

**Universidad de las Ciencias Informáticas**

**Facultad 8**

**Análisis y diseño del sistema de información y  
gestión para la calidad del Software Educativo**

**Trabajo de diploma para optar por el título de  
Ingeniería en Ciencias Informáticas**

**Autora:**

**Yudisney Vázquez Ortiz**

**Tutora:**

**Ing. Licet Gutiérrez Mompíe**

**La Habana, julio 5, 2007**

**Año 49 de la Revolución**

## **Declaración de auditoría**

Por este medio declaro ser autora de la presente tesis y reconozco 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 \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Yudisney Vázquez Ortiz

---

Ing. Licet Gutiérrez Mompíe

A mis padres que me siguen dando vida.

A mis hermanos, familiares y amigos, incondicionales siempre.

## **Agradecimientos**

A los que me alentaron, ayudaron, sorprendieron  
y confiaron en mí a lo largo de toda mi vida estudiantil.

A la Revolución Cubana y a todos los que han luchado por ella,  
permitiendo que haya llegado hasta este nivel sin otra preocupación que mis estudios.

## Resumen

La Universidad de las Ciencias Informáticas (UCI) está realizando numerosos esfuerzos para que sus producciones tengan la calidad requerida. La facultad 8 en el afán de obtener Software Educativos (SE) con alta calidad, requiere la creación de un sistema con el fin de socializar información y gestionar los entregables mínimos para el logro de este propósito; constituyendo la realización del Análisis & Diseño de este sistema el primer paso en aras de alcanzarlo y por tanto el objetivo fundamental de la presente investigación. Para la realización de la propuesta se utilizaron: PHP como lenguaje de programación, *MySQL* como gestor de base de datos (BD), *Apache* como servidor Web, *Rational Unified Process* (RUP) como metodología de desarrollo y como parte esencial de ella *Unified Modeling Language* (UML), que permitió la creación de un conjunto de artefactos que facilitarán la implementación exitosa del sistema.

# Índice de contenidos

<b>INTRODUCCIÓN</b>	1
<b>CAPÍTULO 1.- FUNDAMENTACIÓN TEÓRICA</b>	4
Introducción	4
1.1.- Definición de calidad	4
1.1.1.- Calidad de software	5
1.1.1.1.- Características del software	5
1.1.1.2.- Gestión de calidad	6
1.1.1.3.- Costo de calidad	7
1.2.- Software gestor de calidad	9
1.2.1.- Definición de gestión	9
1.2.2.- Sistema de gestión de la calidad	10
1.2.3.- Sistemas de información de gestión	10
1.3.- Algunos gestores de calidad de software existentes en el mundo	11
1.3.1.- Anaconda	11
1.3.2.- Qualysoft	11
1.4.- Tendencias y tecnologías actuales utilizadas	13
1.4.1.- Plataforma de desarrollo	13
1.4.1.1.- Aplicaciones de escritorio	13
1.4.1.2.- Aplicaciones Web	13
1.4.1.3.- Fundamentación de la selección de la plataforma a utilizar	14
1.4.2.- Sistema de gestión de contenido	15
1.4.3.- Lenguaje de desarrollo Web	16
- Practical Extraction Report Language	16
- Active Server Pages	16
- Hypertext Pre-processor	17
1.4.3.1.- Fundamentación de la selección del lenguaje a utilizar	17
1.4.4.- Servidor Web	19
- Internet Information Services/Server	19
- Apache	19
1.4.4.1.- Fundamentación de la selección del servidor Web a utilizar	20
1.4.5.- Gestor de base de datos	20

- Microsoft SQL Server _____	20
- Oracle _____	21
- MySQL _____	21
- PostgreSQL _____	22
1.4.5.1.- Fundamentación de la selección del gestor de base de datos a utilizar _____	22
1.4.6.- Otras herramientas necesarias _____	23
1.4.7.- Metodología a utilizar _____	24
1.4.7.1.- Definición de metodología _____	24
1.4.7.2.- Proceso Unificado del Rational _____	25
Conclusiones _____	27
<b>CAPÍTULO 2.- CARACTERÍSTICAS DEL SISTEMA _____</b>	<b>29</b>
Introducción _____	29
2.1.- Descripción de los procesos de negocio _____	29
2.2.- Descripción del sistema propuesto _____	30
2.3.- Modelado de dominio _____	31
2.3.1.- Glosario de términos _____	31
2.4.- Requerimientos funcionales _____	32
2.5.- Requerimientos no funcionales _____	33
2.6.- Modelo de Casos de Usos _____	35
2.6.1.- Determinación de los Casos de Usos _____	36
2.7.- Expansión de los Casos de Usos _____	38
Conclusiones _____	58
<b>CAPÍTULO 3.- ANÁLISIS Y DISEÑO DEL SISTEMA _____</b>	<b>59</b>
Introducción _____	59
3.1.- Modelo de análisis _____	59
3.2.- Modelo de diseño _____	61
3.2.1.- Descripción de las clases _____	63
3.2.1.1.- Clases entidad _____	63
3.2.1.2.- Clases de acceso a datos _____	64
3.2.1.3.- Clases manager _____	67
3.2.1.4.- Clase fachada _____	67
3.2.1.5.- Clases lógica de interfaz de usuario _____	67

3.2.1.6.- Clases interfaz de usuario _____	68
3.2.1.7.- Otras clases _____	68
3.2.2.- Diagramas de colaboración por escenarios _____	70
3.2.3.- Diagramas de clases Web por Casos de Uso _____	74
3.2.4.- Diseño de la base de datos _____	78
3.2.4.1.- Descripción de las tablas de la base de datos _____	80
3.2.5.- Principios de diseño _____	82
3.2.6.- Interfaz de la aplicación _____	83
3.2.7.- Tratamiento de errores _____	84
3.2.8.- Seguridad _____	84
Conclusiones _____	85
<b>CAPÍTULO 4.- IMPLEMENTACIÓN Y PRUEBA _____</b>	<b>86</b>
Introducción _____	86
4.1.- Implementación _____	86
4.2.- Prueba _____	89
Conclusiones _____	99
<b>CONCLUSIONES GENERALES _____</b>	<b>100</b>
<b>RECOMENDACIONES _____</b>	<b>101</b>
<b>REFERENCIAS BIBLIOGRÁFICAS _____</b>	<b>102</b>
<b>BIBLIOGRAFÍA _____</b>	<b>105</b>
<b>GLOSARIO DE TÉRMINOS GENERAL _____</b>	<b>106</b>

## Índice de tablas

Tabla 1.- Comparación entre PHP, ASP y Perl	17
Tabla 2.- Servidores Web más utilizados en octubre-noviembre de 2006	20
Tabla 3.- Comparación entre MS SQL Server, Oracle, MySQL y PostgreSQL	22
Tabla 4.- Trabajadores del dominio	32
Tabla 5.- Entidades del dominio	32
Tabla 6.- Justificación de los actores del sistema	35
Tabla 7.- CU Gestionar información	36
Tabla 8.- CU Gestionar proyecto	36
Tabla 9.- CU Gestionar usuario	36
Tabla 10.- CU Gestionar autenticación	37
Tabla 11.- CU Actualizar proyecto	37
Tabla 12.- CU Gestionar entregable	37
Tabla 13.- CU Gestionar reporte	37
Tabla 14.- CU Definir documento	37
Tabla 15.- Expansión de Gestionar información	38
Tabla 16.- Expansión de Gestionar proyecto	39
Tabla 17.- Expansión de Gestionar usuario	43
Tabla 18.- Expansión de Gestionar autenticación	47
Tabla 19.- Expansión de Actualizar proyecto	50
Tabla 20.- Expansión de Gestionar entregable	52
Tabla 21.- Expansión de Generar reporte	55
Tabla 22.- Expansión de Definir documento	56
Tabla 23.- Clase ad_Usuario	64
Tabla 24.- Clase ad_Proyecto	65
Tabla 25.- Clase ad_Entregable	65
Tabla 26.- Clase ad_Media	65
Tabla 27.- Clase ad_Imagen	65
Tabla 28.- Clase ad_SonidoLocucionAudio	65
Tabla 29.- Clase ad_Video	66
Tabla 30.- Clase ad_ArchivoAuxiliar	66
Tabla 31.- Clase ad_ModelacionPedagogica	66
Tabla 32.- Clase ad_AlcanceProducto	66
Tabla 33.- Clase ad_DefinicionEntregable	66
Tabla 34.- Clase ad_Entidad	66
Tabla 35.- Clase ad_Contacto	66
Tabla 36.- CUs asociados a las clases manager	67
Tabla 37.- Descripción de las clases lógica de interfaz de usuario	67
Tabla 38.- Descripción de otras clases utilizadas	68
Tabla 39.- Descripción de la tabla tbRol	80
Tabla 40.- Descripción de la tabla tbUsuario	80
Tabla 41.- Descripción de la tabla tbResponsabilidad_Proyecto	80
Tabla 42.- Descripción de la tabla tbProyecto	80
Tabla 43.- Descripción de la tabla tbEntregable	80
Tabla 44.- Descripción de la tabla tbEtapa	80
Tabla 45.- Descripción de la tabla tbMedia	81
Tabla 46.- Descripción de la tabla tbTipo_Media	81
Tabla 47.- Descripción de la tabla tblImagen	81

Tabla 48.- Descripción de la tabla tbSonidoLocucionAudio_____	81
Tabla 49.- Descripción de la tabla tbVideo _____	81
Tabla 50.- Descripción de la tabla tbAlcance_Producto _____	81
Tabla 51.- Descripción de la tabla tbEntidad _____	82
Tabla 52.- Descripción de la tabla tbDefinicion_Entregable _____	82
Tabla 53.- Descripción de la tabla tbContacto_____	82
Tabla 54.- Descripción de la tabla tbArchivo_Auxiliar _____	82
Tabla 55.- Condiciones de entrada para Gestionar proyecto _____	89
Tabla 56.- Caso de prueba: Crear proyecto entrando datos correctamente _____	89
Tabla 57.- Caso de prueba: Crear proyecto entrando datos incorrectamente _____	89
Tabla 58.- Caso de prueba: Buscar proyectos sin seleccionar un criterio _____	90
Tabla 59.- Caso de prueba: Buscar proyectos seleccionando el criterio de nombre/estado/ambos _____	90
Tabla 60.- Caso de prueba: Cambiar estado de proyecto(s) seleccionado(s) _____	90
Tabla 61.- Caso de prueba: Cambiar estado de proyecto(s) sin seleccionar _____	90
Tabla 62.- Caso de prueba: Eliminar proyecto(s) seleccionado(s) _____	90
Tabla 63.- Caso de prueba: Eliminar proyectos sin seleccionar _____	90
Tabla 64.- Condiciones de entrada para Gestionar usuario _____	91
Tabla 65.- Caso de prueba: Crear usuario con responsabilidad en facultad entrando datos correctamente _____	91
Tabla 66.- Caso de prueba: Crear usuario con responsabilidad en proyecto entrando datos correctamente. _____	91
Tabla 67.- Caso de prueba: Crear usuario dejando de entrar al menos un dato _____	91
Tabla 68.- Caso de prueba: Crear usuario entrando incorrectamente el nombre/usuario/contraseña _____	92
Tabla 69.- Caso de prueba: Buscar usuarios sin seleccionar un criterio de búsqueda. _	92
Tabla 70.- Caso de prueba: Buscar usuarios seleccionando el criterio de nombre y apellidos/usuario/rol/cualquier combinación de ellos _____	92
Tabla 71.- Caso de prueba: Modificar perfil de varios elementos seleccionados _____	92
Tabla 72.- Caso de prueba: Modificar perfil de un elemento seleccionado entrando datos correctamente _____	92
Tabla 73.- Caso de prueba: Modificar perfil de un elemento seleccionado entrando incorrectamente el nombre y apellidos/contraseñas _____	93
Tabla 74.- Caso de prueba: Eliminar usuarios seleccionados _____	93
Tabla 75.- Caso de prueba: Eliminar usuarios sin seleccionar _____	93
Tabla 76.- Condiciones de entrada para Gestionar autenticación _____	93
Tabla 77.- Caso de prueba: Autenticar usuario entrando correctamente los datos _____	93
Tabla 78.- Caso de prueba: Autenticar usuario dejando de entrar al menos un dato ____	93
Tabla 79.- Caso de prueba: Autenticar usuario entrando uno no registrado _____	94
Tabla 80.- Caso de prueba: Autenticar usuario entrando contraseña incorrecta _____	94
Tabla 81.- Caso de prueba: Modificar contraseña entrando correctamente los datos ____	94
Tabla 82.- Caso de prueba: Modificar contraseña dejando de entrar al menos un dato _	94
Tabla 83.- Caso de prueba: Modificar contraseña entrando erróneamente la anterior __	94
Tabla 84.- Caso de prueba: Modificar contraseña entrando las nuevas distintas _____	94
Tabla 85.- Condiciones de entrada para Actualizar proyecto _____	95
Tabla 86.- Caso de prueba: Subir archivo entrando la dirección correctamente _____	95
Tabla 87.- Caso de prueba: Subir archivo dejando la dirección vacía _____	95
Tabla 88.- Condiciones de entrada para Gestionar entregable _____	95
Tabla 89.- Caso de prueba: Insertar nuevo entregable entrando los datos correctos ____	95

Tabla 90.- Caso de prueba: Insertar nuevo entregable entrando el número/resolución incorrectamente _____	96
Tabla 91.- Caso de prueba: Insertar nuevo entregable dejando campo(s) vacío(s) _____	96
Tabla 92.- Caso de prueba: Listar entregables existiendo al menos uno en la BD asociado al proyecto del usuario logueado _____	96
Tabla 93.- Caso de prueba: Listar entregables no existiendo ninguno en la BD asociado al proyecto del usuario logueado _____	96
Tabla 94.- Caso de prueba: Modificar entregable de varios elementos seleccionados__	96
Tabla 95.- Caso de prueba: Modificar entregable de un elemento seleccionado entrando datos correctamente _____	96
Tabla 96.- Caso de prueba: Modificar entregable de un elemento seleccionado entrando incorrectamente el número/resolución/dejando al menos un campo vacío ____	97
Tabla 97.- Caso de prueba: Eliminar entregables seleccionados _____	97
Tabla 98.- Caso de prueba: Eliminar entregables sin seleccionar _____	97
Tabla 99.- Condiciones de entrada para Generar reporte _____	97
Tabla 100.- Caso de prueba: Conformar reporte entrando correctamente valor del parámetro _____	98
Tabla 101.- Caso de prueba: Conformar reporte entrando incorrectamente valor del parámetro _____	98
Tabla 102.- Condiciones de entrada para Definir documento _____	98
Tabla 103.- Caso de prueba: Subir nuevo archivo entrando datos correctamente _____	98
Tabla 104.- Caso de prueba: Subir nuevo archivo entrando datos incorrectamente ____	98
Tabla 105.- Caso de prueba: Subir nuevo archivo entrando nombre de archivo registrado previamente _____	99

## Índice de figuras

Figura 1.- Costo relativo de corregir un error	8
Figura 2.- Impacto de la metodología en el desarrollo software	25
Figura 3.- Fases e iteraciones de la metodología RUP	27
Figura 4.- Diagrama de dominio	32
Figura 5.- Modelo de CUs del sistema	38
Figura 6.- Diagrama de clases del Análisis de Gestionar información	59
Figura 7.- Diagrama de clases del Análisis de Gestionar proyecto	59
Figura 8.- Diagrama de clases del Análisis de Gestionar usuario	60
Figura 9.- Diagrama de clases del Análisis de Gestionar Autenticación	60
Figura 10.- Diagrama de clases del Análisis de Actualizar proyecto	60
Figura 11.- Diagrama de clases del Análisis de Gestionar entregable	60
Figura 12.- Diagrama de clases del Análisis de Generar reporte	60
Figura 13.- Diagrama de clases del Análisis de Definir documento	60
Figura 14.- Arquitectura de tres capas	62
Figura 15.- Diagrama de colaboración: Crear proyecto	70
Figura 16.- Diagrama de colaboración: Cambiar estado de proyecto	70
Figura 17.- Diagrama de colaboración: Eliminar proyectos	70
Figura 18.- Diagrama de colaboración: Evolución de proyecto	70
Figura 19.- Diagrama de colaboración: Crear usuario	71
Figura 20.- Diagrama de colaboración: Modificar perfil de usuario	71
Figura 21.- Diagrama de colaboración: Eliminar usuarios	71
Figura 22.- Diagrama de colaboración: Autenticar usuario	71
Figura 23.- Diagrama de colaboración: Modificar contraseña	72
Figura 24.- Diagrama de colaboración: Insertar nuevo archivo	72
Figura 25.- Diagrama de colaboración: Subir archivo	72
Figura 26.- Diagrama de colaboración: Sobrescribir archivo	72
Figura 27.- Diagrama de colaboración: Llenar entregable	73
Figura 28.- Diagrama de colaboración: Modificar entregable	73
Figura 29.- Diagrama de colaboración: Eliminar entregables	73
Figura 30.- Diagrama de colaboración: Conformar reporte	73
Figura 31.- Diagrama de clases Gestionar información	74
Figura 32.- Diagrama de clases Gestionar proyecto	74
Figura 33.- Diagrama de clases Gestionar usuario	75
Figura 34.- Diagrama de clases Gestionar autenticación	75
Figura 35.- Diagrama de clases Web del CU Actualizar proyecto	76
Figura 36.- Diagrama de clases Web del CU Gestionar entregable	76
Figura 37.- Diagrama de clases Web del CU Generar reporte	77
Figura 38.- Diagrama de clases Web del CU Definir documento	77
Figura 39.- Paquete de acceso a entregables	77
Figura 40.- Diagrama de clases persistentes	78
Figura 41.- Modelo de datos	79
Figura 42.- Diagrama de despliegue	86
Figura 43.- Diagrama de componentes de base de datos	88
Figura 44.- Diagrama de componentes del sistema	88
Figura 45.- Diagrama de componentes del módulo Administración	88
Figura 46.- Diagrama de componentes del Módulo Facultad&Proyecto	88
Figura 47.- Diagrama de componentes de terminales clientes	88

## **Introducción**

La UCI es un centro que complementa la docencia con la actividad productiva, que además ha adquirido en estos casi cinco años valiosos contratos con empresas tanto cubanas como extranjeras, convirtiéndose en parte importante de la Industria Cubana del Software.

Precisamente la Industria Cubana del Software como parte de la Mundial, está inmersa en la actualidad en una oleada de calidad producto de las exigencias cada vez mayores de los clientes, lo que obliga a elevar la calidad de las producciones.

La UCI es resultado de la profunda revolución cultural que se lleva a cabo en el país y es factor fundamental en dicho proceso. Una de sus líneas fundamentales es la producción de SE que tiene en la facultad 8 uno de sus principales exponentes. Ésta produce en la actualidad libros electrónicos y productos de tecnología multimedia (educativos o no) fundamentalmente, productos que son desarrollados bajo la supervisión de líderes de proyectos encargados de que se cumplan los requisitos para los que el software fue implementado y que juntos a la asesora de calidad y al vicedecano de producción de la facultad, constituyen el equipo de control en la producción de software en la misma. Ellos se enfrentan a tres problemas de forma general:

- Inexperiencia, debido a la juventud de la industria cubana del software y de la universidad y facultad principalmente.
- Inexistencia de información centralizada y actualizada referente a cómo controlar la calidad en el proceso de desarrollo del SE.
- Inexistencia de un repositorio donde se archive la información de los proyectos referente al control de la calidad.

La presente investigación surge como necesidad de dar solución al problema que se deriva de la situación existente actualmente en la Facultad: la falta de un sistema de información y gestión de la calidad del SE disminuye la socialización de la información relacionada con la calidad y su control en los proyectos en la facultad 8.

Tomando como idea a defender que si se desarrolla un sistema de información y gestión de la calidad del SE y se aplica, entonces será factible la socialización de la información y el control relacionado con la calidad de los proyectos en la facultad 8.

Para llevar a cabo la propuesta se define como objeto de estudio el proceso de desarrollo del SE y más específicamente como campo de acción la gestión de la calidad en el SE en la facultad 8.

Para resolver el problema se trazó como objetivo general:

- Analizar y diseñar un sistema de información y gestión de la calidad del SE que permita la socialización de la información relacionada con la calidad y su control en los proyectos en la facultad 8.

Derivando en los objetivos específicos:

1. Investigar sobre lo que se ha hecho, y se hace en el mundo, sobre aplicaciones de gestión de la calidad durante el proceso de desarrollo del SE.
2. Investigar cómo se controla la calidad en el proceso de desarrollo del SE en la universidad y en la facultad 8.
3. Hacer una propuesta del flujo de trabajo Análisis & Diseño para el sistema.
4. Desarrollar una primera iteración del sistema en proposición.

Para llevar a cabo dichos objetivos se trazaron una serie de tareas de investigación:

1. Realizar una búsqueda detallada de los software gestores de información y calidad existentes en el mundo, para conocer lo que implementan y determinar si se puede adaptar alguno a las condiciones de la Facultad.
2. Entrevistar a los implicados con el proceso de desarrollo de SE en la universidad y facultad, para conocer cómo se lleva a cabo realmente la gestión de la calidad en él.
3. Hacer levantamiento de requisitos, diagramas de colaboración, diagramas de clases y los diseños lógico y físico de la BD.
4. Implementar parte de los casos de uso (CUs) significativos.

La investigación está distribuida en cuatro capítulos:

- Capítulo 1.- Fundamentación teórica: se hace un estudio del arte sobre el objeto de estudio y campo de acción de la propuesta, evidenciando la importancia del desarrollo del sistema y la selección de la plataforma, el lenguaje, las herramientas y la metodología con las que se llevará a cabo la tarea.

- Capítulo 2.- Características del sistema: se describen las características del sistema y los requisitos con los que debe cumplir, obteniendo como artefacto fundamental su diagrama de CUs.
- Capítulo 3.- Análisis y diseño del sistema: se comienza la construcción de la propuesta, determinando las clases y la BD que soportarán las funcionalidades del mismo.
- Capítulo 4.- Implementación y prueba: se concluye la construcción de la propuesta, determinando cómo se distribuirá el sistema en términos de hardware y los componentes que será necesario implementar.

# Capítulo 1

## Fundamentación teórica

### Introducción

En este capítulo se abordarán temas relacionados con el sistema en proposición. Se definirán los conceptos de calidad y por qué es vital tanto la obtención de productos con un alto grado de ella como su control durante el proceso de desarrollo del mismo; se determinará la importancia de la existencia de un software como el que se propone desarrollar, así como las herramientas y metodología con las que será implementado.

### 1.1.- Definición de calidad

La calidad ha sido un término muy manejado a lo largo del desarrollo industrial, que además ha evolucionado con el propio desarrollo de la sociedad y del hombre como parte indispensable de ella. Ha sido ésta la que le ha impreso un carácter cada vez más importante, a tal punto que en la actualidad no se concibe un producto de cualquier índole sin calidad elevada; no se ha convertido únicamente en uno de los requisitos esenciales del producto sino en un factor estratégico del que dependen la mayor parte de las organizaciones, no sólo para mantener su posición en el mercado sino para asegurar su supervivencia. *Larousse* define la calidad como “valor propio de un objeto o persona, o superioridad en línea, categoría, importancia”.

Es un término que se encuentra, hoy en día, en multitud de contextos y con el que se busca despertar en quien lo escucha una sensación positiva, transmitiendo la idea de excelencia. El concepto técnico de calidad representa una forma de hacer las cosas en las que predomina la preocupación por satisfacer al cliente y por mejorar procesos y resultados. El concepto actual de calidad ha evolucionado hasta convertirse en una forma de gestión que introduce el enfoque de mejora continua en cualquier organización y a todos los niveles de la misma, y que afecta a todas las personas y a todos los procesos. [ANÓNIMO 2006g]

Como un atributo de un elemento, la calidad se refiere a las características mensurables<sup>1</sup>. Sin embargo, el software en su gran extensión, como entidad intelectual, es más difícil de caracterizar que los objetos físicos. [PRESSMAN 2005]

### **1.1.1.- Calidad de software**

Las empresas dedicadas a la producción de software no quedan exentas de esta oleada de calidad que el mundo exige. Para establecer una definición formal de calidad, lo primero es determinar las características del producto que promueve la empresa a la que se va a aplicar, en este caso, las características del software.

“El software de computadora se ha convertido en el *alma mater*. Es la máquina que conduce a la toma de decisiones comerciales. Sirve de base para la investigación científica moderna y de resolución de problemas de ingeniería. Es el factor clave que diferencia los productos y servicios modernos. Está inmerso en sistemas de todo tipo: de transportes, médicos, de telecomunicaciones, militares, procesos industriales, entretenimientos, productos de oficina..., la lista es casi interminable. El software es casi ineludible en un mundo moderno. A medida que nos adentremos en el siglo XXI, será el que nos conduzca a nuevos avances en todo, desde la educación elemental a la ingeniería genética.” [PRESSMAN 2005]

#### **1.1.1.1.- Características del software**

Existen un conjunto de características esenciales que diferencian al software de otros productos, entre las que se listan: [PRESSMAN 2005]

- Se desarrolla, no se fabrica en un sentido clásico, debido a que es un elemento lógico, no físico.
- No se estropea (en el sentido físico) pero se deteriora, su mantenimiento es considerablemente mayor que el del hardware.
- Aunque la industria tiende a ensamblar componentes, la mayoría del software se construye todavía a medida.

El software se enfrenta a innumerables problemáticas, las fundamentales son: [SCALONE 2006]

---

<sup>1</sup> Cosas que se pueden comparar con estándares conocidos como longitud, color, propiedades eléctricas, maleabilidad.

- El aumento constante de su tamaño y complejidad.
- El carácter dinámico e iterativo a lo largo de su ciclo de vida.
- Los insuficientes procedimientos normalizados para estipular y evaluar calidad, costos y productividad.

Estas características apuntan a que la calidad de un producto tan complejo, se torne evidentemente, un poco compleja de alcanzar, pero al fin y al cabo, ¿qué es la calidad de un software?

Según *Pressman* es la “concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.” [PRESSMAN 2005]

Otra definición apunta a que es el grado en el que el software satisface una serie de requerimientos de operación preestablecidos, los estándares de desarrollo especificados con anterioridad y las características inherentes a todo producto de software desarrollado de manera profesional. [MOMPIÉ 2006]

Ambas definiciones indican que: [PRESSMAN 2005]

- Los requisitos del software son la base de las medidas de la calidad.
- Los estándares especificados definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software.
- Existe un conjunto de requisitos implícitos que a menudo no se mencionan (no funcionales) que influyen en la calidad del software.

La obtención de un producto final con una calidad elevada está estrechamente vinculada a la gestión de la misma durante todo el proceso de desarrollo del software.

#### **1.1.1.2.- Gestión de calidad**

La Gestión de la Calidad del Software (SQA) consiste en la auditoría y las funciones de información de la gestión. Tiene como objetivo proporcionar la gestión para informar de los datos necesarios sobre la calidad del producto, lo que permite que se adquiera una visión más profunda y segura de que ésta está cumpliendo sus objetivos. [PRESSMAN 2005]

Los ingenieros de software afrontan la calidad aplicando métodos técnicos sólidos y medidas, realizando las revisiones técnicas formales y llevando a cabo las pruebas de software bien planificadas. [PRESSMAN 2005]

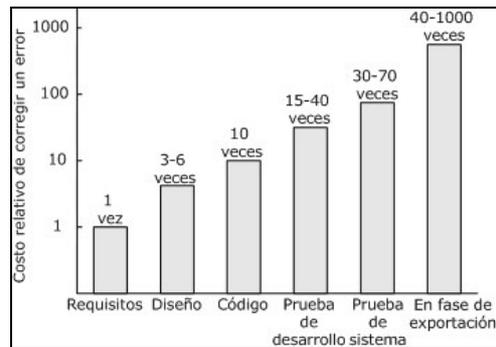
Para lograr que algo tenga la calidad requerida es necesario que sea controlado. El control de calidad es una serie de inspecciones, revisiones y pruebas utilizadas a lo largo del proceso del software para asegurar que cada producto cumpla con los requisitos que le han sido asignados; incluye un bucle de realimentación del proceso que lo creó. La combinación de medición y realimentación permite afinar el proceso cuando los productos de trabajo creados fallan al cumplir sus especificaciones. [PRESSMAN 2005] Es, según este enfoque, una parte importante del proceso de fabricación.

Las revisiones del software juegan un papel fundamental en el control de la calidad, son un filtro en el proceso de ingeniería de software aplicadas en varios momentos de su desarrollo para detectar errores y defectos que puedan ser eliminados y purificar las actividades de ingeniería del software que suceden como resultado del análisis, el diseño y la implementación. Es una forma de aprovechar la diversidad de un grupo de personas para: [PRESSMAN 2005]

- Señalar la necesidad de mejoras en el producto de una persona o equipo.
- Confirmar las partes de un producto en las que no es necesaria o deseable una mejora.
- Conseguir un trabajo técnico de una calidad uniforme, o al menos predecible, con el fin de hacer más manejable el trabajo técnico.

#### **1.1.1.3.- Costo de calidad**

El costo de la calidad incluye todos los costos acarreados en la búsqueda de la calidad o en las actividades relacionadas con su obtención. La figura 1 muestra la proporción con la que crece el costo de corregir un error a medida que avanza el proceso de desarrollo del software. [PRESSMAN 2005]



**Figura 1.-** Costo relativo de corregir un error. [PRESSMAN 2005]

Según la figura anterior, el costo aumenta drásticamente una vez entregado el producto a sus usuarios, lo que ocurre debido a que éste no tiene la calidad requerida. Por tal razón es tan importante controlar la calidad en el proceso de desarrollo para obtener finalmente un software con un alto grado de ella.

Para controlar la calidad durante el proceso de desarrollo software se necesita conocer a fondo las características específicas del producto a desarrollar, en este caso, del SE.

Sánchez J. en su libro Construyendo y Aprendiendo con el Computador, define el SE como cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar. [ÁLVAREZ and RODRÍGUEZ]

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

Otra definición dice que es un programa para ordenadores creado con la finalidad de facilitar los procesos de enseñanza y de aprendizaje. [MARQUÉS 2005]

Entonces para el caso del SE, un producto tiene calidad si logra "...satisfacer las expectativas del estudiante, del profesor, de la familia e incluso de la sociedad..." [MOMPIÉ 2006]; a través de la aptitud para su uso. [ÁLVAREZ and RODRÍGUEZ]

Hasta ahora se ha visto que una de las necesidades inmediatas para la Industria Cubana del Software es la obtención de SE con calidad, siendo su gestión y control a lo largo de todo el proceso de desarrollo del software la única manera de lograrlo.

El principal obstáculo para lograr la calidad deseada es la necesidad de dedicar una cantidad de recursos humanos y un tiempo de los que normalmente no se dispone; por ello se plantea la necesidad de desarrollar herramientas automáticas que ayuden a garantizar una calidad aceptable a un costo razonable. [MARTÍN and PÉREZ]

Una de las posibles herramientas pudiera ser un sistema gestor de información y control para la calidad del SE en unas predefinidas etapas de control. Existen varias clasificaciones que agrupan a los productos software, entre las que, por asociarse a la propuesta, destaca el software de gestión.

## **1.2.- Software gestor de calidad**

Antes de entrar en detalles sobre lo que es un software de gestión, sería bueno determinar qué es la gestión.

### **1.2.1.- Definición de gestión**

Según un vocabulario de negocios es la actividad de apoyo a la dirección que consiste en recopilar, procesar y presentar información relevante para la toma de decisiones. El control de gestión está basado tradicionalmente en instrumentos de tipo contable-financiero, pero contempla, cada vez más, información de índole no financiera. [MÉXICO 2004]

Es el proceso mediante el cual una empresa asegura que las realizaciones, como parte del proceso administrativo, concuerdan con la planificación y, a través del que se han de identificar las desviaciones para tomar las oportunas medidas correctivas, asegurando el empleo eficaz de los recursos obtenidos en el cumplimiento de los objetivos. [TONGA 2000]

Enfocando ambas definiciones al caso de una empresa o entidad desarrolladora de software con el propósito de obtener productos con buena calidad, la gestión sería la recopilación, el procesamiento y la presentación de información relevante que permitiría la:

- Verificación de la planificación hecha para el proceso de desarrollo software.
- Toma de decisiones para emplear de manera eficiente los recursos.

Todo con el fin de lograr software de alta calidad en el tiempo estipulado.

### **1.2.2.- Sistema de gestión de la calidad**

Un sistema de gestión de la calidad implementado y documentado permite controlar los diferentes procesos internos definidos en el campo de actividad de una empresa: [ANÓNIMO 2006c]

- Incrementa la eficiencia y reduce al mínimo el riesgo de no cumplir con las obligaciones para con el cliente.
- Puede medir la eficacia del sistema de gestión de acuerdo con varias normas industriales internacionales a través de la certificación de terceros.
- Establece los cimientos a partir de los que mejorar, de forma gradual, los procesos internos y reforzar la habilidad de la organización para lograr los objetivos estratégicos.

### **1.2.3.- Sistemas de información de gestión**

El proceso de la información comercial constituye la mayor de las tareas de aplicación del software. Los sistemas discretos han evolucionado hacia el software de Sistemas de Información de Gestión (SIG) que acceden a una o más BDs que contienen información comercial. Las aplicaciones en esta área: [PRESSMAN 2005]

- Reestructuran los datos existentes para facilitar las operaciones comerciales o gestionar la toma de decisiones.
- Realizan las tareas convencionales de procesamientos de datos.
- Realizan cálculo interactivo (por ejemplo: el procesamiento de transacciones en puntos de ventas).

Por tanto, conociendo que una de las necesidades inmediatas para la Industria Cubana del Software es la obtención de SE con calidad elevada, se determina que la gestión y control a lo largo de todo el proceso de desarrollo del software es la única manera de lograrlo.

En conclusión, un sistema de información y control para la gestión de la calidad del SE, será el que permitirá controlar el proceso de desarrollo software, enfocado en la mejora continua hasta alcanzar un software con un alto grado de calidad.

### 1.3.- Algunos gestores de calidad de software existentes en el mundo

Producto de lo difícil que resulta guiar y determinar la calidad del proceso de desarrollo del software mundialmente, existen muy pocas aplicaciones que se dediquen a esto. A continuación se listan dos de ellas.

#### 1.3.1.- ANACONDA

Sistema **AN**alizador **A**utomático de **CÓ**digo **N**atural para el **D**esarrollo de **A**plicaciones, que permite la evaluación de la calidad del software de aplicaciones informáticas desarrolladas en el entorno *Natural/Adabas*; analiza la aplicación partiendo del código fuente, verificando que cumple los requisitos de calidad establecidos en la instalación para poder ser implantada y explotada; y proporciona un diagnóstico que refleja el nivel de calidad del software realizado basándose en unas métricas y pesos definidos a partir del conocimiento de los expertos, generando un informe detallado con los errores y recomendaciones. [MARTÍN and PÉREZ]

Se centra en la medición de la calidad del software exclusivamente en la fase de codificación (específicamente en el análisis sintáctico del producto en lugar del semántico); basándose en indicadores de calidad como la autodocumentación de programas, el mantenimiento del código, los errores, la trazabilidad o el cumplimiento de la normativa. [MARTÍN and PÉREZ]

#### 1.3.2.- QUALYSOFT

Sistema elaborado en Cuba y publicado en 1995 por la Empresa de Producción y Desarrollo de Software de Calidad (SOFTCAL) y utilizado por varias empresas productoras de software del país. Brinda los lineamientos generales para el establecimiento y control de la actividad de Aseguramiento de la Calidad del Software (ACS). Constituye un material preliminar de apoyo tanto a los desarrolladores del software y de su documentación como a los evaluadores de su calidad. En su elaboración se han tenido en cuenta las normas ISO 9000-3, ISO 9126, ANSI, IEC e IEEE. [ANÓNIMO 2006]

Es aplicable a todo tipo de software tanto de aplicación como de sistema, independientemente de sus características propias. Válido para cualquier división en etapas que se haga al proceso de desarrollo del software. Es independiente de los métodos, técnicas y herramientas manuales y/o automatizadas que se empleen para la producción del software y de su documentación; de la metodología de desarrollo que se

utilice (orientada a los datos, a las estructuras o a objetos); de los medios y métodos utilizados para la interconexión, comunicación y transferencia de información. En el caso de ambiente de comunicación del procesamiento distribuido es independiente también del ambiente de operación del producto de software, en cuanto al software del sistema y al hardware (macro, mini, microcomputadora). Aunque sí requiere y presupone la utilización de una metodología de desarrollo para la ingeniería del software. [ANÓNIMO 2006i]

Su versión 1.0 constituye una guía que permite al usuario: [ANÓNIMO 2006i]

- Establecer los requisitos del ACS.
- Obtener el consenso necesario de las partes que intervienen en el ACS.
- Ayudar a la obtención del producto de software y su documentación con la calidad requerida.
- Ofrece un modelo cualimétrico para la evaluación de la calidad.

Según lo visto hasta ahora, se puede concluir que:

- La industria actual del software exige que los productos sean entregados con un alto grado de calidad.
- El costo de la calidad crece exponencialmente a medida que avanza el ciclo de desarrollo del software.
- La obtención de un producto con calidad está estrechamente vinculado al control de la misma durante todo el ciclo de desarrollo del software.
- La principal muestra de que el software tiene calidad, es que los usuarios queden satisfechos con el producto.
- El SE ha tomado un auge sin precedentes en la Industria Cubana, producto de su aceptación en las escuelas del país, resultando imperiosa la necesidad de que estos productos tengan un nivel de calidad muy alto; aunque actualmente este objetivo no se halla alcanzado.
- ANACONDA no se adapta a las condiciones de la Universidad porque está hecha para *Natural* y basa su análisis de la calidad en el código solamente.

- QUALYSOFT no se adapta a las condiciones de la Universidad porque es un sistema que se enfoca en dar los lineamientos básicos para el aseguramiento de la calidad, que es un paso importante en la gestión pero no el único.

Todas estas razones justifican la necesidad de proponer el desarrollo de un sistema que:

- Minimice el esfuerzo de la gestión de la calidad al automatizar los procesos para llevarla a cabo.
- Centralice la información necesaria para gestionar la calidad disponible para todos los interesados.

Para lo que se necesita escoger las tecnologías con las que se desarrollará el sistema.

#### **1.4.- Tendencias y tecnologías actuales utilizadas**

##### **1.4.1.- Plataforma de desarrollo**

Hoy en día existen fundamentalmente dos plataformas de desarrollo para las aplicaciones actuales, las llamadas aplicaciones de escritorio (*desktop*) y las Web.

###### **1.4.1.1.- Aplicaciones de escritorio**

Las aplicaciones de escritorio son las tradicionales, para las que se necesita un tipo de instalación en el cliente, consumen enormes cantidades de espacio en disco (en dependencia de la aplicación, claro está), incluyen lógica de procesamiento que necesitan grandes cantidades de memoria y de CPU, que además implican determinados requerimientos en las máquinas clientes (como la cantidad de memoria necesaria para su ejecución) y si se realiza una nueva versión, el cliente debe adquirirla nuevamente.

###### **1.4.1.2.- Aplicaciones Web**

Las aplicaciones Web son aquellas que los usuarios usan para acceder a un servidor Web a través de Internet o de una Intranet; son populares debido a la: [ANÓNIMO 2006a]

- Practicidad del navegador Web como cliente ligero.
- Habilidad para actualizarse y mantenerse sin distribuir e instalar software en miles de potenciales clientes.

Una aplicación Web es un sistema donde la entrada o acción del usuario afecta el estado del negocio. Su arquitectura general es la de un sistema cliente/servidor, donde tanto el cliente (el navegador) como el servidor (el servidor Web), y el protocolo mediante el que

se comunican (HTTP) son estándares. Una de sus ventajas más significativas es su forma de instalación y distribución; normalmente instalar una aplicación Web consiste en configurar los componentes del lado del servidor en la red y no es necesaria una instalación o configuración en el lado del cliente. [BARROSO 2006]

#### **1.4.1.3.- Fundamentación de la selección de la plataforma a utilizar**

Las aplicaciones Web han tenido un enorme auge en los últimos años debido a su relativa facilidad de implementación y de utilización sobre las aplicaciones de escritorio. A continuación se listan sus principales ventajas, que la señalan como la opción correcta para la propuesta de la investigación.

##### Ventajas de las aplicaciones Web [ANÓNIMO 2006a]

- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad *firewall* sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software prestados por lugares geográficos distantes puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándares.

Las aplicaciones Web tienen su base en la arquitectura cliente/servidor, modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios y en el cliente permanece sólo lo particular de cada uno. Sus características fundamentales son: [ANÓNIMO 2006b]

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Sus ventajas principales son que: [ANÓNIMO 2006b]

- El servidor no necesita potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.

#### **1.4.2.- Sistema de gestión de contenido**

Una de las tecnologías que surge a raíz del auge de las aplicaciones Web es el Sistema de Gestión de Contenidos (*Content Management System*, CMS). Éstos permiten la creación y administración de contenidos principalmente en páginas Web. Un CMS es una interfaz que controla una o varias BDs donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. [ANÓNIMO 2007h]

Provee las herramientas necesarias para gestionar el ciclo de vida de los contenidos: creación, gestión, presentación, mantenimiento y actualización. [GUERVOS 2005]

Las ventajas fundamentales de utilizar un CMS son: [PAZ 2006]

- Flexibilidad y escalabilidad en el diseño para su actualización, mediante la inclusión de nuevos módulos que permiten adaptarse a las necesidades del proyecto.
- Seguridad, ya que establece diferentes niveles de acceso a los contenidos, configurables fácilmente.
- Manejo con facilidad de importantes cantidades de información.
- Incorpora un manejo adecuado de la administración, el mantenimiento y la actualización de todo el sitio a través de la distribución del trabajo en varios usuarios, con privilegios explícitos para diferentes áreas.
- Incorpora un manejo sencillo de información dinámica.

Independientemente del tipo de CMS que se utilice, y de las ventajas que ofrecen, el sistema que se pretende construir no puede desarrollarse utilizando la tecnología ofrecida por ninguno debido a la gran desventaja que presentan: no permiten hacer búsquedas en los contenidos que suben los usuarios, siendo esto uno de los principales objetivos del sistema, ya que se pretende que éste permita generar algunos reportes a partir de la información que introduzcan éstos.

En vistas de que no se utilizará un CMS se debe determinar las herramientas para desarrollar la propuesta.

#### 1.4.3.- Lenguaje de desarrollo Web

Las aplicaciones Web producto del desarrollo vertiginoso que han sufrido poseen innumerables herramientas para su desarrollo, entre los principales lenguajes o tecnologías se cuentan: [VIRTUALPYME 2006]

- **Practical Extraction Report Language (Perl):** lenguaje derivado de C que permite manipular fácilmente texto, ficheros y procesos. Se entrega con la mayoría de las plataformas *Unix* y está disponible gratuitamente en todos los sistemas de explotación. La manipulación del lenguaje *Perl* permite programación gráfica, Web, acceso a BDs y redes; lo que hace que sea muy popular entre los administradores de sistemas y programadores de *scripts* CGI. La última versión ha sido modularizada, orientada a objetos, optimizada y con una alta compatibilidad con versiones anteriores.
- **Active Server Pages (ASP):** permite a los programadores de sitios Web y de aplicaciones construir y mantener aplicaciones Web dinámicas conectables a una BD. Incluye *ActiveX Data Objects* (ADO) para el acceso estándar de datos a una amplia variedad de sistemas de gestión de BD. Cuenta con herramientas para enlazar BDs a través de máquinas heterogéneas ODBC, permitiendo una extrema flexibilidad en la configuración del servidor Web/Servidor de Aplicaciones/Sistema de Gestión de Base de Datos (SGBD). Para aplicaciones Web simples, ASP es más rápido y eficaz que *Java* o CGI, y es más potente para aplicaciones más complejas. Su arquitectura modular aporta todos los recursos de desarrollo productivo. Permite estandarizar una solución para Webs no complejas.

- *Hypertext Pre-processor* (inicialmente *Personal Home Page Tools*, PHP): lenguaje de programación que se centra en aplicaciones Web dinámicas. Más de tres millones de sitios consideran que vale la pena utilizarlo y lo han convertido en una plataforma de desarrollo de aplicación rápida (*Rapid Application-Development*, RAD).

#### 1.4.3.1.- Fundamentación de la selección del lenguaje de desarrollo a utilizar

**Tabla 1.-** Comparación entre PHP, ASP y Perl. [ANÓNIMO 2007a; 2007g; BARROSO 2006; LÓPEZ and RAMÍREZ 2004]

Características	PHP	ASP	Perl
<b>Licencia</b>	Distribución libre.	Microsoft.	Artística <sup>1</sup> .
<b>Plataformas en las que funciona</b>	<ul style="list-style-type: none"> <li>- Windows.</li> <li>- Linux/Unix.</li> <li>- Solaris, con el software servidor apropiado.</li> <li>- Mac OS.</li> <li>- OS/2.</li> </ul>	<ul style="list-style-type: none"> <li>- Windows (9x, NT).</li> <li>- Linux/Unix (previo pago y con el software adecuado).</li> </ul>	<ul style="list-style-type: none"> <li>- Windows.</li> <li>- Linux/Unix.</li> <li>- Mac (OS 7-9 y X).</li> <li>- Solaris.</li> </ul>
<b>Gestión de memoria</b>	Automática.	Algunos defectos por solucionar.	Automática.
<b>Servicios que soporta</b>	<ul style="list-style-type: none"> <li>- Encriptado.</li> <li>- FTP.</li> <li>- E-mail.</li> <li>- Compresión de datos.</li> <li>- XML.</li> </ul>	<ul style="list-style-type: none"> <li>- Encriptado.</li> <li>- FTP.</li> <li>- E-mail.</li> </ul> <p>Cada uno es un módulo separado con sus respectivos precios.</p>	<ul style="list-style-type: none"> <li>- Encriptado.</li> <li>- FTP.</li> <li>- E-mail.</li> </ul>
<b>Bases de datos (BDs) que soporta</b>	<ul style="list-style-type: none"> <li>- MySQL.</li> <li>- MS SQL Server.</li> <li>- PostgreSQL.</li> <li>- Oracle.</li> <li>- DB2D.</li> <li>- Ovrimos.</li> <li>- Sybase.</li> <li>- Ingres.</li> <li>- Interbase.</li> <li>- Informix.</li> </ul> <p>El controlador para conexión con las BD va incluido en el intérprete.</p>	<ul style="list-style-type: none"> <li>- Access.</li> <li>- Y otros SGBD con el estándar SQL.</li> </ul> <p>Requiere un controlador para la conexión con las BD.</p>	<ul style="list-style-type: none"> <li>- Oracle.</li> <li>- Sybase.</li> <li>- PostgreSQL.</li> <li>- MySQL.</li> </ul>
<b>Sintaxis</b>	Sigue el modelo	Sigue el modelo	Deriva de C/Shell/

	<i>Java/C++.</i>	<i>Visual Basic.</i>	<i>Lisp/AWK<sup>2</sup>/Sed<sup>3</sup>.</i>
<b>Velocidad de ejecución</b>	De los tres el más rápido.	El tercero más rápido.	El segundo más rápido.
<b>Código</b>	Libre.	Propietario.	Libre.

<sup>1</sup> Licencia del software usada para ciertos paquetes de software libre.

<sup>2</sup> Lenguaje de programación diseñado para procesar datos basados en texto, ya sean ficheros o flujos de datos.

<sup>3</sup> Editor de flujo, potente herramienta de tratamiento de texto para el sistema operativo *Unix* que acepta como entrada un archivo, lo lee y modifica línea a línea mostrando el resultado en pantalla.

Según la tabla 1 PHP y *Perl* son más favorecidos que ASP, no obstante PHP es escogido porque a pesar de su notable facilidad de aprendizaje sobre *Perl*, los principales motivos que lo señalan como la opción correcta son: [WELLING and THOMPSON 2003]

- Desarrollo Web rápido: organizar y ejecutar *scripts* PHP es muy rápido en comparación con lenguajes de programación como C/C++ y *Java*.
- Transportable a la mayoría de las plataformas de servidores Web: desde PHP 4.0 la interfaz del servidor Web (*Server API* o *SAPI*) es abstracta, lo que le permite integrar perfectamente con servidores Web distintos, incluidos *Apache*, *Internet Information Server* (IIS), *iPlanet/Netscape Enterprise Server* y *Zens*.
- Alto rendimiento: desde PHP 4.0 fue renovado para alto rendimiento; un módulo gratuito llamado *Zend Optimizer* de *Zend Technologies* puede optimizar una aplicación y que esta pase a ser entre 40 a 100 veces más rápida que cuando no estaba optimizada.
- Fácil de aprender para desarrolladores nuevos: es como el lenguaje C, con gestión automática de memoria (no tiene punteros). Tiene elementos de *Perl*, *Java* y C++.
- Soporta estándares de Internet: como IMAP, FTP, POP, XML, WDDX, LDAP, NIS y SNMP, lo que significa que puede funcionar de interfaz con distintos estándares y tecnologías con facilidad, todos ellos desde un conjunto de herramientas comunes, sin necesidad de módulos caros de terceras partes.
- Comunidad de código fuente abierto: el código fuente abierto ha resultado ser una aproximación de desarrollo de software muy eficaz, se dispone de acceso al

código fuente, si se desea modificar algo o agregar un elemento al programa, se puede hacer con total libertad sin necesidad de esperar a que el fabricante publique parches.

- Bibliotecas incorporadas: incorpora una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes *GIF* al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con *cookies* y generar documentos PDF, todo con unas pocas líneas de código.
- Su librería estándar es realmente amplia, lo que permite reducir los llamados "costos ocultos", uno de los principales defectos de ASP. [LÓPEZ and RAMÍREZ 2004]
- Tiene una de las comunidades más grandes en Internet, por lo que no es complicado encontrar ayuda, documentación, artículos, noticias y otros recursos. [LÓPEZ and RAMÍREZ 2004]

#### 1.4.4.- Servidor Web

Un servidor Web es un software que suministra páginas Web en respuesta a las peticiones de los navegadores Web. Entre los más utilizados a nivel mundial se encuentran:

- *Internet Information Services/Server (IIS)*: serie de servicios para los ordenadores que funcionan con *Windows*. Originalmente era parte del *Option Pack* para *Windows NT*. Luego fue integrado en otros sistemas operativos de *Microsoft* destinados a ofrecer servicios, como *Windows 2000* o *Windows Server 2003*. *Windows XP Profesional* incluye una versión limitada de IIS. Ofrece servicios como FTP, SMTP, NNTP y HTTP/HTTPS. Procesa páginas de ASP y ASP.NET; aunque también puede incluir PHP o *Perl*. Una de las desventajas que tiene es que solamente se puede utilizar en sistemas de *Windows*. [ANÓNIMO 2006d]
- *Apache*: servidor de código abierto para plataformas *Unix* (BSD, GNU/Linux), *Windows* y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Considerado el servidor número uno del mundo. [KABIR 2003]

#### 1.4.4.1.- Fundamentación de la selección del servidor Web a utilizar

La siguiente tabla muestra la estadística de los sitios activos por servidores Web.

**Tabla 2.-** Servidores Web más utilizados en octubre-noviembre de 2006. [ANÓNIMO 2006f]

Desarrollador	Octubre	Por ciento	Noviembre	Por ciento	Cambio
Apache	28 820 676	61.04	29 149 301	60.45	-0.59
IIS	15 262 010	32.32	15 806 612	32.78	0.46
Sun	249 824	0.53	238 245	0.49	-0.04
Zeus	146 120	0.31	210 544	0.44	0.13

Como se ve en la tabla 2 *Apache* fue el servidor más utilizado en el mundo a finales del 2006, entre las características que lo señalan como la opción correcta están: [KABIR 2003]

- Servidor altamente configurable de diseño modular: se pueden escribir módulos para realizar determinadas funciones lo que implica que haya gran cantidad de ellos disponibles para su utilización.
- Tecnología gratuita de código fuente abierto.
- Trabaja con gran cantidad de código *script* como *Perl* y *PHP*.
- Multiplataformas: funciona en *Linux*, *Unix* y *Windows*.

#### 1.4.5.- Gestor de base de datos

Los SGBD son un tipo de software muy específico dedicado a servir de interfaz entre las BDs, el usuario y las aplicaciones que las utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. [ANÓNIMO 2006h]

En la actualidad, las compañías demandan una clase diferente de solución de BDs. El rendimiento, la escalabilidad y la confiabilidad son esenciales y la anticipación al mercado es crítica. [ANÓNIMO 2006b]

Dentro de los gestores de BDs más utilizados se encuentran:

- *Microsoft SQL Server*: proporciona agilidad a sus operaciones de análisis y administración de datos al permitir a su organización adaptarse rápida y fácilmente para obtener ventaja competitiva en un entorno de cambios constantes. *SQL Server 2000* es un paquete completo de BD y análisis de datos que abre las

puertas al rápido desarrollo de una nueva generación de aplicaciones comerciales de nivel empresarial, que pueden proporcionar a su compañía una ventaja competitiva crítica. Ha obtenido importantes galardones en pruebas de referencia por su escalabilidad y velocidad. Es un producto de BD totalmente habilitado para Web que proporciona una compatibilidad fundamental con el Lenguaje de Marcado Extensible (XML, *Extensible Markup Language*) y la capacidad para realizar consultas en Internet y por encima del servidor de seguridad. [ANÓNIMO 2006b]

- *Oracle*: sistema de administración de BD relacional (RDBMS *Relational Data Base Management System*) fabricado por *Oracle Corporation*. Básicamente es una herramienta cliente/servidor para la gestión de BDs. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que sólo se vea en empresas muy grandes y multinacionales, por lo que no está tan extendido como otras BDs. Está desarrollado en PL/SQL un lenguaje de quinta generación, bastante potente para tratar y gestionar la BD, también por norma general se suele utilizar SQL. Es sin duda una de las mejores BDs existentes en el mercado, es un SGDB robusto, tiene muchas características que garantizan la seguridad e integridad de los datos; que las transacciones se ejecuten de forma correcta, sin causar inconsistencias; ayuda a administrar y almacenar grandes volúmenes de datos; posee gran estabilidad y escalabilidad y es multiplataforma. Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia de gestores de BDs comerciales y de la oferta de otros con licencia Software Libre como *PostgreSQL*, *MySQL* o *FireBird*. Las últimas versiones de *Oracle* han sido certificadas para poder trabajar bajo *Linux*. [PROAÑO 2006]
- *MySQL*: sistema RDBMS, rápido, sólido y muy fácil de usar. Controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios, para proporcionar acceso a dichos datos y para asegurarse de que sólo obtienen acceso a ellos los usuarios con autorización. Utiliza SQL (*Structured Query Language*, Lenguaje de Consulta Estructurado), el lenguaje estándar para la consulta de BDs utilizado en todo el mundo. Existen distribuciones bajo una licencia de código abierto. [WELLING and THOMPSON 2003]

- *PostgreSQL*: motor de BD objeto-relacional que utiliza el lenguaje SQL92/SQL99. Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual. Incluye características de la orientación a objetos como herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, aunque no es un SGBD puramente orientado a objetos. [QUEZADA 2006]

#### 1.4.5.1.- Fundamentación de la selección del gestor de base de datos a utilizar

**Tabla 3.-** Comparación entre *MS SQL Server*, *Oracle*, *MySQL* y *PostgreSQL*. [ANÓNIMO 2007b; 2007c; 2007d; 2007e; 2007i; 2007j]

Características	<i>MS SQL</i>	<i>Oracle</i>	<i>MySQL</i>	<i>PostgreSQL</i>
<b>Licencia</b>	<i>Microsoft Corporation.</i>	<i>Oracle Corporation.</i>	Distribución libre.	Licencia BSD.
<b>Soporte de sistemas operativos</b>	- <i>Windows.</i>	- <i>Windows.</i> - <i>Mac OS X.</i> - <i>Linux/Unix.</i>	- <i>Windows.</i> - <i>Mac OS X.</i> - <i>Linux/Unix.</i> - BSD.	- <i>Windows.</i> - <i>Mac OS X.</i> - <i>Linux/Unix.</i> - BSD. - <i>z/OS.</i>
<b>ACID</b>	Sí.	Sí.	Depende <sup>1</sup> .	Sí.
<b>Integridad referencial</b>	Sí.	Sí.	Depende <sup>1</sup> .	Sí.
<b>Transacciones</b>	Sí.	Sí.	Depende <sup>1</sup> .	Sí.
<b>Unicode</b>	Sí.	Sí.	Sí.	Sí.

<sup>1</sup> Para las transacciones y la integridad referencial debe ser usado el tipo de tabla *InnoDB*; el tipo de tabla por defecto, *MyISAM*, no soporta estas características. Sin embargo, inclusive el tipo de tabla *InnoDB* permite el almacenamiento de valores que excedan el rango de datos; algunas vistas violan la limitación de ACID.

Según la tabla 3 *Oracle* es el SGDB más favorecido desde el punto de vista funcional. Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias). Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas veintidós vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de dos años. [ANÓNIMO 2007f]

Sin tener en cuenta los sistemas propietarios *PostgreSQL* es más potente que *MySQL*. No obstante, según Emilio Rodríguez en su artículo *MySQL vs PostgreSQL*, para BDs

pequeñas (menor de veinte mil registros) *MySQL* ofrece mayor velocidad. [GARCÍA 2007] Además de que la sintaxis de algunos de los comandos de *Postgre* no es nada intuitiva, resultando engorrosas las pequeñas variaciones que presenta en algunos tipos de datos que maneja. [QUEZADA 2006]

Teniendo en cuenta todo esto, y como factor de mayor peso el hecho de la velocidad, se decide escoger *MySQL* que tiene las siguientes características que lo señalan como la opción correcta: [WELLING and THOMPSON 2003]

- Rendimiento: es muy rápido.
- Bajo costo: está disponible de manera gratuita, bajo una licencia de código abierto o por un precio reducido en forma de licencia comercial si resultara necesario para la aplicación.
- Facilidad de uso: además de utilizar SQL, resulta más sencillo de configurar que otros productos similares.
- Portabilidad: se puede utilizar en una gran cantidad de sistemas *Unix* diferentes así como bajo *Windows*.
- Código fuente: se puede obtener y modificar el código fuente según las necesidades.

#### **1.4.6.- Otras herramientas necesarias**

Para confeccionar la propuesta de este trabajo hay que tener en cuenta la utilización de: [BARROSO 2006]

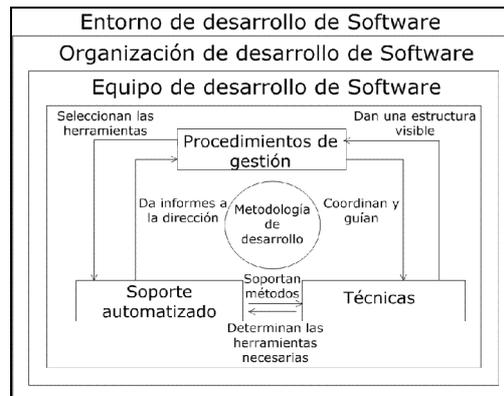
- Un editor de páginas Web: la herramienta de creación de sitios Web más utilizada en la actualidad es el *Macromedia Dreamweaver* (versión 8.0); con la que se puede desarrollar cualquier sitio Web personal con características de sitio profesional y utilizar casi todos los recursos de la Web, así como realizar aplicaciones que se ejecuten en servidor y vinculaciones dinámicas de datos; además de contar con un soporte para aplicaciones PHP5 y utilización de BDs *MySQL*. También cuenta con un amplio soporte para la creación y utilización de CSS (*Cascading Style Sheets*) para lograr un diseño fácil y óptimo. [BARROSO 2006]

- Una herramienta para el trabajo con las imágenes: se escoge *Adobe Photoshop 8*, ya que se considera la aplicación estándar para el tratamiento digital de imágenes. Las continuas mejoras han hecho de este programa uno de los más profesionales para la edición y retoque fotográfico. Tiene un enfoque dirigido hacia los gráficos para la Web. [BARROSO 2006]
- Editor de código PHP: se escoge a *NuSphere* ya que facilita el trabajo de programación con numerosas características de gran utilidad entre las que destacan: completo sistema de ayuda; plantillas de documento y de fragmentos de código frecuentes; código de colores para comandos en PHP, *Perl*, *JavaScript*, SQL y HTML. Además esta herramienta incluye un cliente de FTP y un servidor Web integrados, totalmente configurables según las necesidades de trabajo. [PHP 2007]
- Administración de BD: se escoge a *EMS Manager SQL 2005* ya que es una aplicación de alto desempeño para la administración y desarrollo de servidores de BDs *MySQL*. Entre sus características principales se listan: soporte completo para *MySQL* desde la versión 3.23 hasta la 5.06; administración y navegación rápida de BD; administración fácil de todos los objetos *MySQL*; posee herramientas de manipulación avanzada de datos; administración efectiva de seguridad; excelentes herramientas visuales y de texto para la construcción de consultas; capacidades de exportación e importación de datos; poderoso diseñador visual de BDs; modo guiado para labores de mantenimiento e interfaz de fácil uso. [ARCHIVOSPC 2006]

#### **1.4.7.- Metodología a utilizar**

##### **1.4.7.1.- Definición de metodología**

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software. [ANÓNIMO 2006e]



**Figura 2.-** Impacto de la metodología en el desarrollo software. [ANÓNIMO 2006e]

La figura 2 muestra el impacto de las metodologías en el desarrollo del software, producto de ello es que es tan importante escoger una buena metodología para guiar dicho proceso.

Por sus características y las facilidades que aporta a todo el proceso y teniendo en cuenta que viene acompañado del *Rational Rose Enterprise Edition 2000*, herramienta que soporta cada uno de los procesos, es que se decide escoger a RUP (*Rational Unified Process*).

#### 1.4.7.2.- Proceso Unificado del *Rational*

Existen varias metodologías de desarrollo, pero la más conocida y extendida es la metodología RUP, que como su nombre lo dice, se basa en unificar el proceso de desarrollo software para lograrlo de manera exitosa.

Un proceso de desarrollo software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. [JACOBSON et al. 2004]

#### Características principales de RUP [LÓPEZ and RAMÍREZ 2004]

- Guiado por CUs.
- Centrado en la arquitectura.
- Iterativo e incremental.

- Desarrollo basado en componentes.
- Utilización de un único lenguaje de modelación (UML).
- Proceso integrado.

Se divide en cuatro fases de desarrollo del software [SÁNCHEZ 2004]

- Inicio: con el objetivo de determinar la visión del proyecto.
- Elaboración: con el objetivo de determinar la arquitectura óptima.
- Construcción: con el objetivo de obtener la capacidad operacional inicial.
- Transición: con el objetivo de obtener el *release* del proyecto.

Cada una de estas etapas es desarrollada mediante un ciclo de iteraciones que 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.

El ciclo de vida en cada iteración, es llevada bajo dos disciplinas: [SÁNCHEZ 2004] Ver figura 3.

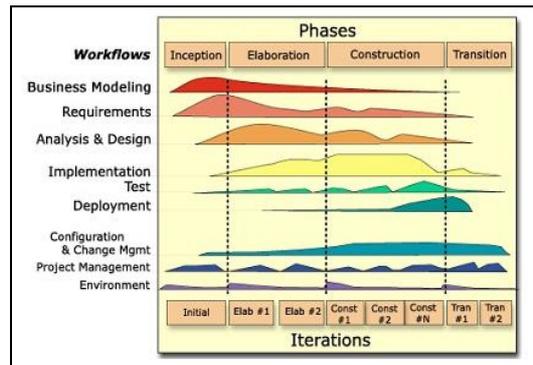
#### Disciplina de desarrollo

- Ingeniería de negocios: permite entender las necesidades del negocio.
- Requerimientos: permite trasladar las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: permite trasladar los requerimientos dentro de la arquitectura de software.
- Implementación: permite crear software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: permite asegurar que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

#### Disciplina de soporte

- Configuración y administración del cambio: permite guardar todas las versiones del proyecto.
- Administrando el proyecto: permite administrar horarios y recursos.

- Ambiente: permite administrar el ambiente de desarrollo.
- Distribución: permite hacer todo lo necesario para la salida del proyecto.



**Figura 3.-** Fases e iteraciones de la metodología RUP. [SÁNCHEZ 2004]

A cada una de estas iteraciones se les clasifica y ordena según su prioridad para convertirlas en entregables al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

#### Elementos de RUP [JACOBSON *et al.* 2004; SÁNCHEZ 2004]

- Actividades: los procesos que se llegan a determinar en cada iteración.
- Trabajadores: las personas o entes involucrados en cada proceso.
- Artefactos: documento, 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.

#### **Conclusiones**

Después de analizada la importancia que tiene elevar la calidad de los productos en el proceso de desarrollo del software, principalmente el educativo; y visto que en la Universidad no existe una herramienta que permita gestionarla y controlarla, se hace necesario un sistema, propuesta de la presente investigación. En este capítulo se profundizó en el conocimiento de algunos conceptos necesarios para la comprensión del trabajo. Además se realizó un análisis de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto y se fundamentaron las elecciones del lenguaje, del sistema gestor de bases de datos y de la metodología a utilizar.

Por los argumentos antes expuestos y dado que se hará uso de la tecnología cliente/servidor sobre plataforma Web para implementar este trabajo, se escogió PHP, MySQL y Apache, porque además de las características expuestas anteriormente, su combinación: [BARROSO 2006]

- Ofrece un alto nivel de prestaciones en la red hoy en día.
- Ofrece al programador una API (*Application Programming Interface*) muy potente que permite el desarrollo de cualquier tipo de aplicación de gestión de datos.
- Es relativamente fácil su uso.
- La seguridad que implementa el Apache más la implementada por MySQL, satisface los requerimientos de seguridad para el sistema propuesto.

# Capítulo 2

## Características del sistema

### Introducción

En el capítulo se hará la descripción de la propuesta del trabajo, para lo que se determinarán los procesos de negocio que tienen que ver con el campo de acción, en base a lo cual se conformará un modelo de dominio. Además se enumerarán los requisitos funcionales y no funcionales que debe tener el sistema en proposición, se identificarán los actores, los CUs y las relaciones existentes entre ellos a través del principal artefacto que se obtiene en el flujo de trabajo Requisitos: el modelo de CUs del sistema.

### 2.1.- Descripción de los procesos de negocio

Para describir los procesos de negocio que se relacionan con el campo de acción de este trabajo es necesario enfocarse en cómo se controla la calidad de los productos durante el proceso de desarrollo del software en la Universidad y del educativo específicamente, en la Facultad 8.

La UCI está formada por diez facultades, cada una con un perfil de producción específico acorde al tipo de software que producen mayormente. Para lograr un alto grado de calidad en los productos que se desarrollan en ellas, la Universidad cuenta con un departamento central de calidad que tiene estipuladas una serie de plantillas y reglas a cumplir, las que son enviadas por correo electrónico a asesores de calidad y vicedecanos de producción de las distintas facultades, los que a su vez las envían a los líderes y encargados de calidad en los proyectos. Todo esto independiente de un grupo dedicado a la calidad en cada facultad, que hace las pruebas al producto una vez concluido su ciclo de desarrollo.

La facultad 8, con su perfil productivo de SE; como parte de la Universidad se rige por las normas que el departamento de calidad tiene. No obstante, no lleva a cabo una buena gestión de la calidad durante el proceso de desarrollo, la que simplemente se reduce a verificar al final del ciclo, si el producto cumple con los requisitos por los que fue implementado.

Además no existe un control de los entregables que se desarrollan en el proceso de desarrollo del SE, lo que influye de manera negativa en el logro de productos con buena calidad.

El sistema propone facilitar la gestión de la calidad en la facultad durante todo el proceso de desarrollo, permitiendo a los encargados de calidad en cada proyecto gestionar los entregables mínimos con los que debe cumplir para culminar una determinada etapa del proceso de desarrollo con calidad aceptable; los que posteriormente serán evaluados por el asesor de calidad de la facultad, quién hará reportes de los mismos permitiendo este intercambio de información, la retroalimentación y el logro, desde el propio comienzo del ciclo de desarrollo del software de un proceso con calidad, lo que se traduce en un producto con calidad elevada finalmente.

## **2.2.- Descripción del sistema propuesto**

Para cumplir los objetivos previamente presentados, el sistema en proposición tendrá tres módulos (administración, gestión de proyectos individuales, gestión de proyectos e información a nivel de facultad) y seis roles (usuario anónimo, usuario registrado, vicedecano de producción, líder de proyecto, encargado de calidad a nivel de proyecto y asesor de calidad a nivel de facultad).

En la BD se controlarán los datos y el rol de cada usuario que exista. Un usuario anónimo puede ver y descargar archivos referentes a cómo controlar la calidad del SE durante su proceso de desarrollo a través de los entregables mínimos que se deben gestionar. Los usuarios registrados pueden ser el Vicedecano de producción, el Líder de proyecto y el Encargado de calidad a nivel de proyecto y el Asesor de calidad a nivel de facultad.

El módulo Administración puede ser accedido por el Vicedecano de producción. Con él se pretende: (1) gestionar los usuarios que van a tener acceso al resto de los módulos, lo que permitirá establecer niveles de privilegios favoreciendo la confiabilidad y la integridad de la información; y (2) la gestión de los proyectos, asignándoles su líder y encargado de calidad.

El módulo Gestión de proyectos individuales puede ser accedido por el Líder de proyecto y el Encargado de calidad. Con él se pretende gestionar la información referente a los proyectos de forma individual, la creación de su expediente y la gestión de los entregables que deben controlar.

El módulo Gestión de proyectos e información a nivel de facultad puede ser accedido por el Asesor de calidad. Con él se pretende gestionar la información de todos los proyectos de la facultad, la gestión de entregables que deben llenar los proyectos y documentos de interés y la gestión de los reportes que se le hacen a cada uno evaluando la calidad durante el proceso de desarrollo. Aunque tiene una parte que no requiere autenticación, que es la que permitirá visualizar la información de interés para la gestión de la calidad.

En resumen, con el sistema se permitirá gestionar los entregables mínimos que debe tener un proyecto de SE para obtener un producto con calidad, a través de su control durante todo el proceso de desarrollo, así como centralizar y tener actualizada la información que permita lograr dicho objetivo.

### **2.3.- Modelado de dominio**

El hecho de que los procesos del negocio tengan muy bajo nivel de estructuración; que sea difícil el establecimiento de reglas para la gestión como tal de los artefactos que se vayan definiendo en el proceso de desarrollo; que no existan flujos de información interconectados y bien definidos; inciden en que no se pueda llevar a cabo como tal un modelado de negocio, haciéndose necesario uno de dominio.

#### **2.3.1.- Glosario de términos**

Facultad: institución que representa un conjunto de estudiantes, profesores y trabajadores que enfocan sus actividades en la docencia y la producción.

Vicedecano de producción: rol encargado de toda la actividad productiva en la Facultad; asigna los proyectos productivos, determina sus líderes y controla su trayectoria durante el proceso de desarrollo.

Asesor de calidad: rol encargado de controlar la calidad de los productos liberados por los proyectos productivos de la Facultad.

Proyecto: elemento organizativo a través del cual se gestiona el desarrollo del software, tiene como resultado la versión de un producto.

Etapas: proceso definido dentro del desarrollo del software.

Entregable: documento que definirá factores y atributos que permitirá controlar la calidad del producto software.

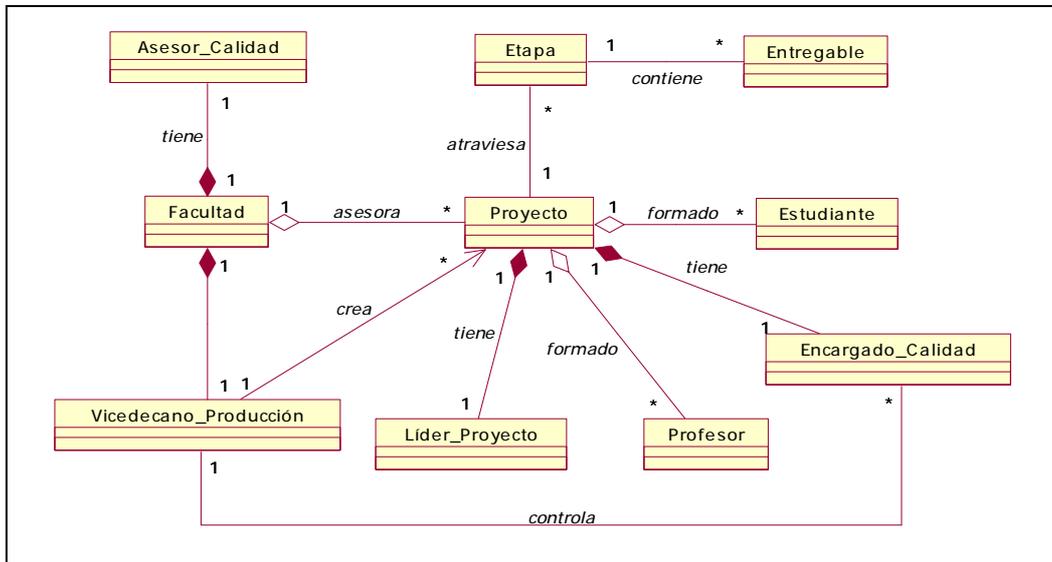
Líder de proyecto: rol encargado de dirigir un proyecto productivo en particular.

Encargado de calidad: rol encargado de controlar la calidad en un proyecto productivo en particular.

Estudiante: rol con responsabilidades en el proyecto que todavía tiene carga de estudio en la carrera.

Profesor: rol con responsabilidades en el proyecto que además da clases en la Facultad.

Documento: datos básicos que identifican el proyecto.



**Figura 4.-** Diagrama de dominio.

Las dos tablas que se muestran a continuación contienen las descripciones de los trabajadores del dominio, sus descripciones se encuentran en la sección 2.3.1.- Glosario de términos, del presente trabajo.

**Tabla 4.-** Trabajadores del dominio.

Líder_Proyecto
Encargado_Calidad
Asesor_Calidad
Vicedecano_Producción

**Tabla 5.-** Entidades del dominio.

Entregable
Proyecto
Documento

## 2.4.- Requerimientos funcionales

De acuerdo con los objetivos planteados, el sistema debe ser capaz de:

1. Mostrar documentación sobre los entregables mínimos que debe gestionar un proyecto multimedia en cada etapa de control, específica del SE, durante el proceso de desarrollo.

2. Descargar archivos.
3. Crear nuevo proyecto.
4. Pasar al historial los proyectos terminados.
5. Eliminar datos de proyectos terminados.
6. Crear nuevos usuarios.
7. Modificar perfil de usuario.
8. Eliminar usuarios.
9. Autenticar usuarios con diferentes niveles de acceso a la información.
10. Cambiar contraseña de usuarios.
11. Llenar expediente de proyecto.
12. Modificar expediente de proyecto.
13. Llenar los entregables en cada una de las etapas en el proyecto.
14. Modificar los entregables en cada una de las etapas en el proyecto.
15. Generar reportes de proyecto.
16. Imprimir reportes de proyecto.
17. Ver evolución del proyecto.
18. Subir nuevos documentos de interés para la gestión y control de la calidad en el proceso de desarrollo del SE.
19. Sobrescribir documentos de interés para la gestión de la calidad en el proceso de desarrollo del SE.

## **2.5.- Requerimientos no funcionales**

Los requisitos no funcionales son las propiedades o cualidades que el producto debe tener y que lo hacen atractivo, usable, rápido o confiable. A continuación se listan los que debe cumplir el sistema en proposición.

- Interfaz externa:
  - Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

- Diseño encuadrado para la resolución 800x600, y preparado para verse en otras resoluciones.
- Usabilidad:
  - El sistema podrá ser usado por cualquier persona interesada en conocer los entregables mínimos, por etapa de control.
  - El sistema podrá ser usado por los encargados de gestionar la calidad en los proyectos productivos relacionados con el SE, por el asesor de calidad de la facultad y el vicedecano de producción.
- Rendimiento:
  - Tiempo de respuesta rápido.
  - Velocidad de procesamiento de la información rápida.
- Soporte:
  - Servidor de BD que soporte volúmenes medios de datos y que tenga buena velocidad de procesamiento, con tiempo de respuesta rápido en accesos concurrentes.
  - Versión de PHP 5.2.0 o superior. Incluido un motor de plantillas *Smarty* v.2.6.9.
  - Por parte del cliente se requiere un navegador capaz de interpretar *JavaScript*.
- Seguridad:
  - Identificar al usuario antes de que pueda realizar cualquier acción excepto ver y descargar archivos.
  - Establecer niveles de permisos para el acceso de los usuarios a los módulos de administración, de gestión de proyectos individuales y de gestión de información y proyectos a nivel de facultad, garantizando que la información sea posible modificarla solamente por quien tiene derecho.
  - Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.

- Proteger contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificar sobre acciones irreversibles (eliminación, modificación).
- Legales:
  - La plataforma escogida para el desarrollo de la aplicación está basada en licencia de software libre.
- Confiabilidad:
  - La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.
- Software:
  - Navegador compatible o superior con *Internet Explorer* 6.0.
  - *Macromedia Dreamweaver* 8.0.
  - *Adobe Photoshop* 8.0.
  - *SQL Manager* 2005, edición 3.5.0.7.
  - *Apache* 2.2.4.
  - PHP 5.2.1.
  - NuSphere 4.6.2.
- Portabilidad:
  - Necesidad de que el sistema sea multiplataforma.

## 2.6.- Modelo de Casos de Usos

Para desarrollar el modelo de CU es necesario determinar los actores y los CU que se plasmarán en él.

**Tabla 6.-** Justificación de los actores del sistema.

Actores	Justificación
<b>Usuario_Anónimo</b>	Usuario que utiliza el sistema sin necesidad de autenticación para buscar información referente a los entregables mínimos que debe tener un proyecto de SE y en cuáles etapas de control; puede ser un estudiante, un profesor o un trabajador de la Universidad.
<b>Usuario_Privilegios</b>	Usuario autenticado previamente que utiliza el sistema para

	gestionar información, acorde a su rol, pudiendo modificar los recursos; puede ser un estudiante o un profesor.
<b>Vicedecano_Producción</b>	Usuario encargado de la gestión de proyectos y usuarios con el objetivo de que tengan sólo acceso a la información acorde a su rol.
<b>Asesor_Calidad</b>	Usuario encargado de gestionar los reportes que se le hacen a los proyectos al evaluar sus entregables y de definir los documentos de interés para la gestión y el control de la calidad en el proceso de desarrollo del SE.
<b>Líder_Proyecto</b>	Usuario encargado de gestionar los datos básicos del proyecto; puede ser un profesor o un trabajador.
<b>Encargado_Calidad</b>	Usuario encargado de gestionar los entregables del proyecto durante todo el ciclo de desarrollo; puede ser un estudiante o profesor encargado de la calidad en el proyecto.

La propuesta del presente trabajo tiene ocho CUs que engloban los diez y nueve requisitos funcionales. En el epígrafe siguiente se describen cada uno de ellos.

### 2.6.1.- Determinación de los Casos de Usos

**Tabla 7.-** CU Gestionar información.

<b>CU-1</b>	Gestionar_Información.
<b>Actor</b>	Usuario_Anónimo, Usuario_Privilegios, Asesor_Calidad, Líder_Proyecto, Encargado_Calidad, Vicedecano_Producción.
<b>Descripción</b>	Un usuario busca algún tipo de información referente a las etapas y entregables, por cada una de ellas, durante el proceso de desarrollo del SE; o de otro tipo de información referente al tema.
<b>Referencia</b>	R1, R2.

**Tabla 8.-** CU Gestionar proyecto.

<b>CU-2</b>	Gestionar_Proyecto.
<b>Actor</b>	Vicedecano_Producción.
<b>Descripción</b>	El Vicedecano_Producción puede crear un nuevo proyecto (asignar un nombre), cambiar su estado de activo a terminado o eliminar los proyectos del historial que lleven mucho tiempo terminados y ya no sean de interés para la facultad; además de ver en cualquier momento la evolución de cada uno de ellos.
<b>Referencia</b>	R3, R4, R5, R17.

**Tabla 9.-** CU Gestionar usuario.

<b>CU-3</b>	Gestionar_Usuario.
<b>Actor</b>	Vicedecano_Producción.
<b>Descripción</b>	El Vicedecano_Producción puede crear nuevos usuarios asignados a los proyectos previamente creados, especificando el rol que permitirá establecer niveles de permisos; además de permitirle modificar los perfiles de los usuarios creados o eliminarlos.
<b>Referencia</b>	R6, R7, R8.

**Tabla 10.-** CU Gestionar autenticación.

<b>CU-4</b>	Gestionar_Autenticacion.
<b>Actor</b>	Asesor_Calidad, Encargado_Calidad Líder_Proyecto, Vicedecano_Producción.
<b>Descripción</b>	Los usuarios con privilegios pueden autenticarse para acceder a los restantes módulos del sistema. Además una vez dentro pueden cambiar su contraseña.
<b>Referencia</b>	R9, R10.

**Tabla 11.-** CU Actualizar proyecto.

<b>CU-5</b>	Actualizar_Proyecto.
<b>Actor</b>	Líder_Proyecto.
<b>Descripción</b>	El Líder_Proyecto puede llenar o modificar el expediente de proyecto que contiene sus datos básicos.
<b>Referencia</b>	R11, R12.

**Tabla 12.-** CU Gestionar entregable.

<b>CU-6</b>	Gestionar_Entregable.
<b>Actor</b>	Encargado_Calidad.
<b>Descripción</b>	El Encargado_Calidad puede llenar y modificar los entregables pertenecientes a su proyecto.
<b>Referencia</b>	R13, R14.

**Tabla 13.-** CU Gestionar reporte.

<b>CU-7</b>	Generar_Reporte.
<b>Actor</b>	Asesor_Calidad.
<b>Descripción</b>	El Asesor_Calidad puede escoger los parámetros para conformar reportes, ver el estado de los proyectos activos e imprimir ambas cosas.
<b>Referencia</b>	R15, R16, R17.

**Tabla 14.-** CU Definir documento.

<b>CU-8</b>	Definir_Documento.
<b>Actor</b>	Asesor_Calidad.
<b>Descripción</b>	El Asesor_Calidad puede subir y modificar los documentos existentes que determinan los entregables a llenar por cada etapa de control u otros de interés referentes al tema.
<b>Referencia</b>	R18, R19.

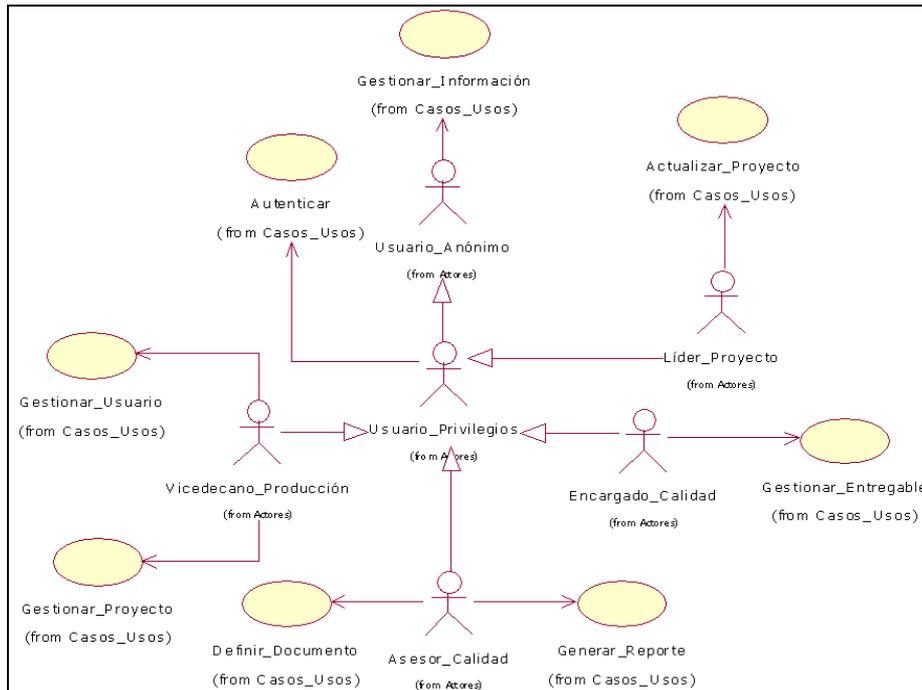


Figura 5.- Modelo de CUs del sistema.

2.7.- Expansión de los Casos de Usos

Tabla 15.- Expansión de Gestionar información.

GESTIONAR INFORMACIÓN			
<b>Actores</b>		Usuario_Anonimo.	
<b>Propósito</b>		Permitir al Usuario_Anonimo ver documentos de interés para la gestión y control de la calidad en el proceso de desarrollo del SE.	
<b>Resumen</b>		El CU inicia cuando el Usuario_Anonimo necesita buscar información referente a cómo se controla la calidad durante el proceso de desarrollo de SE, finalizando el CU cuando logre su cometido.	
<b>Referencias</b>		R1, R2.	
<b>Precondiciones</b>		Sistema disponible.	
<b>Requisitos especiales</b>		-	
Acción del autor		Respuesta del sistema	
1	El Usuario_Anonimo desde la pantalla principal ejecuta la acción Gestionar calidad.	1.1	Se ejecuta la funcionalidad Buscar información, cargándose una página con una serie de documentos de interés para la gestión de la calidad del SE.
2	El Usuario_Anonimo accede a cualquier documento.	2.1	Abre el archivo en formato PDF.
3	El Usuario_Anonimo revisa los documentos, al terminar		

	cierra el sistema, finalizando el CU.		
<b>Prioridad</b>	Secundario, ya que este CU no influye de manera determinante en la gestión de la calidad del proceso de desarrollo del SE.		
<b>Poscondiciones</b>	Búsqueda de información realizada.		

**Tabla 16.-** Expansión de Gestionar proyecto.

<b>GESTIONAR_PROYECTO</b>			
<b>Actores</b>	Vicedecano_Producción.		
<b>Propósito</b>	Permitir al Vicedecano_Producción crear, cambiar estado, ver la evolución de proyectos, cualquiera sea su estado, ver el historial de los terminados y eliminar proyectos de éstos últimos.		
<b>Resumen</b>	El CU inicia cuando el Vicedecano_Producción necesita crear un nuevo proyecto, cambiar su estado de activo a terminado o eliminar alguno que ya no sea de interés para la facultad. Terminando el CU al lograr su objetivo.		
<b>Referencias</b>	R3, R4, R5, R17.		
<b>Precondiciones</b>	Vicedecano_Producción autenticado satisfactoriamente.		
<b>Requisitos especiales</b>	Asegurarse del consentimiento del usuario antes de eliminar un proyecto.		
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
1	<p>El Vicedecano_Producción se encuentra en la Pantalla 1, puede necesitar:</p> <ul style="list-style-type: none"> <li>- Crear proyecto (ir a Escenario: Crear proyecto).</li> <li>- Listar proyectos según los criterios activos y terminados (ir a Escenario: Listar proyectos).</li> </ul>		
			
Pantalla 1.			
<b>ESCENARIO: CREAR PROYECTO</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
2	El Vicedecano_Producción ejecuta la acción Nuevo proyecto.	2.1	Se ejecuta la funcionalidad Crear proyecto, cargándose una página para introducir el nombre que lo identificará, Pantalla 2.

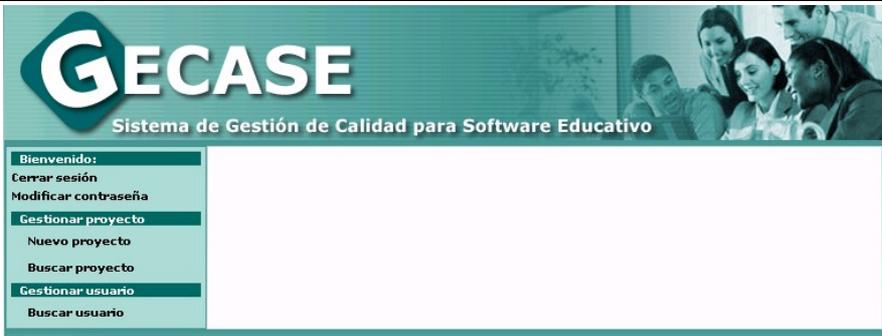
			
Pantalla 2.			
3	El Vicedecano_Producción en Pantalla 2 introduce el nombre del proyecto, al terminar presiona el botón Crear.	3.1	Guarda los datos del proyecto.
		3.2	Muestra un mensaje especificando que el proyecto ha sido insertado, finalizando el CU.
<b>Curso alternativo de los eventos</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
3.a	El Vicedecano_Producción en Pantalla 2 introduce un nombre que ya fue asignado a otro proyecto y presiona el botón Crear.	3.1	Muestra un mensaje diciendo que ya el nombre fue asignado a otro proyecto. Vuelve a la acción 3.
<b>Curso alternativo de los eventos</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
3.b	El Vicedecano_Producción en Pantalla 2 deja el campo vacío y presiona el botón Crear.	3.1	Levanta un cuadro de diálogo especificando que en el campo se debe introducir un nombre, Pantalla 3. Vuelve a la acción 3.
			
Pantalla 3.			
<b>ESCENARIO: LISTAR PROYECTOS</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
2	El Vicedecano_Producción ejecuta la acción Listar Proyectos.	2.1	Se levanta una página que permitirá buscar los proyectos según los criterios nombre y/o estado, Pantalla 4.
			
Pantalla 4.			
3	El Vicedecano_Producción al escoger el criterio de		

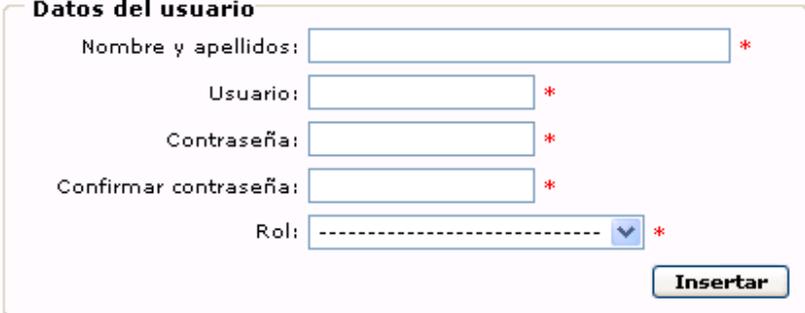
	<p>búsqueda puede:</p> <ul style="list-style-type: none"> <li>- Cambiar estado de un proyecto (ir a Escenario: Cambiar estado de proyecto).</li> <li>- Ver evolución de proyectos (ir a Escenario: Ver evolución).</li> <li>- Eliminar proyecto (ir a Escenario: Eliminar proyecto).</li> </ul>		
4	El Vicedecano_Producción al escoger el criterio de búsqueda que desee.	4.1	Muestra los proyectos buscados según el criterio de búsqueda con botones asociados para Cambiar estado (en caso de que el estado sea Activo), Evolución y Eliminar en Pantalla 5.
 <p>Pantalla 5.</p>			
<b>ESCENARIO: CAMBIAR ESTADO DE PROYECTO</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
5	El Vicedecano_Producción marca el/los proyectos activos que desee pasar al historial y presiona el botón Cambiar estado.	5.1	Levanta un cuadro de diálogo preguntando si está seguro de realizar la acción, Pantalla 6.
 <p>Pantalla 6.</p>			
5	El Vicedecano_Producción está seguro, presiona el botón OK en Pantalla 6.	5.1	Modifica el estado de el/los proyectos.
		5.2	Muestra un mensaje especificando que lo

			operación ha sido realizada con éxito, finalizando el CU.
<b>Curso alternativo de los eventos</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
5.a	El Vicedecano_Producción no está seguro, presiona el botón <i>Cancel</i> en Pantalla 6.	5.1	Vuelve a Pantalla 5.
<b>ESCENARIO: VER EVOLUCIÓN DE UN PROYECTO</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
5	El Vicedecano_Producción marca el proyecto, en cualquiera de los dos criterios, al que desee ver su evolución y presiona el botón Evolución.	5.1	Muestra los entregables que el proyecto tiene hasta el momento gestionados, así como sus datos básicos.
6	El Vicedecano_Producción estudia los datos que desee y presiona el botón Aceptar.	6.1	Vuelve a Pantalla 5.
<b>Curso alternativo de los eventos</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
5.a	El Vicedecano_Producción marca varios proyectos y presiona el botón Evolución.	4.1	Levanta un cuadro de diálogo especificando que debe escoger solo un proyecto para ver su evolución.
<b>ESCENARIO: ELIMINAR PROYECTO</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
5	El Vicedecano_Producción marca el proyecto que desee eliminar y presiona el botón Eliminar.	5.1	Levanta un cuadro de diálogo preguntando si está seguro de eliminar el proyecto, Pantalla 7.
 <p style="text-align: center;">Pantalla 7.</p>			
6	El Vicedecano_Producción está seguro de haber escogido el proyecto correcto, presiona el botón OK.	6.1	Elimina el proyecto, muestra un mensaje especificando que la operación fue realizada, finalizando el CU.
<b>Curso alternativo de los eventos</b>			

Acción del autor		Respuesta del sistema	
6.a	El Vicedecano_Producción no está seguro de haber escogido el proyecto correcto, presiona el botón <i>Cancel</i> .	6.1	Vuelve a la Acción 5.
<b>Prioridad</b>		Crítico, ya que mediante este CU se crean los proyectos a los que se les gestionará posteriormente la calidad durante el proceso de desarrollo, se modifica su estado o se eliminan.	
<b>Poscondiciones</b>		Proyecto creado, cambiado de estado o eliminado.	

**Tabla 17.-** Expansión de Gestionar usuario.

GESTIONAR_USUARIO			
<b>Actores</b>		Vicedecano_Producción.	
<b>Propósito</b>		Permitir al Vicedecano_Producción crear/eliminar usuarios y modificar su perfil.	
<b>Resumen</b>		El CU inicia cuando el Vicedecano_Producción necesita crear/eliminar o modificar el perfil de un usuario del sistema, para lo que puede entrar al módulo de Administración del sistema, que permite el manejo de los usuarios, finalizando el CU cuando logre su cometido.	
<b>Referencias</b>		R6, R7, R8.	
<b>Precondiciones</b>		-Vicedecano_Producción autenticado satisfactoriamente. -Vicedecano_Producción se encuentra en el módulo Administración del sistema.	
<b>Requisitos especiales</b>		Preguntar si está seguro de eliminar un usuario.	
Acción del autor		Respuesta del sistema	
1	El Vicedecano_Producción, desde la Pantalla 1, puede: - Crear nuevo usuario (ir a Escenario: Crear usuario). - Listar usuarios (ir a Escenario: Listar usuarios).		
			
Pantalla 1.			
ESCENARIO: CREAR USUARIO			
Acción del autor		Respuesta del sistema	
2	El Vicedecano_Producción	2.1	Se ejecuta la funcionalidad Crear proyecto.

	ejecuta la acción Nuevo proyecto.		
3	Crea el proyecto, Ver del CU Gestionar proyecto el escenario Crear proyecto.	3.1	Se ejecuta la funcionalidad Crear usuario. Se levanta una página que permite introducir los datos del nuevo usuario, existen dos tipos, los que: (Pantalla 2) - Tienen responsabilidad en un proyecto (Ir a Escenario: Responsabilidad en proyecto). - Tienen responsabilidad en la Facultad (Ir a Escenario: Responsabilidad en facultad).
 <p>Pantalla 2.</p>			
<b>ESCENARIO: CREAR USUARIO, RESPONSABILIDAD EN PROYECTO</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
3	El Vicedecano_Producción introduce todos los datos en la página, Pantalla 2, escoge como rol líder o encargado de calidad en proyecto.	3.1	Muestra un <i>select</i> con los proyectos activos, Pantalla 3.
 <p>Pantalla 3.</p>			
4	El Vicedecano_Producción escoge en Pantalla 3, el proyecto al que le asociará el usuario y presiona el botón Insertar.	4.1	Verifica que el rol del nuevo usuario no esté asignado a otro existente en el mismo proyecto. Se guardan los datos y muestra un mensaje especificando el éxito de la operación, finalizando el CU.
<b>Curso alternativo de los eventos</b>			

Acción del autor		Respuesta del sistema	
		4.a.1	Encuentra que el rol fue asignado a otro usuario en el mismo proyecto. Muestra un mensaje especificando que debe cambiar de rol. Vuelve a la acción 3.
<b>ESCENARIO: CREAR USUARIO, RESPONSABILIDAD EN FACULTAD</b>			
Acción del autor		Respuesta del sistema	
3	El Vicedecano_Producción introduce todos los datos en la página, Pantalla 2, escoge como rol asesor de calidad.	3.1	Verifica que no exista otro asesor de calidad. Se guardan los datos, muestra un mensaje especificando el éxito de la operación (Pantalla 4), finalizando el CU.
<b>Curso alternativo de los eventos</b>			
Acción del autor		Respuesta del sistema	
		3.a.1	Encuentra que existe otro asesor de calidad. Muestra un mensaje especificando que debe cambiar el rol. Vuelve a la acción 3.
<b>Curso alternativo de los eventos</b>			
Acción del autor		Respuesta del sistema	
3.a	El Vicedecano_Producción no introduce todos los datos obligatorios en Pantalla 2 y presiona el botón Crear.	3.1	Levanta un cuadro de diálogo especificando que se deben llenar todos los datos obligatorios, Pantalla 4. Vuelve a la acción 3.
<div style="border: 1px solid gray; padding: 10px;"> <p><b>Datos del usuario</b></p> <p>Nombre y apellidos: <input type="text" value="Osiris Perez Moya"/> *</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <p><b>Microsoft Internet Explorer</b></p> <p> Los campos que tienen asterisco son de carácter obligatorio.</p> <p style="text-align: center;"><input type="button" value="OK"/></p> </div> <p style="text-align: right;"><input type="button" value="Insertar"/></p> </div> <p style="text-align: center;">Pantalla 4.</p>			
<b>ESCENARIO: LISTAR USUARIOS</b>			
Acción del autor		Respuesta del sistema	
2	El Vicedecano_Producción ejecuta la acción Listar usuarios.	2.1	Se levanta una página para escoger un criterio de búsqueda, Pantalla 5.
<div style="border: 1px solid gray; padding: 10px;"> <p><b>Criterio de búsqueda</b></p> <p>Nombre y apellidos: <input type="text"/></p> <p>Usuario: <input type="text"/></p> <p>Rol: <input type="text" value="-----"/> ▼</p> <p style="text-align: right;"><input type="button" value="Buscar"/></p> </div> <p style="text-align: center;">Pantalla 5.</p>			

3	El Vicedecano_Producción escoge un criterio y presiona el botón Buscar, Pantalla 5.	3.1	Se levanta una página con los usuarios que cumplan el criterio, Pantalla 6.																				
<div style="border: 1px solid black; padding: 5px;"> <p><b>Listado de usuarios</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #006666; color: white;"> <th style="width: 5%;"></th> <th style="width: 30%;">Nombre</th> <th style="width: 15%;">Usuario</th> <th style="width: 20%;">Rol</th> <th style="width: 30%;">Proyecto</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>Carlos Vazquez Ortiz</td> <td>cvazquez</td> <td>Líder de proyecto</td> <td>GPI</td> </tr> <tr> <td><input type="checkbox"/></td> <td>ee</td> <td>ee</td> <td>Líder de proyecto</td> <td>D</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Yudermis</td> <td>y</td> <td>Líder de proyecto</td> <td>CICPC</td> </tr> </tbody> </table> <div style="text-align: right; margin-top: 5px;"> <input type="button" value="Modificar perfil"/> <input type="button" value="Eliminar"/> </div> </div> <p style="text-align: center;">Pantalla 6.</p>					Nombre	Usuario	Rol	Proyecto	<input type="checkbox"/>	Carlos Vazquez Ortiz	cvazquez	Líder de proyecto	GPI	<input type="checkbox"/>	ee	ee	Líder de proyecto	D	<input type="checkbox"/>	Yudermis	y	Líder de proyecto	CICPC
	Nombre	Usuario	Rol	Proyecto																			
<input type="checkbox"/>	Carlos Vazquez Ortiz	cvazquez	Líder de proyecto	GPI																			
<input type="checkbox"/>	ee	ee	Líder de proyecto	D																			
<input type="checkbox"/>	Yudermis	y	Líder de proyecto	CICPC																			
4	El Vicedecano_Producción puede en Pantalla 6: - Modificar perfil (ir a Escenario: Modificar perfil). - Eliminar usuario (ir a Escenario: Eliminar usuario).																						
<b>ESCENARIO: MODIFICAR PERFIL</b>																							
<b>Acción del autor</b>		<b>Respuesta del sistema</b>																					
5	El Vicedecano_Producción escoge el usuario al que desea modificar su perfil y presiona el botón Modificar, Pantalla 6.	5.1	Se ejecuta la funcionalidad Modificar perfil.																				
		5.2	Levanta una página con los datos del usuario habilitados para ser modificados, Pantalla 7.																				
<div style="border: 1px solid black; padding: 5px;"> <p><b>Datos del usuario</b></p> <p>Nombre y apellidos: <input style="width: 60%;" type="text"/></p> <p>Nueva contraseña: <input style="width: 40%;" type="password"/></p> <p>Confirmar contraseña: <input style="width: 40%;" type="password"/></p> <div style="text-align: right; margin-top: 5px;"> <input type="button" value="Actualizar"/> </div> </div> <p style="text-align: center;">Pantalla 7.</p>																							
6	El Vicedecano_Producción modifica los datos necesarios y presiona el botón Actualizar, Pantalla 7.	6.1	Se actualizan los datos, finalizando el CU.																				
<b>Curso alternativo de los eventos</b>																							
<b>Acción del autor</b>		<b>Respuesta del sistema</b>																					
5.a	El Vicedecano_Producción escoge varios usuarios y presiona el botón Modificar, Pantalla 7.	5.1	Muestra un mensaje especificando que para modificar el perfil debe escoger solo un usuario, Pantalla 8.																				



Pantalla 8.

ESCENARIO: ELIMINAR USUARIO			
Acción del autor		Respuesta del sistema	
5	El Vicedecano_Producción escoge el/los usuarios que desee eliminar y presiona el botón Eliminar, Pantalla 6.	5.1	Se ejecuta la funcionalidad Eliminar usuario.
		5.2	Levanta un cuadro de diálogo preguntando si está seguro de eliminar el usuario, Pantalla 9.



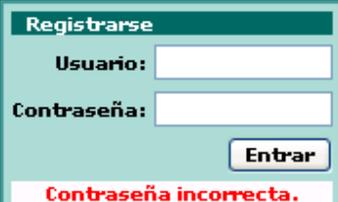
Pantalla 9.

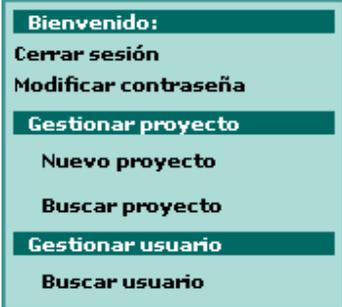
6	El Vicedecano_Producción está seguro de haber escogido el/los usuarios correctos, presiona el botón OK, Pantalla 9.	6.1	Elimina el/los usuarios, finalizando el CU.
---	---	-----	---

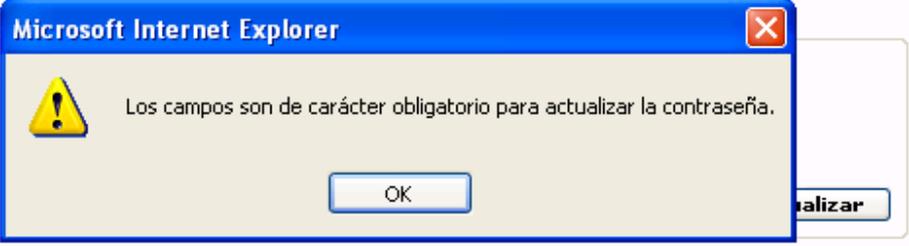
Curso alternativo de los eventos			
Acción del autor		Respuesta del sistema	
6.a	El Vicedecano_Producción no está seguro de haber escogido el/los usuarios correctos, presiona el botón <i>Cancel</i> , Pantalla 9.	6.1	Vuelve a la Pantalla 6.
<b>Prioridad</b>	Crítico, ya que mediante este CU se estratifican los niveles de permisos al sistema.		
<b>Poscondiciones</b>	Usuario creado/eliminado o con el perfil modificado.		

**Tabla 18.-** Expansión de Gestionar autenticación.

GESTIONAR_AUTENTICACION	
<b>Actores</b>	Usuario_Privilegios.
<b>Propósito</b>	Permitir a los actores del sistema entrar a los módulos donde se tienen privilegios o cambiar su contraseña una vez dentro.
<b>Resumen</b>	El CU inicia cuando un actor del sistema necesita trabajar con un módulo para el que se necesita tener privilegios por lo que debe autenticarse o cambiar su contraseña una vez que se

	haya autenticado, finalizando el CU al lograr su cometido.	
<b>Referencias</b>	R9, R10.	
<b>Precondiciones</b>	Sistema instalado satisfactoriamente.	
<b>Requisitos especiales</b>	-	
<b>ESCENARIO: AUTENTICAR USUARIO</b>		
	<b>Acción del autor</b>	<b>Respuesta del sistema</b>
1	El actor desea entrar a algún módulo con privilegios, para lo que debe introducir usuario y contraseña en Pantalla 1.	1.1 Los datos entrados coinciden con los registrados en la BD, se carga una página en dependencia del rol del usuario autenticado (Vicedecano_Producción, Encargado_Calidad o Asesor_Calidad), terminando el CU.
 <p style="text-align: center;">Pantalla 1.</p>		
<b>Curso alternativo de los eventos</b>		
	<b>Acción del autor</b>	<b>Respuesta del sistema</b>
		1.1.a El usuario no existe, en la BD, se muestra un mensaje de error, Pantalla 2. Vuelve a Pantalla 1.
 <p style="text-align: center;">Pantalla 2.</p>		
		1.1.b La contraseña es incorrecta, se muestra un mensaje de error, Pantalla 3. Vuelve a Pantalla 1.
 <p style="text-align: center;">Pantalla 3.</p>		
<b>ESCENARIO: CAMBIAR CONTRASEÑA</b>		

Acción del autor		Respuesta del sistema	
1	El actor necesita cambiar su contraseña, se encuentra logueado en el sistema según su rol, Pantalla 4, y presiona en el vínculo Cambiar contraseña.	1.1	Se ejecuta la funcionalidad Cambiar contraseña, cargándose una página con los campos necesarios para cambiarla, Pantalla 5.
 <p>Pantalla 4.</p>			
 <p>Pantalla 5.</p>			
2	El actor introduce su nueva contraseña y presiona el botón Actualizar, Pantalla 4.	2.1	Levanta un cuadro de diálogo preguntando si está seguro de realizar la acción, Pantalla 6.
 <p>Pantalla 6.</p>			
3	El actor está seguro, presiona el botón OK en Pantalla 6.	3.1	Verifica que las contraseñas coincidan, la actualiza en la BD, finalizando el CU.
<b>Curso alternativo de los eventos</b>			
Acción del autor		Respuesta del sistema	
3.a	El actor no está seguro, presiona el botón <i>Cancel</i> en Pantalla 6.	3.1	Vuelve a la acción 2.

		3.1.a	Verifica que las contraseñas no coinciden, muestra un mensaje especificando que las contraseñas deben ser iguales, Pantalla 7. Vuelve a la acción 2.
 <p>Pantalla 7.</p>			
4.a	El actor presiona el botón OK en Pantalla 7.	4.1.a	Vuelve a la acción 2.
2.a	El actor no introduce contraseña en los campos y presiona el botón Actualizar.	2.1	Muestra un mensaje de error especificando que los campos son de carácter obligatorio, Pantalla 8.
 <p>Pantalla 8.</p>			
<b>Prioridad</b>	Crítico, ya que mediante este CU se restringe a los diferentes actores del sistema que tienen privilegios, a los requerimientos que le son innecesarios; y a aquellos que no lo son les es negado su acceso al módulo.		
<b>Poscondiciones</b>	<ul style="list-style-type: none"> <li>- Acceso dado al usuario si es quien dice ser.</li> <li>- Acceso denegado al usuario si no es quien dice ser.</li> <li>- Contraseña cambiada.</li> </ul>		

**Tabla 19.-** Expansión de Actualizar proyecto.

<b>ACTUALIZAR_PROYECTO</b>	
<b>Actores</b>	Líder_Proyecto.
<b>Propósito</b>	Permitir al Líder_Proyecto llenar y modificar el expediente de proyecto con algunos datos de interés.
<b>Resumen</b>	El CU inicia cuando el Líder_Proyecto entra al sistema por primera vez y necesita llenar el expediente del proyecto o cuando necesita modificarlo. Terminando el CU al lograr su objetivo.
<b>Referencias</b>	R11, R12.
<b>Precondiciones</b>	- Líder_Proyecto autenticado satisfactoriamente.

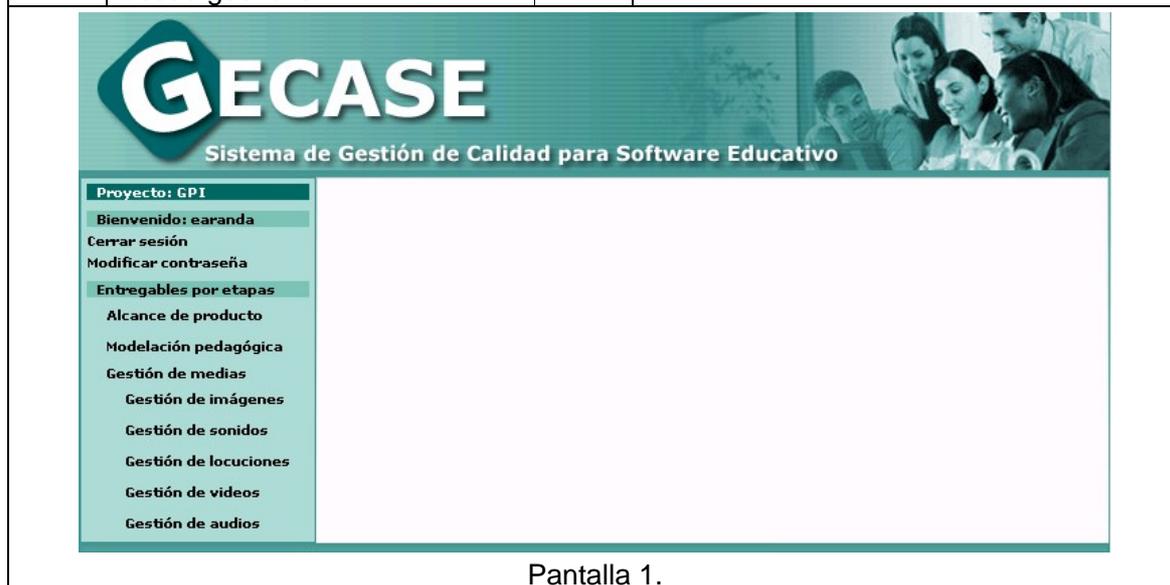
		- Proyecto creado previamente en caso de que se quiera modificar.	
<b>Requisitos especiales</b>		Asegurarse del consentimiento del usuario antes de modificar los datos del proyecto.	
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
1	El Líder_Proyecto se encuentra en la pantalla principal del Líder de proyecto, puede necesitar: - Llenar expediente de proyecto (ir a Escenario: Llenar expediente). - Modificar expediente de proyecto (ir a Escenario: Modificar expediente).		
<b>ESCENARIO: LLENAR EXPEDIENTE DE PROYECTO</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
2	El Líder_Proyecto ejecuta la acción Datos de proyecto.	2.1	Se ejecuta la funcionalidad Llenar expediente, cargándose una página donde aparecen los documentos a subir.
3	El Líder_Proyecto sube los documentos necesarios, al terminar presiona el botón Guardar.	3.1	Guarda los datos del proyecto, finalizando el CU.
<b>Curso alternativo de los eventos</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
3.a	El Líder_Proyecto presiona el botón Guardar sin haber subido ningún documento.	3.1	Levanta un cuadro de diálogo especificando que debe subir al menos un documento.
4	El Líder_Proyecto presiona el botón en Aceptar.	4.1	Regresa a la acción 3.
<b>ESCENARIO: MODIFICAR EXPEDIENTE DE PROYECTO</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
2	El Líder_Proyecto ejecuta la acción Datos de proyecto.	2.1	Muestra los datos del proyecto previamente llenados.
3	El Líder_Proyecto presiona el botón Modificar.	3.1	Se ejecuta la funcionalidad Modificar expediente, habilitándose los datos del expediente para ser modificados.
4	El Líder_Proyecto modifica los datos que crea necesario, al terminar presiona el botón Salvar.	4.1	Levanta un cuadro de diálogo preguntando si está seguro el usuario de modificar los datos.
5	El Líder_Proyecto está seguro de modificar los datos, presiona el botón OK.	5.1	Guarda los nuevos datos, finalizando el CU.
<b>Curso alternativo de los eventos</b>			

Acción del autor		Respuesta del sistema	
5.a	El Líder_Proyecto no está seguro de modificar los datos, presiona el botón <i>Cancel</i> .	5.1	Vuelve a la acción 2.
<b>Prioridad</b>	Secundario, ya que mediante los datos básicos que se gestionan del proyecto, no influyen de manera determinante en el logro de productos con calidad.		
<b>Poscondiciones</b>	Expediente de proyecto actualizado.		

**Tabla 20.-** Expansión de Gestionar entregable.

GESTIONAR ENTREGABLE	
<b>Actores</b>	Encargado_Calidad.
<b>Propósito</b>	Permitir al Encargado_Calidad llenar/subir los entregables en cada una de las etapas de control y modificarlos en caso de que sea necesaria una actualización.
<b>Resumen</b>	El CU inicia cuando el Encargado_Calidad necesita llenar/subir los entregables en cada una de las etapas de control durante el proceso de desarrollo del software; o en caso de que necesite modificar alguno de dichos entregables, finalizando el CU cuando logre su cometido.
<b>Referencias</b>	R13, R14.
<b>Precondiciones</b>	Encargado_Calidad autenticado satisfactoriamente.
<b>Requisitos especiales</b>	Preguntar si está seguro de salvar los cambios realizados al modificar un entregable.

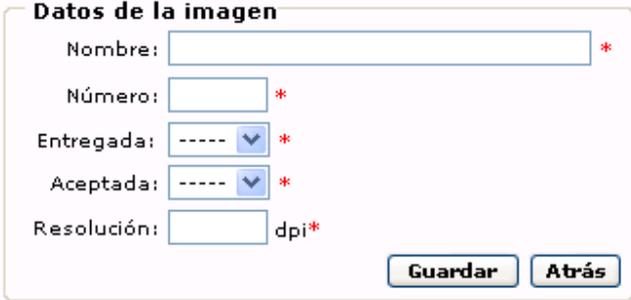
Acción del autor		Respuesta del sistema	
1	El Encargado_Calidad, desde la Pantalla 1 ejecuta la acción asociada al entregable que desee gestionar.	1.1	Se ejecuta la funcionalidad Listar entregables, cargándose una página con la lista de los existentes, Pantalla 2.



Pantalla 1.

<div style="border: 1px solid black; padding: 5px;"> <p><b>Listado de imágenes</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #006666; color: white;"> <th style="width: 5%;"></th> <th style="width: 10%;">Id</th> <th style="width: 35%;">Nombre</th> <th style="width: 10%;">Aceptada</th> <th style="width: 10%;">Entregada</th> <th style="width: 10%;">Resolución</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td>I3</td> <td>GFDGDFG</td> <td style="text-align: center;">Sí</td> <td style="text-align: center;">No</td> <td style="text-align: center;">3 dpi</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td>I34</td> <td>TERTER</td> <td style="text-align: center;">Sí</td> <td style="text-align: center;">No</td> <td style="text-align: center;">234 dpi</td> </tr> </tbody> </table> <p style="text-align: right;"> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/> <input type="button" value="Nueva"/> </p> </div>					Id	Nombre	Aceptada	Entregada	Resolución	<input type="checkbox"/>	I3	GFDGDFG	Sí	No	3 dpi	<input type="checkbox"/>	I34	TERTER	Sí	No	234 dpi
	Id	Nombre	Aceptada	Entregada	Resolución																
<input type="checkbox"/>	I3	GFDGDFG	Sí	No	3 dpi																
<input type="checkbox"/>	I34	TERTER	Sí	No	234 dpi																
Pantalla 2.																					
2	El Encargado_Calidad, desde la Pantalla 2, puede: <ul style="list-style-type: none"> <li>- Modificar entregable (ir a Escenario: Modificar entregable).</li> <li>- Eliminar entregable (ir a Escenario: Eliminar entregable).</li> <li>- Insertar un nuevo entregable (ir a Escenario: Nuevo entregable).</li> </ul>																				
<b>Curso alternativo de los eventos</b>																					
<b>Acción del autor</b>		<b>Respuesta del sistema</b>																			
		1.1.a	Se ejecuta la funcionalidad Listar entregables, no existen entregables registrados, cargándose una página con el formulario para llenar uno. Ir a escenario Nuevo entregable.																		
<b>ESCENARIO: MODIFICAR ENTREGABLE</b>																					
<b>Acción del autor</b>		<b>Respuesta del sistema</b>																			
3	El Encargado_Calidad escoge el entregable que desee actualizar y presiona el botón Modificar.	3.1	Se ejecuta la funcionalidad Modificar entregable, cargándose una página con los datos del entregable habilitados para ser llenados, Pantalla 3.																		
<div style="border: 1px solid black; padding: 5px;"> <p><b>Datos de la imagen</b></p> <p>Nombre: <input type="text" value="Cascada"/> *</p> <p>Número: <input type="text" value="34"/> *</p> <p>Entregada: <input type="button" value="Sí"/> *</p> <p>Aceptada: <input type="button" value="Sí"/> *</p> <p>Resolución: <input type="text" value="31"/> dpi*</p> <p style="text-align: right;"> <input type="button" value="Actualizar"/> <input type="button" value="Atrás"/> </p> </div>																					
Pantalla 3.																					
4	El Encargado_Calidad modifica los datos necesarios presiona el botón Actualizar, Pantalla 3.	4.1	Levanta un cuadro de diálogo pregunta por la seguridad del actor para realizar la acción, Pantalla 4.																		

			
Pantalla 4.			
5	El Encargado_Calidad está seguro y presiona el botón OK.	5.1	Guarda los nuevos datos, finalizando el CU.
<b>Curso alternativo de los eventos</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
5.a	El Encargado_Calidad no está seguro y presiona el botón <i>Cancel</i> .	5.1	Vuelve a Pantalla 3.
<b>ESCENARIO: ELIMINAR ENTREGABLE</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
3	El Encargado_Calidad marca el/los entregable(s) que desee eliminar y presiona el botón Eliminar, Pantalla 2.	3.1	Levanta un cuadro de diálogo preguntando al actor si está seguro de realizar la acción, Pantalla 5.
			
Pantalla 5.			
3	El Encargado_Calidad está seguro y presiona el botón OK.	3.1	Se ejecuta la funcionalidad Eliminar entregable, eliminando los elementos seleccionados. Finaliza el CU.
<b>Curso alternativo de los eventos</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
3.a	El Encargado_Calidad no está seguro y presiona el botón <i>Cancel</i> .	3.1	Vuelve a Pantalla 2.
<b>ESCENARIO: NUEVO ENTREGABLE</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
3	El Encargado_Calidad en Pantalla 2 presiona el botón Nueva(o).	3.1	Se ejecuta la funcionalidad Insertar nuevo entregable, cargándose una página con los datos a llenar en dependencia del que se esté

			gestionando, Pantalla 6.
 <p>Pantalla 6.</p>			
4	El Encargado_Calidad llena todos los datos y presiona el botón Guardar.	3.1	Guarda los nuevos datos, finalizando el CU.
<b>Curso alternativo de los eventos</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
4.a	El Encargado_Calidad deja de llenar al menos un dato y presiona el botón Guardar.	4.1	Levanta un cuadro de diálogo especificando los campos que son obligatorios, Pantalla 7.
 <p>Pantalla 7.</p>			
5	El Encargado_Calidad presiona el botón OK.	5.1	Vuelve a Pantalla 6.
<b>Prioridad</b>	Crítico, ya que mediante este CU se gestionan los entregables que permitirán al Asesor_Calidad posteriormente evaluar el proceso de desarrollo del SE para lograr un producto con calidad en los proyectos de forma individual.		
<b>Poscondiciones</b>	Entregable actualizado.		

**Tabla 21.-** Expansión de Generar reporte.

<b>GESTIONAR REPORTE</b>	
<b>Actores</b>	Asesor_Calidad.
<b>Propósito</b>	Permitir al Asesor_Calidad ver el estado de los proyectos de producción, así como generar reportes que evalúen su proceso de desarrollo e imprimirlos en caso necesario.
<b>Resumen</b>	El CU inicia cuando el Asesor_Calidad necesita evaluar la calidad de un determinado proyecto, para lo que puede ver el

	estado del proyecto, así como generar reportes de él, finalizando el CU cuando logre su cometido.		
<b>Referencias</b>	R15, R16, R17.		
<b>Precondiciones</b>	-Asesor_Calidad autenticado satisfactoriamente. -Expediente de proyecto creado previamente.		
<b>Requisitos especiales</b>	-		
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
1	El Asesor_Calidad ejecuta la acción Escoger parámetros.	1.1	Se carga una página para escoger los parámetros que lo conformarán así como los valores asociados a ellos.
2	El Asesor_Calidad escoge los parámetros necesarios y presiona el botón Generar reporte.	2.1	Se ejecuta la funcionalidad Generar reporte, mostrándose un PDF con los resultados.
3	El Asesor_Calidad revisa los resultados y puede imprimirlos o descargarlos en caso necesario.	3.1	Finaliza el CU.
<b>Prioridad</b>	Crítico, ya que mediante este CU, se generan los reportes que muestran los proyectos que cumplan con una serie de parámetros escogidos por el Asesor_Calidad.		
<b>Poscondiciones</b>	Reportes creados e impresos y/o descargados.		

**Tabla 22.-** Expansión de Definir documento.

<b>DEFINIR DOCUMENTO</b>			
<b>Actores</b>	Asesor_Calidad.		
<b>Propósito</b>	Permitir al Asesor_Calidad modificar los documentos de interés para la gestión y control de la calidad en el proceso de desarrollo del SE.		
<b>Resumen</b>	El CU inicia cuando el Asesor_Calidad necesita modificar algún documento, para lo que puede entrar al módulo de Administración de proyectos por facultad, donde una opción permite llevar a cabo su cometido, finalizando el CU al lograrlo.		
<b>Referencias</b>	R18, R19.		
<b>Precondiciones</b>	-Asesor_Calidad autenticado satisfactoriamente. -Asesor_Calidad se encuentra en el módulo de Administración de proyectos por facultad.		
<b>Requisitos especiales</b>	Preguntar si está seguro de modificar el documento.		
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
1	El Asesor_Calidad puede: - Modificar un documento (ir a Escenario: Modificar documento). - Insertar documento (ir a Escenario: Insertar documento).		Levanta una página con los documentos disponibles y una opción para agregar uno nuevo.

	- Eliminar documento (ir a Escenario: Eliminar documento). Para lo que ejecuta la acción Documentos de interés.		
<b>ESCENARIO: SOBRESCRIBIR DOCUMENTO</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
2	El Asesor_Calidad presiona el botón <i>Browse</i> asociado al documento que desee sobrescribir.	2.1	Se ejecuta la funcionalidad Sobrescribir documento. Se levanta una ventana para buscar el documento a subir.
3	El Asesor_Calidad buscar el archivo y presiona el botón Guardar.	3.1	Se levanta un cuadro de diálogo preguntando si está seguro de modificar el documento existente.
4	El Asesor_Calidad está seguro de modificar el documento, presiona el botón Aceptar.	4.1	Se sobrescribe el archivo, terminando el CU.
<b>Curso alternativo de los eventos</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
4.a	El Asesor_Calidad no está seguro de haber escogido el archivo correcto, presiona el botón <i>Cancel</i> .	4.1	Vuelve a la acción 2.
<b>ESCENARIO: INSERTAR NUEVO DOCUMENTO</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
2	El Asesor_Calidad presiona el botón Nuevo.	2.1	Muestra un formulario que permite introducir el nombre del nuevo documento y un campo de tipo archivo para subirlos.
3	El Asesor_Calidad busca el archivo que desee subir y presiona el botón Guardar.	3.1	Guarda el nuevo a archivo, terminando el CU.
<b>ESCENARIO: ELIMINAR DOCUMENTO</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
2	El Asesor_Calidad presiona el botón Eliminar asociado al documento que desee eliminar.	2.1	Levanta un cuadro de diálogo preguntando si está seguro de realizar la acción.
3	El Asesor_Calidad está seguro, presiona el botón OK.	3.1	Se ejecuta la funcionalidad Eliminar documento, borrando el archivo. Finaliza el CU.
<b>Curso alternativo de los eventos</b>			
<b>Acción del autor</b>		<b>Respuesta del sistema</b>	
3.a	El Asesor_Calidad no está seguro, presiona el botón <i>Cancel</i> .	3.1	Vuelve a la acción 2.
<b>Prioridad</b>	Secundario, ya que este CU no influye de manera decisiva en		

	la gestión de la calidad.
<b>Poscondiciones</b>	Documento sobrescrito y/o subido.

### **Conclusiones**

Este capítulo fue el comienzo del desarrollo de la propuesta de solución que se obtuvo a partir del análisis de los procesos del negocio, de los requisitos que debe cumplir el sistema, agrupados en CUs y que se representaron finalmente en un diagrama de CU, la base del futuro sistema.

# Capítulo 3

## Análisis y diseño del sistema

### Introducción

En este capítulo se modelarán los artefactos de entrada para la implementación de la propuesta. Para ello se hará un estudio que permitirá un esbozo de las clases a implementar a través del Análisis, las que finalmente se representarán en un diagrama de clases Web del Diseño. Además se presentará el modelo de datos, base para construir la BD que soportará la propuesta.

### 3.1.- Modelo de análisis

En este subepígrafe se muestran los diagramas de clases del análisis, agrupados por CU para que pueda entenderse mejor la lógica del negocio, resultando los paquetes:

- Paq. 1: Gestionar\_Informacion.
- Paq. 2: Gestionar\_Proyecto.
- Paq. 3: Gestionar\_Usuario.
- Paq. 4: Gestionar\_Autenticacion.
- Paq. 5: Actualizar\_Proyecto.
- Paq. 6: Gestionar\_Entregable.
- Paq. 7: Generar\_Reporte.
- Paq. 8: Definir\_Documento.

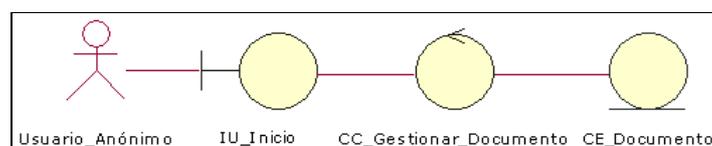


Figura 6.- Diagrama de clases del Análisis de Gestionar información.

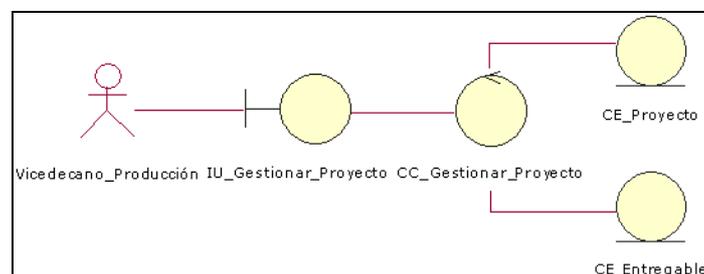
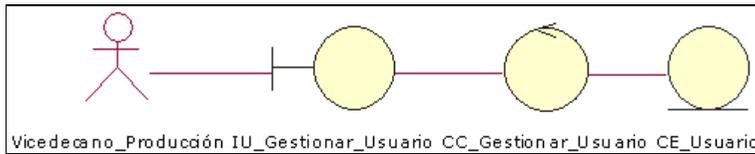
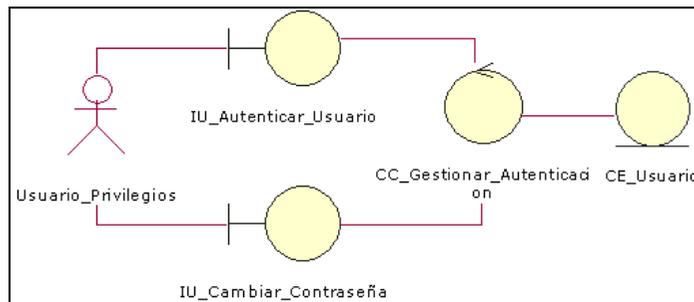


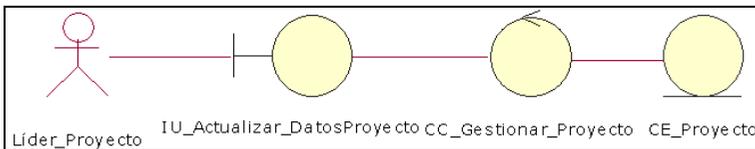
Figura 7.- Diagrama de clases del Análisis de Gestionar proyecto.



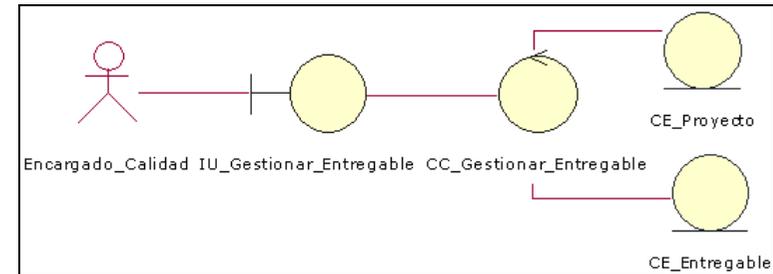
**Figura 8.-** Diagrama de clases del Análisis de Gestionar usuario.



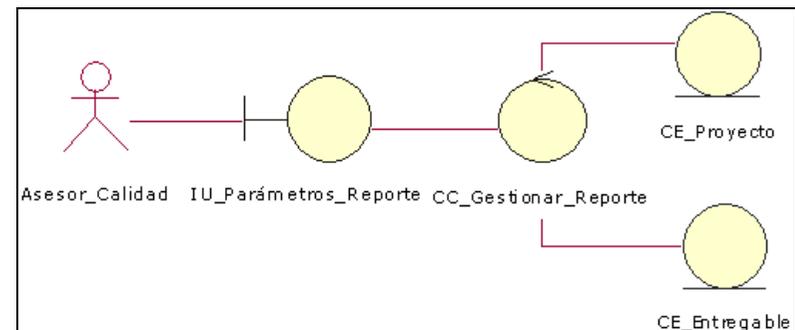
**Figura 9.-** Diagrama de clases del Análisis de Gestionar Autenticación.



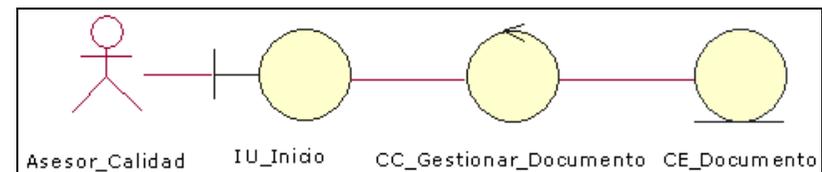
**Figura 10.-** Diagrama de clases del Análisis de Actualizar proyecto.



**Figura 11.-** Diagrama de clases del Análisis de Gestionar entregable.



**Figura 12.-** Diagrama de clases del Análisis de Generar reporte.



**Figura 13.-** Diagrama de clases del Análisis de Definir documento.

### 3.2.- Modelo de diseño

Para hacer un diseño eficiente se tomaron en cuenta un conjunto de patrones, que al ser experiencias de diseñadores expertos en orientación a objetos permiten dar solución a problemas a través de la codificación del conocimiento y principios existentes, facilitando notablemente el trabajo posterior. Los patrones son:

- Soluciones concretas: proponen soluciones a problemas concretos, son teorías genéricas.
- Soluciones técnicas: indican soluciones basadas en programación orientada a objetos (OO).
- Se utilizan en situaciones frecuentes: se basan en la experiencia acumulada por los que resuelven problemas reiterativos.
- No se refleja en el código su uso.

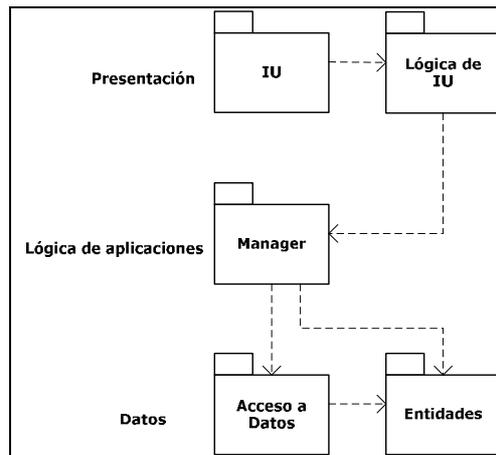
Se utilizaron fundamentalmente tres tipos de patrones:

1. Patrones de organización: patrón de tres capas.
2. Patrones de distribución: cliente/servidor. Básicamente este patrón se definió al escoger la plataforma de desarrollo Web, explicado en el capítulo 1 del presente trabajo.
3. Patrones de diseño: GRASP y GoF.

Se utiliza la arquitectura de tres capas ya que ofrece entre sus principales ventajas las que se listan a continuación: [LARMAN 2004]

- Aísla la lógica de la aplicación y la convierte en una capa intermedia bien definida y lógica del software.
- En la capa de presentación se realiza relativamente poco procesamiento de la aplicación.
- Minimiza las relaciones de los elementos componentes de las diferentes capas (los cambios sobre una capa no influyen, o muy poco, sobre las demás).
- Facilita la reutilización.

La arquitectura de tres capas simplifica la comprensión y organización del desarrollo del sistema, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle de las superiores.



**Figura 14.-** Arquitectura de tres capas.

Los patrones GRASP (*General Responsibility Assignment Software Patterns*, patrones de los principios generales para asignar responsabilidades) permiten asignar correctamente las responsabilidades a cada una de las clases que intervienen en el modelo. Se utilizaron cuatro de los cinco patrones GRASP fundamentales:

- Experto: se asignaron responsabilidades a las clases con la información necesaria para cumplirla, por ejemplo: la clase que puede brindar los valores del identificador o del nombre del usuario es la clase e\_Usuario, que es la que contiene dichos atributos.
- Creador: se asignaron responsabilidades a las clases de crear instancias de otras conociendo que las primeras son las que contienen la información para ello, por ejemplo: la clase m\_Usuario le brinda a la clase ad\_Usuario un objeto de tipo e\_Usuario, por tanto m\_Usuario es la idónea para crear una instancia de e\_Usuario.
- Alta cohesión: se asignaron responsabilidades a las clases de manera que todos sus métodos tuvieran un comportamiento bien definido, por ejemplo: la responsabilidad de la clase ad\_Usuario es la de acceder a los valores y métodos de la clase e\_Usuario, de manera que en la clase ad\_Usuario no existe ningún

método que sea para acceder a e\_Proyecto, por ejemplo, o a cualquier otra clase, a no ser que dichas clases estén estrechamente relacionadas.

- Bajo acoplamiento: cada clase está acoplada a las clases estrictamente necesarias, por ejemplo: la clase sp\_Insertar\_Usuario no conoce la existencia de la clase e\_Usuario que es la que contiene los datos que ella pide, simplemente conoce que la fachada le pasa los valores solicitados.

De los patrones GoF (*Gang of Four Patterns*, patrones de la pandilla de cuatro) se utilizó el patrón Fachada, que provee una interfaz unificada simple para acceder al subsistema *Manager* desde el subsistema Lógica de interfaz de usuario, evitando de esta forma el alto acoplamiento que se crearía entre ambos paquetes y a su vez permitiendo la independencia entre ellos.

### 3.2.1.- Descripción de las clases

Para la definición de las clases de diseño se utilizará como convenio:

- Nombre de la clase: comenzará con un identificador del tipo de clase (entidad: e, acceso a datos: ad, *manager*: m, lógica de interfaz de usuario: sp, interfaz de usuario: cp) seguido de un guión bajo (\_), por ejemplo para la clase entidad Usuario sería: e\_Usuario.
- Parámetros de métodos: comenzarán con una a, ejemplo: ald\_Usuario.
- Los métodos se pondrán con el prefijo que los identifique y el nombre asociado comenzado con mayúscula.
- Las clases, atributos y métodos no se tildarán ni se pondrán con letras específicas del español.

#### 3.2.1.1.- Clases entidad

Las clases entidad poseen un conjunto de atributos que no se describirán en este subepígrafe por ser los mismos del subepígrafe de la descripción de las tablas de la BD, ver subepígrafe 3.2.4.1. Cada clase tiene como métodos:

- Un constructor: se llama igual que la clase y contiene cada uno de los atributos de la misma, excepto el Id en alguna de ellas que será auto-numérico en la BD (las que no cumplan la condición se identificarán con un \*), ejemplo: para la clase

e\_Usuario el constructor sería e\_Usuario(ald\_Usuario, aNombre\_Apellidos, aContrasenna, ald\_Rol).

- Métodos *get*, que retornan el valor del atributo: se pondrán con minúscula, seguidos del nombre del atributo que tiene asociado, ejemplo: getId\_Usuario().
- Métodos *set*, que modifican el valor del atributo: se pondrán con minúscula, seguidos del nombre del atributo que tiene asociado, ejemplo: setId\_Usuario(ald\_Usuario).

En la propuesta se definen como clases entidad:

- e\_Usuario\*.
- e\_Proyecto.
- e\_Entregable.
- e\_Media.
- e\_Imagen.
- e\_SonidoLocucionAudio.
- e\_Video.
- e\_ArchivoAuxiliar.
- e\_ModelacionPedagogica.
- e\_AlcanceProducto.
- e\_DefinicionEntregables.
- e\_Entidad.
- e\_Contacto\*.

### 3.2.1.2.- Clases de acceso a datos

Las clases de acceso a datos son las realizan las consultas a la BD. Existe una clase por cada clase entidad, son clases PHP puras que no contienen atributos, sólo métodos que tienen las consultas necesarias a la BD; a dichos métodos generalmente se les pasa un objeto del tipo entidad asociado a la clase de acceso a datos, por lo que no se pondrán los parámetros a no ser en caso necesario, que se especificará en la descripción.

**Tabla 23.-** Clase ad\_Usuario.

Método	Descripción
Existe_Usuario()	Determina si el Id de usuario fue asignado a otro.
Existe_Rol()	Determina si el Id del rol fue asignado a otro usuario.
Insertar_UsuarioF()	Inserta un usuario en la BD con responsabilidad en la facultad.
Insertar_UsuarioP()	Inserta un usuario en la BD con responsabilidad en un proyecto.
Rol_Usuario()	Retorna el rol de un usuario determinado.
Existen_Usuarios()	Determina si existe al menos un usuario registrado.
Buscar_Nombre()	Retorna el nombre de un usuario determinado.

Eliminar_Usuarios(arregloU)	Elimina un arreglo de usuarios.
Modificar_NombreUsuario()	Modifica el nombre de un usuario determinado.
Modificar_PassUsuario()	Modifica la contraseña de un usuario determinado.
Comparar_Contrasenna()	Compara la contraseña registrada con una pasada.

**Tabla 24.-** Clase ad\_Proyecto.

Método	Descripción
Existe_Nombre_Proyecto()	Determina si existe el nombre del proyecto.
Insertar_Proyecto()	Inserta un nuevo proyecto.
Existen_Proyectos()	Determina si existe al menos un proyecto registrado.
Proyectos_Activos()	Retorna los proyectos con estado activo.
Proyectos_Terminados()	Retorna los proyectos con estado terminado.
Eliminar_Proyectos(arreglo)	Elimina un arreglo de proyectos.
Cambiar_Estado()	Modifica el estado del proyecto de activo a terminado.
Nombre_Proyecto(arreglo)	Retorna los nombres de un arreglo de proyectos determinado.

**Tabla 25.-** Clase ad\_Entregable.

Método	Descripción
Existe_Entregable(idEntreg)	Determina si existe un entregable por su identificador.
Eliminar_Entregables(arr)	Elimina un arreglo de entregables.
Modificar_Imagen()	Modifica los datos de la imagen.

**Tabla 26.-** Clase ad\_Media.

Método	Descripción
Existe_MediaImagen()	Determina si existe al menos una imagen de un proyecto determinado.
Existe_MediaSonido()	Determina si existe al menos un sonido de un proyecto determinado.
Existe_MediaLocucion()	Determina si existe al menos una locución de un proyecto determinado.
Existe_MediaAudio()	Determina si existe al menos un audio de un proyecto determinado.
Existe_MediaVideo()	Determina si existe al menos un video de un proyecto determinado.
Insertar_Imagen()	Inserta una imagen.
Modificar_Imagen()	Modifica los datos de la imagen.

**Tabla 27.-** Clase ad\_Imagen.

Método	Descripción
Existe_Imagen()	Determina si existe al menos una imagen de un proyecto determinado.
Insertar_Imagen()	Inserta una imagen.
Modificar_Imagen()	Modifica los datos de la imagen.

**Tabla 28.-** Clase ad\_SonidoLocucionAudio.

Método	Descripción
Existe_SonidoLocucion()	Determina si existe al menos un sonido, locución o audio de un proyecto determinado.
Insertar_SonidoLocucion()	Inserta un sonido, locución o audio.
Modificar_SonidoLocucion()	Modifica un sonido, locución o audio.

**Tabla 29.-** Clase ad\_Video.

<b>Método</b>	<b>Descripción</b>
Existe_Video()	Determina existe al menos un video de un proyecto determinado.
Insertar_Video()	Inserta los datos de un video.
Modificar_Video()	Modifica los datos de un video.

**Tabla 30.-** Clase ad\_ArchivoAuxiliar.

<b>Método</b>	<b>Descripción</b>
Existe_ArchivoAuxiliar()	Determina si existe el archivo auxiliar.
Insertar_ArchivoAuxiliar()	Inserta un archivo auxiliar.
Sobrescribir_ArchivoAux()	Sobrescribe un archivo auxiliar.
Eliminar_ArchivoAuxiliar()	Elimina un archivo auxiliar.

**Tabla 31.-** Clase ad\_ModelacionPedagogica.

<b>Método</b>	<b>Descripción</b>
Existe_ModelacionPedag()	Determina si existe la modelación pedagógica de un proyecto determinado.
Insertar_ModelacionPedag()	Inserta la modelación pedagógica.
Modificar_ModelacionPeda()	Modifica la modelación pedagógica.

**Tabla 32.-** Clase ad\_AlcanceProducto.

<b>Método</b>	<b>Descripción</b>
Existe_AlcanceProducto()	Determina si existe el alcance del producto de un proyecto determinado.
Insertar_AlcanceProducto()	Inserta el alcance del producto.
Modificar_AlcanceProducto()	Modifica el alcance del producto.

**Tabla 33.-** Clase ad\_DefinicionEntregable.

<b>Método</b>	<b>Descripción</b>
Existe_DefinicionEntregable()	Determina si existe la definición de entregables de un proyecto determinado.
Insertar_DefinicionEntregable()	Inserta la definición de entregables.
Modificar_DefinicionEntregable()	Modifica la definición de entregables.

**Tabla 34.-** Clase ad\_Entidad.

<b>Método</b>	<b>Descripción</b>
Existen_Entidades()	Determina si existe al menos una entidad de un proyecto determinado.
Existe_NombreEntidad()	Determina si el nombre de una entidad ya ha sido asignado a otra previamente.
Existe_IdEntidad()	Retorna una entidad dado su identificador.
Insertar_Entidad()	Inserta una entidad.
Modificar_Entidad()	Modifica una entidad.
Eliminar_Entidades(arreglo)	Elimina un arreglo de entidades.

**Tabla 35.-** Clase ad\_Contacto.

<b>Método</b>	<b>Descripción</b>
Existen_Contactos()	Determina si existe al menos un contacto de un proyecto determinado.
Existe_Contacto()	Determina si existe un contacto dado si carne de identidad.

Insertar_Contacto()	Inserta un contacto.
Modificar_Contacto()	Modifica un contacto.

### 3.2.1.3.- Clases *manager*

La clase *manager* es la que permite la creación de los objetos para pasarlos con los datos necesarios para realizar la consulta a la clase de acceso a datos. Existe una clase *manager* por cada CU, en caso de que éstos estén estrechamente relacionados se designa la misma clase para ambos. Tiene todos los métodos de las clases de acceso a datos asociadas a los CU.

**Tabla 36.-** CUs asociados a las clases *manager*.

Clase	CU asociado
m_Documento	Gestionar_Informacion, Definir_Documento.
m_Proyecto	Gestionar_Poyecto, Actualizar_Proyecto.
m_Usuario	Gestionar_Usuario, Gestionar_Autenticacion.
m_Entregable	Gestionar_Entregable, Generar_Reporte.

### 3.2.1.4.- Clase fachada

Existe una sola clase fachada, interfaz entre los paquetes Lógica de IU y *Manager*; contiene todos los métodos que permiten el acceso de la capa de presentación a los datos a través de la de aplicación.

### 3.2.1.5.- Clases lógica de interfaz de usuario

Se identifican por tener el prefijo sp (*Server Page*). Son páginas PHP puras que tienen la responsabilidad de permitir la captura y actualización de los datos de las interfaces de usuarios, para lo que declaran un variable de tipo Fachada y acceden a través de ella a los métodos necesarios en las capas inferiores. Se determinó una clase lógica por cada interfaz de usuario necesaria.

**Tabla 37.-** Descripción de las clases lógica de interfaz de usuario.

Clase	Descripción
sp_Inicio	Muestra documentos de interés para la gestión de la calidad en el proceso de desarrollo del SE, dando las opciones de verlos, descargarlos e imprimirlos en el CU Gestionar_Informacion y de sobrescribirlos en el CU Definir_Entregables. Además permite la autenticación de los usuarios con privilegios.
sp_Crear_Proyecto	Permite al vicedecano de producción la creación de proyectos y usuarios.
sp_Listar_Proyecto	Permite al vicedecano de producción el listado, el cambio de estado, la vista de la evolución y la eliminación de proyectos.
sp_Gestionar_Usuario	Permite al vicedecano de producción el listado, la modificación del perfil y la eliminación de usuarios.

sp_Datos_Proyecto	Permite subir los datos básicos de un proyecto a través de archivos, dando la opción de sobrescribirlos.
sp_EC	Muestra las opciones que puede realizar el Encargado de calidad.
sp_AC	Muestra las opciones que puede realizar el Asesor de calidad.
sp_LP	Muestra las opciones que puede realizar el Líder de proyecto.
sp_VP	Muestra las opciones que puede realizar el Vicedecano de producción.
sp_Alcance_Producto	Permite la inserción y modificación del alcance del producto, de la definición de entregables, de las entidades involucradas y sus respectivos contactos.
sp_ModelacionPedag	Permite la inserción y modificación de la modelación pedagógica.
sp_Gestion_Imagen	Permite la inserción, el listado, la modificación y la eliminación de las imágenes.
sp_Gestion_Sonido	Permite la inserción, el listado, la modificación y la eliminación de los sonidos.
sp_Gestion_Locucion	Permite la inserción, el listado, la modificación y la eliminación de las locuciones.
sp_Gestion_Audio	Permite la inserción, el listado, la modificación y la eliminación de los audios.
sp_Gestion_Video	Permite la inserción, el listado, la modificación y la eliminación de los videos.
sp_Parametros_Reporte	Permite conformar el reporte dando la opción de escoger los parámetros para ello y sus valores. Conformado finalmente el reporte en un PDF que permitirá visualizarlo o imprimirlo.
sp_Modificar_Pass	Permite modificar la contraseña del usuario logueado.
sp_Cerrar_Sesion	Permite cerrar la sesión del usuario logueado, enviándolo a la página de inicio.

### 3.2.1.6.- Clases interfaz de usuario

Las clases interfaz de usuario se identificarán por tener el prefijo cp (*Client Page*), ejemplo: cp\_Insertar\_Proyecto y están asociadas una por página servidora. Los atributos que contendrán serán los que permitirán la introducción o muestra de peticiones de datos al sistema, que se harán a través de formularios que las componen. Las funciones que pueden contener serán validaciones de campos en *JavaScript* y código *Smarty* embebido para mostrar los datos. Todas las clases contienen un método en *JavaScript* *Accion(x)* que permite determinar la validez de los datos introducidos por el usuario y el envío del formulario con la acción realizada.

### 3.2.1.7.- Otras clases

**Tabla 38.-** Descripción de otras clases utilizadas.

Clase	Descripción
adodb	Permite establecer la conexión con la BD.

Configuracion	Permite establecer los datos que darán acceso a la BD.
Estilo.css	Contiene los diferentes de estilos utilizados en la propuesta.

En los epígrafes siguientes se describirán: (1) los diagramas de colaboración de los principales escenarios que conforman los CUs, que describen las interacciones entre los objetos en un formato de grafo o red, (2) los diagramas de clases de diseño Web por CUs, que muestran las clases y sus interrelaciones, dando una idea de los elementos participantes en la realización de cada CU, (3) el diagrama de clases persistentes que muestra las clases que contienen información de dicho tipo y las relaciones entre ellas y finalmente (4) el modelo de datos que expresa las relaciones de las clases persistentes de manera refinada a través de las tablas de la BD de la propuesta.

3.2.2.- Diagramas de colaboración por escenarios

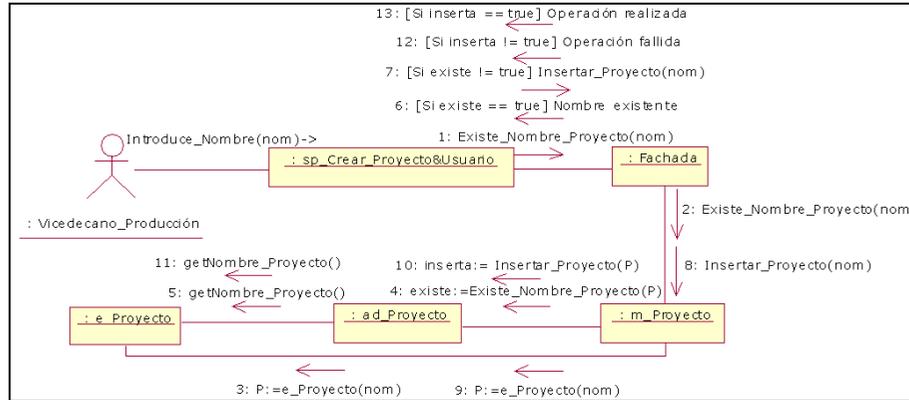


Figura 15.- Diagrama de colaboración: Crear proyecto.

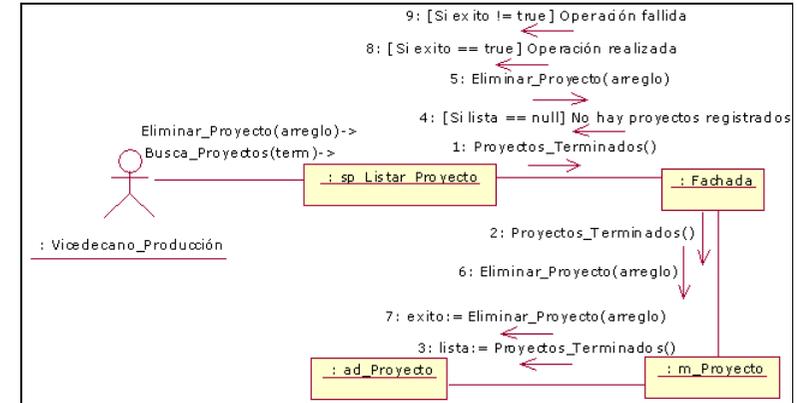


Figura 17.- Diagrama de colaboración: Eliminar proyectos.

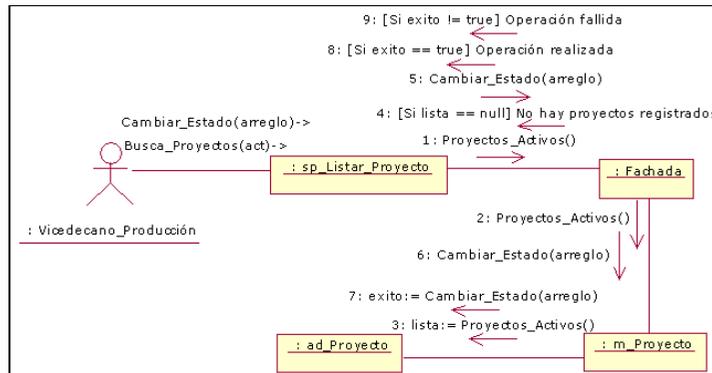


Figura 16.- Diagrama de colaboración: Cambiar estado de proyecto.

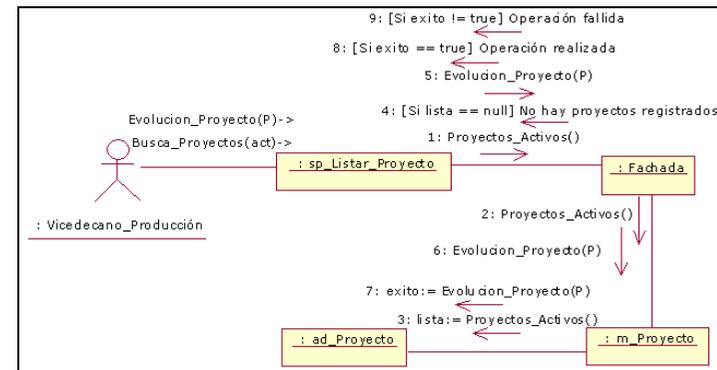


Figura 18.- Diagrama de colaboración: Evolución de proyecto.

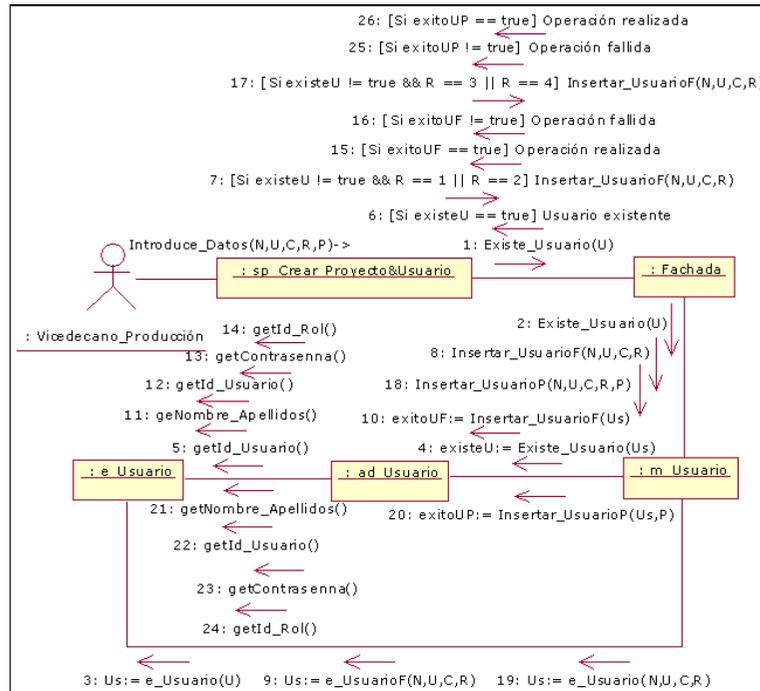


Figura 19.- Diagrama de colaboración: Crear usuario.

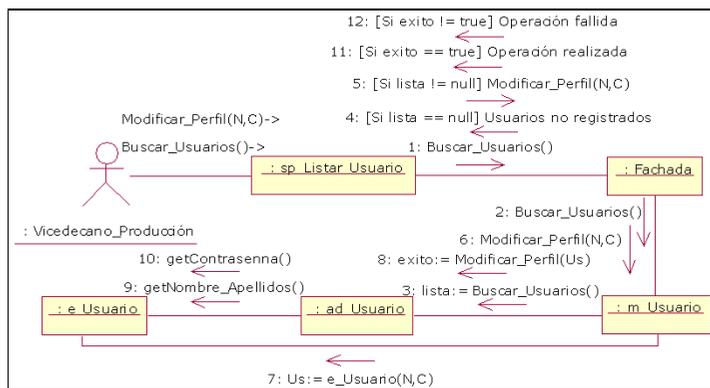


Figura 20.- Diagrama de colaboración: Modificar perfil de usuario.

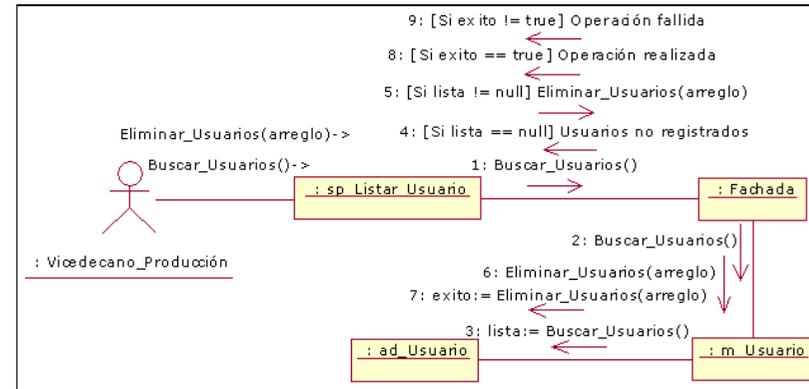


Figura 21.- Diagrama de colaboración: Eliminar usuarios.

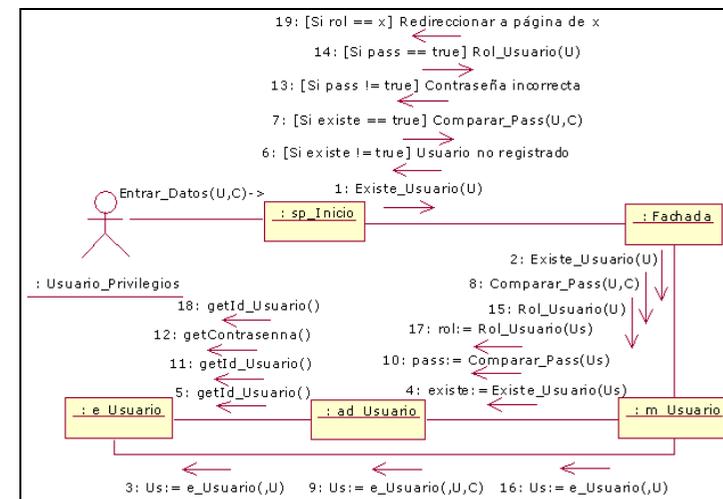


Figura 22.- Diagrama de colaboración: Autenticar usuario.

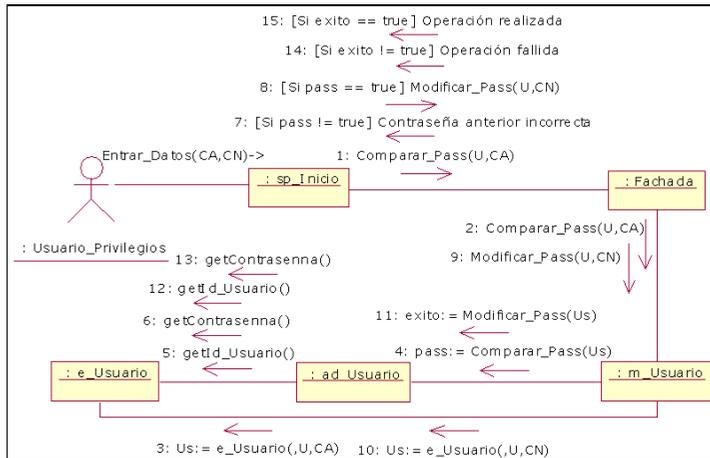


Figura 23.- Diagrama de colaboración: Modificar contraseña.

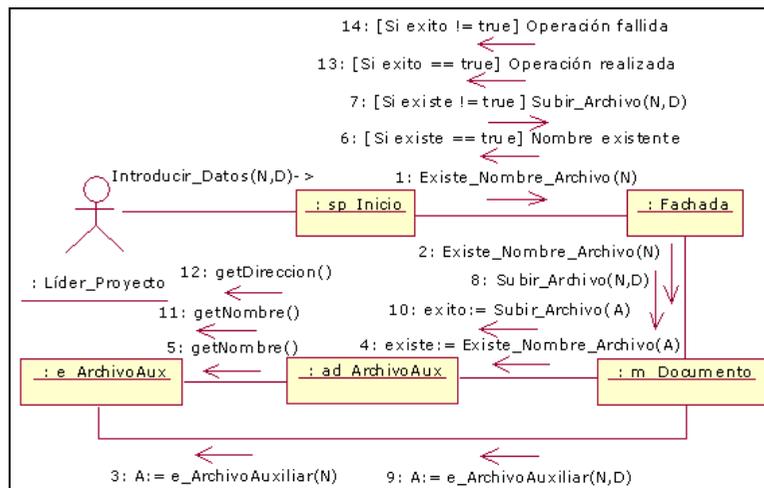


Figura 24.- Diagrama de colaboración: Insertar nuevo archivo.

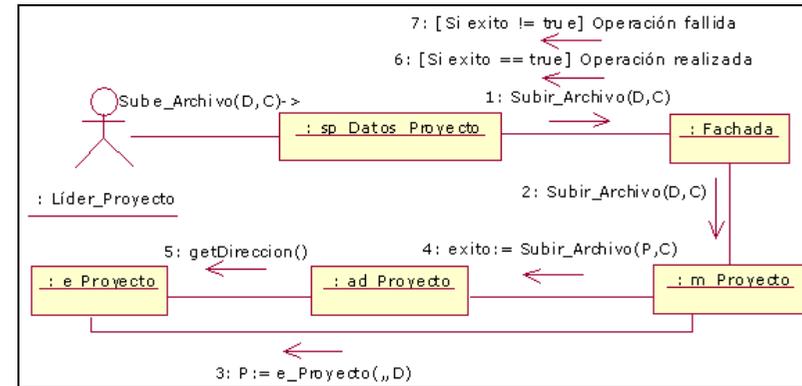


Figura 25.- Diagrama de colaboración: Subir archivo.

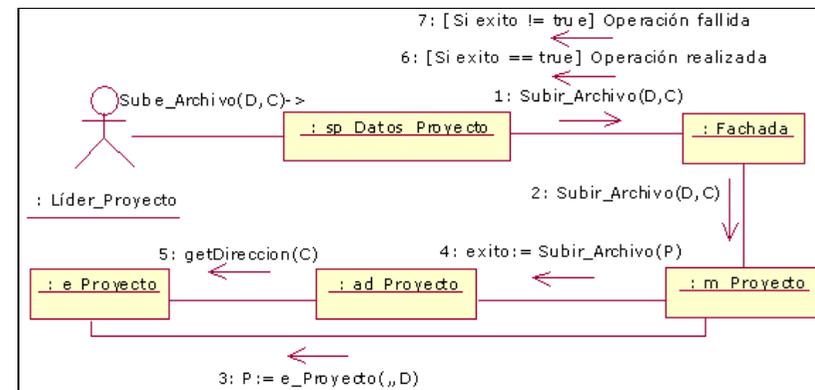


Figura 26.- Diagrama de colaboración: Sobrescribir archivo.

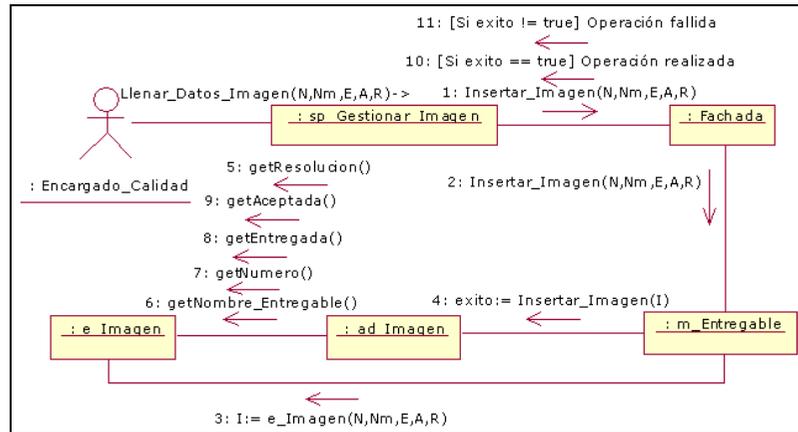


Figura 27.- Diagrama de colaboración: Llenar entregable.

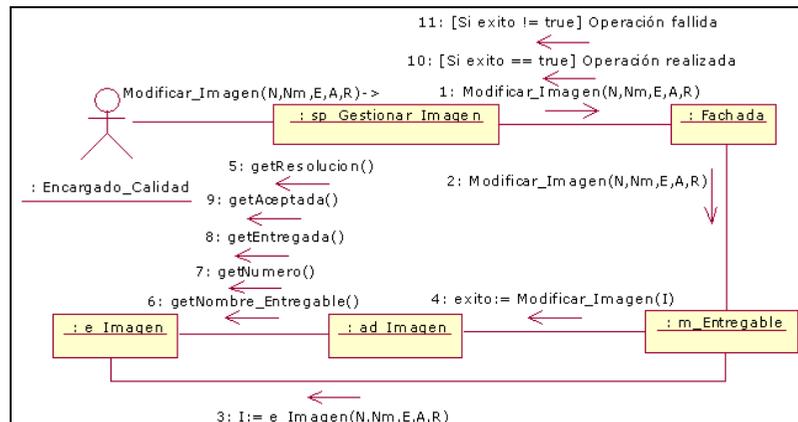


Figura 28.- Diagrama de colaboración: Modificar entregable.

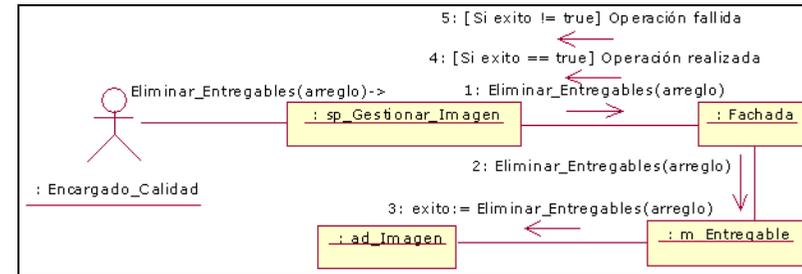


Figura 29.- Diagrama de colaboración: Eliminar entregables.

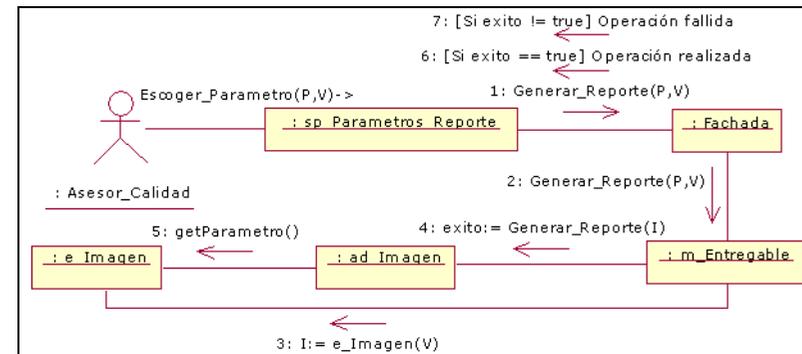
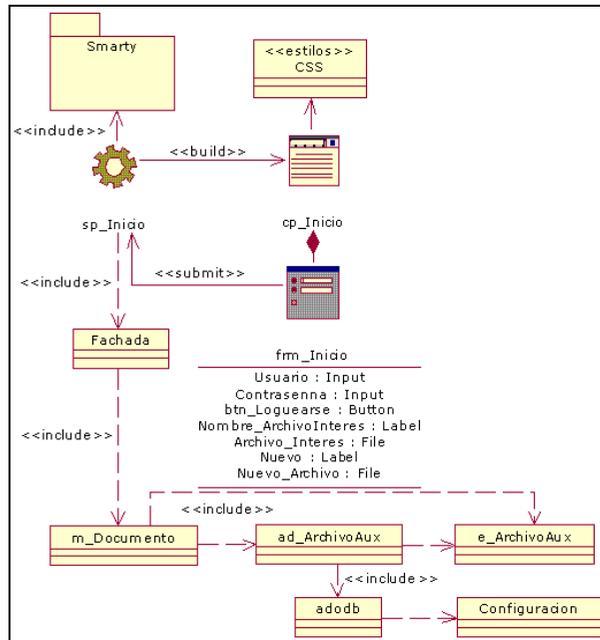


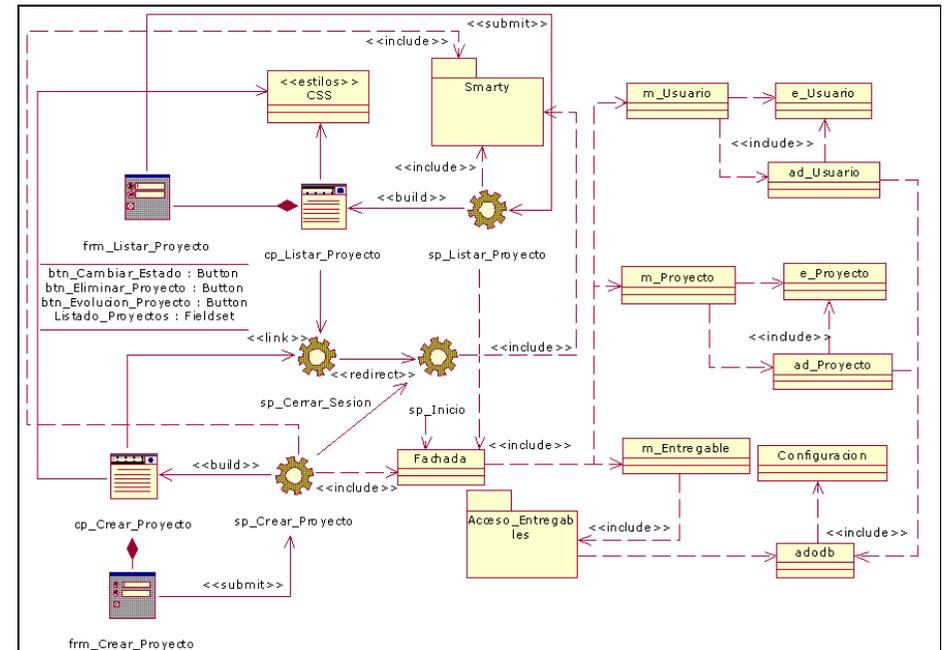
Figura 30.- Diagrama de colaboración: Conformar reporte.

### 3.2.3.- Diagramas de clases Web por Casos de Uso



**Figura 31.-** Diagrama de clases Gestionar información.

Los atributos Nombre\_ArchivoInteres y Archivo\_Interes son una generalización de todos los posibles archivos que el Asesor de Calidad defina como importantes para la gestión de la calidad del SE durante su proceso de desarrollo. En este CU los campos de tipo *File* están ocultos porque sólo el Asesor puede sobrescribir los existentes o añadir uno nuevo.



**Figura 32.-** Diagrama de clases Gestionar proyecto.

Datos\_Usuario contiene los datos de un usuario, ver figura 33. btn\_Atras es la generalización de dichos botones existentes en las pantallas evolución y crear usuario. El paquete Acceso\_Entregables, figura 39, contiene todas las clases entidades de los entregables y sus respectivos accesos a datos. m\_Entregable se relaciona con todas la clases del paquete, las que se relacionan con adodb son las ad. La redirección de Cerrar\_Sesion a Inicio se refiere a mostrar la página de inicio, sólo se pone la servidora para indicarlo, ver figura 31.

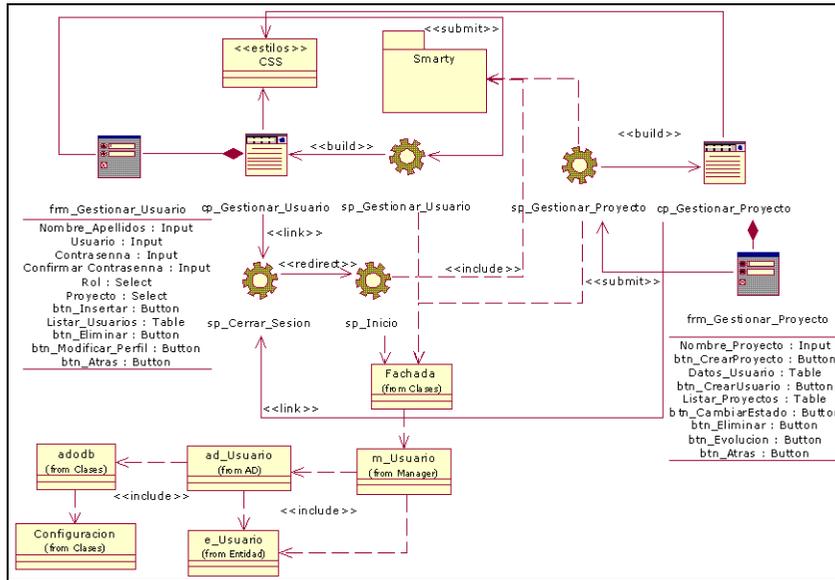


Figura 33.- Diagrama de clases Gestionar usuario.

No se ponen las páginas clientes asociadas a las servidoras de EC, AC, LP y VP y los formularios que las componen por legibilidad, pero se detalla a continuación lo que se puede hacer en cada una:

- EC: el Encargado de calidad puede gestionar los entregables de su proyecto.
- AC: el Asesor de calidad puede generar reportes, definir documentos y buscar proyectos.
- VP: el Vicedecano de producción puede gestionar proyectos y usuarios.
- LP: el Líder de proyecto puede actualizar su proyecto y ver su evolución.

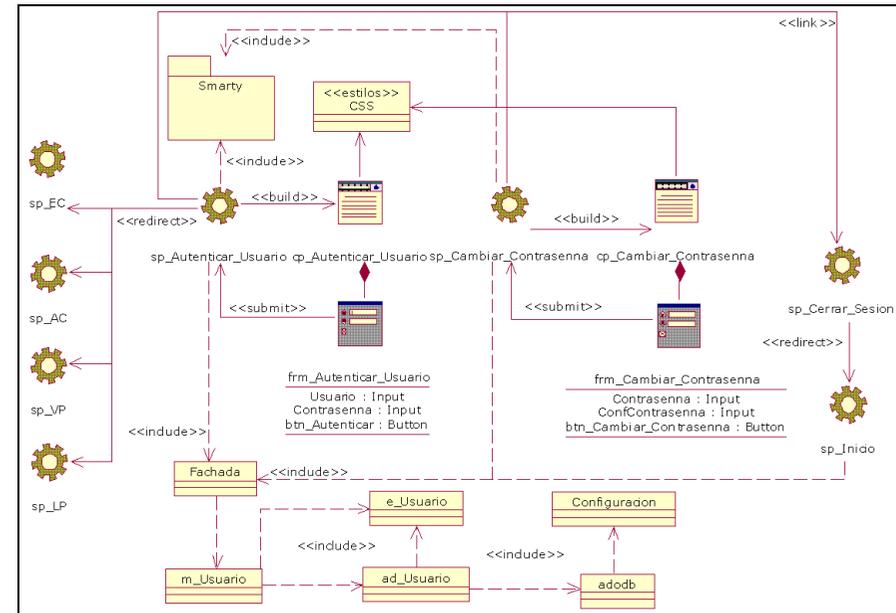
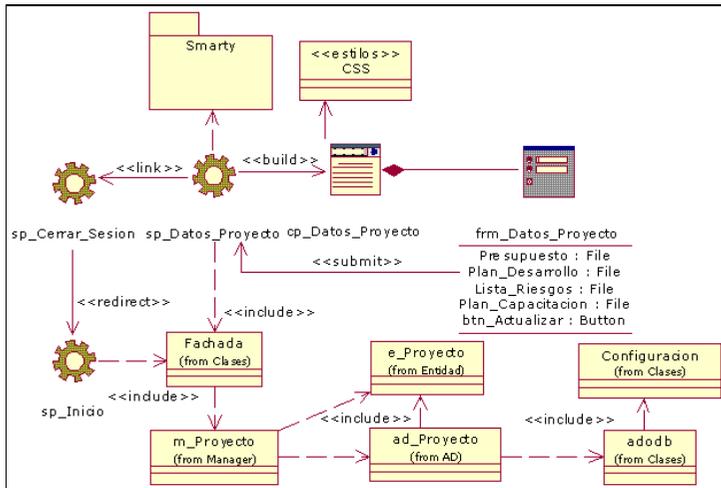
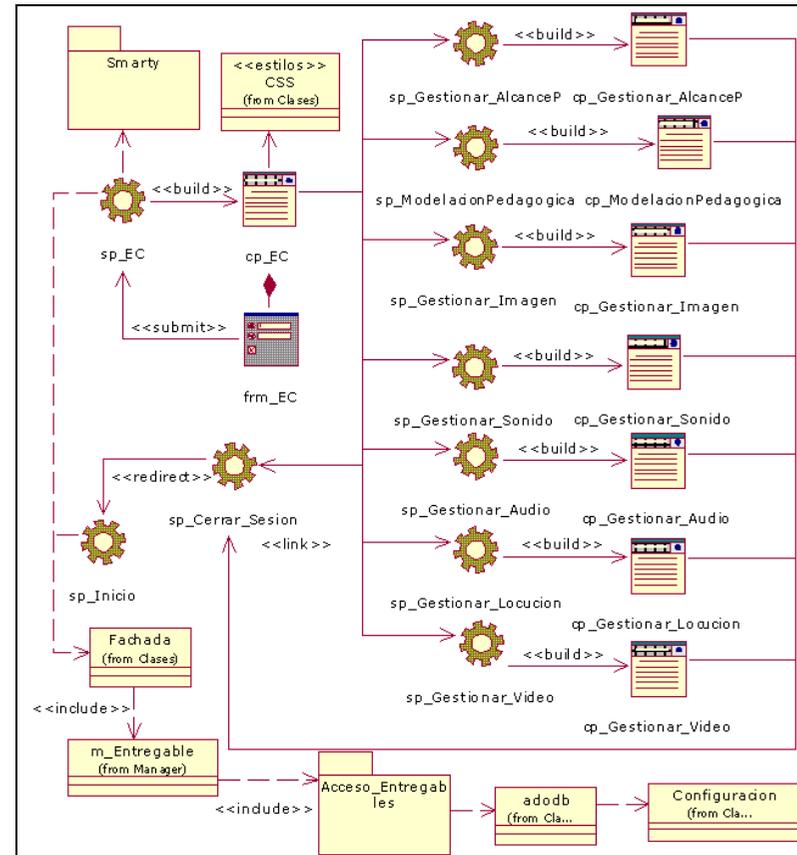


Figura 34.- Diagrama de clases Gestionar autenticación.



**Figura 35.-** Diagrama de clases Web del CU Actualizar proyecto.

En la figura 36 por cuestiones de legibilidad no se pudieron relacionar todas las páginas servidoras con la fachada ni con el paquete *Smarty*, así como las páginas clientes con la hoja de estilos.



**Figura 36.-** Diagrama de clases Web del CU Gestionar entregable.

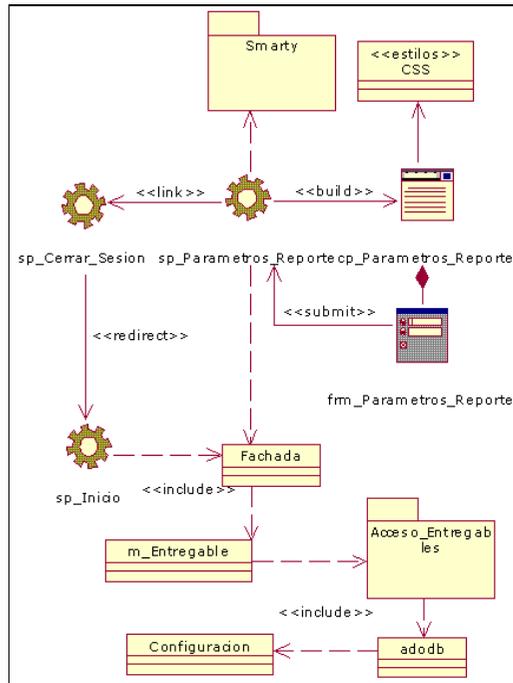


Figura 37.- Diagrama de clases Web del CU Generar reporte.

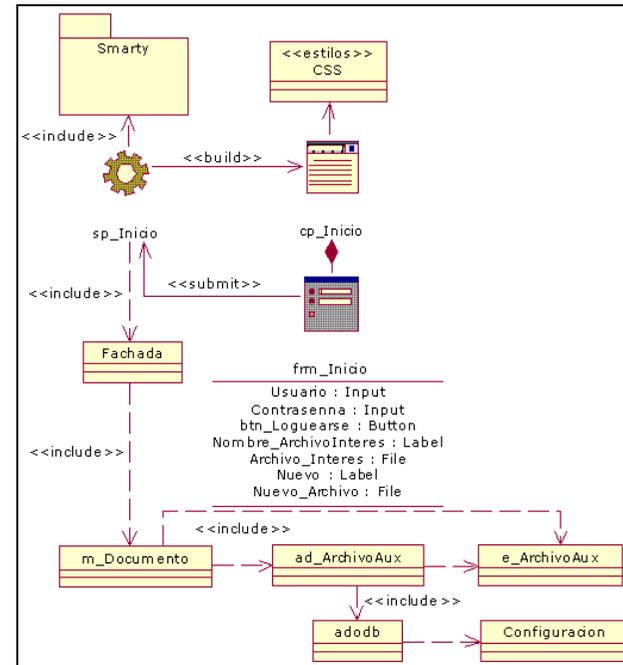


Figura 38.- Diagrama de clases Web del CU Definir documento.

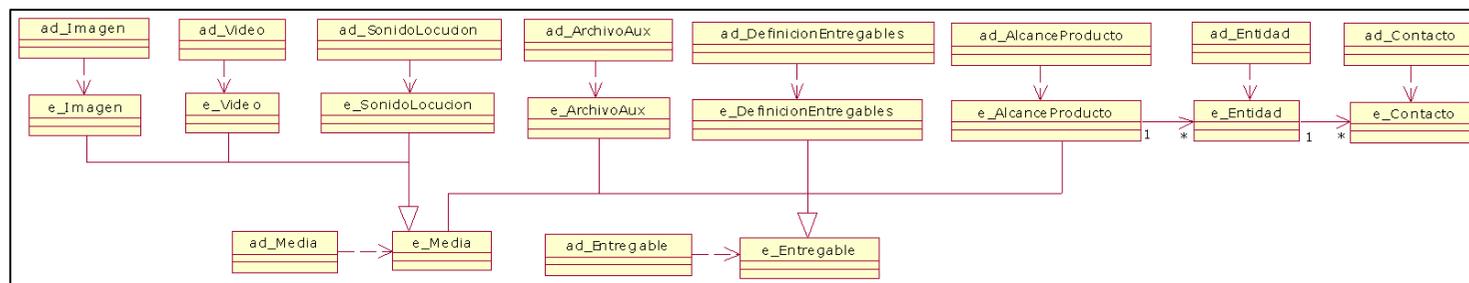


Figura 39.- Paquete de acceso a entregables.

3.2.4.- Diseño de la base de datos

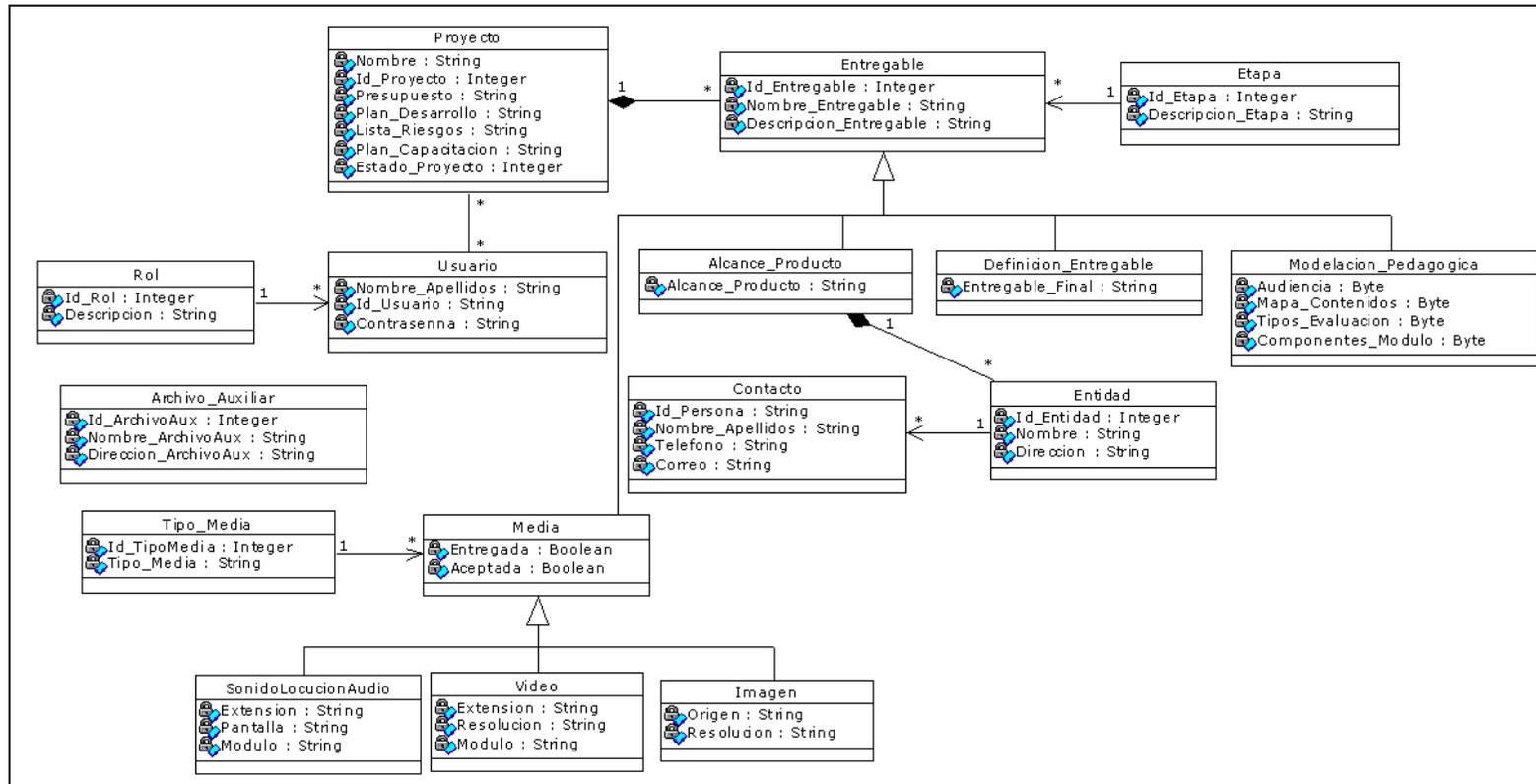


Figura 40.- Diagrama de clases persistentes.

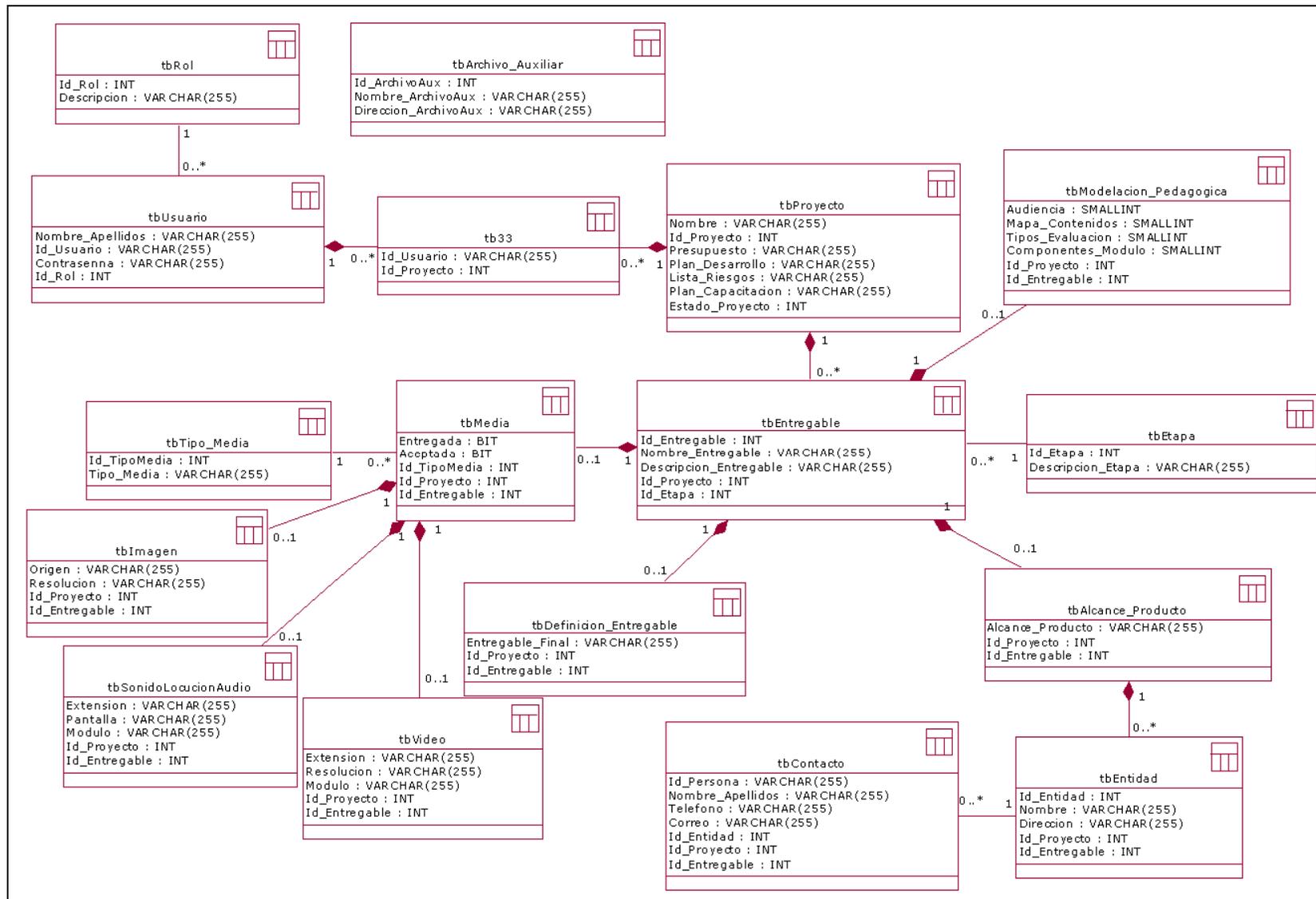


Figura 41.- Modelo de datos.

### 3.2.4.1.- Descripción de las tablas de la base de datos

**Tabla 39.-** Descripción de la tabla tbRol.

<b>Descripción</b>	Nomenclador que almacena los roles que desempeñan los usuarios del sistema.
<b>Atributo</b>	<b>Descripción</b>
Id_Rol	Llave primaria, identificador del rol, auto-numérico.
Descripcion	Nombre del rol, ejemplo: Líder de proyecto.

**Tabla 40.-** Descripción de la tabla tbUsuario.

<b>Descripción</b>	Almacena los datos del usuario.
<b>Atributo</b>	<b>Descripción</b>
Nombre_Apellidos	Nombre y apellidos del usuario.
Id_Usuario	Llave primaria, identificador del usuario, el propio usuario.
Contraseña	Contraseña del usuario.
Id_Rol	Llave foránea, identificador del rol que tiene el usuario.

**Tabla 41.-** Descripción de la tabla tbResponsabilidad\_Proyecto.

<b>Descripción</b>	Almacena los identificadores de los usuarios y los proyectos, almacenando los usuarios que tienen responsabilidades en cada proyecto.
<b>Atributo</b>	<b>Descripción</b>
Id_Usuario	Identificador del usuario, llave foránea-primaria.
Id_Proyecto	Identificador del proyecto, llave foránea-primaria.

**Tabla 42.-** Descripción de la tabla tbProyecto.

<b>Descripción</b>	Almacena los datos de un proyecto.
<b>Atributo</b>	<b>Descripción</b>
Nombre_Proyecto	Nombre del proyecto.
Id_Proyecto	Llave primaria, identificador del proyecto, auto-numérico.
Presupuesto	Dirección del archivo con el presupuesto destinado al proyecto.
Plan_Desarrollo	Dirección del archivo con el plan de desarrollo del proyecto.
Lista_Riesgos	Dirección del archivo con la lista de riesgos del proyecto.
Plan_Capacitacion	Dirección del archivo con el plan de capacitación del proyecto.
Estado_Proyecto	Estado del proyecto: activo o terminado.

**Tabla 43.-** Descripción de la tabla tbEntregable.

<b>Descripción</b>	Almacena los datos de un entregable.
<b>Atributo</b>	<b>Descripción</b>
Id_Entregable	Llave primaria, identificador del entregable, auto-numérico.
Nombre_Entregable	Nombre del entregable.
Descripcion	Breve descripción de en qué consiste el entregable.
Id_Proyecto	Llave primaria-foránea, identificador del proyecto al que pertenece el entregable.
Id_Etapa	Llave foránea, identificador de la etapa.

**Tabla 44.-** Descripción de la tabla tbEtapa.

<b>Descripción</b>	Nomenclador que almacena las etapas por las que atraviesa el proyecto durante el proceso de desarrollo del software.
<b>Atributo</b>	<b>Descripción</b>
Id_Etapa	Llave primaria, identificador de la etapa, auto-numérico.

Descripcion_Etapa	Nombre de la etapa, ejemplo: Gestión Legal.
-------------------	---

**Tabla 45.-** Descripción de la tabla tbMedia.

<b>Descripción</b>	Almacena los datos de una media.
<b>Atributo</b>	<b>Descripción</b>
Entregada	Estado de la media: entregada o no.
Aceptada	Estado de la media: aceptada o no.
Id_TipoMedia	Llave foránea, identificador del tipo de media.
Id_Proyecto	Llave primaria-foránea, identificador del proyecto al que pertenece la media.
Id_Entregable	Llave primaria-foránea, identificador del entregable al que pertenece la media.

**Tabla 46.-** Descripción de la tabla tbTipo\_Media.

<b>Descripción</b>	Nomenclador que almacena el tipo de media.
<b>Atributo</b>	<b>Descripción</b>
Id_TipoMedia	Llave primaria, identificador del tipo de media, auto-numérico.
Tipo_Media	Tipo de media: imagen, sonido_locución, video, animación.

**Tabla 47.-** Descripción de la tabla tblImagen.

<b>Descripción</b>	Almacena los datos de una imagen.
<b>Atributo</b>	<b>Descripción</b>
Origen	Origen de la imagen.
Resolucion	Resolución de la imagen.
Id_Proyecto	Llave primaria-foránea, identificador del proyecto al que pertenece la imagen.
Id_Entregable	Llave primaria-foránea, identificador del entregable al que pertenece la imagen.

**Tabla 48.-** Descripción de la tabla tbSonidoLocucionAudio.

<b>Descripción</b>	Almacena los datos de los sonidos y las locuciones.
<b>Atributo</b>	<b>Descripción</b>
Extension	Extensión de los sonidos y locuciones.
Pantalla	Pantalla en la que se encuentra el/los sonidos y locuciones.
Modulo	Módulo en que se encuentra el/los sonidos y locuciones.
Id_Proyecto	Llave primaria-foránea, identificador del proyecto al que pertenecen los sonidos y locuciones.
Id_Entregable	Llave primaria-foránea, identificador del entregable al que pertenecen los sonidos y locuciones.

**Tabla 49.-** Descripción de la tabla tbVideo.

<b>Descripción</b>	Almacena los datos de un video.
<b>Atributo</b>	<b>Descripción</b>
Extension	Extensión del video.
Resolucion	Resolución del video.
Modulo	Módulo en que se encuentra el video.
Id_Proyecto	Llave primaria-foránea, identificador del proyecto al que pertenece el video.
Id_Entregable	Llave primaria-foránea, identificador del entregable al que pertenece el video.

**Tabla 50.-** Descripción de la tabla tbAlcance\_Producto.

<b>Descripción</b>	Almacena los datos del alcance del producto.
<b>Atributo</b>	<b>Descripción</b>
Alcance_Producto	Breve descripción de lo que se quiere lograr con el producto.
Id_Proyecto	Llave primaria-foránea, identificador del proyecto al que pertenece el alcance del producto.
Id_Entregable	Llave primaria-foránea, identificador del entregable al que pertenece el alcance del producto.

**Tabla 51.-** Descripción de la tabla tbEntidad.

<b>Descripción</b>	Almacena los datos de las entidades asociadas a los proyectos.
<b>Atributo</b>	<b>Descripción</b>
Id_Entidad	Llave primaria, identificador de la entidad asociada a un proyecto.
Id_Proyecto	Llave primaria-foránea, identificador del proyecto.
Id_Entregable	Llave primaria-foránea, identificador del entregable.
Nombre	Nombre de la entidad.
Direccion	Dirección de la entidad.

**Tabla 52.-** Descripción de la tabla tbDefinicion\_Entregable.

<b>Descripción</b>	Almacena la definición de los entregables que desea una entidad con la culminación del proyecto.
<b>Atributo</b>	<b>Descripción</b>
Entregable_Final	Listado de entregables que desea la entidad.
Id_Entidad	Llave foránea, identificador de la entidad que define los entregables finales.
Id_Proyecto	Llave primaria-foránea, identificador del proyecto al que pertenece la definición de los entregables.
Id_Entregable	Llave primaria-foránea, identificador del entregable al que pertenece la definición de los entregables.

**Tabla 53.-** Descripción de la tabla tbContacto.

<b>Descripción</b>	Almacena los datos de los contactos de una entidad.
<b>Atributo</b>	<b>Descripción</b>
Id_Persona	Llave primaria, identificador del contacto, Cl.
Nombre_Apellidos	Nombre y apellidos del contacto.
Telefono	Teléfono del contacto.
Correo_Electronico	Correo electrónico del contacto.
Id_Entidad	Llave foránea, identificador de la entidad a la que pertenece el contacto.

**Tabla 54.-** Descripción de la tabla tbArchivo\_Auxiliar.

<b>Descripción</b>	Almacena los datos de un archivo auxiliar.
<b>Atributo</b>	<b>Descripción</b>
Id_ArchivoAux	Llave primaria, identificador del archivo auxiliar.
Nombre_ArchivoAux	Nombre del archivo auxiliar.
Direccion_ArchivoAux	Dirección en el servidor del archivo auxiliar.

### 3.2.5.- Principios de diseño

El diseño del sistema debe centrarse en el usuario, por lo que utiliza ciertos principios generales que garantizan la usabilidad en los diseños para aplicaciones Web.

- Principio de uso equiparable: las características de privacidad, garantía y seguridad estarán disponibles para todos los usuarios, y el diseño gráfico será atractivo para todos.
- Principio de la flexibilidad: se ofrecerán posibilidades de elección en los métodos de uso que faciliten al usuario la exactitud y precisión y se adapte su ritmo.
- Principio de la información perceptible: se usará para presentar la información esencial el modo verbal. Se proporcionará contraste entre la información esencial y sus alrededores ampliando la legibilidad de la información esencial.
- Principio de tolerancia al error: se dispondrá de mecanismos para minimizar riesgos y errores utilizando elementos comunes. Se posibilita la reversibilidad y la recuperabilidad de las acciones, por ejemplo se debe tener el consentimiento del usuario para realizar acciones irreversibles como la modificación y/o eliminación de datos.
- Principio de esfuerzo de acceso y uso: se minimizarán las acciones repetitivas, proporcionando una línea de visión clara hacia los elementos importantes.

### **3.2.6.- Interfaz de la aplicación**

La página principal de la aplicación muestra la información referente a la gestión de la calidad en el proceso de desarrollo. Además se da la posibilidad, bien visible, de loguearse para acceder al resto de los módulos. Se utilizan los colores verde, en tonos claros y fuertes, y el blanco.

Para el diseño de los menús se utilizó la información escrita en la parte izquierda del sistema. Todas las páginas tienen enlace a la página principal.

Se utilizó una hoja de estilos para guardar la configuración del diseño para todas las páginas y cada uno de sus elementos. Como tipo de letra la Verdana de tamaño 10 y 12, que permite una separación entre las letras óptima para presentar información.

Los formularios de entradas ocupan el centro superior, encerrados en un *fieldset* para, diferenciarlos del resto de la página. Se incluye una breve explicación del objetivo del formulario y alguna especificación con respecto a las entradas. Los formularios con muchas entradas se organizan y dividen de acuerdo a lo que representan.

En general se realizan múltiples operaciones en cada página de forma que el usuario no tenga que moverse tanto dentro del sistema para llevar a cabo un flujo de trabajo completo.

### **3.2.7.- Tratamiento de errores**

El tratamiento de errores garantiza el buen funcionamiento del sistema facilitando su uso. La propuesta se modeló de manera que las operaciones de inserción, eliminación y modificación de registros en la BD se realizaran de forma correcta; para lo que fueron establecidos mecanismos de visualización de la información para evitar, en gran medida, la introducción manual de datos y de esta forma minimizar los errores, siendo válida toda la información entrada.

Para la inserción se implementarán funciones *JavaScript* que validen la entrada de datos y en caso de haber errores se mostrarán mensajes especificándolos.

Los errores que pueden ser generados por el SGBD serán capturados antes de mostrarse al usuario, dándosele tratamiento en el sistema para mostrárselos de manera que sea entendible para él.

### **3.2.8.- Seguridad**

La seguridad de los datos en el sistema está sustentada sobre la base del establecimiento de niveles de permisos en los módulos que lo componen.

La página de inicio del sistema puede ser accedida por todos los usuarios que lo deseen pues contiene información sobre la gestión de la calidad en el proceso de desarrollo del SE. Para acceder al resto de las páginas se requiere de la autenticación de los usuarios, lo que se ha logrado con la creación de un módulo de administración en el que se definen los usuarios con acceso: vicedecano de producción, asesor de calidad, líderes y encargados de calidad de los proyectos. Dichos usuarios inclusive dentro del sistema sólo podrán acceder a los recursos necesarios, por ejemplo el encargado de calidad no puede ver los datos básicos gestionados por el líder de proyecto en su proyecto.

Otro de los mecanismos implementados es el uso del algoritmo MD5 para encriptar las contraseñas de los usuarios que se almacenan en la BD como una cadena de treinta y dos caracteres, evitando la legibilidad y garantizando la privacidad. Con este algoritmo no es posible obtener dos mensajes de salida iguales con diferentes mensajes de entrada, ni

obtener un mensaje a partir de uno de treinta y dos caracteres. En esencia, el algoritmo es una manera de verificar la integridad de los datos, se dice que es un método confiable.

Se propone mantener salvadas de la BD en otros discos duros, tanto del nodo servidor como de máquinas clientes. Dichas salvadas pueden realizarse diaria o semanalmente, según se defina posteriormente. Producto de que es innegable la posibilidad de que se pierda la información por problemas de orden tecnológico como la contaminación por virus, los fallos en dispositivos del sistema o la durabilidad de los soportes electrónicos se propone que el servidor se destine solamente a eso, evitando el uso de otros servicios que incrementan en gran medida las posibilidades.

### **Conclusiones**

En este capítulo se mostraron varios diagramas para llevar a cabo el proceso de diseño del sistema, para lo que se utilizaron los de clases Web que explican la lógica del sistema, y se diseñó el diagrama de clases persistentes que permitió hacer el modelo de datos para el diseño de la BD que se utilizará.

# Capítulo 4

## Implementación y prueba

### Introducción

En este capítulo se definirán los modelos de despliegue e implementación, así como las pruebas de caja negra (que se le hicieron al sistema durante su desarrollo), etapa final de la fase de construcción del proceso de desarrollo.

### 4.1.- Implementación

El modelo de despliegue detalla la configuración de los diferentes dispositivos hardware que participan en la ejecución del sistema. Como muestra la figura 42, el sistema se distribuirá en un Servidor que contendrá los servicios Web y de datos debido a que la base de datos no es grande ni existirá tampoco

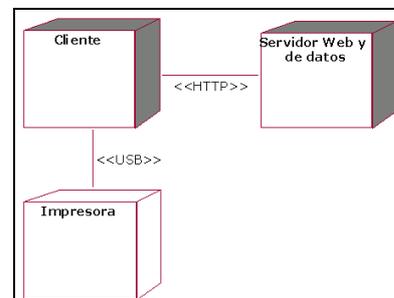


Figura 42.- Diagrama de despliegue.

gran flujo de información. El Cliente solicita los servicios al Servidor vía HTTP y tendrá conectada, no necesariamente, una impresora, en el caso de que necesite imprimir algún recurso.

El modelo de implementación describe los componentes y las relaciones de dependencias de compilación entre ellos.

La Figura 43 muestra la relación del sistema como un componente y la BD.

En la Figura 44 se detallan las relaciones entre paquetes, determinados por módulos; los que engloban: administración (CUs Gestionar proyecto, usuario y autenticación), facultad&proyecto (CUs Gestionar información, Generar reporte, Definir documento, Actualizar proyecto y Gestionar entregable). Describe las dependencias entre los paquetes desde las terminales clientes hasta los recursos de la BD.

La figura 45 muestra las relaciones de dependencia entre los componentes del paquete Administración. En la 46 no se muestran las relaciones entre los componentes del paquete Facultad&Proyecto porque no existen, las relaciones de dependencia de los paquetes de este módulo se establecen con los de otros paquetes, lo que se visualiza mejor en el diagrama en terminales clientes, ver figura 47.

La figura 47 especifica los componentes que integran cada uno de los paquetes que se ejecutan de alguna manera en el cliente, los que se conectan con la BD (componente que contiene genéricamente todos los que intervienen en la conexión y realización de consultas a la BD). Los restantes paquetes contienen:

- *Manager*: la fachada, que depende de los cuatro *manager* (ver tabla 36) también contenidos en el paquete.
- Entidades: las clases entidades, ver subepígrafe 3.2.1.1.
- Acceso\_Datos: las clases de acceso a datos, ver tablas desde la 23 hasta la 35.
- Ficheros\_Conexión: las clases adobd, Configuración.

Los componentes que se muestran contienen cada uno las clases servidoras y clientes de la realización de un escenario dentro de un CU, por ejemplo: el componente Listar\_Proyecto.php contiene las clases sp\_Listar\_Proyecto y cp\_Listar\_Proyecto.

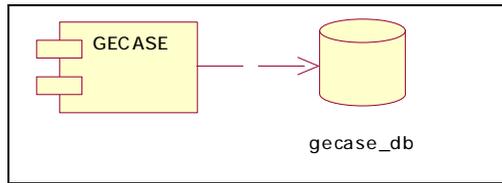


Figura 43.- Diagrama de componentes de base de datos.

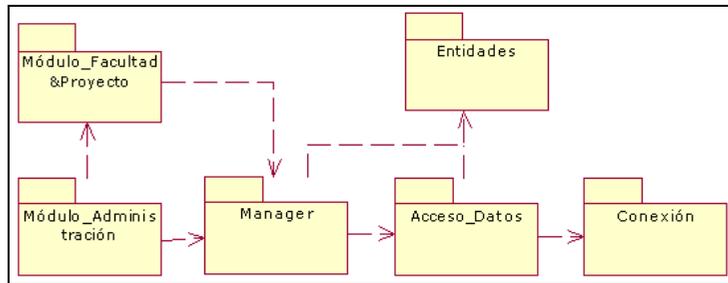


Figura 44.- Diagrama de componentes del sistema.

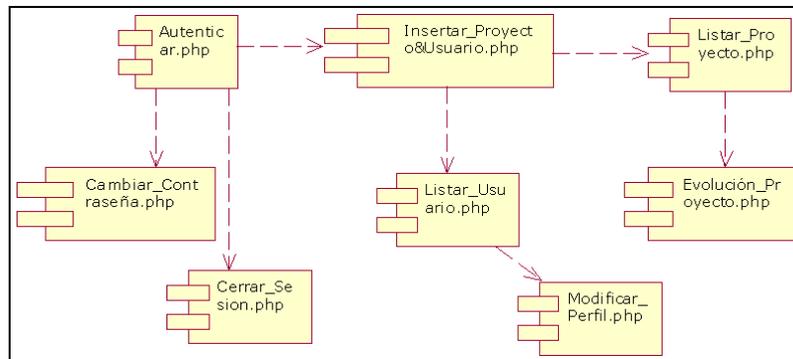


Figura 45.- Diagrama de componentes del módulo Administración.

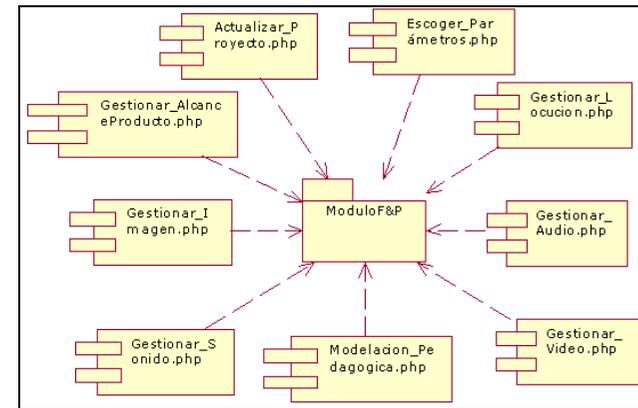


Figura 46.- Diagrama de componentes del Módulo Facultad & Proyecto.

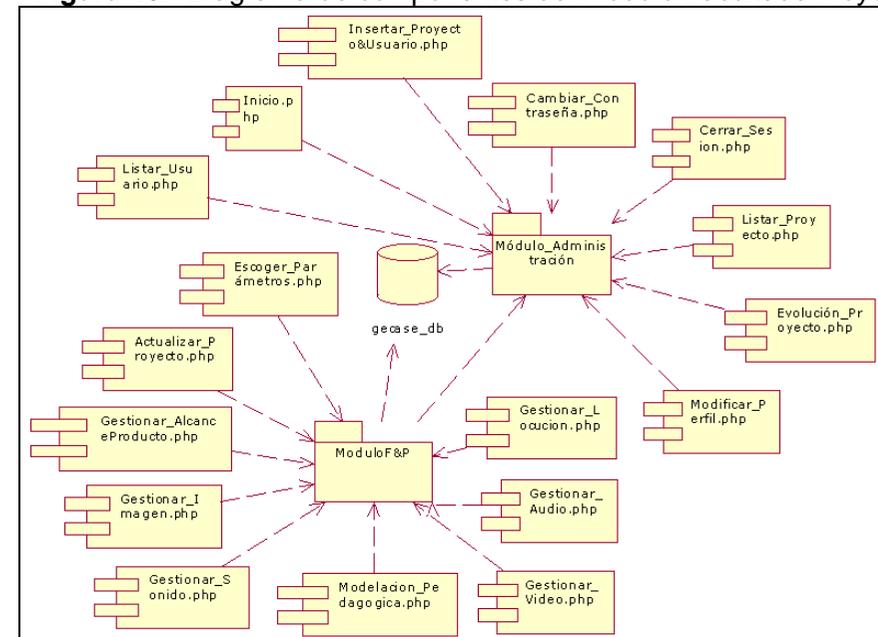


Figura 47.- Diagrama de componentes de terminales clientes.

## 4.2.- Prueba

Los casos de prueba (CP) especifican una forma de probar el sistema, incluyendo la entrada o resultado con el que se ha de probar y las condiciones bajo las que ha de probarse. Las pruebas de caja negra son pruebas de comportamiento observable externamente del sistema que (1) verifican las especificaciones funcionales y no consideran la estructura interna del programa y (2) son hechas sin el conocimiento interno del producto.

Los CP se realizarán por CU, se harán tres tipos fundamentales:

- Creación de elementos: se probarán a la hora de introducir los datos correcta e incorrectamente.
- Modificación de elementos: se probarán a la hora de introducir los datos correcta e incorrectamente.
- Realización de acciones sobre elementos listados: se probarán a la hora de seleccionar elementos en correspondencia con la acción a realizar.

### 4.2.1.- CU Gestionar información.

Este CU no tiene CP ya que lo que permite es descargar archivos existentes en la BD, no necesita introducción previa de datos por parte del usuario, por lo que no cabe la posibilidad de cometer errores.

### 4.2.2.- CU Gestionar proyecto.

**Tabla 55.-** Condiciones de entrada para Gestionar proyecto.

Condición de entrada	Casos válidos	Casos no válidos
Nombre de proyecto	Letras, números y caracteres extraños.	Campo vacío.
Estado del proyecto	Activos, terminados.	----
Elementos seleccionados	Marcar al menos un elemento para realizar acción.	No marcar ningún elemento.

**Tabla 56.-** Caso de prueba: Crear proyecto entrando datos correctamente.

<b>Entrada</b>	El Vicedecano de producción introduce el nombre del proyecto a crear: - Nombre: GPI.
<b>Resultado</b>	Se inserta en la BD el nombre del proyecto.
<b>Condiciones</b>	-

**Tabla 57.-** Caso de prueba: Crear proyecto entrando datos incorrectamente.

<b>Entrada</b>	El Vicedecano de producción no introduce ningún nombre en el campo: - Nombre: __
----------------	---

<b>Resultado</b>	Muestra un mensaje de error especificando que el nombre es un campo obligatorio.
<b>Condiciones</b>	-

**Tabla 58.-** Caso de prueba: Buscar proyectos sin seleccionar un criterio.

<b>Entrada</b>	El Vicedecano de producción no escoge ningún criterio de búsqueda: - Nombre: __ - Estado: ----
<b>Resultado</b>	Muestra un mensaje de error especificando que debe seleccionar un criterio de búsqueda.
<b>Condiciones</b>	-

**Tabla 59.-** Caso de prueba: Buscar proyectos seleccionando el criterio de nombre/estado/ambos.

<b>Entrada</b>	El Vicedecano de producción escoge el criterio de búsqueda nombre/estado/ambos: - Nombre: GPI/__/GPI - Estado: ----/Activo/Activo
<b>Resultado</b>	Muestra los datos del proyecto que coincida(n) con los valores de los criterios escogidos.
<b>Condiciones</b>	Proyecto registrado previamente con las características descritas.

**Tabla 60.-** Caso de prueba: Cambiar estado de proyecto(s) seleccionado(s).

<b>Entrada</b>	El Vicedecano de producción marca al menos un elemento y presiona el botón Cambiar estado.
<b>Resultado</b>	Muestra un mensaje preguntando si está seguro de realizar la acción, en caso aprobatorio cambia el estado del/los elemento(s) seleccionado(s) de activo(s) a terminado(s).
<b>Condiciones</b>	Proyecto(s) activo(s) registrado(s) en la BD.

**Tabla 61.-** Caso de prueba: Cambiar estado de proyecto(s) sin seleccionar.

<b>Entrada</b>	El Vicedecano de producción no marca ningún elemento y presiona el botón Cambiar estado.
<b>Resultado</b>	Muestra un mensaje de error especificando que para realizar la acción debe seleccionar al menos un elemento.
<b>Condiciones</b>	-

**Tabla 62.-** Caso de prueba: Eliminar proyecto(s) seleccionado(s).

<b>Entrada</b>	El Vicedecano de producción marca al menos un elemento y presiona el botón Eliminar.
<b>Resultado</b>	Muestra un mensaje preguntando si está seguro de realizar la acción, en caso aprobatorio elimina el/los elemento(s) seleccionado(s).
<b>Condiciones</b>	Proyecto(s) terminado(s) registrado(s) en la BD.

**Tabla 63.-** Caso de prueba: Eliminar proyectos sin seleccionar.

<b>Entrada</b>	El Vicedecano de producción no marca ningún elemento y presiona el botón Eliminar.
<b>Resultado</b>	Muestra un mensaje de error especificando que para realizar la acción debe seleccionar al menos un elemento.
<b>Condiciones</b>	-

**4.2.3.- CU Gestionar usuario.**

**Tabla 64.-** Condiciones de entrada para Gestionar usuario.

Condición de entrada	Casos válidos	Casos no válidos
Nombre y apellidos	Letras.	Campo vacío, otros caracteres diferentes de letras.
Usuario	Primer caracter letra, el resto pueden ser letras, números y guiones bajos.	Campo vacío, primer carácter diferente de letra, el resto caracteres extraños.
Contraseña	Cualquier caracter.	Campo vacío.
Confirmar contraseña	Cualquier caracter.	Campo vacío.
Rol	Escoger alguno.	No escoger ninguno.
Proyecto	Escoger alguno.	No escoger ninguno.

**Tabla 65.-** Caso de prueba: Crear usuario con responsabilidad en facultad entrando datos correctamente.

<b>Entrada</b>	El Vicedecano de producción introduce los datos del usuario a crear: <ul style="list-style-type: none"> <li>- Nombre y apellidos: Risell Ramirez Ramos</li> <li>- Usuario: rramirezr</li> <li>- Contraseña: ***</li> <li>- Confirmar contraseña: ***</li> <li>- Rol: Asesor de calidad</li> </ul>
<b>Resultado</b>	Inserta el usuario en la BD.
<b>Condiciones</b>	No existe usuario registrado en la BD con dicho usuario.

**Tabla 66.-** Caso de prueba: Crear usuario con responsabilidad en proyecto entrando datos correctamente.

<b>Entrada</b>	El Vicedecano de producción introduce los datos del usuario a crear: <ul style="list-style-type: none"> <li>- Nombre y apellidos: Carlos Vazquez Ortiz</li> <li>- Usuario: cvazquez</li> <li>- Contraseña: ***</li> <li>- Confirmar contraseña: ***</li> <li>- Rol: Líder de proyecto</li> <li>- Proyecto: GPI</li> </ul>
<b>Resultado</b>	Inserta el usuario en la BD.
<b>Condiciones</b>	No existe usuario registrado en la BD con dicho usuario.

**Tabla 67.-** Caso de prueba: Crear usuario dejando de entrar al menos un dato.

<b>Entrada</b>	El Vicedecano de producción introduce los datos del usuario a crear dejando al menos un campo vacío: <ul style="list-style-type: none"> <li>- Nombre y apellidos: Carlos Vazquez Ortiz</li> <li>- Usuario: cvazquez</li> <li>- Contraseña: ***</li> <li>- Confirmar contraseña:___</li> <li>- Rol: Líder de proyecto</li> <li>- Proyecto: GPI</li> </ul>
<b>Resultado</b>	Muestra un mensaje de error especificando que los campos con asterisco son de carácter obligatorio.
<b>Condiciones</b>	-

**Tabla 68.-** Caso de prueba: Crear usuario entrando incorrectamente el nombre/usuario/contraseña.

<b>Entrada</b>	El Vicedecano de producción introduce los datos del usuario a crear: <ul style="list-style-type: none"> <li>- Nombre y apellidos: Carlos Vazquez31/Carlos Vazquez/Carlos</li> <li>- Usuario: cvazquez/1cvazquez/cvazquez</li> <li>- Contraseña: ***/***/**</li> <li>- Confirmar contraseña: ***/***/*****</li> <li>- Rol: Líder de proyecto/Líder de proyecto/Líder de proyecto</li> <li>- Proyecto: GPI</li> </ul>
<b>Resultado</b>	Muestra un mensaje de error especificando que el nombre sólo puede contener letras/el usuario debe comenzar con una letra y debe contener sólo letras, número y guiones bajos/las contraseñas no coinciden.
<b>Condiciones</b>	-

**Tabla 69.-** Caso de prueba: Buscar usuarios sin seleccionar un criterio de búsqueda.

<b>Entrada</b>	El Vicedecano de producción no escoge ningún criterio de búsqueda: <ul style="list-style-type: none"> <li>- Nombre y apellidos: __</li> <li>- Usuario: __</li> <li>- Rol: ----</li> </ul>
<b>Resultado</b>	Muestra un mensaje de error especificando que debe seleccionar un criterio de búsqueda.
<b>Condiciones</b>	-

**Tabla 70.-** Caso de prueba: Buscar usuarios seleccionando el criterio de nombre y apellidos/usuario/rol/cualquier combinación de ellos.

<b>Entrada</b>	El Vicedecano de producción escoge el criterio de búsqueda nombre y apellidos/usuario/rol/cualquier combinación de ellos: <ul style="list-style-type: none"> <li>- Nombre y apellidos: Carlos Vazquez/__/__/Carlos Vazquez</li> <li>- Usuario: __/cvazquez/__/cvazquez</li> <li>- Rol: __/__/Líder de proyecto/__</li> </ul>
<b>Resultado</b>	Muestra los datos del usuario que coincida(n) con los valores de los criterios escogidos.
<b>Condiciones</b>	Usuario registrado previamente con las características descritas.

**Tabla 71.-** Caso de prueba: Modificar perfil de varios elementos seleccionados.

<b>Entrada</b>	El Vicedecano de producción marca varios elementos y presiona el botón Modificar perfil.
<b>Resultado</b>	Muestra un mensaje de error especificando que para realizar la acción debe seleccionar sólo un elemento.
<b>Condiciones</b>	-

**Tabla 72.-** Caso de prueba: Modificar perfil de un elemento seleccionado entrando datos correctamente.

<b>Entrada</b>	El Vicedecano de producción marca un elemento y presiona el botón Modificar perfil. Introduce en el formulario los datos que necesite modificar: <ul style="list-style-type: none"> <li>- Nombre y apellidos: Julio M. Vazquez</li> <li>- Contraseña: ***</li> <li>- Confirmar contraseña: ***</li> </ul>
<b>Resultado</b>	Muestra un mensaje preguntando si está seguro de realizar la acción, en caso aprobatorio modifica los datos.

<b>Condiciones</b>	Usuario logueado.
--------------------	-------------------

**Tabla 73.-** Caso de prueba: Modificar perfil de un elemento seleccionado entrando incorrectamente el nombre y apellidos/contraseñas.

<b>Entrada</b>	El Vicedecano de producción marca un elemento y presiona el botón Modificar perfil. Introduce en el formulario los datos que necesite modificar: <ul style="list-style-type: none"> <li>- Nombre y apellidos: Julio3 M. Vazquez/Julio M. Vazquez</li> <li>- Contraseña: ***/***</li> <li>- Confirmar contraseña: ***/*</li> </ul>
<b>Resultado</b>	Muestra un mensaje de error especificando que el nombre debe contener solamente letras/las contraseñas no coinciden.
<b>Condiciones</b>	Usuarios existentes en la BD.

**Tabla 74.-** Caso de prueba: Eliminar usuarios seleccionados.

<b>Entrada</b>	El Vicedecano de producción marca al menos un elemento y presiona el botón Eliminar.
<b>Resultado</b>	Muestra un mensaje preguntando si está seguro de realizar la acción, en caso aprobatorio elimina el/los elemento(s) seleccionado(s).
<b>Condiciones</b>	Usuarios registrados en la BD.

**Tabla 75.-** Caso de prueba: Eliminar usuarios sin seleccionar.

<b>Entrada</b>	El Vicedecano de producción no marca ningún elemento y presiona el botón Eliminar.
<b>Resultado</b>	Muestra un mensaje de error especificando que para realizar la acción debe seleccionar al menos un elemento.
<b>Condiciones</b>	Usuarios registrados en la BD.

#### 4.2.4.- CU Gestionar autenticación.

**Tabla 76.-** Condiciones de entrada para Gestionar autenticación.

Condición de entrada	Casos válidos	Casos no válidos
Usuario	Letras en el primer caracter, letras, número y guiones bajos en el resto.	Campo vacío, otros caracteres extraños.
Contraseña/contraseña anterior/nueva contraseña/confirmar contraseña	Cualquier caracter.	Campo vacío.

**Tabla 77.-** Caso de prueba: Autenticar usuario entrando correctamente los datos.

<b>Entrada</b>	El Usuario con privilegios introduce los datos para autenticarse: <ul style="list-style-type: none"> <li>- Usuario: cvazquez</li> <li>- Contraseña: ***</li> </ul>
<b>Resultado</b>	Autentica al usuario, redireccionando la página a la principal del rol que juega en la facultad o proyecto.
<b>Condiciones</b>	Usuario registrado previamente en la BD.

**Tabla 78.-** Caso de prueba: Autenticar usuario dejando de entrar al menos un dato.

<b>Entrada</b>	El Usuario con privilegios introduce los datos para autenticarse: <ul style="list-style-type: none"> <li>- Usuario: cvazquez</li> <li>- Contraseña: __</li> </ul>
<b>Resultado</b>	Muestra un mensaje de error especificando que ambos campos son de caracter obligatorio para autenticarse.

<b>Condiciones</b>	-
--------------------	---

**Tabla 79.-** Caso de prueba: Autenticar usuario entrando uno no registrado.

<b>Entrada</b>	El usuario introduce los datos para autenticarse: - Usuario: lpozo - Contraseña: ***
<b>Resultado</b>	Muestra un mensaje de error especificando que el usuario no está registrado.
<b>Condiciones</b>	Usuario no registrado en la BD.

**Tabla 80.-** Caso de prueba: Autenticar usuario entrando contraseña incorrecta.

<b>Entrada</b>	El Usuario con privilegios introduce los datos para autenticarse: - Usuario: cvazquez - Contraseña: ****
<b>Resultado</b>	Muestra un mensaje de error especificando que la contraseña es incorrecta.
<b>Condiciones</b>	Usuario registrado en la BD con otra contraseña.

**Tabla 81.-** Caso de prueba: Modificar contraseña entrando correctamente los datos.

<b>Entrada</b>	El Usuario con privilegios introduce los datos para cambiar su contraseña: - Contraseña anterior: *** - Nueva contraseña: **** - Confirmar contraseña: ****
<b>Resultado</b>	Muestra un mensaje preguntando si está seguro de modificar la contraseña, en caso afirmativo realiza la acción.
<b>Condiciones</b>	Usuario logueado con contraseña coincidente con la anterior.

**Tabla 82.-** Caso de prueba: Modificar contraseña dejando de entrar al menos un dato.

<b>Entrada</b>	El Usuario con privilegios introduce los datos para cambiar su contraseña: - Contraseña anterior:___ - Nueva contraseña: **** - Confirmar contraseña: ****
<b>Resultado</b>	Muestra un mensaje de error especificando que los campos con asterisco son de caracter obligatorio.
<b>Condiciones</b>	-

**Tabla 83.-** Caso de prueba: Modificar contraseña entrando erróneamente la anterior.

<b>Entrada</b>	El Usuario con privilegios introduce los datos para cambiar su contraseña: - Contraseña anterior:* - Nueva contraseña: **** - Confirmar contraseña: ****
<b>Resultado</b>	Muestra un mensaje de error especificando que la contraseña anterior no coincide con la registrada.
<b>Condiciones</b>	Contraseña registrada en la BD diferente de la pasada en el formulario.

**Tabla 84.-** Caso de prueba: Modificar contraseña entrando las nuevas distintas.

<b>Entrada</b>	El Usuario con privilegios introduce los datos para cambiar su contraseña: - Contraseña anterior:*** - Nueva contraseña: ***** - Confirmar contraseña: ****
<b>Resultado</b>	Muestra un mensaje de error especificando que las contraseñas no

	coinciden.
<b>Condiciones</b>	Contraseña anterior coincidente con la registrada en la BD.

#### 4.2.5.- CU Actualizar proyecto.

**Tabla 85.-** Condiciones de entrada para Actualizar proyecto.

Condición de entrada	Casos válidos	Casos no válidos
Dirección del Presupuesto, Plan de desarrollo, Lista de riesgos y Plan de capacitación	Letras y caracteres extraños.	Campo vacío.

**Tabla 86.-** Caso de prueba: Subir archivo entrando la dirección correctamente.

<b>Entrada</b>	El Líder de proyecto introduce la dirección del archivo a subir: - Dirección del archivo: C:\Documents and Settings\yvazquez\My Documents\Plan de desarrollo.doc
<b>Resultado</b>	Se guarda en la BD el archivo asociado al proyecto del usuario registrado.
<b>Condiciones</b>	Usuario logueado con un proyecto asignado.

**Tabla 87.-** Caso de prueba: Subir archivo dejando la dirección vacía.

<b>Entrada</b>	El Líder de proyecto no introduce la dirección del archivo a subir: - Dirección del archivo:___
<b>Resultado</b>	Muestra un mensaje de error especificando que la dirección del archivo es de carácter obligatorio.
<b>Condiciones</b>	-

#### 4.2.6.- CU Gestionar entregable.

Para este CU que engloba un conjunto de entregables con diferentes tipos de entradas las pruebas son las mismas, lo que varía es el juego de datos a entrar por lo que los CP se centrarán en la gestión de imágenes, como referencia para el resto de los entregables.

**Tabla 88.-** Condiciones de entrada para Gestionar entregable.

Condición de entrada	Casos válidos	Casos no válidos
Nombre	Letras.	Campo vacío, otros caracteres.
Número	Números.	Campo vacío, otros caracteres.
Entregada	Sí, No	---
Aceptada	Sí, No	---
Resolución	Números	Campo vacío, otros caracteres.

**Tabla 89.-** Caso de prueba: Insertar nuevo entregable entrando los datos correctamente.

<b>Entrada</b>	El Encargado de calidad llena los datos para insertar el entregable: - Nombre: Sol naciente - Número: 31 - Entregada: No - Aceptada: Sí - Resolución: 120
<b>Resultado</b>	Inserta la imagen en la BD.
<b>Condiciones</b>	Usuario logueado con un proyecto asignado.

**Tabla 90.-** Caso de prueba: Insertar nuevo entregable entrando el número/resolución incorrectamente.

<b>Entrada</b>	El Encargado de calidad llena los datos para insertar el entregable: <ul style="list-style-type: none"> <li>- Nombre: Sol naciente</li> <li>- Número: 31a/31</li> <li>- Entregada: No</li> <li>- Aceptada: Sí</li> <li>- Resolución: 120/1a2</li> </ul>
<b>Resultado</b>	Muestra un mensaje de error especificando que el número/resolución sólo debe contener números.
<b>Condiciones</b>	-

**Tabla 91.-** Caso de prueba: Insertar nuevo entregable dejando al menos un campo vacío.

<b>Entrada</b>	El Encargado de calidad llena los datos para insertar el entregable: <ul style="list-style-type: none"> <li>- Nombre: Sol naciente</li> <li>- Número: __</li> <li>- Entregada: ---</li> <li>- Aceptada: Sí</li> <li>- Resolución: 120</li> </ul>
<b>Resultado</b>	Muestra un mensaje de error especificando que los campos con asterisco son de carácter obligatorio.
<b>Condiciones</b>	-

**Tabla 92.-** Caso de prueba: Listar entregables existiendo al menos uno en la BD asociado al proyecto del usuario logueado.

<b>Entrada</b>	El Encargado de calidad presiona el vínculo del entregable que desee gestionar.
<b>Resultado</b>	Muestra un listado con los entregables asociados al proyecto que han sido gestionados.
<b>Condiciones</b>	Entregables registrados en la BD, gestionados previamente.

**Tabla 93.-** Caso de prueba: Listar entregables no existiendo ninguno en la BD asociado al proyecto del usuario logueado.

<b>Entrada</b>	El Encargado de calidad presiona el vínculo del entregable que desee gestionar.
<b>Resultado</b>	Muestra un formulario con los datos del entregable a llenar.
<b>Condiciones</b>	Entregables no registrados en la BD.

**Tabla 94.-** Caso de prueba: Modificar entregable de varios elementos seleccionados.

<b>Entrada</b>	El Encargado de calidad marca varios elementos y presiona el botón Modificar.
<b>Resultado</b>	Muestra un mensaje de error especificando que para realizar la acción debe seleccionar sólo un elemento.
<b>Condiciones</b>	-

**Tabla 95.-** Caso de prueba: Modificar entregable de un elemento seleccionado entrando datos correctamente.

<b>Entrada</b>	El Encargado de calidad marca un elemento y presiona el botón Modificar. En el formulario que se levanta llena los datos necesarios: <ul style="list-style-type: none"> <li>- Nombre: Sol naciente</li> <li>- Número: 31</li> </ul>
----------------	---

	<ul style="list-style-type: none"> <li>- Entregada: Sí</li> <li>- Aceptada: Sí</li> <li>- Resolución: 120</li> </ul>
<b>Resultado</b>	Muestra un mensaje preguntando si está seguro de realizar la acción, en caso afirmativo modifica los datos en la BD.
<b>Condiciones</b>	Entregable registrado en la BD previamente.

**Tabla 96.-** Caso de prueba: Modificar entregable de un elemento seleccionado entrando incorrectamente el número/resolución/dejando al menos un campo vacío.

<b>Entrada</b>	El Encargado de calidad marca un elemento y presiona el botón Modificar. En el formulario que se levanta llena los datos necesarios: <ul style="list-style-type: none"> <li>- Nombre: Sol naciente/Sol naciente/___</li> <li>- Número:31/3a1/31</li> <li>- Entregada: Sí/Sí/___</li> <li>- Aceptada: Sí/Sí/Sí</li> <li>- Resolución: 120/a12/120</li> </ul>
<b>Resultado</b>	Muestra un mensaje de error especificando que el número/resolución debe contener solamente números/los campos con asterisco son de carácter obligatorio.
<b>Condiciones</b>	-

**Tabla 97.-** Caso de prueba: Eliminar entregables seleccionados.

<b>Entrada</b>	El Encargado de calidad marca al menos un elemento y presiona el botón Eliminar.
<b>Resultado</b>	Muestra un mensaje preguntando si está seguro de realizar la acción, en caso aprobatorio elimina el/los elemento(s) seleccionado(s).
<b>Condiciones</b>	Entregable registrados en la BD asociados al proyecto del usuario logueado.

**Tabla 98.-** Caso de prueba: Eliminar entregables sin seleccionar.

<b>Entrada</b>	El Encargado de calidad no marca ningún elemento y presiona el botón Eliminar.
<b>Resultado</b>	Muestra un mensaje de error especificando que para realizar la acción debe seleccionar al menos un elemento.
<b>Condiciones</b>	Entregable registrados en la BD asociados al proyecto del usuario logueado.

#### 4.2.7.- CU Generar reporte.

Para este CU se escogen los parámetros necesarios, que estarán listados en un *select*, y estarán asociados a cada campo de los entregables contemplados en la BD, para conformar el reporte que el Asesor de calidad estime conveniente, pasándosele el valor de cada parámetro y dando la posibilidad de añadir uno nuevo. Los CP definidos en el subepígrafe se harán para un solo parámetro, siendo lo mismo a probar en caso de que se existan más.

**Tabla 99.-** Condiciones de entrada para Generar reporte.

Condición de entrada	Casos válidos	Casos no válidos
----------------------	---------------	------------------

Parámetro	Cualquiera de los elementos listados en el <i>select</i> .	Campo vacío.
Valor del parámetro	Letras y números.	Campo vacío, otros caracteres.

**Tabla 100.-** Caso de prueba: Conformar reporte entrando correctamente valor del parámetro.

<b>Entrada</b>	El Asesor de calidad introduce el dato del parámetro: - Resolución de la imagen: 120
<b>Resultado</b>	Se realizan las consultas necesarias en la BD conformando un archivo PDF con los proyectos que cumplan con el valor pasado para el parámetro.
<b>Condiciones</b>	Proyectos registrados con entregables que cumplan con el parámetro y valor pasados.

**Tabla 101.-** Caso de prueba: Conformar reporte entrando incorrectamente valor del parámetro.

<b>Entrada</b>	El Asesor de calidad introduce el dato del parámetro: - Resolución de la imagen: 12a0/___
<b>Resultado</b>	Muestra un mensaje de error especificando que el valor del parámetro está incorrecto/el campo es de carácter obligatorio.
<b>Condiciones</b>	-

#### 4.2.8.- CU Definir documento.

**Tabla 102.-** Condiciones de entrada para Definir documento.

Condición de entrada	Casos válidos	Casos no válidos
Nombre del archivo	Letras, números y caracteres extraños.	Campo vacío.
Dirección del archivo	Letras, números y caracteres extraños.	Campo vacío.

**Tabla 103.-** Caso de prueba: Subir nuevo archivo entrando datos correctamente.

<b>Entrada</b>	El Asesor de calidad introduce los datos del archivo a subir: - Nombre del archivo: Entregables en la etapa de gestión de medias - Dirección del archivo: C:\Documents and Settings\yvazquezo\My Documents\Entregables en la etapa de gestión de medias.doc
<b>Resultado</b>	Se inserta en la BD el nombre del proyecto.
<b>Condiciones</b>	- El archivo debe tener como extensión .doc o .pdf. - Nombre de archivo no existente.

**Tabla 104.-** Caso de prueba: Subir nuevo archivo entrando datos incorrectamente.

<b>Entrada</b>	El Asesor de calidad introduce los datos del archivo a subir: - Nombre del archivo: __/Etapa de contratación - Dirección del archivo: C:\Documents and Settings\yvazquezo\My Documents\Entregables en la etapa de gestión de medias.doc/ C:\Documents and Settings\yvazquezo\My Documents\Entregables en la etapa de gestión de medias.ppt/___
<b>Resultado</b>	Muestra un mensaje de error especificando que el nombre es de carácter obligatorio/el archivo sólo puede ser .doc o .pdf/la dirección del archivo es de carácter obligatorio.
<b>Condiciones</b>	-

**Tabla 105.-** Caso de prueba: Subir nuevo archivo entrando nombre de archivo registrado previamente.

<b>Entrada</b>	El Asesor de calidad introduce los datos del archivo a subir: <ul style="list-style-type: none"> <li>- Nombre del archivo: Entregables en la etapa de gestión de medias</li> <li>- Dirección del archivo: C:\Documents and Settings\yvazquez\My Documents\Entregables en la etapa de gestión de medias.doc</li> </ul>
<b>Resultado</b>	Muestra un mensaje de error especificando que el nombre del archivo ya está asignado a otro documento.
<b>Condiciones</b>	Nombre de archivo registrado previamente en la BD.

### Conclusiones

En este capítulo se mostraron los diagramas de despliegue e implementación para completar, junto a los diagramas de clases del capítulo anterior, la confección de la propuesta del sistema. Se propusieron además, los CP imprescindibles para probar el correcto funcionamiento del sistema.

## **Conclusiones generales**

La gestión de la calidad a lo largo del proceso de desarrollo del SE juega un papel primordial en el logro de productos con un alto grado de ella. La facultad 8 de la UCI no es ajena a esta conclusión, teniendo como necesidad inmediata desarrollar un sistema que facilite dicho proceso de gestión; razón por la cual surgió la presente investigación con el propósito de iniciar el ciclo de producción del sistema esperado.

Para ello se realizó un estudio de algunos sistemas de gestión de calidad existentes en el mundo, determinándose que ninguno podía adaptarse a las condiciones de la Facultad. Además se hizo una revisión de las posibles tecnologías a utilizar para el futuro desarrollo del sistema, seleccionando las adecuadas para el cumplimiento de los requerimientos y necesidades del sistema; se analizaron y diseñaron las clases que lo conformarán y se implementaron parte de las funcionalidades definidas como significativas.

Por todo lo anterior se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente.

## **Recomendaciones**

Independientemente de que se haya alcanzado el objetivo trazado al inicio de la investigación se recomienda:

- Implementar el resto de las funcionalidades del sistema que no fueron desarrolladas en la primera iteración.
- Desarrollar la ayuda del sistema.
- Facilitar la búsqueda de los entregables, mostrándolos al inicio de cada sesión en un menú.
- Facilitar la navegación en la gestión de los entregables haciendo una paginación de diez elementos.
- Asignar los usuarios a través del dominio UCI.
- Implementar un control de acceso a las páginas por usuario, para complementar la seguridad existente.
- Profundizar la implementación del tratamiento de errores, de manera que el usuario conozca en todo momento el estado del proyecto.

## Referencias bibliográficas

- ÁLVAREZ, D. M. and K. C. RODRÍGUEZ. *Software educativo. Su influencia en la escuela cubana*, 2007, ]. Disponible en: <http://www.monografias.com/trabajos31/software-educativo-cuba/software-educativo-cuba.shtml>
- ANÓNIMO. *About Perl*, 2007a. [2007]. Disponible en: <http://www.perl.org/about.html>
- . *ACID*, 2007b. [2007]. Disponible en: <http://es.wikipedia.org/wiki/ACID>
- . *Aplicación Web*, 2006a. [2006, ]. Disponible en: [http://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_web](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)
- . *Aplicaciones disponibles para su web*, 2006b. [2006,]. Disponible en: <http://www.virtualpyme.com/desarrollo.htm>
- . *Berkeley Software Distribution*, 2007c. [2007, ]. Disponible en: <http://es.wikipedia.org/wiki/BSD>
- . *Comparación de sistemas administradores de bases de datos relacionales*, 2007d. [2007, ]. Disponible en: [http://es.wikipedia.org/wiki/Comparaci%C3%B3n\\_de\\_sistemas\\_administradores\\_de\\_bases\\_de\\_datos\\_relacionales](http://es.wikipedia.org/wiki/Comparaci%C3%B3n_de_sistemas_administradores_de_bases_de_datos_relacionales)
- . *Gestión de la calidad*, 2006c.
- . *Integridad referencial*, 2007e. [2007]. Disponible en: <http://es.wikipedia.org/wiki/integridadreferencial>
- . *Internet Information Server*, 2006d. [2006, ]. Disponible en: [http://es.wikipedia.org/wiki/Internet\\_Information\\_Server](http://es.wikipedia.org/wiki/Internet_Information_Server)
- . *Metodologías de desarrollo de software clases de HI*, 2006e. [2006]. Disponible en:
- . *November 2006 Web Server Survey*, 2006f. [2006, ]. Disponible en: [http://news.netcraft.com/archives/2006/11/01/november\\_2006\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2006/11/01/november_2006_web_server_survey.html)
- . *Oracle*, 2007f. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Oracle>
- . *Perl*, 2007g. [2007, ]. Disponible en: <http://es.wikipedia.org/wiki/Perl>
- . *¿Qué es la calidad?*, 2006g. [2006, ]. Disponible en: <http://calidad.umh.es/curso/concepto.htm>
- . *Sistema de gestión de base de datos*, 2006h. [2006, ]. Disponible en: [http://es.wikipedia.org/wiki/Sistema\\_de\\_gesti%C3%B3n\\_de\\_base\\_de\\_datos](http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos)
- . *Sistema de gestión de contenido*, 2007h. [2007, ]. Disponible en: [http://es.wikipedia.org/wiki/Sistema\\_de\\_gesti%C3%B3n\\_de\\_contenido](http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenido)

- . *Sistema para la evaluación y certificación del software*, 2006i. [2006, ]. Disponible en: [http://www.pcm.gob.pe/PORTAL\\_ONGEI/publica/metodologias/Lib5086/cap0205.htm](http://www.pcm.gob.pe/PORTAL_ONGEI/publica/metodologias/Lib5086/cap0205.htm)
- . *Transacción (base de datos)*, 2007i. [2007]. Disponible en: [http://es.wikipedia.org/wiki/Transacci%C3%B3n\\_de\\_base\\_de\\_datos](http://es.wikipedia.org/wiki/Transacci%C3%B3n_de_base_de_datos)
- . *Unicode*, 2007j. [2007]. Disponible en: <http://es.wikipedia.org/wiki/Unicode>
- ARCHIVOSPC. *EMS SQL Manager 2005 for MySQL 3.6.5.1*, 2006. [2007]. Disponible en: <http://www.archivospc.com/programas/archivos/EMS%20SQL%20Manager%2005%20for%20MySQL.php>
- BARROSO, A. *Sistema Automatizado del Registro Mercantil de Cuba: SAREMC*. Ciudad Habana, CUJAE, 2006. p.
- GARCÍA, E. J. R. *MySQL vs PostgreSQL*, 2007. [2007.]. Disponible en: <http://emilio.aesinformatica.com/2007/03/15/mysql-vs-postgresql>
- GUERVOS, J. J. M. *Introducción a los sistemas de gestión de contenidos*, 2005. [2007, ]. Disponible en: <http://geneura.ugr.es/~jmerelo/tutoriales/cms/>
- JACOBSON, I.; G. BOOCH, et al. *El Proceso Unificado de Desarrollo de Software*. Félix Varela, 2004. p.
- KABIR, M. J. *La biblia de Servidor Apache 2*. Española. 2003. p.
- LARMAN, C. *UML y patrones*. 2004. p.
- LÓPEZ, N. M. S. and Y. S. RAMÍREZ. *Sistema de catalogación y recuperación de recursos de información*. Facultad de Ingeniería Industrial e Ingeniería Informática. Ciudad Habana, Instituto Superior Politécnico José Antonio Echeverría, 2004. p.
- MARQUÉS, P. *El software educativo*, Universidad Autónoma de Barcelona, 2005. p.
- MARTÍN, E. and F. S. PÉREZ. *Sistema ANACONDA para el análisis automático de la calidad del software*.
- MÉXICO, E. *Vocabulario de Negocios*, 2004. [2007,]. Disponible en: <http://www.esmas.com/emprendedores/glosario/400189.html>
- MOMPIÉ, L. G. *Ideas para la concepción de un modelo de evaluación de calidad de software educativo*, Universidad de las Ciencias Informáticas, 2006. p.
- PAZ, L. E. B. *Sistemas gestores de contenidos*, 2006. [2007,]. Disponible en: <http://www.sociedadinformacion.unam.mx/index.jsp?pagina=tecnologias&action=vrArticulo&aid=209>

- PHP, P. *NuSphere PHPEd*, 2007. [2007]. Disponible en:  
[http://www.programacionphp.net/recursos-programas/programas-de-editores-de-php/NuSphere-PHPEd-3.3.3\\_12.html](http://www.programacionphp.net/recursos-programas/programas-de-editores-de-php/NuSphere-PHPEd-3.3.3_12.html)
- PRESSMAN, R. S. *Ingeniería del Software, un enfoque práctico*. Quinta. Félix Varela, 2005. p.
- PROAÑO, D. J. B. *Análisis comparativo de bases de datos de código abierto vs código cerrado (determinación de índices de comparación)*, 2006.
- QUEZADA, C. F. L. *Cuando queda chico un motor base de datos... ¿Qué hacer?*, 2007,]. Disponible en: <http://www.mmug.cl/articulos.php?id=293&tod=1>
- SÁNCHEZ, M. A. M. *Metodologías de desarrollo de software*, 2004. [2006, ]. Disponible en:  
[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)
- SCALONE, F. *Estudio comparativo de los modelos y estándares de calidad del software*. Facultad Regional de Buenos Aires. Buenos Aires, Universidad Tecnológica Nacional, 2006. 488. p.
- TONGA. *Glosario Iberoamericano de Contabilidad de Gestión*, 2000. [2007,]. Disponible en:  
<http://www.observatorio-iberoamericano.org/Revista%20Iberoamericana%20de%20Contab%20Gesti%C3%B3n/N%C2%BA%201/Glosario.htm>
- VIRTUALPYME. *Aplicaciones disponibles para su web*, 2006. [2006,]. Disponible en:  
<http://www.virtualpyme.com/desarrollo.htm>
- WELLING, L. and L. THOMPSON. *Desarrollo Web con PHP y MySQL*. Española. Anaya, 2003. p.

## Bibliografía

- González, S.C. and R.A.H. León, *El paradigma cuantitativo de la investigación científica*. 2002, Ciudad Habana: Eduniv.
- Guerrero, L.A. *CC40B - Análisis y Diseño Orientado a Objetos. Patrones de diseño*. 2004 [cited 2007 abril 25]; Available from: <http://www.dcc.uchile.cl/~luguerre/cc40b/clase14.html>.
- Mañas, J.A. *Prueba de programas*. 1994 [cited 2007 mayo, 20]; Available from: <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>.
- López, N.M.S. and Y.S. Ramírez, *Sistema de catalogación y recuperación de recursos de información*, in *Facultad de Ingeniería Industrial e Ingeniería Informática*. 2004, Instituto Superior Politécnico José Antonio Echeverría: Ciudad Habana.
- Peralta, M. (2004) *Estimación del esfuerzo basada en casos de uso*. Volume, 16.
- Pérez, Y.P., *Metodología para la gestión de contratación en proyectos de desarrollo de Software Educativo*. 2007, Universidad de las Ciencias Informáticas: Ciudad Habana.
- Lascano, E. *Desarrollo de aplicaciones de cliente servidor en dos, tres y n capas, metodologías y tecnologías*. 2007 [cited 2007 mayo 14]; Available from: [http://www.estrategiasysoluciones.com/archivos/aplicaciones2\\_3\\_n\\_capas.pdf](http://www.estrategiasysoluciones.com/archivos/aplicaciones2_3_n_capas.pdf).

## Glosario de términos general

**ACID:** Acrónimo de *Atomicity, Consistency, Isolation and Durability*, Indivisibilidad, Consistencia, Aislamiento y Durabilidad.

**Adabas:** *Adaptable Database System*, base de datos jerárquica de alto rendimiento creada por la empresa alemana Software AG en 1969.

**ADO:** *ActiveX Data Objects*, uno de los mecanismos que usan los programas de computadoras para comunicarse con las bases de datos, darles órdenes y obtener resultados de ellas.

**Aislamiento:** Propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.

**ANSI:** *American National Standards Institute*, Instituto Nacional Estadounidense de Estándares. Organización sin ánimo de lucro que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos. Es miembro de la ISO y la IEC.

**API:** *Application Programming Interface*, Interfaz de Programación de Aplicaciones. Conjunto de especificaciones de comunicación entre componentes software; conjunto de llamadas al sistema que ofrecen acceso a los servicios del sistema desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software.

**Aplicación:** Programa con el que el usuario final interactúa a través de una interfaz, realizando tareas útiles para éste.

**Arquitectura software:** Conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. Establece los fundamentos para que los desarrolladores trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema.

**ACS:** Aseguramiento de la Calidad del Software. Ver SQA.

**Atomicidad:** Ver Indivisibilidad.

**Auditoría:** Proceso de recoger, agrupar y evaluar evidencias para determinar si un Sistema de Información salvaguarda el activo empresarial, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización y utiliza eficientemente los recursos.

**BSD:** *Berkeley Software Distribution*, sistema operativo derivado de *Unix* nacido a partir de las aportaciones realizadas a ese sistema por la Universidad de California en *Berkeley*.

**Capa de presentación:** Presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

**Capa de aplicación:** Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

**Capa de datos:** Es donde residen los datos. Está formada por uno o más gestor de bases de datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

**CGI:** *Common Gateway Interface*, en castellano Interfaz Común de Pasarela, importante tecnología de la *World Wide Web* que permite a un cliente (explorador Web) solicitar datos de un programa ejecutado en un servidor Web. Especifica un estándar para transferir datos entre el cliente y el programa; es un mecanismo de comunicación entre el servidor Web y una aplicación externa.

**Ciclo de vida de desarrollo software:** Cubre cuatro fases en el siguiente orden: inicio, elaboración, construcción y transición.

**Cliente:** Persona, organización o grupo de personas que encarga la construcción de un sistema, ya sea empezando desde cero, o mediante un refinamiento de versiones sucesivas.

**Código fuente abierto:** Software que distribuye de forma libre su código fuente, de manera que los desarrolladores pueden hacer variaciones, mejoras o reutilizarlo en otras aplicaciones. Conocido también como *free software*.

**Componente:** Parte física y reemplazable del sistema que cumple y proporciona la realización de un conjunto de interfaces, ejemplo: ficheros de código fuente, *scripts*, ficheros de código binario, ejecutables y similares.

**Consistencia:** Propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.

**Despliegue:** Cuando varios trabajos relativamente independientes (flujos de control, procesos) se distribuyen entre diferentes dispositivos hardware (procesadores).

**Durabilidad:** Propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

**Estándar ISO:** *International Organization for Standardization*, Organización Internacional para la Estandarización compuesta por representantes de los organismos de normalización nacionales, que produce normas internacionales industriales y comerciales, con el propósito de facilitar el comercio, el intercambio de información y contribuir con unos estándares comunes para el desarrollo y transferencia de tecnologías.

**Factibilidad de un proyecto:** Determina si el proyecto puede ser llevado a cabo con los recursos disponibles y teniendo en cuenta sus riesgos.

**Fase:** Periodo de tiempo entre dos hitos principales en un proceso de desarrollo.

**FTP:** *File Transfer Protocol*, protocolo de transferencia de ficheros entre sistemas conectados a una red TCP basado en la arquitectura cliente/servidor, de manera que un equipo cliente se puede conectar a uno servidor para descargar ficheros desde él o para enviarle sus propios archivos independientemente del sistema operativo utilizado en cada equipo.

**GPL:** *General Public License*, licencia que permite el uso y modificación del código para desarrollar software libre, no propietario.

**Herramienta:** Software que se utiliza para automatizar las actividades definidas en el proceso.

**Herramienta CASE:** *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador. Diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

**Hito:** Punto en el que han de tomarse importantes decisiones de negocio.

**HTML:** *HyperText Markup Language*, lenguaje basado en marcas que indican las características del texto, utilizado para definir documentos de hipertexto en Web.

**HTTP:** *HyperText Transfer Protocol*, protocolo cliente-servidor utilizado para el intercambio de páginas Web, Ver HTML.

**IEC:** *International Electrotechnical Commission*, Comisión Electrotécnica Internacional, organización de normalización en los campos eléctrico, electrónico y tecnologías relacionadas. Numerosas normas se desarrollan conjuntamente con la ISO (normas ISO/IEC).

**IEEE:** *The Institute of Electrical and Electronics Engineers*, el Instituto de Ingenieros Eléctricos y Electrónicos, asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, ingenieros en sistemas e ingenieros en telecomunicación.

**Indicadores de calidad:** Métrica o combinación de ellas que proporciona una visión profunda del proceso del software, del proyecto de software o del producto en sí.

**Indivisibilidad:** Propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias [Dada por la IEEE].

**Ingeniería:** Conjunto de conocimientos y técnicas cuya aplicación permite la utilización racional de los materiales y de los recursos naturales, mediante invenciones, construcciones u otras realizaciones provechosas para el hombre.

**Ingeniería de software:** Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software.

**Integridad referencial:** Propiedad que garantiza que una entidad (fila o registro) siempre se relaciona con otras entidades válidas, es decir, que existen en la base de datos.

Implica que en todo momento dichos datos sean correctos, sin repeticiones innecesarias, datos perdidos y relaciones mal resueltas.

**Interoperabilidad:** Condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos. La interoperabilidad de la Web es una condición necesaria para que los usuarios (humanos o mecánicos) tengan un acceso completo a la información disponible.

**Licencia BSD:** Licencia de software otorgada principalmente para los sistemas BSD. Pertenece al grupo de licencias de Software Libre; 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.

**Macrocomputadora:** *Mainframe*, computadora grande, potente y costosa usada principalmente por una gran compañía para el procesamiento de una gran cantidad de datos; por ejemplo, para el procesamiento de transacciones bancarias.

**Métrica:** Mediciones para el software que se pueden aplicar al proceso de desarrollo con el intento de mejorarlo sobre una base continua. Se utilizan para la estimación, el control de la calidad, la evaluación de productividad y el control de proyectos. Ayuda a evaluar la calidad de los resultados de trabajos técnicos y en la toma de decisiones tácticas a medida que el proyecto evoluciona.

**Microcomputadora:** Computadora que tiene un microprocesador unidad central de procesamiento. Desde el lanzamiento de la computadora personal de IBM el término computadora personal se aplica a las microcomputadoras orientadas a los consumidores.

**Minicomputadora:** Más conocidos como servidores, se desarrollaron en los años 70 y 80; eran el eslabón entre las microcomputadoras de poca potencia y las computadoras centrales de gran capacidad.

**Natural:** Lenguaje de programación de cuarta generación con el que se maneja *Adabas*.

**Navegador Web:** *Navigator* o *browser*. Aplicación que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores Web de todo el mundo a través de Internet.

**Parche:** Actualización de un programa usado para solucionar problemas o la usabilidad de una versión previa de la aplicación. Esto puede incluir cualquier programa desde un procesador de texto, videojuego hasta un sistema operativo.

**Patrón:** Solución común a un problema común de determinado contexto.

**Peso:** Nivel de complejidad asociado al elemento asignado.

**Plataforma de desarrollo:** Entorno común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Comúnmente se encuentra relacionada directamente a un sistema operativo, sin embargo, también es posible encontrarla ligada a una familia de lenguajes de programación o a una API.

**Proceso de desarrollo:** Definición del conjunto completo de actividades necesarias para transformar los requisitos de un usuario en un producto. Plantilla para crear proyectos.

**Producto:** Artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables y documentación.

**ODBC:** *Open DataBase Connectivity*, estándar de acceso a bases de datos desarrollado por *Microsoft Corporation*, con el objetivo de hacer posible el acceso a cualquier dato de cualquier aplicación, sin importar qué SGBD los almacene.

**RAD:** *Rapid Application-Development*, Desarrollo Rápido de Aplicaciones, proceso de desarrollo de software, el método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE. Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.

**Release:** Revisión o versión.

**Requisito:** Condición o capacidad que debe cumplir un sistema.

**Requisito de rendimiento:** Impone condiciones de comportamiento sobre un requisito funcional, como velocidad, rendimiento, mantenimiento, tiempo de respuesta y uso de memoria.

**Requisito funcional:** Especifica una acción que debe ser capaz de realizar el sistema sin considerar las restricciones físicas. Especifica el comportamiento de entrada/salida de un sistema.

**Seguridad:** Asegurar que los recursos del sistema de una organización sean utilizados de la manera que se decidió y que la información que se considera importante no sea fácil de acceder por cualquier persona que no se encuentre acreditada.

**SGBD transaccional:** SGBD capaz de mantener la integridad de los datos, haciendo que las transacciones no puedan finalizar en un estado intermedio; cuando por alguna causa

el sistema debe cancelar la transacción, empieza a deshacer las órdenes ejecutadas hasta dejar la BD en su estado inicial (llamado punto de integridad), como si la orden de la transacción nunca se hubiese realizado.

**SQA:** Patrón de acciones planificado y sistemático que se requiere para asegurar la calidad del software.

**Tablas MyISAM:** Tipo de tabla en un SGBD que no permite transacciones, subconsultas y tampoco ofrece integridad en los datos.

**Tablas InnoDB:** Tipo de tabla en un SGBD que permite transacciones, subconsultas y ofrece integridad en los datos a costa de un alto rendimiento.

**TIC:** Tecnologías que se necesitan para la gestión y transformación de la información, y muy en particular el uso de ordenadores y programas que permiten crear, modificar, almacenar, proteger y recuperar dicha información.

**Transacciones:** Conjunto de órdenes que se ejecutan formando una unidad de trabajo en forma indivisible o atómica.

**UML:** *Unified Modeling Language*, lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

**Unicode:** Estándar industrial con el objetivo de proporcionar el medio por el cual un texto en cualquier forma e idioma pueda ser codificado para el uso informático.

**Usuario:** persona que utiliza o trabaja con algún objeto o que es destinataria de algún servicio público o privado, empresarial o profesional.

**Versión:** Es habitual que una aplicación software sufra modificaciones, mejoras o correcciones. El número de versión suele indicar el avance de los cambios. Suelen ser números correlativos y frecuentemente son dos cifras separadas por un punto.

**XML:** *eXtensible Markup Language*, lenguaje de marcas extensible, metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C). Simplificación y adaptación del SGML que permite definir la gramática de lenguajes específicos. No es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.