

**Universidad de las Ciencias Informáticas**

**Facultad 8**



*Sistema para automatizar la gestión de contenidos en aplicaciones  
con tecnología Multimedia.*

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

**Autores:**

Héctor Nicolás Vega Ríos  
Joaquín Ernesto Cisneros Gómez

**Tutores:**

Ing. Renier Portelles Cobas  
Ing. Michel Miranda Cairo

Ciudad de la Habana, Junio del 2008.

“Año 50 de la Revolución”

Curso 2007-2008

## **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 firmo la presente a los 26 días del mes de Junio del año 2008.

\_\_\_\_\_  
Firma del Autor  
Héctor Nicolás Vega Ríos

\_\_\_\_\_  
Firma del Autor  
Joaquín Ernesto Cisneros Gómez

\_\_\_\_\_  
Firma del Tutor  
Ing. Renier Portelles Cobas

\_\_\_\_\_  
Firma del Tutor  
Ing. Michel Miranda Cairo

## *Agradecimientos*

*A nuestros padres, a nuestros abuelos, a nuestra familia, a las personas que hicieron posible el desarrollo de este trabajo.*

*A nuestros tutores que han sabido guiarnos durante el desempeño de este trabajo de diploma.*

*A todos los que nos acompañaron durante el transcurso de la vida que de una manera u otra formaron parte del logro de este sueño.*

## *Dedicatoria*

*Joaquín:*

*A mis padres, que han sabido guiarme en todo momento durante el transcurso de mi existencia.*

*A mi novia, por quererme tanto y hacerme sentir muy feliz en todo momento.*

*A mis abuelas porque constituyen una parte muy importante en mi vida.*

*A mi hermanita, que es lo más lindo que me ha pasado desde que vino al mundo.*

*A mis tías y mi tío, por mantenerse siempre unidos en todo momento, y brindarme el apoyo necesario para lograr este sueño.*

*A mis primos y primas, que me recuerdan constantemente la importancia de tener una familia.*

*A mi padrastro que ha sabido brindar apoyo incondicional a mi mamá y mi hermanita.*

*A mis amigos.*

*A todos los que se sientan parte de este resultado.*

*Héctor:*

*A mi abuela Isabel, porque estoy seguro de que pocas personas me hayan querido tanto como ella.*

*A mi abuelo Héctor Ríos, por quererme tanto y por guiarme con su ejemplo, ya que es una de las personas que más admiraré en mi vida.*

*A mi mamá Yamina y mi papá Nicolás por guiarme y apoyarme incondicionalmente en todo momento.*

*A mi hermanita Linda por quererme tanto.*

*A mi tía Betty y a mi primo Raciélito por el amor que me han sabido dar.*

*A toda mi familia que de una manera u otra me han guiado hasta esta meta.*

*A todos mis amigos.*

## **RESUMEN**

La gestión de la información es uno de los pasos más importantes a la hora de crear un software con tecnología multimedia. En la Facultad 8 de la Universidad de las Ciencias Informáticas (UCI) se realiza esta fase en ficheros con formatos: PDF, Word, Excel; lo que ha causado problemas con la organización y la utilización de dicha información. Por lo antes expuesto, se realizó el análisis, diseño e implementación de una aplicación para gestionar los contenidos y generar los archivos XML asociados a la multimedia en desarrollo. La implementación de este sistema ofrecerá las ventajas de tener centralizada la información de los productos que se desarrollan en la facultad y el control de la misma, así como lograr un estándar en la estructura de los archivos XML utilizados para el desarrollo de este tipo de software, facilitando el trabajo a los programadores, ganando en eficiencia y tiempo de desarrollo. El documento que se presenta a continuación recoge los resultados del trabajo investigativo realizado. Se emplea, para la solución de la aplicación, RUP como metodología de desarrollo; PHP como lenguaje en el lado del servidor; JavaScript y CSS del lado del cliente, y MySql Server como gestor de base de datos.

### **Palabras Claves**

Sistema de Gestión de Información, Gestión de la Información de los Software con Tecnología Multimedia.

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA</b> .....	<b>3</b>
<b>1.1 - INTRODUCCIÓN</b> .....	<b>3</b>
<b>1.2 - SISTEMAS AUTOMATIZADOS SIMILARES EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN</b> .....	<b>3</b>
1.2.1 - ESCRIPNAUTA.....	3
1.2.2 - MALTED .....	3
1.2.3 - MACROMEDIA BREEZE .....	4
<b>1.3 - PROCESO DE PRODUCCIÓN DE SOFTWARE CON TECNOLOGÍA MULTIMEDÍA EN LA UCI</b> .....	<b>4</b>
<b>1.4 - CONCEPTOS IMPORTANTES</b> .....	<b>5</b>
<b>1.5 - TECNOLOGÍA CLIENTE/SERVIDOR</b> .....	<b>5</b>
1.5.1 - TECNOLOGÍAS DEL LADO DEL SERVIDOR .....	6
1.5.1.1 - <i>Lenguajes de programación del lado del servidor</i> .....	6
1.5.1.2 - <i>Servidores</i> .....	9
1.5.1.3 - <i>Gestores de base de datos</i> .....	11
1.5.1.4 - <i>Plataforma (XAMPP)</i> .....	13
1.5.2 - TECNOLOGÍAS DEL LADO DEL CLIENTE .....	14
1.5.2.1 - <i>Lenguajes del lado del cliente</i> .....	14
<b>1.6 - UML (UNIFIED MODELING LANGUAGE)</b> .....	<b>14</b>
<b>1.7 - METODOLOGÍAS A UTILIZAR</b> .....	<b>15</b>
1.7.1 - RATIONAL UNIFIED PROCESS (RUP) .....	15
1.7.2 – EXTREME PROGRAMMING (XP) .....	16
1.7.3 - MICROSOFT SOLUTION FRAMEWORK (MSF) .....	17
<b>1.8 - PATRÓN MODELO VISTA CONTROLADOR (MVC)</b> .....	<b>19</b>
<b>1.9 - HERRAMIENTAS CASE</b> .....	<b>21</b>
1.9.1 - RATIONAL ROSE .....	22
1.9.2 - VISUAL PARADIGM .....	23
1.9.3 - ARGO UML .....	23
1.9.4 - POSEIDÓN .....	24
<b>1.10 - HERRAMIENTAS</b> .....	<b>24</b>
1.10.1 - ZEND STUDIO .....	24
1.10.2 - FIREWORKS .....	25
1.10.3 - DREAMWEAVER .....	25
<b>1.11 - CONCLUSIONES</b> .....	<b>26</b>
<b>CAPÍTULO 2 MODELACIÓN DEL NEGOCIO</b> .....	<b>27</b>

<b>2.1 - INTRODUCCIÓN</b> .....	<b>27</b>
<b>2.2 - MODELO DEL NEGOCIO ACTUAL</b> .....	<b>27</b>
<b>2.3 - DESCRIPCIÓN DE LOS PROCESOS DEL NEGOCIO</b> .....	<b>28</b>
<b>2.4 - REGLAS DEL NEGOCIO</b> .....	<b>29</b>
<b>2.5 - ENTIDADES DEL NEGOCIO</b> .....	<b>29</b>
<b>2.6 - ACTORES DEL NEGOCIO</b> .....	<b>29</b>
<b>2.7 - TRABAJADORES DEL NEGOCIO</b> .....	<b>29</b>
<b>2.8 - CASOS DE USO DEL NEGOCIO</b> .....	<b>30</b>
2.8.1 - CASO DE USO DEL NEGOCIO .....	30
<b>2.9 - MODELO DE CASOS DE USO DEL NEGOCIO</b> .....	<b>30</b>
<b>2.10 - DESCRIPCIÓN DE LOS CASOS DE USO DEL NEGOCIO</b> .....	<b>31</b>
<b>2.11 - DIAGRAMA DE ACTIVIDADES (ANEXO 1)</b> .....	<b>32</b>
<b>2.12 - DIAGRAMA DE CLASES DEL MODELO DE OBJETOS</b> .....	<b>32</b>
<b>2.13 - REQUERIMIENTOS</b> .....	<b>33</b>
2.13.1 - REQUISITOS FUNCIONALES .....	33
2.13.2 - REQUERIMIENTOS NO FUNCIONALES .....	34
<b>2.14 - ACTORES DEL SISTEMA</b> .....	<b>35</b>
<b>2.15 - DIAGRAMA DE CASOS DE USO DEL SISTEMA</b> .....	<b>36</b>
<b>2.16 - DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA</b> .....	<b>37</b>
2.16.1 - DESCRIPCIÓN DEL CUS "ADMINISTRAR USUARIOS" (ANEXO 2) .....	37
2.16.2 - DESCRIPCIÓN DEL CUS "AUTENTICAR USUARIOS" (ANEXO 2) .....	37
2.16.3 - DESCRIPCIÓN DEL CUS "ADMINISTRAR PROYECTO" .....	37
2.16.4 - DESCRIPCIÓN DEL CUS "OBTENER REPORTE DEL ESTADO DEL PROYECTO" .....	39
2.16.5 - DESCRIPCIÓN DEL CUS "ADMINISTRAR MÓDULOS" .....	40
2.16.6 - DESCRIPCIÓN DEL CUS "ADMINISTRAR INFORMACIÓN DEL MÓDULO TEMAS" .....	41
2.16.7 - DESCRIPCIÓN DEL CUS "ADMINISTRAR INFORMACIÓN DEL MÓDULO GALERÍA" .....	43
2.16.8 - DESCRIPCIÓN DEL CUS "ADMINISTRAR INFORMACIÓN DEL MÓDULO EJERCICIOS" .....	45
2.16.9 - DESCRIPCIÓN DEL CUS "ESTRUCTURAR CONTENIDO" .....	46
<b>2.17 - CONCLUSIONES</b> .....	<b>47</b>
<b>CAPÍTULO 3 ANÁLISIS Y DISEÑO</b> .....	<b>48</b>
<b>3.1 - INTRODUCCIÓN</b> .....	<b>48</b>
<b>3.2 - ANÁLISIS</b> .....	<b>48</b>
3.2.1 - MODELO DE CLASES DE ANÁLISIS .....	48
3.2.2 - Diagrama de clases del análisis .....	48



3.2.2.1 - Diagramas de clases de análisis de Gestionar Usuarios .....	48
3.2.2.2 - Diagramas de clases de análisis de Autenticar Usuarios .....	49
3.2.2.3 - Diagramas de clases de análisis de Gestionar proyecto .....	49
3.2.2.5 - Diagramas de clases de análisis de Administrar Módulos .....	51
3.2.2.6 - Diagramas de clases de análisis de Administrar Información del Módulo Temas .....	52
3.2.2.7 - Diagramas de clases de análisis de Administrar Información del Módulo Galería .....	53
3.2.2.8 - Diagramas de clases de análisis de Administrar Información del Módulo Ejercicios .....	54
3.2.2.9 - Diagramas de clases de análisis de Estructura Información .....	55
3.2.3 - DIAGRAMAS DE COLABORACIÓN. (ANEXO 3) .....	55
<b>3.3 - DISEÑO .....</b>	<b>55</b>
3.3.1 - MODELO DE CLASES DEL DISEÑO .....	55
3.3.2 - DIAGRAMAS DE CLASES DE DISEÑO .....	56
3.3.2.1 - Diagrama de clases de diseño del Caso de Uso Gestionar Usuarios .....	56
3.3.2.2 - Diagrama de clases de diseño del Caso de Uso Autenticar Usuarios .....	58
3.3.2.3 - Diagrama de clases de diseño del Caso de Uso Gestionar Proyecto .....	58
3.3.2.4 - Diagrama de clases de diseño del Caso de Uso Obtener Reporte .....	62
3.3.2.5 - Diagrama de clases de diseño del Caso de Uso Administrar Módulos .....	63
3.3.2.6 - Diagrama de clases de diseño del Caso de Uso Administrar Información del Módulo Temas .....	65
3.3.2.7 - Diagrama de clases de diseño del Caso de Uso Administrar Información del Módulo Galería .....	67
3.3.2.8 - Diagrama de clases de diseño del Caso de Uso Administrar Información del Módulo Ejercicios .....	69
3.3.2.9 - Diagrama de clases de diseño del Caso de Uso Estructura Información .....	71
3.3.3 - DESCRIPCIÓN DE LAS CLASES (ANEXO 4) .....	71
3.3.4 - PRINCIPIOS DE DISEÑO .....	71
3.3.4.1 - Interfaz de Usuario .....	71
3.3.4.2 - Tratamiento de Errores .....	72
3.3.4.3 - Seguridad .....	73
3.3.5 - DIAGRAMA DE CLASES PERSISTENTES (VER ANEXO 5) .....	73
3.3.6 - MODELO DE DATOS (VER ANEXO 6) .....	73
3.3.7 - CONCLUSIONES .....	73
<b>CAPÍTULO 4 IMPLEMENTACIÓN .....</b>	<b>74</b>
<b>4.1 - INTRODUCCIÓN .....</b>	<b>74</b>
<b>4.2 - MODELO DE IMPLEMENTACIÓN .....</b>	<b>74</b>
4.2.1 - DIAGRAMA DE COMPONENTES POR PAQUETES .....	75
4.2.2 - DIAGRAMA DE COMPONENTES AUTENTICAR USUARIO .....	76
4.2.3 - DIAGRAMA DE COMPONENTES GESTIONAR USUARIOS .....	76
4.2.4 - DIAGRAMA DE COMPONENTES GESTIONAR PROYECTOS .....	77
4.2.5 - DIAGRAMA DE COMPONENTES ADMINISTRAR MÓDULOS .....	78
4.2.6 - DIAGRAMA DE COMPONENTES ADMINISTRAR INFORMACIÓN MÓDULO TEMA .....	79
4.2.7 - DIAGRAMA DE COMPONENTES ADMINISTRAR INFORMACIÓN MÓDULO GALERÍA .....	80
4.2.8 - DIAGRAMA DE COMPONENTES ADMINISTRAR INFORMACIÓN MÓDULO EJERCICIO .....	81
4.2.9 - DIAGRAMA DE COMPONENTES OBTENER REPORTE .....	81
4.2.10 - DIAGRAMA DE COMPONENTES ESTRUCTURAR CONTENIDO .....	82

---

<b>4.3 - DIAGRAMA DE DESPLIEGUE .....</b>	<b>82</b>
4.3.1 - DIAGRAMA DE DESPLIEGUE.....	83
4.4 - CONCLUSIONES .....	83
<b>CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD.....</b>	<b>84</b>
<b>5.1 - INTRODUCCIÓN .....</b>	<b>84</b>
<b>5.2 - PLANIFICACIÓN MEDIANTE PUNTOS DE CASOS DE USO .....</b>	<b>84</b>
<b>5.3 - COSTO.....</b>	<b>90</b>
<b>5.4 - BENEFICIOS TANGIBLES E INTANGIBLES .....</b>	<b>90</b>
5.4.1 - TANGIBLES.....	90
5.4.2 – INTANGIBLES .....	90
<b>5.5 - ANÁLISIS COSTO-BENEFICIO .....</b>	<b>91</b>
<b>5.6 - CONCLUSIONES .....</b>	<b>91</b>
<b>CONCLUSIONES GENERALES .....</b>	<b>92</b>
<b>RECOMENDACIONES .....</b>	<b>93</b>
<b>BIBLIOGRAFÍA</b>	
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	
<b>GLOSARIO DE TÉRMINOS</b>	
<b>ANEXOS</b>	

## **Introducción**

En la actualidad la producción de software con tecnología multimedia es el proceso que da como resultado una aplicación donde se combinan diferentes tipos de información como texto, imágenes, videos, entre otras, relacionados de manera coherente, todo referente a un tema deseado.

Nuestro país es uno de los que produce esta herramienta con diversos objetivos donde se abarcan diferentes temáticas tales como política, sociales, económicas, salud, ciencia, deportes, entre otras. La UCI no se encuentra exenta de ello, la misma se ha convertido en una de las productoras de aplicaciones con tecnología multimedia en el país, en especial la facultad 8.

Como todo proceso la creación de una multimedia consta de un flujo de trabajo. Este flujo, actualmente se caracteriza por la sucesión de diferentes etapas, una de las cuales es la gestión de la información a la hora de obtener un producto, que en ocasiones requiere de un largo espacio de tiempo obstruyendo el trabajo eficiente para lograr un producto en menor tiempo con igual calidad. La información que se manipula relacionada con las cuestiones relativas a la gestión de la información de multimedia, se almacena en muchos casos en papel, documentos Word, Excel, PDF, que en ocasiones puede provocar con el paso del tiempo, pérdidas de datos, demoras a la hora de realizar una búsqueda de la información proveniente de diferentes fuentes, viéndose afectada la seguridad del contenido debido a la aparición de imprevistos tales como: virus, problemas en el Sistema Operativo, errores humanos, entre otros; además aumenta el tiempo que se consume durante la etapa de implementación al tener que analizar, organizar y estructurar manualmente los contenidos en ficheros. En este caso los programadores utilizan parte del tiempo que está previsto para la implementación en la elaboración de ficheros que sirven para almacenar contenido, reduciéndole en gran medida el volumen de trabajo a la hora de la implementación del producto.

Se puede decir que estos problemas se ven reflejado en la Facultad 8 a la hora de producir aplicaciones con tecnología multimedia, por tanto el **problema científico** queda enmarcado en:

¿Cómo crear una herramienta que agilice el proceso de gestión de la información y ficheros XML que sirvan de estándar para la implementación de aplicaciones con tecnología multimedia?

El **objeto de estudio** estará dirigido al proceso de gestión de la información para los software con tecnología multimedia, siendo el **campo de acción** el proceso de gestión de la información para los software con tecnología multimedia en la Facultad 8 de la UCI.

Con la investigación se quiere **defender la idea** de que si se crea una herramienta capaz de automatizar el proceso de gestión de la información y si se obtienen ficheros con los contenidos de un software con tecnología multimedia, facilitará la implementación del producto disminuyendo el tiempo de desarrollo.

**El objetivo general** de este trabajo para dar solución al problema en cuestión es desarrollar una Aplicación Web en la Facultad 8 de la UCI, que permita gestionar la información relacionada con la producción de software con tecnología multimedia, generándose además una estructura de archivos que contengan la información.

Se trazaron como **objetivos específicos**: realizar un estudio sobre los términos conceptuales y los software de gestión, realizar el diseño y la implementación de los módulos administración, gestión y creación de ficheros de contenido y conformar la documentación del sistema con vista al desarrollo de posteriores versiones.

Con el fin de alcanzar los objetivos propuestos se definieron las siguientes **tareas**: realizar un estudio del arte; estudiar el proceso de gestión de la información para la producción de software con tecnología multimedia; seleccionar la metodología más adecuada para el desarrollo del sistema; seleccionar las herramientas apropiadas para diseñar la aplicación; realizar el análisis y diseño del sistema; diseñar una base de datos que soporte las funcionalidades del sistema informático que se pretende desarrollar y realizar la implementación del sistema.

El presente documento está compuesto por 5 capítulos:

En el Capítulo 1 "*Fundamentación Teórica*" se exponen los fundamentos generales que sirven de soporte teórico en la solución del problema. Se analizan las herramientas y lenguajes de programación para comprobar que son las idóneas para el desarrollo Aplicación. Además se plantea la metodología a emplear en el desarrollo del mismo.

En el Capítulo 2 "*Modelación del negocio*" se exponen las políticas y condiciones que deben cumplirse, entendidas como reglas del negocio asociadas al campo de acción. Se describen los actores, trabajadores del negocio y el modelo de objetos.

En el Capítulo 3 "*Diseño del Sistema*" se presentan los diagramas de clases de análisis y diagramas de diseño Web que describen la interacción entre las distintas páginas.

En el Capítulo 4 "*Implementación*" se representa el diagrama de componentes que detalla la forma en que está estructurado el sistema y de despliegue que brindan una mayor vista para la implementación del producto.

En el Capítulo 5 "*Estudio de la Factibilidad*" se exponen un estudio de la factibilidad del sistema utilizando el método de análisis por Punto de Casos de Uso.

## **CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA**

### **1.1 - Introducción**

En este capítulo se realiza un análisis sobre la situación problemática planteada, además del estado del arte entorno al objeto de investigación y al campo de acción. Se hace referencia a las tendencias, técnicas, tecnologías, metodologías y software que se utilizan en la actualidad que ayudarían a formar parte de la propuesta de solución del problema a resolver.

### **1.2 - Sistemas automatizados similares existentes vinculados al campo de acción**

En la actualidad con el desarrollo creciente de la informática y los medios de comunicación a lo largo de todo el mundo, muchas empresas e instituciones han comenzado a desarrollar herramientas que permiten gestionar información para desarrollar software con tecnologías multimedia.

#### **1.2.1 - Escriptnauta**

Escriptnauta: (Sistema de creación y generación de presentaciones multimedia) desarrollado en España por la Universidad de las Islas Baleares, entre los años 2002 y 2003. Nació con la filosofía de llegar al máximo número de usuarios posible sin discriminación de sistema operativo y/o navegador, basándose en estándares y software libre; dirigido a la creación de multimedia en líneas. Su objetivo primordial es la elaboración de una herramienta que permita la creación remota y cooperativa de presentaciones multimedia, integrando texto, imágenes y sonidos mediante el uso de un navegador Web estándar y una conexión a Internet; también el sistema permite modificar cualquier elemento de animaciones, en cualquier momento y en cualquier lugar. Esta aplicación ha sido íntegramente desarrollada con Apache, PHP, MySQL, Ming y utilidades de ImageMagick. (1)

El software presenta una serie de dificultades como son, que los usuarios para ver los programas con tecnologías multimedia necesitan del acceso a la red de redes (Internet), pues esta aplicación no permite crear multimedia de forma off-line. Lo cual dificulta la distribución de los productos finales, pues la misma no se puede realizar en CD, DVD u otros dispositivos convencionales de almacenamiento de datos, además si falla el sistema que soporta la aplicación, por culpa de diversos factores como son, virus, fallos en los servidores de datos, fallos humanos, fallos en las redes, se perdería la información almacenada, así como el acceso a las multimedia creadas y dificultaría el uso de las mismas por los clientes.

#### **1.2.2 - Malted**

Malted es una herramienta informática de autor para la creación y ejecución de unidades didácticas multimedia e interactivas para ser utilizadas por los usuarios como prácticas de aprendizaje en aulas dotadas tecnológicamente. Esta herramienta ha sido desarrollada en particular para la enseñanza de

idiomas , pero su uso se puede extender a otras materias del currículo escolar; además cuenta con numerosas plantillas que permiten generar fácilmente buena parte de los ejercicios usualmente empleados en la enseñanza de idiomas, articulando además la incorporación de contenidos renovables y modificables, así como de texto, imágenes, gráficos, audio, vídeo y una aplicación para grabar la voz mediante un micrófono externo. (2)

Esta herramienta tiene la ventaja de ser multiplataforma y además permite organizar las actividades en secuencias lógicas siguiendo un plan de desarrollo didáctico. La desventaja está en que solamente se puede realizar productos enfocados a la rama de los idiomas reduciendo su campo de aplicación a un margen muy estrecho del conocimiento.

### **1.2.3 - Macromedia Breeze**

Macromedia presenta la tecnología Breeze, que hace que la comunicación web sea valiosa, efectiva y útil. Breeze proporciona un sistema de comunicación web dinámico que permite a los usuarios mantener reuniones en tiempo real, clases guiadas por un instructor, y seminarios, así como para ofrecer presentaciones informativas bajo demanda y cursos e-learning. Breeze aprovecha Microsoft PowerPoint y Macromedia Flash Player para que aproximadamente 500 millones de usuarios web obtengan de forma inmediata reuniones online y contenido dinámico sin tener que descargar plug-ins especiales o media players. (3)

Macromedia Breeze está disponible en versiones de prueba, "hosting" y con licencias, por lo que presenta la desventaja de ser un software propietario y necesita una conexión en tiempo real para su utilización.

## **1.3 - Proceso de producción de software con tecnología multimedia en la UCI**

La Universidad de las Ciencias Informáticas es uno de los pilares más importantes en la rama de producción de software con tecnología multimedia, dentro de ella la Facultad 8 es destacada en este ámbito productivo. La creación de un software con tecnología multimedia es un proceso que lleva una serie de pasos fundamentales.

Para realizar una multimedia en la facultad el cliente solicita la creación de un software con tecnología multimedia, pero si el mismo no presenta la información en un guión, de esta manera debe existir un guionista (especialista) que sería el encargado en ponerse de acuerdo con el cliente de cómo quedaría el guión para la aplicación, se escogen las herramientas a utilizar, y se hace un mapa de navegación para tener una vista previa de como va a quedar el sistema, luego a medida que el equipo de desarrollo va interactuando con la información brindada van surgiendo un serie de funcionalidades que serían implementadas por los programadores posteriormente, creándose una serie de hitos en este

proceso de producción, se consulta al cliente para ver si está de acuerdo con el trabajo realizado y se llevan a cabo una serie de pruebas para corroborar la calidad del software.

Dentro de este proceso productivo existe un paso que es imprescindible a la hora de obtener el producto, este es la gestión de la información; que en la Facultad 8 de la UCI, en ocasiones se realiza de manera que puede ocasionar consecuencias adversas para el desarrollo del software, pues cuando se hace entrega de la información, esta es almacenada en formatos como: PDF, Word, Excel, entre otras; que por sus características pueden peligrar; luego al no producirse una buena entrega de la información pueden suceder una serie de contratiempos que dificultarían la obtención de un buen resultado en el tiempo requerido.

#### **1.4 - Conceptos importantes**

Cuando se realiza una aplicación web es necesario el análisis de algunos conceptos fundamentales a la hora de trabajar sobre ella, entre ellos están:

**HTTP:** (HiperText Transfer Protocol) Protocolo de Transmisión Hipertexto. Protocolo de comunicaciones utilizado por los programas clientes y servidores de WWW para comunicarse entre sí.

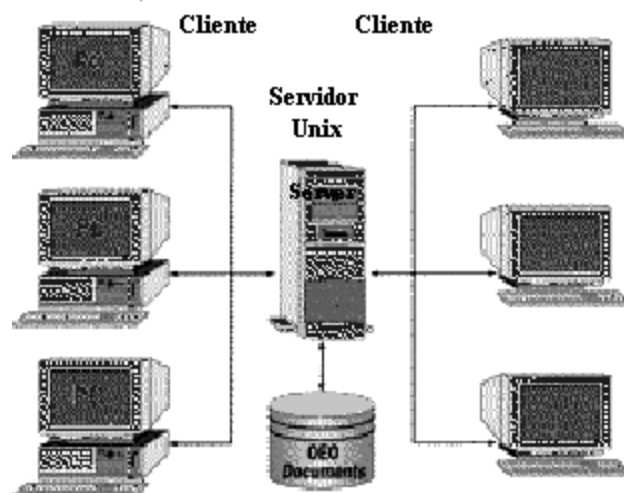
**URL:** (Uniform Resource Locators) Es la cadena de caracteres con la cual se asigna dirección única a cada uno de los recursos de información disponibles en Internet. (4)

**CGI:** (Common Gateway Interface) Un script CGI es una pequeña aplicación que se ejecuta en el servidor, normalmente a petición de una página web alojada en el mismo. Los CGI proporcionan la posibilidad de acceder a bases de datos, intercambio de información a través de formularios HTML, gestión de claves de accesos, utilidades de búsqueda, etc. Los scripts CGI pueden elaborarse en cualquier lenguaje, aunque el más usado es el PERL y Visual Basic. (5)

**LOG:** Son archivos en los que se recogen las visitas que tienen las páginas de un sitio web. (6)

#### **1.5 - Tecnología Cliente/Servidor**

La Tecnología Cliente/Servidor **desde un punto de vista conceptual** es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información y vista desde la perspectiva de **términos de arquitectura** son los distintos aspectos que caracterizan a una aplicación (proceso, almacenamiento, control y operaciones de entrada y salida de datos) en el sentido más amplio, están situados en más de un computador, los cuales se encuentran interconectados mediante una red de comunicaciones.



Las características principales de este modelo están en que el Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes. Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma. Un servidor da servicio a múltiples clientes en forma concurrente. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.

Cliente/Servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo. (7)

### 1.5.1 - Tecnologías del lado del servidor

En el desarrollo de software utilizando la tecnología de Cliente/Servidor se debe tener presente los diferentes lenguajes de programación y aplicaciones que se pueden utilizar en ambas partes. De la gran variedad que existen actualmente se seleccionaron los más importantes y adecuados a ser utilizados.

#### 1.5.1.1 - Lenguajes de programación del lado del servidor

Entre los lenguajes del lado del servidor podemos encontrar algunos que son de gran utilidad como PHP, JSP, PERL, ASP, entre otros; cada uno de ellos explota más a fondo ciertas características que lo hacen más o menos útiles para desarrollar distintas aplicaciones, estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, siendo los encargados del acceso a Bases de Datos (BD), tratamiento de la información, entre otros; además son aquellos lenguajes que son



reconocidos, ejecutados e interpretados por el propio servidor enviando al cliente un archivo HTML en un formato comprensible para él.

**PHP (Hypertext Pre-procesos):** es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor. El programa PHP es ejecutado y el resultado es enviado al navegador. El resultado es normalmente una página HTML, no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP.

Se ha probado que a pesar de que es un lenguaje interpretado, el desempeño es superior o igual a los lenguajes compilados como ASP o JSP. Los dos mayores puntos fuertes de PHP son su simplicidad y la inmensa funcionalidad que provee. Incorpora la elegante sintaxis del lenguaje C y el poder de las estructuras de Perl sin su complejidad. PHP posee un inmenso arsenal de funciones para cada necesidad. PHP puede ser extendido debido a que posee un conjunto bien definido de APIs en C. PHP puede hacer muchísimo con tan solo unas pocas líneas de programación. Además posee algunas características que lo convierten en el lenguaje adecuado a la hora de realizar una aplicación web, como son:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

- PHP es un potente lenguaje y el intérprete, tanto incluido en el servidor Web como módulo o ejecutado como un binario CGI, puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor Web sea seguro por defecto.(8)

**JSP (Java Server Pages):** Es una interfaz de programación de aplicaciones de servidores Web. En una página jsp se entremezclan bloques de HTML estáticos, y HTML dinámico generados con Java que se ejecutan en el servidor. Una página jsp puede procesar formularios Web, acceder a bases de datos y redireccionar a otras páginas. Las páginas jsp son transformadas a un servlet y después compiladas. El contenedor JSP proporciona un motor que interpreta y procesa una página JSP como un servlet (en tomcat, dicho motor se llama jasper). Al estar basadas en los servlets, las distintas peticiones a una misma página jsp son atendidas por una única instancia del servlet.

Características:

- Permiten separar la parte dinámica de la estática en una página web.
- Las páginas jsp se almacenan en el servidor en archivos con extensión .jsp.
- La sintaxis también se puede expresar en formato XML
- En una página jsp hay varios objetos implícitos.
- Cada página JSP es compilada automáticamente hacia un servlet por el motor JSP la primera vez que se accede a esa página.
- Desde una página JSP se puede llamar a un componente JavaBean donde se puede implementar la lógica de negocio. (9)

**PERL:** Siglas de "**Practical Extraction Report Language**". Lenguaje de programación derivado de una herramienta UNIX, basa su potencial en su manera de manejar texto, fue inventado en 1986 para manejar grandes cantidades de texto, además es sencillo de aprender y manejar; se dice de él que es una manera de pensar más que un lenguaje. Comparte sintaxis con otro lenguaje de programación, el C.

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el web. Además permite extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Características:

- Es un lenguaje libre de uso, eso quiere decir que es gratuito.
- Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows.

- Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como Javascript o ASP. Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito.
- Es extensible a partir de otros lenguajes, ya que desde Perl podremos hacer llamadas a subprogramas escritos en otros lenguajes. También desde otros lenguajes se puede ejecutar código Perl.
- Una diferencia fundamental de Perl con respecto a los otros lenguajes es que no limita el tamaño de los datos con los que trabaja, el límite lo pone la memoria que en ese momento se encuentre disponible. (10)

**ASP (Active Server Pages):** Siglas correspondientes a "Servidor de Páginas Activas". Es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente.

Active Server Pages (ASP) y ASP.NET es un intento de Microsoft para introducirse en el mercado del desarrollo Web, y viene a ser como su estándar para su servidor Web, ISS. Asp ha sido atacado por la comunidad open source desde que este apareció. La realidad es que ASP ha sido implementado en otras plataformas y que cuando está funcionando bajo su servidor predeterminado IIS (Internet Information Server) es relativamente rápido.

El hecho de que puedas escribir módulos para ASP usando Visual Basic y COM podría parecer una solución realmente atractiva si no fuera por el hecho de que ASP solamente es compatible con IIS. En otras plataformas hay muchas características de ASP que son realmente lentas. (11)

### **1.5.1.2 - Servidores**

Cuando llamamos a un equipo **servidor** nos referimos a un equipo que va a proporcionar determinados servicios a los usuarios de una red local o de Internet. Lleva un equipamiento de prestaciones más avanzadas que un equipo de sobremesa y sobre todo incorpora un sistema operativo de servidor. (12)

**Servidor web** es el programa que, utilizando el protocolo de comunicaciones HTTP, es capaz de recibir peticiones de información de un programa cliente (navegador), recuperar la información solicitada y enviarla al programa cliente para su visualización por el usuario. (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.

Este software libre posee una gran popularidad, y esta grandemente reconocido en muchos ámbitos empresariales y tecnológicos, por diferentes razones como:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.

- Es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber, sin ningún secreto, sin ninguna puerta trasera).
- Altamente configurable de diseño modular y muy sencillo ampliar las capacidades del servidor Web. Actualmente existen muchos módulos que son adaptables a este, y están ahí para que se instalen cuando sean necesarios. Otra cosa importante es que cualquiera que posea una experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- Trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte de Perl tanto con soporte CGI como con soporte modo Perl. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.(14)

**IIS (Internet Information Server) de Windows** es el servidor de páginas web de Microsoft. Este viene de forma gratuita con el sistema operativo Windows NT, 2000 y XP y descargable para los demás.

Características:

- Microsoft ha mejorado sustancialmente su software estrella en el campo de los servicios Web. Los avances vienen motivados sobre todo por la seguridad y el rendimiento, aunque todavía adolece de algunos agujeros de seguridad.
- Las características agregadas en seguridad se aprovechan de las últimas tecnologías de cifrado y métodos de autenticación mediante certificados de cliente y servidor. Una de las formas que tiene IIS de asegurar los datos es mediante SSL (Secure Sockets Layer). Esto proporciona un método para transferir datos entre el cliente y el servidor de forma segura, permitiendo también que el servidor pueda comprobar al cliente antes de que inicie una sesión de usuario.
- Otra característica nueva es la autenticación implícita que permite a los administradores autenticar a los usuarios de forma segura a través de servidores de seguridad y proxy.

- IIS 5.0 también es capaz de impedir que aquellos usuarios con direcciones IP conocidas obtengan acceso no autorizado al servidor, permitiendo especificar la información apropiada en una lista de restricciones.
- La administración de la seguridad del servidor IIS es una tarea fácilmente ejecutable a base de asistentes para la seguridad. Se pueden definir permisos de acceso en directorios virtuales e incluso en archivos, de forma que el asistente actualizará los permisos NTFS para reflejar los cambios.

Este servidor posee la desventaja de ser propietario, sin embargo viene de forma gratuita incluido en algunos de los sistemas operativos de Microsoft. (15)

### **1.5.1.3 - Gestores de base de datos**

**MySQL Database Server** es la base de datos de código fuente abierto más usada del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. La exclusiva separación del core server del manejador de tablas, permite funcionar a MySQL bajo control estricto de transacciones o con acceso a disco no transaccional ultrarrápido.

- MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto monto de información en una red corporativa. MySQL Server permite agregar, acceder y procesar datos guardados en un computador.
- 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. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.
- Es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones.
- MySQL Database Server es muy rápido, confiable y fácil de usar. El mismo tiene un conjunto de características desarrolladas en cercana cooperación con sus usuarios. MySQL Server fue desarrollado inicialmente para manejar grandes bases de datos mucho más rápidamente que

las soluciones existentes y ha sido usado exitosamente por muchos años en ambientes de producción de alta demanda. A través de constante desarrollo, MySQL Server ofrece hoy una rica variedad de funciones. Su conectividad, velocidad y seguridad hacen a MySQL altamente satisfactorio para trabajar con bases de datos. (16)

**Oracle** es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

Desventaja:

- El principal rival comercial de Oracle es el SQL Server de Microsoft Corporation, el cual ofrece las mismas características a un menor costo.
- Es un software propietario por lo que su código fuente no permite modificaciones.
- Ha sido criticado por algunos especialistas la seguridad de la plataforma, y las políticas de suministro de parches de seguridad.
- Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySql o Firebird.(17)

**Microsoft SQL Server 7.0** constituye un lanzamiento determinante para los productos de bases de datos de Microsoft, continuando con la base sólida establecida por SQL Server 6.5. Como la mejor base de datos para Windows NT, SQL Server es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software (ISVs) que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos.

SQL Server 7.0 disminuye el costo total de propiedad a través de características como administración multi-servidor y con una sola consola; ejecución y alerta de trabajos basadas en eventos; seguridad integrada; y scripting administrativo. Esta versión también libera al administrador de base de datos para aspectos más sofisticados del trabajo al automatizar las tareas de rutina. Al combinar estos poderosos servicios de administración con las nuevas características de configuración automática, Microsoft SQL Server 7.0 es la elección ideal de automatización de sucursales y aplicaciones de base

de datos insertadas. Posee la desventaja de ser un software propietario, ya que pertenece como su nombre lo indica a la compañía Microsoft. (19)

**PostgreSQL:** Sistema de administración de bases de datos relacionales de código abierto que utiliza un subconjunto de instrucciones del lenguaje SQL. (41)

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos PostgreSQL soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario. PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos. PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido. PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL. (42)

#### **1.5.1.4 - Plataforma (XAMPP)**

Un programa que permite instalar Apache, PHP y MySQL en un solo paso y de manera sencilla en sistemas Linux, Windows, Mac OS X y Solaris. Permite contar con estos programas correctamente configurados, sin tener que preocuparnos en los pormenores de cada uno.

El XAMPP es un paquete de aplicaciones que consta de:

- Servidor Web Apache (Intérprete de lenguaje HTML).
- Servidor de Base de Datos MySql.
- Intérprete de Lenguaje PHP.
- Aplicación para el Manejo de Bases de Datos Online phpMyAdmin.
- Servidor FTP Filezilla.
- Sevidor SSH OpenSSL.

El XAMPP es la plataforma ideal para instalar un perfecto servidor web sin tener los más mínimos conocimientos de un Sistema Operativo Multiproceso, Multitarea y Multiusuario (Unix, Linux y sus derivados), lo cual nos permite tener un Servidor de Prueba en forma local, es decir, en nuestro computador, incluso sin tener conexión a Internet.(20)

## **1.5.2 - Tecnologías del lado del cliente**

A igual que en el servidor en el lado del cliente se utiliza lenguajes, aplicaciones que son necesarias para un buen desarrollo de un sistema. A continuación se detallan algunos de ellos.

### **1.5.2.1 - Lenguajes del lado del cliente**

**JavaScript:** Lenguaje de scripts desarrollado por Netscape. Este tipo de lenguajes es muy útil para la elaboración de pequeños programas, muy utilizados para resaltar los contenidos de una página Web. Es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. (21)

Tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado. Además posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones, utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. (22)

**HTML (HiperText Markup Language):** Lenguaje basado en marcas o etiquetas diseñado para la creación y publicación de páginas web. HTML es una sintaxis para definir los elementos que se quieren mostrar en una página, y su estructura y ubicación, así como las relaciones entre ellos y con otros elementos de la web mediante hipervínculos. (23)

**XML (eXtensible Markup Language):** Es considerado como un metalenguaje de definición de documentos estructurados mediante marcas o etiquetas. Se trata de un estándar del W3C cuyo objetivo es crear unas reglas básicas para permitir el intercambio de información estructurada entre aplicaciones, y en particular, entre aplicaciones web. (24)

## **1.6 - UML (Unified Modeling Language)**

UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos. Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software.

UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelaje como OMT (Object Modeling Technique) o Booch sí definen procesos concretos. En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no puede ser el mismo el proceso para crear una aplicación



en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión, por poner un ejemplo.

UML no define un proceso concreto que determine las fases de desarrollo de un sistema, las empresas pueden utilizar UML como el lenguaje para definir sus propios procesos y lo único que tendrán en común con otras organizaciones que utilicen UML serán los tipos de diagramas. UML es un método independiente del proceso. Los procesos de desarrollo deben ser definidos dentro del contexto donde se van a implementar los sistemas. (30)

## **1.7 - Metodologías a utilizar**

Una de las múltiples definiciones de metodología, extraída del Diccionario de la Real Academia de la Lengua (1992: 1366) se refiere al término como "Ciencia del método", "Conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal". (27)

### **1.7.1 - Rational Unified Process (RUP)**

**Rational Unified Process:** es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable. Las mejores prácticas del Rational Unified Process, (RUP), son un conjunto de procesos web-enabled de ingeniería de software que dan guía para conducir las actividades de desarrollo del equipo. Como una plataforma de procesos que abarca todas las prácticas de la industria, el RUP permite seleccionar fácilmente el conjunto de componentes de proceso que se ajustan a las necesidades específicas del proyecto. Se podrán alcanzar resultados predecibles unificando el equipo con procesos comunes que optimicen la comunicación y creen un entendimiento común para todas las tareas, responsabilidades y artefactos. Desde un único sitio web centralizado de intercambio, el Software Rational, las plataformas, herramientas y expertos de dominios proveen los componentes de proceso necesarios para el éxito. (28)

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

- Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción: En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- Transición: El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

### **Disciplina de Desarrollo**

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

### **Disciplina de Soporte**

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.
- Distribución: Hacer todo lo necesario para la salida del proyecto

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Los elementos de RUP son:

- Actividades: Son los procesos que se llegan a determinar en cada iteración.
- Trabajadores: Vienen hacer las personas involucradas en cada proceso.
- Artefactos: Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

### **1.7.2 – Extreme Programming (XP)**

Programación Extrema por sus siglas en inglés Extreme Programming (XP), es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo y pequeños equipos. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Características de XP, la metodología se basa en:

- Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.

- Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

### **¿Qué es lo que propone XP?**

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

### **Derechos del Cliente**

- Decidir que se implementa
- Saber el estado real y el progreso del proyecto
- Añadir, cambiar o quitar requerimientos en cualquier momento
- Obtener lo máximo de cada semana de trabajo
- Obtener un sistema funcionando cada 3 o 4 meses

### **Derechos del Desarrollador**

- Decidir como se implementan los procesos
- Crear el sistema con la mejor calidad posible
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos
- Estimar el esfuerzo para implementar el sistema
- Cambiar los requerimientos en base a nuevos descubrimientos

### **Lo fundamental en este tipo de metodología es:**

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. (29)

### **1.7.3 - Microsoft Solution Framework (MSF)**

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se

centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF tiene las siguientes características:

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

- **Modelo de Arquitectura del Proyecto:** Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.
- **Modelo de Equipo:** Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.
- **Modelo de Proceso:** Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.
- **Modelo de Gestión del Riesgo:** Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.
- **Modelo de Diseño del Proceso:** Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y

físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.

- **Modelo de Aplicación:** Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores. (29)

### **1.8 - Patrón Modelo Vista Controlador (MVC)**

El patrón Modelo-Vista-Controlador (MVC) descompone una aplicación interactiva en tres grandes bloques:

El **modelo** contiene los datos y la funcionalidad de la aplicación. Es independiente de la representación de los datos.

Los objetos de negocio son los objetos del Modelo que implementan la lógica de negocio. Sus funciones fundamentales son:

- Realizar la validación de los datos introducidos por el usuario, tanto sintáctica (numérico, fecha, etc.) como lógica (importe menor que saldo, etc.)
- Ejecutar la petición realizada por el usuario. Para ello podrá valerse de transacciones contra Sistemas Host, consultas a bases de datos, consultas a proveedores de contenidos, etc.
- Generar los objetos que utilizarán las Vistas para mostrar los resultados obtenidos.
- Garantizar la navegación y el flujo de pantallas correcto.

Las **vistas** muestran la información al usuario de una cierta forma. Existen todas las que se necesite definir, es la encargada de interactuar con el usuario y se corresponde con lo que tradicionalmente se conoce como Interfaz de Usuario.

Para este caso las vistas serán del tipo PHP. Las PHP's son las encargadas de generar las páginas HTML que constituyen la visualización de los resultados generados por la petición realizada por el usuario. Para recibir la información que deben mostrar acceden a objetos generados por el Modelo. La recepción de la información se puede hacer de varias maneras, aunque las más habituales son:

- Accediendo a objetos almacenados dentro del Modelo como propiedades.
- El Modelo almacena los resultados como atributos de la petición.
- El Modelo escribe los resultados en la "sesión".

Cada vista tiene un **controlador** asociado. Los controladores reciben entradas en forma de eventos que responden a mandos realizados por el usuario a través del ratón o del teclado. El control traduce estos eventos a peticiones a la vista o al modelo.

Cada petición se identifica mediante un parámetro. En base a esta identificación, el Controlador decide qué objeto u objetos de negocio (Modelo) debe ejecutar para resolver la petición.

Tras la ejecución de los objetos de negocio, y en función del resultado devuelto por estos, el Controlador determina qué PHP se usará para visualizar el resultado, generando una redirección que concluirá con la generación del código HTML (48)

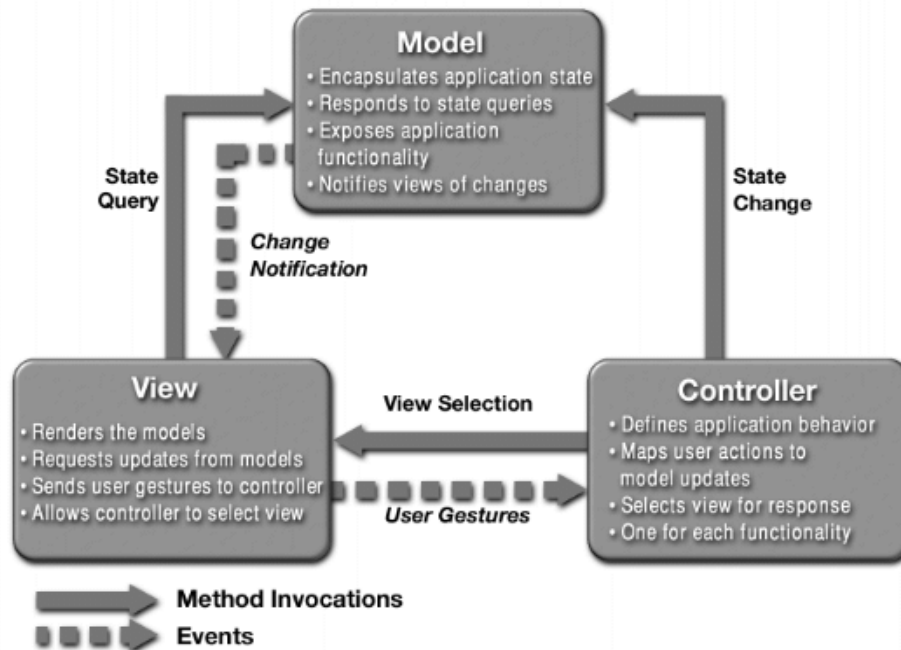
**Ventajas:**

- Múltiples vistas del mismo modelo.
- Vistas sincronizadas.
- Flexibilidad para cambiar las vistas y los controladores.
- La aplicación puede soportar distintos tipos de interfaz de usuario.
- Aumenta en gran medida el nivel de reusabilidad de código. Facilita una evolución continuada de los sistemas, sin puntos de ruptura, ya que un cambio en un sistema afectará a uno o más componentes pero nunca afectará significativamente al *core* de la aplicación.
- Facilidad de desarrollo y acortamiento del “Tiempo de Comercialización” gracias a la paralelización de tareas.

**Desventajas:**

- Complejidad creciente.
- Cambios innecesarios. Puede ser que no todas las vistas estén interesadas en los cambios.
- Conexión entre la vista y el controlador. Hay que usar los dos a la vez.
- Acceso ineficiente a los datos en la vista. Puede necesitar varias llamadas al modelo para actualizar todos sus datos.

Tanto la vista como el controlador son específicos de una plataforma. Algunas herramientas de diseño de interfaces de usuario incorporan parte del procesamiento de eventos entrada. El controlador deja de ser necesario (49).



### 1.9 - Herramientas CASE.

La ingeniería de sistemas asistida por ordenador es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo, su objetivo es acelerar el proceso para el que han sido diseñadas, en el caso de CASE (Computer Aided Software Engineering) para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

La tecnología CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información y se plantean los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta se consigue agilizar el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

Automatizar:

- El desarrollo del software.

- La documentación.
- La generación del código.
- El chequeo de errores.
- La gestión del proyecto.

Permitir:

- La reutilización del software.
- La portabilidad del software.
- La estandarización de la documentación. (31)

### **1.9.1 - Rational Rose**

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto:

- Concepción y formalización del modelo.
- Construcción de los componentes.
- Transición a los usuarios.
- Certificación de las distintas fases. (32)

Características Rational Rose:

- Mantiene la consistencia de los modelos del sistema software.
  - Generación Documentación automáticamente.
  - Generación de Código a partir de los Modelos.
  - Ingeniería Inversa (crear modelo a partir código).
  - Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
  - Capacidad de análisis de calidad de código.
  - Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos
  - Capacidad de crear definiciones de tipo de documento XML para el uso en la aplicación.
  - Integración con otras herramientas de desarrollo de Rational.
  - Publicación web y generación de informes para optimizar la comunicación dentro del equipo.
- (33)



### **1.9.2 - Visual Paradigm**

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Es un software no propietario por convirtiéndose en una herramienta fundamental a tener en cuenta a la hora de ser utilizada.

Características:

- Ingeniería de ida y vuelta.
- Soporte de UML.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++, Esquemas XML, XML, .NET.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas de flujo de datos.
- Generación de bases de datos. Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos, desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XML.
- Editor de figuras.
- Herramientas y plugins de modelado UML.(34)

### **1.9.3 - Argo UML**

Es una aplicación de diagramado de UML escrita en Java y publicada bajo la Licencia BSD open source.

Características:

- Características de extensibilidad mejoradas de UML 1.4.

- Diagramas de secuencia.
- La mayoría de las funciones ahora soportan la selección múltiple de los elementos del modelo.
- Soporte de Internacionalización para inglés, alemán, francés, español y ruso.
- Soporte para el lenguaje de generación de Código: Java, PHP, Python, C++ y Csharp (C#).
- Ingeniería inversa.
- Generación de ficheros PNG, GIF, JPG, SVG, EPS desde diagramas.
- Soporte para comentarios para múltiples elementos.
- Todos los diagramas de UML 1.4 están soportados.

Desventaja:

Está escrito en Java por lo que se puede ejecutar en cualquier plataforma que tenga una máquina virtual java, convirtiéndose esto en su mayor problema, ya que al ser el lenguaje Java compilado, se convierte en una aplicación que necesita muchos recursos de la computadora por lo que solo es usable en máquinas muy potentes. (50)

#### **1.9.4 - Poseidón**

Es una herramienta para modelar cualquier clase de sistema que tenga o no que ver con programación.

Licencia: Community Edition y Professional Edition.

Características:

- Soporta diagramas UML.
- Opciones avanzadas de impresión.
- Soporta gráficos en la mayoría de formatos.
- Varios idiomas.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones. (50)

### **1.10 - Herramientas**

#### **1.10.1 - Zend Studio**

Zend Studio 5 se ha diseñado para una amplia gama de programadores y existen dos ediciones: Standard y Professional. Zend Studio 5, concebido con el fin de crear aplicaciones altamente fiables, proporciona una facilidad de uso inigualable, escalabilidad, fiabilidad, y la extensión que los programadores profesionales y de empresas requieren para desarrollar, distribuir, depurar y administrar aplicaciones PHP de negocios.

Características:

- Compatible con las plataformas Linux, MAC y Windows.
- Excelente completamiento de código, coloreado en la sintaxis del código, administración avanzada de proyectos, múltiples lenguajes, incorpora el Framework de Zend, PHP Documentor, manual de PHP. Integración con subversión, los navegadores, integración avanzada con FTP. Soporte para Web Services, PHP4, PHP5 y SQL.
- Agiliza nuestro trabajo, cuenta con un buen Depurador, infinitas opciones que permiten un desarrollo profesional de nuestras aplicaciones. (26)

### **1.10.2 - Fireworks**

Fireworks es una suite de diseño vectorial que agrupa texto, diseño, ilustración, edición de imágenes, URL, JavaScript, y herramientas de animación. Diseñado desde un primer momento para satisfacer las necesidades de los diseñadores Web profesionales, Fireworks incluye opciones tan avanzadas como: exportación de previsualización visual, control total sobre las paletas de color y la compresión de las mismas, generación automática de botones de estado y mensajes en JavaScript, control total sobre textos y efectos que se pueden editar en cualquier momento. (47)

### **1.10.3 - Dreamweaver**

Potente software de creación de Macromedia que permite crear fácilmente sitios que contienen gráficos y elementos multimedia. Es uno de los mejores programas para crear animaciones JavaScript y DHTML.

Características:

Aunque sea un experto programador de HTML el usuario que lo maneje, siempre se encontrarán en este programa razones para utilizarlo, sobretodo en lo que a productividad se refiere.

Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas.
- Javascript para crear efectos e interactividades.
- Inserción de archivos multimedia.

Además es un programa que se puede actualizar con componentes, que fabrica tanto Macromedia como otras compañías, para realizar otras acciones más avanzadas.

El programa es realmente satisfactorio, incluso el código generado es de buena calidad.

Al ser tan avanzado, puede resultar un poco difícil su manejo para personas menos experimentadas en el diseño de webs.

Funciona sobre las plataformas Windows 95/98/Me/2000/XP y es multilinguaje.

Tiene la desventaja de ser un software perteneciente a la empresa Adobe, por lo que posee la característica de ser un software propietario. [25]

### **1.11 - Conclusiones**

Después de un estudio de las tendencias y tecnologías actuales descritas anteriormente, se llegó a la conclusión de que es necesaria la creación de una aplicación web con tecnología cliente-servidor, que ayude a la gestión de la información en el proceso de creación de un software con tecnología multimedia en la facultad 8 de la Universidad de las Ciencias Informáticas.

Para la realización del software se utilizará como lenguaje del lado del servidor PHP, por las ventajas y facilidades de uso antes expuestas. Como Gestor de base de datos está el MySQL Server, el cual posee una gran compatibilidad con el lenguaje PHP, y como servidor web el Apache por su fácil manejo y ser multiplataforma al igual que PHP y MySQL Server.

Se utilizará JavaScript, HTML, XML, y otros lenguajes que son incorporados en la creación de un sitio web, brindando la posibilidad de enriquecer el sistema con las funcionalidades propias cada uno.

Como metodología a utilizar se tiene a RUP por ser una de las más utilizadas a nivel mundial y poseer características que describen el flujo de trabajo y las etapas de desarrollo de un software, esta se encuentra en estrecha relación con el Lenguaje de Modelado Unificado UML, que juntos forman una solución en el proceso de creación del sistema a realizar. Para modelar el sistema antes de proceder a construirlo el Rational Rose proporciona una visión de las principales funcionalidades, actividades y tareas que forman parte del proceso de desarrollo de software.

Como patrón de arquitectura se hará uso del Modelo - Vista - Controlador (MVC) por su integridad, facilidad de mantenimiento del código y ventajas que presenta.

Se utilizarán diferentes herramientas que apoyarán el desarrollo de la aplicación entre las cuales se pueden mencionar el Zend Studio por tener un excelente completamiento de código, el Dreamweaver que constituye un potente software de creación de Macromedia que permite crear fácilmente sitios que contienen gráficos y elementos multimedia y Fireworks que es una suite de diseño vectorial que agrupa texto, diseño, ilustración, edición de imágenes, URL, JavaScript, y herramientas de animación.

## **CAPÍTULO 2 MODELACIÓN DEL NEGOCIO**

### **2.1 - Introducción**

A la hora de comenzar a desarrollar el sistema o la aplicación es necesario comprender como se llevan a cabo actualmente en la Facultad 8 los procesos en cuestión, a fin de lograr una mejor comprensión del problema a resolver y el común entendimiento entre clientes y desarrolladores; para lo cuál se realiza la modelación del negocio.

El modelo del negocio posibilita obtener una visión más clara del proceso en cuestión, por ello en este capítulo se exponen las políticas y condiciones que deben cumplirse, entendidas como reglas del negocio asociadas al campo de acción. Se describen los actores, trabajadores del negocio y el modelo de objetos.

### **2.2 - Modelo del negocio actual**

El cliente solicita al Vicedecano de producción el desarrollo de un software con tecnología multimedia, el Vicedecano de producción analiza la solicitud del cliente y le da una respuesta, en caso de que la respuesta del Vicedecano de producción sea positiva, el mismo aprueba el proyecto y convoca a una reunión donde participarán las personas que estarán implicadas en el desarrollo de la aplicación como son el líder del proyecto, el analista de procesos del negocio, entre otros.

El cliente entrega la información referente al producto al líder del proyecto, esta información puede estar estructurada, en un guión de contenido, en un guión técnico o puede ser solamente una información general sobre el producto.

Si la información brindada por el cliente no viene estructurada en un guión de contenido o en un guión técnico, el líder del proyecto pasa a realizar el plan de proyecto y el listado de riesgos, el analista de procesos del negocio confecciona el listado de requisitos, se le presenta al guionista los datos necesarios del software con tecnología multimedia y comienza la confección del guión de contenido, el resultado obtenido se le presenta al cliente y este aprueba o no el guión de contenido confeccionado, en caso de no ser aprobado dicho guión de contenido se pasa nuevamente a la confección del mismo, sino se le entrega al arquitecto de la información que será el responsable de realizar el guión técnico y se le entrega además al diseñador que será el responsable de elaborar el diseño. Si el cliente trae la información en un guión de contenido, se realiza por parte del líder de proyecto el plan de proyecto y el listado de riesgos, este guión de contenido pasa directamente al arquitecto de la información y al diseñador para realizar el guión técnico y el diseño respectivamente.

Si el cliente trae la información estructurada en un guión técnico se pasa a realizar el plan de proyecto y el listado de riesgos por parte del líder del proyecto.

Con el guión técnico realizado y el diseño de la aplicación se precede a la entrega de dicha información a los programadores, los que comienzan a interactuar con la misma y deciden el modo de almacenamiento de la información con la que cuenta el software.

Los programadores con la información almacenada y el diseño realizado inician la implementación del sistema. Se obtiene como resultado una serie de hitos que se le aplican una serie de pruebas para verificar la calidad del software, se presenta al cliente los resultados y este aprueba o no el trabajo desarrollado

## **2.3 - Descripción de los procesos del negocio**

### **2.3.1 - Aprobar proyecto**

El proceso comienza cuando el cliente solicita la creación del software con tecnología multimedia, entrega la información inicial y comienza con la descripción de las características principales de la multimedia, el Vicedecano de producción se pone de acuerdo con el cliente, y se obtiene como resultado que el proyecto sea aprobado o no.

### **2.3.2 - Unificar información**

Este proceso comienza después de haber sido aprobado el desarrollo del software con tecnología multimedia. El líder del proyecto se reúne con el cliente y algunas de las personas que forman parte del equipo de trabajo para obtener la información necesaria para la confección del software. Esta información puede o no estar recogida en un guión de contenido o un guión técnico.

### **2.3.3 - Confeccionar guión de contenido**

Este proceso comienza cuando el guionista, se encarga de confeccionar el guión de contenido con la información brindada en el proceso Unificar Información. El resultado de este proceso es el guión de contenido listo para ser revisado por el cliente.

### **2.3.4 - Aprobar guión de contenido**

Este proceso comienza cuando el guión de contenido es entregado al cliente. En este proceso el cliente se encarga de revisar el guión de contenido.

En el caso que el cliente esté de acuerdo con la información estructurada en el guión de contenido, se aprueba el mismo, sino se pasa nuevamente al paso de rectificar el guión de contenido.

### **2.3.5 - Confeccionar guión técnico**

El proceso confeccionar guión técnico comienza después de haber sido aprobado el guión de contenido, el cual es entregado al arquitecto de la información que es el responsable de confeccionar el guión técnico. En este proceso se obtiene como resultado guión técnico.

## 2.4 - Reglas del Negocio

- Existencia y disponibilidad de recursos tecnológicos.
- El perfil del proyecto a realizar debe estar en concordancia con el de la facultad.
- Un líder de proyecto no puede participar en más de un proyecto al mismo tiempo.
- Un analista de procesos del negocio no puede participar en más de un proyecto al mismo tiempo.
- Un guionista no puede participar en más de un proyecto al mismo tiempo.
- Un arquitecto de la información no puede participar en más de un proyecto al mismo tiempo.

## 2.5 - Entidades del Negocio:

- Plan de proyecto.
- Lista de riesgos.
- Listado de requisitos.
- Guión de contenido.
- Guión técnico.

## 2.6 - Actores del negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. De acuerdo con esta idea un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor.

Actores del Negocio	Descripción
Cliente	Persona que solicita la creación del software con tecnología multimedia, es el responsable de brindar información necesaria para el desarrollo del software.

## 2.7 - Trabajadores del negocio

Un trabajador define el comportamiento y las responsabilidades de un individuo que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio.

Trabajadores del Negocio	Descripción
Vicedecano de producción.	Persona responsable de aprobar el desarrollo del software y garantizar la calidad del producto obtenido.
Líder de proyecto.	Persona que recopila la información necesaria para el desarrollo del software, dirige el equipo de trabajo, confecciona el plan de proyecto y el listado de riesgos.
Analista de procesos del negocio.	Persona encargada de realizar el listado de requisitos.
Guionista.	Persona que confecciona el guión de contenido del producto.
Arquitecto de la información.	Persona que se encarga de estructurar la información en el guión técnico.

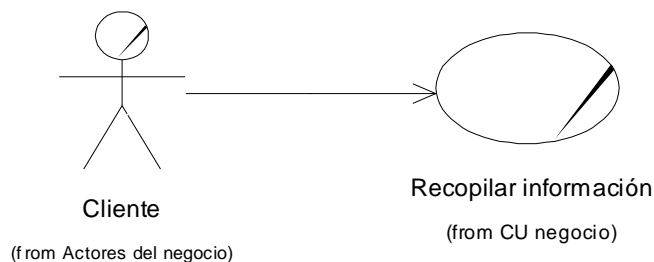
## 2.8 - Casos de uso del negocio

Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables.

### 2.8.1 - Caso de uso del negocio

- Recopilar información.

## 2.9 - Modelo de casos de uso del negocio





## 2.10 - Descripción de los casos de uso del negocio

Descripción del caso de uso “**Recopilar Información**”

<b>Nombre del caso de uso del negocio:</b>	Recopilar información.
<b>Actores del negocio:</b>	Cliente.
<b>Propósito:</b>	Obtener la información.
<b>Resumen:</b> El caso de uso se inicia cuando el cliente solicita al Vicedecano de producción la realización de un software con tecnología multimedia, en este caso el cliente brinda información que es recibida por el líder de proyecto, el cuál en conjunto con otras personas del equipo de desarrollo recopilarán la información del software a realizar.	
<b>Casos de uso asociados:</b>	
<b>Flujo de trabajo</b>	
<b>Acción del actor</b>	<b>Respuesta del negocio</b>
<p><b>1-</b> El cliente solicita el desarrollo de un software con tecnología multimedia y brinda las características del producto que desea obtener.</p> <p><b>3-</b> El cliente entrega información referente al producto, así como contenido específico de la multimedia como son los textos y las medias.</p> <p><b>8-</b> Se realiza una revisión del guión de contenido por parte del cliente.</p> <p><b>9-</b> El cliente acepta el guión de contenido.</p>	<p><b>2-</b> El Vicedecano de producción acepta la solicitud, le informa al cliente que se realizará la aplicación.</p> <p><b>4-</b> El líder del proyecto recoge toda la información del producto.</p> <p><b>5-</b> El líder del proyecto realiza el plan de proyecto y el listado de riesgos.</p> <p><b>6-</b> El analista de procesos del negocio realiza el listado de requisitos.</p> <p><b>7-</b> El guionista confecciona el guión de contenido.</p> <p><b>10-</b> El arquitecto de la información recibe el guión de contenido.</p> <p><b>11-</b> El arquitecto de la información realiza el guión técnico.</p>
<b>Prioridad:</b>	Alto

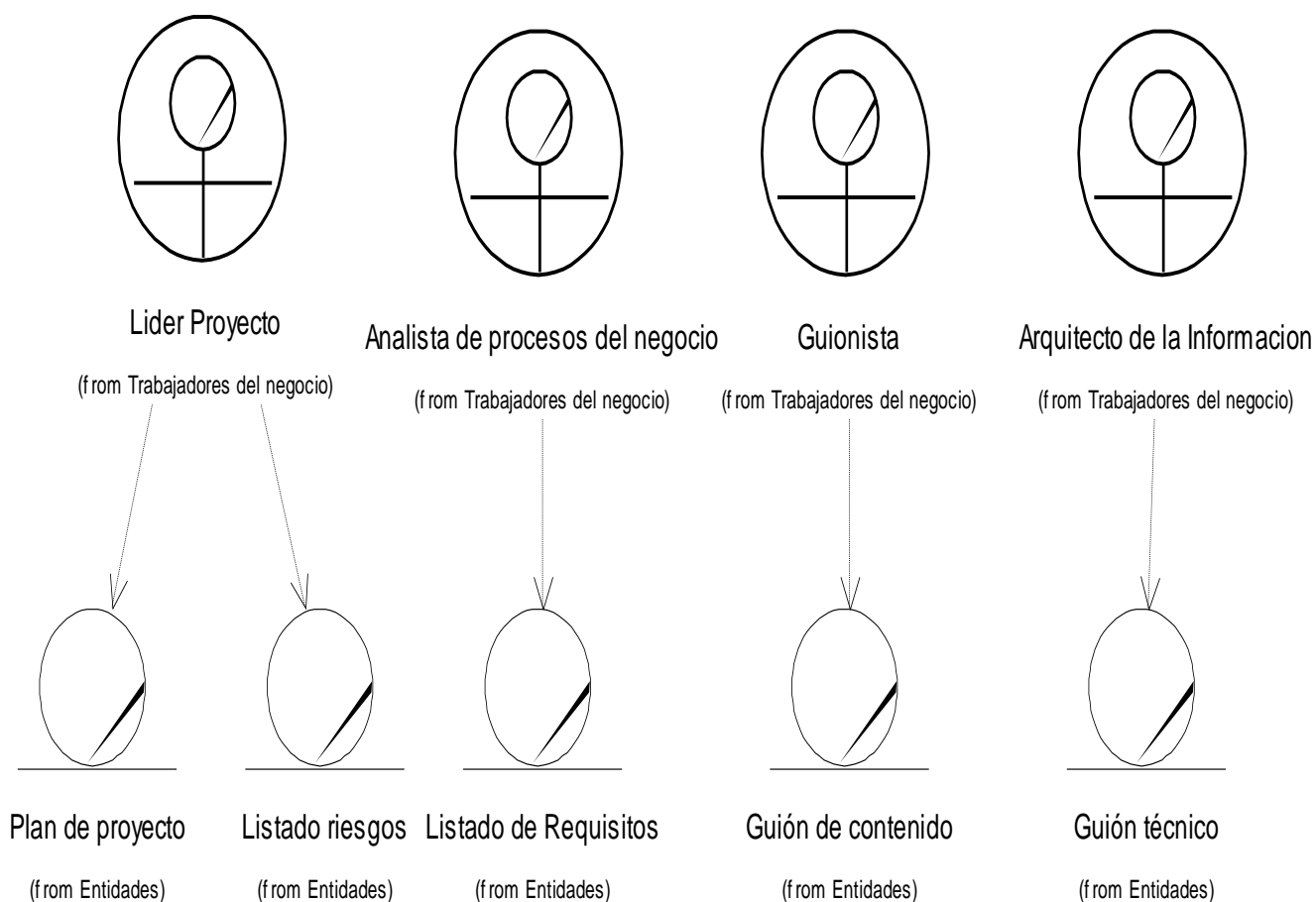
<b>Mejoras:</b>	
<b>Cursos alternos:</b>	
<b>Acción del actor</b>	
<p><b>3.1-</b> El cliente entrega la información en un guión de contenido. Se realiza el paso 5. Ir a paso 10.</p> <p><b>3.2-</b> El cliente entrega la información en un guión técnico. Se realiza paso 5 y finaliza el CUN.</p> <p><b>9.1-</b> El cliente informa que no esta de acuerdo con la propuesta de guión de contenido. Ir al paso 7.</p>	<p><b>2.1-</b> El Vicedecano de producción no acepta la solicitud e informa al cliente.</p>

### 2.11 - Diagrama de Actividades (Anexo 1)

Un diagrama de actividad demuestra la serie de actividades que deben ser realizadas en un proceso del negocio, así como las distintas rutas que pueden irse desencadenando. Este es dividido en calles, donde cada calle representa el trabajador que está llevando a cabo la actividad y muestra cómo se utilizan las entidades del negocio.

### 2.12 - Diagrama de clases del modelo de objetos

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe cómo cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo. [40]



## 2.13 - Requerimientos

### 2.13.1 - Requisitos funcionales

Los requisitos funcionales, son capacidades o condiciones que el sistema debe cumplir, definen qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas.

**RF1:** Administrar usuario.

**1.1** - Administrar usuario administrador de sistema.

**1.2** - Administrar usuario Vicedecano de producción.

**1.3** - Administrar usuario líder de proyecto.

**1.4** - Administrar usuario documentador.

**RF2:** Autenticar usuario.

**RF3:** Administrar proyecto.

**3.1** - Crear proyecto.

**3.2** - Actualizar datos de proyecto.

**3.3** - Eliminar proyecto.

**3.4** – Mostrar Datos de Proyecto.

**RF4:** Obtener reporte del estado del proyecto.

**RF5:** Administrar módulos.

**RF6:** Administrar información del módulo Temas.

**RF7:** Administrar información del módulo Juegos.

**RF8:** Administrar información del módulo Galería.

**RF9:** Estructurar contenido.

### **2.13.2 - Requerimientos no funcionales**

#### **2.13.2.1 - Interfaz Externa**

- Debe contar con una interfaz amigable y de fácil manejo para los usuarios.
- Ajustarse a los estándares de diseño Web.
- Debe verse en cualquier resolución de pantalla de forma adecuada.
- Contar con una gran interactividad.

#### **2.13.2.2 - Usabilidad**

- El texto deberá tener un tamaño adecuado para ser visto a una distancia de un metro en la resolución de 800x600.

#### **2.13.2.3 - Rendimiento**

- El sistema debe ser capaz de procurar una respuesta consistente a sus usuarios para una determinada acción en un tiempo mínimo (fracciones de segundo).
- Debe tener una gran capacidad de recuperación ante posibles fallas.
- Debe permanecer funcionando las 24 horas como aplicación distribuida sobre la Web.

#### **2.13.2.4 - Soporte**

- Fácil mantenimiento y de configuración sencilla.

#### **2.13.2.5 - Seguridad**

- Se podrá acceder al sistema solamente después de autenticarse.
- Contar con un sistema de permisos y usuarios para el acceso a la información.
- Chequeo de seguridad sobre las operaciones no reversibles (insertar modificar).
- Tiene que ser capaz de anular cualquier acción incorrecta que atente contra la integridad de los datos.

### 2.13.2.6 - Legales

- La plataforma sobre la que se va a ejecutar el sistema esté bajo la licencia de software libre.

### 2.13.2.7 - Confiabilidad

- Los reportes que se obtendrán deben ser 100% precisos y reales.
- Garantiza un control estricto sobre el tráfico de información.
- Chequeo constante de la integridad y consistencia en los datos referenciados.

### 2.13.2.8 - Software

- El cliente debe tener uno de los siguientes navegadores: Mozilla, Netscape (versión 7.1 o superior), Microsoft Internet Explorer (para Windows 5.0 o superior),
- Para el servidor Web: Apache (versión 2.0.55 o superior) con las librerías para ejecutar PHP (versión 5.1.4).

### 2.13.2.9 - Hardware

- Tarjeta de red.
- Para el servidor Web: 1 GB RAM.
- Capacidad de disco duro, preferiblemente mayor a los 80 GB.

## 2.14 - Actores del sistema

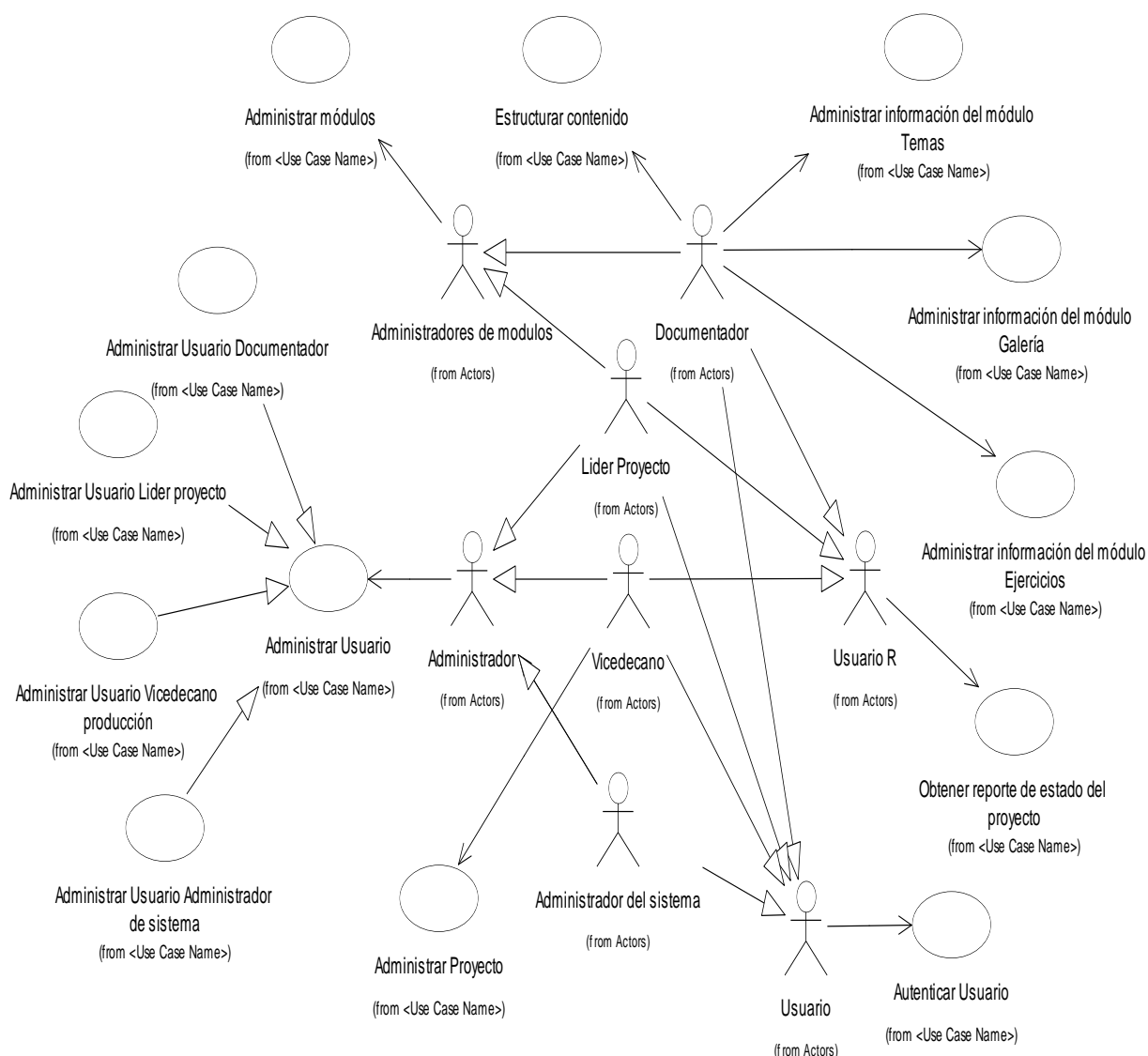
Los actores del sistema definen el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que interactúan con el mismo intercambiando información con este.

Actores del sistema	Descripción
Administrador del sistema.	Persona encargada de administrar la aplicación y brindar acceso a las personas que desarrollarán nuevas versiones del producto.
Vicedecano de producción.	Persona encargada de gestionar los datos de los proyectos que se realizan en la facultad.
Líder de proyecto.	Persona encargada de registrar la información de las personas que integran el equipo de desarrollo del proyecto del cual es líder, así como la verificación y control del estado de desarrollo del proyecto.

<p>Documentador.</p>	<p>Persona encargada de introducir la información de los guiones técnicos en el sistema para la obtención de los ficheros en formato XML.</p>
----------------------	---

### 2.15 - Diagrama de casos de uso del sistema

Los Casos de Uso del sistema (CUS) son un conjunto de secuencia de acciones que un sistema ejecuta y que produce un resultado observable para un actor. Con otras palabras, son “fragmentos” de funcionalidad que el sistema ofrece a los actores que interactúan con el mismo, reportándoles algunos beneficios.



## 2.16 - Descripción de los casos de uso del sistema

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

### 2.16.1 - Descripción del CUS “Administrar usuarios” (Anexo 2)

### 2.16.2 - Descripción del CUS “Autenticar usuarios” (Anexo 2)

### 2.16.3 - Descripción del CUS “Administrar Proyecto”

<b>Nombre del CU</b>	Administrar proyecto
<b>Actores</b>	Vicedecano de producción(Inicia)
<b>Resumen</b>	El caso de uso se inicia cuando el Vicedecano de producción decide crear, actualizar, mostrar datos o eliminar proyecto.
<b>Referencias</b>	RF3
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
El Vicedecano de producción selecciona la opción de Gestionar Proyecto.	El sistema muestra al Vicedecano de producción las posibles acciones a realizar. <b>Crear proyecto.</b> <b>Actualizar proyecto.</b> <b>Eliminar proyecto.</b> <b>Mostrar datos de proyecto.</b>
<b>Sección “Crear proyecto”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El Vicedecano de producción selecciona la opción de crear proyecto.	2. El sistema muestra una interfaz por la cual se introducirán los datos.
3. El Vicedecano de Producción entra datos del proyecto y presiona el botón insertar.	4. El sistema verifica que los campos: tema del proyecto, nombre del cliente, nombre de la empresa a la que se le realiza el contrato, observaciones, tipo de multimedia (informativo o educativo),

	teléfono de cliente, correo electrónico de cliente, fecha de inicio y fecha fin estén llenos.
	5. El sistema verifica en la base de datos que el proyecto no esté registrado.
	6. Los datos del proyecto se almacenan en el sistema.
<b>Sección “Crear Proyecto”: Flujo Alternativo</b>	
2.1- El Vicedecano de producción cancela la acción y se termina el CU.	4.1 Si los campos no están llenos se emite un mensaje y se remite al paso 2.
	5.1 Si el proyecto está registrado se emite un mensaje de error y se remite al paso 2.
<b>Sección “Actualizar Proyecto”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El Vicedecano de producción selecciona la opción actualizar proyecto.	2. El sistema muestra lo proyectos que están en la base de datos.
3. El Vicedecano de producción selecciona el proyecto a actualizar.	
4. El Vicedecano de producción realiza las modificaciones deseadas.	
5. El Vicedecano de producción presiona el botón actualizar.	6. Se verifica que los campos: nombre del proyecto, tema del proyecto, nombre del cliente, nombre de la empresa a la que se le realiza el contrato, tipo de multimedia (informativo o educativo), fecha de inicio y fecha fin estén llenos.
	7. Se actualiza la información.
<b>Sección “Actualizar proyecto”: Flujo Alternativo</b>	
3.1. El Vicedecano de producción cancela la acción y se termina el CU.	6.1. Se emite un mensaje para que llene todos los campos y se remite al paso 5.
<b>Sección “Mostrar datos de proyecto”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>



1. El Vicedecano de producción selecciona la opción mostrar datos de proyecto.	2. El sistema carga una interfaz donde se muestran los nombres de los proyectos registrados en la base de datos.
3. El Vicedecano de producción selecciona el proyecto del cuál desea obtener los datos y presiona el botón “Ver datos de proyecto”.	4. El sistema muestra en una interfaz con los datos del proyecto seleccionado.
<b>Sección “Mostrar datos de proyecto”: Flujo Alternativo</b>	
3.1- El Vicedecano de producción cancela la acción y se termina el CU.	
<b>Sección “Eliminar proyecto”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El Vicedecano de producción selecciona la opción eliminar proyecto.	2. El sistema muestra los nombres del los proyectos registrados en la base de datos.
3. El Vicedecano de producción selecciona el proyecto a eliminar.	4. El sistema muestra un mensaje para confirmar si desea eliminar el proyecto.
	5. Se eliminan lo datos del proyecto.
<b>Sección “Eliminar proyecto”: Flujo Alternativo</b>	
3.1. El Vicedecano de producción cancela la acción y se termina el CU.	4.1. Cancela la acción y se remite al paso 2.
<b>Precondiciones</b>	El Vicedecano de producción debe estar previamente autenticado.
<b>Poscondiciones</b>	Se crea, actualiza, muestra o elimina un proyecto.
<b>Prioridad:</b>	Alta

#### 2.16.4 - Descripción del CUS “Obtener reporte del estado del proyecto”

<b>Nombre del CU</b>	Obtener reporte del estado del proyecto.
<b>Actores</b>	Usuario(Inicia)
<b>Resumen</b>	El CU se inicia cuando el usuario interactúa con el sistema para obtener un reporte de la situación en que se encuentra un proyecto y finaliza cuando se muestran los datos del reporte.
<b>Referencias</b>	RF 4

<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción Obtener Reporte.	2. Si el usuario que interactúa con el sistema es el Vicedecano de Producción el sistema muestra una interfaz con los nombres de los proyectos existentes en la base de datos para obtener un reporte actual de la situación del proyecto.
3. El Vicedecano de Producción selecciona el proyecto del que desea obtener el reporte.	4. El sistema muestra el reporte del proyecto.
<b>Curso Alternativo de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
3.1- Presiona botón cancelar.	2.1 Si el usuario es líder de proyecto o documentador, el sistema muestra una interfaz donde el usuario puede obtener reporte actual del proyecto al que pertenece, se ejecuta el paso 4.
<b>Precondiciones</b>	El usuario debe estar previamente autenticado.
<b>Poscondiciones</b>	Se muestran los datos del reporte.
<b>Prioridad</b>	Media.

### 2.16.5 - Descripción del CUS “Administrar módulos”

<b>Nombre del CU</b>	Administrar módulos
<b>Actores</b>	Administrador de módulos (Inicia)
<b>Resumen</b>	El CU se inicia cuando el administrador de módulos interactúa con el sistema para determinar la cantidad de información que tendrá la multimedia.
<b>Referencias</b>	RF 5
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

1. El administrador de módulos selecciona la opción administrar módulos Temas, Galería o Ejercicios.	2. El sistema muestra al actor una interfaz donde podrá insertar la información del plan que tendrán los módulos de la multimedia.
3. El administrador de módulos escribe los datos del módulo y presiona el botón insertar.	4. El sistema verifica que todos los campos estén llenos.
	5. El sistema almacena la información especificada del módulo en cuestión.
<b>Curso Alterno de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1.1. Presiona botón cancelar y termina el caso de uso.	4.1. Si los campos no están llenos se emite un mensaje de error y se remite al paso 2.
<b>Precondiciones</b>	El usuario debe estar previamente autenticado.
<b>Poscondiciones</b>	Se almacenan los datos específicos de los módulos.
<b>Prioridad</b>	Alta

### 2.16.6 - Descripción del CUS “Administrar información del módulo Temas”

<b>Nombre del CU</b>	Administrar información del módulo Temas
<b>Actores</b>	Documentador(Inicia)
<b>Resumen</b>	El caso de uso se inicia cuando el Documentador interactúa con el sistema y decide administrar la información del módulo Temas.
<b>Referencias</b>	RF6
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
El documentador selecciona la opción de Administrar Módulo Temas.	El sistema muestra al documentador las posibles acciones a realizar. <b>Registrar información.</b> <b>Eliminar información.</b>
<b>Sección “Registrar información”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

1. El documentador selecciona la opción de registrar información de módulo Temas.	2. El sistema muestra una interfaz donde especifica los temas a los cuales se le insertará información.
3. El documentador escoge el tema al cual se insertará información, introduce el texto, y da clic en el botón insertar.	4. El sistema almacena el texto correspondiente al tema.
5. El documentador da clic en el botón adicionar imágenes al tema.	6. El sistema carga una interfaz que brinda la posibilidad de insertar imágenes al tema así como su descripción.
7. El documentador carga la imagen a insertar y presiona el botón para subir la imagen.	8. El sistema verifica que no se exceda la cantidad de imágenes especificadas en la administración para el tema.
9. Se introduce la descripción correspondiente a la imagen y da clic en el botón enviar.	10. El sistema verifica que no se exceda la cantidad de descripciones especificadas en la administración para el tema.
	11. El sistema muestra la imagen insertada.
<b>Sección “Registrar información”: Flujo Alternativo</b>	
3.1. El documentador cancela la acción y se termina el CU. 3.2. En caso de campos vacíos se remite al paso 3.	8.1 Si el tema tiene todas las imágenes insertadas se muestra un mensaje para especificar que al tema se le insertaron todas las imágenes.
	10.1 Si el tema tiene todas las descripciones de imágenes insertadas se muestra un mensaje para especificar que al tema se le insertaron todas las descripciones.
<b>Sección “Eliminar información”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El documentador selecciona la opción eliminar información del módulo Temas.	2. El sistema muestra una interfaz donde muestra el nombre de los temas que pertenecen al proyecto.

3. El documentador selecciona el tema que será eliminado.	4. Se muestra un mensaje para confirmar la eliminación del tema.
.	5. Se elimina de la base de datos la información del tema seleccionado.
<b>Sección “Eliminar información”: Flujo Alternativo</b>	
3.1. El documentador cancela la acción y se termina el CU.	4.1. El documentador cancela la acción y pasa al paso 2.
<b>Precondiciones</b>	El documentador debe estar previamente autenticado.
<b>Poscondiciones</b>	Se registra, actualiza o elimina la información del módulo Temas
<b>Prioridad:</b>	Alta

### 2.16.7 - Descripción del CUS “Administrar información del módulo Galería”

<b>Nombre del CU</b>	Administrar información del módulo Galería
<b>Actores</b>	Documentador(Inicia)
<b>Resumen</b>	El caso de uso se inicia cuando el documentador interactúa con el sistema y decide administrar la información del modulo Galería.
<b>Referencias</b>	RF7
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
El documentador selecciona la opción de administrar módulo Galería.	El sistema permite al documentador realizar las posibles acciones como: <b>Registrar información.</b> <b>Eliminar información.</b>
<b>Sección “Registrar información”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El documentador selecciona la opción de registrar información de módulo Galería que puede ser de imágenes, videos o las descripciones correspondientes a las medias.	2. El sistema muestra una interfaz donde brinda la posibilidad de subir la información.

3. El documentador carga la información a guardar y presiona el botón enviar.	4. El sistema verifica en la base de datos que la cantidad de información almacenada, (imágenes o videos) no exceda la especificada en el caso de uso administrar módulos.
5. El documentador inserta la descripción asociada a la información almacenada y da clic en el botón enviar.	6. El sistema verifica en la base de datos que la cantidad de información almacenada (descripciones) no exceda la especificada en el caso de uso administrar módulos.
	7. El sistema almacena la información introducida.
<b>Sección “Registrar información”: Flujo Alternativo</b>	
3.1- El documentador cancela la acción y se termina el CU.	4.1 En caso de información completa muestra un mensaje de módulo lleno y finaliza caso de uso.
	6.1 En caso de información completa muestra un mensaje de módulo lleno y finaliza caso de uso.
<b>Sección “Eliminar información”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El documentador selecciona la opción eliminar información del módulo Galería.	2. Se muestra un mensaje para confirmar la eliminación de la información.
	3. Se elimina de la base de datos la información seleccionada de la galería.
<b>Sección “Eliminar información”: Flujo Alternativo</b>	
3.1. El documentador cancela la acción y se termina el CU.	4.1. El documentador cancela la acción y se remite al paso 2.
<b>Precondiciones</b>	El documentador debe estar previamente autenticado.
<b>Poscondiciones</b>	Se registra, actualiza o elimina la información del módulo Galería.
<b>Prioridad:</b>	Alta

### 2.16.8 - Descripción del CUS “Administrar información del módulo Ejercicios”

<b>Nombre del CU</b>	Administrar información del módulo Ejercicios.
<b>Actores</b>	Documentador(Inicia)
<b>Resumen</b>	El caso de uso se inicia cuando el documentador interactúa con el sistema y decide administrar la información del módulo Ejercicios.
<b>Referencias</b>	RF8
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
El documentador selecciona la opción de Administrar Módulo Ejercicios.	El sistema muestra al documentador las posibles acciones a realizar. <b>Registrar información.</b> <b>Eliminar información.</b>
<b>Sección “Registrar información”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El documentador selecciona la opción de registrar información de módulo Ejercicios.	2. El sistema muestra una interfaz donde especifica los temas a los cuales se le insertarán Ejercicios.
3. El documentador escoge el tema al cual se le insertarán ejercicios.	4. El sistema muestra los ejercicios pertenecientes al tema.
5. El documentador escoge el ejercicio al cual se le insertarán las actividades.	6. El sistema muestra una serie de campos por donde se insertarán la información de las actividades.
7. El documentador introduce los datos de las actividades y presiona el botón insertar.	8. El sistema verifica que los campos estén llenos.
	9. El sistema almacena la información introducida.
<b>Sección “Registrar información”: Flujo Alternativo</b>	
3.1- El documentador cancela la acción y se termina el CU.	8.1 En caso que lo campos no estén llenos se muestra un mensaje de campos vacios.

<b>Sección “Eliminar información”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El documentador selecciona la opción eliminar información del módulo Ejercicios.	2. El sistema muestra el nombre de los temas a los cuales se le eliminarán ejercicios.
3. El documentador selecciona el tema para eliminarle ejercicios y presiona un botón para ver los ejercicios asociados al tema.	4. El sistema muestra los ejercicios asociados al tema.
5. El documentador escoge el ejercicio a eliminar y presiona el botón Eliminar.	6. Se muestra un mensaje de verificación de eliminación
	7. Se elimina el ejercicio.
<b>Sección “Eliminar información”: Flujo Alternativo</b>	
3.1. El documentador cancela la acción y se termina el CU.	6.1. El documentador cancela la acción y pasa al paso 2.
<b>Precondiciones</b>	El documentador debe estar previamente autenticado.
<b>Poscondiciones</b>	Se registra, actualiza o elimina la información del módulo Ejercicios.
<b>Prioridad:</b>	Media

### 2.16.9 - Descripción del CUS “Estructurar contenido”

<b>Nombre del CU</b>	Estructurar contenido
<b>Actores</b>	Documentador (Inicia)
<b>Resumen</b>	El CU se inicia cuando el documentador interactúa con el sistema para estructurar la información almacenada en la base de datos.
<b>Referencias</b>	RF 9
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El documentador selecciona la opción estructurar contenido.	2. Se estructura la información.
<b>Curso Alternativo de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
<b>Precondiciones</b>	El documentador debe estar previamente



	autenticado.
<b>Poscondiciones</b>	Se obtiene la información estructurada.
<b>Prioridad</b>	Media.

### 2.17 - Conclusiones

En este capítulo se inició el desarrollo de la propuesta de solución, donde se comenzó el análisis de los procesos del negocio, obteniéndose un listado con las principales funcionalidades que debe tener la aplicación. Se presentó el Diagrama de Casos de Uso del Sistema, y por último se realizaron las descripciones de las acciones de los actores que intervienen en el sistema con los casos de uso con los que interactúan.

## CAPÍTULO 3 ANÁLISIS Y DISEÑO

### 3.1 - Introducción

Luego de definir y describir en el capítulo anterior las principales funcionalidades que va a tener la aplicación, se hace imprescindible saber como se desarrollará el sistema. En este capítulo se quiere obtener el análisis y diseño del sistema propuesto. Se presentan los diagramas de clases de análisis y diagramas de diseño Web que describen la interacción entre las distintas páginas.

### 3.2 - Análisis

En el análisis se refinan y estructuran los requisitos obtenidos con anterioridad, profundizando el equipo del proyecto en el dominio de la aplicación lo que les permitirá una mayor comprensión del problema para modelar la solución.

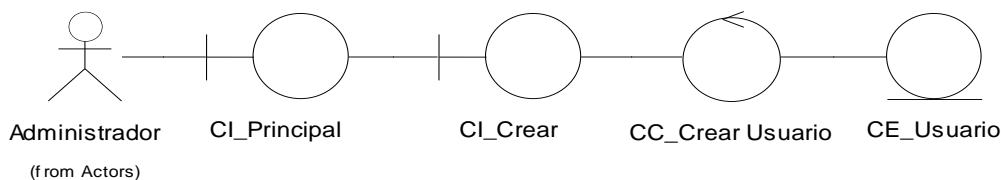
#### 3.2.1 - Modelo de clases de análisis

En la construcción del modelo de análisis se tuvo que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construyó el Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes, proporcionando la estructura a la vista interna y sirvió como una primera aproximación del diseño.

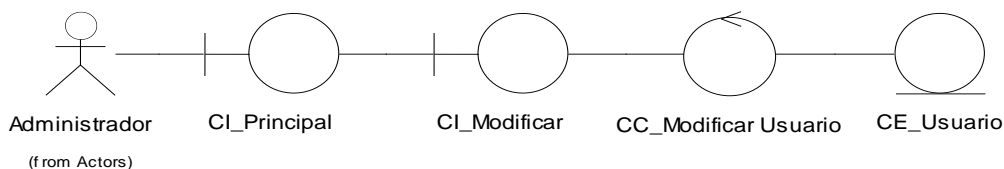
#### 3.2.2 - Diagrama de clases del análisis

##### 3.2.2.1 - Diagramas de clases de análisis de Gestionar Usuarios

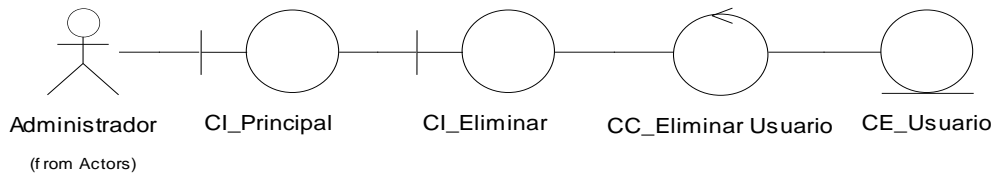
- Diagrama de clases de análisis del Caso Crear Usuarios



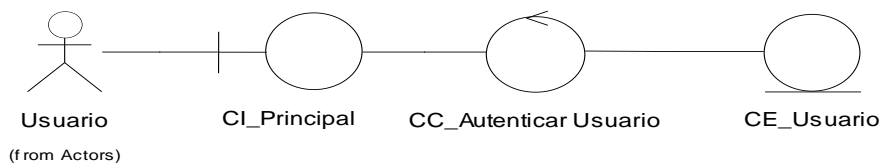
- Diagrama de clases de análisis del Caso Modificar Usuarios



- **Diagrama de clases de análisis del Caso Eliminar Usuarios**

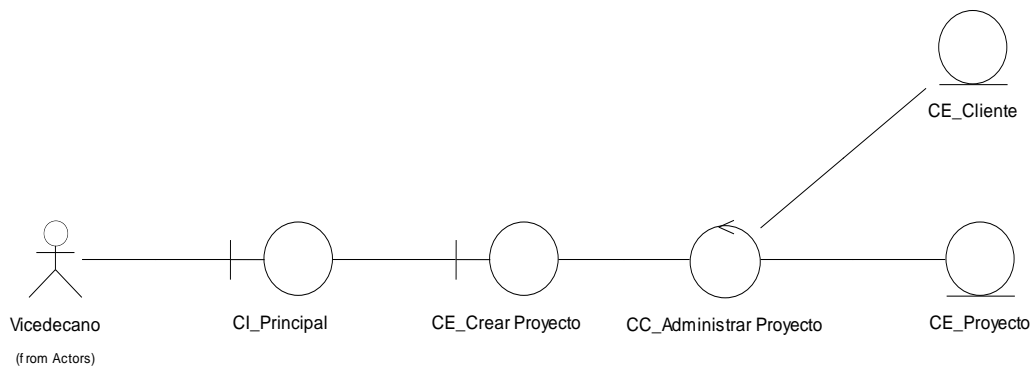


### 3.2.2.2 - Diagramas de clases de análisis de Autenticar Usuarios

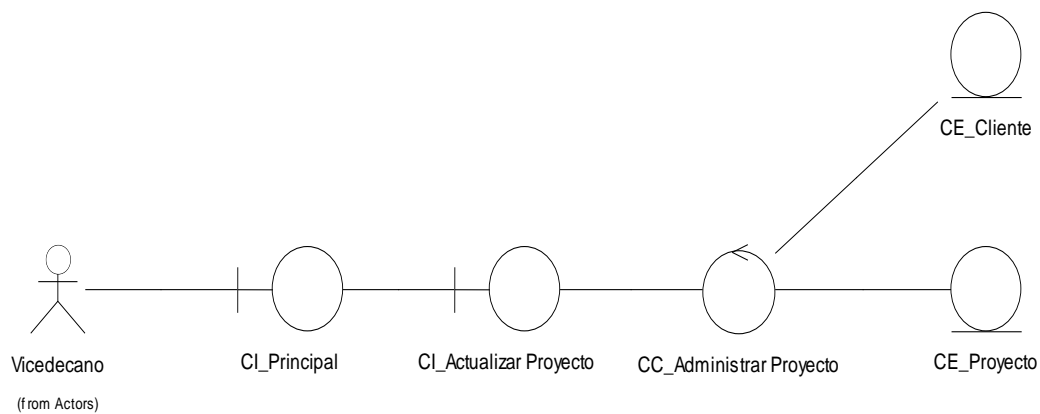


### 3.2.2.3 - Diagramas de clases de análisis de Gestionar proyecto

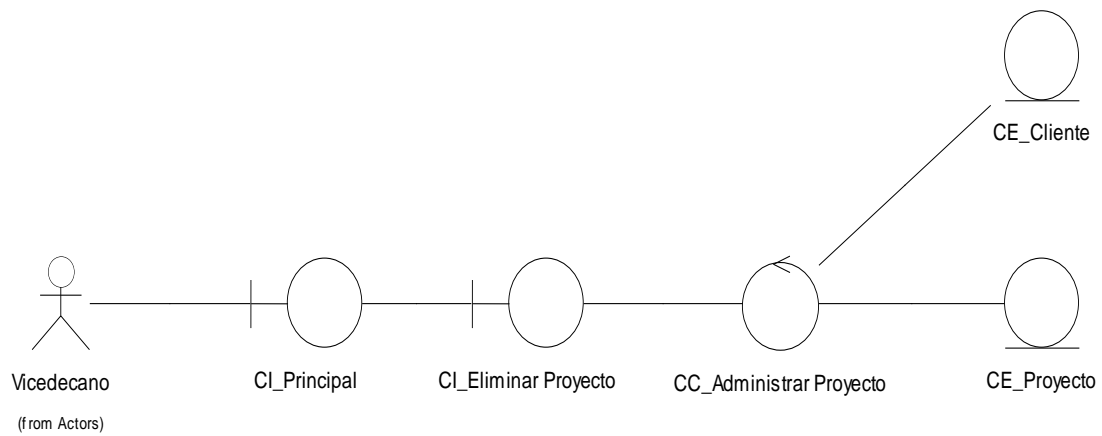
- **Diagrama de clases de análisis del Caso Crear Proyecto**



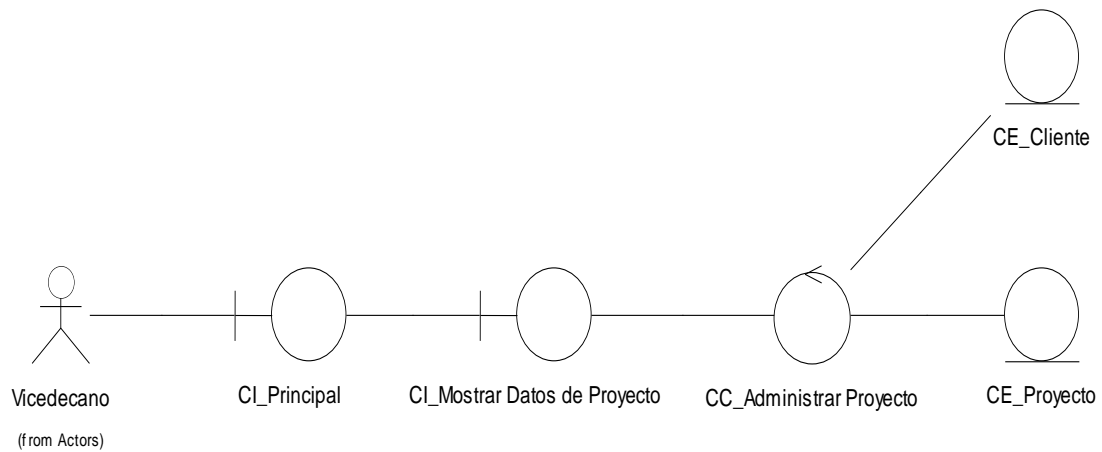
- **Diagrama de clases de análisis del Caso Actualizar Datos de Proyecto**



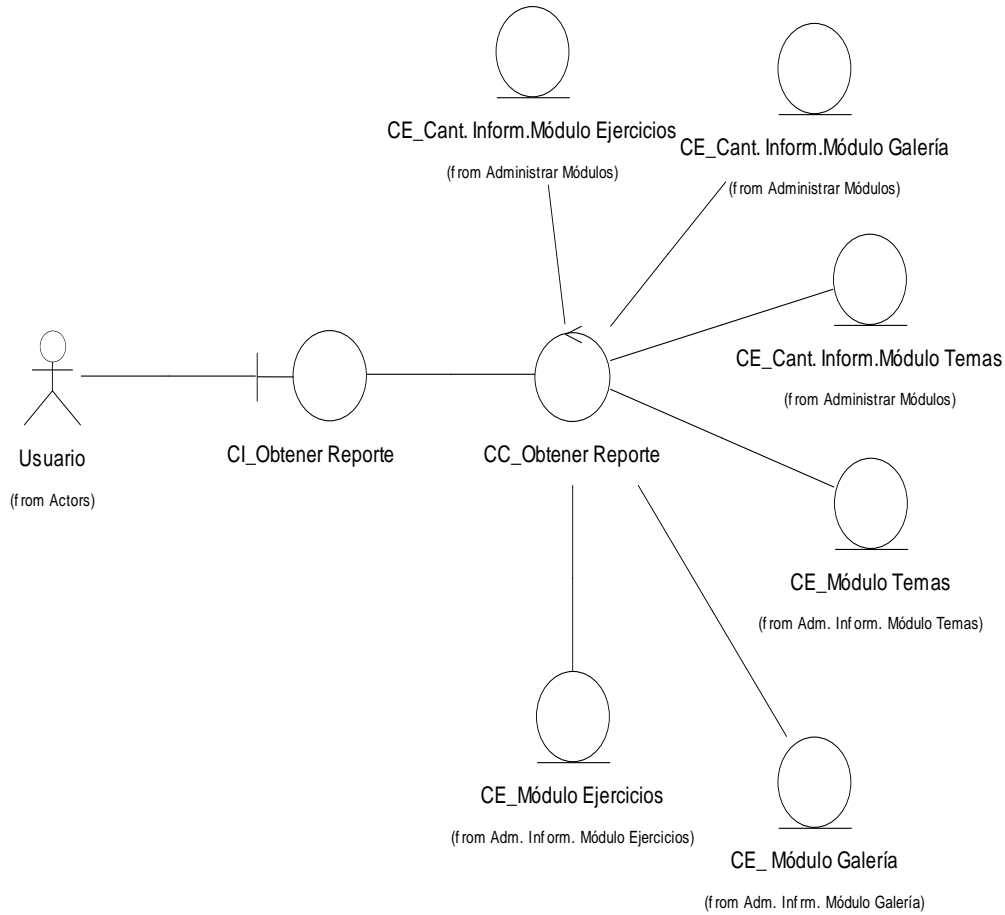
- **Diagrama de clases de análisis del Caso Eliminar Proyecto**



- **Diagrama de clases de análisis del Caso Mostrar Datos Proyecto**

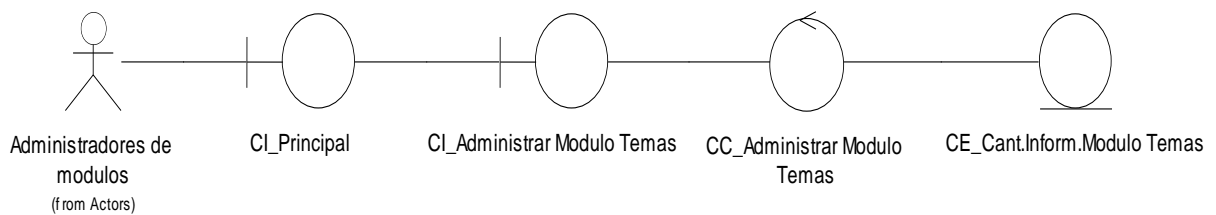


### 3.2.2.4 - Diagramas de clases de análisis de Obtener Reporte

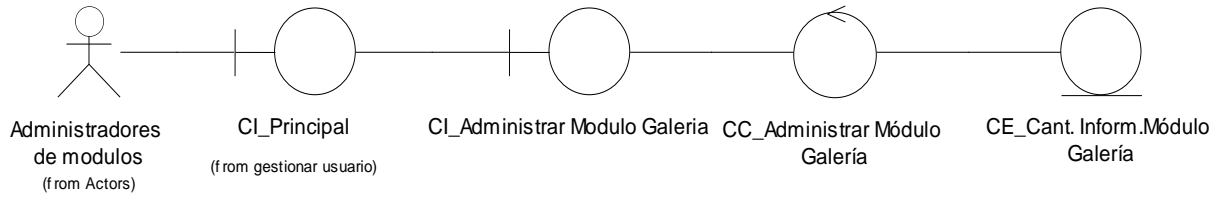


### 3.2.2.5 - Diagramas de clases de análisis de Administrar Módulos

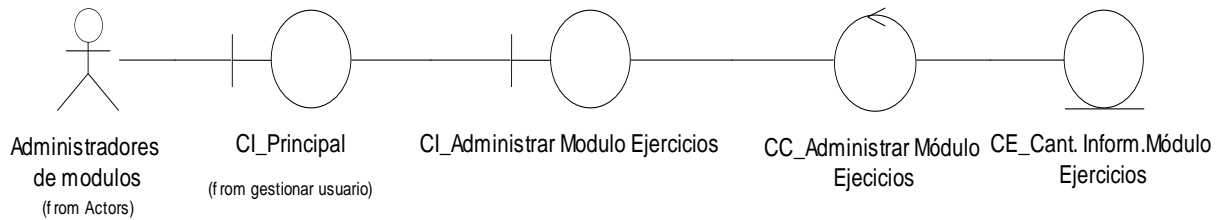
- **Diagrama de clases de análisis del Caso Administrar Módulo Temas**



• Diagrama de clases de análisis del Caso Administrar Módulo Galería

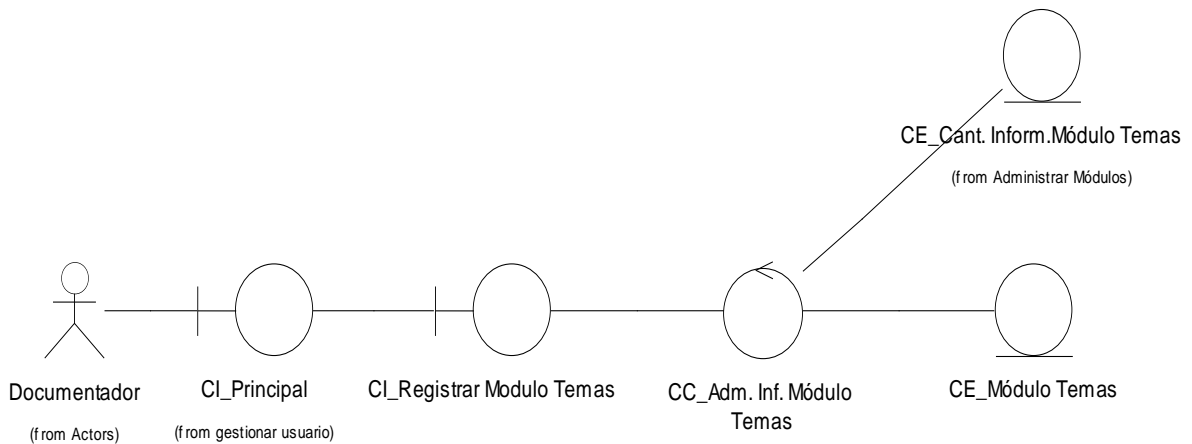


• Diagrama de clases de análisis del Caso Administrar Módulo Ejercicios

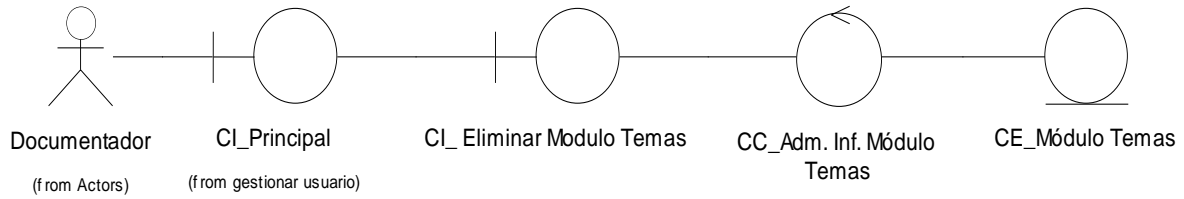


3.2.2.6 - Diagramas de clases de análisis de Administrar Información del Módulo Temas

• Diagrama de clases de análisis del Caso Registrar Información del Módulo Temas

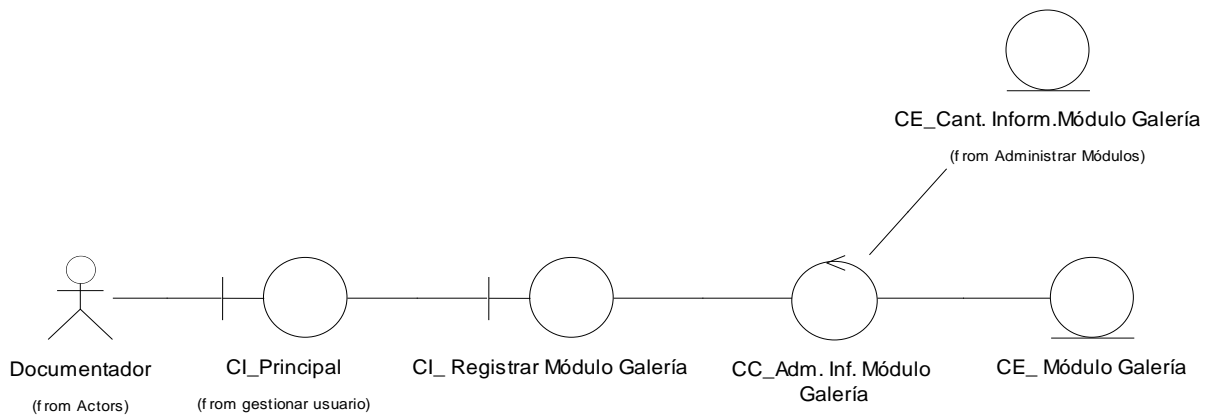


- **Diagrama de clases de análisis del Caso Eliminar Información del Módulo Temas**

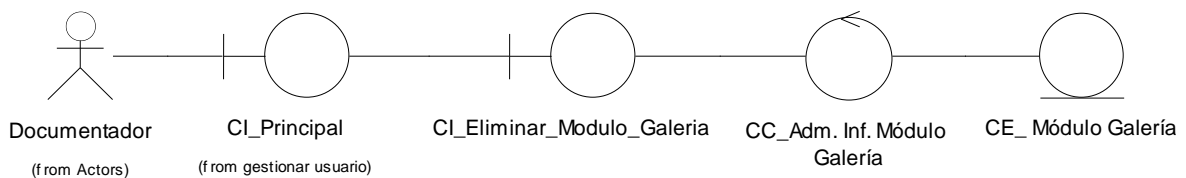


### 3.2.2.7 - Diagramas de clases de análisis de Administrar Información del Módulo Galería

- **Diagrama de clases de análisis del Caso Registrar Información del Módulo Galería**



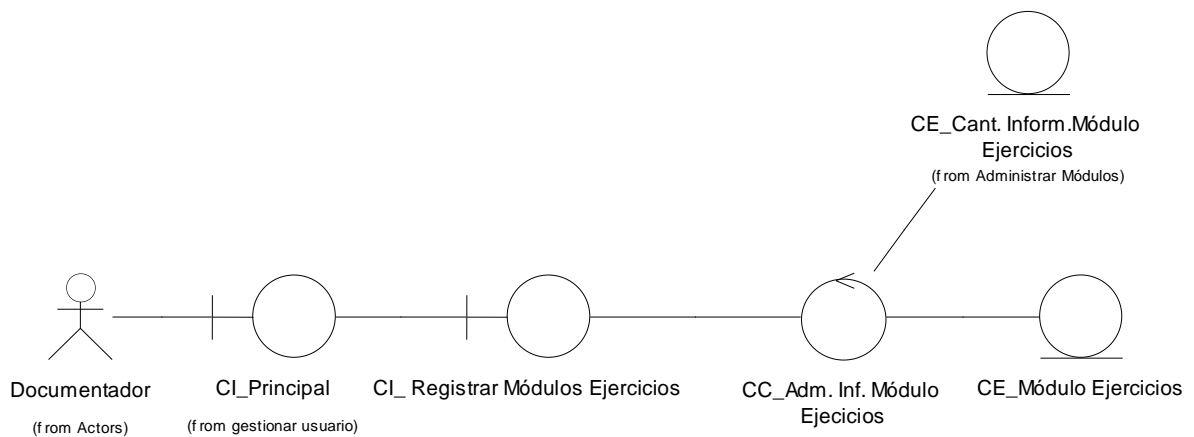
- **Diagrama de clases de análisis del Caso Eliminar Información del Módulo Galería**



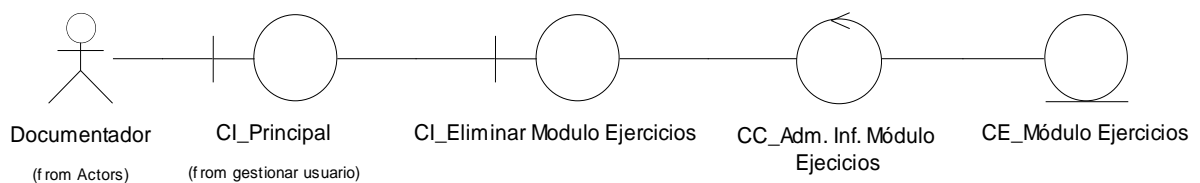
### 3.2.2.8 - Diagramas de clases de análisis de Administrar Información del Módulo

#### Ejercicios

- **Diagrama de clases de análisis del Caso Registrar Información del Módulo Ejercicios**

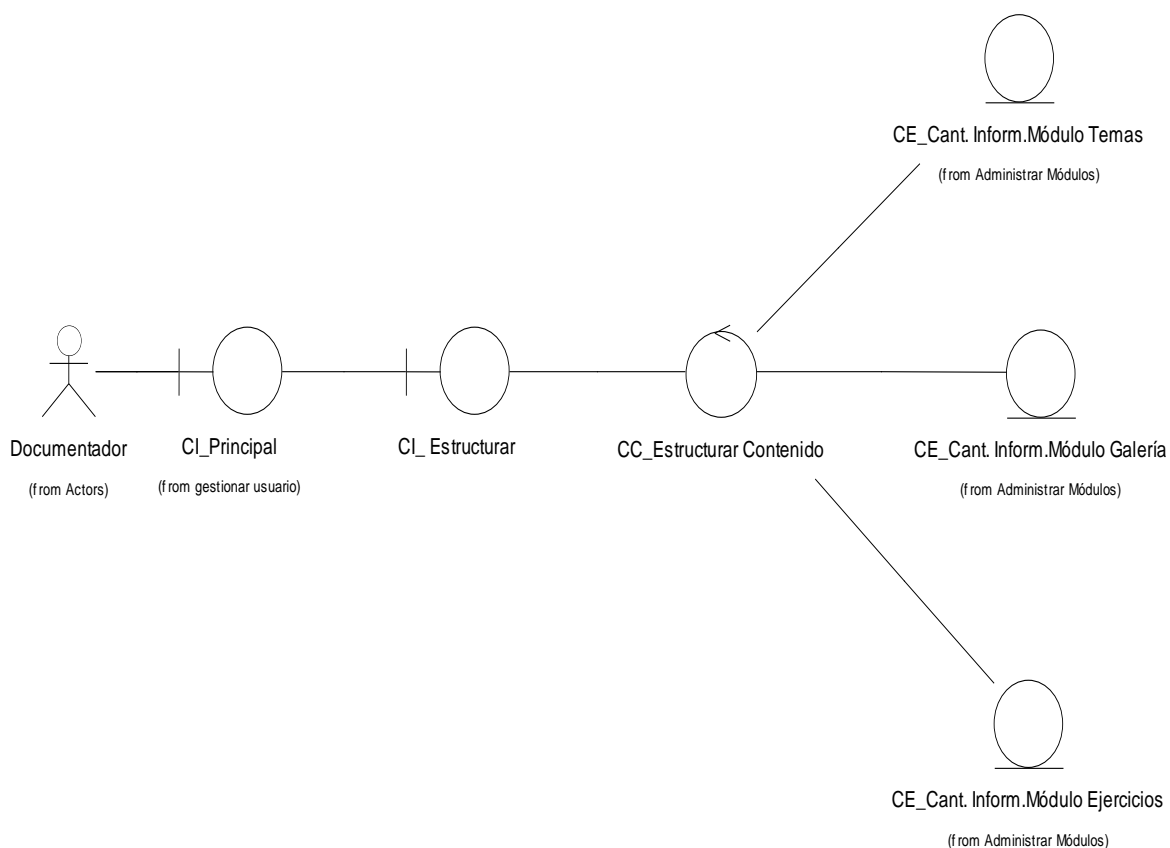


- **Diagrama de clases de análisis del Caso Eliminar Información del Módulo Ejercicios**





### 3.2.2.9 - Diagramas de clases de análisis de Estructurar Información



### 3.2.3 - Diagramas de colaboración. (Anexo 3)

#### 3.3 - Diseño

Durante el diseño se toman decisiones estratégicas y tácticas para cumplir los requerimientos funcionales y no funcionales de un sistema. Uno de los objetivos del diseño es crear una entrada apropiada y un punto de partida para la implementación del sistema.

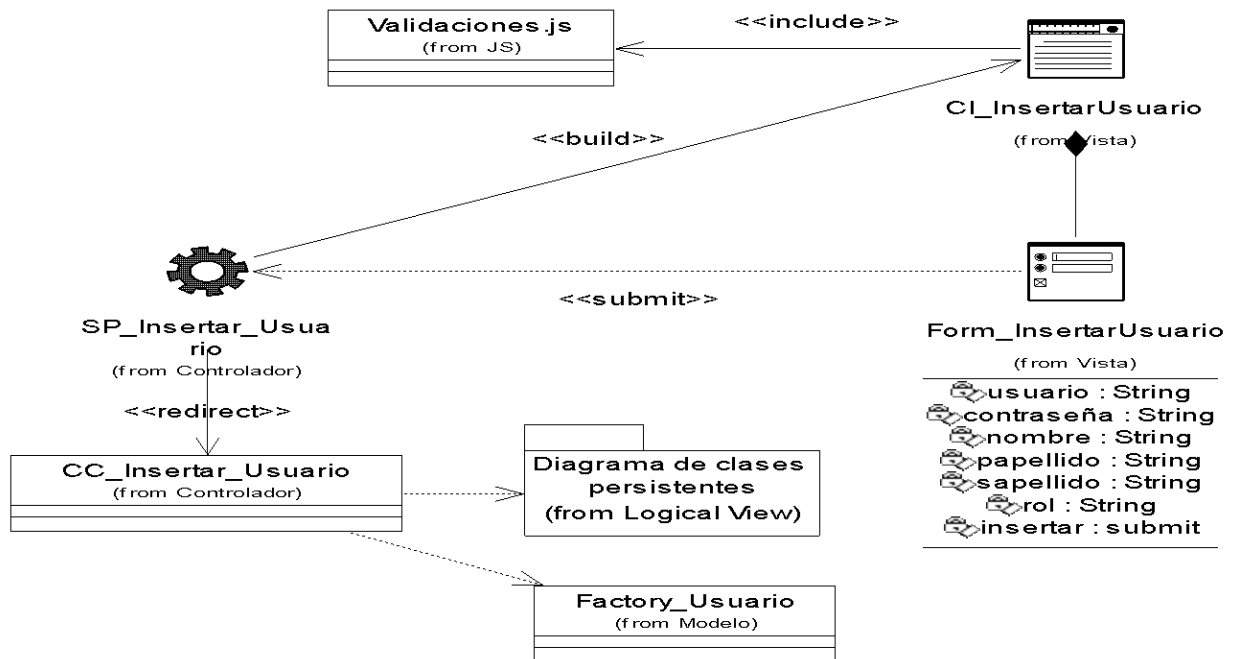
#### 3.3.1 - Modelo de clases del diseño

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. En el caso de las aplicaciones Web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. El diagrama de clases para las Aplicaciones Web, difiere un poco del resto de las aplicaciones que se construyen, puesto que en ellas son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación.

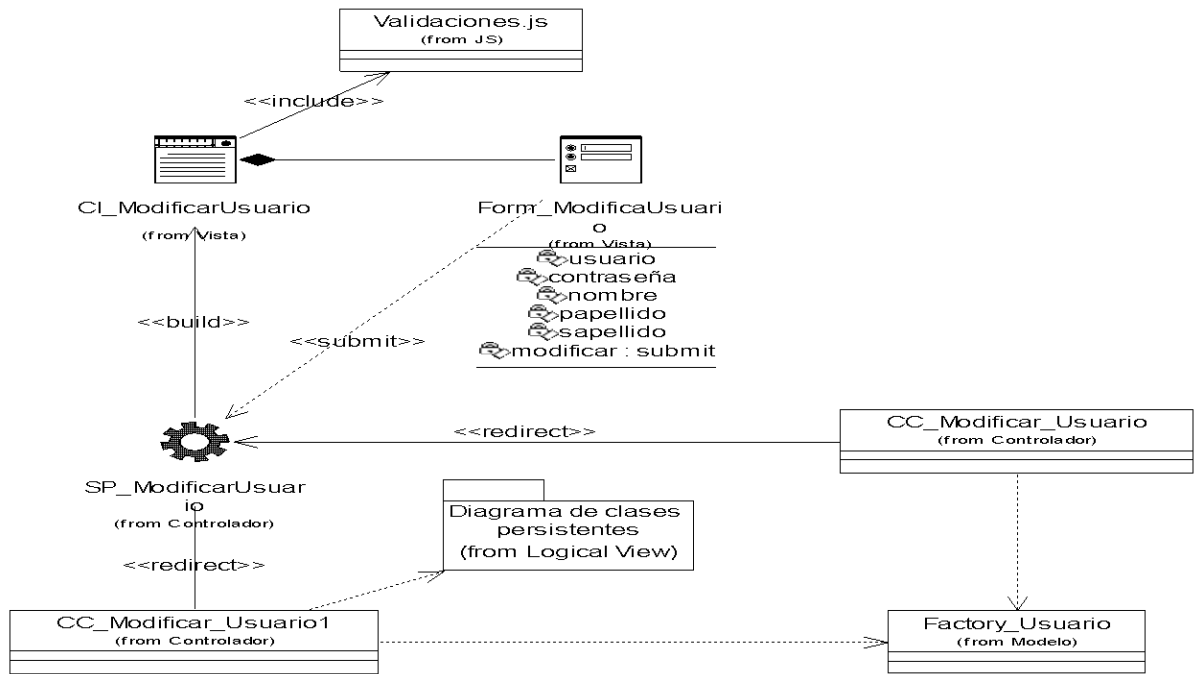
### 3.3.2 - Diagramas de clases de diseño

#### 3.3.2.1 - Diagrama de clases de diseño del Caso de Uso Gestionar Usuarios

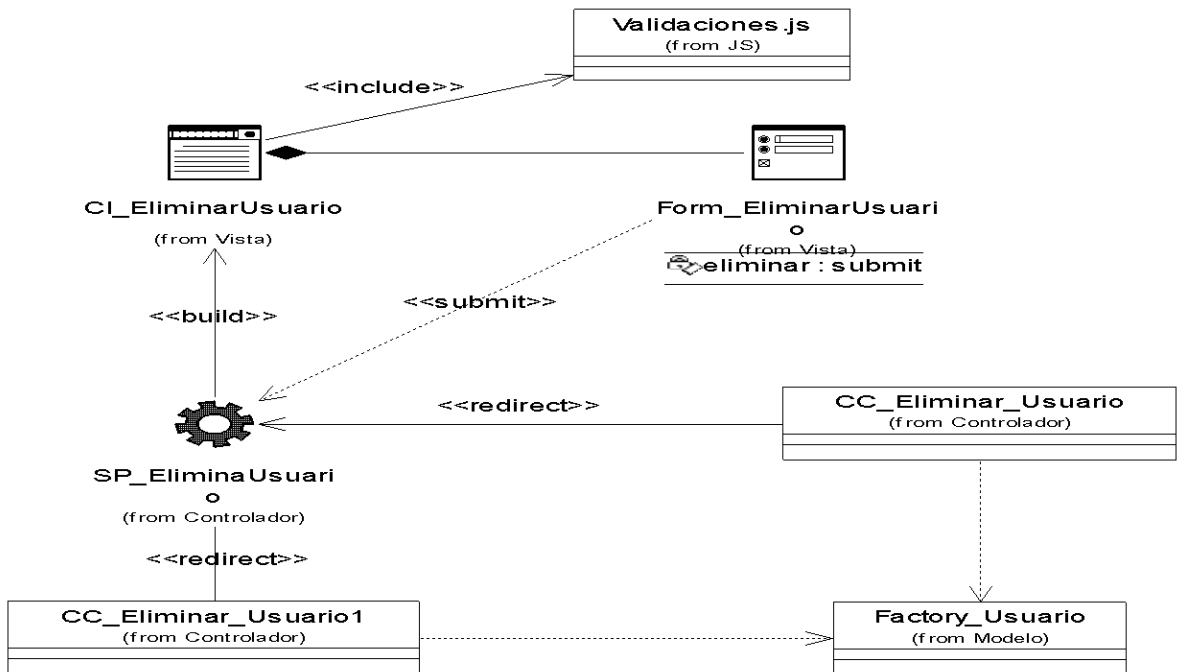
- Diagrama de clases de diseño del Caso Insertar Usuario



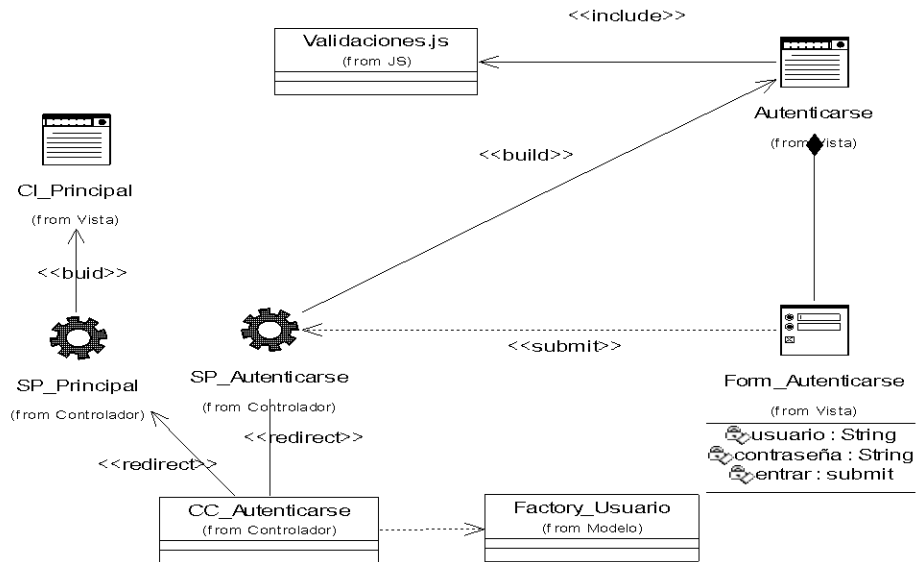
• Diagrama de clases de diseño del Caso Modificar Usuario



• Diagrama de clases de diseño del Caso Eliminar Usuario

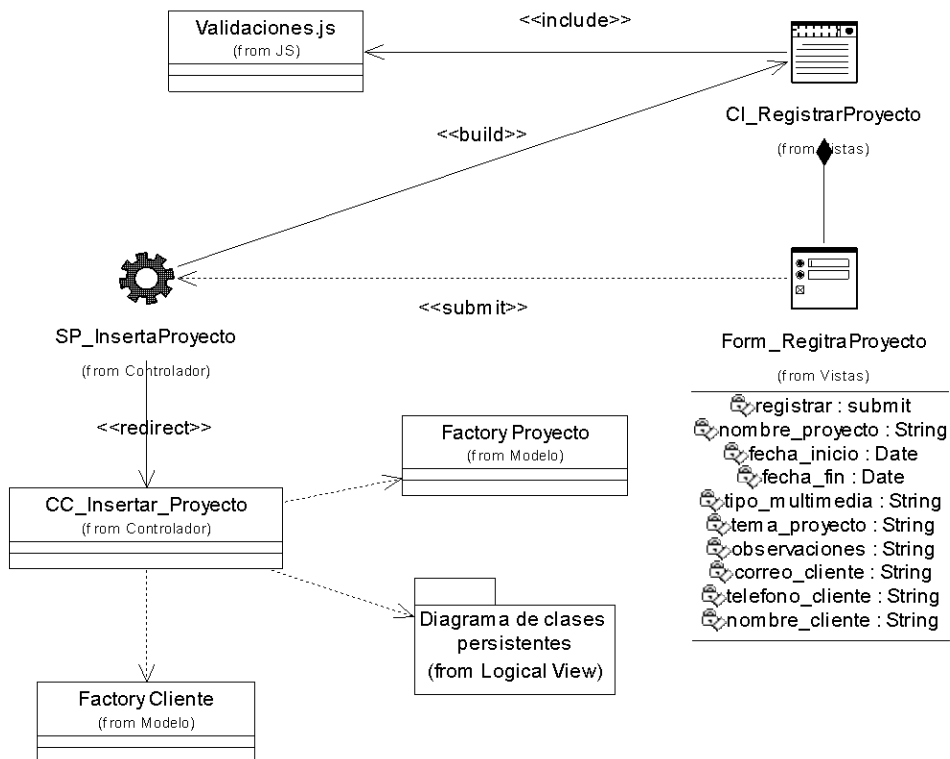


### 3.3.2.2 - Diagrama de clases de diseño del Caso de Uso Autenticar Usuarios

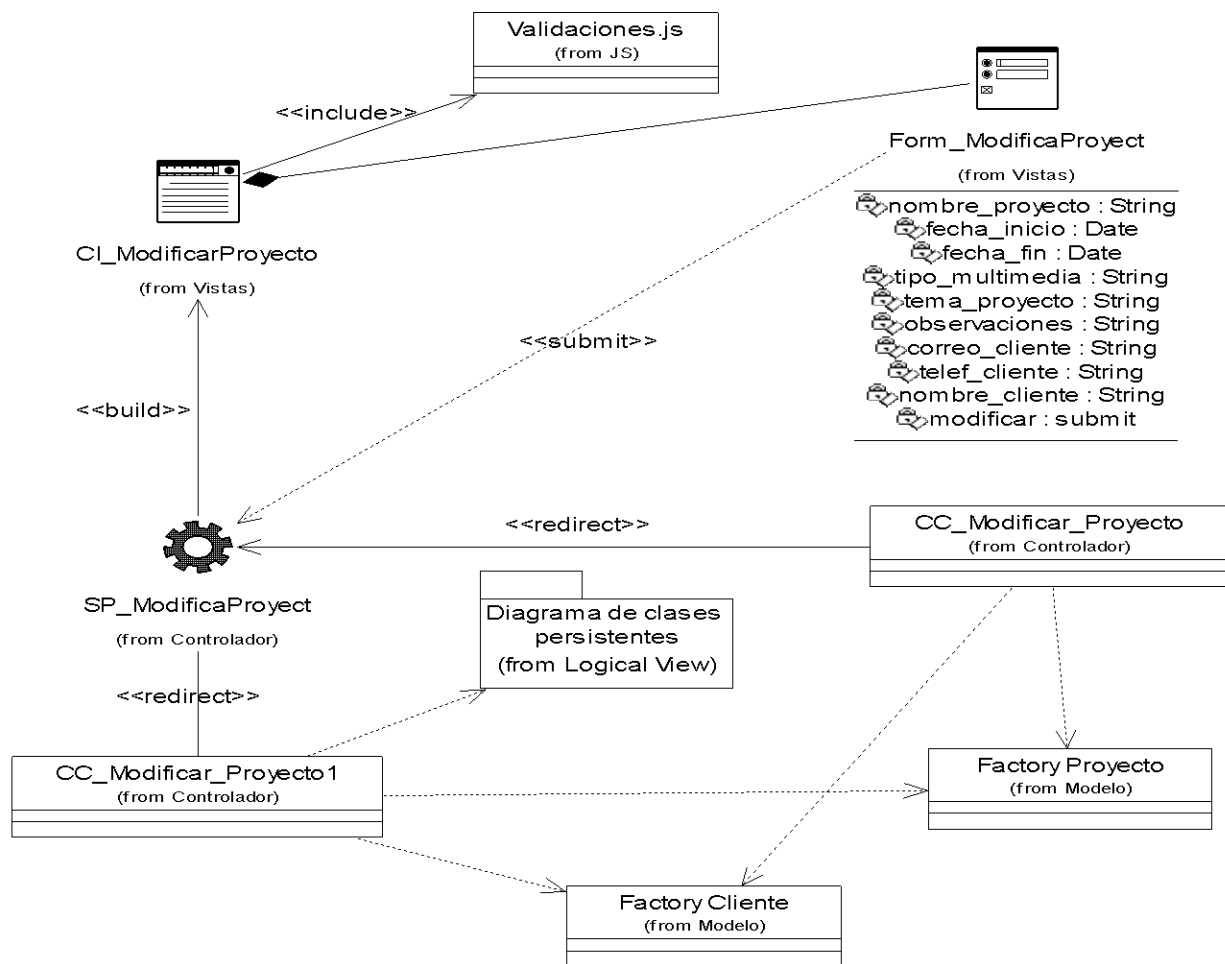


### 3.3.2.3 - Diagrama de clases de diseño del Caso de Uso Gestionar Proyecto

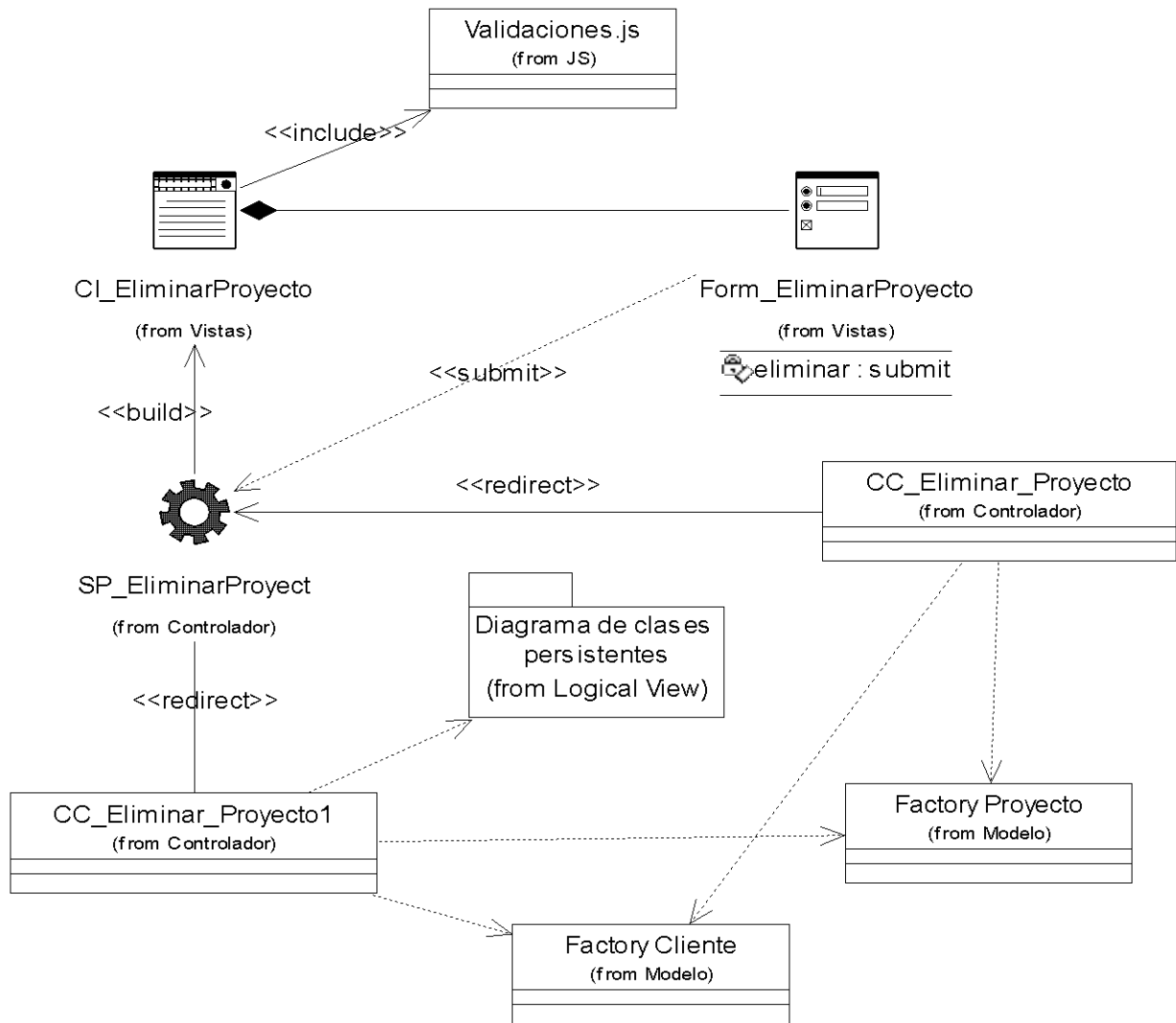
- Diagrama de clases de diseño del Caso Insertar Proyecto



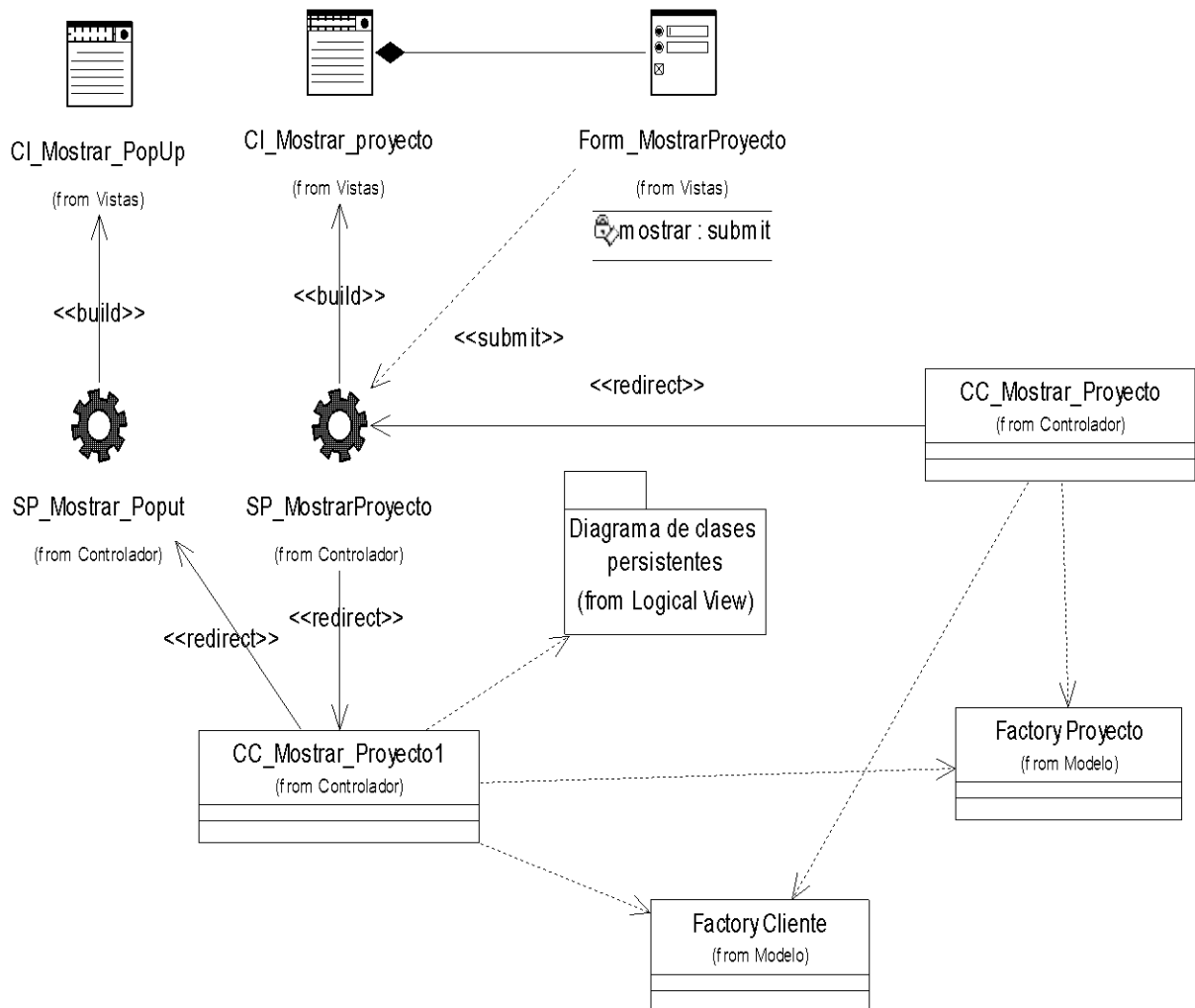
• Diagrama de clases de diseño del Caso Modificar Proyecto



• Diagrama de clases de diseño del Caso Eliminar Proyecto

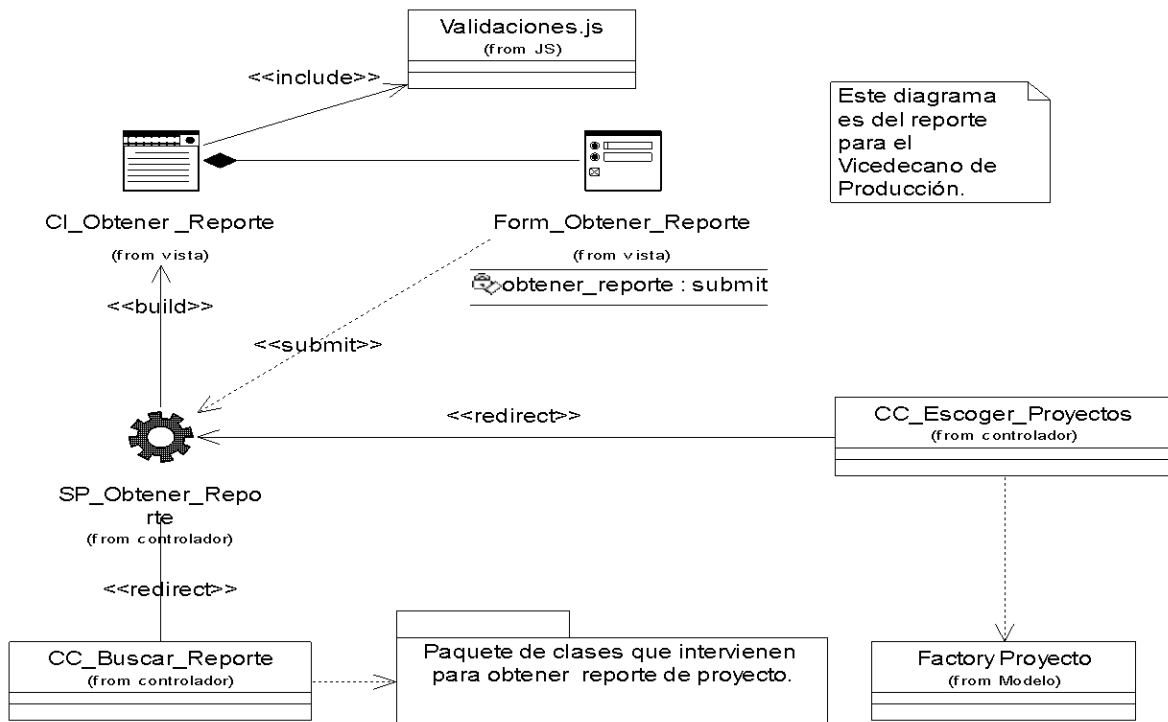


• Diagrama de clases de diseño del Caso Mostrar Proyecto

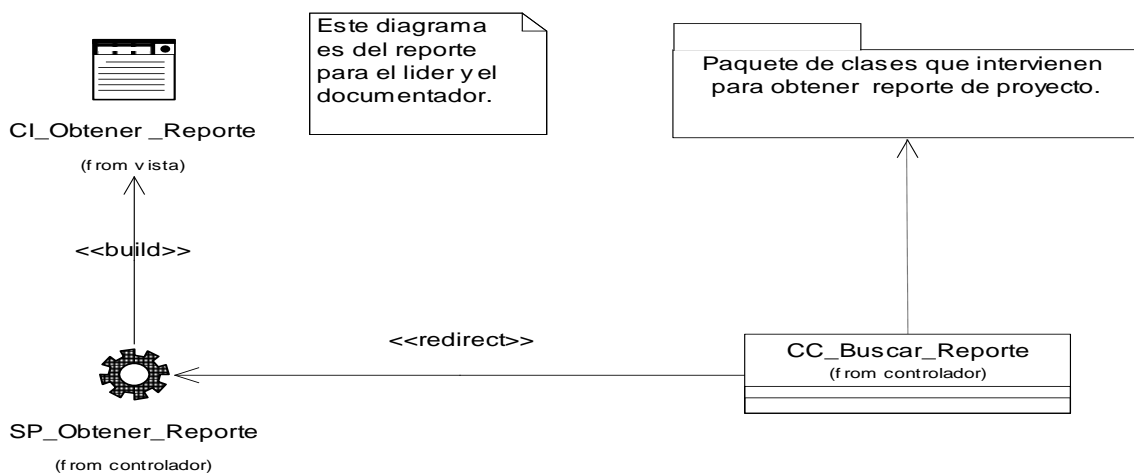


### 3.3.2.4 - Diagrama de clases de diseño del Caso de Uso Obtener Reporte

- Diagrama de clases de diseño del Caso de Uso Obtener Reporte (Vicedecano Producción)



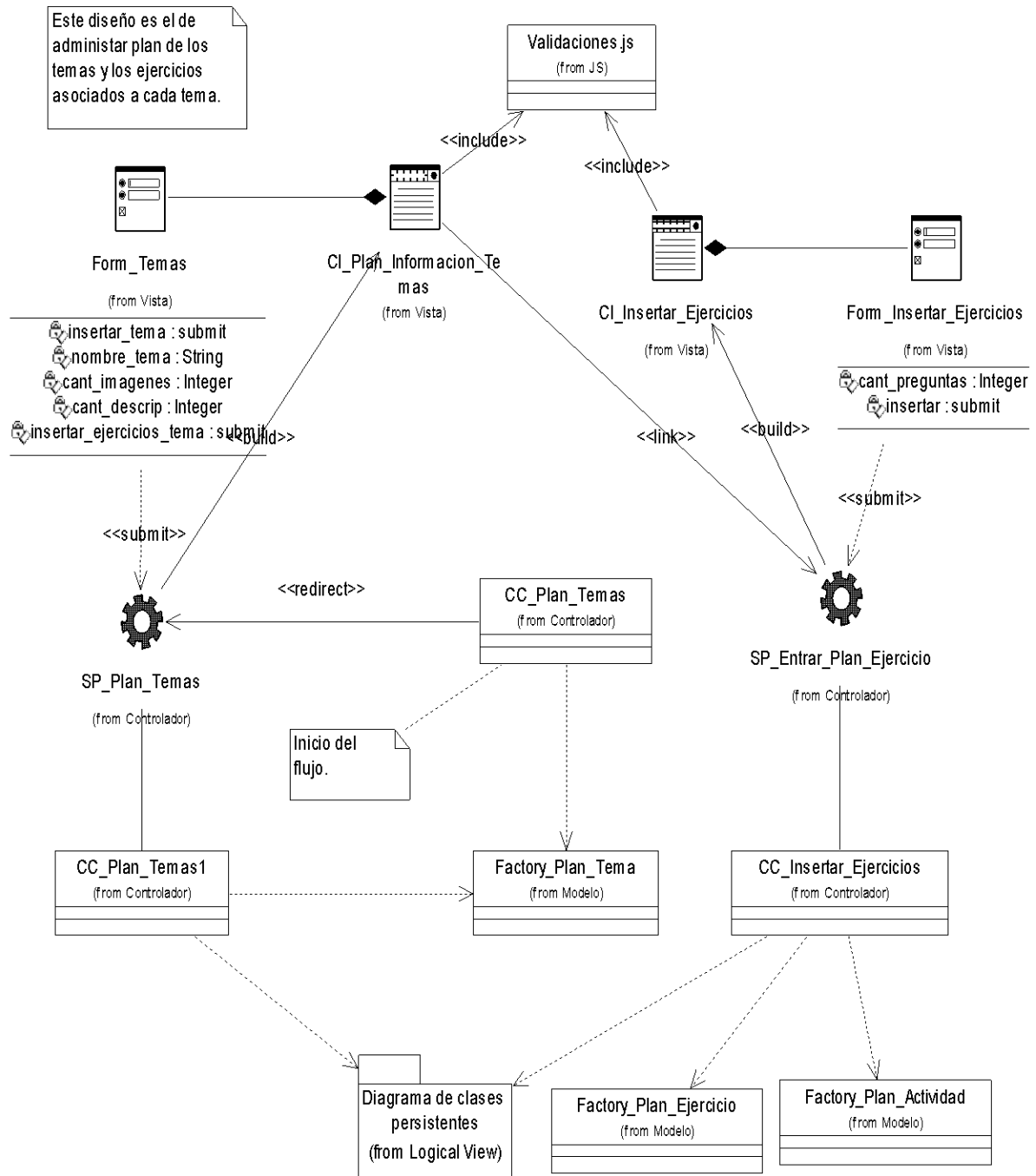
- Diagrama de clases de diseño del Caso de Uso Obtener Reporte (Líder Proyecto, Documentador)



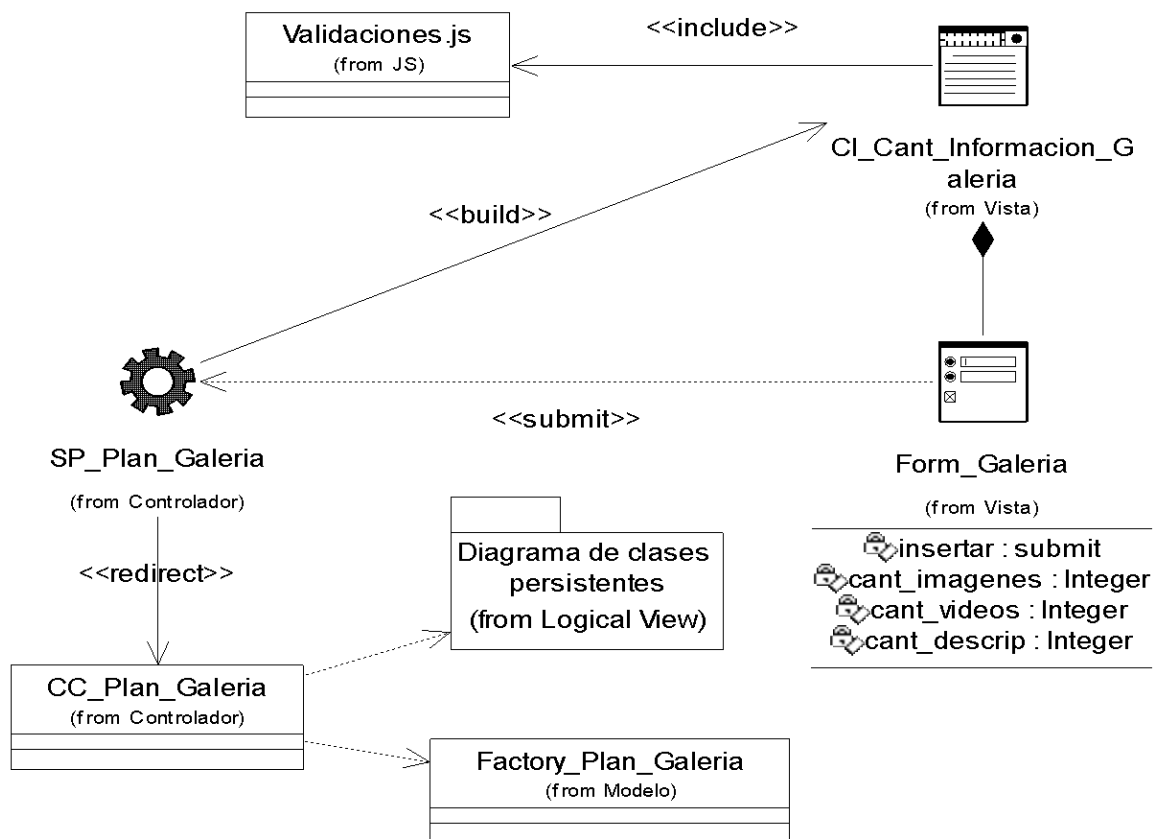


### 3.3.2.5 - Diagrama de clases de diseño del Caso de Uso Administrar Módulos

- Diagrama de clases de diseño del Caso Administrar Módulo Temas y Ejercicio

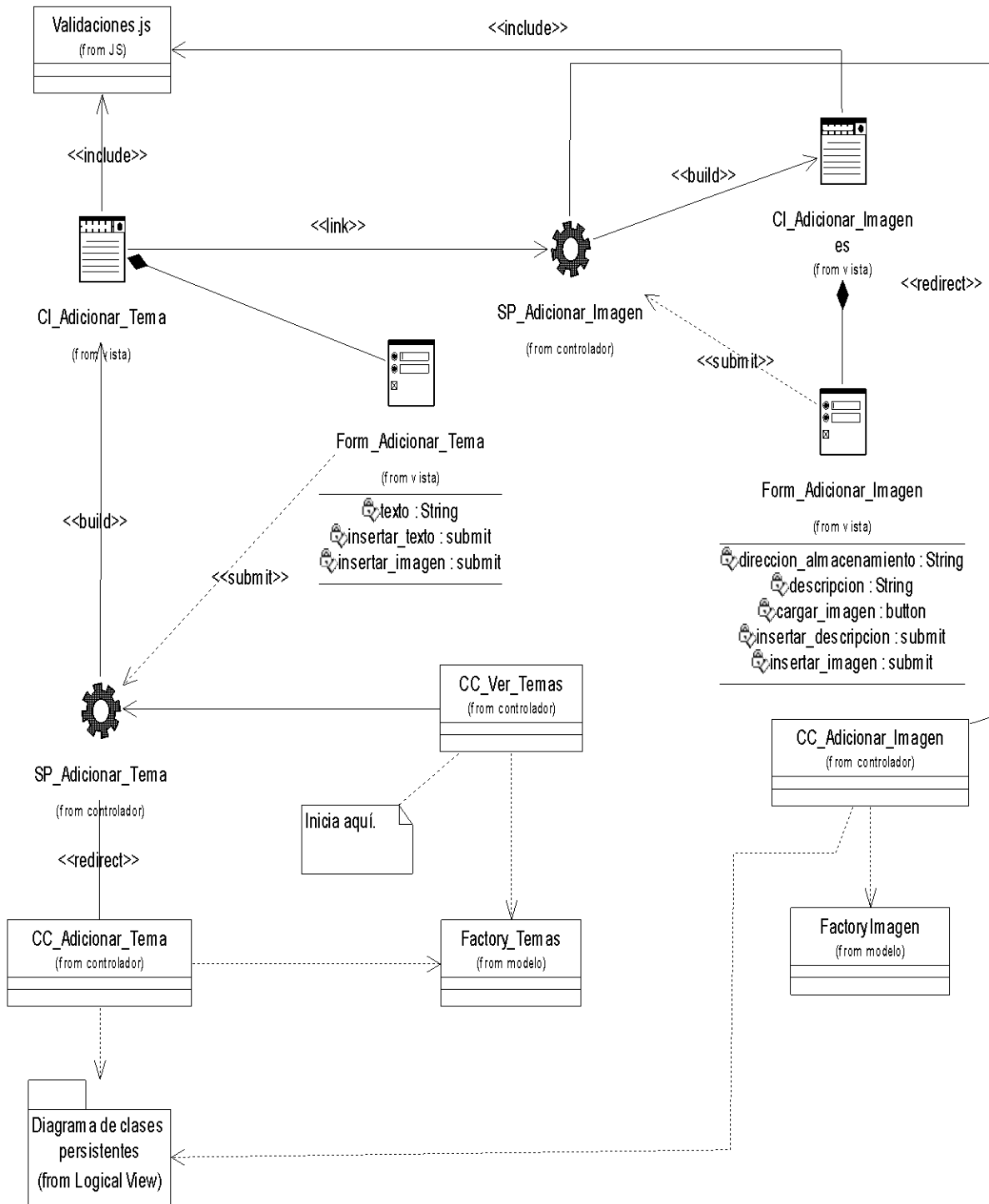


• Diagrama de clases de diseño del Caso Administrar Módulo Galería

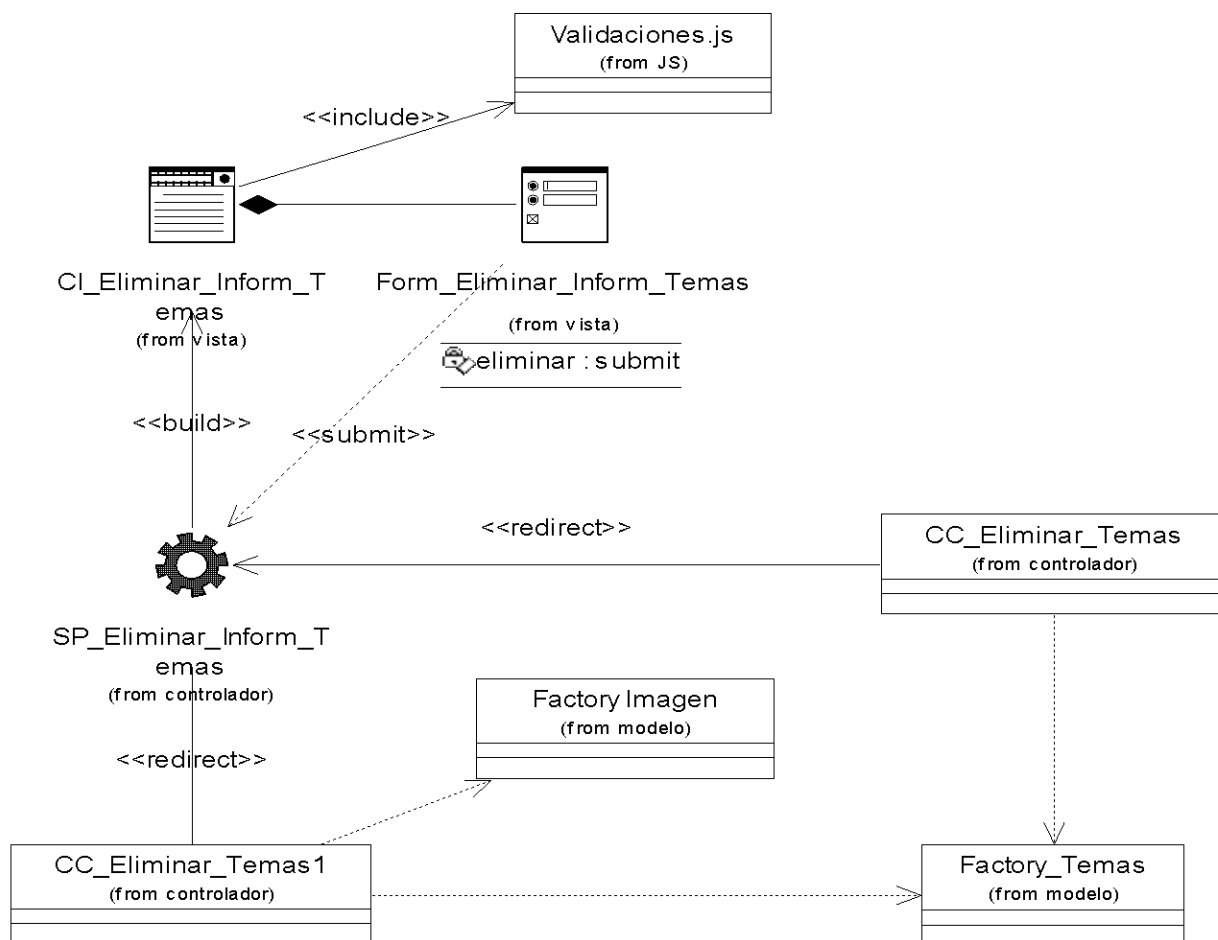


### 3.3.2.6 - Diagrama de clases de diseño del Caso de Uso Administrar Información del Módulo Temas

- Diagrama de clases de diseño del caso Insertar Información del Módulo Temas

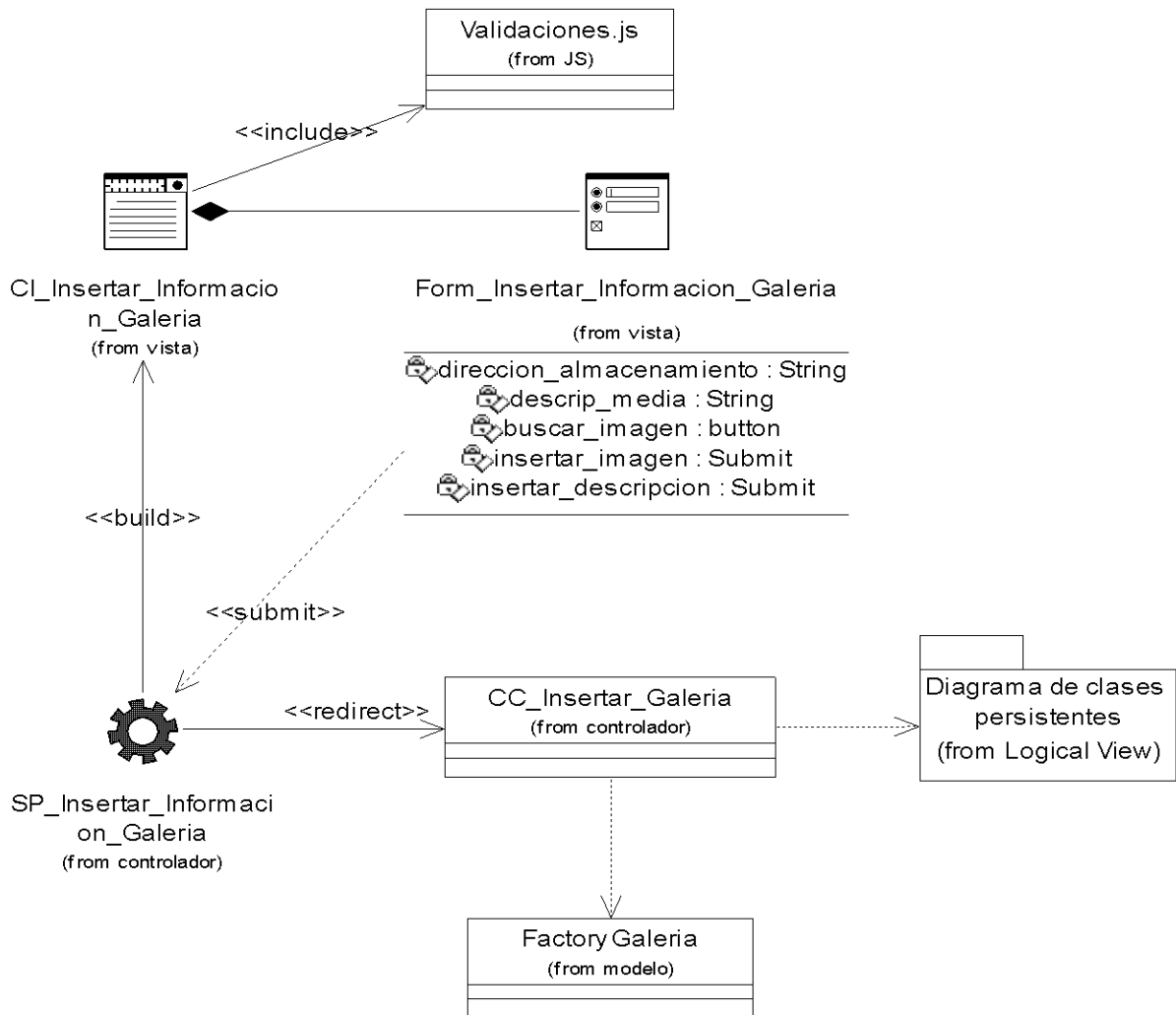


• Diagrama de clases de diseño del caso Eliminar Información del Módulo Temas

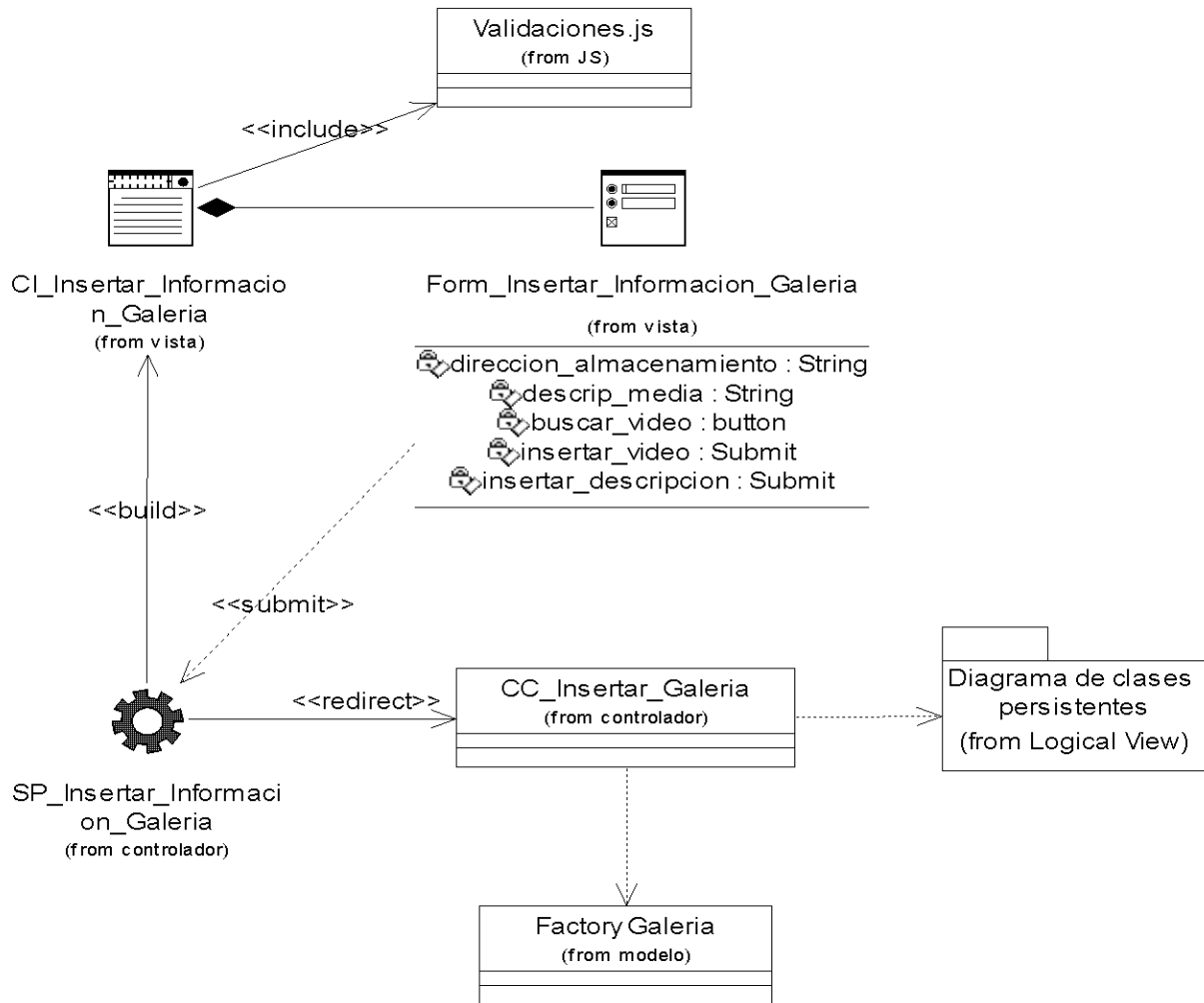


### 3.3.2.7 - Diagrama de clases de diseño del Caso de Uso Administrar Información del Módulo Galería

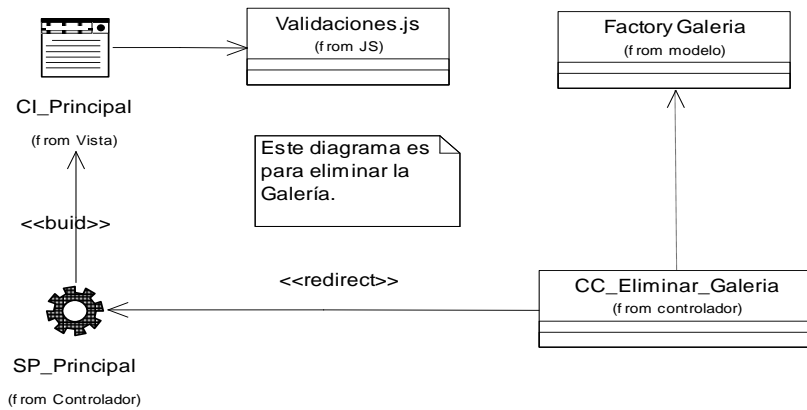
- Diagrama de clases de diseño del caso Insertar Información del Módulo Galería Imágenes



• Diagrama de clases de diseño del caso Insertar Información del Módulo Galería Videos

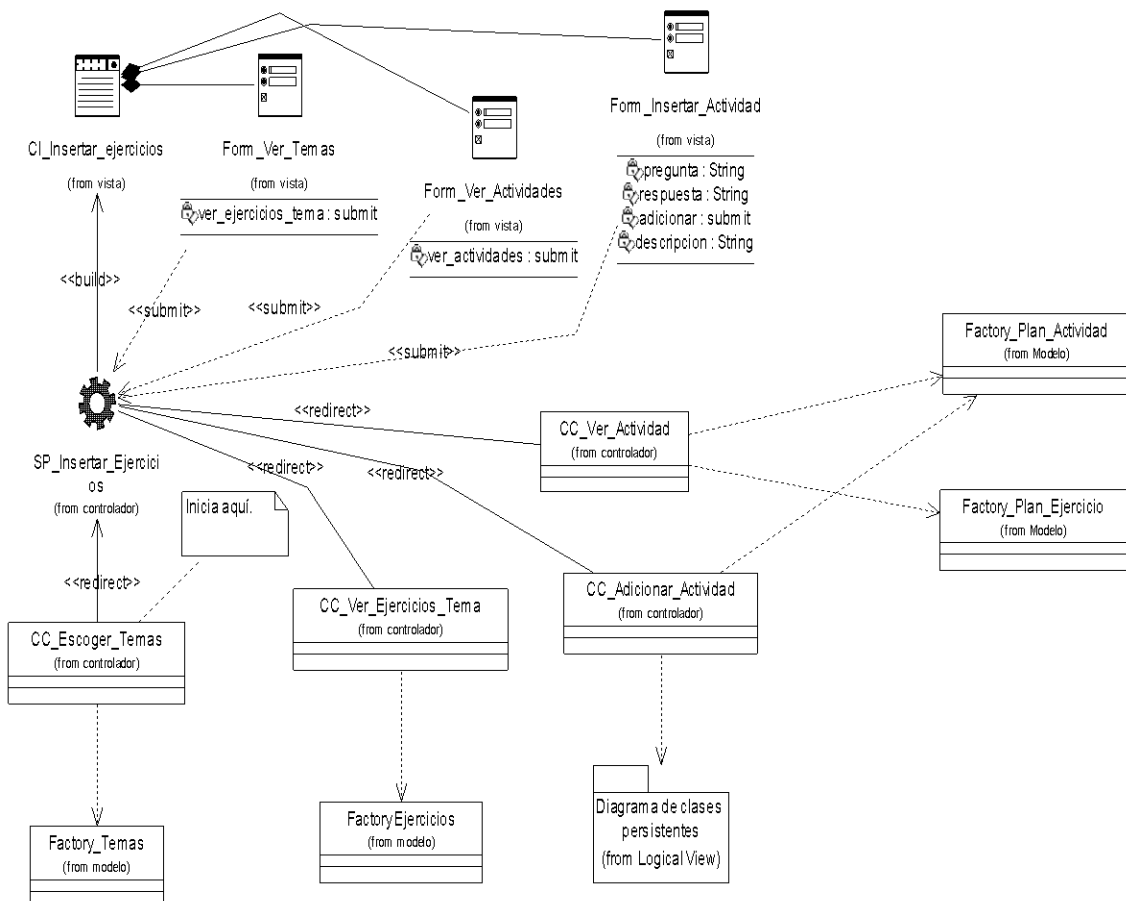


• Diagrama de clases de diseño del caso Eliminar Información del Módulo Galería

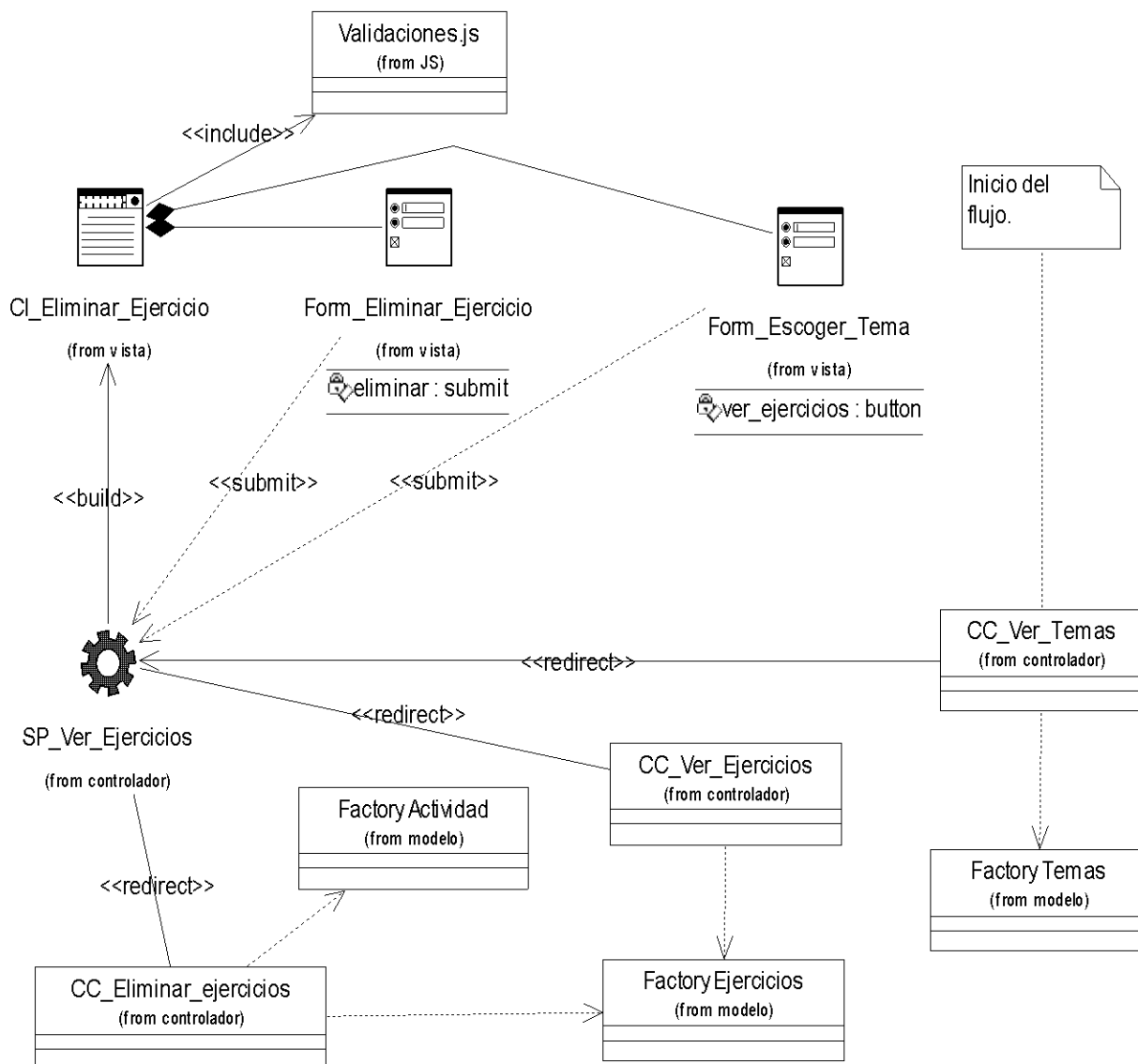


3.3.2.8 - Diagrama de clases de diseño del Caso de Uso Administrar Información del Módulo Ejercicios

• Diagrama de clases de diseño del caso Insertar Información del Módulo Ejercicios

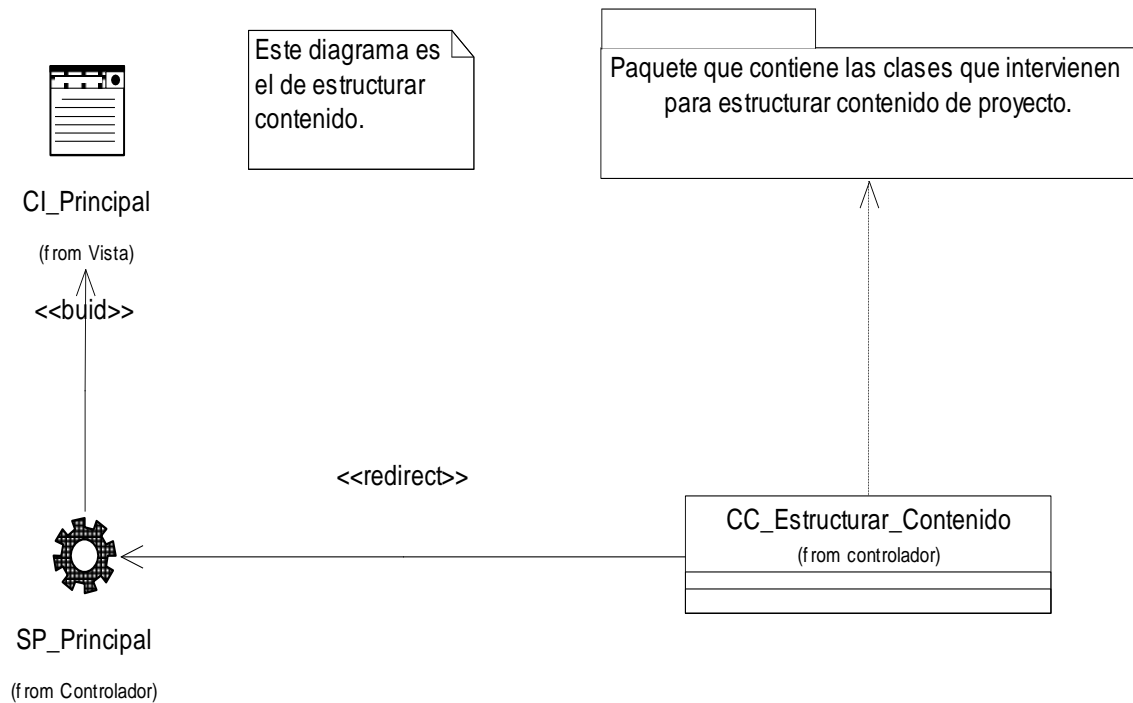


• Diagrama de clases de diseño del caso Eliminar Información del Módulo Ejercicios





### 3.3.2.9 - Diagrama de clases de diseño del Caso de Uso Estructurar Información



### 3.3.3 - Descripción de las Clases (Anexo 4)

### 3.3.4 - Principios de Diseño

En el diseño de una aplicación web se hace necesario regirse por un grupo de principios de diseño que influyen en gran medida tanto en el éxito o fracaso del mismo, entre estos principios están el de Interfaz del Sistema, la Seguridad y el de Tratamiento de Errores, a continuación se describen los mismos que son los seguidos en el desarrollo de dicho sistema.

#### 3.3.4.1 - Interfaz de Usuario

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema, se puede definir como: “el conjunto de trabajos y pasos que seguirá el usuario, durante todo el tiempo que se relacione con el programa, detallando lo que verá y escuchará en cada momento, y las acciones que realizará, así como las respuestas que el sistema dará”.

La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

Para el desarrollo de la interfaz se tiene en cuenta los siguientes aspectos:

- Reducir la carga a la memoria.

- Atajos a Usuarios expertos.
- Obtener información de retroalimentación.
- Diseño de diálogos que conducen a una conclusión.
- Previsión de errores y manejo de errores simples.
- Lograr deshacer acciones fácilmente.
- Que el usuario sintiera la sensación de control.

Una de las premisas fundamentales de la aplicación es la ventaja que proporciona las interfaces Web sobre las interfaces de comando. Ya que las interfaces Web:

- Proporcionan un ambiente amigable.
- Conducen a un aprendizaje más natural.
- Establecen un “sentimiento” (sobre todo en la uniformidad del ambiente) al usuario que enriquece su experiencia en el uso de la aplicación.

Además de estos principios, se tuvieron en cuenta las siguientes características:

- Utilizar una misma tipografía, forma y estilo en todas las páginas.
- La facilidad del usuario de poder navegar desde cualquier punto a otro dentro de la aplicación.
- Se tuvo presente siempre el ancho de banda y por ello se utilizaron formato de imágenes de compresión favorables.
- La simplicidad y consistencia, favoreciendo la usabilidad de la aplicación.
- Navegación simple en todas las páginas de la aplicación, de forma tal que siempre sea accesible por el usuario.
- Estabilidad y uniformidad del diseño, para así poder ubicar al usuario dentro del mismo y hacerlo sentir parte de él.

Se debe utilizar una hoja de estilos para guardar la configuración del diseño para todas las páginas, para los botones y las líneas se utilizaron estos estilos, eliminando así el número de imágenes que demoren la presentación de la página. Se deben realizar múltiples operaciones en cada página, de forma que el usuario no tenga que moverse tanto dentro de la aplicación, para completar una operación. [43]

#### **3.3.4.2 - Tratamiento de Errores**

El correcto tratamiento de los errores en un sistema, influye notablemente en el buen funcionamiento de este, a la vez que se garantiza la integridad de la información. Para lograr esto, fueron previstos todos los errores posibles que pudiera generar el sistema a partir de la interacción del usuario con este, así como de problemas inesperados que pudieran surgir. Muchos de estos posibles errores son

tratados a partir de funciones JavaScript, y otros se obtienen a partir de la ejecución de la página en el servidor de la aplicación, estos últimos aparecen de forma textual en la misma página en la que se encuentra el usuario en ese momento. En cualquiera de los dos casos, los mensajes de error son mostrados con un texto claro, que señala de forma explícita y legible la respuesta del sistema ante cualquier acción que se ejecute. Además de esto, se muestran mensajes de confirmación ante acciones que son irreversibles, como es el caso de la eliminación de datos, a la vez que se muestran mensajes para indicar cuándo una acción fue realizada con éxito. [44]

### **3.3.4.3 – Seguridad**

La seguridad del sistema, se basa principalmente en el empleo de la autenticación como una acción de obligatorio cumplimiento para el posterior uso del mismo, por lo que cada usuario del sistema tendrá un nombre para identificarse y una contraseña, los cuáles serán verificados antes de darle acceso a las funcionalidades de sistema y si alguno de estos datos son incorrectos, se denegará dicho acceso. La contraseña pasa por un proceso de encriptación en el cliente, siguiendo el algoritmo MD5 y posteriormente se verifica en el servidor, si la misma ha sido encriptada o no, para en caso negativo encriptarla y manejarla de esta forma. Por otra parte, cada usuario posee un rol, el cual es chequeado en cada página del sistema, con vista a darle o no acceso a la misma en dependencia de cual sea este rol. También, son validados cada uno de los campos de los formularios con el objetivo de evitar que se introduzcan o seleccionen datos no permisibles por el sistema. Además se evita la incorrecta ejecución de alguna consulta, a la vez que se asegura que una acción de inserción de determinada información en varias tablas de la base de datos se realice incorrectamente debido a que si alguna parte de esta información no se inserta con éxito, esto puede provocar que no se cuente con la información deseada posteriormente. [45]

### **3.3.5 - Diagrama de Clases Persistentes (Ver Anexo 5)**

### **3.3.6 - Modelo de Datos (Ver Anexo 6)**

### **3.3.7 - Conclusiones**

Concluye el capítulo de análisis, habiéndose hecho el modelo de clases de Análisis y los diagramas de interacción del análisis. También se concluye así con la etapa de diseño del sistema donde se graficaron los diagramas de diseño Web del sistema y se hizo referencia al tratamiento de errores generales del sistema y se graficaron además los diagramas de clases persistentes y modelo de datos que representa un análisis más profundo a tener en cuenta a la hora de implementar. Con todos estos elementos, se tiene una idea más precisa sobre los elementos constitutivos del sistema que se propone.

## **CAPÍTULO 4 IMPLEMENTACIÓN**

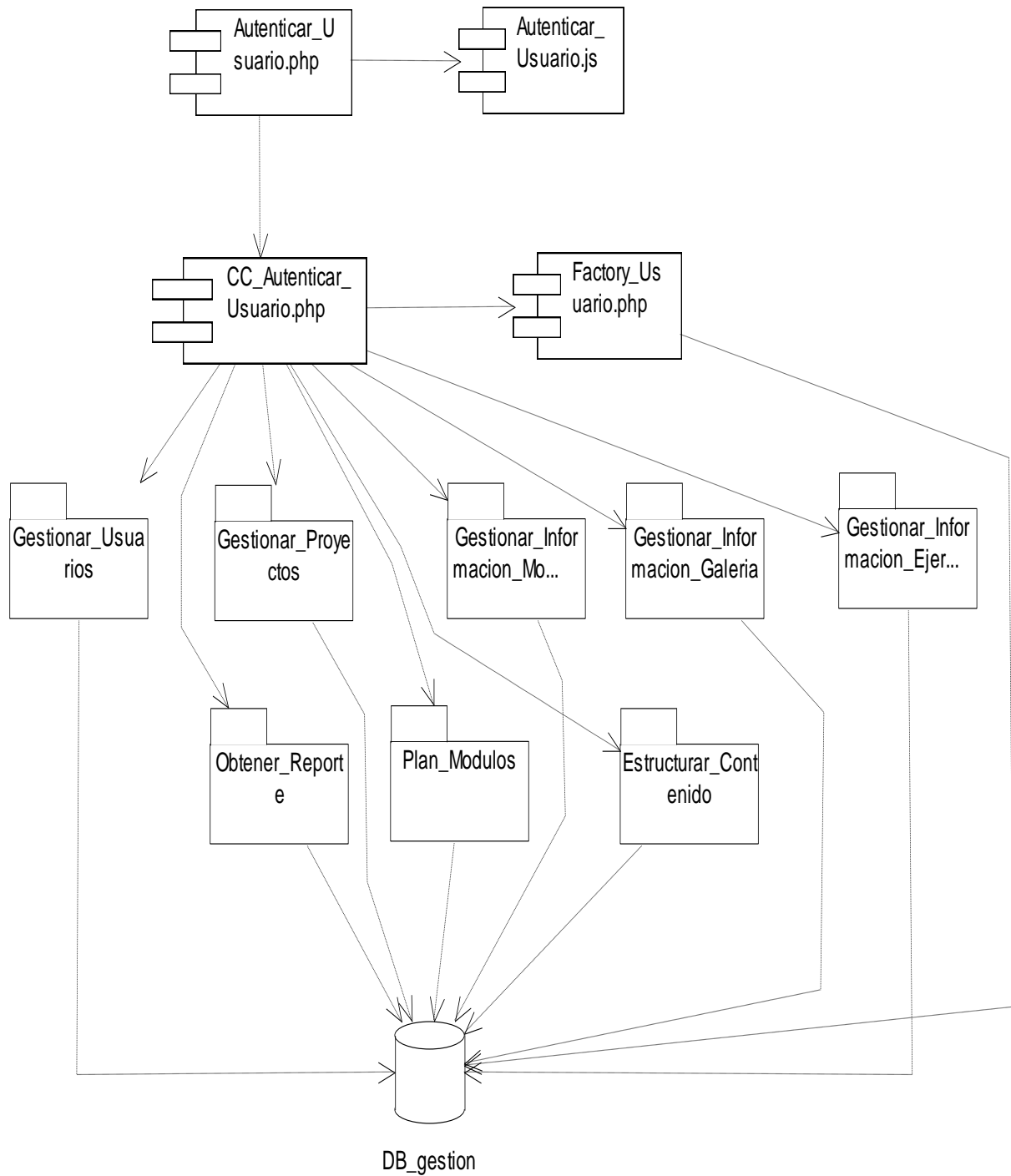
### **4.1 - Introducción**

En este capítulo se pasará a la implementación del sistema en el lenguaje de programación escogido basándose en el capítulo anterior, donde se abordó el análisis y diseño de la aplicación. En este capítulo se modela los diagramas de componentes y de despliegue que brindan una mayor vista para la implementación del producto.

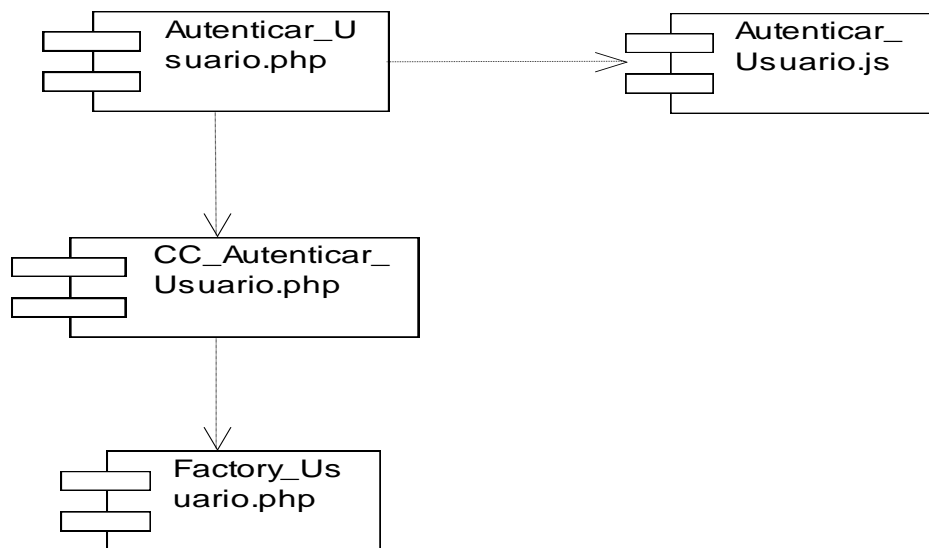
### **4.2 - Modelo de Implementación**

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

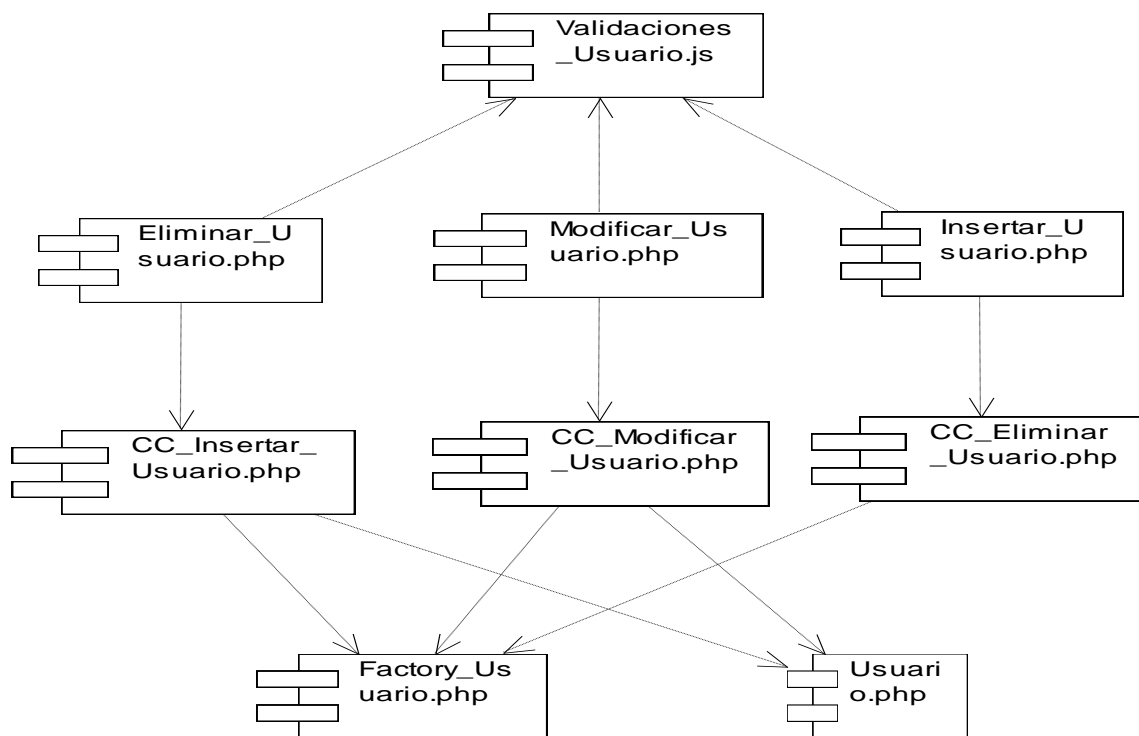
### 4.2.1 - Diagrama de Componentes por paquetes



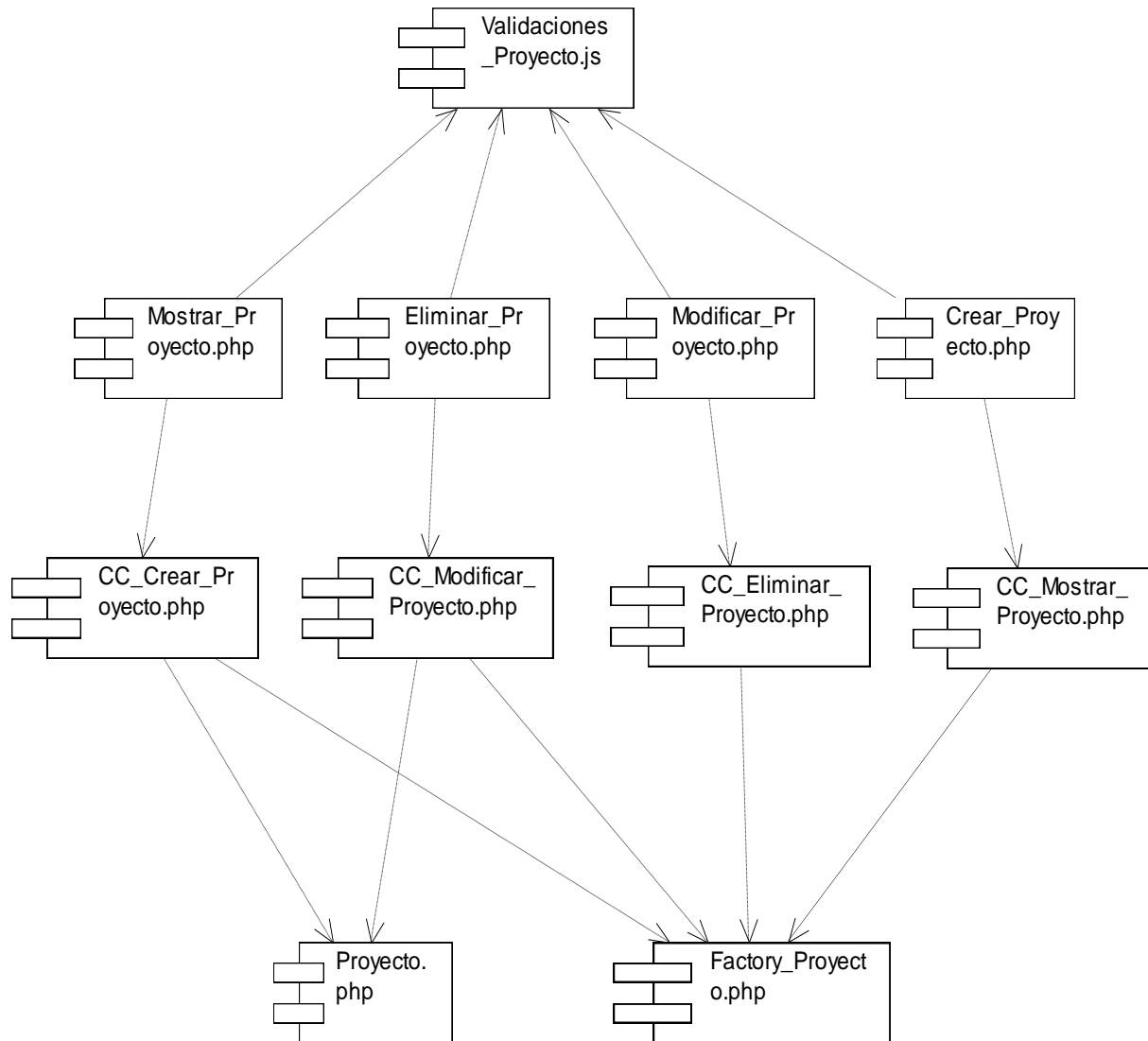
### 4.2.2 - Diagrama de Componentes Autenticar Usuario



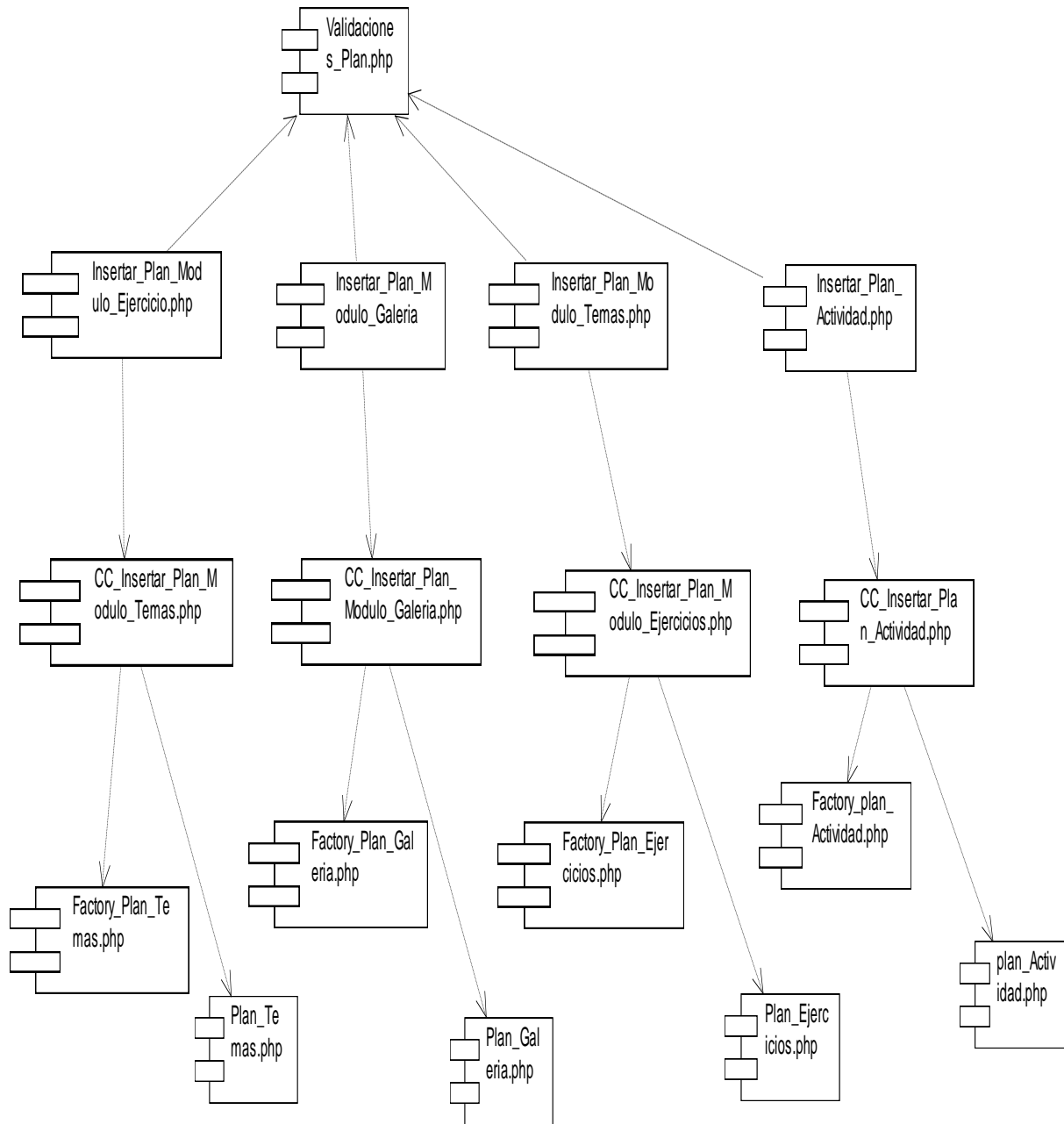
### 4.2.3 - Diagrama de Componentes Gestionar Usuarios



#### 4.2.4 - Diagrama de Componentes Gestionar Proyectos

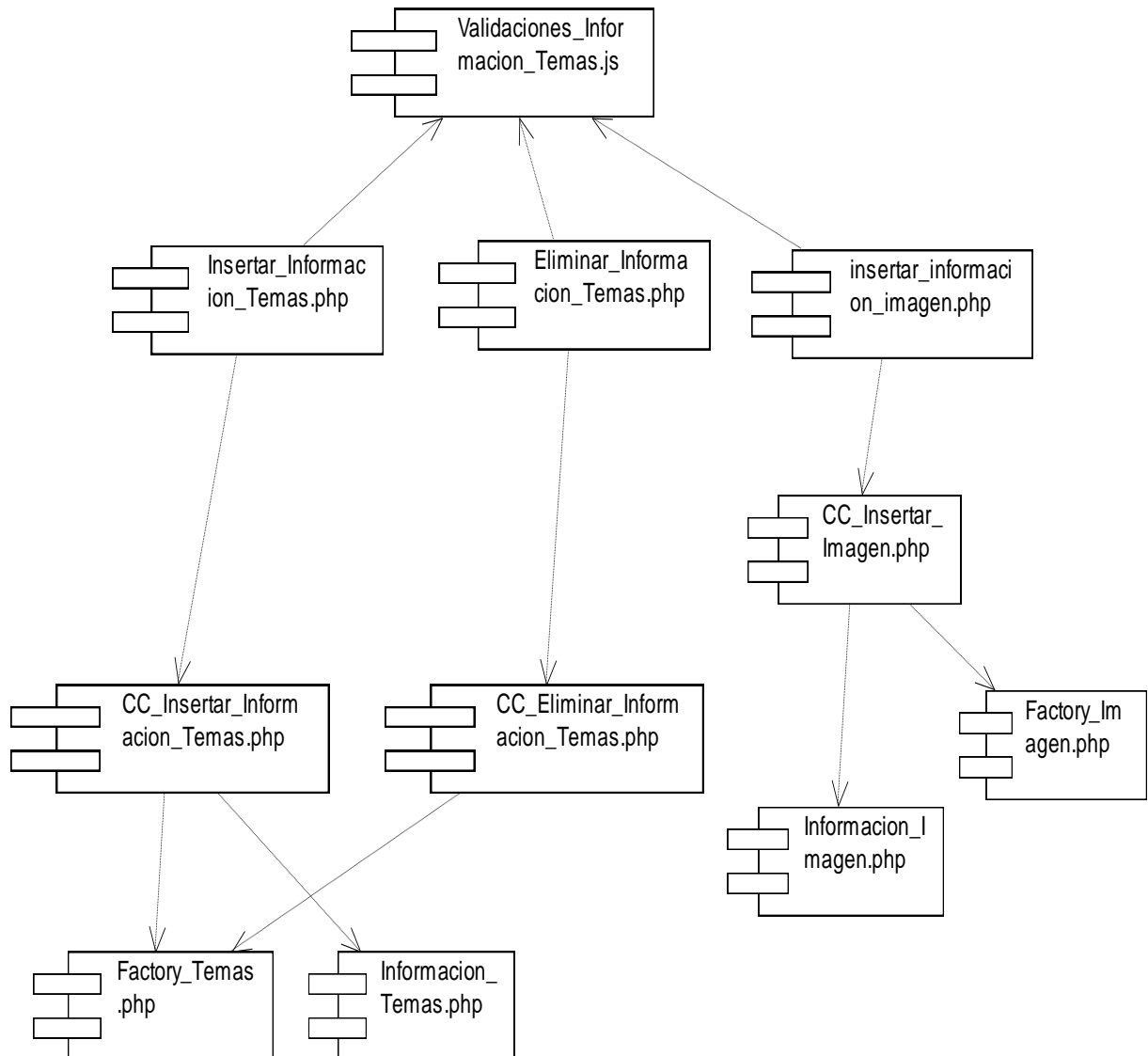


### 4.2.5 - Diagrama de Componentes Administrar Módulos

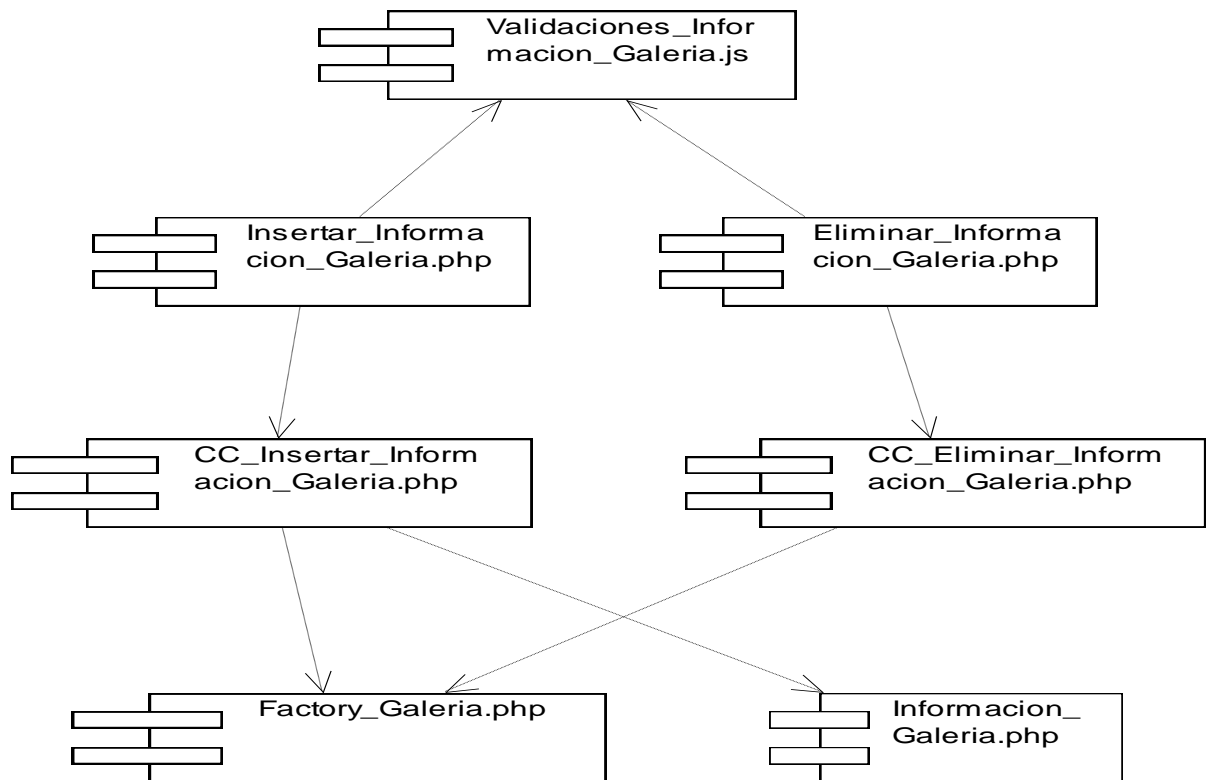




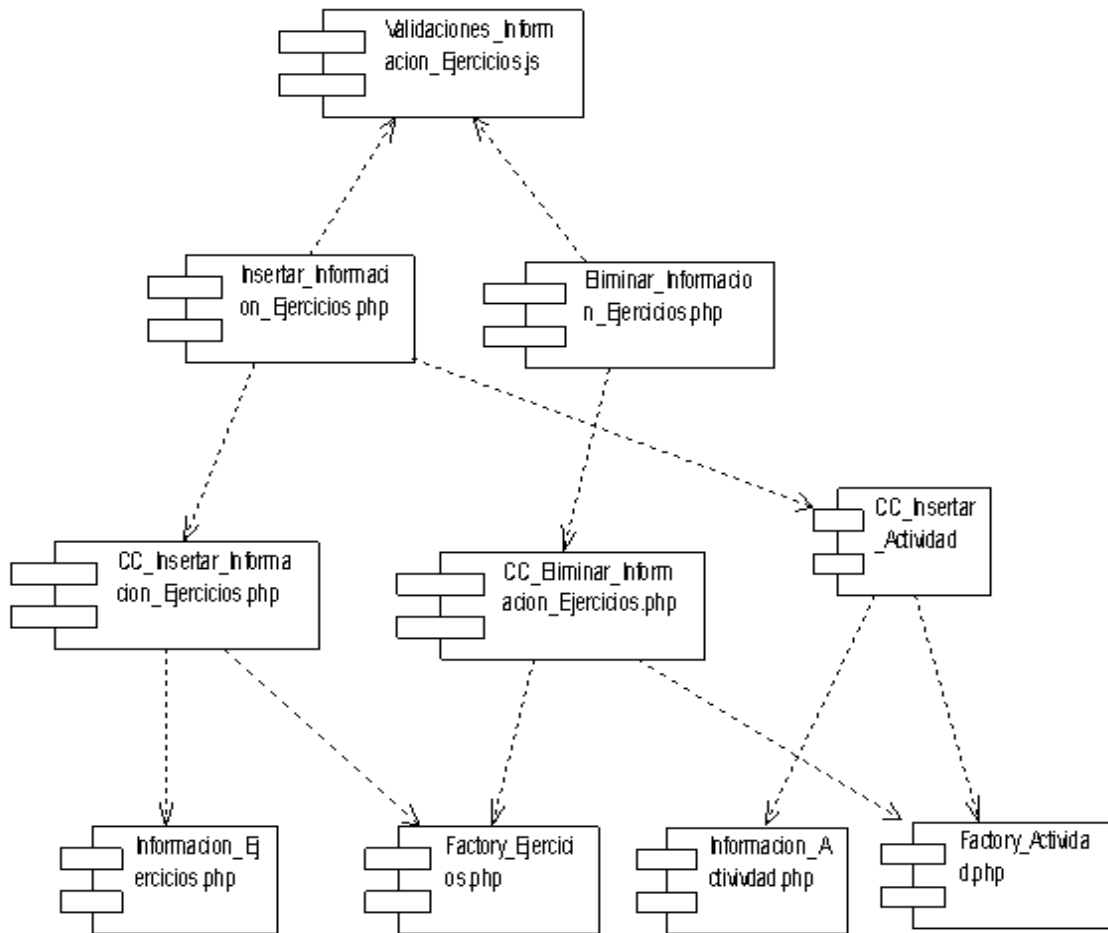
#### 4.2.6 - Diagrama de Componentes Administrar Información Módulo Tema



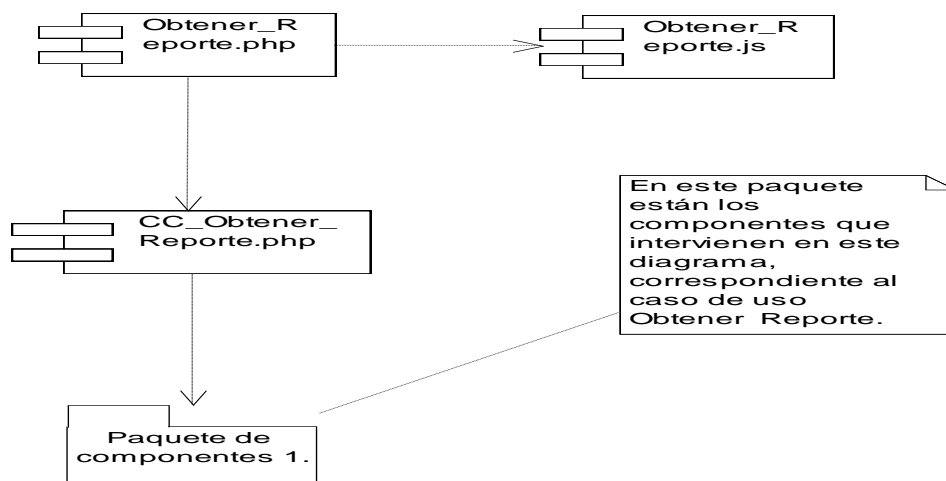
#### 4.2.7 - Diagrama de Componentes Administrar Información Módulo Galería



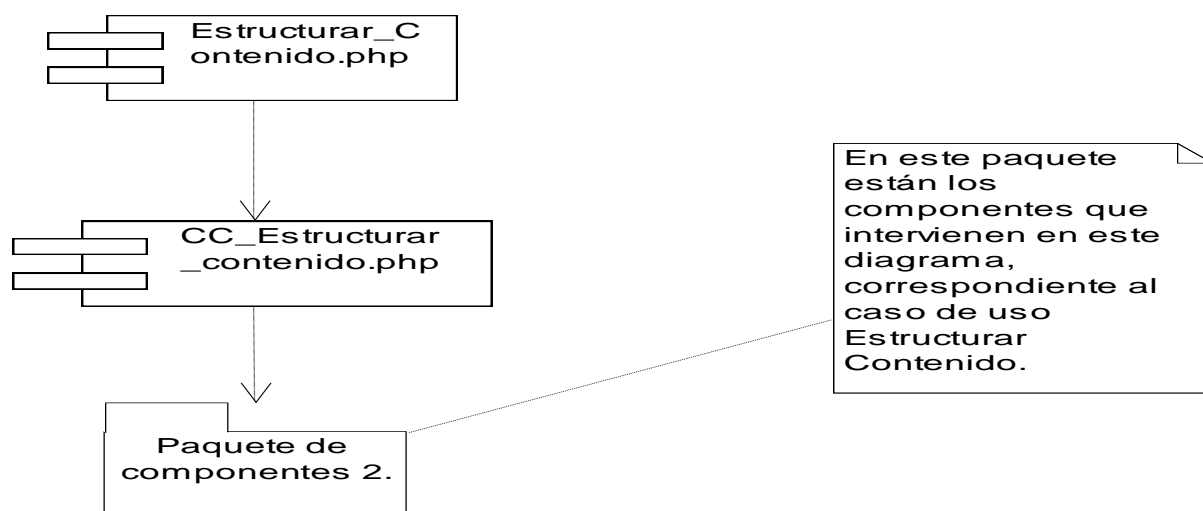
#### 4.2.8 - Diagrama de Componentes Administrar Información Módulo Ejercicio



#### 4.2.9 - Diagrama de Componentes Obtener Reporte



#### 4.2.10 - Diagrama de Componentes Estructurar Contenido

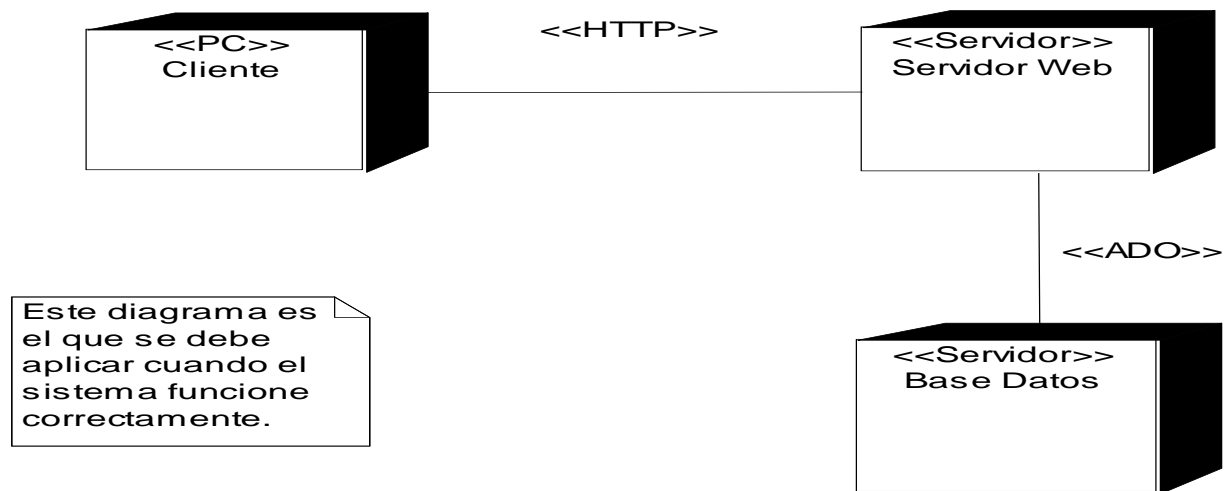


#### 4.3 - Diagrama de despliegue

El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

La mayoría de las veces el modelado de la vista de despliegue implica modelar la topología del hardware sobre el que se ejecuta el sistema. Aunque UML no es un lenguaje de especificación hardware de propósito general, se ha diseñado para modelar muchos de los aspectos hardware de un sistema a un nivel suficiente para que un ingeniero software pueda especificar la plataforma sobre la que se ejecuta el software del sistema. [46]

### 4.3.1 - Diagrama de despliegue



### 4.4 - Conclusiones

En este capítulo fue presentado cómo está construido el sistema a partir de los diagramas de componentes de cada caso de uso, así como, los diagramas de componentes de todo el sistema en general. También fue mostrado el diagrama de despliegue, el cual ilustra cuáles serán los nodos que serán usados para la implantación de la aplicación.

## CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD

### 5.1 - Introducción

En la realización de un proyecto es muy importante realizar la estimación del esfuerzo humano, el tiempo de desarrollo que se requiere así como su costo. En el presente capítulo se llevará a cabo un estudio de la factibilidad del sistema utilizando el método de análisis por Punto de Casos de Uso.

### 5.2 - Planificación mediante Puntos de Casos de Uso

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objector y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

**Cálculo de Puntos de Caso de Uso sin ajustar.**

$$UUCP = UAW + UUCW$$

Donde:

**UUCP:** Puntos de Casos de Uso sin ajustar.

**UAW:** Factor de Peso de los Actores sin ajustar.

**UUCW:** Factor de Peso de los Casos de Uso sin ajustar.

**Calculando UAW: Factor de Peso de los actores.**

Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Descripción	Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3

En este caso son 4 actores de tipo complejo, Administrador del Sistema, Vicedecano de Producción, Líder de Proyecto y Documentador.

$$UAW = 4 * 3$$

$$UAW = 12$$

**Calculando UUCW: Factor de Peso de los Casos de Uso sin ajustar.**

Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo.

Una transacción está representada por uno o más pasos del flujo de eventos principal del caso de uso, pudiendo existir más de una transacción dentro del mismo caso de uso.

Tipo de Caso de Uso	Descripción	Peso
Simple	El caso de uso contiene de 1 a 3 transacciones.	5
Medio	El caso de uso contiene de 4 a 7 transacciones.	10
Complejo	El caso de uso contiene más de 8 transacciones.	15

Caso de Uso	Transacciones	Peso
Autenticar Usuarios	2	5
Gestionar Usuarios	4	10
Gestionar Proyecto	8	15
Gestionar Plan Módulos	2	5
Gestionar Información Módulo Temas	7	10
Gestionar Información Módulo Galería	4	10
Gestionar Información Módulo Ejercicios	7	10
Obtener Reporte	2	5
Estructurar Contenido	1	5

$$UUCW = 4*5+4*10+1*15=75$$

**Calculo de Puntos de Caso de Uso sin ajustar.**

$$UUCP = UAW + UUCW$$

**UUCP = 12+75**

**UUCP = 87**

**Calculando los Puntos de Caso de Uso ajustados.**

Los puntos de Casos de Uso se calculan con la siguiente fórmula:

**UCP = UUCP \* TCF \* EF**

**Donde:**

**UCP:** Puntos de Casos de Uso ajustados.

**UUCP:** Puntos de Casos de Uso sin ajustar.

**TCF:** Factor de complejidad técnica.

**EF:** Factor ambiente.

**Calculando TCF: Factor de complejidad técnica.**

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

En la siguiente tabla se muestran los factores, el peso de cada uno de ellos y el valor asignado:

Factor	Descripción	Peso	Valor asignado	Comentario
T1	Sistema distribuido.	2	3	El sistema será utilizado desde muchas máquinas a la vez.
T2	Tiempo de respuesta.	1	3	La velocidad es limitada por la conectividad.
T3	Eficiencia del usuario final.	1	3	Requiere de conocimiento para tratar con la tecnología y el sistema.
T4	Procesamiento interno complejo.	1	2	No existen cálculos complejos.
T5	El código debe ser reutilizable.	1	2	El código será reutilizable.
T6	Facilidad de instalación.	0.5	3	No conlleva a una instalación compleja.
T7	Facilidad de uso.	0.5	4	Gran facilidad de uso.



T8	Portabilidad.	2	1	No se requiere que el sistema sea portable.
T9	Facilidad de cambio.	1	2	Costo moderado de mantenimiento.
T10	Concurrencia.	1	2	Existe concurrencia.
T11	Incluye objetivos especiales de seguridad.	1	3	Sistema de autenticación.
T12	Provee acceso directo a terceras partes.	1	0	No presenta terceras partes.
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	3	Presenta facilidad de uso pero por su importancia se adiestrarán.

$$TCF = 0.6 + 0.01 * \sum (P_i * V_i)$$

$$TCF = 0.6 + 0.01 * 31.5$$

$$TCF = 0.189$$

**Calculando EF: Factor de ambiente.**

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Factor	Descripción	Peso	Valor asignado	Descripción
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	2	El grupo está familiarizado con el modelo.
E2	Experiencia en la aplicación.	0.5	0	La mayoría del grupo ha trabajado tiempo en ésta aplicación.

E3	Experiencia en orientación a objetos.	1	3	La mayoría del grupo programa en objetos.
E4	Capacidad del analista líder.	0.5	0	El líder es un Ing. Informática.
E5	Motivación.	1	3	Existe motivación del grupo.
E6	Estabilidad de los requerimientos.	2	4	Existe probabilidad de cambios.
E7	Personal part-time.	-1	3	Todo el grupo trabaja a tiempo completo.
E8	Dificultad del lenguaje de programación.	-1	3	Se usará lenguaje PHP.

$$EF = 1.4 - 0.03 * \sum (Pi * Vi)$$

$$EF = 1.4 - 0.03 * (11)$$

$$EF = 1.07$$

#### Calculando lo Puntos de Caso de Uso ajustados

$$UCP = UUCP * TCF * EF$$

$$UCP = 87 * 0.189 * 1.07$$

$$UCP = 17.6$$

#### De los Puntos de Casos de Uso a la estimación del esfuerzo.

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CP$$

**Donde:**

**E:** Esfuerzo estimado en horas-hombre.

**UCP:** Puntos de Casos de Uso ajustados.

**CP:** Factor de conversión.

#### Para calcular el factor de conversión es necesario tener como guía el siguiente criterio:

Se contabilizan cuantos factores de los que afectan al factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.

Se contabilizan cuántos factores de los que afectan al factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

- Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Resumiendo:

CP = 20 horas-hombre (si Total EF  $\leq$  2)

CP = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CP = abandonar o cambiar proyecto (si Total EF  $\geq$  5)

En el presente caso existen 3 factores por debajo de 3 desde E1 hasta E6 y solo un factor de E7 y E8 esta por encima de 3, por lo que el total de factores a tener en cuenta es 4 y el factor conversión sería de 28 horas-hombre.

Calculando el esfuerzo del flujo de trabajo:

$$E = UCP * CP$$

$$E = 17.6 * 28$$

$$E = 492.8 \text{ horas-hombre.}$$

Se debe tener en cuenta que éste método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso.

Finalmente, para una estimación más completa de la duración total del producto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Además se considera que este esfuerzo representa un porcentaje del esfuerzo total del proyecto, de acuerdo a los valores porcentuales:

Actividad	Porcentaje	Horas/Hombre
Análisis	30 %	493
Diseño	40 %	657
Programación	30 %	493
Total	100 %	1643

#### Convirtiendo a Horas/Hombre

Como la jornada laboral de un día de trabajo es de 8 horas y en un mes se trabaja un aproximado de 24 días, entonces una persona en 1 mes trabaja 192 horas, por tanto:

$$Et = E \text{ (Horas-Hombres)}/192\text{horas-mes}$$

Quedaría,  $Et = 1643 \text{ horas} - \text{Hombres} / 192 \text{ horas} - \text{mes} = 8.56 = 9 \text{ mes} - \text{hombres}$ .

Si en el proyecto trabajan dos hombres entonces el tiempo de desarrollo es:

Tiempo de desarrollo =  $Et / \text{cantidad de hombres}$

Tiempo de desarrollo =  $9 / 2 = 4.5 \text{ meses}$

El tiempo a emplear para el desarrollo de la aplicación es de 4.5 meses.

### **Salario**

Para determinar el salario mensual se tiene en cuenta que los desarrolladores del sistema pueden ser ingenieros recién graduados, por lo que se toma como salario mensual: \$349.

### **5.3 - Costo**

Como se definió anteriormente el salario promedio de un ingeniero, y son dos personas los desarrolladores del proyecto, entonces para hallar el costo total sería:

$Ct = \text{Salario mensual} * \text{Cantidad de hombres} * \text{Tiempo de desarrollo}$

$Ct = \$349 * 2 * 4.5$

$Ct = \$3141$

### **5.4 - Beneficios tangibles e intangibles**

#### **5.4.1 - Tangibles**

Teniendo en cuenta que la aplicación en cuestión no es un producto desarrollado para la comercialización, ya que surge con la idea de apoyar en la facultad el proceso de gestión de la información de los software con tecnología multimedia no es válido mencionar beneficios económicos.

Se puede decir que el costo por desarrollar la aplicación es de \$3141 MN (moneda nacional) o 126CUC (convertible), el cual es perfectamente reparable si en un futuro se comercializara.

#### **5.4.2 – Intangibles**

Como beneficios intangibles de la aplicación Gestión de Información de Software con Tecnología Multimedia se señalan los siguientes:

- Se agiliza el proceso de gestión de la información del software con tecnología multimedia.
- Permite garantizar una mayor centralización de la información de los software con tecnología multimedia que se desarrollan en la facultad.
- Se obtiene un fichero XML que facilita el trabajo a los programadores a la hora de desarrollar un software con tecnología multimedia.
- Estandarización de los software con tecnología multimedia.

### **5.5 - Análisis costo-beneficio**

Dadas las características de la aplicación se puede decir que se requiere de un tiempo aceptable para el desarrollo del software, además no necesita de grandes gastos de recursos ya que será un software de tamaño pequeño.

El ambiente del software se ve favorecido con un diseño que permite al usuario desarrollar una perfecta navegabilidad sin motivos de perderse dentro del mismo.

Es factible desarrollar una aplicación para informatizar y centralizar la información de los software con tecnología multimedia, ya que agiliza el proceso de gestión de la información de las multimedias así como la implementación de las mismas.

### **5.6 - Conclusiones**

En este capítulo se realiza un estudio de la factibilidad del producto, se puntualizan los costos a incurrir, los recursos humanos implicados, el tiempo de desarrollo y los beneficios intangibles que aporta la terminación del producto "Gestión de la Información de los Software con Tecnología Multimedia". Además con la realización de este capítulo se expresa claramente la ventaja que implica la implementación de este producto.

Los resultados obtenidos al concluir los cálculos y estimaciones realizadas en este capítulo, se muestran en la siguiente tabla:

<b>Parámetros</b>	<b>Valores</b>
Esfuerzo	28 mes-hombre
Tiempo de desarrollo	4.5 meses
Cantidad de hombres	2 hombres
Salario	\$225
Costo Total MN	\$2025
Costo Total CUC	\$81

## **CONCLUSIONES GENERALES**

Como resultado de este trabajo se dio cumplimiento a los objetivos trazados, implementándose una aplicación web al nivel requerido, siguiendo los requisitos solicitados por el cliente. Dicha aplicación proporciona una mayor eficiencia en el proceso de gestión de la información, brinda una mayor centralización de la información a la hora de almacenarla, se estructura un archivo XML con el cual los programadores ganan en tiempo de implementación ya que no deben preocuparse por la realización del mismo.

A partir del análisis de las tendencias y tecnologías actuales que se utilizan para el desarrollo de aplicaciones informáticas, se elaboró como propuesta la utilización de PHP como lenguaje de programación del lado del servidor, del lado del cliente javascript y CSS, además de utilizar MySQL como gestor de base datos y Apache como servidor web para la construcción de la propuesta de solución, en conjunto con otras herramientas.

Se optó como metodología idónea para llevar a cabo todo el proceso de desarrollo del software a RUP (Proceso Unificado de Rational), utilizando las representaciones de UML para modelar las distintas fases por las que transita el proyecto que se desea desarrollar.

Se realizó la modelación del negocio propuesto, definiendo los actores y trabajadores así como las actividades que son objeto de automatización. Se definieron los requisitos funcionales y no funcionales del sistema para la posterior estructuración de los casos de uso del sistema.

Se realizó el análisis y diseño del sistema a través de diagramas de clases web, clases persistentes, entre otros. Se generó el modelo de datos a partir del diagrama de clases persistentes, el cual muestra la representación física de la base de datos del sistema.

Además se plantearon los principios de diseño a través de los estándares de interfaz de aplicación, de codificación, así como los patrones de diseño y tratamientos de errores. Se construyó finalmente el diagrama de despliegue y componentes.

Luego de este largo proceso de trabajo se puede decir que el sistema brinda la solución requerida ante la problemática que la originó y que su utilización representará una mejora considerable en el control y eficiencia de los procesos que automatiza dentro del flujo de trabajo de Gestión de la Información para los Software con Tecnología Multimedia.

## **RECOMENDACIONES**

- Continuar con el estudio y análisis del proceso de Gestión de la Información de los software con tecnología multimedia en la facultad 8 de la Universidad de Ciencias Informáticas, con el objetivo de ampliar las funcionalidades del sistema que se propone.
- Integrar el sistema propuesto con el IDE para multimedia que se está desarrollando en el polo de Software Educativo.
- Crear un proyecto I+D en la facultad que amplíe el desarrollo del sistema propuesto.
- Realizar un estudio para una posible adaptación de un framework para la continuidad del sistema.

## **BIBLIOGRAFÍA**

- Conallen, J. (2000). "Modeling web application architectures with UML"
- Espinosa, J. (2002). "Introducción a n-Capas con VFP y VB." Retrieved 12 de mayo, 2008, from <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art20.asp>
- Gussy. (2006). "Sistemas informáticos. N-tier. Arquitecturas empresariales. Desarrollo de aplicaciones. Sistema de negocios integrados. E-bussines." Retrieved 21 de febrero, 2008, from <http://apuntes.rincondelvago.com/desarrollo-de-n-capas.html>
- Lanvin, D. F. (2005). "Arquitectura Web." Retrieved 1 de junio, 2008, from <http://www.di.uniovi.es/~dflanvin/docencia/dasdi/teoria/Transparencias/06.%20Arquitectura%20Web.pdf>
- Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda Edición por Prentice Hall.
- MARRERO, D. Modelado de aplicaciones Web con UML. En: Conferencia de Ingeniería de Software, Diciembre 2002, ISPJAE (CEIS).
- Martínez, R. (2003). Manual de PHP
- Matos, Rosa María. Introducción al trabajo con Base de Datos. Asignatura de Sistemas de Gestión de Base de Datos.
- Rational, C. (2003). Ayuda Ampliada de RUP. Estados Unidos.
- Sánchez, M. A. M. (2004). "Metodologías De Desarrollo De Software." Retrieved 1 de abril, 2008, from [http://www.informatizate.net/articulos/pdfs/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.pdf](http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf)
- Senarega, L. L. T. A. M. (2006). SISTEMA DE GESTIÓN DE CONTENIDOS XML.



## REFERENCIAS BIBLIOGRÁFICAS

1. *Escripnauta: sistema de creación y generación de presentaciones multimedia*. [Online] [Cited: 12 4, 2007.] <http://mnm.uib.es/gallir/escripnauta/escripnauta.pdf>.
2. *Malted*. [Online] [Cited: 12 4, 2007.] <http://malted.cnice.mecd.es/presentacion/QMalted.htm>.
3. *Macromedia Breeze*. [Online] [Cited: 12 4, 2007.] <http://www.desarrolloweb.com/articulos/1502.php>.
4. *Agencia Española de Protección de Datos*. [Online] [Cited: 12 2, 2007.]  
<https://www.agpd.es/index.php?idSeccion=541>.
5. *Diccionario Informático*. [Online] [Cited: 1 10, 2008.]  
<http://www.monografias.com/trabajos22/diccionario-informatico/diccionario-informatico.shtml>.
6. *Glosario de términos de marketing en Internet*. [Online] [Cited: 2 4, 2008.] <http://www.marketing-xxi.com/glosario-de-terminos-de-marketing-en-internet-149.htm>.
7. [Online] [Cited: 12 3, 2007.] <http://www.inei.gob.pe/biblioineipub/bancopub/inf/Lib5038/indice.htm>.
8. *Características de PHP*. [Online] [Cited: 2 6, 2008.]  
<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
9. *Características JSP*. [Online] [Cited: 12 4, 2007.]  
[www.dei.inf.uc3m.es/docencia/p\\_s\\_ciclo/pa4/practicas/jsp.pdf](http://www.dei.inf.uc3m.es/docencia/p_s_ciclo/pa4/practicas/jsp.pdf).
10. *Desarrollo Web*. [Online] [Cited: 12 5, 2007.] <http://www.desarrolloweb.com/articulos/541.php>.
11. *Diferencias entre PHP ASP y otros lenguajes*. [Online] [Cited: 2 6, 2008.]  
[http://www.tufuncion.com/diferencias\\_lenguajes](http://www.tufuncion.com/diferencias_lenguajes).
12. *Introducción a Internet Information Server*. [Online] [Cited: 2 4, 2008.]  
<http://www.adrformacion.com/curso/intranet/leccion1/InternetInformationServer.htm>.
13. *Glosario de términos sobre Internet y Spam*. [Online] [Cited: 2 4, 2008.]  
<https://www.agpd.es/index.php?idSeccion=541>.
14. *Una Introducción a APACHE*. [Online] [Cited: 2 4, 2008.]  
[http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/).
15. *Características de IIS*. [Online] [Cited: 2 4, 2008.]  
<http://fferrer.dsic.upv.es/cursos/Windows/Avanzado/ch08s02.html>.
16. *Información del Producto*. [Online] [Cited: 2 6, 2008.] [http://www.software-shop.com/in.php?mod=ver\\_producto&prdID=217](http://www.software-shop.com/in.php?mod=ver_producto&prdID=217).
17. *Glosario SQL*. [Online] [Cited: 2 7, 2008.]  
[http://www.yoled.com/ys\\_htm\\_art\\_glosario\\_sql.php#LetraO](http://www.yoled.com/ys_htm_art_glosario_sql.php#LetraO).
18. *Oracle Database*. [Online] [Cited: 2 7, 2008.]  
<http://www.oracle.com/global/es/products/database/index.html>.

19. *SQL Server*. [Online] [Cited: 2 7, 2008.]  
<http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml>.
20. *Apache, PHP y MySQL*. [Online] [Cited: 2 7, 2008.]  
<http://ticscolegionatales.blogspot.com/2007/04/plataforma-xampp-3-medio-ferrer.html>.
21. *Pequeño Diccionario Informático*. [Online] [Cited: 2 6, 2008.]  
<http://www.monografias.com/trabajos22/diccionario-informatico/diccionario-informatico.shtml>.
22. *¿Qué es Javascript?* [Online] [Cited: 2 6, 2008.]  
<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
23. *Diccionario de siglas*. [Online] [Cited: 12 2, 2007.]  
<http://www.elcodigo.net/tutoriales/diccionario.html>.
24. *Glosario*. [Online] [Cited: 2 6, 2008.] <http://webmaster.lycos.es/glossary/D/>.
25. *Dreamweaver*. [Online] [Cited: 2 6, 2008.] <http://www.desarrolloweb.com/articulos/332.php>.
26. *Editores web que facilitan tu trabajo*. [Online] [Cited: 2 7, 2007.]  
<http://www.radiocaribe.co.cu/secundaria/informatica/342.htm>.
27. *Glosario Formador Ocupacional*. [Online] [Cited: 2 4, 2008.]  
<http://opeatal.com/formacion/mod/glossary/view.php?id=377&mode=author&hook=ALL&sortkey=FIRSTNAME&sortorder=asc&fullsearch=0&page=6>.
28. *Rational Unified Process*. [Online] [Cited: 2 7, 2008.]  
<http://www.rational.com.ar/herramientas/rup.html>.
29. *Metodologías* . [Online] [Cited: 2 4, 2008.]  
[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).
30. *Ingeniería de SoftwareUML*. [Online] [Cited: 2 7, 2008.]  
<http://www.monografias.com/trabajos5/insof/insof.shtml>.
31. *Herramientas Case*. [Online] [Cited: 2 4, 2008.]  
<http://www.monografias.com/trabajos14/herramicase/herramicase.shtml>.
32. *Slideshow transcript*. [Online] [Cited: 2 6, 2008.] [http://www.slideshare.net/vivi\\_jocadi/rational-rose/](http://www.slideshare.net/vivi_jocadi/rational-rose/).
33. *Rational Rose Enterprise*. [Online] [Cited: 2 6, 2008.]  
<http://www.rational.com.ar/herramientas/roseenterprise.html>.
34. *Visual Paradigm International*. [Online] [Cited: 2 6, 2008.]  
[http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(Iglesia\\_Anglicana\)\\_%5BMac\\_OS\\_X\\_cuenta\\_14717\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/).
35. *Diccionario básico de Informática*. [Online] [Cited: 2 4, 2008.]  
<http://usuarios.lycos.es/Resve/diccioninform.htm>.

36. *Glosario sobre Linux*. [Online] [Cited: 2 4, 2008.] <http://www.escomposlinux.org/glosario/>.
37. *Glosario*. [Online] [Cited: 2 7, 2008.] <http://webmaster.lycos.es/glossary/R/>.
38. *glosario-de-blogging*. [Online] [Cited: 2 10, 2008.] <http://www.blogmundi.com/glosario-de-blogging/>.
39. *Diccionario básico de Informática*. [Online] [Cited: 2 10, 2008.] <http://www.nachocabanes.com/diccio/ndic.php#C>.
40. Jacobson, I., G. Booch, and J. Rumbaugh. *El Proceso Unificado de Desarrollo de software*. 2000.
41. *Glosario SQL*. [Online] [Cited: 4 1, 2008.] [http://www.yoled.com/ys\\_htm\\_art\\_glosario\\_sql.php#LetraP](http://www.yoled.com/ys_htm_art_glosario_sql.php#LetraP).
42. *Características de PostgreSQL*. [Online] [Cited: 4 1, 2008.] <http://www.sobl.org/traduccion/practical-postgres/node19.html>.
43. Curbelo, Fidel. *Sistema de Gestión de Información de la facultad 8. Módulo para la Gestión de la Residencia Estudiantil*. Ciudad Habana : s.n., 2008.
44. —. *Sistema de Gestión de la Información de la Facultad 8. Módulo para la Gestión de la Residencia Estudiantil*.
45. —. *Sistema de Gestión de la Información de la Facultad 8. Módulo para la Gestión de la Residencia Estudiantil*. Ciudad Habana : s.n., 2008.
46. *Diagrama de despliegue*. [Online] [Cited: 5 12, 2008.] <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>.
47. *El mejor programa de diseño gráfico para Web*. [Online] [Cited: 5 24, 2008.] <http://adobe-fireworks.softonic.com/>.
48. *Patrón Modelo-Vista-Controlador*. [Online] [Cited: 5 24, 2008.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
49. *Patrón Modelo-Vista-Controlador*. [Online] [Cited: 5 24, 2008.] [http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_3317.as](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.as).
50. *UML*. [Online] [Cited: 06 17, 2008.] <http://www.scribd.com/doc/2080534/UML>.

## **GLOSARIO DE TÉRMINOS**

**BSD:** Es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Pertenece al grupo de licencias de software Libre. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre. (35)

**GPL:** (Licencia Pública General) es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. (36)

**RDBMS:** Relational Database Management System (Sistema de gestión de bases de datos relacionales). (37)

**PLUGINS:** Pequeños archivos que añaden mayores funcionalidades y características a tu blog. (38)

**CGI:** Pequeños programas situados en un servidor Web, para poder ser utilizados desde las páginas Web que se hospeden en dicho servidor (Common Gateway Interface). (39)

**E-LEARNING:** Término que cubre un amplio grupo de aplicaciones y procesos, tales como aprendizaje basado en web, aprendizaje basado en ordenadores, aulas virtuales y colaboración digital. Incluye entrega de contenidos vía Internet, intranet/extranet, audio y vídeo grabaciones, transmisiones satelitales, TV interactiva, CD-ROM y más. (39)

**CORE:** Núcleo.

**HOSTING:** El alojamiento web (en inglés web hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía Web. Los Web Host son compañías que proporcionan espacio de un servidor a sus clientes. (35)