sistema. Estos casos de uso son resultado de la captura de requisitos funcionales que son los que indican las funcionalidades que debe cumplir el sistema.

2.5.1 Definición de los actores del sistema a automatizar

Actores del Sistema	Justificación
Usuario	Es un usuario que generaliza el rol de autenticación al sistema, de salir del sistema correctamente.
Estudiante	Es el encargado de realizar la gestión del Cuaderno de Ingeniería.
Profesor	Es el encargado principalmente de controlar las actividades registradas por cada uno de los estudiantes en sus respectivos cuadernos de ingeniería.
Administrador	Es el encargado de administrar la aplicación.

Tabla 2.4: Actores del sistema y su justificación

2.5.2 Representación del diagrama de casos de uso del sistema

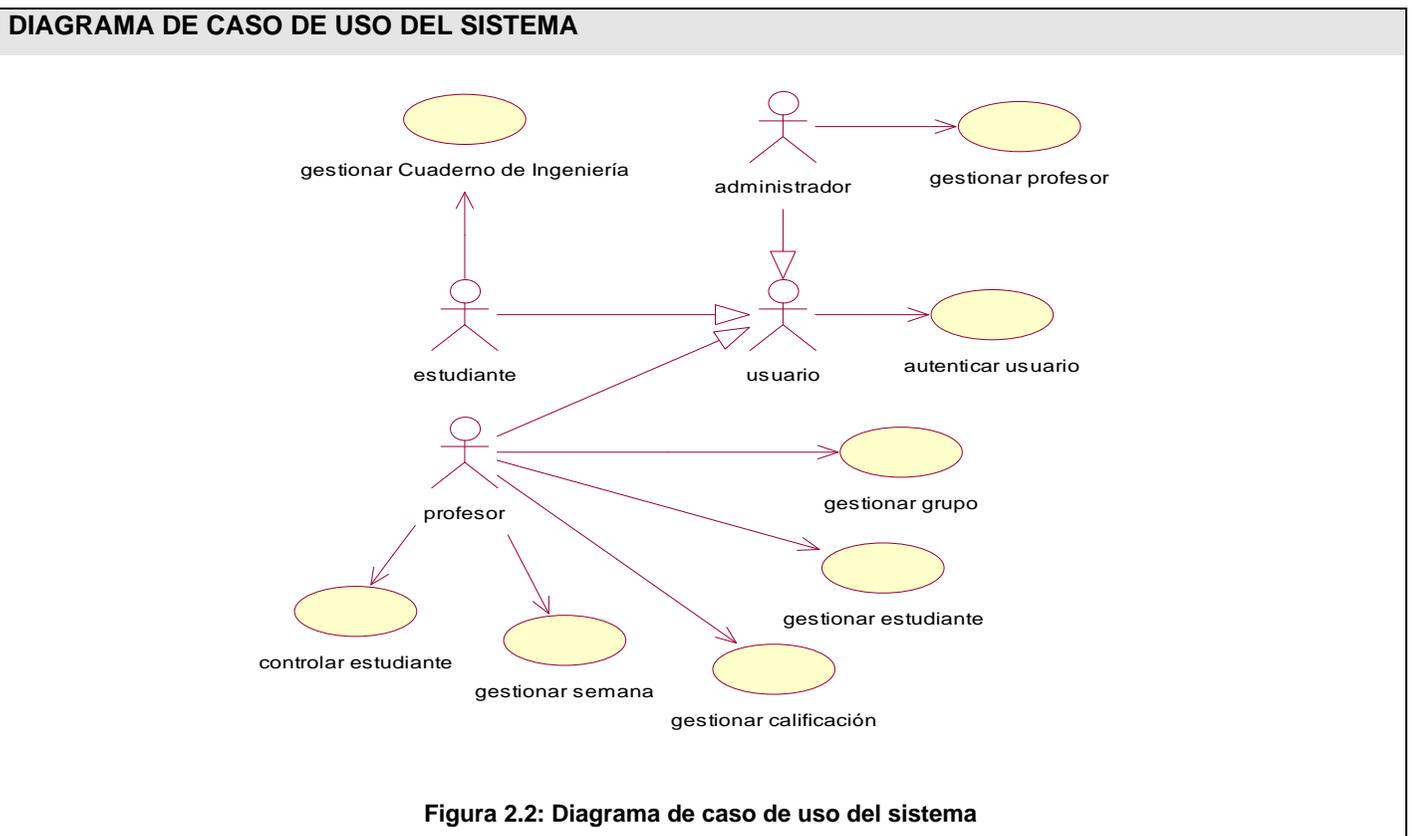


Figura 2.2: Diagrama de caso de uso del sistema

2.5.3 Descripción de los casos de uso del sistema

La descripción de los casos de uso del sistema, describen de forma detallada la interacción de los actores del sistema con el sistema, ésta se divide por escenario y cada escenario es un requerimiento, un caso de uso debe tener varios escenarios, ([Ver anexo V](#)).

2.6 Conclusiones

A través de este capítulo se mostró el objeto de estudio, situación problemática, objeto de automatización, así como un enfoque general de sistemas automatizados existentes vinculados al campo de acción y el análisis comparativo de las soluciones existentes con la propuesta dada en este trabajo.

Además se realizó el diagrama de casos de uso del negocio, modelo de objeto, los requisitos tanto funcionales como no funcionales los cuales definen las funcionalidades y restricciones del sistema, diagrama de casos de uso del sistema, y sus descripciones.

CAPÍTULO 3

Análisis y diseño del sistema

3.1 Introducción

Este capítulo trata acerca de traducir los requisitos a una especificación que describe cómo implementar el sistema, conformando así el análisis como paso intermedio para llegar a la elaboración de un diseño que cumpla con todos los requisitos establecidos.

En este capítulo se muestran los diagramas clases del análisis, así como en el diseño se mostrará los diagramas de clases del diseño de los casos de uso y diagramas de interacción dentro de éste se realizó el diagrama de secuencia del diseño. Se muestran además el diseño de la base de datos, dentro de este, el modelo lógico y el modelo físico.

3.2 Diagrama de clases del análisis

Los diagramas de clases de análisis contienen a las clases (que se centran en los requisitos funcionales), y a las relaciones entre las distintas clases y el actor. Estas clases se dividen en tres:

- 1- Clase Interfaz mediante la cual el actor interactúa con el sistema.
- 2- Clase de control la cual coordina la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.
- 3- Clase entidad la cual modela información que posee larga vida y que es a menudo persistente.

3.2.1 Representación de los diagramas de clase del análisis

Justificación de la clase Plantilla_cuaderno tanto en el análisis como en el diseño.

La clase Plantilla_cuaderno se usa en representación de todas las plantillas del Cuaderno de Ingeniería debido a que son un gran número de plantillas, con esta representación se logra una mejor comprensión y

claridad del diagrama. Ella está presente en los casos de uso: Gestionar Cuaderno de Ingeniería y en Controlar estudiante.

DIAGRAMA DE CLASES DE ANÁLISIS, caso de uso: Autenticar usuario

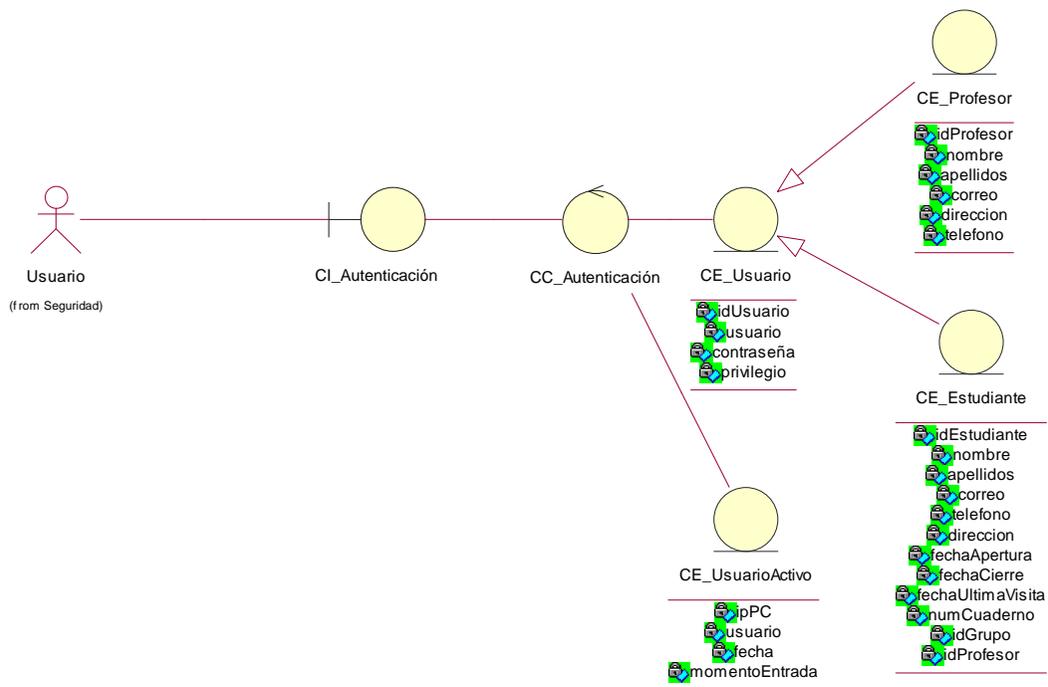


Figura 3.1: Diagrama de clases de análisis, Autenticar usuario

DIAGRAMA DE CLASES DE ANÁLISIS, caso de uso: Gestionar profesor

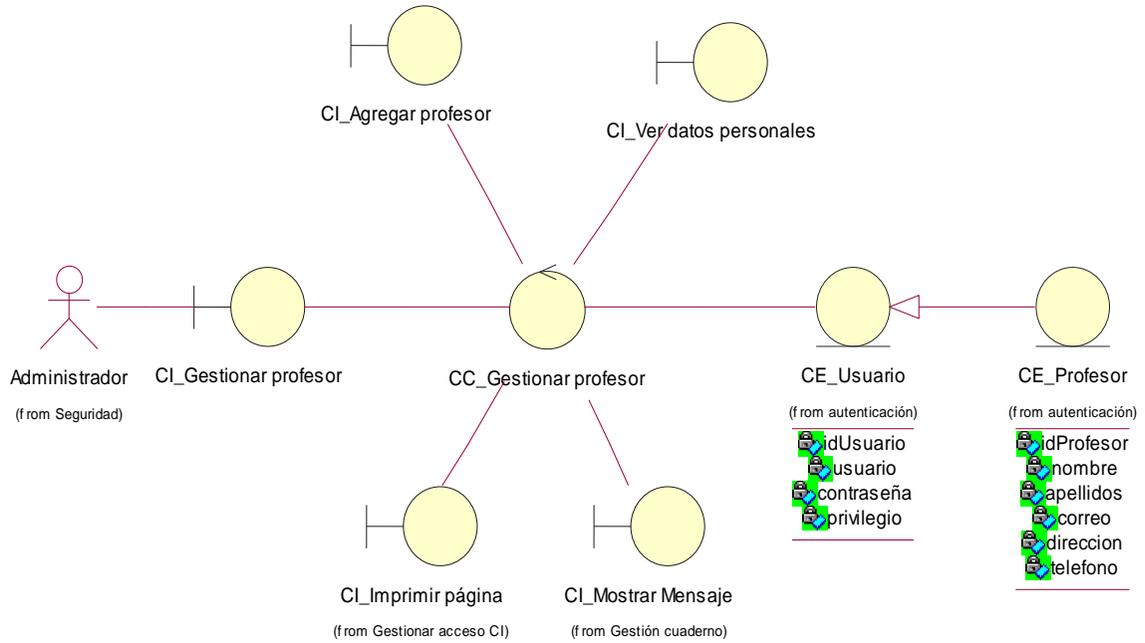


Figura 3.2: Diagrama de clases de análisis, Gestionar profesor

DIAGRAMA DE CLASES DE ANÁLISIS, caso de uso: Gestionar Cuaderno de Ingeniería

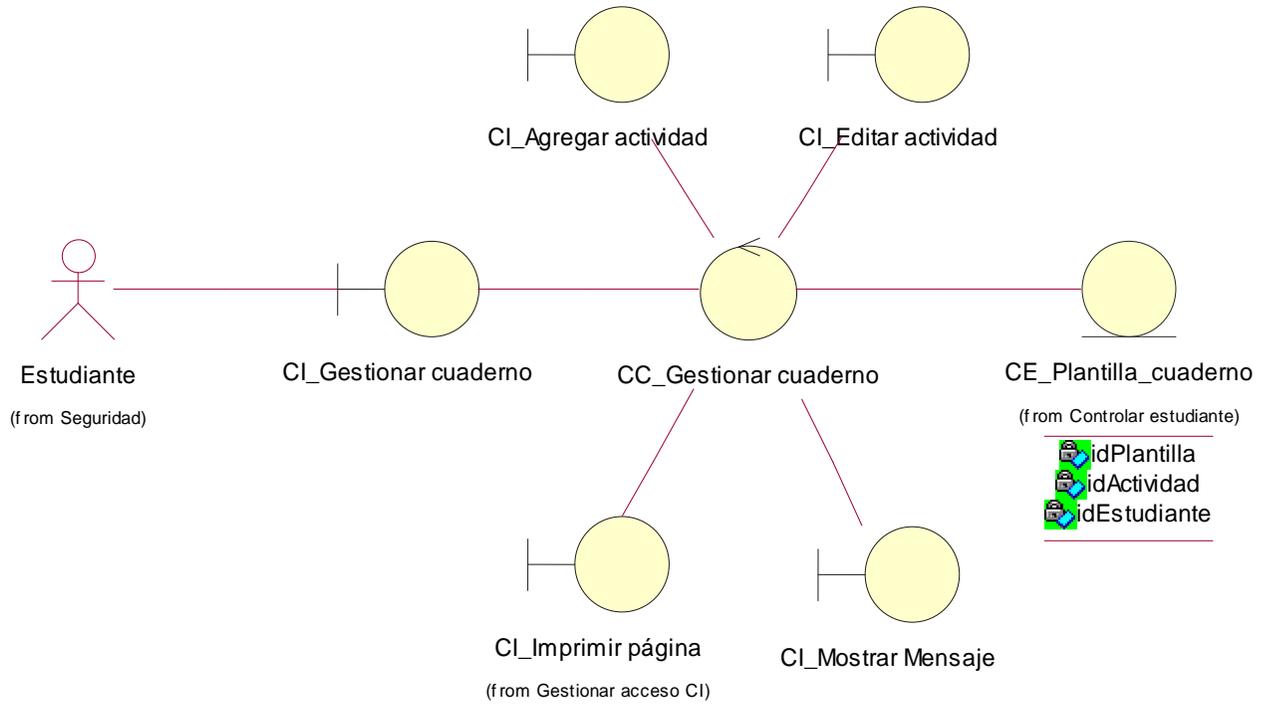


Figura 3.3: Diagrama de clases de análisis, Gestionar Cuaderno de Ingeniería

DIAGRAMA DE CLASES DE ANÁLISIS, caso de uso: Gestionar grupo

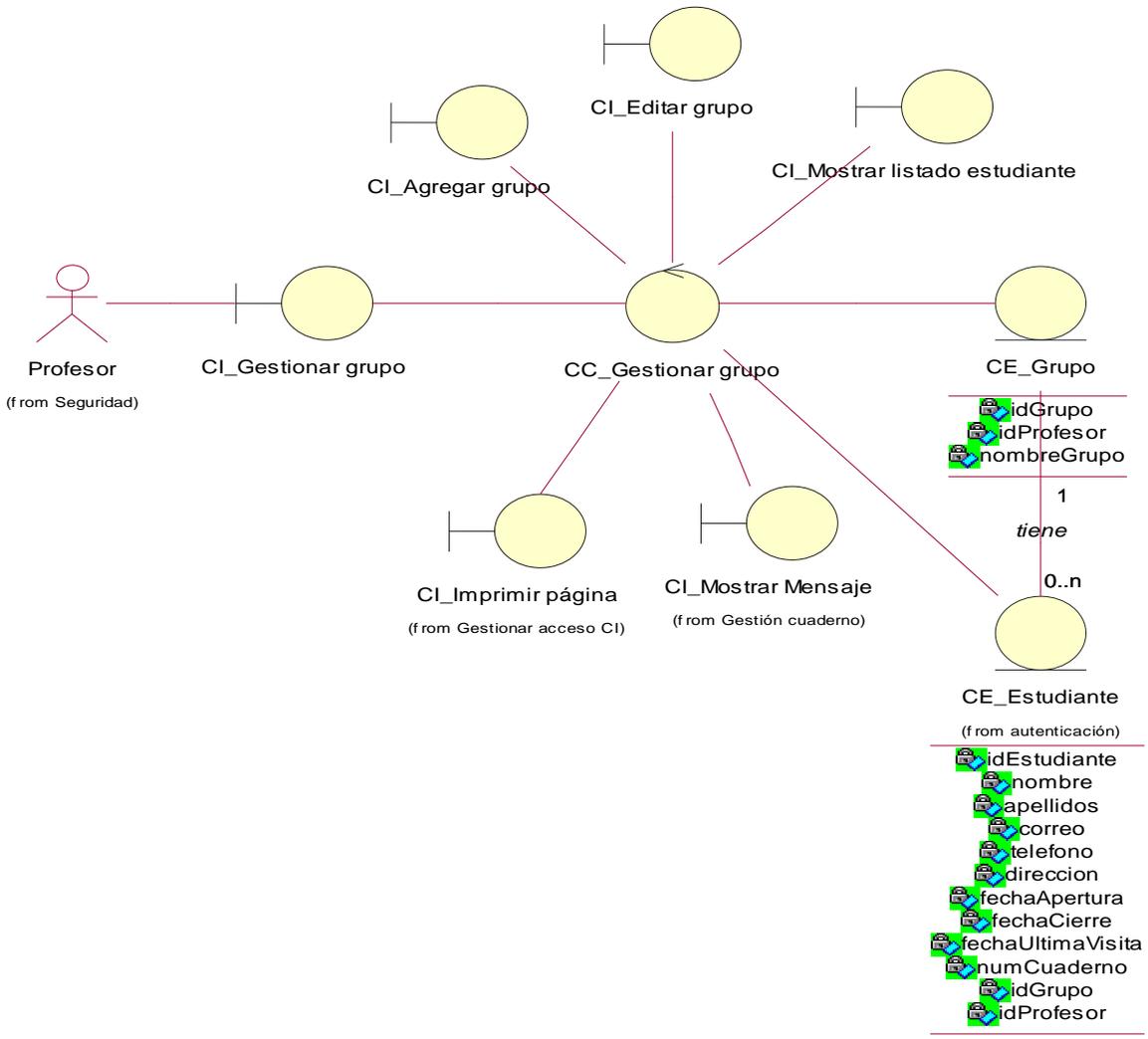


Figura 3.4: Diagrama de clases de análisis, Gestionar grupo

DIAGRAMA DE CLASES DE ANÁLISIS, caso de uso: Gestionar estudiante

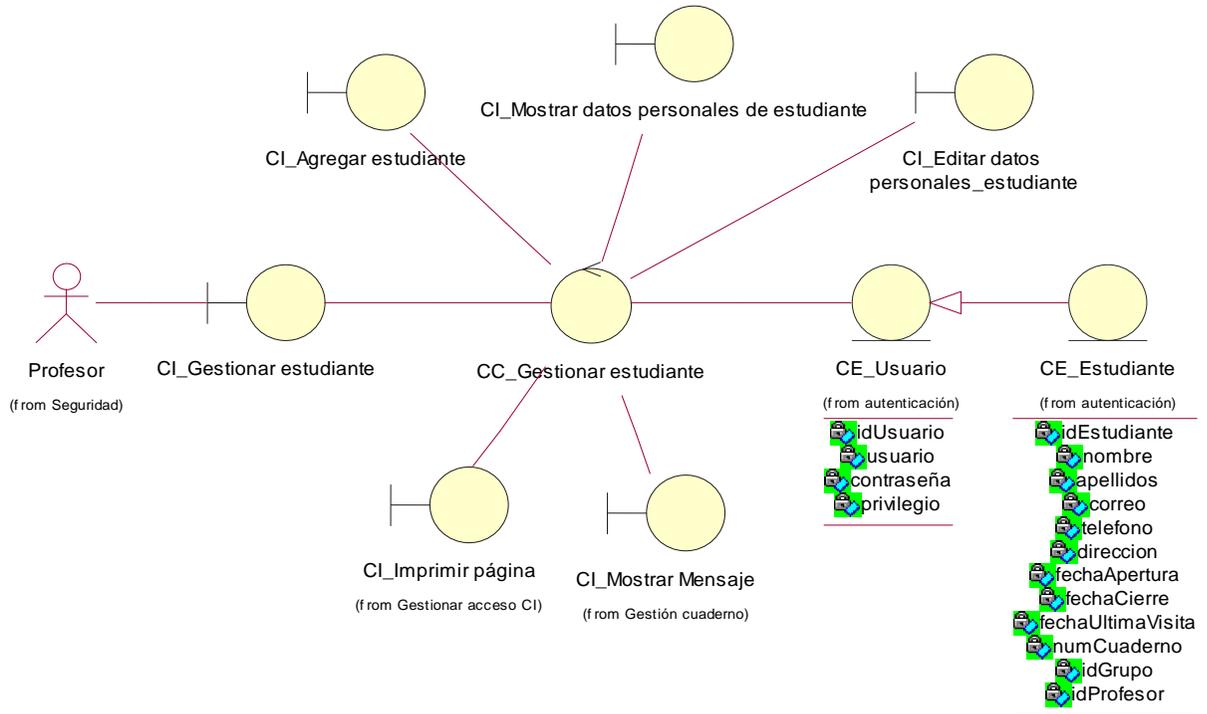


Figura 3.5: Diagrama de clases de análisis, Gestionar estudiante

DIAGRAMA DE CLASES DE ANÁLISIS, caso de uso: Gestionar semana

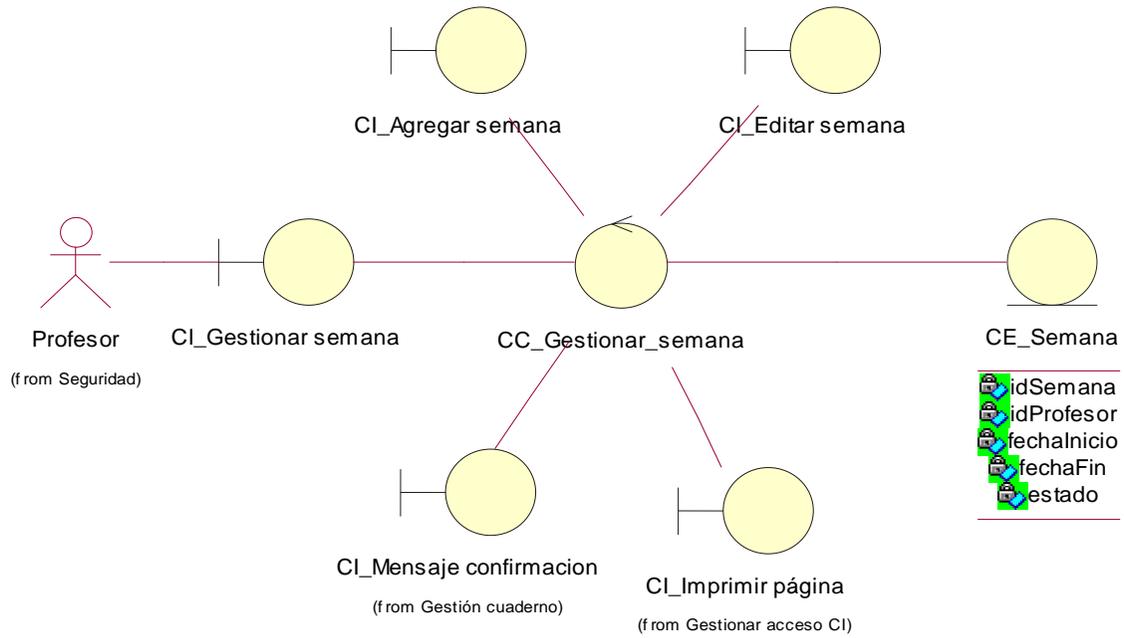


Figura 3.6: Diagrama de clases de análisis, Gestionar semana

DIAGRAMA DE CLASES DE ANÁLISIS, caso de uso: Controlar estudiante

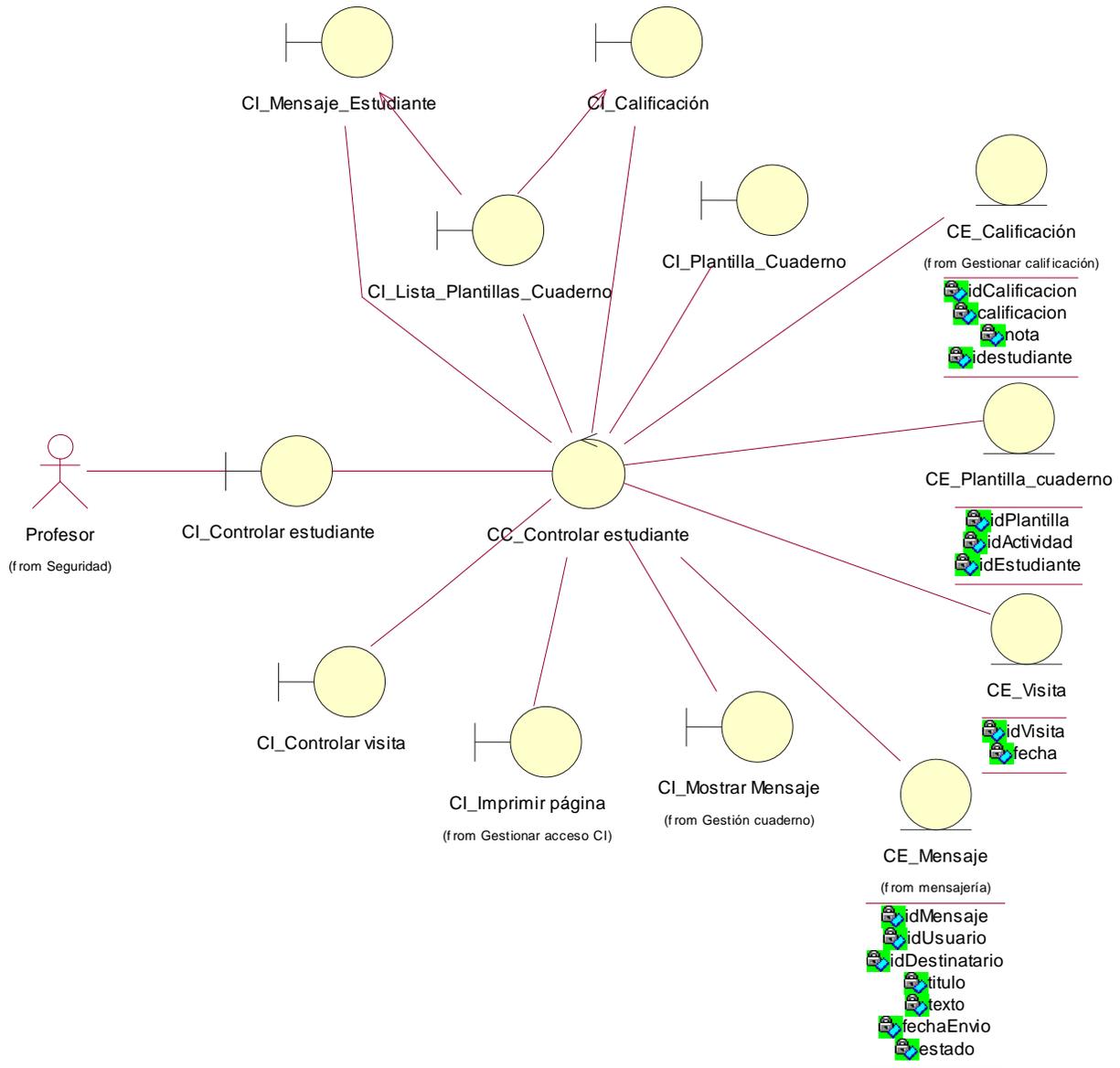
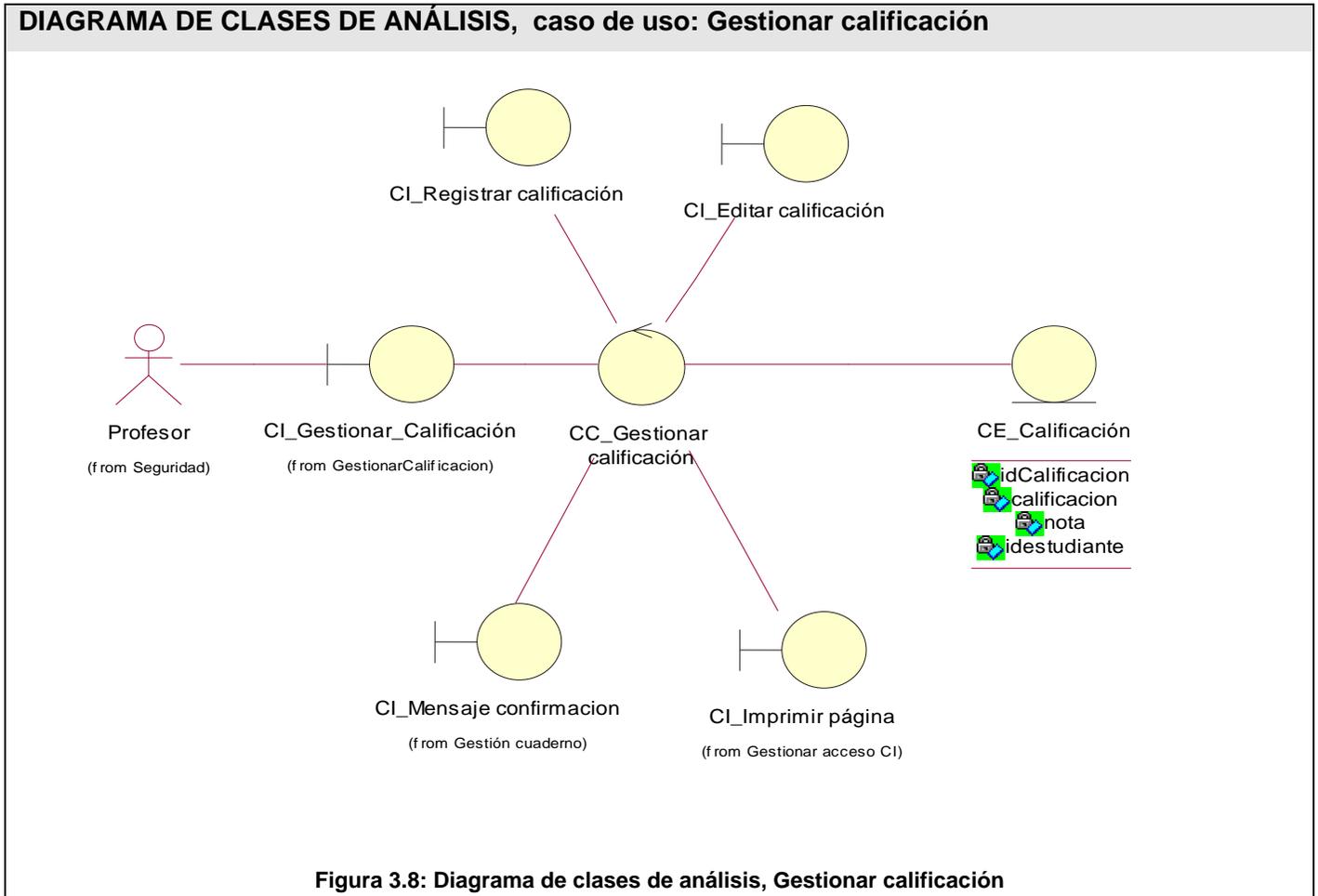


Figura 3.7: Diagrama de clases de análisis, Gestionar estudiante



3.3 Diagrama de Secuencia del diseño

Los diagramas de secuencia, muestran la ordenación temporal de los mensajes modelando la interacción entre los objetos. Los objetos suelen ser instancias con nombre o anónimas de clases, pero también pueden representarse instancias de otros elementos, tales como colaboraciones, componentes y nodos.

3.3.1 Representación de los diagramas de secuencia

En los diagramas de secuencia del diseño se muestra más detalladamente el funcionamiento del sistema internamente, ya es más vinculado al lenguaje de programación y a la arquitectura, ([Ver anexo VI](#))

3.4 Diagramas de clases del diseño

Los diagramas de clase del diseño representan a las clases de diseño y las relaciones entre ellas. Las clases diseño contienen atributos y métodos.

Cuando se trata de una aplicación web UML ofrece una serie de estereotipos para representar las partes que contiene dicha aplicación. Por ejemplo en nuestra aplicación:

- Las páginas de código HTML, son las páginas clientes, que están compuestas por formularios y contienen código JavaScript.
- Las páginas de código PHP, son las páginas servidoras.
- Las tablas de la base de datos.

En los distintos diagramas de clases del diseño se representa el patrón de arquitectura Modelo Vista Controlador (MVC).

3.4.1 Representación de los diagramas de clases del diseño

DIAGRAMA DE CLASES DE DISEÑO, caso de uso: Autenticar usuario

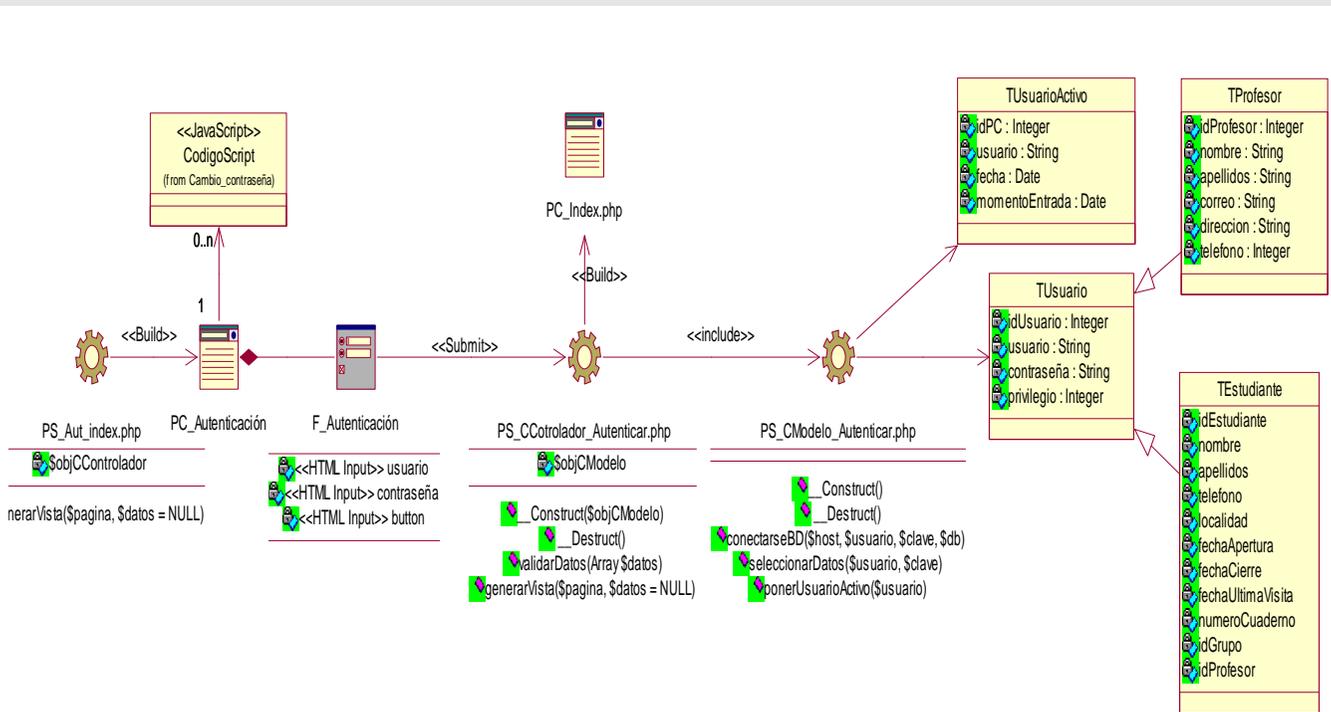


Figura 3.9: Diagrama de clases de diseño, Autenticar usuario

DIAGRAMA DE CLASES DE DISEÑO, caso de uso: Gestionar profesor

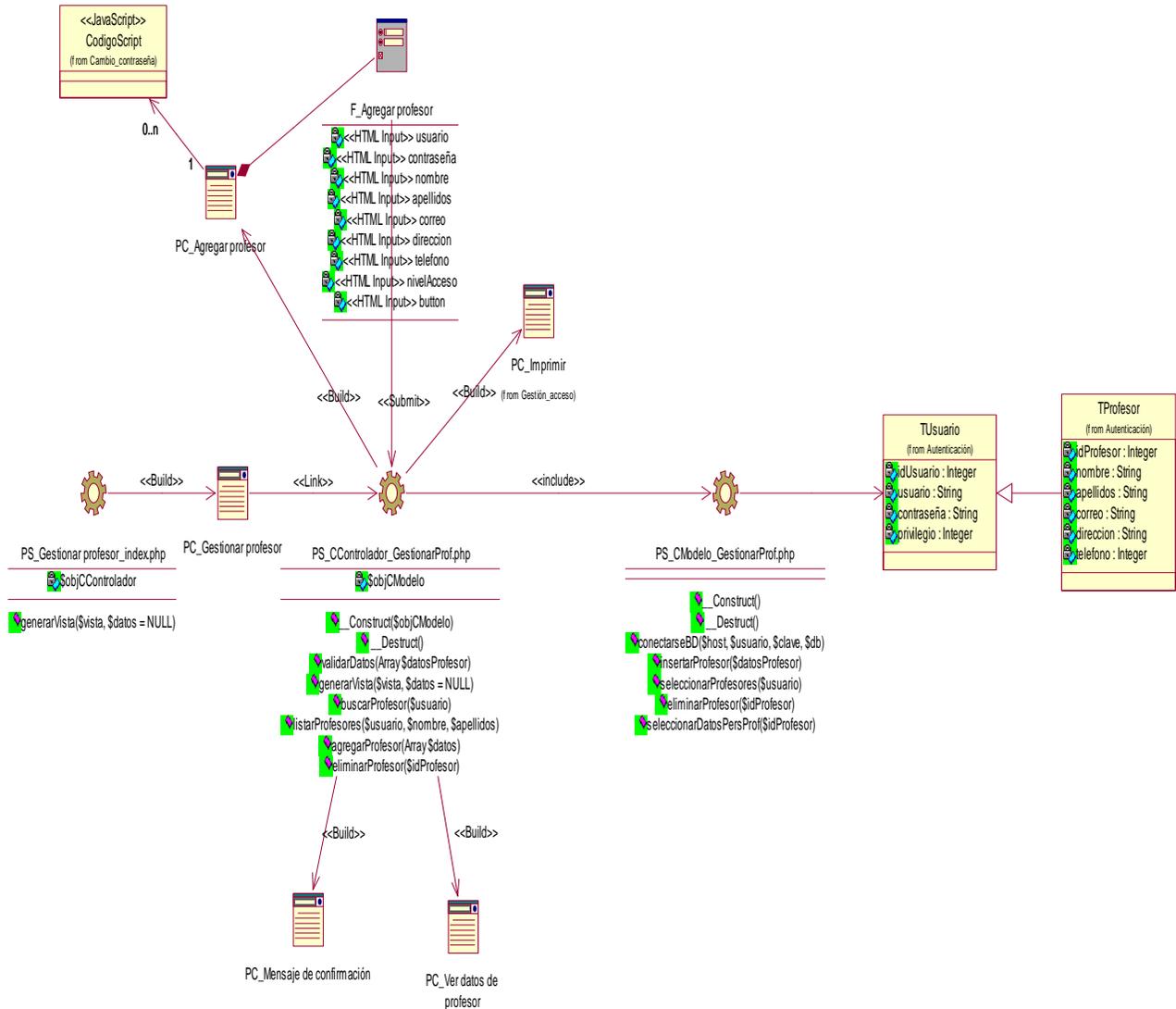


Figura 3.10: Diagrama de clases de diseño, Gestionar profesor

DIAGRAMA DE CLASES DE DISEÑO, caso de uso: Gestionar Cuaderno de Ingeniería

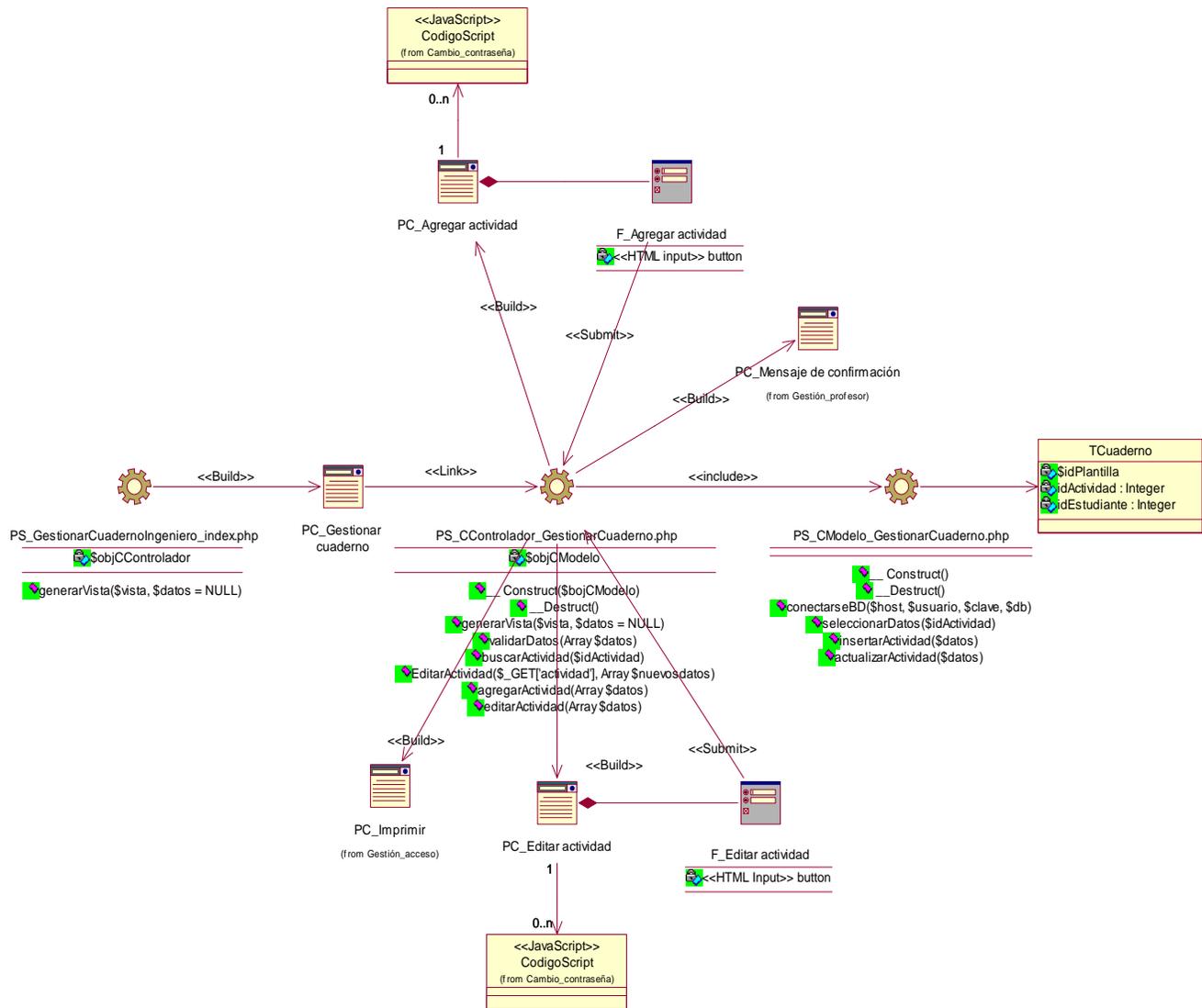


Figura 3.11: Diagrama de clases de diseño, Gestionar Cuaderno de Ingeniería

DIAGRAMA DE CLASES DE DISEÑO, caso de uso: Gestionar grupo

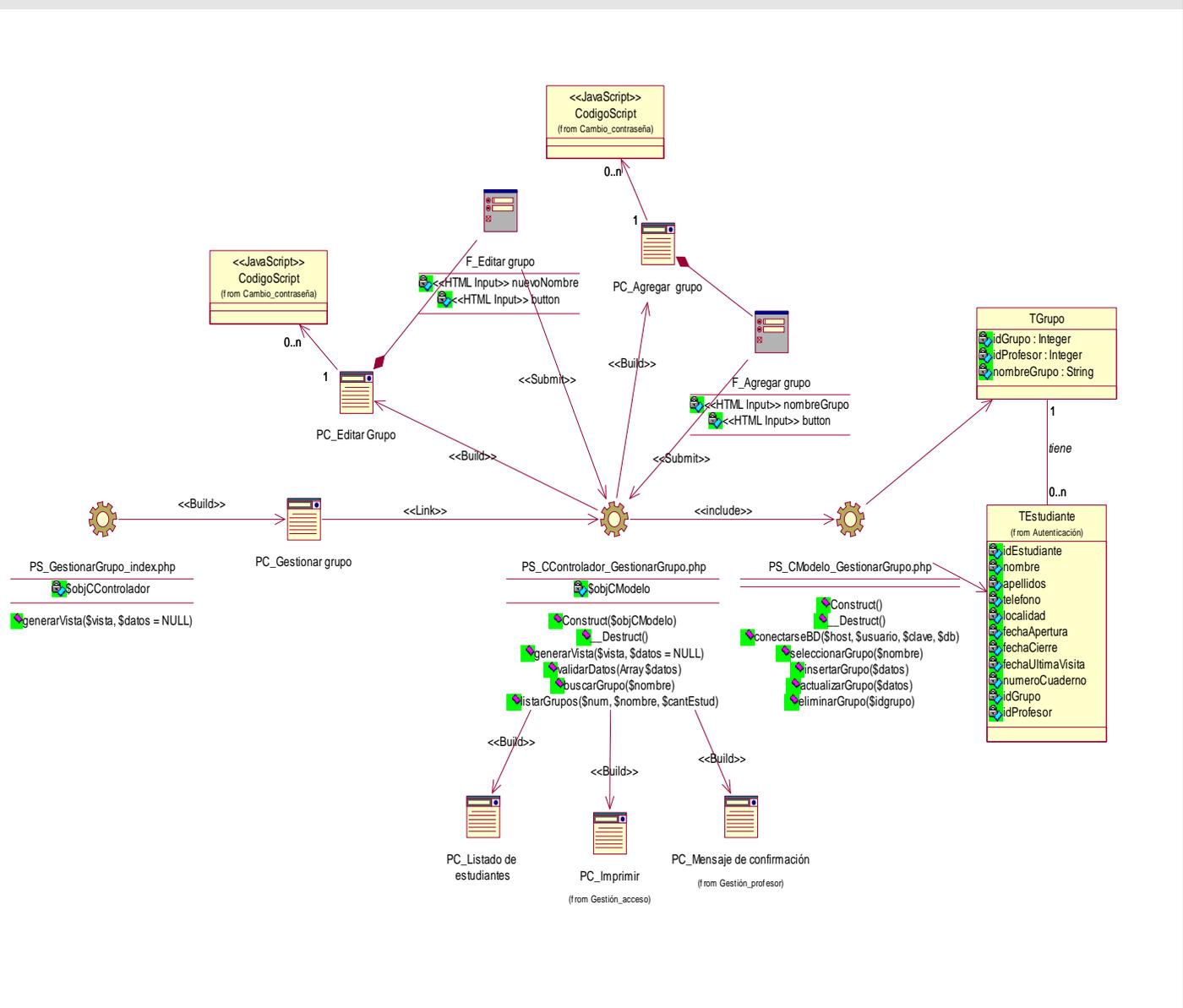


Figura 3.12: Diagrama de clases de diseño, Gestionar grupo

DIAGRAMA DE CLASES DE DISEÑO, caso de uso: Gestionar estudiante

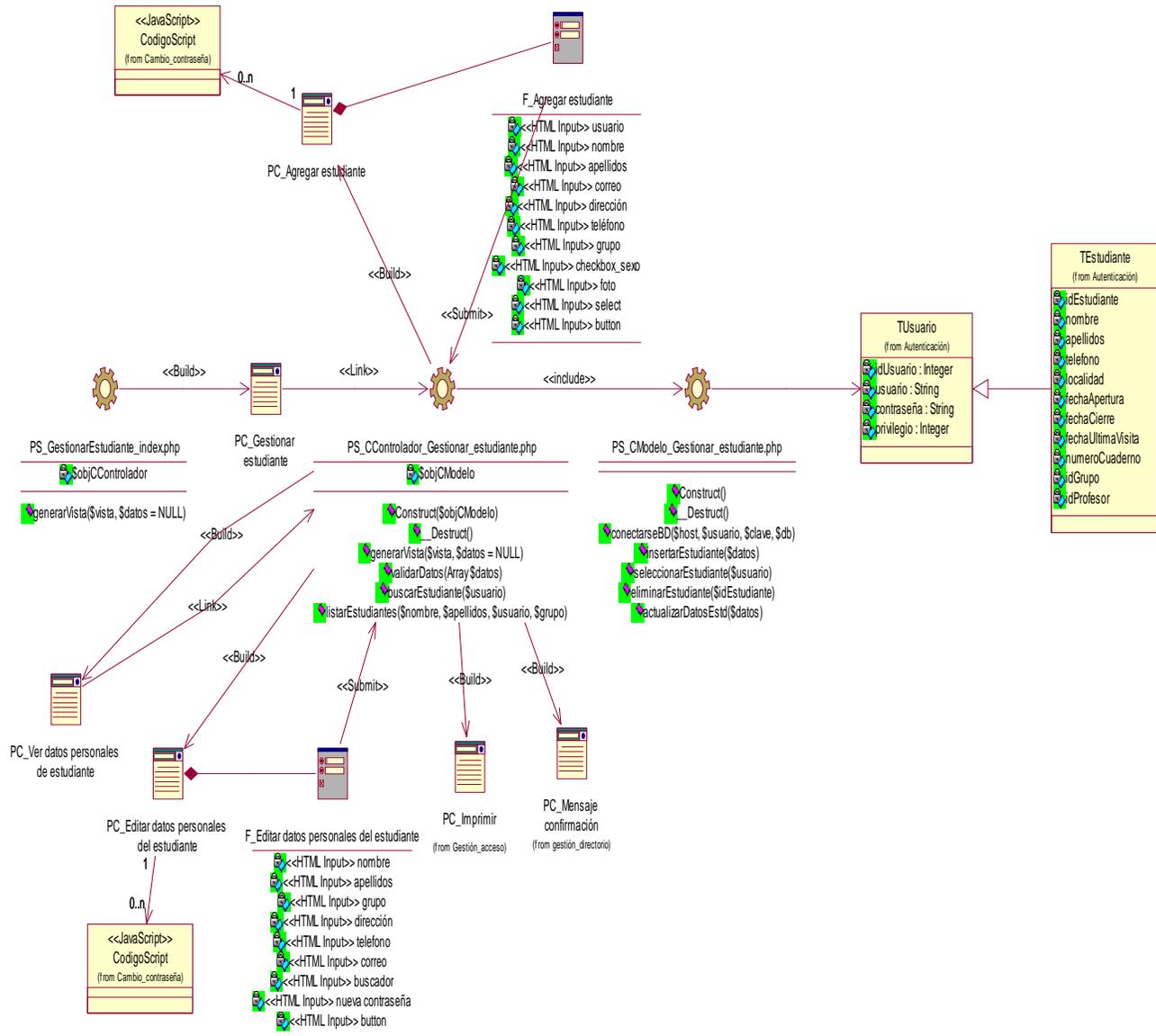


Figura 3.13: Diagrama de clases de diseño, Gestionar estudiante

DIAGRAMA DE CLASES DE DISEÑO, caso de uso: Gestionar semana

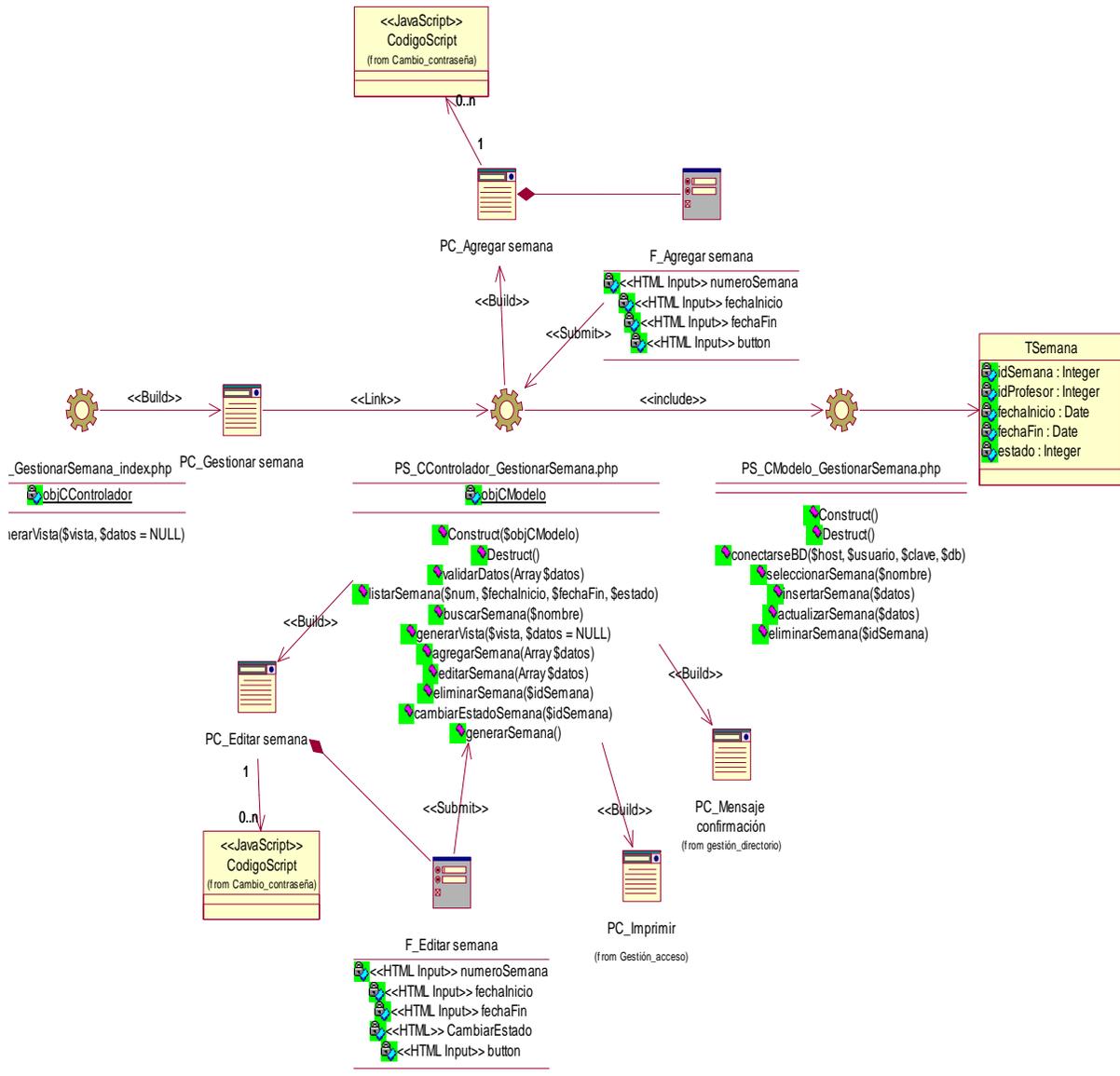


Figura 3.14: Diagrama de clases de diseño, Gestionar semana

DIAGRAMA DE CLASES DE DISEÑO, caso de uso: Controlar estudiante

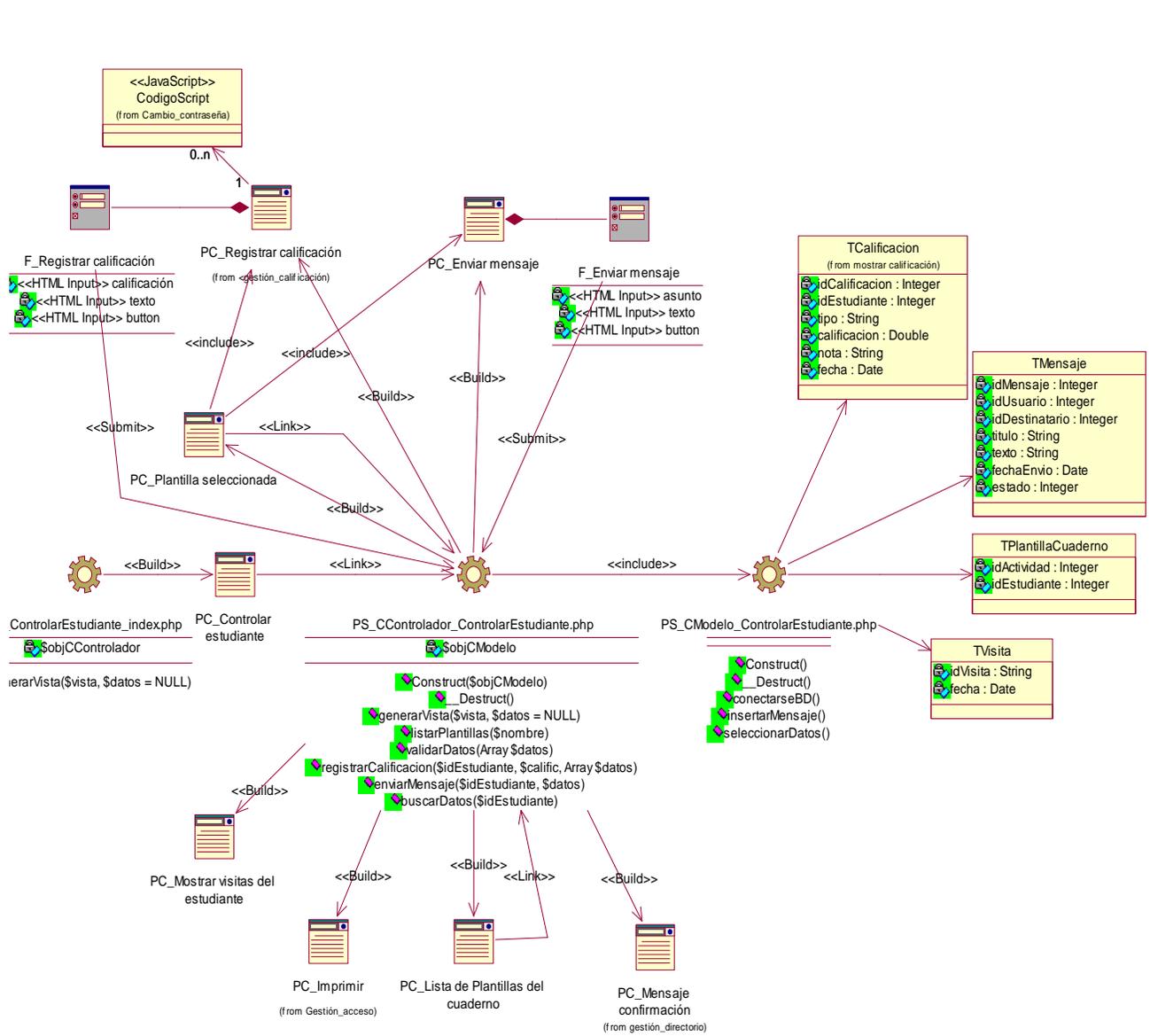


Figura 3.15: Diagrama de clases de diseño, Controlar estudiante

DIAGRAMA DE CLASES DE DISEÑO, caso de uso: Gestionar calificación

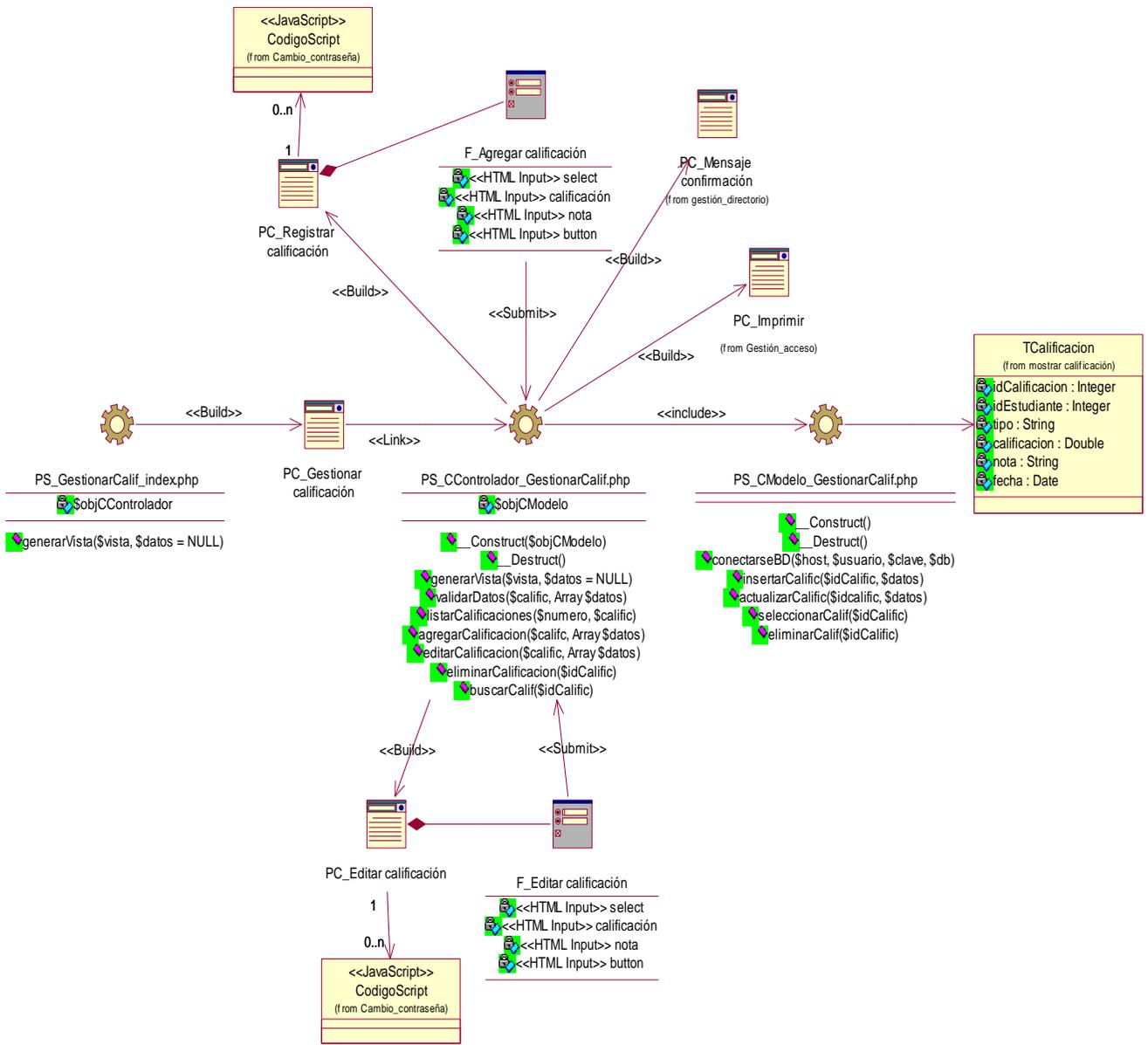


Figura 3.16: Diagrama de clases de diseño, Gestionar calificación

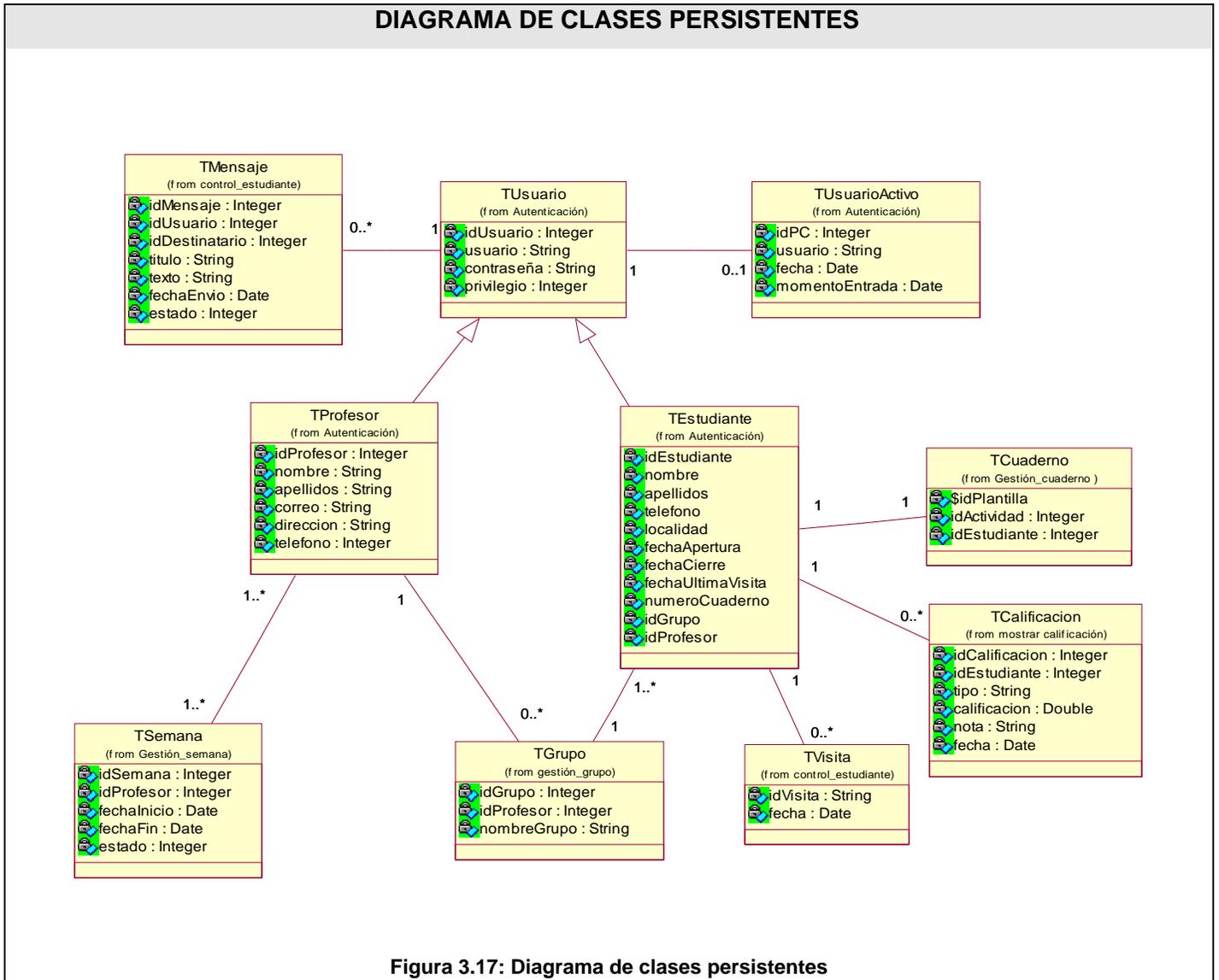
3.5 Diseño de la Base de datos

En el diseño de la base de datos, se muestran el modelo lógico y el modelo físico de la base de datos del sistema.

3.5.1 Modelo Lógico

El modelo lógico es el diagrama de clases persistentes que serán las tablas que compondrán la base de datos del sistema y que son seleccionadas en el diagrama de clases del diseño. A partir de este modelo se genera el modelo físico que es el que representa las relaciones de las tablas de la base de datos después de haber pasado por un proceso de mapeo donde las relaciones se transforman según su naturaleza y pueden o no incrementarse el número de tablas de la base de datos.

3.5.1.1 Representación del modelo lógico de datos

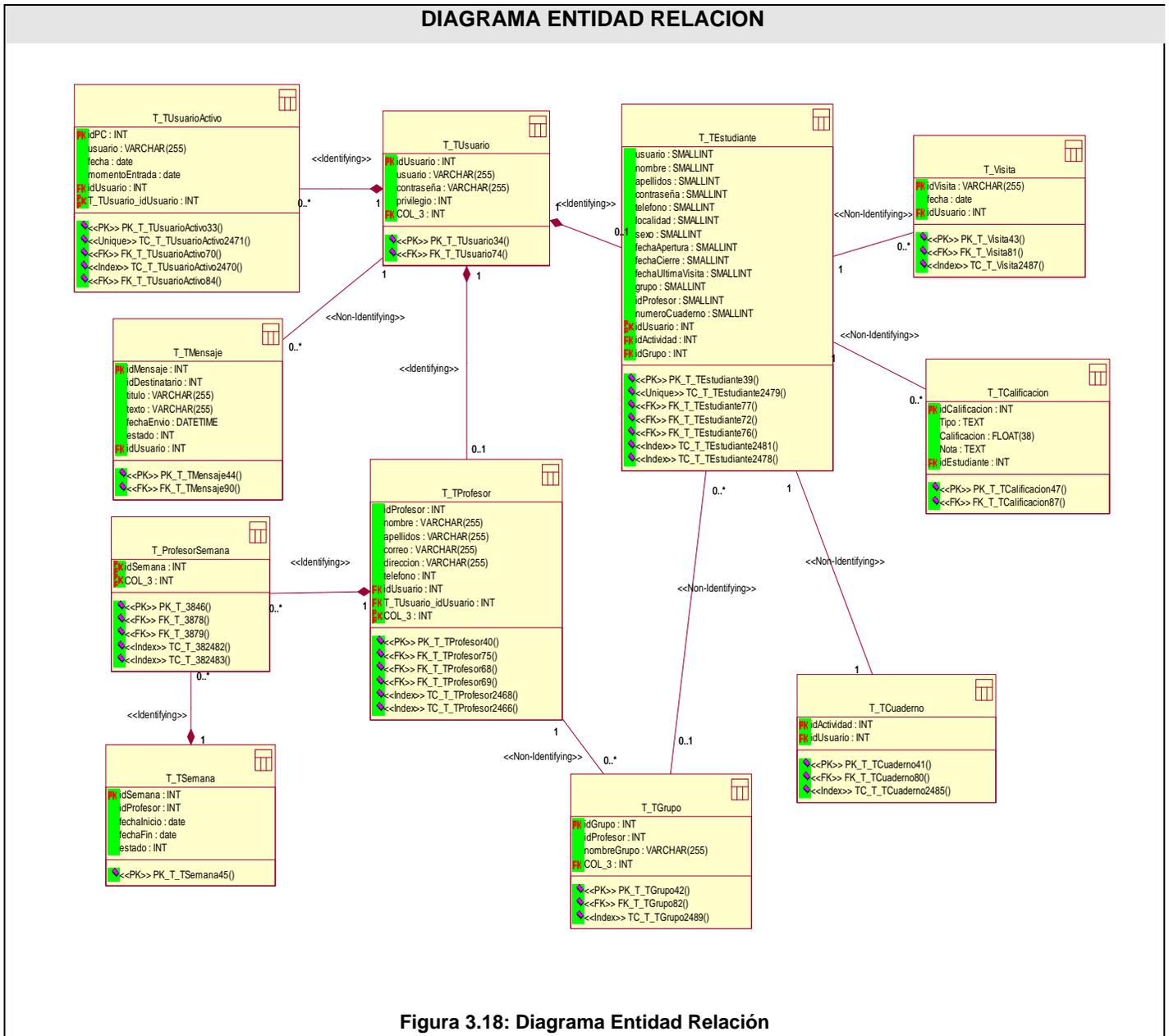


3.5.2 Modelo Físico

Son las tablas que contiene la base de datos del sistema con sus atributos y relaciones entre las tablas y es donde se selecciona el atributo identificador o llave de la tabla. Este modelo de datos se realizó a partir del diagrama de clases persistentes.

Como la tabla TCuaderno, es una de las clases persistentes, pasa a representar las plantillas del Cuaderno de Ingeniería en el modelo de datos, en ella se ponen los atributos que tienen estas plantillas en común.

3.5.2.1 Representación del modelo físico



3.6 Conclusiones

En el transcurso de este capítulo se mostraron los diferentes diagramas tanto en el análisis como en el diseño del sistema, quedando formada la arquitectura del sistema que es la arquitectura MVC la cual se manifiesta en los distintos diagramas de clases del diseño, y se muestra el modelo de datos que refleja las relaciones y las distintas tablas de la base de datos que conforman al sistema.

Del mismo modo se abordan otros aspectos como son: los principios de diseño, los tratamientos de errores aplicados al sistema, los estándares en la interfaz y la concepción general de la ayuda del sistema.

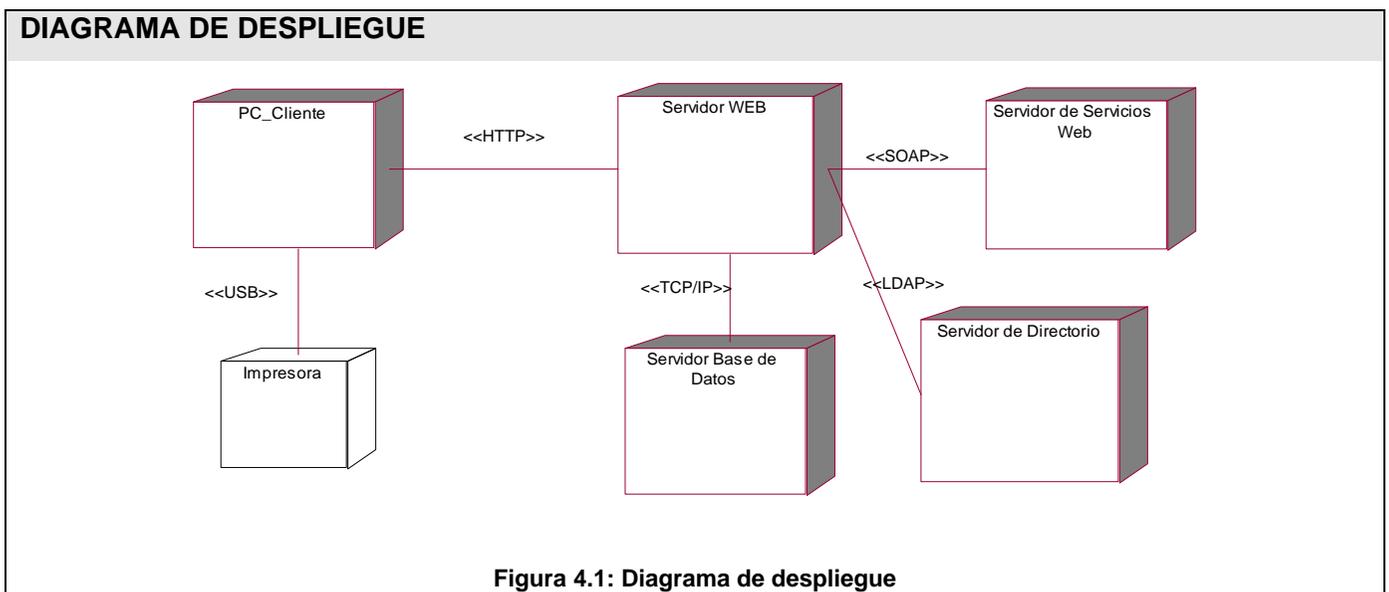
4.1 Introducción

En este capítulo se mostrarán los distintos diagramas en la implementación del sistema como son: el diagrama de despliegue quedando plasmado la estructura física del sistema, el diagrama de componentes. Luego se realizarán las descripciones de los casos de prueba de integración para verificar y garantizar el correcto funcionamiento del sistema.

4.2 Modelo de Despliegue

El modelo de despliegue, es usado para modelar la topología del hardware sobre el que se ejecuta el sistema y su distribución física, los distintos dispositivos y procesadores se representan a través de nodos. En este caso el sistema debe disponer de computadoras clientes, un servidor para aplicaciones Web y un servidor para Base de Datos y la impresora.

4.2.1 Representación del diagrama de Despliegue



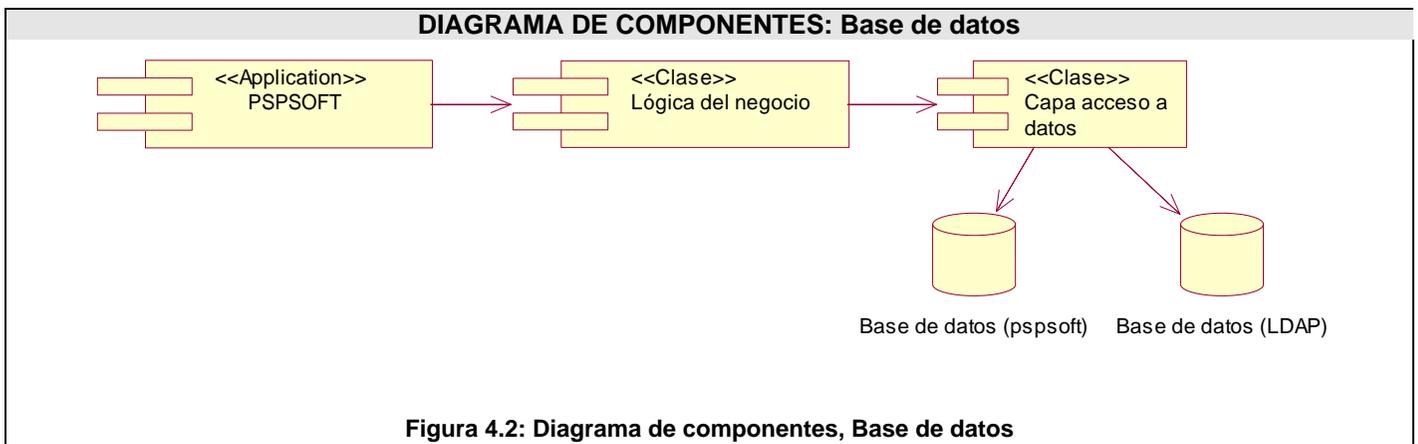
4.3. Diagrama de componentes

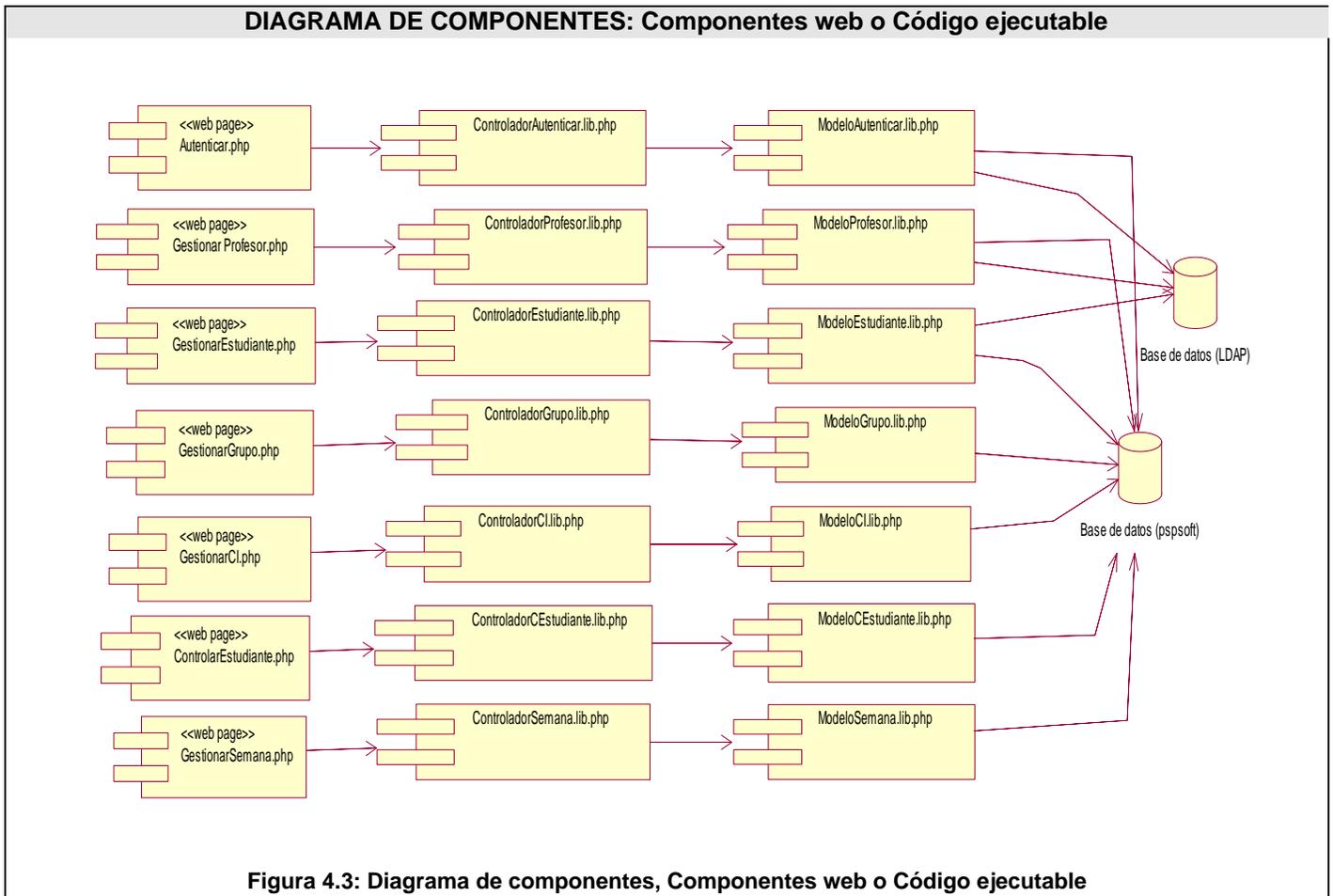
El diagrama de componentes se usa para representar a los elementos físicos o componentes y sus relaciones de dependencias. Ellos incluyen código fuente, y ejecutables. Los componentes representan todo tipo de elementos de software como por ejemplo: las librerías, archivos, bibliotecas cargadas dinámicamente.

Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

A continuación se muestran los diagramas de componente de acceso a datos y componentes web.

4.3.1 Representación del diagrama de Componentes





4.4 Casos de prueba de integración

Los casos de prueba son un conjunto de condiciones mediante las cuales se comprueba si los requisitos funcionan correctamente.

Las descripciones de los casos de prueba de integración realizadas a continuación contienen el nombre del caso de uso, el caso de prueba, se describe el estado de la información almacenada, el resultado que se obtiene y las condiciones que se deben cumplir en el momento que se ejecuta el caso de uso.

4.4.1 Descripción de los casos de prueba de integración

Caso de uso: Autenticar usuario

Caso de uso: Autenticar usuario
Caso de prueba: Autenticar usuario existente.
Entrada: El usuario desea entrar al sistema y se autentica poniendo su usuario y contraseña. El usuario está registrado.
Resultado: Se permite la entrada del usuario otorgándosele los permisos correspondientes.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el usuario.
Tabla 4.1: Descripción del caso de prueba, Autenticar usuario existente, en el CU, Autenticar usuario

Caso de uso: Autenticar usuario
Caso de prueba: Autenticar usuario no existente.
Entrada: El usuario desea entrar al sistema y se autentica poniendo su usuario y contraseña. El usuario no está registrado.
Resultado: No se permite la entrada del usuario y se muestra un mensaje confirmando que el usuario no está registrado.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el usuario.
Tabla 4.2: Descripción del caso de prueba, Autenticar usuario no existente, en el CU, Autenticar usuario

Caso de uso: Autenticar usuario
Caso de prueba: Usuario o contraseña incorrecta de usuario registrado.
Entrada: El usuario desea entrar al sistema y se autentica poniendo su usuario y contraseña. El usuario está registrado.
Resultado: No se permite la entrada del usuario y se muestra un mensaje confirmando que el usuario o la contraseña no están correctos.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el usuario.
Tabla 4.3: Descripción del caso de prueba, Autenticar usuario no existente, en el CU, Autenticar usuario

Caso de uso: Gestionar profesor

Caso de uso: Gestionar profesor
Caso de prueba: Registrar profesor no existente.
Entrada: El administrador desea registrar al profesor y entra los datos del mismo. El profesor no está registrado.
Resultado: Se registra al profesor, se muestra un mensaje confirmando la operación y se muestra la lista de profesores actualizada.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el administrador.
Tabla 4.4: Descripción del caso de prueba, Registrar profesor no existente, en el CU, Gestionar profesor

Caso de uso: Gestionar profesor
Caso de prueba: Registrar profesor existente.
Entrada: El administrador desea registrar al profesor y entra los datos del mismo. El profesor está registrado.
Resultado: No se registra al profesor, se muestra un mensaje confirmando que el profesor ya está registrado y se muestra la lista de profesores sin actualizar.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el administrador.
Tabla 4.5: Descripción del caso de prueba, Registrar profesor existente, en el CU, Gestionar profesor

Caso de uso: Gestionar profesor
Caso de prueba: Eliminar profesor existente.
Entrada: El administrador desea eliminar al profesor y selecciona la opción para eliminar. El profesor está registrado.
Resultado: Se elimina al profesor, se muestra un mensaje confirmando la operación y se muestra la lista de profesores actualizada.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el administrador.
Tabla 4.6: Descripción del caso de prueba, Eliminar profesor existente, en el CU, Gestionar profesor

Caso de uso: Gestionar profesor
Caso de prueba: Ver datos personales del profesor.

<p>Entrada:</p> <p>El administrador desea ver los datos personales del profesor y selecciona la opción de ver datos personales.</p> <p>El profesor está registrado.</p>
<p>Resultado:</p> <p>Se muestra los datos personales del profesor.</p>
<p>Condiciones:</p> <p>No se permite que se ejecuten otras instancias del caso de uso para el administrador.</p>
<p>Tabla 4.7: Descripción del caso de prueba, Ver datos personales del profesor, en el CU, Gestionar profesor</p>

<p>Caso de uso: Gestionar profesor</p>
<p>Caso de prueba: Imprimir listado de profesores.</p>
<p>Entrada:</p> <p>El administrador desea imprimir la lista de los profesores y selecciona la opción para imprimir.</p> <p>Los profesores están registrados.</p>
<p>Resultado:</p> <p>Se muestra una interfaz con la lista de los profesores y se manda a ejecutar la orden de imprimir.</p>
<p>Condiciones:</p> <p>No se permite que se ejecuten otras instancias del caso de uso para el administrador.</p>
<p>Tabla 4.8: Descripción del caso de prueba, Imprimir listado de profesores, en el CU, Gestionar profesor</p>

Caso de uso: Gestionar Cuaderno de Ingeniería

<p>Caso de uso: Gestionar Cuaderno de Ingeniería.</p>
<p>Caso de prueba: Registrar actividad no existente.</p>
<p>Entrada:</p> <p>El estudiante desea agregar una actividad en algunas de las plantillas y entra los datos correspondientes.</p> <p>La plantilla está habilitada.</p>

La actividad no está registrada.

Resultado:

Se registra la actividad, se muestra un mensaje confirmando la operación y se muestra la plantilla con las actividades actualizada.

Condiciones:

No se permite que se ejecuten otras instancias del caso de uso para el estudiante.

Tabla 4.9: Descripción del caso de prueba, Registrar actividad no existente, en el CU, Gestionar Cuaderno de Ingeniería

Caso de uso: Gestionar Cuaderno de Ingeniería.

Caso de prueba: Registrar actividad existente.

Entrada:

El estudiante desea agregar una actividad en algunas de las plantillas y entra los datos correspondientes.

La plantilla está habilitada.

La actividad está registrada.

Resultado:

No se agrega la actividad, se muestra un mensaje confirmando que la actividad ya existe y se muestra la plantilla sin cambios.

Condiciones:

No se permite que se ejecuten otras instancias del caso de uso para el estudiante.

Tabla 4.10: Descripción del caso de prueba, Registrar actividad existente, en el CU, Gestionar Cuaderno de Ingeniería

Caso de uso: Gestionar Cuaderno de Ingeniería.

Caso de prueba: Editar actividad existente.

Entrada:

El estudiante desea editar una actividad en algunas de las plantillas y entra los datos correspondientes.

<p>La plantilla está habilitada.</p> <p>La actividad está registrada en la plantilla.</p>
<p>Resultado:</p> <p>Se actualiza la actividad, se muestra un mensaje confirmando la operación y se muestra la plantilla con las actividades actualizada.</p>
<p>Condiciones:</p> <p>No se permite que se ejecuten otras instancias del caso de uso para el estudiante.</p>
<p>Tabla 4.11: Descripción del caso de prueba, Editar actividad existente, en el CU, Gestionar Cuaderno de Ingeniería</p>

<p>Caso de uso: Gestionar Cuaderno de Ingeniería.</p>
<p>Caso de prueba: Imprimir datos.</p>
<p>Entrada:</p> <p>El estudiante desea imprimir alguna de las plantillas con las actividades registradas.</p> <p>La plantilla está habilitada.</p> <p>Hay actividades registradas en la plantilla.</p>
<p>Resultado:</p> <p>Se muestra un interfaz con la plantilla seleccionada y sus actividades y manda a ejecutar la orden de imprimir.</p>
<p>Condiciones:</p> <p>No se permite que se ejecuten otras instancias del caso de uso para el estudiante.</p>
<p>Tabla 4.12: Descripción del caso de prueba, Imprimir datos, en el CU, Gestionar Cuaderno de Ingeniería</p>

Caso de uso: Gestionar Grupo

<p>Caso de uso: Gestionar grupo</p>
<p>Caso de prueba: Agregar grupo no existente.</p>
<p>Entrada:</p> <p>El profesor desea agregar un nuevo grupo y entra los datos correspondientes.</p>

El grupo no está registrado.
Resultado: Se registra al grupo, se muestra un mensaje confirmando la operación y se muestra la lista de grupos actualizada.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.13: Descripción del caso de prueba, Agregar grupo no existente, en el CU, Gestionar grupo

Caso de uso: Gestionar grupo
Caso de prueba: Agregar grupo existente.
Entrada: El profesor desea agregar un nuevo grupo y entra los datos correspondientes. El grupo está registrado.
Resultado: No se registra el grupo, se muestra un mensaje confirmando que el grupo ya existe y se muestra la lista de grupos sin actualizar.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.14: Descripción del caso de prueba, Agregar grupo existente, en el CU, Gestionar grupo

Caso de uso: Gestionar grupo
Caso de prueba: Editar grupo existente.
Entrada: El profesor desea editar un grupo y entra los datos correspondientes. El grupo está registrado.
Resultado: Se actualiza el grupo, se muestra un mensaje confirmando la operación y se muestra la lista de grupos

actualizada.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.15: Descripción del caso de prueba, Editar grupo existente, en el CU, Gestionar grupo

Caso de uso: Gestionar grupo
Caso de prueba: Eliminar grupo existente.
Entrada: El profesor desea eliminar un grupo y selecciona la opción para eliminar. El grupo está registrado.
Resultado: Se elimina el grupo, se muestra un mensaje confirmando la operación y se muestra la lista de grupos actualizada.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.16 Descripción del caso de prueba, Eliminar grupo existente, en el CU, Gestionar grupo

Caso de uso: Gestionar grupo
Caso de prueba: Mostrar listado de estudiantes registrados.
Entrada: El profesor desea ver el listado de sus estudiantes y selecciona la opción para ver el listado. El grupo está registrado. Los estudiantes están registrados.
Resultado: Se muestra la lista de estudiantes.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.17: Descripción del caso de prueba, Mostrar listado de estudiantes registrados, en el CU, Gestionar grupo

Caso de uso: Gestionar grupo
Caso de prueba: Mostrar listado de estudiantes no registrados.
Entrada: El profesor desea ver el listado de sus estudiantes y selecciona la opción para ver el listado. EL grupo está registrado. No hay estudiantes registrados.
Resultado: Se muestra un mensaje confirmando que no hay estudiantes registrados.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.18: Descripción del caso de prueba, Mostrar listado de estudiante no registrados, en el CU, Gestionar grupo

Caso de uso: Gestionar grupo.
Caso de prueba: Imprimir lista de grupos.
Entrada: El profesor desea imprimir la lista de grupos. Hay grupos registrados.
Resultado: Se muestra una interfaz con la lista de grupos y se manda a ejecutar la orden de imprimir.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.19: Descripción del caso de prueba, Imprimir lista de grupos, en el CU, Gestionar grupo

Caso de uso: Gestionar estudiante

Caso de uso: Gestionar estudiante.
Caso de prueba: Registrar estudiante no existente.
Entrada: El profesor desea agregar un nuevo estudiante.

<p>EL grupo está registrado. El estudiante no está registrado.</p>
<p>Resultado: Se registra el estudiante, muestra un mensaje confirmando la operación y muestra la lista de estudiantes actualizada.</p>
<p>Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.</p>
<p>Tabla 4.20: Descripción del caso de prueba, Registrar estudiante no existente, en el CU, Gestionar estudiante</p>

<p>Caso de uso: Gestionar estudiante.</p>
<p>Caso de prueba: Registrar estudiante existente.</p>
<p>Entrada: El profesor desea agregar un nuevo estudiante. El grupo está registrado. El estudiante está registrado.</p>
<p>Resultado: No se registra el estudiante, muestra un mensaje confirmando que el estudiante ya existe y muestra la lista de estudiantes sin actualizar.</p>
<p>Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.</p>
<p>Tabla 4.21: Descripción del caso de prueba, Registrar estudiante existente, en el CU, Gestionar estudiante</p>

<p>Caso de uso: Gestionar estudiante.</p>
<p>Caso de prueba: Eliminar estudiante existente.</p>
<p>Entrada: El profesor desea eliminar a un estudiante y selecciona la opción para eliminar. El grupo está registrado. El estudiante está registrado.</p>

<p>Resultado:</p> <p>Se elimina el estudiante, muestra un mensaje confirmando la operación y muestra la lista de estudiantes actualizada.</p>
<p>Condiciones:</p> <p>No se permite que se ejecuten otras instancias del caso de uso para el profesor.</p>
<p>Tabla 4.22: Descripción del caso de prueba, Eliminar estudiante existente, en el CU, Gestionar estudiante</p>

<p>Caso de uso: Gestionar profesor</p>
<p>Caso de prueba: Ver datos personales del estudiante.</p>
<p>Entrada:</p> <p>El profesor desea ver los datos personales del estudiante y selecciona la opción de ver datos personales.</p> <p>El estudiante está registrado.</p>
<p>Resultado:</p> <p>Se muestra los datos personales del estudiante.</p>
<p>Condiciones:</p> <p>No se permite que se ejecuten otras instancias del caso de uso para el profesor.</p>
<p>Tabla 4.23: Descripción del caso de prueba, Ver datos personales del profesor, en el CU, Gestionar profesor</p>

<p>Caso de uso: Gestionar estudiante.</p>
<p>Caso de prueba: Editar los datos personales del estudiante.</p>
<p>Entrada:</p> <p>El profesor desea editar los datos personales del estudiante.</p> <p>El grupo está registrado.</p> <p>El estudiante está registrado.</p>
<p>Resultado:</p> <p>Se actualiza los datos personales del estudiante, muestra un mensaje confirmando la operación y muestra los datos personales del estudiante actualizados.</p>

Condiciones:
No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.24: Descripción del caso de prueba, Editar los datos personales del estudiante, en el CU, Gestionar estudiante

Caso de uso: Gestionar estudiante.
Caso de prueba: Imprimir lista de estudiante.
Entrada:
El profesor desea imprimir la lista de estudiantes.
El grupo está registrado.
Hay estudiantes registrados.
Resultado:
Se muestra la interfaz con la lista de estudiantes y se manda a ejecutar la orden de imprimir.
Condiciones:
No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.25: Descripción del caso de prueba, Imprimir lista de estudiantes, en el CU, Gestionar estudiante

Caso de uso: Gestionar Semana

Caso de uso: Gestionar semana.
Caso de prueba: Agregar semana no existente.
Entrada:
El profesor desea agregar una nueva semana.
La semana no está registrada.
Resultado:
Se registra la semana, se muestra un mensaje confirmando la operación y se muestra la lista de semanas actualizada.
Condiciones:
No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.26: Descripción del caso de prueba, Agregar semana no existente, en el CU, Gestionar semana

Caso de uso: Gestionar semana.
Caso de prueba: Agregar semana existente.
Entrada: El profesor desea agregar una nueva semana. La semana está registrada.
Resultado: No se registra la semana, se muestra un mensaje confirmando que la semana ya existe y se muestra la lista de semanas sin actualizar.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.27: Descripción del caso de prueba, Agregar semana existente, en el CU, Gestionar semana

Caso de uso: Gestionar semana.
Caso de prueba: Editar semana existente.
Entrada: El profesor desea editar alguna semana. La semana está registrada.
Resultado: Se actualiza la semana, se muestra un mensaje confirmando la operación y se muestra la lista de semanas actualizada.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.28: Descripción del caso de prueba, Editar semana existente, en el CU, Gestionar semana

Caso de uso: Gestionar semana.
Caso de prueba: Eliminar semana existente.
Entrada: El profesor desea eliminar alguna semana y selecciona la opción para eliminar.

La semana está registrada.
Resultado: Se elimina la semana, se muestra un mensaje confirmando la operación y se muestra la lista de semanas actualizada.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.29: Descripción del caso de prueba, Eliminar semana existente, en el CU, Gestionar semana

Caso de uso: Gestionar semana.
Caso de prueba: Generar semana.
Entrada: El profesor desea generar una semana y selecciona la opción para generar semana. La semana no está registrada.
Resultado: Se genera la semana, se muestra un mensaje confirmando la operación y se muestra la lista de semanas actualizada.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.30: Descripción del caso de prueba, Generar semana, en el CU, Gestionar semana

Caso de uso: Gestionar semana.
Caso de prueba: Cambiar estado de la semana.
Entrada: El profesor desea habilitar o inhabilitar una semana y selecciona la opción para cambiar estado. La semana está registrada.
Resultado: Se cambia el estado de la semana, se muestra un mensaje confirmando la operación y se muestra la lista

de semana, mostrando el estado actualizado.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.31: Descripción del caso de prueba, Cambiar estado de la semana, en el CU, Gestionar semana

Caso de uso: Gestionar semana.
Caso de prueba: Imprimir lista de semanas.
Entrada: El profesor desea imprimir la lista de semanas y selecciona la opción para imprimir. Hay semanas registradas.
Resultado: Se muestra la interfaz con la lista de las semanas y se ejecuta la orden de imprimir.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.32: Descripción del caso de prueba, Imprimir lista de semanas, en el CU, Gestionar semana

Caso de uso: Controlar estudiante

Caso de uso: Controlar estudiante.
Caso de prueba: Revisar plantilla del cuaderno con actividades registradas.
Entrada: El profesor desea revisar las plantillas del cuaderno y selecciona la plantilla para revisar las actividades que tiene registradas. El estudiante está registrado. Las actividades están registradas en la plantilla.
Resultado: Se muestra la plantilla con las actividades registradas en ella.
Condiciones:

No se permite que se ejecuten otras instancias del caso de uso para el profesor.

Tabla 4.33: Descripción del caso de prueba, Revisar plantilla del cuaderno con actividades registradas, en el CU, Controlar estudiante

Caso de uso: Controlar estudiante.

Caso de prueba: Revisar plantilla del cuaderno sin actividades registradas.

Entrada:

El profesor desea revisar las plantillas del cuaderno y selecciona la plantilla para revisar las actividades que tiene registradas.

El estudiante está registrado.

Las actividades no están registradas en la plantilla.

Resultado:

Se muestra la un mensaje confirmando que no hay actividades registradas.

Condiciones:

No se permite que se ejecuten otras instancias del caso de uso para el profesor.

Tabla 4.34: Descripción del caso de prueba, Revisar plantilla del cuaderno sin actividades registradas, en el CU, Controlar estudiante

Caso de uso: Controlar estudiante.

Caso de prueba: Enviar un mensaje al estudiante.

Entrada:

El profesor desea enviar un mensaje al estudiante y selecciona la opción para enviar dicho mensaje.

Hay estudiantes registrados.

Resultado:

Se envía el mensaje al estudiante, se muestra un mensaje de confirmación y se muestra la lista de mensajes recibidos.

Condiciones:

No se permite que se ejecuten otras instancias del caso de uso para el profesor.

Tabla 4.35: Descripción del caso de prueba, Agregar semana no existente, en el CU, Controlar estudiante

Caso de uso: Controlar estudiante.

Caso de prueba: Controlar visitas realizadas por el estudiante.

Entrada:

El profesor desea ver la cantidad de visitas que ha realizado el estudiante al cuaderno y selecciona la opción para ver las visitas del estudiante.

El estudiante está registrado.

Hay visitas registradas.

Resultado:

Se muestra la cantidad de visitas que el estudiante ha realizado.

Condiciones:

No se permite que se ejecuten otras instancias del caso de uso para el profesor.

Tabla 4.36: Descripción del caso de prueba, Controlar visitas realizadas por el estudiante, en el CU, Controlar estudiante

Caso de uso: Controlar estudiante.

Caso de prueba: Controlar visitas realizadas por el estudiante.

Entrada:

El profesor desea ver la cantidad de visitas que ha realizado el estudiante al cuaderno y selecciona la opción para ver las visitas del estudiante.

El estudiante está registrado.

No hay visitas registradas.

Resultado:

Se muestra un mensaje confirmando que el estudiante no ha realizado ninguna visita.

Condiciones:

No se permite que se ejecuten otras instancias del caso de uso para el profesor.

Tabla 4.37: Descripción del caso de prueba, Controlar visitas realizadas por el estudiante, en el CU, Controlar estudiante

Caso de uso: Controlar estudiante.
Caso de prueba: Imprimir datos.
<p>Entrada:</p> <p>El profesor desea imprimir alguna de las platillas del cuaderno o las visitas que ha realizado el estudiante al cuaderno.</p> <p>El estudiante está registrado.</p> <p>Los datos están registrados.</p>
<p>Resultado:</p> <p>Se muestra la interfaz con las actividades registradas en la plantilla seleccionada o la cantidad de visitas realizadas por el estudiante y se ejecuta la orden de imprimir.</p>
<p>Condiciones:</p> <p>No se permite que se ejecuten otras instancias del caso de uso para el profesor.</p>
Tabla 4.38: Descripción del caso de prueba, Imprimir datos, en el CU, Controlar estudiante

Caso de uso: Gestionar calificación

Caso de uso: Registrar calificación.
Caso de prueba: Agregar calificación.
<p>Entrada:</p> <p>El profesor desea registrar una calificación al estudiante y selecciona la opción para registrar la calificación.</p> <p>El estudiante está registrado.</p>
<p>Resultado:</p> <p>Se agrega la calificación, se muestra un mensaje de confirmación y se muestra la lista de calificaciones.</p>
<p>Condiciones:</p> <p>No se permite que se ejecuten otras instancias del caso de uso para el profesor.</p>
Tabla 4.39: Descripción del caso de prueba, Agregar calificación, en el CU, Controlar estudiante

Caso de uso: Registrar calificación.
Caso de prueba: Editar calificación existente.
Entrada: El profesor desea editar una calificación al estudiante y selecciona la opción para editar la calificación. El estudiante está registrado. La calificación está registrada.
Resultado: Se actualiza la calificación, se muestra un mensaje de confirmación y se muestra la lista de calificaciones.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.40: Descripción del caso de prueba, Editar calificación existente, en el CU, Controlar estudiante

Caso de uso: Registrar calificación.
Caso de prueba: Eliminar calificación existente.
Entrada: El profesor desea eliminar una calificación al estudiante y selecciona la opción para eliminar la calificación. El estudiante está registrado. La calificación está registrada.
Resultado: Se elimina la calificación, se muestra un mensaje de confirmación de la operación y se muestra la lista de calificaciones.
Condiciones: No se permite que se ejecuten otras instancias del caso de uso para el profesor.
Tabla 4.41: Descripción del caso de prueba, Eliminar calificación existente, en el CU, Controlar estudiante

Caso de uso: Registrar calificación.
Caso de prueba: Imprimir lista de calificaciones.

<p>Entrada:</p> <p>El profesor desea imprimir la lista de calificaciones del estudiante y selecciona la opción para imprimir. El estudiante está registrado. Hay calificaciones registradas.</p>
<p>Resultado:</p> <p>Se muestra la interfaz con la lista de calificaciones registradas, se muestra un mensaje de confirmación de la operación y se ejecuta la orden de imprimir.</p>
<p>Condiciones:</p> <p>No se permite que se ejecuten otras instancias del caso de uso para el profesor.</p>
<p>Tabla 4.42: Descripción del caso de prueba, Imprimir lista de calificaciones, en el CU, Controlar estudiante</p>

4.5 Conclusiones

En este capítulo se mostró el diagrama de despliegue, los distintos diagramas de componentes y se realizaron las descripciones de los casos de prueba de integración, quedando plasmado el cumplimiento de la implementación y la prueba del sistema.

CAPÍTULO **5**

Estudio de factibilidad

5.1 Introducción

La estimación de un proyecto es muy importante para evitar la pérdida de esfuerzo, tiempo y dinero. Su propósito es establecer planes razonables para desarrollar la Ingeniería de Software y manejar los cambios de los proyectos de Software.

Para realizar una buena estimación se debe tener en cuenta la actividad de estimar los resultados del proyecto y los valores de costo, tiempo y recursos requeridos. Este capítulo contiene el método de estimación Cocomo II para la planificación del sistema, obteniendo entre otras cosas, el esfuerzo y costo del proyecto. Comprende además el análisis del beneficio tangible e intangible y el análisis del costo teniendo en cuenta los beneficios para así llegar a un resultado sobre su factibilidad.

5.2 Planificación

Entrada Externa

Nombre de la Entrada Externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (Simple, Media y Compleja)
Agregar profesor	3	8	Medio
Agregar actividad en Cuaderno de Ingeniería	3	30	Alto
Editar actividad en el Cuaderno de Ingeniería	3	30	Alto
Agregar Grupo	3	2	Bajo
Editar Grupo	3	1	Bajo
Agregar estudiante	3	8	Alto
Editar Datos estudiante	3	6	Medio

Agregar Semana	3	5	Medio
Editar semana	3	4	Medio
Registrar calificación	3	3	Bajo
Enviar mensaje	3	2	Bajo
Editar calificación	3	2	Bajo

Salida Externa

Nombre de la Salida Externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (Simple, Media y Compleja)
Listado de profesores	3	5	Medio
Listado de actividades del cuaderno	3	30	Alto
Listado de estudiantes	3	5	Medio
Listado de las plantillas del Cuaderno de Ingeniería	3	15	Medio
Visitas al Cuaderno de Ingeniería	3	3	Bajo
Listado de actividades de plantilla	3	31	Alto
Listado de calificaciones	3	4	Medio
Listado de las semanas	3	4	Medio
Listado de grupos	3	3	Bajo

Petición

Nombre de la Petición	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (Simple, Media y Compleja)
Obtener_Profesor	3	5	Bajo
Agregar_Profesor	3	8	Medio
Eliminar_Profesor	3	8	Medio
Obtener_Actividad_Cuaderno	3	30	Bajo
Agregar_Actividad	3	31	Bajo
Actualizar_Actividad	3	30	Medio
Obtener_Estudiante	3	5	Medio
Agregar_Estudiante	3	8	Medio
Actualizar_Estudiante	3	7	Medio
Eliminar_Estudiante	3	8	Medio
Obtener_Plantilla_Cuaderno	3	15	Medio
Obtener_Calificacion	3	4	Bajo
Agregar_Calificacion	3	4	Bajo
Actualizar_Calificacion	3	3	Medio
Eliminar_Calificacion	3	4	Medio
Obtener_Semana	3	4	Bajo
Agregar_Semana	3	5	Bajo
Actualizar_Semana	3	4	Medio
Eliminar_Semana	3	5	Medio
Obtener_Grupo	3	3	Bajo
Agregar_Grupo	3	2	Bajo
Actualizar_Grupo	3	1	Bajo
Eliminar_Grupo	3	2	Bajo

Fichero Interno

Nombre del Fichero Interno	Cantidad de records	Cantidad de elementos de datos	Clasificación (Simple, Media y Compleja)
Usuario	2	3	Bajo
Profesor	4	8	Medio
Cuaderno de Ingeniería	4	31	Alto
Grupo	6	8	Medio
Estudiante	5	9	Medio
Semana	6	5	Bajo
Mensajes	2	4	Medio
Calificación	4	4	Medio

Interfaz Externa

Nombre de la Interfaz Externa	Cantidad de records	Cantidad de elementos de datos	Clasificación (Simple, Media y Compleja)
LDAP	6	6	Medio

5.2.2 Puntos de función desajustados

Elementos	Simple	X Peso	Medios	X Peso	Complejos	X Peso	Subtotal de puntos de función
Ficheros Lógicos Internos	4	28	3	30	1	15	73
Ficheros de Interfaces	0	0	1	6	0	0	6

Externas							
Entradas Externas	6	30	3	30	3	45	105
Salidas Externas	3	30	5	50	2	30	110
Peticiones	11	55	12	120	0	0	175
Total							469

5.3 Cálculo de las instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

Características	Valor
Puntos de función desajustados	469
Lenguaje	PHP
Instrucciones fuentes por puntos de función	36
Instrucciones fuentes	16.8 KSLOC

5.3.1 Cálculos

Cálculo de:	Valor
Esfuerzo	22 Hombres/Mes
Tiempo de desarrollo	9.5 Meses
Cantidad de hombres	2 Hombres
Salario Medio	\$ 50
Costo	\$ 2 200

$$EM = \prod_{i=1}^7 E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED = 0.495$$

$$SF = \sum_{i=1}^5 SFi = PREC + FLEX + RESL + TEAM + PMAT = 16.55$$

$$A = 2.94; B = 0.91; C = 3.67; D = 0.28$$

$$E = B + 0.01 * SF = 0.91 + 0.01 * 16.55 = 1.0755$$

Esfuerzo

$$PM = A * (Size)^E * EM = 2.94 * (16.8)^{1.07} * 0.495 = 21.9 \approx 22 \text{ Hombres / Mes}$$

Cálculo del tiempo de desarrollo

$$F = D + 0.2 * 0.01 * \sum_{i=1}^5 SFi = 0.28 + 0.2 * 0.01 * 16.55 = 0.313$$

$$TDEV = C * PM^F = 3.67 * (22)^{0.31} = 9.5 \text{ Meses}$$

Cálculo de la cantidad de hombres

$$CH = PM / TDEV = 22 / 9.5 = 2.3 \approx 2 \text{ Hombres}$$

Costo

$$CHM = CH * \text{Salario Promedio} = 2 * \$50 = \$ 100$$

$$\text{Costo} = CHM * PM = \$ 100 * 22 = \$ 2 200$$

5.4 Beneficios tangibles e intangibles

PSPSOFT, no es sistema automatizado con fines comerciales, su objetivo fundamental es apoyar docentemente a los estudiantes en la asignatura Gestión de Software en la universidad, inicialmente.

El beneficio principal de este sistema es obtener una aplicación web sencilla, flexible y amena, que permita gestionar el Cuaderno de Ingeniería de forma rápida, dinámica y precisa, permitiendo ser un trabajo más cómodo para los usuarios del sistema y quedando así registrados los datos de forma tal que pueda acceder a ellos en un tiempo y un esfuerzo menor.

Por lo que los beneficios inmediatos son intangibles:

- ✓ Disminución de errores a la hora de registrar los datos.
- ✓ Disminución de tiempo y esfuerzo en la realización de la tarea.
- ✓ Disminución de los gastos, pues es más económico gestionar y procesar información digital que copias duras.
- ✓ Disminución de la pérdida de información.
- ✓ Fácil detención de problemas.
- ✓ Fácil y rápido acceso a la información almacenada.

5.5 Análisis de costo

Implementar un producto informático resulta en parte costoso, entender los beneficios que traería consigo, es lo que conduce a la decisión final de su elaboración. El sistema propuesto está orientado fundamentalmente a la docencia, por lo que su mayor beneficio no es económico sino de orden social.

Una vez desplegado el sistema contribuirá a aumentar la eficiencia y calidad de los servicios con un gasto mínimo de recursos al disminuir el tiempo necesario empleado en los registros, actualizaciones, eliminaciones de registros y consultas a la diversa información.

La tecnología empleada en el desarrollo del sistema es libre por lo que no hay que incurrir en gastos de licencia para su uso. El sistema es portable por lo que el cambio a una nueva plataforma no sería problema ya que su arquitectura está diseñada en capas siguiendo el patrón MVC.

Analizando el costo del proyecto con los beneficios que trae consigo, además de analizar su tecnología se concluye que es factible el desarrollo del mismo.

5.6 Conclusiones

En este capítulo se realizó el estudio de factibilidad, analizando el esfuerzo, costo del proyecto, los beneficios tangibles e intangibles y el análisis de costo teniendo en cuenta en este último los beneficios, costo y tecnología, deduciendo que la factibilidad de la realización de la aplicación.

La herramienta propuesta aportará beneficios a la universidad a los estudiantes que trabajen con el Cuaderno de Ingeniería, y después de haber realizado las correspondientes pruebas al sistema se puede extender a otras universidades e instalaciones del país que hagan uso de dicho cuaderno.

Conclusiones

Con el desarrollo de este trabajo se implementó una aplicación Web con el objetivo de apoyar a la docencia, automatizando el Cuaderno de Ingeniería. Con esto se logra un mejor funcionamiento en el manejo del cuaderno y una mayor rapidez en cuanto al trabajo con los datos.

Se alcanzaron además los siguientes resultados:

- ✓ Estudiantes, fácil desempeño en el manejo del Cuaderno de Ingeniería y disminución de errores al registrar los datos.
- ✓ Estudiantes de proyectos de producción, vinculación real con la asignatura.
- ✓ Profesores, mayor control del cuaderno y evaluar el desempeño de cada estudiante.

El sistema satisface los requerimientos del usuario final, cumple con las exigencias de usabilidad, disponibilidad, agradable interfaz, facilidad para aprender a usarlo y garantiza su sostenibilidad y fácil mantenimiento.

Es importante destacar que el sistema fue abalado por el departamento de Ingeniería y Gestión de software de la universidad, se validó del mismo por los estudiantes y profesores de 4to año en el curso 2006-2007. Se presentó en diferentes eventos como IV y V Jornada Científica Estudiantil obteniendo relevante a nivel de universidad por dos años consecutivos.

Recomendaciones

- ✓ Proponer, tras corroborar un desempeño exitoso, la utilización y generalización de este sistema en las diferentes universidades y lugares que se lleva a cabo el PSP en nuestro país.

- ✓ Aplicar la herramienta a estudiante de primer año de la carrera.

Referencias bibliográficas

- [1]. Nuevas tecnologías, 2006. Disponible en URL: http://cice.uci.cu/Gestionconoc/Teleformación/ApliTICeduc/word/Nuevas_tecnologías.doc [Fecha de consulta 3 febrero 2007]
- [2]. Tecnologías de la información y la comunicación, 2007. Disponible en URL: http://es.wikipedia.org/wiki/Tecnologías_de_la_información_y_la_comunicación [Fecha de consulta 3 febrero 2007]
- [3]. **Dr. Pere Marquès Graells**. Impacto de las TIC en la educación: Funciones y Limitaciones, 2000. Disponible en URL: <http://dewey.uab.es/pmarques/siyedu.htm> [Fecha de consulta 3 febrero 2007]
- [4]. 2005. Disponible en URL: <http://processdash.sourceforge.net/> [Fecha de consulta 9 febrero 2007]
- [5]. Introducción a la Gestión de Software. Principios de calidad establecidos en la UCI, 2006-2007. Disponible en URL: <http://teleformacion.uci.cu/mod/resource/view.php?id=10847> [Fecha de consulta 9 febrero 2007]
- [6]. The Team Software Process (TSP) and the Personal Software Process (PSP), 2006. Disponible en URL: <http://www.sei.cmu.edu/tsp/> [Fecha de consulta 9 febrero 2007]
- [7]. **Humphrey Watts S**. Introducción a Proceso Software Personal (PSP), 1997, 301
- [8]. **Mateu Carles**. Desarrollo de aplicaciones web, 2004. Disponible en URL: <http://www.uoc.edu/masters/esp/img/873.pdf> [Fecha de consulta 15 febrero 2007]
- [9]. Aplicación web, 2006. Disponible en URL: http://es.wikipedia.org/wiki/Aplicación_web [Fecha de consulta 15 febrero 2007]

- [10]. Cliente-servidor, 2006. Disponible en URL: <http://es.wikipedia.org/wiki/Cliente-servidor> [Fecha de consulta 15 febrero 2007]
- [11]. PHP, 2006. Disponible en URL: <http://es.wikipedia.org/wiki/PHP> [Fecha de consulta 15 febrero 2007]
- [12]. Servidor HTTP Apache, 2006. Disponible en URL: http://es.wikipedia.org/wiki/Servidor_HTTP_Apache [Fecha de consulta 17 febrero 2007]
- [13]. **Vegas Jesús**. El Servidor Web, 2002. Disponible en URL: <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html> [Fecha de consulta 17 febrero 2007]
- [14]. Una Introducción Apache, 2006. Disponible en URL: http://linux.ciberaula.com/articulo/linux_apache_intro/ [Fecha de consulta 17 febrero 2007]
- [15]. **Merelo Cuervos Juan Julian**. Introducción al lenguaje XML, 2004. Disponible en URL: <http://geneura.ugr.es/~jmerelo/xml/> [Fecha de consulta 18 febrero 2007]
- [16]. XML, 2006. Disponible en URL: http://es.wikipedia.org/wiki/XML#Objetivos_y_usos_de_XML [Fecha de consulta 18 febrero 2007]
- [17]. JavaScript, 2006. Disponible en URL : <http://es.wikipedia.org/wiki/JavaScript> [Fecha de consulta 18 febrero 2007]
- [18]. Sistema de Gestión de base de datos, 2006. Disponible en URL: http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos [Fecha de consulta 20 febrero 2007]

- [19]. Servidores de base de datos de software libre, 2006. Disponible en URL: http://72.14.203.104/search?q=cache:Ziwd9t86NcEJ:www.cidsol.org/downloads.php3Fcat_id3D226download_id3D3+gestores+base+de+datos+2B+mas+usados&hl=es&gl=cu&ct=clnk&cd=3&lr=lang_es [Fecha de consulta 20 febrero 2007]
- [20]. MySql, 2002. Disponible en URL: <http://www.softwareshop.com/Productos/MySQL/mysql.html> [Fecha de consulta 20 febrero 2007]
- [21]. MySql, 2006. Disponible en URL: <http://es.wikipedia.org/wiki/MySQL> [Fecha de consulta 20 febrero 2007]
- [22]. **Pecos Daniel.** PostgreSQL, 2006. Disponible en URL: http://www.netpecos.org/docs/mysql_postgres/x15.html [Fecha de consulta 20 febrero 2007]
- [23]. **Welicki León.** Patrones y antipatrones: una Introducción Parte II, 2005. Disponible en URL: http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.asp#M15 [Fecha de consulta 21 febrero 2007]
- [24]. Rational Unified Process, 2006. Disponible en URL: http://es.wikipedia.org/wiki/Rational_Unified_Process [Fecha de consulta 21 febrero 2007]
- [25]. Introducción a la Ingeniería de Software, 2005-2006. Disponible en URL: http://teleformacion.uci.cu/mod/resource/view.php?id=10817&subdir=/Conferencias_IS1_05-06 [Fecha de consulta 21 febrero 2007]
- [26]. Rational Rose. Disponible en URL: http://www.indudata.com/1rational_rose.htm#4 [Fecha de consulta 21 febrero 2007]
- [27]. Lenguaje Unificado de Modelado, 2006. Disponible en URL: <http://es.wikipedia.org/wiki/UML> [Fecha de consulta 21 febrero 2007]

Bibliografía

- ✓ **Álvarez Miguel Ángel.** Evaluando Zend Studio, 2003. Disponible en URL: <http://www.maestrosdelweb.com/editorial/zendstudio/> [fecha de consulta 25 febrero 2007]
- ✓ Aplicación web, 2006. Disponible en URL: http://es.wikipedia.org/wiki/Aplicación_web [fecha de consulta 15 febrero 2007]
- ✓ Cliente-servidor, 2006. Disponible en URL: <http://es.wikipedia.org/wiki/Cliente-servidor> [fecha de consulta 15 febrero 2007]
- ✓ **Dr. Pere Marquès Graells.** Impacto de las TIC en la educación: Funciones y Limitaciones, 2000. Disponible en URL: <http://dewey.uab.es/pmarques/siyedu.htm> [fecha de consulta 3 febrero 2007]
- ✓ **Humphrey Watts S.** Introducción a Proceso Software Personal (PSP), 1997, 301
- ✓ Introducción a la Gestión de Software. Principios de calidad establecidos en la UCI, 2006-2007. Disponible en URL: <http://teleformacion.uci.cu/mod/resource/view.php?id=10847> [fecha de consulta 9 febrero 2007]
- ✓ Introducción a la Ingeniería de Software, 2005-2006. Disponible en URL: http://teleformacion.uci.cu/mod/resource/view.php?id=10817&subdir=/Conferencias_IS1_05-06 [fecha de consulta 21 febrero 2007]
- ✓ **Jacobson Ivar, Booch Grady, Rumbaugh James.** El Proceso Unificado De Desarrollo De Software, 1998, 435
- ✓ JavaScript, 2006. Disponible en URL : <http://es.wikipedia.org/wiki/JavaScript> [fecha de consulta 18 febrero 2007]

- ✓ **Larman Craig.** UML y Patrones, 1997, 503
- ✓ Lenguaje Unificado de Modelado, 2006. Disponible en URL: <http://es.wikipedia.org/wiki/UML> [fecha de consulta 21 febrero 2007]
- ✓ Macromedia Dreamweaver, 2006. Disponible en URL: <http://es.wikipedia.org/wiki/Dreamweaver> [fecha de consulta 25 febrero 2007]
- ✓ **Mateu Carles.** Desarrollo de aplicaciones web, 2004. Disponible en URL: <http://www.uoc.edu/masters/esp/img/873.pdf> [fecha de consulta 15 febrero 2007]
- ✓ **Merelo Cuervos Juan Julian.** Introducción al lenguaje XML, 2004. Disponible en URL: <http://geneura.ugr.es/~jmerelo/xml/> [fecha de consulta 18 febrero 2007]
- ✓ MySql, 2002. Disponible en URL: <http://www.softwareshop.com/Productos/MySQL/mysql.html> [fecha de consulta 20 febrero 2007]
- ✓ MySql, 2006. Disponible en URL: <http://es.wikipedia.org/wiki/MySQL> [fecha de consulta 20 febrero 2007]
- ✓ Nuevas tecnologías, 2006. Disponible en URL: http://cice.uci.cu/Gestionconoc/Teleformación/ApliTICeduc/word/Nuevas_tecnologías.doc [fecha de consulta 3 febrero 2007]
- ✓ **Pecos Daniel.** PostgreSql, 2006. Disponible en URL: http://www.netpecos.org/docs/mysql_postgres/x15.html [fecha de consulta 20 febrero 2007]
- ✓ PHP, 2006. Disponible en URL: <http://es.wikipedia.org/wiki/PHP> [fecha de consulta 15 febrero 2007]
- ✓ Rational Unified Process, 2006. Disponible en URL: http://es.wikipedia.org/wiki/Rational_Unified_Process [fecha de consulta 21 febrero 2007]

- ✓ Rational Rose. Disponible en URL: http://www.indudata.com/1rational_rose.htm#4 [fecha de consulta 21 febrero 2007]
- ✓ Servidor HTTP Apache, 2006. Disponible en URL: http://es.wikipedia.org/wiki/Servidor_HTTP_Apache [fecha de consulta 17 febrero 2007]
- ✓ Sistema de Gestión de base de datos, 2006. Disponible en URL: http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos [fecha de consulta 20 febrero 2007]
- ✓ Servidores de base de datos de software libre, 2006. Disponible en URL: http://72.14.203.104/search?q=cache:Ziwd9t86NcEJ:www.cidsol.org/downloads.php3Fcat_id3D226download_id3D3+gestores+base+de+datos+2B+mas+usados&hl=es&gl=cu&ct=clnk&cd=3&lr=lang_es [fecha de consulta 20 febrero 2007]
- ✓ Tecnologías de la información y la comunicación, 2007. Disponible en URL: http://es.wikipedia.org/wiki/Tecnologías_de_la_información_y_la_comunicación [fecha de consulta 3 febrero 2007]
- ✓ The Team Software Process (TSP) and the Personal Software Process (PSP), 2006. Disponible en URL: <http://www.sei.cmu.edu/tsp/> [fecha de consulta 9 febrero 20]
- ✓ Una Introducción Apache, 2006. Disponible en URL: http://linux.ciberaula.com/articulo/linux_apache_intro/ [fecha de consulta 17 febrero 2007]
- ✓ **Vegas Jesús.** El Servidor Web, 2002. Disponible en URL: <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html> [fecha de consulta 17 febrero 2007]

- ✓ **Welicki León.** Patrones y antipatrones: una Introducción Parte II, 2005. Disponible en URL: http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.asp#M15 [fecha de consulta 21 febrero 2007]
- ✓ XML, 2006. Disponible en URL: http://es.wikipedia.org/wiki/XML#Objetivos_y_usos_de_XML [fecha de consulta 18 febrero 2007]
- ✓ 2005. Disponible en URL: <http://processdash.sourceforge.net/> [fecha de consulta 9 febrero 2007]